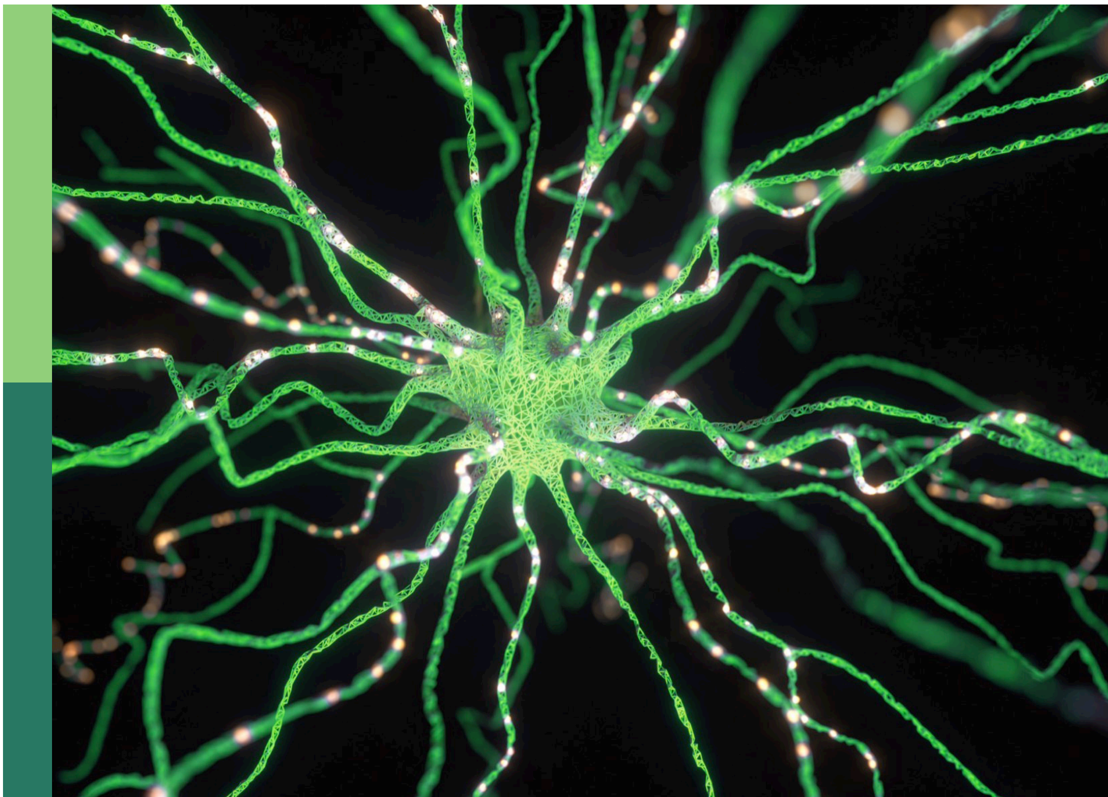
frontiers | Research Topics

Efficient deep neural network for intelligent robot system: Focusing on visual signal processing

Edited by
Xiao Bai, Praveen Kumar Donta, Xin Ning and Weijun Li

Published in
Frontiers in Neurorobotics



ISSN 1664-8714
ISBN 978-2-8325-2269-1
DOI 10.3389/978-2-8325-2269-1

About Frontiers

Frontiers is more than just an open access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers journal series

The Frontiers journal series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the *Frontiers journal series* operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the *Frontiers journals series*: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area.

Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers editorial office: frontiersin.org/about/contact

Efficient deep neural network for intelligent robot system: Focusing on visual signal processing

Topic editors
Xiao Bai — Beihang University, China
Praveen Kumar Donta — Vienna University of Technology, Austria
Xin Ning — Institute of Semiconductors, Chinese Academy of Sciences (CAS), China
Weijun Li — Institute of Semiconductors, Chinese Academy of Sciences (CAS), China

Citation
Bai, X., Donta, P. K., Ning, X., Li, W., eds. (2023). *Efficient deep neural network for intelligent robot system: Focusing on visual signal processing*.
Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-2269-1

Table of contents

TYPE Editorial
PUBLISHED 13 April 2023
DOI 10.3389/fnbot.2023.1191655

OPEN ACCESS

EDITED AND REVIEWED BY
Alois C. Knoll,
Technical University of Munich, Germany

*CORRESPONDENCE
Xin Ning
✉ ningxin@semi.ac.cn

RECEIVED 22 March 2023
ACCEPTED 27 March 2023
PUBLISHED 13 April 2023

CITATION
Bai X, Ning X, Donta PK and Li W (2023) Editorial: Efficient deep neural network for intelligent robot system: Focusing on visual signal processing.
Front. Neurorobot. 17:1191655.
doi: 10.3389/fnbot.2023.1191655

Editorial: Efficient deep neural network for intelligent robot system: Focusing on visual signal processing

Xiao Bai[1], Xin Ning[2]*, Praveen Kumar Donta[3] and Weijun Li[2]

[1]School of Computer Science and Engineering, State Key Laboratory of Software Development Environment, Jiangxi Research Institute, Beihang University, Beijing, China, [2]Institute of Semiconductors, Chinese Academy of Sciences, Beijing, China, [3]Distributed Systems Group, TU Wien (Vienna University of Technology), Vienna, Austria

KEYWORDS

deep neural network, intelligent robot system, image processing, pattern recognition, visual navigation

Editorial on the Research Topic
Efficient deep neural network for intelligent robot system: Focusing on visual signal processing

With the availability of a vast amount of data in the public domain and the advancement of computing power, deep neural network (DNN) models are being increasingly utilized in machine learning tasks related to visual signal processing. However, the extensive use of larger DNN model architectures to enhance learning abilities and performance indicators for various tasks is hindered by their high complexity and computing power requirements. This impedes their efficient deployment on edge platforms and for real-time operations. To overcome these challenges and enable widespread deployment of DNNs in intelligent robot systems, researchers are now focusing on developing efficient DNN models that can improve their training and running speeds.

The purpose of this Special Issue is to collect high-quality articles on the recent development and trend of efficient DNN for intelligent robot system based on visual signal processing, and disseminate the outcomes and products from this topic to a wide range of communities, helping peers and non-expert readers understand the highly efficient design of DNNs. Researchers from all over the world actively participate and contributed a lot of manuscripts. After carefully and professionally reviewing all submissions, 12 high-quality manuscripts are accepted.

One contribution in this topic is about model pruning. Wu et al. propose a novel filter pruning method based on filter similarity to address the limitations of current criterion-based methods used for inference acceleration and hardware compatibility. It achieves significant FLOPs and parameter reduction with no loss in accuracy on different benchmark datasets and network architectures.

Six contributions are about the research of lightweight models and algorithms for classical image processing and computer vision tasks. Lan et al. propose a physical-model guided self-distillation network (PMGSDN) for single image dehazing. Experimental results on synthetic and real-world images show that the proposed method outperforms other methods and achieves high-quality dehazed results with clear textures and good color fidelity. Kumari and Mustafi develop a robust digital watermarking algorithm that uses an informed watermark retrieval architecture, fractional Fourier transform, blind source separation, and a heuristic algorithm. The algorithm's performance is evaluated against common attacks such as JPEG compression and Gaussian noise, and the optimal fractional domain is found using a genetic algorithm. Dai et al. propose SiamHFFT, a lightweight object tracking algorithm capable of handling small targets in complex scenarios. The proposed algorithm uses a hierarchical feature fusion transformer to extract multi-level features from a lightweight backbone, which allows for comprehensive feature representations in an end-to-end manner. The model achieves state-of-the-art results on various benchmark datasets and operates at a rate of 29 FPS on a CPU, making it practical for real-world applications. Zhong et al. propose a lightweight facial expression recognition model based on the Northern Goshawk Optimization algorithm and the bidirectional LSTM neural network, which improves recognition accuracy and can be effectively applied to facial expression recognition. Lu et al. present a facial image inpainting method using a multistage GAN and the global attention mechanism. The proposed method can effectively restore incomplete facial images by enhancing feature mining and semantic expression, using skip connections, encoder-decoder structure, and a local refinement network. Comparative experiments demonstrate that the proposed method generates realistic inpainting results with high PSNR and SSIM, indicating the model's performance and efficiency. Lin et al. address the shortage of boxing coaches in Chinese campuses by proposing a novel solution that employs human pose estimation technology to train interns. Specifically, they develop a model transfer technique that utilizes channel patching to enhance the accuracy of pose key points by an average of 1–20% and 3D accuracies by 0.3–0.5% compared to 2D baselines. The proposed method is not only practical but also effective for boxing pose estimation.

Five contributions focus on the implementation of lightweight models to address other signals. Zheng presents a writing feature abstraction process based on ON-LSTM and attention mechanism for sentiment analysis, addressing the problem of ignoring syntactic and tag semantics information in emotional text feature extraction. The study shows the high application potential of deep learning models for dynamic user sentiment analysis. Wang and Chen investigate teachers' acceptance of robotics education and its relationship to the effectiveness and sustainability of robotics education using the UTAUT model and deep learning algorithms. The study also found that deep learning models such as mDAE and AmDAE reduced training time compared to existing noise-reducing autoencoder models. Teng et al. address the lack of technical and algorithmic support in music therapy for cancer patients and design a neural network robotic system based on breast cancer patients to analyze the effect of music relaxation training on alleviating adverse reactions after chemotherapy. The research provides reference for the next development of neural network robot system in the medical field. Chen and Fan utilize the neural Turing machine model to investigate the tensile properties of metallic materials, and they improve the model to achieve faster and more explicit results. The improved model demonstrates potential for practical applications in the exploration of metal material tensile properties testing technology. Xue et al. develop a modular system for robots to collaborate with humans in using tools. The system uses a multi-layer instance segmentation network to find task-related operating areas and identify tools based on the state of the robot in the collaborative task, generating a state semantic region. The system performs well on an untrained real-world tool dataset and is validated using a robot platform based on Sawyer.

Overall, all papers published in this Special Issue show that efficient DNN for intelligent robot system have developed very fast in recent years. We hope that this topic can provide some references and novel ideas for researchers in this field. It should be emphasized that for such a rapidly developing research field, the work that has been done so far is only a drop in the ocean. The manuscripts we collect this time can only be a small leaf in the Amazon rainforest.

We would like to thank all the authors for their innovative contributions, and all the reviewers for their professional, crucial, yet constructive comments. Also, we wish to express our thanks to Mr Hang Ran, PhD students at Institute of Semiconductors, Chinese Academy of Sciences, for his assistance in this process. Last, we wish to express our gratitude to the editorial team of *Frontiers in Neurorobotics* for their support throughout this venture. We hope you enjoy this collection of papers and that the Special Issue can stimulate further research and development in this area.

Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 6190143).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

TYPE Brief Research Report
PUBLISHED 15 September 2022
DOI 10.3389/fnbot.2022.1000646

OPEN ACCESS

EDITED BY
Xin Ning,
Institute of Semiconductors
(CAS), China

REVIEWED BY
Zhenjun Xu,
Qingdao Agricultural University, China
Jun Tian,
Dongguan University of
Technology, China

*CORRESPONDENCE
Xuewen Chen
chenxw197088@163.com

RECEIVED 22 July 2022
ACCEPTED 22 August 2022
PUBLISHED 15 September 2022

CITATION
Chen X and Fan W (2022) Testing technology for tensile properties of metal materials based on deep learning model.
Front. Neurorobot. 16:1000646.
doi: 10.3389/fnbot.2022.1000646

Testing technology for tensile properties of metal materials based on deep learning model

Xuewen Chen[1]* and Weizhong Fan[2,3]

[1]Guangdong Engineering Polytechnic College, Guangzhou, China, [2]Huajin New Materials Research Institute (Guangzhou) Co., Ltd., Guangzhou, China, [3]Guangdong Hongbang Metal Aluminum Co., Ltd., Guangzhou, China

The properties of metallic materials have been extensively studied, and nowadays the tensile properties testing techniques of metallic materials still have not found a suitable research method. In this paper, the neural Turing machine model is first applied to explore the tensile properties of metallic materials and its usability is demonstrated. Then the neural Turing machine model was improved. The model is then improved so that the required results can be obtained faster and more explicitly. Based on the improved Neural Turing Machine model in the exploration of tensile properties of metal materials, it was found that both H-NTM and AH-NTM have less training time than NTM. A-NTM takes more training time than AH-NTM. The improvement reduces the training time of the model. In replication, addition, and multiplication, the training time is reduced by 6.0, 8.8, and 7.3%, respectively. When the indentation interval is 0.5–0.7 mm, the error of the initial indentation data is large. The error of the tensile properties of the material obtained after removing the data at this time is significantly reduced. When the indentation interval is 0.8–1.5 mm, the stress is closer to the real value of tensile test yield strength 219.9 Mpa and tensile test tensile strength 258.8 Mpa. this paper will improve the neural Turing machine model in the exploration of metal material tensile properties testing technology has some application value.

KEYWORDS

neural Turing machine, metallic material, tensile experiment, inspection technique, hard sigmoid

Introduction

As a material, metallic materials are widely used in human production, life and social development (Stock et al., 2018). Metallic materials have many properties, such as high elasticity, high toughness, and high hardness (Kumar et al., 2020). In the metal industry, metallic materials are usually divided into pure metals and alloys (Suryanarayana, 2019). The physical properties of metal materials are usually tested when making the corresponding materials and equipment. The physical testing of metals is carried out according to industry standards and using scientific methods. Therefore, it is necessary to strengthen the research on the physical properties testing technology of metals and to improve the corresponding technical measures (Xu et al., 2021).

The effects on the properties of metallic materials are usually reflected within the metallic material. When people process metal materials, the metal material is easily affected by the tensile speed. Generally metal inclusions, metal crystals and other impurities are present inside the metallic material. These magazines lead to problems such as crystal misalignment and poor bonding within the metallic material (Ford et al., 2019). Metallic materials generally have a relatively consistent overall performance, but the tensile properties of metallic materials will be affected when external elastic deformation or plastic deformation occurs during processing (Khalid et al., 2022). According to Regan et al. (2020), who studied plastic materials, it was found that plastic deformation of materials can be accomplished by stretching. The external processing can cause the relative sliding of the metal material beyond the slip threshold, and this phenomenon will cause the crystalline and crystallographic motion of the metal crystal. This process will have a velocity of motion. When the metal material is stretched, the strength will also increase when the stretching temperature increases, and there will be a time lag in the stretching process. At a slow rate of stretching, the technical material can withstand a tensile force of 200 kN.

When the stretching speed is increased, applying 200 kN tension to the metal material will cause dislocation intensive reduction of the material tensile properties and fracture of the metal material. Yuan and Fan (2019) found that a reasonable choice of speed and pressure is required when stretching metal materials. When the metal is stretched, this operation needs to ensure that the metal crystal slip is produced and the tensile properties of the material are taken into account. And to avoid the fracture of the metal material during the operation. The properties of metallic materials have been widely studied, but for their tensile properties testing techniques, they are currently difficult to find a suitable method for researchers to explore simple, accurate and fast testing techniques for the tensile properties of metallic materials. In today's era of exponential growth of data, the value laws behind the data are often buried under the vast amount of information. How to uncover the potential value through the surface phenomenon and exploit it has become the focus of current technology research (Bai et al., 2022). Yao and Guan (2018) stated that natural language processing is a popular area of data research. In terms of algorithm implementation, machine learning methods have received wide attention from scholars both at home and abroad. Neural networks have features such as automatic feature extraction and strong description ability (Ning et al., 2022). Among many machine learning methods, neural networks have become a dark horse in the machine learning community. Neural networks have made breakthroughs in many research areas. In the research of Neural Turing Machine (NTM), a Neural Turing Machine (NTM) is a kind of neural network with Turing-complete properties. It has the ability to fit functions and can theoretically implement any function. According to Gangal et al. (2021), it was found that the most important difference between NTM and physical Turing machines is that a Neural Turing Machine is an algorithm that can pass gradients backwards. The physical concept of a Turing machine uses the 0 or 1 representation of data in a computer to compute all logical functions (Malekmohamadi Faradonbe et al., 2020). The same feature as all algorithms is that neural Turing machines, like all neural network algorithms, use mainly real numbers (Boce et al., 2022). Neural Turing machines use activation functions with smoother function images to make the neural network properties appear continuously non-linear. Such non-linear neural networks composed of real numbers are easier to train (Huang et al., 2020). Neural Turing machines combine physical Turing machine ideas and smooth activation functions to perform the operations associated in physical Turing machines. Another difference from physical Turing machines is that physical Turing machines read the instructions to be executed continuously in one direction in a sequential manner (Mühlhoff, 2020). In contrast, during the addressing of a neural Turing machine, the neural Turing machine can computationally generate a displacement that shifts the center of gravity of the current attention to the left or right, rather than simply to one direction (Faradonbeh and Safi-Esfahani, 2019). The focus of NTM is on the management of external memory. NTM extends the functionality of standard controllers by reading and writing external memory as a result of addressing. Thus, they can make the NTM implement the memory management function. According to Sharma et al. (2020), it was found that the addressing mechanism of NTM also makes the controller in NTM to generate certain attention. Thus, NTM can improve the model's ability to process sequences. Deep Reinforcement Leaning (DRL) is used to solve the problem of too many states in reinforcement learning (Wang et al., 2022). Deep reinforcement learning methods construct a function with parameters to fit the value assessment of state actions (Quan et al., 2020). Deep reinforcement learning obtains action chains with corresponding values by trying different strategies, which in turn can tune the parameters of the value function. Thus, they can make the prediction of the value function converge to the actual value (Bai et al., 2021). It has also become a trend to add deep learning to NTM as the optimal strategy can be obtained through the value function (Wang et al., 2021). In the study by Gross et al. (2021), this study used the NTM mechanism to improve the network model structure. A data copy experiment and a data repetitive copy experiment were designed in the study. The effectiveness of the attention mechanism generated by NTM was verified from the experimental results. The metal material tensile property testing technique has been widely explored, so combining neural network applied to metal material tensile property testing technique is rarely studied and the applicability study under this combined neural Turing model is almost absent.

In summary, this problem is explored for the tensile properties testing technique of metal materials. In this study, an improved neural Turing machine model is proposed. The model uses the Hard sigmoid function instead of the sigmoid activation function in NTM. This approach makes the model computationally simple and easy to optimize. This approach ensures that the core structure of the NTM remains unchanged, while reducing the computational effort of the model and speeding up the model training. In this paper, the improved neural Turing machine model is applied to the problem of exploring the tensile properties testing technology for metal materials. In the study, it is found that the improved neural Turing machine can reduce the training time of the model. When the indentation interval of metal material is 0.5–0.7 mm, the error of the tensile property results obtained after removing the initial indentation data is significantly reduced and is closer to the real value. When the indentation interval is 0.8–1.5 mm, the accuracy of fitting the results using the default range is higher.

System model

Introduction to the neural turing machine model and formulas

Introduction of neural turing machine

A neural Turing machine is a neural network architecture with the addition of an external storage matrix. The external storage matrix enhances the neural network's ability to remember long input sequences, forming an attention mechanism similar to the Seq2Seq model. This external memory-based architecture is consistent with computer Turing machines. Only in contrast to computer Turing machines, an end-to-end microscopic neural network model of NTM can be trained using gradient descent method for network modeling.

The main components of the NTM are the controller, the read/write side, and an external memory (Urien, 2019). The controller in the NTM is equivalent to the CPU in a computer, and the external memory is equivalent to the memory of a computer. The read/write side is equivalent to the IO device of the computer. The controller modifies and reads the memory blocks through the read/write side. During the operation of the computer, the CPU addresses the data according to the control signals from the controller, and the CPU determines where in the memory to read and write the data information. Unlike actual machines, there is no concept of bootability for computer operations on memory. In NTM, all read and write operations to the memory block matrix are derivable (Vishwakarma and Lee, 2018).

The output of the NTM controller controls the entire workflow of the NTM. The implementation of the controller is a neural network. This means that it can be a recurrent neural network. It can also be a fully connected or convolutional network. It is the neural network controller that interacts with the entire system input and output. The read and write sides of the NTM calculate the weights of each vector in the external memory matrix for the current state based on the control signals from the controller. The values of the memory matrix in the NTM are affected by all the inputs up to the current moment. the memory matrix in the NTM is a real matrix. the memory matrix in the NTM is the object of direct operations by the read and write side. The process of reading and writing against the memory matrix in the sequence model is represented as an attention mechanism.

Formulation of the neural turing machine

$M^{(t)}$ denotes the memory matrix of size N × E at moment t, where N denotes the number of memory cells and E denotes the size of each memory cell. $w^{(t)}$ denotes the weight vector output through the read head at the moment of t. $w^{(t)}$ whose the i-th dimensional element $w_i^{(t)}$ represents the weight occupied by the i-th memory cell and satisfies the following constraint.

$$\sum_{i=1}^{N} w_i^{(t)} = 1, 0 \leq w_i^{(t)} \leq 1, i = 1, 2, \ldots N \tag{1}$$

Then the reading vector $r^{(t)}$ at moment t is calculated according to Equation (2).

$$r^{(t)} = w^{(t)} M^{(t)} \tag{2}$$

At the moment of t, the write head outputs the weight vector $w^{(t)}$, the E dimensional elimination vector $e^{(t)}$ and the E dimensional $a^{(t)}$. $e^{(t)}$ each element belongs to the interval (0, 1). Then the value of the memory matrix can be calculated according to Equations (3)–(5) as follows:

$$\begin{aligned} e^{(t)} &= \sigma(W^e h^{(t)} + b^e) && (3) \\ a^{(t)} &= W^a h^{(t)} + b^a && (4) \\ M^{(t)} &= M^{(t-1)} |1 - w^{(t)} (e^{(t)})^T| + w^{(t)} (a^{(t)})^T && (5) \end{aligned}$$

where, 1 in Equation (5) denotes an all-1 matrix of size N × E, denotes the output of the controller at the moment of t, W^e, b^e, W^a, and b^a are the weights and biases corresponding to the elimination vector and the additive vector, respectively. From Equation (5), it can be seen that each element of the memory matrix is reset to 0 when each element of $e^{(t)}$ and $w^{(t)}$is equal to 1, and then a new memory vector is written.

When each element of $e^{(t)}$ and $w^{(t)}$ is equal to 0, each element of the memory matrix remains unchanged.

The addressing mechanism based on location addressing is introduced into NTM. In this paper, the addressing mechanism of NTM, that is, the weight vector occupied by the ith memory unit, is summarized as the following four formulas:

$$\begin{aligned} C_i^{(t)} &= \frac{\exp\left(\beta^{(t)} k(k^{(t)}, M_i^{(t)})\right)}{\sum_j \exp\left(\beta^{(t)} k(k^{(t)}, M_i^{(t)})\right)} && (6)\\ G_i^{(t)} &= g^{(t)} C_i^{(t)} + \left(1 - g^{(t)}\right) w_i^{(t-1)} && (7)\\ \tilde{w}_i^{(t)} &= \sum_{j=0}^{N-1} G_j^{(t)} s_{i-j}^{(t)} && (8)\\ w_i^{(t)} &= \frac{\tilde{w}_i^{(t)\gamma(t)}}{\sum_j \tilde{w}_j^{(t)\gamma(t)}} && (9) \end{aligned}$$

The *K* function in Equation (6) represents the cosine similarity function.

$$k(u, v) = \frac{u \cdot v}{||u||||v||} \tag{10}$$

Equations (6) and (9) involves the five parameters $k^{(t)}$, $\beta^{(t)}$, $g^{(t)}$, $s^{(t)}$,$\gamma^{(t)}$ and according to the previous section, have their specific physical meaning. In the structure of NTM, they each correspond to a single layer of neural networks whose inputs are controller outputs $h^{(t)}$. Where $k^{(t)}$ corresponds to a linear activation function, $\beta^{(t)}$, $g^{(t)}$, $s^{(t)}$, $\gamma^{(t)}$ corresponding to the activation functions 1+ReLU, sigmoid, softmax, and 1+ReLU, respectively. the following equation gives the definition of these five parameters:

$$\begin{aligned} k^{(t)} &= W^k h^{(t)} + b^k && (11)\\ \beta^{(t)} &= 1 + \text{ReLU}\left(W^\beta h^{(t)} + b^\beta\right) && (12)\\ g^{(t)} &= \text{sigmoid}\left(W^g h^{(t)} + b^g\right) && (13)\\ s^{(t)} &= \text{softmax}\left(W^s h^{(t)} + b^s\right) && (14)\\ \gamma^{(t)} &= 1 + \text{ReLU}\left(W^\gamma h^{(t)} + b^\gamma\right) && (15) \end{aligned}$$

shows the output of the time-step t controller, and the W and b appearing in Equation are the weights and biases corresponding to each parameter, respectively.

Improvement of neural Turing machine

In order to speed up the training of the model, this paper uses hard sigmoid function instead of sigmoid activation function in NTM. Hard sigmoid function has the main properties of sigmoid activation function. Hard sigmoid function also has the characteristics of simple calculation and easy optimization of relu activation function. The hard sigmoid function is used as the activation function, which ensures that the core structure of NTM does not change. It also reduces the calculation of the model.

The relu activation function is defined as follows:

$$g\,(x) = \max\,(0, x) \tag{16}$$

Relu is a piecewise linear function. When x is a negative number, G (x) is equal to zero; When x is >0, G (x) is equal to X.

Sigmoid activation function. The activation function related to this article is sigmoid function. It is defined as follows:

$$\sigma\,(x) = \frac{1}{1 + e^{-x}} \tag{17}$$

Its characteristics are: the value range of the function is an interval (0, 1), the function is derivable, and the exp function and division must be calculated for both the function value and the derivative value. Compared with linear function, sigmoid activation function has a huge amount of computation and saturation at both ends of the function.

According to literature (Mao, 2020), this paper notes that Kaiser et al. introduced the truncation mechanism in N-GPUs. Gate truncation mechanism used by Kaiser means that in the gate mechanism of N-GPUs, the sigmoid function is replaced by the function defined in Equation (18).

$$\sigma'(x) = \max(0, \min(1, 1.2\sigma(x) - 0.1)) \tag{18}$$

Hard sigmoid function is a piecewise linear approximation function of sigmoid function. Its definition is shown in Equation (19). According to the introduction of literature (Darabi et al., 2018), this definition comes from courbariaux.

$$\sigma'(x) = \max(0, \min(1, (x + 1)/2)) \tag{19}$$

Introduction and formula of tensile properties of metal materials

Tensile properties are one of the important mechanical properties of metallic materials, the yield strength, tensile strength, elongation and sectional shrinkage of metals can be measured by tensile tests and other performance indicators. The relationship between elongation and section shrinkage during the uniform deformation phase has been derived in previous studies under the assumption of constant volume; after necking, the "true stress-strain" curve is also plotted under incorrect strain values.

The stress-strain relationship of the hardening section of power hardening and line hardening metal materials is shown in Equations (20) and (21), respectively.

$$\sigma_R = K\varepsilon_R^n \tag{20}$$

K is the strength coefficient of power hardening material, which is fitted by the least square method.

$$\sigma_R = E_2\varepsilon + b \tag{21}$$

E_2 is the tangent modulus of linear hardening material, and b is the material constant. For metallic materials satisfying the linear elastic power hardening model, the yield strength is the intersection of the elastic deformation line and the plastic deformation curve. Generally, it is determined by the offset of 0.2% of the elastic segment. The tensile strength is calculated by the concept of tensile instability:

$$\sigma_y = E\left(\varepsilon_y - 0.002\right) = K\varepsilon_y^n \tag{22}$$

$$\sigma_u = k\left(\frac{n}{e}\right)^n \tag{23}$$

In the formula, the elastic modulus E has been calculated by indentation contact stiffness, contact projection circle area and other indentation parameters before calculation. E is the base of natural logarithm. For metal materials that meet the line hardening model, the yield strength can be obtained by Meyer's law. The tensile strength is obtained according to the concepts of volume incompressibility and instability during tensile deformation (Ye et al., 2020).

$$\sigma_y = \beta_m A \tag{24}$$

$$\sigma_u = \frac{E_2}{\exp[(E_2 - b)/E_2]} \tag{25}$$

A is the Meyer index and the material yield parameter, respectively, which is obtained by non-linear regression of the Meyer equation. m is the material constant, which is related to the type of metal material. For carbon steel and austenitic stainless steel, this value is usually taken as 0.2285 and 0.1910.

Analysis and discussion

Analysis and discussion of neural turing machine

The experiments introduced in this section compare the performance of different models in algorithm learning tasks, including RNN, LSTM, GRU, and NTM. The experimental tasks include replication, addition and multiplication. Each task trains a model independently. In the copy task, the model trained 50,000 Batches; In the addition task, the model trained 150,000 Batches; In the multiplication task, the model trained 300,000 batches. The performance differences of different models are compared in Figure 1.

The training time of different models is compared in Figures 1A–C. In Figures 1A–C, from top to bottom are copy, addition and multiplication, respectively. The models include RNN, LSTM, GRU, and NTM. The unit of time is minutes. Pink represents RNN, orange represents LSTM, blue represents GRU, and purple represents NTM. It can be seen from the figure that training NTM takes the longest time and training RNN takes the least time. The training time increases in the order of RNN, GRO, LSTM and NTM. In the replication task, it takes 28 min to train RNN, 46 min to train Gru, 49 min to train LSTM, and 101 min to train NTM; In addition to this task, it takes 267 min to train RNN, 188 min to train Gru, 200 min to train LSTM, and 658 min to train NTM; In multiplication tasks, it takes 140 min to train RNN, 374 min to train GRU, 439 min to train LSTM and 1,829 min to train NTM. Although the accuracy of NTM on the test set is higher than GRU, LSTM and NTM, the performance of the model is still relatively poor.

NTM has a long training time. In the replication task, the time required to train NTM is between 2 and 5 times that of other models. In addition to task, the time needed to train NTM is between 3 and 6 times that of other models. In the multiplication task, the time required to train NTM is between 4 and 9 times that of other models.

Improved neural turing machine analysis discussion

Figures 1D–F show the comparison of the training time of different models. four network structures are involved in this experiment as follows: (1) ordinary NTM. (2) NTM using Hard sigmoid activation function, which is referred to as H-NTM in the experiment. (3) NTM trained using adaptive curriculum learning strategy based on adaptive curriculum scaling, referred to as A-NTM. (4) H-NTM trained using adaptive curriculum-based scaling of curriculum learning strategies, referred to as AH-NTM. In Figures 1D–F, from top to bottom, are replication, addition, and multiplication, respectively. The models include AH-NTM, A-NTM, H-NTM, and NTM. Time units are minutes. Orange represents AH-NTM, yellow represents A-NTM, cyan represents NTM, and purple represents H-NTM. As can be seen from the figure, the time required to train NTM and A-NTM is very close, and the time required to train H-NTM and AH-NTM is very close. In the replication task, NTM takes 101 min; A-NTM takes 103 min; H-NTM takes 93 min; and AH-NTM takes 95 min. In the addition task, NTM takes 658 min; A-NTM both take 657 min; H-NTM takes 604 min and AH-NTM takes

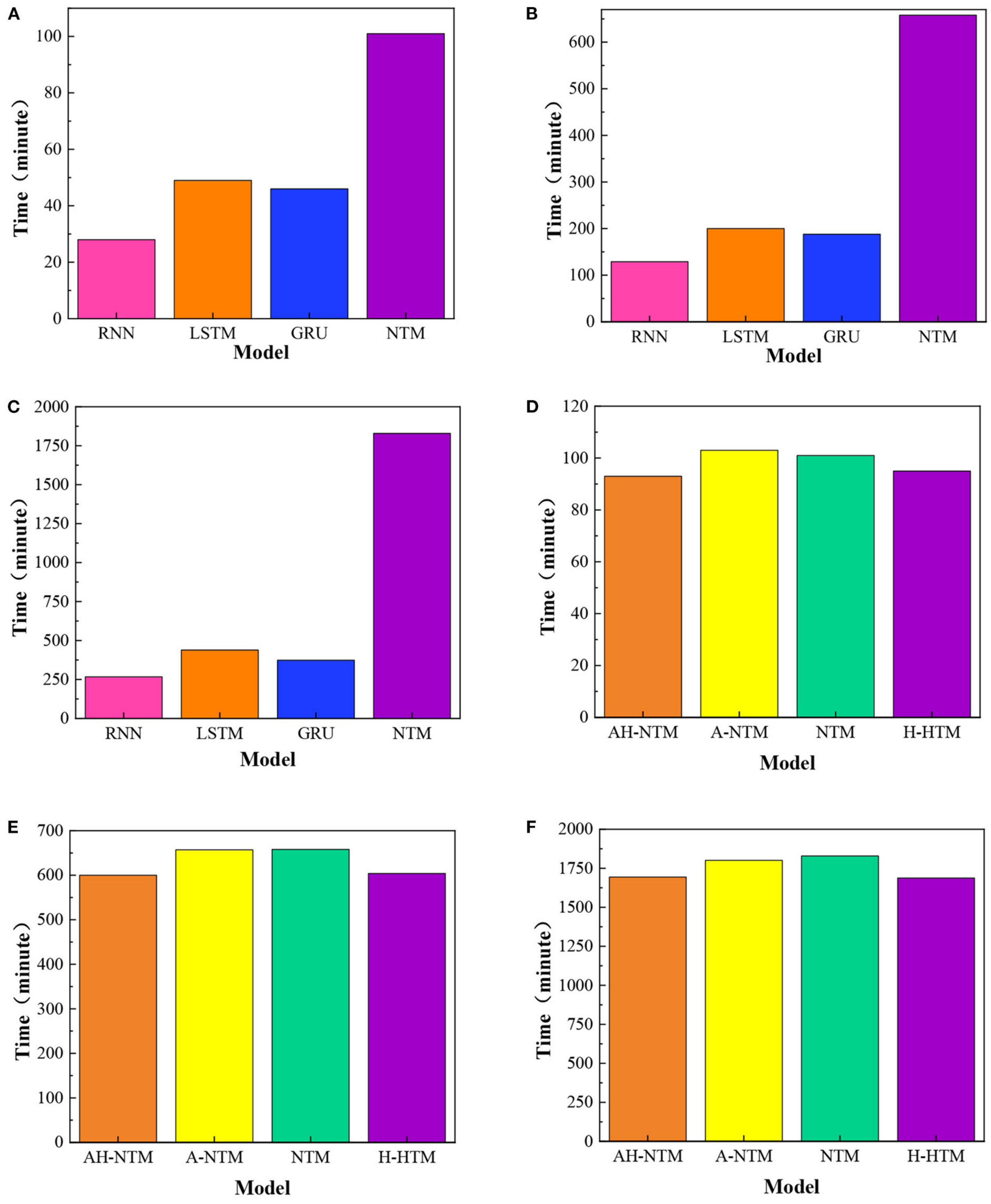


FIGURE 1
Comparison of training time of different models, **(A)** replication; **(B)** addition; **(C)** multiplication; **(D)** replication; **(E)** addition; **(F)** multiplication.

600 min. In the multiplication task, NTM requires 1,829 min; both A-NTM require 1,801 min; H-NTM requires 1,688 min, and AH-NTM requires 1,694 min.

Figures 1D–F shows that the training time of H-NTM and AH-NTM is less than that of NTM and A-NTM takes more training time than AH-NTM. The specific figures are as follows:

in the copying task, the training time of H-NTM is 8.0% less than that of NTM; the training time of AH-NTM is 6.0% less than that of NTM. In the addition task, the training time of H-NTM was reduced by 8.2% compared to NTM, and the training time of AH-NTM was reduced by 8.8% compared to NTM. In the multiplication task, the training time of H-NTM was reduced by 7.7% compared to NTM, and the training time of AH-NTM was reduced by 7.3% compared to NTM. In summary, the training time of the model is reduced by using the Hard sigmoid activation function instead of the sigmoid activation function in the NTM. The combination of the Hard sigmoid activation function and the adaptive course scaling-based course learning strategy reduces the training time of the model.

Analysis and discussion of tensile properties of metal materials

In this experiment, the indentation interval is small and taken as 0.5, 0.55, 0.6, and 0.65 mm variables for the experimental study. The results of the second indentation test in this study were mainly influenced by the raised material surface. Its initial indentation load value is much larger than the true value, and the calculated distribution pattern of the characterized stress-strain data points deviates from the power-law intrinsic structure relationship of the aluminum alloy material. In order to improve the accuracy of the fitting results of this experiment. In this study, the results corresponding to the small initial indentation depth can be excluded. On this basis, the data points in the latter part of the study were selected to be more reasonably distributed for fitting. The smaller number of test points means that more initial data points are eliminated. For example, the 14 points represent the remaining 14 data sets after removing the stress-strain points obtained at the indentation depth of 10 μm. The indentation results at 0.5 mm indentation interval were used for the experimental analysis. The influence of the data selection range on the fitting and calculation results was investigated. The experimental study found that the most significant effect was caused by the first indentation at this working condition. As shown in Figure 2A, the stress of tensile strength and yield strength showed an increasing trend with the increase of fitting data points. As the number of points increases, the further away from the true value of the test. The true value of tensile test yield strength stress is 219.9 Mpa. The true value of the tensile strength stress of another group of tensile tests is 258.8 Mpa. This indicates that the indentation results with a small indentation depth are the main factor affecting the final calculation results. However, the number of test fitting points should be >10.

The yield strength obtained from this experiment using a 10-point fit was 4.32%. The relative error of the tensile strength was 4.17%. The relative errors of the initial results in the experiment were 15.17 and 14.68%. The comparison of the initial results with the 10-point fit indicates that reducing the choice of fitted data points in this case can significantly improve the accuracy of the calculated results. The 10-point fits were performed separately for the indentation results at different indentation intervals in the experiments. The variations of the calculated yield strength and tensile strength results are shown in Figure 2B. From the results, it can be obtained that the error of the initial indentation data is larger when the indentation interval is 0.5–0.7 mm. The error of the material tensile property results obtained after removing this section of data is significantly reduced and is closer to the true value. The data obtained were closer to the true value of the tensile test yield strength of 219.9 Mpa. another set of data was closer to the tensile test tensile strength of 258.8 Mpa. when the indentation interval was 0.8–1.5 mm, the accuracy of fitting the results using the default range was higher. The test stress data is infinitely closer to the true value with less error. Reducing the fitted data points will increase the calculated value of indentation and reduce its accuracy to some extent.

Conclusion

Nowadays, metal material properties have become a hot research problem. Based on the assistance of neural Turing machine model, an improved neural Turing machine model is proposed in this paper. The model is applied to the exploration of tensile properties testing techniques for metallic materials. The model allows us to get the required results faster and more explicitly. It is found that the accuracy of fitting the results using the default range is higher when the press-in interval is 0.8 mm-1.5 mm. The specific findings of this study are as follows.

(1) In the experiments of neural Turing machine, the training time of four different models, RNN, LSTM, GRU and NTM, was compared for three different experimental tasks of replication, addition and multiplication. The analysis reveals that the training time of NTM is longer. In the replication task, the time required to train NTM is 2–5 times longer than the other models, respectively. In the addition task, it took 3–6 times as long to train as the others. In the multiplication task, it took 4–9 times longer to train than the others.
(2) In the experiments of the improved neural Turing machine, the training time was compared for four different models, NTM, H-NTM, A-NTM, and AH-NTM, for three different experimental tasks of replication, addition, and multiplication. The analysis shows the training time of H-NTM and AH-NTM is less than that of NTM. The training time of A-NTM is more than that of AH-NTM. The improvement of them reduces the training time of the models. In replication, the training time of AH-NTM is reduced by 6.0% compared to NTM, respectively. In

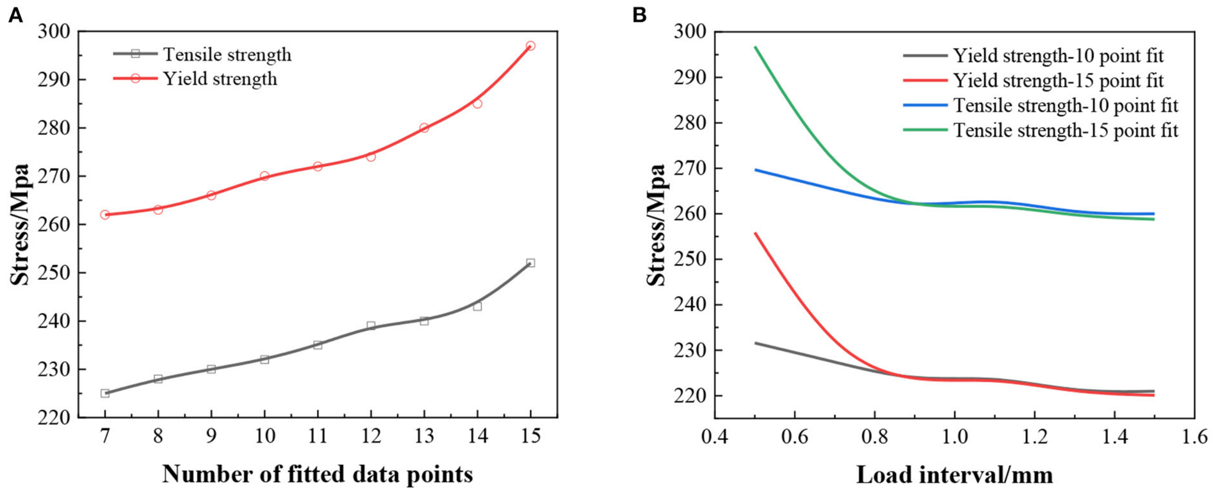


FIGURE 2
Calculated results of metal tensile properties, **(A)** material tensile properties for different data ranges at 0.5 mm press-in interval; **(B)** re-choice of fitted data.

addition, its training time was reduced by 8.8%, and in multiplication, its training time was reduced by 7.3%.

(3) When the indentation interval is 0.5–0.7 mm, the error of the initial indentation data is larger. This value is closer to the real value of the tensile test yield strength 219.9 Mpa and the real value of tensile test tensile strength 258.8 Mpa. When the indentation interval is 0.8–1.5 mm, the accuracy of fitting the results with the default range is higher, whose values are infinitely close to the true values. Reducing the number of fitted data points will increase the calculated value of indentation and reduce its accuracy to some extent.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

XC and WF validated their proposed ideas by designing experiments and analyzing the results in detail and then completed the paper writing. All authors read and approved the final draft.

Conflict of interest

Author WF was employed by Huajin New Materials Research Institute (Guangzhou) Co., Ltd. and Guangdong Hongbang Metal Aluminum Co., Ltd.

The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bai, X., Wang, X., Liu, X., Liu, Q., Song, J., Sebe, N., et al. (2021). Explainable deep learning for efficient and robust pattern recognition: a survey of recent developments. *Pattern Recogn.* 120, 108102. doi: 10.1016/j.patcog.2021.108102

Bai, X., Zhou, J., Ning, X., and Wang, C. (2022). *3D Data Computation and Visualization*. Elsevier, 102169.

Boce, H., Fono, A., and Kutyniok, G. (2022). Inverse problems are solvable on real number signal processing hardware. arXiv preprint.

Darabi, S., Belbahri, M., Courbariaux, M., and Nia, V. P. (2018). *Bnn+: Improved Binary Network Training*. Available online at: openreview.net

Faradonbeh, S. M., and Safi-Esfahani, F. (2019). A review on neural turing machine. *arXiv preprint*.

Ford, M. J., Ambulo, C. P., Kent, T. A., Markvicka, E. J., Pan, C., Malen, J., et al. (2019). A multifunctional shape-morphing elastomer with liquid metal inclusions. *Proc. Natl. Acad. Sci.* 116, 21438–21444. doi: 10.1073/pnas.1911021116

Gangal, A., Kumar, P., Kumari, S., Li, W., Bai, X., and Wang, Y. (2021). Neural Computing. *arXiv preprint*.

Gross, J. E., Caceres, S., Poch, K., Hasan, N. A., Davidson, R. M., Epperson, L. E., et al. (2021). Healthcare-associated links in transmission of nontuberculous mycobacteria among people with cystic fibrosis (HALT NTM) study: rationale and study design. *PLoS ONE* 16, e0261628. doi: 10.1371/journal.pone.0261628

Huang, Z., Zhu, X., Ding, M., and Zhang, X. (2020). Medical image classification using a light-weighted hybrid neural network based on PCANet and DenseNet. *Ieee Access* 8, 24697–24712. doi: 10.1109/ACCESS.2020.2971225

Khalid, M. Y., Arif, Z. U., Ahmed, W., and Arshad, H. (2022). Evaluation of tensile properties of fiber metal laminates under different strain rates. *Proc. Inst. Mech. Eng. E J. Proc. Mech. Eng.* 236, 556–564. doi: 10.1177/09544089211053063

Kumar, J., Singh, D., Kalsi, N. S., Sharma, S., Pruncu, C. I., Pimenov, D. Y., et al. (2020). Comparative study on the mechanical, tribological, morphological and structural properties of vortex casting processed, Al–SiC–Cr hybrid metal matrix composites for high strength wear-resistant applications: fabrication and characterizations. *J. Mater. Res. Technol.* 9, 13607–13615. doi: 10.1016/j.jmrt.2020.10.001

Malekmohamadi Faradonbe, S., Safi-Esfahani, F., and Karimian-Kelishadrokhi, M. A. (2020). review on neural turing machine (NTM). *SN Comput. Sci.* 1, 1–23. doi: 10.1007/s42979-020-00341-6

Mao, J. (2020). *Efficient Neural Network Based Systems on Mobile and Cloud Platforms*. Duke University.

Mühlhoff, R. (2020). Human-aided artificial intelligence: or, how to run large computations in human brains? Toward media sociology of machine learning. *New Media Soc.* 22, 1868–1884. doi: 10.1177/1461444819885334

Ning, X., Tian, W., Yu, Z., Li, W., Bai, X., and Wang, Y. (2022). HCFNN: high-order coverage function neural network for image classification. *Pattern Recogn.* 108873. doi: 10.1016/j.patcog.2022.108873

Quan, H., Li, Y., and Zhang, Y. A. (2020). novel mobile robot navigation method based on deep reinforcement learning. *Int. J. Adv. Robotic Syst.* 17, 1729881420921672. doi: 10.1177/1729881420921672

Regan, B., Aghajamali, A., Froech, J., Tran, T. T., Scott, J., Bishop, J., et al. (2020). Plastic deformation of single-crystal diamond nanopillars. *Adv. Mater.* 32, 1906458. doi: 10.1002/adma.201906458

Sharma, R., Kumar, A., Meena, D., and Pushp, S. (2020). Employing differentiable neural computers for image captioning and neural machine translation. *Proc. Comp. Sci.* 173, 234–244. doi: 10.1016/j.procs.2020.06.028

Stock, T., Obenaus, M., Kunz, S., and Kohl, H. (2018). Industry 4.0 as enabler for a sustainable development: a qualitative assessment of its ecological and social potential. *Proc. Safety Environ. Prot.* 118, 254–267. doi: 10.1016/j.psep.2018.06.026

Suryanarayana, C. (2019). Mechanical alloying: a novel technique to synthesize advanced materials. *Research.* 1–17. doi: 10.34133/2019/4219812

Urien, P. (2019). "Introducing innovative bare metal crypto terminal for blockchains and bigbang paradigm," in *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 1–4.

Vishwakarma, G., and Lee, W. (2018). Exploiting JTAG and its mitigation in IOT: a survey. *Future Int.* 10, 121. doi: 10.3390/fi10120121

Wang, C., Wang, X., Zhang, J., Zhang, L., Bai, X., Ning, X., et al. (2022). Uncertainty estimation for stereo matching based on evidential deep learning. *Pattern Recogn.* 124, 108498. doi: 10.1016/j.patcog.2021.108498

Wang, X., Wang, C., Liu, B., Hasan, N. A., Davidson, R. M., Epperson, L. E., et al. (2021). Multi-view stereo in the deep learning era: a comprehensive revfiew. *Displays* 70, 102102. doi: 10.1016/j.displa.2021.102102

Xu, T., Wang, K., and Song, S. (2021). Measurement uncertainty and representation of tensile mechanical properties in metals. *Metals* 11, 1733. doi: 10.3390/met11111733

Yao, L., and Guan, Y. (2018). An improved LSTM structure for natural language processing. *IEEE Int. Conf. Safety Prod. Inform. (IICSPI)* 2018, 565–569. doi: 10.1109/IICSPI.2018.8690387

Ye, J.-Y., Zhang, L.-W., and Reddy, J. (2020). Large strained fracture of nearly incompressible hyperelastic materials: enhanced assumed strain methods and energy decomposition. *J. Mech. Phys. Solids* 139, 103939. doi: 10.1016/j.jmps.2020.103939

Yuan, S., and Fan, X. (2019). Developments and perspectives on the precision forming processes for ultra-large size integrated components. *Int. J. Extreme Manufact.* 1, 022002. doi: 10.1088/2631-7990/ab22a9

frontiers | Frontiers in Neurorobotics

TYPE Brief Research Report
PUBLISHED 29 September 2022
DOI 10.3389/fnbot.2022.1006755

Check for updates

OPEN ACCESS

EDITED BY
Xin Ning,
Institute of Semiconductors
(CAS), China

REVIEWED BY
Yuning Tao,
South China University of
Technology, China
Quanyong Wang,
Guangdong Research Institute of
Petrochemical and Fine Chemical
Engineering, China

*CORRESPONDENCE
Qi Zheng
zhengq0727@163.com

RECEIVED 29 July 2022
ACCEPTED 25 August 2022
PUBLISHED 29 September 2022

CITATION
Zheng Q (2022) Efficient recognition of dynamic user emotions based on deep neural networks. *Front. Neurorobot.* 16:1006755. doi: 10.3389/fnbot.2022.1006755

Efficient recognition of dynamic user emotions based on deep neural networks

Qi Zheng*

School of Communication, Zhengzhou Normal University, Zhengzhou, China

The key issue at this stage is how to mine the large amount of valuable user sentiment information from the massive amount of web text and create a suitable dynamic user text sentiment analysis technique. Hence, this study offers a writing feature abstraction process based on ON-LSTM and attention mechanism to address the problem that syntactic information is ignored in emotional text feature extraction. The study found that the Att-ON-LSTM improved the micro-average F1 value by 2.27% and the macro-average F value by 1.7% compared to the Bi-LSTM model with the added attentivity mechanisms. It is demonstrated that it can perform better extraction of semantic information and hierarchical structure information in emotional text and obtain more comprehensive emotional text features. In addition, the ON-LSTM-LS, a sentiment analysis model based on ON-LSTM and tag semantics, is planned to address the problem that tag semantics is ignored in the process of text sentiment analysis. The experimental consequences exposed that the accuracy of the ON-LSTM and labeled semantic sentiment analysis model on the test set is improved by 0.78% with the addition of labeled word directions compared to the model Att-ON-LSTM without the addition of labeled semantic information. The macro-averaged F1 value improved by 1.04%, which indicates that the sentiment analysis process based on ON-LSTM and tag semantics can effectively perform the text sentiment analysis task and improve the sentiment classification effect to some extent. In conclusion, deep learning models for dynamic user sentiment analysis possess high application capabilities.

KEYWORDS

deep learning models, dynamic users, sentiment analysis, text extraction, tag semantics

Introduction

In recent years, online social media and mobile smart terminals have emerged in large numbers and developed rapidly. They provide people with new communication process and interactive spaces, making people's lifestyles change dramatically (Aprem and Krishnamurthy, 2017). A growing number of people tend to use smart terminals to obtain information, exchange ideas and spread information through online social media. The process of information dissemination is no longer the one-way communication of traditional media, but interactive communication (Nduhura and Prieler, 2017). Among them, there is a large amount of text as the simplest and most direct carrier for human beings to express their thoughts and spread knowledge in the Internet. It involves hot events, product reviews, news information and many other aspects. They contain rich emotional information and attitudinal views, with high social value and commercial application value (Liu et al., 2017). Emotion, a physiological and psychological state that results from a combination of feelings, thoughts, and behaviors (Chao et al., 2017). Such as happiness, sadness and anger are also among the chief factors that influence human behavior. People discuss certain hot topics and express their opinions on social platforms. They publish reviews of products or services on shopping platforms, etc. These unstructured online texts contain a lot of valuable information about users' emotions (Angioli et al., 2019). These emotional messages inherently have a certain abstraction and are difficult to be processed directly. It has attracted the attention of many researchers to find out how to extract this effective information from the huge amount of web texts and then apply it effectively in real life, resulting in the emergence of dynamic user text sentiment analysis techniques.

Initial studies on sentiment analysis focused on coarse-grained research on sentiment polarity dichotomization or trichotomization. With the gradual research on text sentiment analysis techniques and the desire to have a more comprehensive understanding of users' psychological states, the study of text sentiment analysis gradually shifted to more fine-grained multi-categorization research. Emotion recognition is the foundation and prerequisite of emotion classification. In the vast amount of realistic web texts, there are filled with abundant unemotional texts and emotional texts. Emotionless text is an objective description of things or events without any emotion. Emotional texts contain personal emotions such as happiness, anger, sadness and so on. Emotional texts are the main object of textual emotion analysis, therefore, it is necessary and chief to identify the presence or absence of emotions in a large amount of real texts and filter out emotional texts. Sachdev et al. (2020) collected a corpus of blog posts, which were annotated with word-level emotion categories and intensities, and used a knowledge-based approach to identify sentences with and without emotions with an accuracy of 73.89%. Wallgrün et al. (2017) constructed a corpus oriented to microblogging texts, annotated whether the microblogging texts contain emotion information or not. At the same time, they performed a multi-label annotation of the emotion categories contained in the microblog texts with emotions. They summarized the results of the NLP&CC2013 Chinese microblog sentiment analysis evaluation task on sentiment recognition, which facilitated the research related to sentiment analysis. Huang et al. (2017) planned a text emotion recognition process based on syntactic information, which expands the performance of emotion recognition by making full use of syntactic information through lexical annotation sequences and syntactic trees. Emotion classification, as chief research direction of emotion analysis, is based on emotion recognition to classify texts containing emotion information at a finer granularity and obtain the specific emotion category (e.g., happy, angry, sad, etc.) expressed by the user in the text. The majority of early sentiment classification studies utilized lexicon- and rule-based process to determine sentiment categories. Fine-grained multi-class sentiment classification is the difficulty and focus of sentiment analysis, researchers have studied sentiment classification from different perspectives, such as construction of sentiment corpus (Fraser and Liu, 2014; Kawaf and Tagg, 2017), author sentiment and reader sentiment prediction (Chang et al., 2015; Yoo et al., 2018), document-level sentimentality organization, sentence-level sentimentality organization and word-level sentiment classification (Liu et al., 2020; Zhang L., et al., 2021). Alves and Pedrosa (2018) planned a process based on frequency and co-occurrence information to classify the sentiment of headline texts by making full use of the co-occurrence relationship between contextual words and sentiment keywords. Subsequently, various researchers have used traditional machine learning-based process for sentiment classification studies. Pan et al. (2017) planned a multi-label K-nearest neighbor (KNN) based sentiment classification process that explores the polarity of words, sentence subject-verb-object components, and semantic frames as features for their impact on sentiment classification. To differ from this, Aguado et al. (2019) have taken into accounts the interaction of sentiments between sentences and use a coarse-to-fine analysis strategy to do sentiment classification of sentences, first obtaining the set of possible sentiments embedded in the target sentence roughly using a multi-label K-nearest neighbor approach, and then refining the sentiment category of the target sentence by combining the sentiment transfer probabilities of neighboring sentences. Rao et al. (2016) utilized a maximum entropy model to model words and multiple sentiment categories in a text to estimate the relationship between words and sentiment categories to classify sentiment in short texts. Traditional machine learning process extract text features manually through feature engineering, while deep learning uses representation learning process to extract text features automatically without relying on artificial features. Effective feature extraction is the core of research on emotion classification process, and most of research works show that reasonable use of deep learning techniques based on neural networks to extract rich semantic information in emotional texts contributes to the effectiveness of text emotion classification. Abdul-Mageed and Ungar (2017) constructed a large-scale fine-grained sentiment analysis dataset employing Twitter data and designed a represent neural network based on gating units to achieve 24 classes of fine-grained sentiment classification. Kim and Huynh (2017) experimentally explored text emotion classification utilizing the LSTM model as well as its variant nested long-short-term memory network (Nested LSTM) model, respectively, which showed that the Nested LSTM model facilitates better accuracy of sentiment classification. Wang et al. (2016) planned neural network (NN) model based on bilingual attentively mechanisms for the problem of emotion classification in bilingual mixed text, where the LSTM model is used to construct a document-level text illustration and the attention mechanism captures semantically rich words in both monolingual and bilingual texts. In addition, some studies have applied a joint multi-task learning approach to the task of emotion classification. Awal et al. (2021) incorporated emotion classification and emotion cause detection as two subtasks into a unified framework through a joint learning model, trained simultaneously to extract the emotional features needed for emotion classification and the event features needed for emotion cause detection. Yu et al. (2018) planned a dual-attention-based transfer learning process that aims to improve the performance of emotion classification using sentiment classification. At present, there are plenty of results for research on text sentiment analysis process, but sentiment analysis still faces many challenges due to the colloquial and irregular nature of online texts and the complexity of sentiment itself (Ning et al., 2021). For text representation, most approaches use pre-trained models such as Word2Vec and GloVe to obtain word directions, which are simple, efficient, and can characterize contextual semantics well, but suffer from the problem of multiple meanings of a word (Ning et al., 2022). For text feature extraction, commonly used NNs such as CNN and Bi-LSTM extract semantic features while ignoring the syntactic hierarchy features of the text (Ning et al., 2020). Most approaches only symbolize sentiment category labels and act as a supervisory role in the classification process, while ignoring semantic information contained in the labels themselves, which is undoubtedly a "semantic waste". In this paper, we explore and improve the text sentiment analysis process based on the above three problems.

Based on this, the main research of this paper is to use deep learning techniques to accomplish the task of emotional analysis of online text, based on the ordered neuronal long and short term memory network (ON-LSTM) and attention mechanism, and incorporating the semantic information of emotional category labels to build the emotional analysis model ON- LSTM-LS. First, the text features are extracted based on ON-LSTM and attention mechanism. Then the sentiment analysis model based on ON-LSTM and label semantics. In this study, for online social text, we build the sentiment analysis model ON-LSTM-LS based on ordered neuron long and short term memory network and attention mechanism, and incorporate the semantic information of sentiment category labels to improve the performance of text sentiment analysis.

Theory and model construction

Text representation

Text data usually consists of a set of unstructured or semi-structured strings. Since computers cannot directly recognize and process text strings, they need to numerate or directionize the text, i.e., text representation. Text representation enables computers to process real text efficiently and is a fundamental and chief step in the study of text sentiment analysis. In Chinese, words are generally considered to be the most basic semantic units of text. Therefore, general research for Chinese text should first perform word separation operation, and then the words in the text are represented afterwards.

Word direction representation

NN-based distributed representation is also recognized as word direction, word embedding or distributed representation of words. This process models the target word, context of target word and the relationship between them, and represents the target word as a low-dimensional solid real-valued direction in continuous space. Compared with matrix-based distributed representation and cluster-based distributed representation, word direction representation can contain more and more complex semantic information. It is extensively used in various normal language dispensation errands.

The word directions are gained by training the language model, which uses a single-layer NN to perform the solution of the binary language model while obtaining the word direction representation. Based on this, the NN language model NNLM is planned (Wang et al., 2022). The model takes the first k words of the present word w_t, $w_{t-k-1}, \ldots, w_{t-1}$, as input and uses a NN to predict the conditional likelihood of the occurrence of the present word w_t to obtain a word direction representation while training the language model. As the NNLM is capable of handling only fixed-length sequences, lacking flexibility. Due to the slow training speed, the researchers improved the NNLM. They planned two words direction training models, which are successive bag-of-words model CBOW and skip-word model. They open source a tool Word2Vec for word direction computation. Differs from the NNLM in which the representation of the present word w_t depends on its predecessor. In CBOW and Skip-gram models, the representation of the present word w_t depends on k words before and after it. In the CBOW and Skip-gram models, representation of present word w_t depends on k words before and after it.

Both CBOW and Skip-gram models have same three-layer hierarchy: input layer, mapping layer, output layer, and structure diagram are exposed in Figure 1A. In the input layer, the input words are randomly initialized as N-dimensional directions. They enter the hidden layer after a simple linear operation, and then likelihood distribution of the target words is output by hierarchical Softmax.

The CBOW model is to predict the provisional likelihood of the occurrence of the word w_t by the context $\left\{w_{t-k}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+k}\right\}$ of present word w_t, which

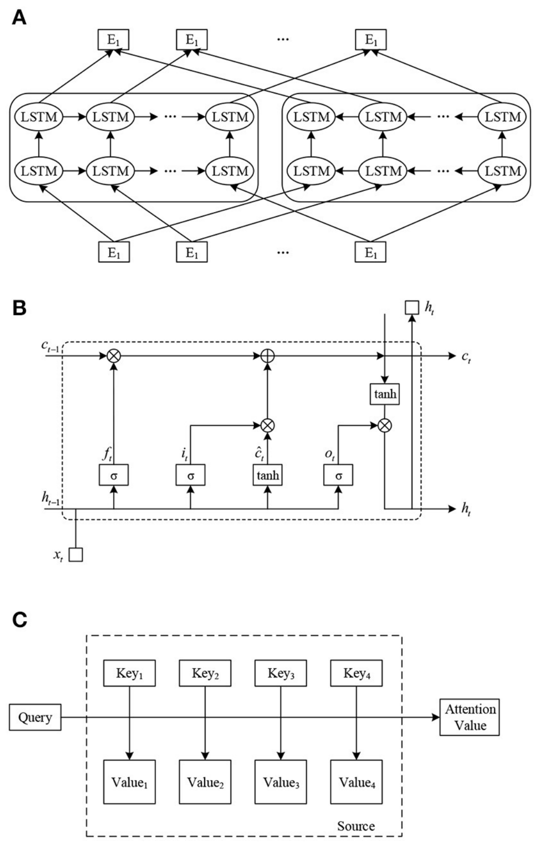


FIGURE 1
Model structure diagram. **(A)** ELMo model structure diagram. **(B)** LSTM model structure diagram. **(C)** Attention model structure diagram.

is calculated as:

$$p\left(w_t|c_t\right) = \frac{\exp\left(e\prime(w_t)^T x\right)}{\sum\limits_{i=1}^{|V|} \exp\left(e\prime(w_i)^T x\right)} \tag{1}$$

$$x = \sum_{i \in k} e\left(w_i\right) \tag{2}$$

where c_t denotes the words $\left\{w_{t-k}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+k}\right\}$ in the context window, $e\left(w_t\right)$ is input direction of word w_t, $e\prime\left(w_t\right)$ is output direction of word w_t, x is the context direction, V is the corpus word list.

The Skip-gram model uses the present word w_t to predict the conditional likelihood of each word in the vocabulary to occur in its context (Wang et al., 2021) and is calculated as:

$$p\left(w_j|w_t\right) = \frac{\exp\left(e\prime\left(w_j\right)^T e\left(w_t\right)\right)}{\sum_{i=1}^{|V|} \exp\left(e\prime(w_i)^T e\left(w_t\right)\right)} \tag{3}$$

To limit the number of restrictions to restore the training efficiency of the model, Word2Vec is optimized using two techniques, hierarchical Softmax and negative sampling. In the training process of model, representation of words is constantly updated by the output direction of the mapping layer is the direction of words.

Considering that Word2Vec only considers contextual co-occurrence features within a finite window when learning, global statistical features are ignored. Researchers propose the GloVe word embedding technique that fuses global and local contextual features of text. gloVe belongs to matrix-based distribution representation, which is a global logarithmic bilinear regression model. They construct global word-word co-occurrence matrices based on the statistical information between words in the corpus. They use a global matrix decomposition and local context window approach for unsupervised training of non-zero positions in the co-occurrence matrix, and a back-propagation algorithm to solve the word directions. Compared with Word2Vec, GloVe has stronger scalability.

Word2Vec and GloVe use pre-training techniques to represent each word as a word direction with a contextual semantic representation. They solve the problems of sparse data, dimensional disaster, and lack of semantic information representation that exist in traditional one-hot directions. At the same time, they have become the mainstream text representation techniques because of their simplicity and efficiency. They have greatly contributed to the development of natural language processing research.

Dynamic word direction technique

While the word direction representation techniques, represented by Word2Vec and GloVe, have greatly advanced the development of natural language processing, they also have some problems. One of the biggest problems is polysemy. Word2Vec and GloVe learn word directions as static word directions, i.e., the word-direction relationship is one-to-one. In other words, no matter how the context of a word changes, the trained word direction is uniquely determined and does not change in any way as the context changes. This static word direction cannot solve the problem of multiple meanings of a word. To solve the problem, researchers have conducted some exploratory research. They have planned a dynamic word direction technique based on language model. Dynamic word directions are not fixed, but change at any time according to the contextual background. ELMo, a language model that can be used to train dynamic word directions, is presented next.

ELMo (Embedding from Language Models) is a novel deep contextualized word representation model planned by Peters et al. By training the model, high quality dynamic word directions can be gained. The model is trained using a deep bi-directional language model (biLM) to obtain different word representations for different contextual inputs, i.e., to generate word directions dynamically.

ELMo uses a two-stage training process. The first stage is to pre-train a language model on a large corpus using a multilayer biLM before training on a specific task. This language model is equivalent to a "dynamic word direction generator" that generates specific word directions for a specific task. In the second stage, the word directions generated by the language model in the first stage are added to the downstream task as a feature supplement for task-specific training. The model structure of ELMo is exposed in Figure 1B. Where $(\mathrm{E}_1, \mathrm{E}_2, \ldots, \mathrm{E}_\mathrm{N})$ are the static Word Embedding of the input word sequence, $(\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_\mathrm{N})$ are the dynamic Word Embedding of the output gained by pre-training the ELMo model.

The ELMo model employs bivectorial LSTM language model to train word representation in the first stage. Specifically, suppose that given a word sequence $(w_1, w_2, \ldots, w_N)$ of length N, the forward LSTM language model calculates the sequence likelihood of the occurrence of the word at the $1, 2, \ldots, k-1$ by taking the word sequence at the given first position as:

$$p\left(w_1, w_2, \ldots, w_N\right) = \prod_{k=1}^{N} p\left(w_k | w_1, w_2, \ldots, w_{k-1}\right) \tag{4}$$

The backward LSTM language model calculates the sequence likelihood of the occurrence of words at position $k+1, k+2, \ldots, k+N$ by taking the sequence of words at position k given as:

$$p\left(w_1, w_2, \ldots, w_N\right) = \prod_{k=1}^{N} p\left(w_k | w_{k+1}, w_{k+2}, \ldots, w_N\right) \tag{5}$$

The combination of frontward LSTM language model and backward LSTM language model constitutes the bi-directional language model biLM, which is required to maximize the following objective function during the training process as:

$$\sum_{k=1}^{N} \left(\begin{array}{l} \log p\left(w_k | w_1, w_2, \ldots, w_{k-1}; \Theta_x, \vec{\Theta}_{lstm}, \Theta_s\right) \\ + \log p\left(w_k | w_{k+1}, w_{k+2}, \ldots, w_N; \Theta_x, \overleftarrow{\Theta}_{lstm}, \Theta_s\right) \end{array} \right) \tag{6}$$

where Θ_x denotes the word direction matrix, Θ_x is the parameter of the softmax layer, $\vec{\Theta}_{lstm}$ and $\bar{\Theta}_{lstm}$ denote frontward LSTM language model and the backward LSTM language model, respectively.

The L-layer biLM model is used in the pre-training, and the word representations gained from each layer have different features. For each input word, the word representation with features such as syntactic semantics is output for it by pre-training with the L-layer biLM model as:

$$\begin{aligned} R_k &= \left\{x_k^{LM}, \vec{h}_{k,j}^{LM}, \bar{h}_{k,j}^{LM} | j = 1, \ldots, L\right\} \\ &= \left\{h_{k,j}^{LM} | j = 0, \ldots, L\right\} \end{aligned} \tag{7}$$

where $h_{k,0}^{LM}$ is the word layer, $h_{k,j}^{LM} = \left|\vec{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM}\right|$.

A language model and word representations for each hidden layer will be gained after the first stage of training. In the second stage, the sentences in the downstream task will be used as input to the dimensional ELMo. For each word in the sentence, ELMo combines the word representations of all hidden layers into a direction by calculating the weights of each hidden layer. That is, a direction of words in the present context. It is formalized as:

$$\mathrm{ELMo}_k^{\mathrm{task}} = E\left(R_k; \Theta^{\mathrm{task}}\right) = \gamma^{\mathrm{task}} \sum_{j=0}^{L} s_j^{\mathrm{task}} h_{k,j}^{LM} \tag{8}$$

where γ^{task} is the validated global scaling factor and s_j^{task} is the softmax normalized weighting factor.

ELMo is able to dynamically generate different word directions for the similar word in dissimilar circumstances. It conforms to the word direction of the present context. It solves the problem of encoding multiple meanings of a word to some extent and performs well in several natural language processing tasks.

Deep learning models

The concept of deep learning (DL) (Wang et al., 2021) originated from the study of artificial NNs. It involves of multiple layers of artificial NNs connected to be able to extract effective feature representation information from a large amount of input data. Deep learning mimics the way the human brain operates. It learns from experience and has the ability to excel in representation learning. It has been successfully applied in several research fields.

Represent NNs

Represent neural network (RNN) is a NN with short-term memory capability to process temporal information of varying lengths. It is widely used for several tasks in natural language processing. RNNs use neurons with self-feedback to process temporal information of arbitrary length by unfolding multiple times. Where x_t denotes input direction at the time of t, h_t denotes state of the hidden layer at the time of t. o_t denotes output direction at the time of t. U denotes Input layer to hidden layer weight matrix, V denotes Value matrix of weights from hidden layer to output layer. W denotes value of the weight matrix of the preceding moment of hidden layer as input value of the current moment.

The RNN gives an output o_t with present network hidden layer state h_t for input x_t at t. The value of h_t at time t depends upon not only x_t, as well as also on the hidden layer state h_{t-1} at previous moment, calculated as:

$$o_t = g\left(Vh_t\right) \tag{9}$$

$$h_t = f\left(Ux_t + Wh_{t-1}\right) \tag{10}$$

where f represents a non-Linear activation function such as sigmod or tanh. The network parameters are shared at different moments and are trained by the backpropagation over time algorithm (BPTT).

Theoretically, RNNs are capable of handling text sequences of arbitrary length. However, in practice, when the length of text sequences is too long, the problem of gradient explosion or gradient dispersion occurs. It makes parameter updating difficult, which in turn prevents RNNs from learning long-range dependency information. It also leads to biased learning results of long-range dependencies, i.e., RNNs learn short-term dependence.

Short long-term memory network

A variation of RNN, Long-Short-Term Memory Network (LSTM) can effectively tackle the problem of gradient explosion or gradient dispersion in RNN (Yan et al., 2021). The improvement of LSTM for RNN is twofold. On the one hand, during the training process of the network, RNN has only one state h_t at the moment t, LSTM adds a state c_t on this basis, c_t represents the memory state of the represent unit, which involves a small number of function operations and thus can store long distance information. On the other hand, LSTM introduces a gating mechanism and designs three gate structures, forgetting gate, input gate, and output gate, which enable represent NN to selectively forget some unchief information while remembering the past information through the interaction between the three gates, and thus learn longer distance dependencies.

The cell of LSTM has certain memory function because of it. It also called a memory cell. The structure of the LSTM loop cell is exposed in Figure 1B. where x_t represents input direction of the memory cell at moment t. h_t represents output direction of the memory cell at time t. represents present information after updating the memory. f_t, o_t and i_t represent the forgetting gate, output gate and input gate, respectively.

The forgetting gate f_t selectively forgets a portion of the cell state information through the sigmod layer, i.e., it determines how much of the cell condition c_{t-1} of previous moment needs to be retained in the cell state c_t of the present moment. The calculation formula is as follows:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{11}$$

Input gate i_t selectively records new inputs in the memory cells through the sigmod layer. It determines how much of the present network's input x_t needs to be saved into the cell state c_t. Also, the cell state c_t of the present input is determined based on the output h_{t-1} of hidden layer state and x_t at the present moment, which is calculated as follows:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{12}$$

$$\hat{c}_t = tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{13}$$

The present memory $\hat{c}_t$ and the historical memory c_{t-1} need to be combined before the output gate to update the cell state c_t at the present moment. The forgetting gate allows chief information from long ago to be preserved, and the input gate allows irrelevant information from the present input to be filtered and forgotten.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{14}$$

The output gate o_t inputs the input information and the present cell state update to the next hidden layer h_t through the sigmod layer, i.e., it controls how much of the cell state c_t needs to be output to the hidden layer state h_t at the present moment. The calculation formula is as follows:

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{15}$$

$$h_t = o_t \circ tanh\left(c_t\right) \tag{16}$$

In the above equations, W_f, W_i and $W_\circ$ are weight matrices from input layer to the forgetting gate, input gate and output gate, respectively. U_f, U_i and U_o are weight matrices from output layer to the forgetting gate, input gate and output gate, respectively. W_c is connection weight from input layer to the LSTM loop unit. U_c is the connection weight from the previous node to the present node of the LSTM loop unit, b_f, b_i, b_o and b_c are all offsets.

Weight parameters in RNN are shared across time steps, which is why there is gradient explosion or dispersion. In contrast, there are multiple paths of gradient propagation in LSTM. In which the process of cell state update at the present moment is carried out by element-by-element multiplication and summation. Its gradient flow is relatively stable, thus greatly reducing the risk of gradient explosion or dispersion. Thus, LSTM is able to handle long-range temporal information.

The input gate determines degree of retention of the input information. The forgetting gate determines the extent to which memory information is forgotten. The output gate, on the other hand, controls the extent to which internal memory is output to the outside. Each of the three gating switches has its own role, which enables LSTM to effectively use historical information and establish long-range temporal dependencies. In turn, it is widely used in tasks related to sequential problems.

Gated circulation unit

Gated represent unit (GRU) is a represent NN based on another gating mechanism. The basic design idea of GRU is the same as LSTM, and it can be said to be a variant of LSTM. Its difference lies in two main aspects. On the one hand, the structure of GRU is relatively simple, using two gate structures. The reset gate determines how much historical information requires to be forgotten. The update gate determines how much of the history information can be saved to the present state. On the other hand, GRU directly passes the hidden state to the next cyclic unit without using an output gate.

Where x_t represents input direction at the present moment, h_{t-1} represents state at previous moment, $\tilde{h}_t$ is the candidate state at the present moment, r_t and z_t represent reset gate and update gate, respectively, and output direction h_t at the present moment is calculated as in Eqs. (17–20).

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1}\right) \tag{17}$$

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1}\right) \tag{18}$$

$$\tilde{h}_t = tanh\left(W_h x_t + U_h\left(r_t \circ h_{t-1}\right)\right) \tag{19}$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t \tag{20}$$

Reset gates help capture short-term dependencies. The update gate, on the other hand, helps to capture long-term dependencies. Therefore, GRU can handle long and short-term dependencies in sequences. At the same time, GRU has one less gate structure compared to LSTM, and the number of parameters is relatively less, and the overall training speed is improved.

Attentional mechanisms

The Attention mechanism assigns higher weights to task-relevant information by weighting (Zhang M., et al., 2021). While assigning lower weights to task-irrelevant information, and then filtering the relatively chief information from the large amount of information. Introducing the attention mechanism into machine translation tasks in the natural language domain achieves significant effect improvement. Since then, the attention mechanism has received some attention in natural language processing. Researchers have combined it with DNN to extract features that are more relevant to the task and thus expand the performance of the model. For example, in sentiment classification, a set of directions or matrices with parameters are used to characterize the importance of words in a text sentence. In the process of extracting features, the features that are more relevant to the sentiment classification are extracted based on the importance of the words.

The core idea of the attention mechanism is to move from “focus on all” to “focus”. It focuses the limited attention on the chief information related to the task, so that effective information can be gained quickly. As exposed in Figure 1C, the Source can be observed as the content deposited in a memory, whose rudiments consist of (address Key, value Value), and given a query with Key=Query, the Value conforming to the query is removed, i.e., the Attention value. The specific calculation process is:

(1) The similarity between the Query and each Key is calculated as the weight coefficient of the Value conforming to each Key. The similarity is usually calculated by dot product, cosine, multilayer perceptron network, etc.

$$Sim_i = F\left(Query,\ Key_i\right) \tag{21}$$

(2) The similarity gained in the previous stage is normalized using softmax and transformed into a similarity whose sum of all similarity weights is 1, thus highlighting the weights of chief elements. The calculation formula is as follows, and is the weight coefficient conforming to each value.

$$\alpha_i = softmax\left(Sim_i\right) = \frac{e^{Sim_i}}{\sum_{j=1}^{L_x} e^{Sim_j}} \tag{22}$$

(3) The normalized weighting coefficients are weighted and summed with the conforming Value to obtain the final attention value.

$$Attention\left(Query,\ Source\right) = \sum_{i=1}^{L_x} \alpha_i \cdot Value_i \tag{23}$$

Presently, the values of Key and Value are same in the research for natural language processing. And in the commonly used self-attention mechanism (self-Attention), Query (Q), Key (K) and Value (V) are all from the same input and all three are the same, noted as Q = K = V. In text analysis, for example, a sentence is input, and the self-attention mechanism requires that each word in the sentence is computed with all the words in the judgment to learn the dependences among the words within the sentence. The computation is formalized as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{24}$$

where d_k is the dimension of Q and K directions, $\sqrt{d_k}$ is a scaling factor, and the same division of $\sqrt{d_k}$ is to avoid the softmax gradient from decreasing or even disappearing due to excessive dot product.

Experiment and analysis

ON-LSTM-A based text feature extraction process

To confirm the dependability of the experimental consequences, this chapter uses the public dataset provided by the NLP&CC2013 Chinese microblog sentiment analysis evaluation task. The dataset is derived from Sina Weibo, with a total of 14,000 microblog texts. The dataset is divided into 7 categories of emotions and labeled with "none" for the text without emotions. In this chapter, the experiment classifies the text with emotion, so the text without emotion is removed. The distribution of the number of each type of emotional samples in the experimental data set is exposed in Table 1. From them, 60% were arbitrarily designated as the training set and 40% as the test set for the experiments.

This experiment uses the same evaluation metrics as the evaluation task: macro-averaged F1 values ($Macro_{F\ measure}$) and micro-averaged F1 values ($Macro_{F\ measure}$), which are calculated as follows:

$$Macro_{Percision} = \frac{1}{7}\sum_i \frac{\#system\_orrect\,(emotion = i)}{\#system\_roposed\,(emotion = i)} \tag{25}$$

$$Macro_{\mathrm{Re}\,call} = \frac{1}{7}\sum_i \frac{\#system\_orrect\,(emotion = i)}{\#gold\,(emotion = i)} \tag{26}$$

$$Macro_{F\_measure} = \frac{2 \times Macro_{Percision} \times Macro_{Recall}}{Macro_{Percision} + Macro_{Recall}} \tag{27}$$

$$Micro_{Percision} = \frac{\sum_i \#system\_orrect\,(emotion = i)}{\sum_i \#system\_roposed\,(emotion = i)} \tag{28}$$

$$Micro_{Recall} = \frac{\sum_i \#system\_correct\,(emotion = i)}{\sum_i \#gold\,(emotion = i)} \tag{29}$$

$$Micro_{F\_measure} = \frac{2 \times Micro_{Percision} \times Micro_{Recall}}{Micro_{Percision} + Micro_{Recall}} \tag{30}$$

Where #gold is the number of manually labeled results, #system_correct is the number of correctly classified tweets, #system_planned is the total number of tweets forecasted by the model for the present category, *i* is one of the seven categories of sentiment.

To confirm the effectiveness of the Att-ON-LSTM model planned in this study, a comparison experiment with the following similar models was designed.

LSTMl, which encodes text sequences from front to back in a unidirectional manner. Only the influence of the above information on the below information is considered, and the contextual semantic features of the text sequences are extracted.

Bi-LSTM, encoding text sequences from both positive and negative directions, extracting to the relationship before and after the text, for the more reasonable text sequence features.

ON-LSTM: ordered neuron long and short-term memory network, capable of extracting text semantics along with hierarchical syntactic information of text to obtain comprehensive text features.

Bi-LSTM+Attention, which adds attention mechanism to Bi-LSTM to extract contextual semantic features, making the model focus more on the features of words related to classification.

ON-LSTM+Attention: that is, the sentiment analysis model Att-ON-LSTM based on ON-LSTM and attention mechanism planned in this chapter. Three layers of ON-LSTM network are used to extract text features, and attention mechanism is used to increase the attention of words related to classification. The results of the comparison experiments are exposed in Table 2.

The experimental results show that ON-LSTM has improved micro F1 values and macro F1 values compared to LSTM and Bi-LSTM. The ON-LSTM+Attention model, i.e., Att-ON-LSTM planned in this section, improves the micro-average F1 value by 2.27% and the macro-average F value by 1.7% compared with the Bi-LSTM model with the added attention mechanism. It is indicated that the text features extracted by Att-ON-LSTM are more comprehensive and can effectively improve the effect of text emotion classification. In this section, the Att-ON-LSTM model is experimentally validated by comparing it with similar baseline models. The study illustrates that the ON-LSTM network can better extract text features, which helps to improve the effect of sentiment analysis.

TABLE 1 Emotional sample data statistics table.

Emotional category	Happiness	Sadness	Joy	Disgust	Anger	Fear	Surprise
Number of samples	1,487	1,132	2,155	1,360	671	151	348

TABLE 2 Comparison results of similar models.

Models	$Micro_{F\_measure}$	$Macro_{F\_measure}$
LSTM	0.3039	0.0665
Bi-LSTM	0.3008	0.2632
ON-LSTM	**0.3144**	**0.2781**
Bi-LSTM+Attention	0.3090	0.2893
ON-LSTM+Attention(ours)	**0.3317**	**0.2910**

TABLE 3 Experimental environment configuration.

Experimental environment	Configuration
CPU	19
Memory	64GB
Video card	NVIDIA GTX 2080ti
Development languages	python 3.7.4
Deep learning tools	TensorFlow 1.15.0

Research on sentiment analysis process based on ON-LSTM and label semantics

Experimental environment and parameter setting

This chapter uses Google's open source deep learning framework TensorFlow to complete the experiment, the specific experimental environment configuration is shown in Table 3.

The model parameter settings in the experiments of this chapter are adjusted according to the performance of the validation set, and the relevant parameter settings are shown in Table 4.

Experimental data set

To ensure the reliability of the experimental results, the CLUE Emotion Analysis Dataset provided by the Chinese Language Understanding Benchmark Assessment was used for the experimental data in this chapter. The corpus in this dataset comes from Sina Weibo and contains a total of 39,661 emotion samples, each with an emotion category label. There are seven types of emotion category labels: happiness, sadness, like, disgust, anger, fear, and surprise. In this experiment, 80% of the dataset is selected. Serving as the training set, 10% being used as the validation set, while the remaining 10% being used as the test set, specific distribution of the sample data is shown in Table 5.

Evaluation indicators

The sentiment classification problem is a multi-category problem and the following metrics are used in this experiment: Accuracy, Precision P, Recall R and F1 value. Accuracy usually measures the performance of the model on the whole data set. Accuracy cannot be reflected when the model is biased and is always wrong in certain categories of judgments. Therefore, it needs to be judged for each category using the P, R, and F1 values, and then averaged, and thus judged for the model as a whole. There are two types of averaging: Macro-average and Micro-average. Macro-average considers each category equally, increasing the impact of categories with less data. Micro-average considers each sample to be classified equally and is more influenced by common categories. In order to better measure the classification effect of the model for each category, the F1 value of macro-average is chosen as the evaluation index in this paper.

For the entire data set, Accuracy is calculated as:

$$Accuracy = \frac{\sum_{i=1}^{7} a_{E_i}}{N} \tag{31}$$

where $\sum_{i=1}^{7} a_{E_i}$ denotes the sum of the number of correctly forecasted samples in each category and N is the total number of samples in the dataset.

For the sentiment category, Precision, Recall and F1-measure are calculated as:

$$P = \frac{a}{a+c} \tag{32}$$

$$R = \frac{a}{a+b} \tag{33}$$

$$F1 = \frac{2 \cdot P \cdot R}{P+R} \tag{34}$$

Macro-averaging treats each category equally and calculates the arithmetic mean of the indicators for each category as:

$$Macro - P = \frac{1}{7}\sum_{i=1}^{7} P_i \tag{35}$$

$$Macro - R = \frac{1}{7}\sum_{i=1}^{7} R_i \tag{36}$$

TABLE 4 Model-related parameter settings.

Parameter name	Value
Word vector dimension	768
Maximum number of words in a sentence	100
ON-LSTM training dimension	[100,128]
Learning rate	0.01
Dropout ratio	0.25
Batch	256
Number of iterations	30

TABLE 5 Statistics on the distribution of samples in the data set.

Emotional category	Training set	Validation set	Test set	Total	Percentage (%)
Happy	7,975	1,006	978	9,959	25.11
Sad	11,210	1,394	1,448	14,052	35.43
Pleasant	3,657	430	453	4,540	11.45
Disgusted	3,896	509	471	4,876	12.29
Anger	3,657	458	447	4,562	11.50
Fear	525	69	67	661	1.67
Surprise	808	100	103	1,011	2.55
Total	31,728	3,966	3,967	39,661	100.00

$$Macro - F1 = \frac{2 \times Macro - P \times Macro - R}{Macro - P + Macro - R} \quad (37)$$

Analysis of experimental results

To confirm the effectiveness of the sentiment analysis model based on ON-LSTM and label semantics and its performance, two sets of experiments were designed in this study. The first group is an ablation experiment to confirm the effect of label semantics on the effect of sentiment classification. The second group is a similar model comparison experiment, which compares the ON-LSTM-LS model with other sentiment analysis models based on deep learning to confirm the effectiveness and performance of the ON-LSTM-LS model planned in this chapter.

A. Ablation experiment results and analysis.

The set of experiments is based on the Att-ON-LSTM model in Chapter 3, and the semantics of labeled word directions, the semantics of labeled semantics expanded text, and optimization with a weighted loss function are added in turn to confirm the effect of labeled semantics on the effect of emotion classification. The contrasting models are as follows.

(a) ON-LSTM + Attention: The model is Att-ON-LSTM, a sentiment analysis model based on ON-LSTM and attention mechanism in Section ON-LSTM-A based text feature extraction process it uses the combination of three-layer ON-LSTM network and attention mechanism to extract the sentiment features in the text as the baseline model for this group of experiments;

(b) ON-LSTM + Attention + Label: add word direction semantic features of labeled words to the Att-ON-LSTM model. It is combined with the sentiment features of the text to guide the model for sentiment classification using label semantics;

(c) ON-LSTM + Attention + Label + Extra: This model expands the label semantics using the label semantic expansion process. It obtains richer label semantic features and expects to improve the effect of emotion classification;

(d) ON-LSTM+Attention+Label+Extra+Customloss: This model is the ON-LSTM-LS model planned in this chapter. The cross-entropy loss function with weights is used on the basis of the model in (3) to alleviate the problem of sample imbalance.

The experimental results of the above model on the test set are exposed in Table 6. Compared with the model Att-ON-LSTM without adding labeled semantic information, the accuracy of the model improves by 0.78% after adding labeled word directions. The macro-average F1 value is improved by 1.04%. Its shows that the semantic information of tag words is helpful to improve the effect of sentiment classification. However, the improvement effect is limited because the semantic information of the labeled words is not sufficient. After using the label semantic expansion process to enrich the semantic features of the labels, the accuracy of the model is improved by 1.79%. The average F1 value of the macro was improved by 2.02%. It indicates that the label semantic expansion process planned for this paper is effective. The optimized model ON-LSTM-LS using the cross-entropy loss function with weights has further improved the effect of emotion classification. Its accuracy is improved by 5.83% relative to the baseline model Att-ON-LSTM. The macro-average F1 value is improved by 6.38%, further demonstrating that the ON-LSTM-LS model can effectively improve the effectiveness of text sentiment analysis.

B. Comparative experimental results and analysis of similar models

This group of experiments compares the ON-LSMT-LS model with related similar models to confirm the validity and accuracy of the ON-LSTM-LS model planned in this chapter. The comparison models are as follows:

(1) LSTM: Long Short-Term Memory Network Model, which encodes text sequences from front to back. It is

TABLE 6 Results of ablation experiments.

Models	Accuracy	Macro-F1
ON-LSTM + Attention **(baseline)**	0.4691	03300
ON-LSTM + Attention + Label	0.4769	0.3404
ON-LSTM+Attention+ Label + Label Extra	0.4870	03502
ON-LSTM+Attention+ Label + Label Extra + Cutom Loss **(ours)**	**0.5229**	**0.3938**
ON-LSTM + Attention **(baseline)**	0.4691	03300

able to learn the distant information in the text and extract the one-way semantic features of the text.

(2) CNN: Convolutional neural network model, which is able to extract n-elements features at different locations in a sentence by convolutional operation. It is able to learn textual relations within a certain distance through pooling operation and has an advantage over LSTM in terms of training speed.

(3) LSTM-CNN: a hybrid model of long- and short-term memory network and convolutional neural network, the long- and short-term memory network is used to extract the global features of text. Convolutional neural network is used to extract local features of text, and then complete the learning of text contextual features.

(4) Bi-LSTM: Bi-directional long- and short-term memory network model, which encodes text sequences from both positive and negative directions. It can learn the association relationship between the preceding and following words in the text and extract the contextual semantic features of the text.

(5) Att-Bi-LSTM: Emotion analysis model based on Bi-LSTM and attention mechanism, which adds attention mechanism to the bidirectional long and short-term memory network. Its makes the model pay more attention to the words that have more influence on the classification effect during the learning process.

(6) ON-LSTM: ordered neuron long and short-term memory network model. It is able to extract the text semantics while extracting the hierarchical syntactic information of the text to obtain a more comprehensive text feature.

TABLE 7 Experimental results of comparing similar models.

Models	Accuracy	Macro-F1
LSTM	03,250	0.0764
CNN	0.3893	0.2747
LSTM-CNN	0.4034	0.2633
Bi-LSTM	0.4279	03582
Att-Bi-LSTM	0.4650	0.3264
ON-LSTM	0.4325	0.3251
Att-ON-LSTM	0.4691	03300
ON-LSTM-LS **(ours)**	**0.5229**	**0.3938**

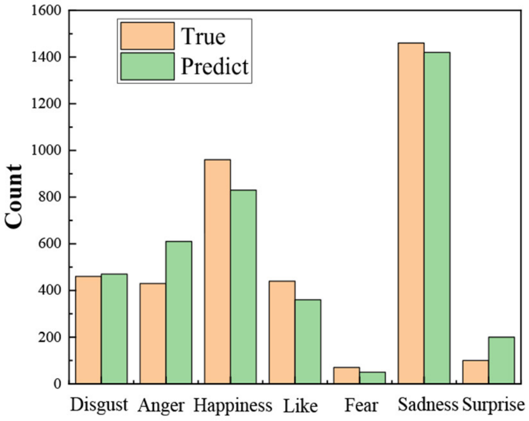


FIGURE 2
Distribution of forecasted categories and true categories.

“The experimental results of the above models on the test set are exposed in Table 7. The model ON-LSTM-LS planned in this paper has a substantial improvement in accuracy and macro-average F1 value compared with similar baseline model LSTM, CNN, LSTM-CNN, Bi-LSTM, and Att-Bi-LSTM. The accuracy is improved by 19.79, 13.36, 11.95, 9.5, and 5.79%, respectively. Its shows that using three-layer ON-LSTM to extract semantic features of emotional text and combining them with semantic features of labels can improve the effectiveness of emotional classification more substantially. Compared with ON-LSTM and Att-ON-LSTM, the accuracy is improved by 9.04 and 5.28%, respectively. The macro-average F1 value score was improved by 6.87 and 6.38%. The study illustrates that the semantic information of labels is helpful for the improvement of sentiment analysis. As a whole, the ON-LSTM-LS model planned in this paper outperforms similar models for sentiment classification. The study shows that the ON-LSTM-LS model can effectively perform the text sentiment classification task with certain advantages.

Figure 2 shows the distribution of the predicted and true results of the ON-LSTM-LS model for different categories on the test set. The difference between the actual and predicted values of the seven emotions is small. However, due to data imbalance, the model has better classification results for two categories, disgust and sadness, than the prediction results for the remaining five categories. And the prediction effect of two emotions, happy and angry, was poor. In the next work, different methods should be explored to alleviate the data imbalance problem.

Conclusion

How to extract this effective information from the huge amount of web texts and then apply it effectively in real life. This problem has led to the emergence of dynamic user text sentiment analysis techniques. To address the problem that syntactic information is ignored in emotional text feature extraction, this paper proposes a text feature extraction process based on ON-LSTM and attention mechanism. It is proved that it can better extract the semantic and hierarchical information in the emotional text and obtain more comprehensive emotional text features. The experimental results show that the sentiment analysis process based on ON-LSTM and tag semantics can effectively complete the text sentiment analysis task and improve the sentiment classification effect to a certain extent. The specific findings of the study are as follows:

(1) Att-ON-LSTM, compared with the Bi-LSTM model with added attention mechanism, improved the micro-average F1 value by 2.27% and the macro-average F value by 1.7%. The text features extracted by Att-ON-LSTM are more comprehensive and can effectively improve the effect of text emotion classification;

(2) The experimental results of ON-LSTM and the sentiment analysis model with labeled semantics on the test set showed a 0.78% improvement in the accuracy of the model with the addition of labeled word directions compared to the model Att-ON-LSTM without the addition of labeled semantic information. The macro-average F1 value was improved by 1.04%;

(3) The accuracy of the model improved by 1.79% after the label semantic features were enriched using the label semantic expansion process. The macro-average F1 value was improved by 2.02%. The sentiment classification of the model ON-LSTM-LS, which was optimized using the cross-entropy loss function with weights, was further improved. Its accuracy was improved by 5.83% relative to the baseline model Att-ON-LSTM.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

QZ planned the idea of revising the paper, designed the comparison experiment, and analyzed the results, then wrote the paper.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdul-Mageed, M., and Ungar, L. (2017). "EmoNet: Fine-Grained Emotion Detection with Gated Represent Neural Networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada: Association for Computational Linguistics, p. 718–728. doi: 10.18653/v1/P17-1067

Aguado, L., Dieguez-Risco, T., Villalba-García, C., and Hinojosa, J. (2019). Double-checking emotions: valence and emotion category in contextual integration of facial expressions of emotion. *Biol. Psychol.* 146, 107723. doi: 10.1016/j.biopsycho.2019.107723

Alves, P. S., and Pedrosa, R. (2018). Neurologist-level classification of stroke using a Structural Co-Occurrence Matrix based on the frequency domain. *Comput. Electr. Eng.* 71, 398–407. doi: 10.1016/j.compeleceng.2018.07.051

Angioli, R., Casciello, M., Lopez, S., et al. (2019). Assessing HPV vaccination perceptions with online social media in Italy. *Int. J. Gynecol. Cancer* 29, ijgc-2018-000079. doi: 10.1136/ijgc-2018-000079

Aprem, A., and Krishnamurthy, V. (2017). Utility change point detection in online social media: a revealed preference framework. *IEEE Transac. Signal Process.* 65, 1869–1880. doi: 10.1109/TSP.2016.2646667

Awal, M. R., Cao, R., Lee, K. W., and Mitrovic, S. (2021). "Angrybert: Joint learning target and emotion for hate speech detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. (Cham: Springer).

Chang, Y. C., Chen, C. C., Hsieh, Y. L., Chen, C., and Hsu, W. (2015). "Linguistic template extraction for recognizing reader-emotion and emotional resonance writing assistance," in *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7rd International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015).*

Fraser, A., and Liu, Y. (2014). *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: Tutorials*.

Huang, L., Shou-Shan, L. I., and Zhou, G. D. (2017). Emotion recognition of chinese microblogs with syntactic information. *Comput. Sci*.

Kawaf, F., and Tagg, S. (2017). The construction of online shopping experience: a repertory grid approach. *Comput. Human Behav.* 72, 222–232. doi: 10.1016/j.chb.2017.02.055

Kim, Y. G., and Huynh, X. P. (2017). Discrimination between genuine versus fake emotion using long-short term memory with parametric bias and facial landmarks. *IEEE Int. Conf. Comput. Vis. Workshop*, 3065–3072. doi: 10.1109/ICCVW.2017.362

Liu, F., Zheng, L., and Zheng, J. (2020). HieNN-DWE: a hierarchical neural network with dynamic word embeddings for document level sentiment classification. *Neurocomputing* 403, 21–32. doi: 10.1016/j.neucom.2020.04.084

Liu, Y., Wang, J., Jiang, Y., Sun, J., and Shang, J. (2017). Identifying impact of intrinsic factors on topic preferences in online social media: a nonparametric hierarchical Bayesian approach. *Inf. Sci.* 423, 219–234. doi: 10.1016/j.ins.2017.09.041

Nduhura, D., and Prieler, M. (2017). When I chat online, I feel relaxed and work better: exploring the use of social media in the public sector workplace in Rwanda. *Telecommun. Policy.* 41, 708–716. doi: 10.1016/j.telpol.2017.05.008

Ning, X., Gong, K., Li, W., et al. (2020). Feature refinement and filter network for person re-identification. *IEEE Transac. Circ. Syst. Video Technol.* 31, 3391–3402. doi: 10.1109/TCSVT.2020.3043026

Ning, X., Gong, K., Li, W., et al. (2021). JWSAA: joint weak saliency and attention aware for person re-identification, *Neurocomputing* 453, 801–811. doi: 10.1016/j.neucom.2020.05.106

Ning, X., Tian, W., Yu, Z., Li, W., Bai, X., and Wang, Y. (2022). HCFNN: high-order coverage function neural network for image classification. *Pattern Recogn.* 131, 108873. doi: 10.1016/j.patcog.2022.108873

Pan, Z., Wang, Y., and Ku, W. (2017). A new k-harmonic nearest neighbor classifier based on the multi-local means. *Expert Syst. Appl.* 67, 115–125. doi: 10.1016/j.eswa.2016.09.031

Rao, Y., Xie, H., Li, J., Jin, F., Wang, F., and Li, Q. (2016). Social emotion classification of short text via topic-level maximum entropy model. *Inf. Manage.* 53, 978–986. doi: 10.1016/j.im.2016.04.005

Sachdev, S., Sita, T. L., Shlobin, N. A., Shlobin, N., Gopalakrishnan, M., Sucholeiki, R., et al. (2020). Completion corpus callosotomy with stereotactic radiosurgery for drug-resistant, intractable epilepsy: case report. *World Neurosurg.* 143, 440–444. doi: 10.1016/j.wneu.2020.08.102

Wallgrün, J., Karimzadeh, M., Maceachren, A. M., and Pezanowski, S. (2017). GeoCorpora: building a corpus to test and train microblog geoparsers. *Int. J. Geogr. Inf. Sci.* 32, 1–29. doi: 10.1080/13658816.2017.1368523

Wang, C., Ning, X., Sun, L., Zhang, L., Li, W., and Bai, X. (2022). Learning discriminative features by covering local geometric space for point cloud analysis. *IEEE Transac. Geosci. Remote Sens.* 60, 1–15. doi: 10.1109/TGRS.2022.3170493

Wang, X., Wang, C., Liu, B., et al. (2021). Multi-view stereo in the Deep Learning Era: a comprehensive revfiew. *Displays* 70, 102102. doi: 10.1016/j.displa.2021.102102

Wang, Z., Zhang, Y., Lee, S., Li, S., and Zhou, G. (2016). A bilingual attention network for code-switched emotion prediction. *Int. Conf. Comput. Linguist.*

Wu, C., Zhang, Y., Jia, J., and Zhu, W. (2017). Mobile Contextual Recommender System for Online Social Media. *IEEE Transac. Mobile Comput.* 16, 3403–3416.

Yan, C., Pang, G., Bai, X., et al. (2021). Beyond triplet loss: person re-identification with fine-grained difference-aware pairwise loss. *IEEE Transac. Multimedia* 24, 1665–1677. doi: 10.1109/TMM.2021.3069562

Yoo, S. Y., Song, J., and Jeong, O. R. (2018). Social media contents based sentiment analysis and prediction system. *Expert Syst. Appl.* 105, 102–111. doi: 10.1016/j.eswa.2018.03.055

Yu, J., Marujo, L., Jiang, J., Karuturi, P., and Brendel, W. (2018). "Improving multi-label emotion classification via sentiment classification with dual attention transfer network," in *Proceedings of the 2018 Conference on Empirical Process in Natural Language Processing*.

Zhang, L., Sun, L., Yu, L., Dong, X., Chen, J., Cai, W., et al. (2021). ARFace: attention-aware and regularization for face recognition with reinforcement learning. *IEEE Transac. Biometr. Behav. Identity Sci.* 4, 30–42. doi: 10.1109/TBIOM.2021.3104014

Zhang, M., Palade, V., Wang, Y., et al. (2021). Attention-based word embeddings using Artificial Bee Colony algorithm for aspect-level sentiment classification. *Inf. Sci.* 545, 713–738. doi: 10.1016/j.ins.2020.09.038

frontiers | Frontiers in Neurorobotics

TYPE Brief Research Report
PUBLISHED 26 October 2022
DOI 10.3389/fnbot.2022.1009093

OPEN ACCESS

EDITED BY
Weijun Li,
Chinese Academy of Sciences, China

REVIEWED BY
S. A. Edalatpanah,
Ayandegan Institute of Higher Education (AIHE), Iran
Yuning Tao,
South China University of Technology, China

*CORRESPONDENCE
Charles Chen
cchen20220606@163.com

RECEIVED 01 August 2022
ACCEPTED 30 September 2022
PUBLISHED 26 October 2022

CITATION
Wang X and Chen C (2022) Case report: Quantitative recognition of virtual human technology acceptance based on efficient deep neural network algorithm.
Front. Neurorobot. 16:1009093.
doi: 10.3389/fnbot.2022.1009093

Case report: Quantitative recognition of virtual human technology acceptance based on efficient deep neural network algorithm

Xu Wang[1] and Charles Chen[2]*

[1]School of Government, Sun Yat-sen University, Guangzhou, China, [2]Guangzhou Foreign Language School, Guangzhou, China

With the advancement of artificial intelligence, robotics education has been a significant way to enhance students' digital competency. In turn, the willingness of teachers to embrace robotics education is related to the effectiveness of robotics education implementation and the sustainability of robotics education. Two hundred and sixty-nine teachers who participated in the "virtual human education in primary and secondary schools in Guangdong and Henan" and the questionnaire were used as the subjects of study. UTAUT model and its corresponding scale were modified by deep learning algorithms to investigate and analyze teachers' acceptance of robotics education in four dimensions: performance expectations, effort expectations, community influence and enabling conditions. Findings show that 53.68% of the teachers were progressively exposed to robotics education in the last three years, which is related to the context of the rise of robotics education in schooling in recent years, where contributing conditions have a direct and significant impact on teachers' acceptance of robotics education. The correlation coefficients between teacher performance expectations, effort expectations, community influence, and enabling conditions and acceptance were 0.290 ($p = 0.000 < 0.001$), -0.144 ($p = 0.048 < 0.05$), 0.396 ($p = 0.000 < 0.001$), and 0.422 ($p = 0.000 < 0.001$) respectively, indicating that these four core dimensions both had a significant effect on acceptance. Optimization comparison results of deep learning models show that mDAE and AmDAE provide a substantial reduction in training time compared to existing noise-reducing autoencoder models. It is shown that time-complexity of the deep neural network algorithm is positively related to the number of layers of the model.

KEYWORDS

digital virtual human, acceptance, deep learning, UTAUT model, neural network algorithm

Introduction

The beginning of worldwide research on artificial intelligence (AI) traces back to the Dartmouth Symposium held in 1956, in which the American scholar McCarthy defined the concept of AI from an engineering perspective, and AI has accumulated enormous potential over the past 60 years. Education, as one of the important fields of AI applications, is moving toward a new ecology of AI education. In other words, AI is leading the transformation of education and becoming an essential factor in promoting the development of education information technology integration and innovation (Wenge, 2021; Kim and Shim, 2022). As the most revolutionary technology nowadays, it will be of great benefits in optimizing the teaching environment, intelligent assessment, personalized tutoring, identifying classroom deficiencies and enhancing the learning experience to facilitate accurate teaching. There is no doubt that this will shock the traditional education objectives, contents and processes. Therefore, along with the rapid development of the intelligent era, the education field will be facing greater challenges and should make full use of AI technology to deepen education reform comprehensively and build an intelligent, lifelong and personalized talent training system so that education can better serve and develop people (Woolf et al., 2013; Aoun, 2017; Ahmad et al., 2021).

With the new wave of AI development, practice and application of avatar education in the basic education stage is becoming increasingly popular and gradually becoming an important vehicle for the development of AI (Benitti, 2012). As a concrete manifestation of the change in the basic approach and methodology of teaching, it not only plays an important role in promoting students' innovative spirit, computational thinking, practical skills and social skills, but also facilitates the development of interesting learning courses and the construction of personalized (student-specific) learning environments. This is a key element in the effective practice and promotion of robotics education, with virtual teachers playing a pivotal role in delivering robotics courses and guiding students in robotics competitions. Currently, China's education and teaching model has transformed from the "teacher-centered" and then "student-centered" unilateral teaching to the current "dominant-subject" model. The dual-focused education and teaching model has evolved into the current "lead-subject" model. Only when teachers play their "leading role" well can students effectively play their "main effect." The starting point for effective practice and application of robotics education in primary and secondary schools is the teacher, which requires not only careful planning of teaching activities and selection of appropriate media and technologies, but also the embedding of new concepts and ideas to compensate for the limitations of traditional teaching models and to expand the advantages of robotics education for teachers and students, thus effectively enhancing teaching effectiveness. Takuya Hashimoto, a Japanese scholar, introduced a self-developed robot teacher into elementary school science classroom teaching, where participating students could discuss relevant issues with the robot teacher, showing that robots have greater potential in elementary school science classroom teaching, not only to enhance learners' knowledge acquisition, but also to improve students' creativity and questions (Hashimoto et al., 2013). Russian academic Elena Ospennikova used a quasi-experimental approach to examine the possibilities of robotics education in science and mathematics curricula. The study selected 186 students from grades 7 to 9 as target subjects and over three years of experimental observation concluded that robotics is a key element in the multidisciplinary orientation of the teaching and learning process in schools (Ospennikova et al., 2015). Andri Ioannou introduced Nao, a humanoid robot developed by Aldebaran Robotics in France, to the education of children with autism (ASD), based on an in-depth analysis of the advantages of combining humanoid robot education with the development of social communication skills in children with autism. After a four-session intervention with a boy with ASD, the robot was found to be an effective way to promote independence and emotional expression in the education of children with ASD (Ioannou et al., 2015). Deep neural networks are an extremely popular research direction in artificial intelligence since 2012, as well as artificial intelligence algorithms for effective analysis and processing of big data (Wang et al., 2021). Its advantages include overcoming the disadvantages of time-consuming and labor-intensive manual feature design, more effective (exponential) distributed data learning by pre-training layer-by-layer data to obtain the primary features of each layer. Compared with shallow modeling approaches, deep modeling enables more detailed and efficient representation of actual complex nonlinear problems. This technique shows potential to efficiently solve quantitative recognition techniques (Foad et al., 2022; Wang et al., 2022).

Objectively, research on robotics education in China has accelerated in the early twenty-first century, however, the key to making robotics education truly effective lies in the ability of teachers to accept and use robotics-supported teaching models. Since teachers' understanding of the concept of informational teaching and learning of the implementation content are internal factors that limit the development of their informational teaching skills (Smith and Sivo, 2012). As a result, studying the acceptance of virtual (robot) education by primary and secondary school teachers as well as grasping its influencing factors are beneficial to the development of robotics education in primary and secondary schools. For this reason, based on the teachers' own perspective, this study draws on the integrated information technology acceptance model (UTAUT model) and the technology acceptance theory model, optimizes the construction of the teachers' acceptance model of robotics

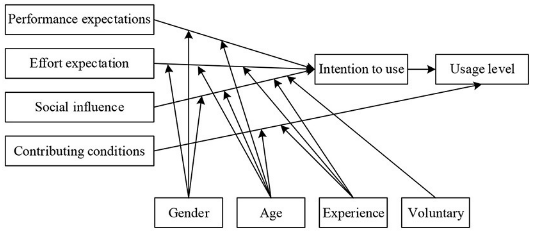


FIGURE 1
UTAUT model.

education based on deep learning algorithms, analyzes the influencing factors of primary and secondary school teachers' acceptance of virtual human education using the questionnaire method, as well as proposes corresponding countermeasures to provide reference for the effective implementation of robotics education at all levels of teaching.

Models and research methods

Deep learning-based construction of UTAUT model

UTAUT model

From the domestic and international studies on teacher IT acceptance models, it is found that UTAUT model is widespread in the field of IT acceptance research. However, by combing through the studies related to robotics education and teacher acceptance, finding that there are fewer studies exploring its effective promotion and implementation from the influence of teachers in the main body of robotics education. Therefore, based on the theoretical basis of the UTAUT model and characteristics of robotics teaching, this study explores the factors influencing teachers' acceptance of robotics education from the teachers' perspective.

The UTAUT model was first proposed by Venkatesh et al. (2003). The model contains four core determinants of performance expectations, effort expectations, community influence and enabling conditions and four moderating variables of age, gender, experience, and voluntariness. As shown in Figure 1. This model explains 70% of technology adoption and usage behavior, outperforms previous technology acceptance models, which is now extensively applied to explore user acceptance behavior.

To investigate the factors influencing teachers in carrying out robotics education, this study remained using the four core determinants in the UTAUT model. Since the development of robotics education in China is oriented to competition or club activities, both teachers and students have little access to robotics. Most teachers had little experience in using robotics and were not highly motivated to do so autonomously. As a result, two moderating variables, experience and voluntariness, were removed, while teaching experience and IT proficiency were added as moderating variables in conjunction with the technical characteristics of robotics education in primary and secondary schools and expert interviews. In addition, considering that acceptance includes both individual's own behavior and individual's attitude toward the object, both usage intention and behavior in the original model are collectively referred to as acceptance level, A theoretical model of factors influencing the acceptance of teacher robotics education is proposed, as shown in Figure 2.

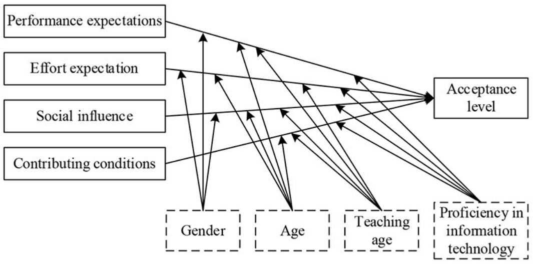


FIGURE 2
Theoretical model of factors influencing teachers' acceptance of robotics education.

Improvising approach based on deep learning

Deep learning network model involves inputting the original input data into a neural network containing multiple implicit layers, through nonlinear operations in the middle multiple implicit layers, where final output of the implicit layers is the deeper, abstract depth features learned from the input data through this deep network model (Guan et al., 2020). However, certain datasets without initial labels to whether the initial labels are involved in the whole network training process will be divided into three categories of deep feature learning, namely supervised feature learning, semi-supervised feature learning and unsupervised feature learning, where supervised feature learning of which can also be referred to as classification, semi-supervised feature learning between the two, which refers to the presence of both labeled and unlabeled data in the trained data, unsupervised supervised feature learning is also known as clustering (Gu et al., 2014).

The Expectation Maximization (EM) algorithm was first proposed by Dempster et al. The EM algorithm has a wide range of applications. The EM algorithm is utilized in numerous algorithms in machine learning (Intisar and Watanobe, 2018; Goulden et al., 2019). Such as the K-means, Support Vector Machine (SVM) (Ukil, 2007), GMM, Hidden Markov Mode (HMM) (Arica and Vural, 2000), Topic Generation Model LDA

(Latent Dirichlet Allocation) (Hoffman et al., 2010), as well as various other models in which parameter estimation EM algorithm is used. It refers to solving some target parameters from the entire data set including hidden variables by iterative iterations employing a strategy of great likelihood estimation. Iteration of the EM algorithm is done by two main steps, E-step (Expectation Step) and M-step (Maximization Step). The expectation of each step of the expectation maximization algorithm is to calculate expectation of the model based on the hidden state of the model, after which Gaussian distribution of the conjectured hidden data is calculated, then fixed model parameters using maximum likelihood estimation to calculate the complete result containing both observed and hidden data, followed by the execution of M-step to finally obtain the parameters of the Gaussian mixture model. The E- and M-step are iterated until the parameters of solved Gaussian mixture model are approximately unchanged. Algorithm convergence is achieved and optimal expectation, covariance matrix and weights of each Gaussian distribution are obtained for the Gaussian mixture model. Expectation of the log-likelihood function of the mixture model is illustrated by the initial values of model parameters that have been selected, as defined in Equation (1):

$$E_Q\left[\log p(\theta|Y,Q)|\theta^{(i)},Y\right]=\int \log[p(\theta|Y,Q)]p\left(Q|\theta^{(i)},Y\right)dQ \tag{1}$$

where, Q denotes implicit data that fail to be observed, $\theta^{(i)}$ denotes posterior standard deviation after the $i+1$st iteration. Conditional expectation probabilities of the joint distribution of the hybrid model can be expressed by Equation (2) as follows:

$$L(\theta,\theta_i)=\sum^{m}\sum P\left(z_i|x_i,\theta_j\right)\log P\left(x_i,z_i|\theta\right) \tag{2}$$

Extreme values of the parameters of the log-likelihood function with conditional probabilities can be bounded by Equation (3) as follows:

$$\theta^{j+1}=\arg\max_{\theta}L\left(\theta,\theta^{j}\right) \tag{3}$$

The above E- and M-step are iterated continuously, terminating the iteration when $\theta^{(i)}$ and $\theta^{(i+1)}$ are infinitely close to each other.

Theoretical model optimization of technology acceptance based on Elman neural network

Elman neural network model

Compared with the ordinary neural network structure, a new takeover layer is incorporated in the Elman network, where the implicit layer transmits processed data to the takeover layer, which memorizes incoming information from the implicit layer and uses received data together with the input layer input at the next moment as the input to the implicit layer at the next moment (Cheng et al., 2002). By storing it through the takeover layer and outputting it to the hidden layer at the next moment, it makes neural network have dynamic memory recognition of historical input data and enhances its ability to treat dynamic information. Its specific mathematical model is:

$$\begin{cases} h(k)=g\left(w_3\cdot q(k)\right) \\ q(k)=f\left[w_1\cdot q_c(k)+w_2(u\cdot(k-1))\right] \\ q_c(k)=q(k-1) \end{cases} \tag{4}$$

where, h represents output of the output layer, $g()$ represents transfer function of the output layer, w_3 represents weight of data received by the output layer that is processed by the implicit layer, q represents state of the implicit layer, and k represents current moment. In the second equation, f represents processing function of the implicit layer, Sigmoid is chosen in most cases, w_1 represents weight of data processed by the implicit layer in the total received data of the takeover layer, w_2 represents weight of the information received by the input layer transmitted to the implicit layer, u represents input of the input layer; q_c in the second and third equations refers to the state output for the takeover layer, and *k-1* in q indicates the previous moment.

Diffusion of innovation theory

Diffusion of Innovation Theory (DIT) was first introduced in 1962 by Everett M. Rogers, an American scholar, who used certain channels to make members of a social group more open to adopting new concepts and things. It emphasizes that an innovation is a thought or concept that can be perceived as novel by an individual or a social community. Diffusion of innovation is the process by which a new product spreads through a social system over a period of time through appropriate communication channels. DIT is divided into five groups of adopters based on the sequence of adoption and usage of innovations: (1) Innovation pioneers: first to adopt and use innovations with a spirit of discovery, accounting for 2.5% of the total; (2) Early adopters: highly visible, adopting and using innovations after the innovation pioneers, accounting for 13.5% of the total; (3) Early adopters: those who take longer to adopt and use innovations with more deliberation than innovation pioneers and early adopters, 34% of the total; (4) Late adopters: those who accept decisions only when they are clearly guided by the norms in the social system, 34% of the total; (5) Conservatives: those who are the last in the social network system to adopt and use innovations, 16 % of the total.

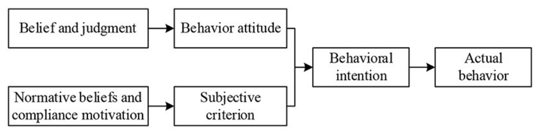


FIGURE 3
TRA model.

Theory of rational behavior

Theory of Rational Behavior (TRA) was co-proposed by American scholars Fishbein and Ajzen in the 1970s to explore the correlation between an individual's internal attitude toward a behavior and the actual performance of that behavior. TRA model has its origins in psychology and covers three basic assumptions: first, social groups are rational and able to accept and utilize knowledge and experience they acquire based on a systemic and holistic view; second, unconscious latent variables do not influence actual behaviors of social groups; and third, individuals themselves entirely determine their own conscious behaviors. The TRA model is given in Figure 3, from which it can be noticed that behavioral intentions in the TRA model can effectively infer actual behaviors used by individuals; while individual attitude and subjective norm that an individual displays when performing a certain behavior can effectively infer one's behavioral intention.

Theory of planned behavior

Theory of Planned Behavior (TPB) was first proposed by American psychologist Ajzen in 1985 to compensate for the limitations of the TRA model (Mahlaole, 2021). TPB model is considered as an extension and improvement of the TRA model, which can make fuller predictions and more convincing explanations of human behavior, as presented in Figure 4. The discrepancy between TPB and TRA lies in the predictors of individual behavioral intentions. In addition to subjective norms and attitudes, which are included in the TRA model, TPB also adds potential variable of perceived behavioral control (PBC). It refers to the perceived ease of performing a behavior. When individuals perceive that they have more opportunities and resources, their internal expectation of behavioral control increases, while the perceived constraints are reduced.

PSD learning algorithm

By analogy with the Widrow-Hoff (WH) learning rule the following equation can be obtained (Hinton and Nowlan, 1990):

$$\Delta w_i = \alpha x_i \left(y_d - y_o \right) \tag{5}$$

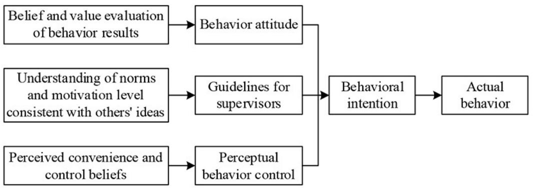

Belief and value evaluation of behavior results
Behavior attitude
Understanding of norms and motivation level consistent with others' ideas
Guidelines for supervisors
Perceived convenience and control beliefs
Perceptual behavior control
Behavioral intention
Actual behavior

FIGURE 4
TPB model.

where, w_i denotes weight of the ith input counterpart, α denotes learning rate, y_d denotes desired sequence, y_0 denotes the actual output sequence, and x_i denotes sequence of inputs. Since the actual output is a sequence containing pulse spikes, it is challenging for derivation, and a derivable continuous value is obtained by convolving a sharp pulse with a convolution kernel when the PSD rule, defining as:

$$K\left(t - t^j\right) = V_0 \cdot \left(\exp\left(\frac{-\left(t - t^j\right)}{\tau_s} \right) - \exp\left(\frac{-\left(t - t^j\right)}{\tau_f} \right) \right) \tag{6}$$

Research methodology

Questionnaire design

In this study, based on the relevant mature scales from existing studies, we designed independently measurement items for each variable in the context of the real situation of robotics education in less developed regions. In order to ensure the reliability and validity of questionnaires, author conducted two rounds of research. In the first round, 35 robotics teachers were randomly selected, followed by a revision of the questionnaire based on the initial research results to better match the real situation of robotics teachers in less developed regions. Final developed formal questionnaire consisted of the following two components with 33 question items. The first part is a survey of basic information of primary and secondary school teachers, with 15 question items, including gender, teaching age, title, school nature, school location, proficiency in information technology, frequency and barriers to robotics education, etc. The second part is a survey of factors affecting teachers' acceptance of robotics education, including five dimensions, namely, performance expectancy (PE), effort expectancy (EE), community influence (SI), enabling conditions (FC), and acceptance (AD), with a total of 18 questions. To ensure robotics teachers' recognition of the questionnaire answers, these measurement questions were in the form of a five-point Likert scale, with 1–5 indicating strongly disagree, disagree, neutral, agree, and strongly agree, respectively.

Questionnaire reliability and validity analysis

Reliability (reliability) focuses on the accuracy, consistency and stability of the recovered sample data. That is, the magnitude of the variability of the measurement results by the random errors generated during the measurement process. Before the formal questionnaire is distributed, reliability testing is normally conducted to purify the content of the questionnaire. The value indicating reliability index is called reliability coefficient, which is correlation coefficient between the results obtained by two or more tests, mostly distributed in the range of 0–1. Reliability tests mainly include test-retest reliability and internal reliability, in which a scale is repeatedly tested on the same target object at different times and the degree of similarity of the test results is then determined. However, repeated tests possibly have the following problems: first, there will be variability in the measurement subject's own cognitive level after having one subject experience; second, the measurement subject may change somewhat when a subject is measured twice or more. Therefore, most experts and scholars use internal reliability to calculate reliability size. Metrics for detecting internal reliability are generally θ coefficient, *Ω* coefficient, Cronbach Alpha confidence coefficient (Cronbach's α), and total correlation coefficient of calibration items (CITC). Among them, CITC and Cronbach's α are the more commonly used methods for reliability evaluation.

- Cronbach's *α* reliability analysis

Cronbach's alpha captures both degrees of internal consistency and correlation among test items and is defined in Equation (7).

$$\alpha = \frac{\mathrm{K}}{\mathrm{K}-1}\left(1-\frac{\sum S_i^2}{S^2}\right) \tag{7}$$

where, α represents Cronbach's alpha coefficient, K represents total number of questionnaire items, S_i^2 represents variance corresponding to the *i*th measurement item, and S^2 represents variance of the whole questionnaire item scores.

When a measurement questionnaire involves several unrelated contents (i.e., different dimensions), it is required to test the internal reliability corresponding to each dimension separately, and on this basis to calculate the internal reliability of the whole questionnaire, instead of directly calculating the alpha coefficient 1 of the whole questionnaire. the reason for this is primarily because questions under the same dimension all reflect the characteristics of a certain aspect and have a high correlation, while the whole questionnaire needs to examine the comprehensive consideration of a certain "coverage," thus there are differences between the one and the other. Larger Cronbach's alpha values indicate better correlation among the items. In general, Cronbach's alpha values >0.9 indicate excellent reliability, as well as Cronbach's alpha values between 0.7 and 0.9 indicate good reliability, meaning that the questionnaire scale is still acceptable. However, if Cronbach's alpha value of each measurement dimension (subscale) is <0.6 and Cronbach's alpha value of the total scale is <0.7, it is determined that internal consistency of the scale is inferior and questionnaire needs to be redesigned. Based on the summary of several researchers' views on Cronbach's alpha value, Ming-Lung Wu divided reliability value testing criteria in detail, as shown in Table 1.

TABLE 1 Cronbach alpha test criteria.

Cronbach's alpha value	Credibility
Cronbachα≥0.9	Extremely credible
0.7≤Cronbachα <0.9	Credible (more common)
0.5≤Cronbachα <0.7	Credible (most common)
0.4≤Cronbachα <0.5	Credible
0.3≤Cronbachα <0.4	Less credible
Cronbachα <0.3	Not credible, should be deleted

- Total item statistics analysis

CITC is designed to measure correlation coefficients between each item and its dimension, in order to remove "junk" items from the questionnaire and clean up the content. There is no unanimous opinion on the evaluation criteria of CITC. Foreign scholar Cronbach considered that questions with CITC <0.5 should be discarded, while domestic scholar Lu Zhendai considered that questions with CITC >0.3 should be retained. The criteria for eliminating items in this study were based on two principles proposed by Cronbach: first, CITC is <0.5; second, alpha coefficient of the deleted item exceeds alpha coefficient of the variable to which it belongs, i.e., the reliability of the potential variable corresponding to the item has improved significantly. Results are shown in Table 2, and items were removed when they met both of these principles. To ensure the scientific validity of the study, two rounds of research were conducted with teachers participating in the "Virtual Human Education in Guangdong and Henan Primary and Secondary Schools" as the research subjects. Thirty-five teachers were selected for the first round of research, formal research was conducted by online questionnaire, 203 questionnaires were collected, of which 190 were valid. The survey results showed that 64.21% of the teachers who participated in robotics education training were male teachers; 78.42% of the teachers were aged between 26 and 45; 96.84% of the teachers' education was concentrated in college and undergraduate level, and only 2.11% of the teachers were graduate and above; 72.11% of the teachers' teaching experience was concentrated in 6–15 years, 15 years or above; 68.42% of the teachers' titles were concentrated in secondary school level 2 and secondary school level 1. In terms of the surveyed teachers' proficiency in IT, 61.58% were competent; teachers from public schools accounted for 98.95%, while teachers from private schools accounted for only 1.05%; rural teachers accounted for 38.95%, while urban teachers accounted for 61.05%. In addition, in terms of the school year in which the teachers serve the students, due to the shortage of teachers in robotics education in Henan Province, there is still a phenomenon that the same teacher teaches students in different grades; therefore, total number of teaching grades involved in the surveyed teachers is >190, covering 67.89, 34.21, and 11.05% of elementary, middle, and high schools, respectively.

Research results and analysis

Relationship analysis of teachers' acceptance of robotics education and influencing factors

Based on the results obtained from the sample data processing analysis to validate initial model and research hypotheses, this study revealed that effort expectation, perceived enjoyment, and performance expectation were factors that directly influenced teachers' acceptance of robotics education, while enabling conditions, community influence, and innovation expectation significantly and indirectly influenced acceptance, and perceived enjoyment could also indirectly influence acceptance through community influence, which will be analyzed and discussed in detail next.

Effect of performance expectations and effort expectations on acceptance

Performance expectations and effort expectations in robotics education positively and directly affect teachers' acceptance (Hl, H2), i.e., the higher performance expectations (PE) or effort expectations (EE) that teachers place on robotics education, the stronger their acceptance of robotics education. This conclusion is not only consistent with the original UTAUT model, but also with earlier research findings (Almaiah et al., 2019; Raffaghelli et al., 2022). Analysis of the paths revealed that performance expectations (path coefficient of 0.117) had a slightly stronger effect on acceptance than effort expectations (path coefficient of 0.101), generally speaking, teachers' willingness to attempt to introduce a new teaching model into their actual classrooms will heavily consider whether the model contributes to their performance levels. For virtual human teachers, if the implementation of robotics education causes them to feel a shift in their role and facilitates their salary, title, or promotion opportunities, which truly leads to more professional development, then this will undoubtedly strengthen their belief in practicing and applying robotics education. Descriptive statistical analysis of the core variables showed that teachers scored higher on the performance expectation level for questions PE-1 and PE-4, with scores of 3.73 and 3.68, respectively, indicating that most teachers perceived robotics education as both a key component in transforming their teaching role and an ideal platform for their professional growth, which facilitated their acceptance of robotics education. However, scores for PE-2 and PE-3 were low, at 3.38 and 3.10, respectively, which indicates that teachers in the current regional basic education level basically do not receive additional rewards for carrying out robotics education, and to some extent, it may also weaken teachers' enthusiasm to carry out robotics education. Therefore, in the process of promoting the practical application of robotics education, teachers' awareness of the concept of robotics education needs to be strengthened. In the process of actively organizing training in robotics project instruction and teaching skills, attention needs to be paid to the role of robotics education in teachers' professional development and to improving relevant assessment and reward mechanisms.

Effects of innovation expectations and facilitating conditions on effort expectations

Enabling conditions and innovation expectations in robotics education positively influenced virtual teachers' own effort expectations (H7, H8), i.e., more innovative teachers or more adequate accommodations already in place, the greater teachers' effort expectations, which enhanced their acceptance of robotics education. This is consistent with earlier research findings as well. The path analysis indicated that innovation expectations (path coefficient of 0.329) acted slightly more on effort expectations than enabling conditions (path coefficient of 0.294). Innovation expectations refer to the degree of teachers' innovativeness and problem-solving intentions when confronted with a new technology or a new pedagogical paradigm, which contributes to teachers' beliefs about the acceptance of a new technology or a new pedagogical paradigm (Rosenbusch et al., 2019). In general, if teachers frequently follow the latest developments of emerging technologies and are particularly willing to experiment with the introduction of new educational ideas into the actual classroom when they are exposed to them, they may not deplete excessive time to pay attention to whether such teaching ideas will affect the teaching order and their own emotions, but whether they are able to understand it or apply it or encounter obstacles to overcome it better as soon as possible, and such teachers are relatively more confident in accepting new teachers are relatively confident in accepting new technologies or teaching ideas, their willingness to try new teaching models is largely to satisfy their curiosity. Conversely, if teachers are reluctant to introduce new teaching models into the classroom, they may subconsciously believe that implementing models will make the classroom disorderly and stressful in guiding students in the process of project practice, which in turn will increase their teaching tasks. Therefore,

TABLE 2 Results of exploratory factor analysis.

Sample	Ingredients				
	1	2	3	4	5
EE1	−0.145	0.654	0.073	−0.012	0.032
EE2	−0.022	0.779	−0.038	0.123	−0.024
EE3	−0.081	0.678	−0.125	−0.042	0.386
EE4	−0.076	0.687	0.168	0.028	−0.298
EE5	0.089	0.668	0.036	−0.125	−0.325
PE1	0.037	0.038	0.134	0.884	0.039
PE2	0.020	−0.006	0.084	0.788	0.062
PE3	0.373	−0.002	−0.025	0.587	0.042
FC1	0.168	−0.098	0.305	0.101	0.808
FC2	0.312	−0.081	0.202	0.045	0.798
S11	0.219	0.068	0.827	0.085	0.152
S12	0.102	0.018	0.878	0.067	0.152
S13	0.219	0.041	0.698	0.192	0.119
AD1	0.702	−0.205	0.258	0.021	0.142
AD2	0.788	−0.198	0.231	0.078	0.143
AD3	0.888	−0.087	0.064	0.010	0.095
AD4	0.878	0.015	0.119	0.052	0.087
AD5	0.787	0.095	0.079	0.118	0.058

TABLE 3 Descriptive statistical analysis of model variables.

Variant	Average value	Standard deviation	Cronbach's α	Sum of Cronbach's α
Performance Expectations	3.26	0.78	0.785	0.782
Effort Expectations	2.92	0.59	0.760	
Community Impact	3.39	0.85	0.808	
Enabling conditions	3.08	0.94	0.842	
Acceptance level	3.87	0.58	0.889	

teachers should be trained to be creative and innovative at the level of their subjectivity and practical activities in receiving robotics education.

Effects of perceived pleasantness and innovation expectations

There is a positive direct effect of teachers' perceived pleasantness on robotics education on their level of acceptance (H6), i.e., the stronger teachers' perceived pleasantness on robotics education, the stronger their internal level of acceptance, which is consistent with earlier findings (Adieze, 2016). By comparing path coefficient values for each factor, it is observed that the direct effect of perceived pleasantness on acceptance is as high as 0.852 ($p = 0.000<0.05$), which is significantly higher than the effect of each other variable. One possible reason for this is that virtual teachers have a strong interest in novelty or new teaching models, they want to be more enjoyable and stimulate their curiosity and exploration, rather than teaching in the traditional test-based education model for a long time, whereas robotics education as an existing new teaching concept can largely lead them to explore new knowledge and stimulate their curiosity, which makes it prone to feel enjoyable and enhance their motivation and interest in teaching, which in turn significantly enhances their belief that they tend to accept the model. As indicated by the

TABLE 4 Correlation coefficients of teachers' acceptance of robotics education and its various influencing factors.

Dimension	Performance expectation	Effort expectations	Community impact	Enabling conditions	Acceptance level
Performance expectations	1				
Effort expectations	0.003	1			
Community impact	0.308***	0.045	1		
Enabling conditions	0.232***	−0.169	0.418***	1	
Acceptance level	0.287***	−0.142*	0.397***	0.412***	1

*$p < 0.05$, **$p < 0.01$, ***$p < 0.001$.

TABLE 5 Compound regression analysis of acceptance and coefficients.

Model	*R*	Square *R*	Adjusted *R*	Square Std. Error of the Estimate
1	0.432	0.196	0.191	2.689
2	0.506	0.254	0.250	2.587
3	0.530	0.281	0.270	2.655

descriptive statistical analysis of core variables, teachers' scores on the question items PP-1, PP-2, and PP-3 in the perceived pleasantness dimension were roughly comparable, with scores of 3.80, 3.82, and 3.75 respectively, indicating that most teachers have favorable perceived pleasantness of robotics education and are willing to actively attempt robotics education, however, they are still between neutral and agree (mean value of 3.79 between 3 and 4). Therefore, in the process of promoting practical applications of robotics education, teachers' perceived enjoyment of robotics education can be further enhanced by training practical activities.

Analysis of teacher acceptance of robotics education

Descriptive statistical analysis of questionnaires

Results of the study demonstrated that the scores for each variable ranged from 2.92 to 3.87, with the highest score for acceptance (3.87) and the lowest score for effort expectancy (2.92). The standard deviation of the variables is <1.0, which indicates that scores of the variables are densely distributed around mean values, and mean values are well-represented, as shown in Table 3.

Variance and regression analysis of teachers' acceptance of robotics education

Taking into account the different background characteristics of elementary and secondary school teachers, one-way ANOVA with independent samples *t*-test was employed in this study to explore the variability in the factors exhibited by teachers from different backgrounds. Results indicated that there were no significant differences in performance expectations, community influence, enabling conditions, and acceptance among teachers of different ages and titles, with significant differences only in effort expectations, suggesting that teachers with older ages and higher titles would perceive robotics instruction as more complex. There were significant differences in effort expectation and acceptance among teachers of different teaching ages, while teachers with more than 15 years of teaching experience showed a phenomenon of "low effort expectation and high acceptance," indicating that teachers of higher teaching ages may perceive many barriers to robotics education, however, it is possible that they want to break through the limitations of the old teaching methods and are more receptive to new things. Teachers with different levels of IT acceptance reached significant differences in terms of effort expectations, enabling conditions, and acceptance levels. There were no significant differences between public and private teachers, urban and rural teachers on these five variables.

Relevance between variables was measured in this study by applying Pearson's product-difference correlation coefficient, and correlations basically existed between all dimensions (Um and Crompton, 1986). Among them, acceptance was positively correlated with performance expectancy, community influence and enabling conditions; since the questions about effort expectancy designed in this study were biased toward the reverse questions, as in the case of conducting robotics education that tends to create uncontrolled, stressful and time-consuming classrooms, results of negative correlation between effort expectancy and acceptance coincided with the design of this experiment; moreover, correlations between effort expectancy and acceptance were weak, results of which are shown in Table 4.

To further validate the hypothesized model, multiple regression analysis was utilized in this study in an attempt

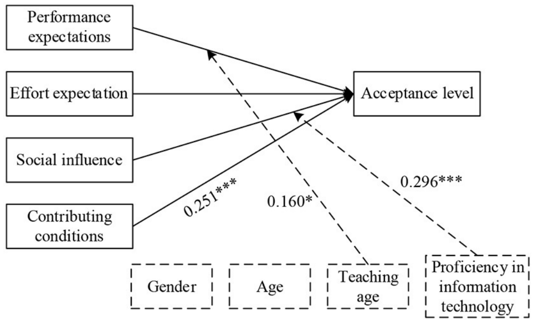


FIGURE 5
The path of influencing factors of virtual human teacher education acceptance.

to examine the causal relationships among the influencing factors. As seen in Table 4, the correlation coefficients between teacher performance expectations, effort expectations, community influence, enabling conditions, and acceptance were 0.290 ($p = 0.000<0.001$), -0.144 ($p = 0.048<0.05$), 0.396 ($p = 0.000<0.001$), and 0.422 ($p = 0.000<0.001$), respectively, indicating that that all four core dimensions had a significant effect on acceptance. In addition, previous one-way ANOVAs showed that moderating variables such as teachers' teaching experience and IT proficiency also had significant effects on acceptance, therefore, multiple regression analysis was attempted in this study to explore the specific effects of these variables on teachers' acceptance, and regression results are presented in Table 5. As seen in this, model 3 explained 28.2% of the results, while the adjusted R2 was finally chosen to explain 27.1% of the results, considering the sample size and the number of independent variables. In particular, enabling conditions had a significant correlation with acceptance; teaching age moderated effects of performance expectations on acceptance; community influence on acceptance was moderated by IT acceptance; effort expectations did not have a direct effect on acceptance of teacher robotics education. Through multiple regression analysis, a path diagram of factors influencing the acceptance of teacher robotics education can be obtained, as shown in Figure 5.

Effect analysis of neural network optimization

To verify effectiveness of the proposed algorithms in this study. Existing deep neural networks composed of noise-reducing autoencoding (DAE), marginalized depth autoencoder (mDAE) and marginalized depth autoencoder with adaptive noise (AmDAE) were compared in experiments under the same conditions. Experiments were conducted to compare the algorithmic performance of the three algorithms, with statistics on the average time required to train the three deep neural network models once. Different implied layer building numbers of deep neural networks with different methods of model training time are shown in Figure 6.

As shown in Figure 6, mDAE and AmDAE have a substantial reduction in training time compared to the existing noise-reducing autoencoder model, reflecting the lower time complexity of the marginalization method, while the improved AmDAE and mDAE models take little difference in training time; model training time basically increases approximately linearly with the number of layers as the number of training layers increases for different standard MNIST variant datasets, indicating that the time complexity of the deep neural network algorithm is positively correlated with the number of layers of the model.

Conclusion

Based on the UTAUT model, this study focuses on the main factors influencing the acceptance of virtual human education by teachers in order to promote application of robotics education in educational teaching activities, by taking some colleges and universities in Guangdong and Henan provinces as examples and draws the following basic conclusions.

- Robotics education is mostly taught by IT teachers, and there is a paradox of "low knowledge and high frequency." Results of study showed that 53.68% of teachers were gradually introduced to robotics education in the last three years, which is related to the background of the rise of robotics education in school education in recent years.
- Descriptive statistical analysis, analysis of variance and reliability tests were conducted on the formal sample data using the appropriate software, while AMOS 22.0 software was used to test the correctness and rationality of the theoretical model and the 15 research hypotheses to obtain the final model that established the factors influencing the acceptance of robotics education by the virtual human teachers. Effects of influencing factors were in descending order: community influence < enabling conditions < effort expectations < performance expectations < innovation expectations < perceived pleasantness.
- Multiple regression analysis of model optimization based on neural networks showed that model 3 explained 28.2% of the results. Meanwhile, its explanatory ratio for the outcome reached 27.1%. In this case, enabling conditions have a significant correlation with acceptance; teaching age moderates effects of performance expectations on acceptance; community influence on acceptance is moderated by IT acceptance; and effort expectations do not directly affect acceptance of teacher robotics education.

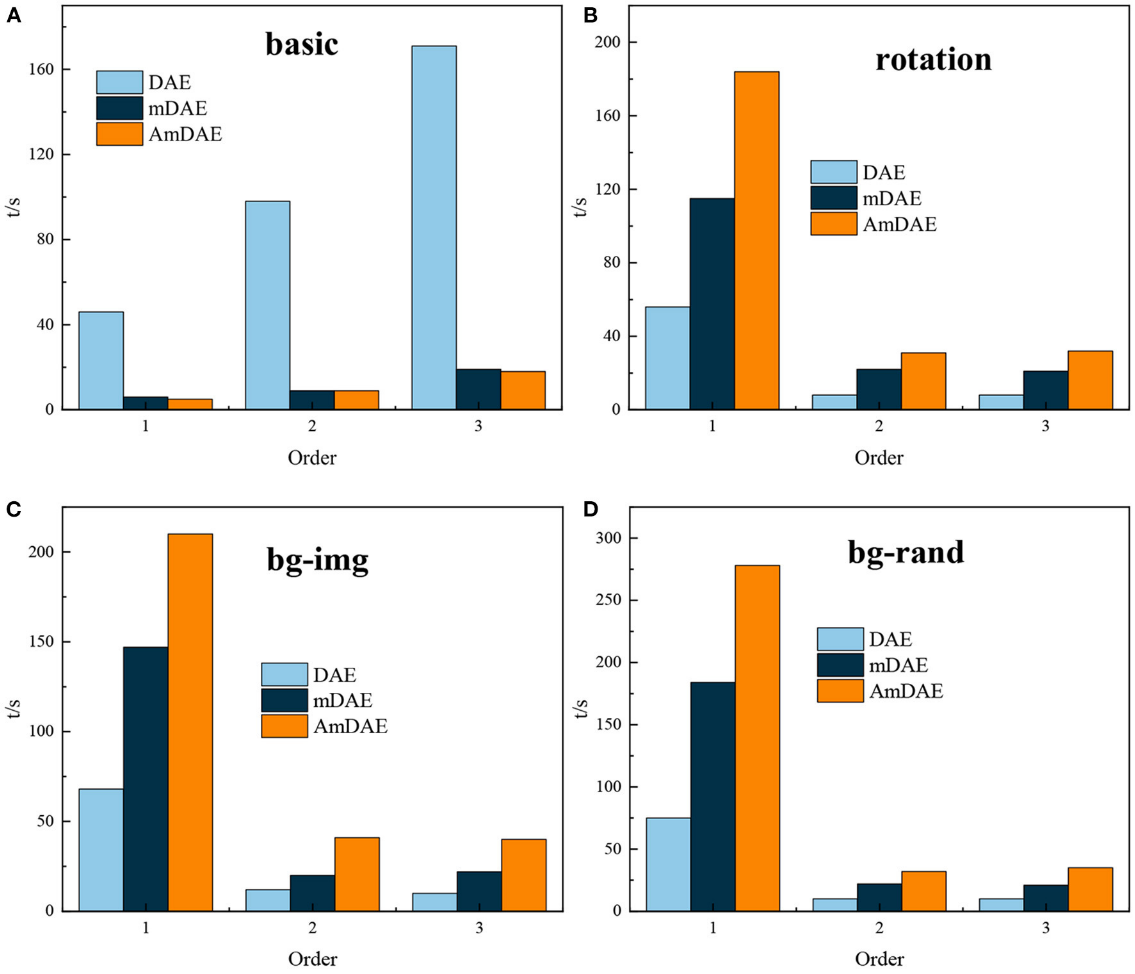


FIGURE 6
Time required for model training. **(A)** basic model, **(B)** rotating model, **(C)** background image, and **(D)** offset model.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Ethics statement

The studies involving human participants were reviewed and approved by Sun Yat-sen University. The patients/participants provided their written informed consent to participate in this study.

Author contributions

XW and CC put forward the core concepts and architecture of this manuscript. XW wrote this article. Both authors contributed to the article and approved the submitted version.

Funding

This study was supported by Guangzhou Higher Education Innovation and Entrepreneurship Education Project (Creation based on Virtual Simulation and Artificial Intelligence New Entrepreneurship Curriculum and Teaching System Construction), Project No. 2019KC113.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Adieze, C. (2016). Effects of edutainment, scaffolding instructional models and demonstration method on students' academic performance in business studies in secondary schools in Abia South Senatorial Zone in Abia State, Nigeria. *Int. J. Educ. Benchmark* 2, 72–84.

Ahmad, S. F., Rahmat, M. K., Mubarik, M. S., Alam, M. M., and Hyder, S. I. (2021). Artificial intelligence and its role in education. *Sustainability* 13:12902. doi: 10.3390/su132212902

Almaiah, M. A., Alamri, M. M., and Al-Rahmi, W. (2019). Applying the UTAUT model to explain the students' acceptance of mobile learning system in higher education. *IEEE Access* 7, 174673–174686. doi: 10.1109/ACCESS.2019.2957206

Aoun, J. E. (2017). *Robot-proof: Higher Education in the Age of Artificial Intelligence*. America: MIT press. doi: 10.7551/mitpress/11456.001.0001

Arica, N., and Vural, F. Y. (2000, September). "A shape descriptor based on circular Hidden Markov Model," in *Proceedings 15th International Conference on Pattern Recognition, ICPR-2000, Vol. 1,* Milan: IEEE, 924–927.

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: a systematic review. *Comput. Educ.* 58, 978–988. doi: 10.1016/j.compedu.2011.10.006

Cheng, Y. C., Qi, W. M., and Cai, W. Y. (2002, November). "Dynamic properties of Elman and modified Elman neural network, in *Proceedings International Conference on Machine Learning and Cybernetics, Vol. 2,* IEEE, 637–640.

Foad, B., Elzohery, R., and Novog, D. R. (2022). Demonstration of combined reduced order model and deep neural network for emulation of a time-dependent reactor transient. *Ann. Nucl. Energy* 171:109017. doi: 10.1016/j.anucene.2022.109017

Goulden, M. C., Gronda, E., Yang, Y., Zhang, Z., Tao, J., Wang, C., et al. (2019). CCVis: visual analytics of student online learning behaviors using course clickstream data. *Elect. Imaging* 2019:681. doi: 10.2352/ISSN.2470-1173.2019.1.VDA-681

Gu, K., Zhai, G., Yang, X., and Zhang, W. (2014, October). "Deep learning network for blind image quality assessment," in *2014 IEEE International Conference on Image Processing (ICIP),* America: IEEE, 511–515. doi: 10.1109/ICIP.2014.7025102

Guan, S., Lei, M., and Lu, H. (2020). A steel surface defect recognition algorithm based on improved deep learning network model using feature visualization and quality evaluation. *IEEE Access* 8, 49885–49895. doi: 10.1109/ACCESS.2020.2979755

Hashimoto, T., Kobayashi, H., Polishuk, A., and Verner, I. (2013, March). "Elementary science lesson delivered by robot," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI),* IEEE, 133–134. doi: 10.1109/HRI.2013.6483537

Hinton, G. E., and Nowlan, S. J. (1990). The bootstrap Widrow-Hoff rule as a cluster-formation algorithm. *Neural Comput.* 2, 355–362. doi: 10.1162/neco.1990.2.3.355

Hoffman, M., Bach, F., and Blei, D. (2010). Online learning for latent dirichlet allocation. *Adv. Neural Inf. Process. Syst.* 23.

Intisar, C. M., and Watanobe, Y. (2018, September). "Classification of online judge programmers based on rule extraction from selforganizing feature map," in *2018 9th International Conference on Awareness Science and Technology (iCAST).* IEEE, 313–318. doi: 10.1109/ICAwST.2018.8517222

Ioannou, A., Kartapanis, I., and Zaphiris, P. (2015). Social robots as co-therapists in autism therapy sessions: a single-case study. *Lect. Notes Comput. Sci.* 9388, 255–263. doi: 10.1007/978-3-319-25554-5_26

Kim, J., and Shim, J. (2022). Development of an AR-based AI education app for non-majors. *IEEE Access* 10, 14149–14156. doi: 10.1109/ACCESS.2022.3145355

Mahlaole, S. T. (2021). Effects of gender on students' entrepreneurial intentions: a theory of planned behaviour perspective. *Open J. Bus. Manage.* 10, 57–76. doi: 10.4236/ojbm.2022.101004

Ospennikova, E., Ershov, M., and Iljin, I. (2015). Educational robotics as an inovative educational technology. *Proced. Soc. Behav. Sci.* 214, 18–26. doi: 10.1016/j.sbspro.2015.11.588

Raffaghelli, J. E., Rodríguez, M. E., Guerrero-Roldán, A. E., and Bañeres, D. (2022). Applying the UTAUT model to explain the students' acceptance of an early warning system in higher education. *Comput. Educ.* 182:104468. doi: 10.1016/j.compedu.2022.104468

Rosenbusch, N., Gusenbauer, M., Hatak, I., Fink, M., and Meyer, K. E. (2019). Innovation offshoring, institutional context and innovation performance: a meta-analysis. *J. Manage. Stud.* 56, 203–233. doi: 10.1111/joms.12407

Smith, J. A., and Sivo, S. A. (2012). Predicting continued use of online teacher professional development and the influence of social presence and sociability. *Br. J. Educ. Technol.* 43, 871–882. doi: 10.1111/j.1467-8535.2011.01223.x

Ukil, A. (2007). "Support vector machine," in *Intelligent Systems and Signal Processing in Power Engineering.* (Berlin, Heidelberg: Springer), 161–226. doi: 10.1007/978-3-540-73170-2_4

Um, S., and Crompton, J. L. (1986). The importance of testing for a significant difference between two pearson product-moment correlation coefficients. *J. Leis. Res.* 18, 206–209. doi: 10.1080/00222216.1986.11969658

Venkatesh, V., Morris, M. G., Davis, G. B., and Davis, F. D. (2003). User acceptance of information technology: toward a unified view. *MIS Q.* 27, 425–478. doi: 10.2307/30036540

Wang, C., Wang, X., Zhang, J., Zhang, L., Bai, X., Ning, X., et al. (2022). Uncertainty estimation for stereo matching based on evidential deep learning. *Pattern Recognit.* 124:108498. doi: 10.1016/j.patcog.2021.108498

Wang, X., Wang, C., Liu, B., Zhou, X., Zhang, L., Zheng, J., et al. (2021). Multi-view stereo in the deep learning era: a comprehensive revfiew. *Displays* 70:102102. doi: 10.1016/j.displa.2021.102102

Wenge, M. (2021). Artificial intelligence-based real-time communication and Ai-Multimedia services in higher education. *J. Multiple-Valued Log. Soft Comput.* 36, 231–248.

Woolf, B. P., Lane, H. C., Chaudhri, V. K., and Kolodner, J. L. (2013). AI grand challenges for education. *AI Mag.* 34, 66–84. doi: 10.1609/aimag.v34i4.2490

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 11 November 2022
DOI 10.3389/fnbot.2022.1054481

OPEN ACCESS

EDITED BY
Praveen Kumar Donta,
Vienna University of Technology, Austria

REVIEWED BY
Pallavi Joshi,
Raisoni Group of Institutions, India
P. Suman Prakash,
G. Pullaiah College of Engineering and Technology (GPCET), India

*CORRESPONDENCE
Rani Kumari
phdcs10058.17@bitmesra.ac.in

RECEIVED 26 September 2022
ACCEPTED 27 October 2022
PUBLISHED 11 November 2022

CITATION
Kumari R and Mustafi A (2022) The spatial frequency domain designated watermarking framework uses linear blind source separation for intelligent visual signal processing.
Front. Neurorobot. 16:1054481.
doi: 10.3389/fnbot.2022.1054481

The spatial frequency domain designated watermarking framework uses linear blind source separation for intelligent visual signal processing

Rani Kumari* and Abhijit Mustafi

Department of Computer Science, Birla Institute of Technology, Ranchi, India

This paper develops a digital watermarking algorithm using an informed watermark retrieval architecture. The developed method uses the fractional Fourier transform to embed the watermark in the space-frequency domain and extracts the watermark using blind source separation techniques. The watermark embedding is further enhanced using a heuristic algorithm to increase the strength of the watermarking system. We use genetic algorithm to find the optimal fractional domain by minimizing the coefficient of RMSE between the input image and the watermarked image. The algorithm's performance against various common attacks, e.g., JPEG compression and Gaussian noise, is presented to estimate the algorithm's robustness.

KEYWORDS

digital watermarking, fractional Fourier transform, blind source separation, genetic algorithm, robustness

Introduction

With the proliferation of computers and other digital devices in society, the rapid growth of the internet, the demand for free availability of copyrighted digital entities, and weak cyber laws existing in most nations, digital piracy has grown to massive proportions in the past few years. Authorities worldwide are struggling to contain this phenomenon, costing many industries a vast amount of money in terms of revenue. Even though many systems have been developed to fight this menace, the effectiveness of these systems remains in question for large-scale public use. Many of these systems utilize the tenets of cryptography to encrypt digital entities before distributing them across networks (public key and private key infrastructures). However, such systems are helpless if rightful owners of copyrighted digital entities make illegal copies of the material and proceed to distribute it illegally. Cryptographic systems are quite capable of preventing abuse of digital entities during the distribution phase but are not very effective once the decryption process has been performed.

Watermarking and steganographic systems attempt to fill this void and complement the functioning of cryptographic algorithms. These algorithms control the authenticity of digital entities even after decryption and allow the ownership of digital entities to be verified. This can be of extreme importance not only for public use but also in sensitive application domains like defense and intelligence. Many of these domains need to maintain stringent control on distributed digital entities, and verifying ownership is crucial. Establishing a source-verified or informed watermarking system is considered an extremely efficient solution in such cases, and developing such systems has become an area of active research.

Various methods have been cited in the literature to embed watermarks in signals. These techniques can be broadly categorized as time (space) domain techniques and frequency domain techniques. For digital images, spatial domain watermarks are embedded by directly manipulating the intensity values of the individual pixels to cause little change in the visual perceptibility of the image. Methods in this category include LSB (least significant bit) based techniques, XOR-based techniques, color manipulation, etc. In the frequency domain, the image is first transformed to the frequency domain using the Fourier transform before the transform coefficients are manipulated to embed the watermark. Watermarking techniques have recently been developed in the time-frequency domain and using the fractional Fourier transform. Wavelet-based techniques have also been proven to be highly effective in digital watermarking. The efficiency of spread spectrums in channel communication has also drawn the attention of researchers working in watermarking and steganography. Spread spectrums provide an excellent diffusion mechanism for spreading watermark signatures across a random set of frequencies and be extremely robust.

In this paper, we present the design of a watermarking system for digital images using the fractional Fourier transform (FrFT) and blind source separation (BSS). The developed system embeds a watermark in the space-frequency domain, and the retrieval of the watermark is accomplished using a BSS technique. The embedded watermark is inserted using a mixing matrix whose coefficients are optimized using genetic algorithms to ensure that the watermark is not visually perceivable and the integrity of the pixels in the watermarked image is minimally compromised in comparison to the original image.

Related work

In recent decades a digital revolution has occurred that no one could have imagined a few years ago. Massive digitalization has revolutionized the way we work and our social interactions (Woods and Gonzalez, 2002). While digitalization has made data easier to store and preserve, it has also posed cognitive issues of global significance, requiring massive computational infrastructure and efficient algorithms. The storing of textual documents in digital archives is a case in point. The prevailing trend was to scan text documents as images and submit them to storage repositories.

Now, we have tools and techniques that allow us to create digital texts that can be encoded using ASCII and UTF-8/16 (Djurovic et al., 2001). Scanned text documents are convenient to store but processing them is difficult. Text extraction from scanned images is not harrowing, but it is far from perfect. The massive production and storage of digital documents is an issue for organizations worldwide. The internet's advent has created new pathways for generating virtual text, with the web offering several alternatives (Cox et al., 2008). Web pages, social media sites, encyclopedias, and other internet sources are widespread. If new digital techniques are not investigated, the speed at which documents are produced could surpass our computational capabilities. In this paper, we investigate another aspect of the digital revolution which has to do with security in the form of watermarking.

A watermarking technique for identifying copyright infringement was developed by Komatsu and Tominaga (1989), specifically for digital entities. The idea of storing a watermark generated with spread spectrum methods and matrix transformations that was imperceptible in grayscale images and resistant to tempering was given by Boland et al. (1995). It has suggested a technique for embedding robust watermarks in images. Recent years have seen the development of watermarking methods in both frequency and spatial domains. Several methods have been used in spatial domain algorithms, including paired pixel manipulation, LSB substitution in the host images, and textured block coding. LSB substitution and several variations are the most effective and computationally efficient methods for embedding watermarks (Lu et al., 2003). However, LSB replacement is still not considered a robust algorithm, notwithstanding these attempts to improve it. A secure spatial domain technique was devised by Lin (2000) to survive challenging attacks, including JPEG compression.

The frequency-domain digital watermarking algorithms offer significantly more security than their spatial counterparts. The algorithms transform images from the time domain to the transform domain and then embed the watermark into the frequency domain (Abraham and Paul, 2017). Many of these methods involve the discrete Fourier transform (Candan et al., 2000), discrete cosine transform (Hernandez et al., 2000), or discrete wavelet transform (Xia et al., 1998). Zhang et al. (2020) effectively encrypted color images using a 2D discrete Fourier transform and a blind watermarking method. Fares et al. (2020) proposed a method for color images in the Fourier transform domain where embedding was carried out independently in each image plane. The discrete wavelet transform was effectively employed by Xia et al. (1997) to embed watermarks in multiresolution images. In this method, the significant coefficients in the high and medium frequency bands of the DWT image were assigned pseudo-random codes.

The fractional Fourier transform, and its applications have been explored extensively by researchers. One of the most well-known early introductions to the transformation is given by Namias (1980). He also suggested embedding watermarks in images using the technique of phase shift keying to make the watermarking more imperceptible. Kumar et al. (2013) created a blind digital image watermarking system based on the FrFT. The study proved that the FrFT could be used to provide good imperceptibility and resilience to complex JPEG compression attacks. Based on the FrFT, Mustafi and Ghorai (2013) proposed a new method for denoising medical images. The presented method uses blind source separation and fractional Fourier transform algorithm to eliminate noise from medical images, resulting in enhanced and robust denoising. Lang and Zhan (2014) proposed a new blind image watermarking method based on the FrFT for embedding a visually unidentifiable watermark into an image. The original cover image is divided into non-overlapping blocks for watermarking, each modified using a 2D fractional Fourier transform of two fractional orders. Kumari and Mustafi (2020) presented a straightforward digital watermarking method based on the fractional Fourier transform. This presented work provides a more secure information hiding technique that is robust, undetectable and has a more extensive data concealing capacity to meet the needs of each recipient across a broad spectrum of frequencies and spatial domains. Kumari and Mustafi (2021) developed a powerful image watermarking method based on the 2D discrete fractional Fourier transform. The method includes a twofold transform technique to improve its robustness to attacks. PSO was used to determine the best fractional ordering for embedding watermarks in the cover image. Kumari and Mustafi (2022) described an effective solution for image denoising using the FrFT. Images are denoised using a parallel set of filters and FrFT to extract the watermark. Simulations show that their approach is as accurate as current denoising techniques.

Recent watermarking techniques have also used nature-based algorithms (Naheed et al., 2014). Most watermarking techniques require lengthy searches to determine the ideal location for the watermark (in space or frequency domain). Due to local optima and high dimensionality, standard search strategies are often inadequate. Shieh et al. (2004) proposed a genetic algorithm-based watermarking approach in the transform domain. GA optimizes conflicting requirements. Watermarking using GA seems to be simple. They also evaluate their approach using GA's fitness function, which considers robustness and invisibility. Simulations illustrate GA's robustness under attacks and improvement in watermarked image quality. Wang et al. (2011) provided an optimal image watermarking methodology employing a multiobjective genetic algorithm in accordance with the multiobjective nature of image watermarking. A multiobjective genetic algorithm was employed to autonomously determine the optimal watermarking parameters, and a variable-length mechanism was used to seek the best watermark embedding locations. The optimization results indicate that multiobjective watermarking can increase the performance of watermarking algorithms without the problem of determining optimal parameters. Naheed et al. (2014) devised reversible watermarking to enhance embedding strength and invisibility. GA and PSO-based reverse interpolation watermarking provide for medical and standard images. Experimental data show that the suggested technique improves perceptual quality and the size of the watermark payload. An effective blind digital watermarking system based on a genetic algorithm is given by Alvarez et al. (2018). The experimental results suggest that, compared to previous approaches in the literature, the scheme maintains invisibility, security, and robustness more frequently. Calculations showed that the proposed watermarking method is robust to several attacks caused by salt and pepper, Gaussian noise, and jpeg compression.

Blind source separation (BSS) methodologies are used to separate audio, image, or any other source signal from a group of observation or mixed signals without identifying the mixing procedure and source signal characteristics (Sanchez, 2002). Separation is performed using several algorithms. However, BSS employing Non-Negative Matrix Factorization is frequently used. Non-Negative Matrix Factorization algorithms (Silva et al., 2020) still have difficulties with solution space convergence and separation quality. BSS was described by Belouchrani et al. (1997) as the recovery of a set of sources from a mixture without knowledge of the original signals or mixing technique. When a single source is recovered from several mixtures, the problem is called blind source separation (Choi et al., 2002). Recent studies by Silva et al. (2020) developed an output-only operational modal analysis method based on blind source separation. The method uses each pixel as a measurement point. This increases sensor density by orders of magnitude. Using extracted modal data, a simple method is provided to magnify and visualize independent vibration modes. The results show that the proposed technique can decompose, visualize, and rebuild weakly stimulated vibration modes.

Overview of fractional Fourier transform and blind source separation

The fractional Fourier transform coupled with BSS can provide an extremely efficient framework for developing watermarking and steganographic systems. The FrFT, with its ability to provide a singular domain space-frequency representation of a signal (Ozaktas et al., 2001), can quickly disperse an embedded watermark in the host entity to prevent localization attacks in the spatial domain. Such watermarks are usually challenging but simple to retrieve using blind source separation. The algorithm's simplicity is enhanced because BSS does not require any apriori knowledge at the retrieval end to extract the watermark. Consequently, the watermarking system is not only extremely robust but also simple to operate for the authorized user.

Fractional Fourier transform

The fractional Fourier transform (FrFT) is an extension of the Fourier transform (FT) with an additional degree of freedom $\alpha(0 \leq \alpha \leq 1)$, known as the order of the transform (Namias, 1980). This extra degree of freedom allows the FrFT to generate a powerful time (spatial) frequency signal representation. The FrFT has been compared to the short-term Fourier transform, Wigner Distribution, and wavelets in literature. As FrFT is a specialization of the Fourier transforms, it is also reversible and follows Parseval's theorem (Mustafi and Ghorai, 2013). Though not as closed and mathematically consistent as its analog counterpart, the discrete and two-dimensional version of the transform also exists under moderate limitations that are almost always satisfied.

The formal definition of the FrFT employs a forward transformation kernel K_α, which is defined in Eq. (1) (Bultheel and Sulbaran, 2004).

$$\mathbf{K}_\alpha\ (\mathbf{t,u}) = \begin{cases} \delta\ (\mathbf{t-u}) & \alpha \textbf{ is a multiple of } \mathbf{2\pi} \\ \delta\ (\mathbf{t+u}) & (\alpha+\pi) \textbf{ is a multiple of } \mathbf{2\pi} \\ \sqrt{\frac{\mathbf{1-jcot(\alpha)}}{\mathbf{2\pi}}}\,\mathbf{e}^{\mathbf{j}\left(\frac{\mathbf{u^2+t^2}}{2}\right)\cot(\alpha)-\mathbf{j\ ut\ cosec}(\alpha)} & \textbf{else} \end{cases} \tag{1}$$

Using this forward transformation kernel, the FrFT of order α is defined in the regular form as

$$\mathbf{F}_\alpha\ (\mathbf{u}) = \int_{-\infty}^{\infty} \mathbf{f}\,(\mathbf{t})\ \mathbf{K}_\alpha\ (\mathbf{t,u})\ \mathbf{dt} \tag{2}$$

The Euler representation, which is used to further simplify the expression in Eq. (1)

$$\sqrt{\frac{\mathbf{1-jcot(\alpha)}}{\mathbf{2\pi}}} = \sqrt{\frac{\mathbf{-je^{j\alpha}}}{\mathbf{2\pi\ sin(\alpha)}}} \tag{3}$$

According to Eq. (1), the Fourier transform is a special case of the FrFT coinciding with the first order (i.e., $\alpha = 1$) FrFT, whereas the zeroth order ($\alpha = 0$) FrFT is the signal's representation of the space domain. The FrFT provides a joint space-frequency signal representation for all other orders. Figure 1 is a visual representation of the FrFT's functioning.

The FrFT is simple to apply successively and mathematically; the successive application of the FrFT is denoted (Ozaktas et al., 2001) as

$$\mathbf{F}_{\alpha\mathbf{1}}(\mathbf{F}_{\alpha\mathbf{2}}) = \mathbf{F}_{\alpha\mathbf{1}+\alpha\mathbf{2}} \tag{4}$$

From Eq. (4), the computation of the inverse FrFT for the domain a is simply another FrFT with order $-\alpha$ i.e.,

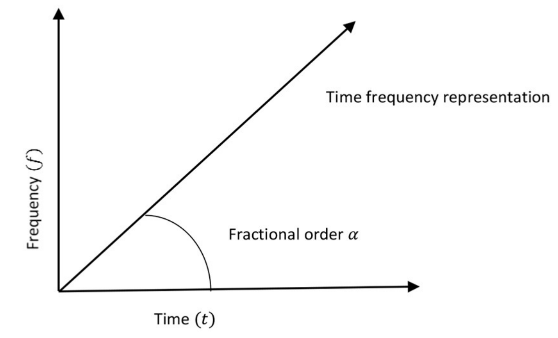


FIGURE 1
The time-frequency plane (Namias, 1980).

$$\mathbf{F}_{-\alpha}\ (\mathbf{F}_\alpha) = \mathbf{F}_{\alpha-\alpha} = \mathbf{F_0} = \mathbf{I} \tag{5}$$

Eq. (4) and Eq. (5) are the basis for developing the watermarking system. Interestingly, a discrete representation of the FrFT exists and can be expressed in terms of the discrete Hermite-Gaussian function (Kutay et al., 1997). The mathematical representation of the discrete form of the FrFT forward kernel is given as

$$\mathbf{F}^\alpha\ [\mathbf{m,n}] = \sum_{\mathbf{k=0,k\neq(N-1+(N)_2)}}^{\mathbf{N}} \mathbf{u_k}\ [\mathbf{m}]\ \mathbf{e}^{-\mathbf{j}\frac{\pi \mathbf{k}\alpha}{2}}\,\mathbf{u_k}[\mathbf{n}] \tag{6}$$

Where $u_k[n]$ is the *kth* discrete Hermite Gaussian function and $(N)_2 \equiv N\ mod2$

Blind source separation

Blind source separation refers to extracting source signals from a linear or non-linear mixture without any apriori knowledge about the sources (Yeredor, 2000). Mathematically we conceptualize the discrete linear BSS problem as

$$\mathbf{X = AS} \tag{7}$$

S is a matrix representing the collection of *N* source signals known to exist at *M* discrete points. Thus, *S* can be visualized as

$$\mathbf{S} = \left\{ \begin{matrix} \mathbf{S_{11}} & \mathbf{S_{12}} & \cdots & \mathbf{S_{1m}} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{S_{n1}} & \mathbf{S_{n2}} & \cdots & \mathbf{S_{nm}} \end{matrix} \right\} \tag{8}$$

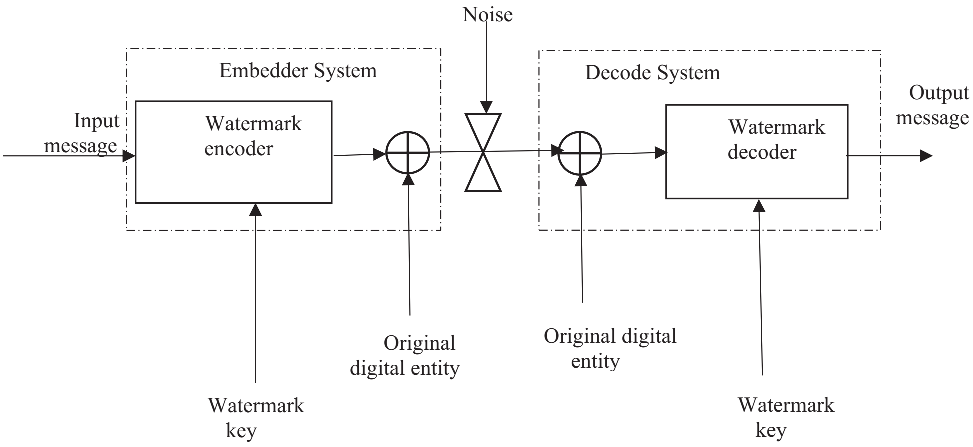


FIGURE 2
Watermark system architecture for informed watermarking (Cox et al., 2008).

The term A in Eq. (7) is an unknown matrix of dimension $n \times n$ made of real coefficients. The mixed signals output by the mixing signals are represented by the individual rows of the matrix X. It is obvious that X is a linear mixture of all the source symbols s_i, $1 \leq i \leq n$ and is a collection of signal mixtures observed by the receptors. The extension to the case of two-dimensional signals is straightforward. To solve the BSS problem, we must find the coefficients of the matrix S. However, in doing so, we work without any knowledge about A. Consequently, the only observed quantity available to us is matrix X. It can be observed that a solution to the linear BSS-problem reduces to finding the coefficient of the matrix A. Mathematically, it can be written as

$$\mathbf{X} = \mathbf{AS} \tag{9}$$

$$\mathbf{A}^{-1}\mathbf{X} = \mathbf{A}^{-1}\mathbf{AS} \tag{10}$$

$$\mathbf{S} = \mathbf{A}^{-1}\mathbf{X} \tag{11}$$

Thus, knowledge about the coefficients of A is sufficient to completely recover S, which is the goal of any BSS algorithm. The linear BSS problem is the one that has found the most relevance in real-world applications, though a lot of research has also focused on solving the non-linear BSS problem (Silva et al., 2020).

Various techniques have been proposed in the literature to solve the BSS problem (Song et al., 2019). Some more common approaches have focused on higher-order statistics or cumulants, the mutual information between the extracted sources, non-gaussianity of signals, principal component analysis, etc. In the present work, BSS has been used to extract the watermark from the watermarked image.

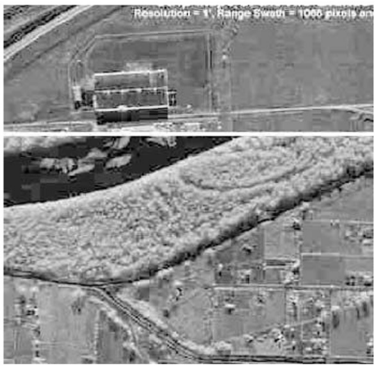
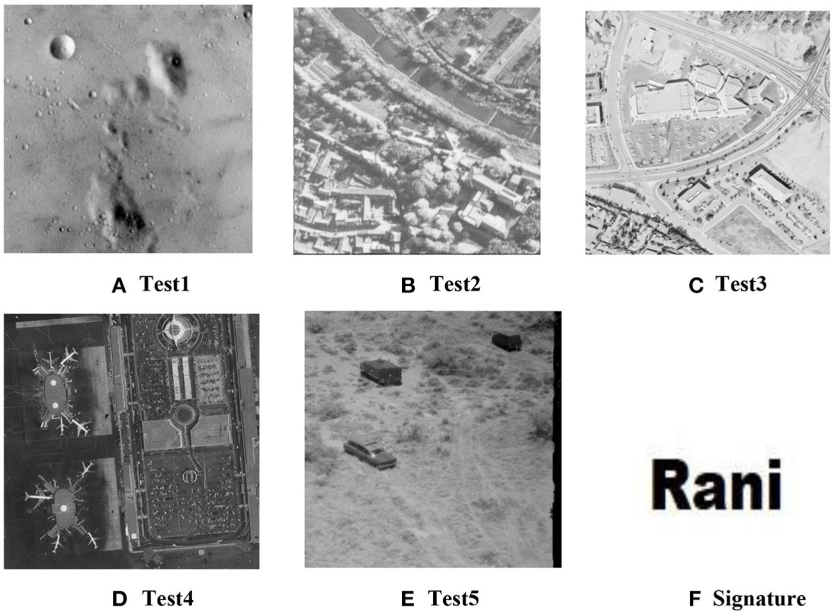


FIGURE 3
A SAR image of an agricultural field.

Proposed algorithm

Figure 2 shows the architecture of an informed watermarking system. Such systems are characterized by the fact that extraction of watermarks (Bo et al., 2011) necessitates the participation of the original watermarked digital entity or some variation. Thus, the system allows for *"source verification,"* which is an essential asset in ownership verification.

Figure 3 shows an example of a typical satellite image used in defense and intelligence. The image is typical as it shows several different features, all of which are distinguishable by their gray values, even though the image itself is only greyscale. The reader may also observe that the grab shows significant blocking effects and periodic noise symptoms. The common issues with images captured under non-optimal circumstances include sub-optimal



A Test1

B Test2

C Test3

D Test4

E Test5

F Signature

FIGURE 4
Cover images **(A–E)** and signature watermark image **(F)**.

environmental conditions, inappropriate lighting, acquisition device malfunction, etc. Digitally watermarking such images is a challenge due to several reasons. The induced watermark should result in a minimum variation of greyscale values of pixels, as these values are often interpreted automatically or semi-automatically. The watermark should also induce minimal distortion and artifacts, given the already impure nature of the images.

Even though it is common to embed random patterns (usually a collection of k random numbers) as watermarks in digital images, the retrieval process is cumbersome for many applications. In high-security application domains, it makes sense to embed visually perceptible watermarks (Wang et al., 2011) that can instantly be recognized in keeping with traditional paper watermarks. Thus, the proposed method embeds binary images as watermarks in the host image. One added advantage of this is that the method can be used for steganography even though such a use cannot be recommended except for the most trivial of cases. This is because the recovered watermark may lack integrity from the original watermark, which is not desirable for a robust steganography method. Figure 4 shows some of the other test images used in this paper and the watermark that was embedded in each of these test images.

Embedding procedure

The proposed method embeds the watermark image in the α^{th} FrFT domain of the host image. The embedding is done with the help of a mixing matrix A of size 2 × 2, as explained in Section Blind source separation. The process of embedding the watermark is illustrated using the flowchart shown in Figure 5. The signature watermark image is padded with zeros to be the same dimension as the target or host image. In our experiments, we found that scaling the signature image's greyscale to have the same mean as the target image increases the effectiveness of the algorithm.

Embedding can be done with the watermark when the target image is first transformed into the α^{th} FrFT domain, and then the scaled and padded watermark image is multiplicatively introduced into the FrFT domain using a mixing matrix A. The resultant modified FrFT is returned to the spatial domain using the inverse FrFT transform, equivalent to performing an FrFT with $-\alpha$. We see the resultant flowchart of watermark embedding in the target image, which found the output of two watermarked images. In an informed watermarking system, one of the resultant images is stored in a secure repository while the other can be distributed for use. By choosing a suitably high FrFT domain, the embedded watermark can be dispersed intricately in the space-frequency domain, making it very difficult to remove. In the experimental results presented, the FrFT domain chosen was $\alpha = 0.75$. Mathematically the embedding process is expressed (Belouchrani et al., 1997) as

$$\begin{bmatrix} \mathbf{W1} \\ \mathbf{W2} \end{bmatrix} = \mathbf{F}^{-\alpha}\left\{\begin{bmatrix} \mathbf{a}_{11}\&\mathbf{a}_{12} \\ \mathbf{a}_{21}\&\mathbf{a}_{22} \end{bmatrix}\begin{bmatrix} \mathbf{F}^{\alpha}\left\{\left|\mathbf{f}\left(\mathbf{x,y}\right)\right|\right\} \\ \left|\mathbf{h}\left(\mathbf{x,y}\right)\right| \end{bmatrix}\right\} \quad (12)$$

Where $W1$ and $W2$ are two distinct representations of the watermarked image, a_{ij} are the coefficients of the mixing matrix

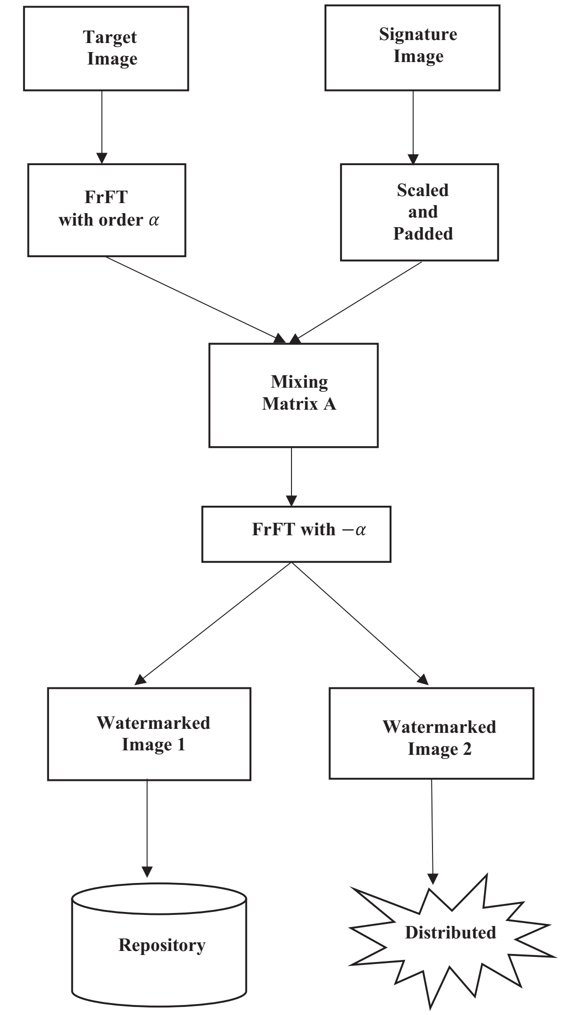


FIGURE 5
Flowchart of proposed system.

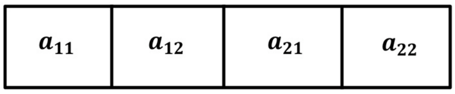


FIGURE 6
The chromosome representing the coefficients of the mixing matrix A.

and $|f(x,y)|$ and $|h(x,y)|$ are column matrix representations of the target image and watermark image zero-padded to be of equal length. F^{α} is the α^{th} order fractional Fourier transform as discussed in section Fractional Fourier transform.

Though much research has been devoted to the possibility of using the FrFT as a tool for the digital watermarking of images (Zhang et al., 2020), not much work has focused on using the BSS in conjunction with the FrFT. An ant colony approach to optimizing the FrFT coefficients has been discussed in Al-Qaheri et al. (2010). However, using both these tools, a high level of robustness can be provided to the watermarking process. While the FrFT can diffuse the watermark over the spatial-frequency domain simultaneously, the BSS can be used to recover the watermark back with minimal effort. However, optimizing the mixing matrix to allow the embedding of an entire image requires exact tuning of the mixing matrix. One of the issues in choosing a mixing matrix-based approach to embed the watermark image in an FrFT domain is the distortion that may be induced in the watermarked image once it is brought back to the spatial domain using the inverse FrFT transform. Without any constraints to guide the choice of the coefficients of matrix A, the watermarked image usually displays many distortions. The distortions can clearly be seen in the form of wave-like structures in the top half of the image.

To reduce such distortions, the matrix coefficients must be chosen carefully. Without any apriori knowledge regarding these coefficients, the method employs the genetic algorithm to choose the optimal set of coefficients.

Genetic algorithm-based coefficient optimization for mixing matrix

Genetic Algorithms (GAs) have long been considered an extremely efficient optimization tool for large search spaces. The functioning of these algorithms is based on the natural law of evolution and the concept of the *"survival of the fittest."* Genetic Algorithms iterate through generations by creating a population of candidates which tend to propagate the best traits of the previous generation of candidates (Goldberg, 1989). The algorithm employs several steps to ensure that the current population converges to an optimal solution, even over large search spaces having numerous variables. The mutation operator ensures that GAs is not trapped in local maxima and can quickly converge to global solutions even in large search spaces. GAs is considered ideal for cases where a near-optimal solution has to be established in a short period.

Basics of genetic algorithm

1. Initialize population
2. Create initial population
3. Evaluate individuals in the initial population.
4. Create new population
5. Select-fit individuals for reproduction
6. Generate offspring with genetic operator crossover.
7. Mutate offspring.
8. Evaluate offspring.

In the proposed method, the four coefficients of the mixing matrix $A = \begin{bmatrix} a_{11}\&a_{12} \\ a_{21}\&a_{22} \end{bmatrix}$, $0 < a_{pq} < \infty$ are used to create a multi-gene chromosome for use in the GA. A typical

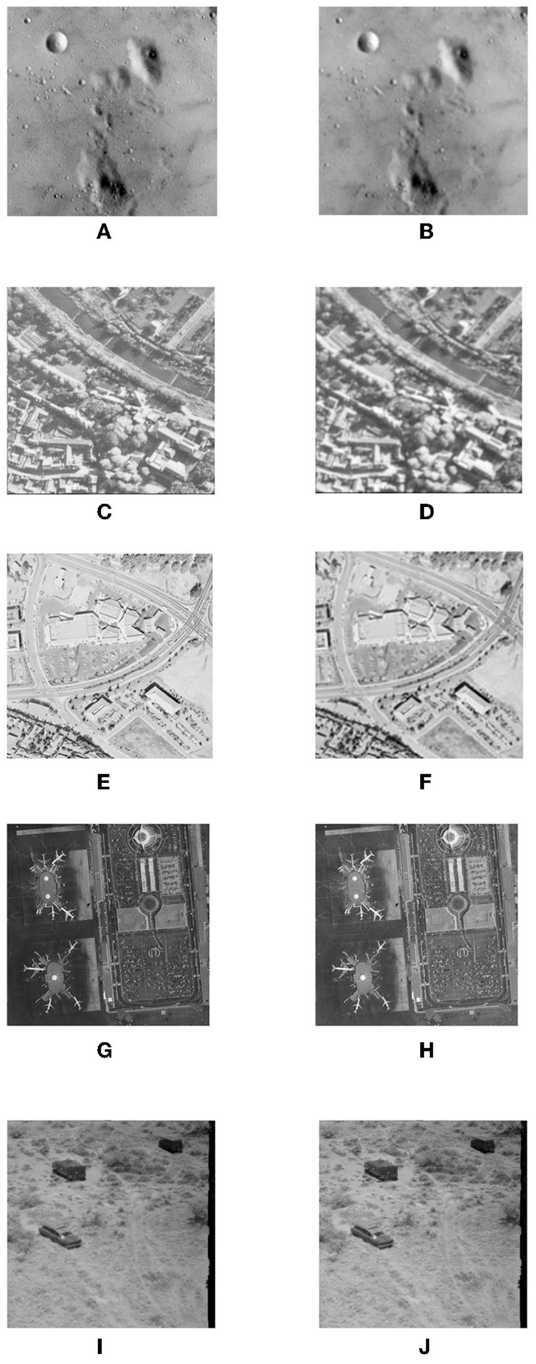


FIGURE 7
Watermarked images using GA-based mixing matrix coefficient optimization and its objective evaluation of imperceptibility for different images. **(A)** Original Test1. **(B)** Watermarked Image (PSNR = 51.2 and SSIM = 0.98). **(C)** Original Test2. **(D)** Watermarked Image (PSNR = 53.9 and SSIM = 0.95). **(E)** Original Test3. **(F)** Watermarked Image (PSNR = 53.9 and SSIM = 0.96). **(G)** Original Test4. **(H)** Watermarked Image (PSNR = 50.6 and SSIM = 0.96). **(I)** Original Test5. **(J)** Watermarked Image (PSNR = 53.8 and SSIM = 0.96).

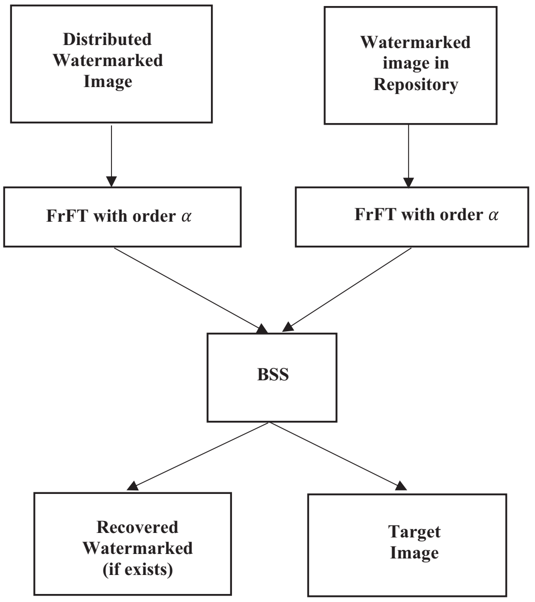

Distributed Watermarked Image
Watermarked image in Repository
FrFT with order α
FrFT with order α
BSS
Recovered Watermarked (if exists)
Target Image

FIGURE 8
Schematic representation of watermark extraction.

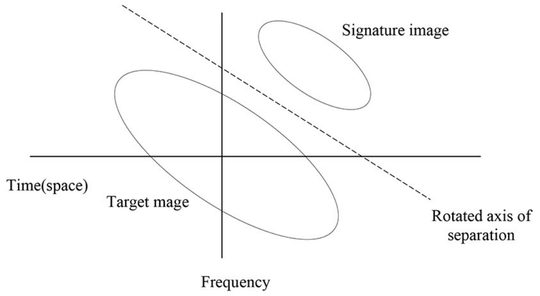


FIGURE 9
A view of the Wigner's plane in the watermarking scheme.

chromosome is shown in Figure 6 (Shieh et al., 2004). The upper limit of the coefficients was restricted to 500 for experimental purposes. The fitness function used to converge the search is defined as the average RMSE of the two images in comparison to the original target image and is mathematically defined as

$$\mathrm{F}=\frac{1}{2}\left[\sum_{\mathrm{i}=1}^{2}\sqrt{\frac{1}{\mathrm{MN}}\sum_{\mathrm{x}=0}^{\mathrm{N}-1}\sum_{\mathrm{y}=0}^{\mathrm{M}-1}\left[\mathrm{T}\left(\mathrm{x,y}\right)-\hat{\mathrm{T}}_{\mathrm{i}}(\mathrm{x,y})\right]^{2}}\right] \quad (13)$$

In Eq. (13), $T\left(x,y\right)$ represents the original image and $\hat{T}_i(x,y)$ represents one of the watermarked images. The images are considered of size $M \times N$ for generality, but in our experiments, only square images were used for the sake of computational

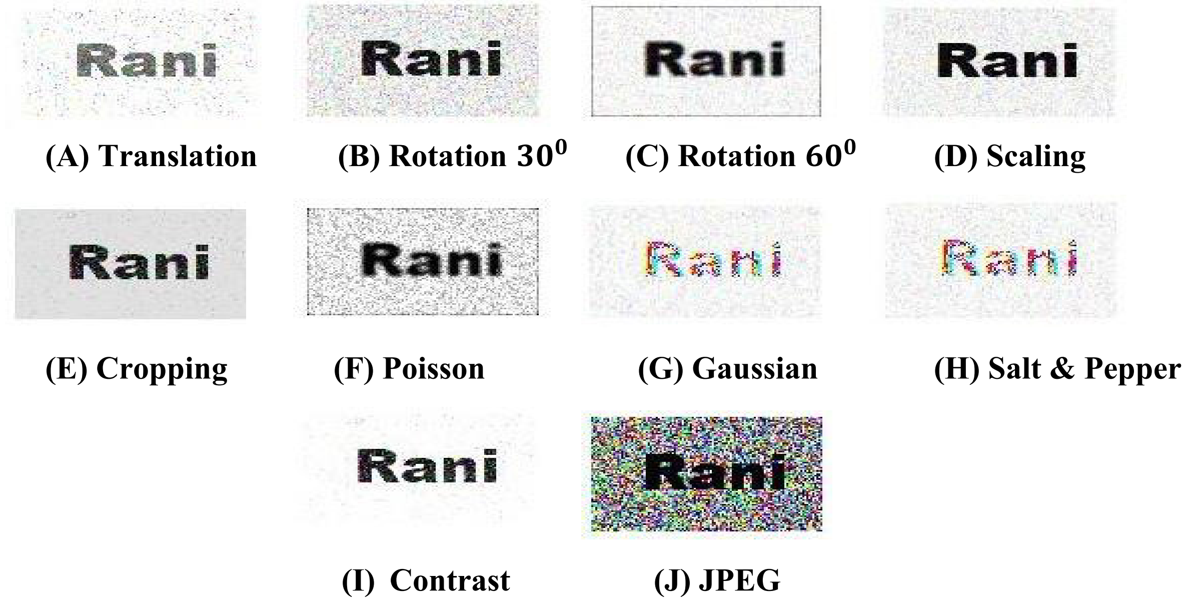


FIGURE 10
Schematic representation of extracted watermark after performing some geometric attacks. **(A)** Translation. **(B)** Rotation 30^0. **(C)** Rotation 60^0. **(D)** Scaling. **(E)** Cropping. **(F)** Poisson. **(G)** Gaussian. **(H)** Salt and Pepper. **(I)** Contrast. **(J)** JPEG.

simplicity. Using the GA-based approach (Wang et al., 2011), the mixing matrix used to embed the watermark produced minimal distortions in the watermarked images when using reasonably small-sized watermarked images. However, the distortions show a marked increase when the size of the watermark image is increased relative to the target image. As the sole aim of the signature image is to verify the authenticity and ownership of the target image, even small watermarks are more than sufficient for this purpose.

Figures 7B,D,F,H,J shows all watermarked images using the GA-based optimization method. The lack of wave-like distortions in the top half of the image is noticeable (compared with Figures 7A,C,E,G,I respectively). It is evident that the watermarked image is heavily distorted for any signature image larger than approximately 30% of the size of the target image. Very slight wave-like disturbances in the image are noticeable toward the top left corner of the image. However, they are quite insignificant in the context of the overall image.

Watermark extraction

The watermark extraction in the proposed method is straightforward to perform. The process inverts the extraction method and is depicted using the flowchart shown in Figure 8. It is interesting to note that the actual extraction employs the technique of BSS, and the knowledge of the mixing matrix A used at the encoding end is not required during the retrieval process. The FrFT domain α in which the watermark was embedded functions as the key in the watermarking scheme, as shown in Figure 2. Our experimental setup used the highly efficient FASTICA package to perform the BSS. Figure 10 shows the recovered watermark using the retrieval process. Even though traces of the frequency component of the target image are visible in the retrieved watermark, the extracted watermark has been extracted with remarkable clarity.

Due to its time (space) frequency capabilities embedding watermarks in the FrFT domains is a highly robust way of securing images. Figure 9 shows a visual representation of watermark embedding in the FrFT domain. The figure shows the target and signature images in the Wigner plane (Xia et al., 1997), with the two axes representing space and frequency, respectively. It is seen that the watermark image cannot be separated from the target image in either the spatial domain or the frequency domain alone. Thus, targeted attacks to remove the watermark in any of these domains are not likely to succeed, and this significantly increases the robustness of the method. Figure 9 also shows that the oblique FrFT axis, which corresponds to a rotation in the Wigner plane, can separate the two components, and it is in this domain that we perform BSS to extract the watermark. Thus, knowledge of the correct FrFT domain is essential in extracting the watermark. Figure 13 shows the PSNR of the recovered watermark for different FrFT domains. The actual embedding of the watermark was performed in the FrFT domain $\alpha = 0.75$. The plot clearly shows that the watermark has

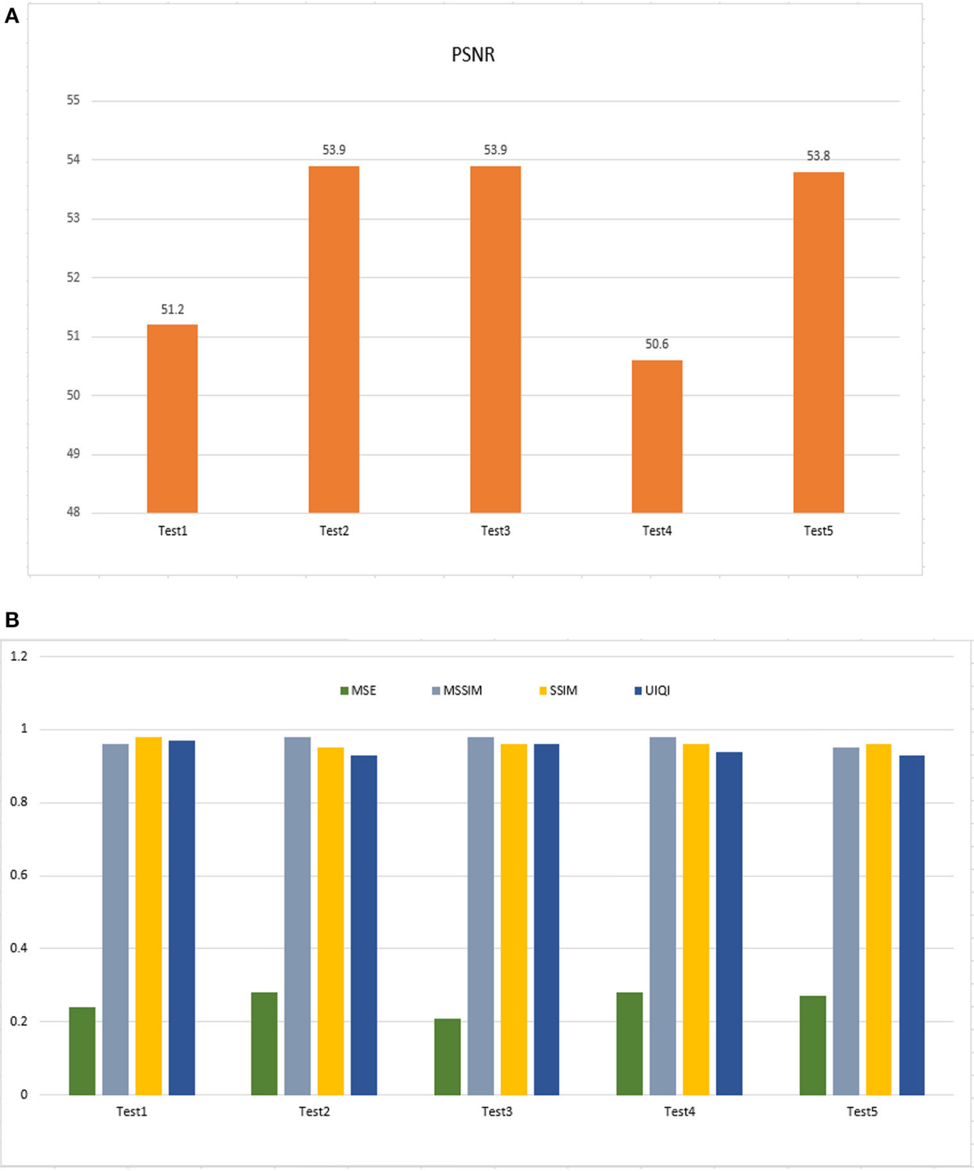


FIGURE 11
(A) Performance outcome of imperceptibility (PSNR). **(B)** The outcome of MSE (Mean Square Error), MSSIM (Mean Structure Similarity Index Measure), SSIM (Structure Similarity Index Measure), and UIQI (Universal Index, Quality Index).

failed to be extracted for all other domains except for the domain in which it was embedded.

Experimental results and discussions

To evaluate the robustness of the proposed watermarking method, the watermarked images were subjected to several common signal processing attacks (Lu et al., 2003). Due to the visual nature of the embedded watermark, the quality of the retrieved watermark is subject to human interpretation rather than statistical parameters, e.g., PSNR (Kumari and Mustafi, 2021) and RMSE (Alvarez et al., 2018). This is often an advantage for end-users. The experimental results show that the watermark can be successfully extracted at the retrieval end in almost all cases.

Figure 10 summarizes the performance of the method for various test cases of simulated attacks. In Figure 10 results, the watermarked image has been blurred using a Gaussian filter (Zhang et al., 2020). Such filters are very mild and do not have a significant abrasive effect on the image. The recovered watermark shows a high degree of clarity, as seen in

Figure 10G. However, the watermark is still reasonably extracted and is visually recognizable. A similar result is observed in the case of salt and pepper noise. The watermark is still perceptible even though salt and pepper noise distorts the entire frequency spectrum and often adversely affects many watermarking schemes. In Figure 10E, the watermarked image shows a simulated cropping effect, replacing a section of the image with black grayscale values. The crop position is intentionally chosen as standard embedding using a mixing matrix that would place the watermark at the top left of the target image. Even though the recovered watermark shows marked distortions, it is quite recognizable even by the naked eye, and the authenticity of the image can be validated.

As can be seen, the method is quite robust and can detect the watermark in the case of most attacks. In a few cases, like Gaussian filtering and JPEG compression (Naheed et al., 2014), the choice of the FrFT domain affected the performance of the method, and the retrieval was found to be more efficient in FrFT domains.

The proposed method can easily be extended to multiple plane formats like RGB images, where two alternative methods for embedding the watermark may be adopted. The watermark may be embedded in one of the three planes (which increases the robustness of the algorithm to a small extent), or the watermark image can be partitioned and embedded in all three planes. The second method is interesting as the choice of the FrFT domain α can be different for the three planes. Another possible improvement can be to choose an optimal FrFT domain to embed the watermark. Our experiments observed that the best results were obtained for the higher FrFT domains, but some domains performed better than others. The choice of the FrFT domain can again be performed using a heuristic or meta-heuristic algorithm, e.g., GA. However, even for randomly chosen non-optimized FrFT domains, the method is found to be highly competent.

Performance evaluation criteria

Imperceptibility, robustness, payload, and security are four attributes that determine the quality of an image watermarking scheme (Fares et al., 2020). Further, the algorithmic complexity is also often considered an important parameter while judging the efficiency of a watermarking algorithm.

Quality metrics

While the quality of an image watermarking scheme can be judged by the human visual system (HVS) using our latent sense of perception (Shih, 2017), several mathematical techniques have been suggested in the literature to measure the performance of an image watermarking scheme quantitatively.

TABLE 1 Evaluation of different geometric attacks with their respective robustness results.

Attacks type	WPSNR	NCC	SM	BER
Translation	50.8	1	1	0
Rotation **30^0**	52.4	1	1	0
Rotation **60^0**	51.5	1	1	0.114
Scaling	50.48	1	1	0
Cropping	49.6	1	1	0
Poisson	51.3	0.95	1	0
Gaussian	51.3	1	1	0.14
Salt and pepper	50.8	0.93	1	0
Speckle	51.3	0.94	1	0
Contrast	51.8	1	1	0
JPEG (q = 100)	50.12	1	1	0

In the current work, we have employed four conventional performance metrics to evaluate the imperceptibility and robustness of the proposed algorithm. These quality metrics (Woods and Gonzalez, 2002) are peak signal-to-noise ratio (PSNR), structural similarity index measurement (SSIM), normalized cross-correlation (NC), and bit error rate (BER). Among these, PSNR and SSIM have been used to evaluate the imperceptibility of a digital watermark, while NC and BER test the robustness of the proposed method. A brief description of these quality metrics is provided in the following sections.

Imperceptibility and capacity test

The tests outlined in the previous section were performed to evaluate the proposed method. Figure 11A shows PSNR values for experimental images. PSNR readings remain high, proving the watermark is imperceptible. Figure 11B shows MSE values between 0.21 and 0.28, indicating a minimal loss in watermarked image quality. Maximum UIQI values are close to 1 (0.93–0.97). This illustrates that watermarked images always seem to be like the originals. Regarding structural similarity, the original and watermarked are comparable, and the highest value for both SSIM and MSSIM is 0.98, indicating high perceptual quality.

Robustness test

Robustness can be determined by examining the extracted watermark after the watermarked image has been attacked (Shih, 2017). We evaluated the algorithm against geometric attacks. Table 1 depicted the watermark and extracted the watermark's robustness after some attacks. In Figure 12A, we had shown the WPSNR values for the experiment conducted. In Figure 12B, performance outcome of NCC (Normalized Cross-Correlation),

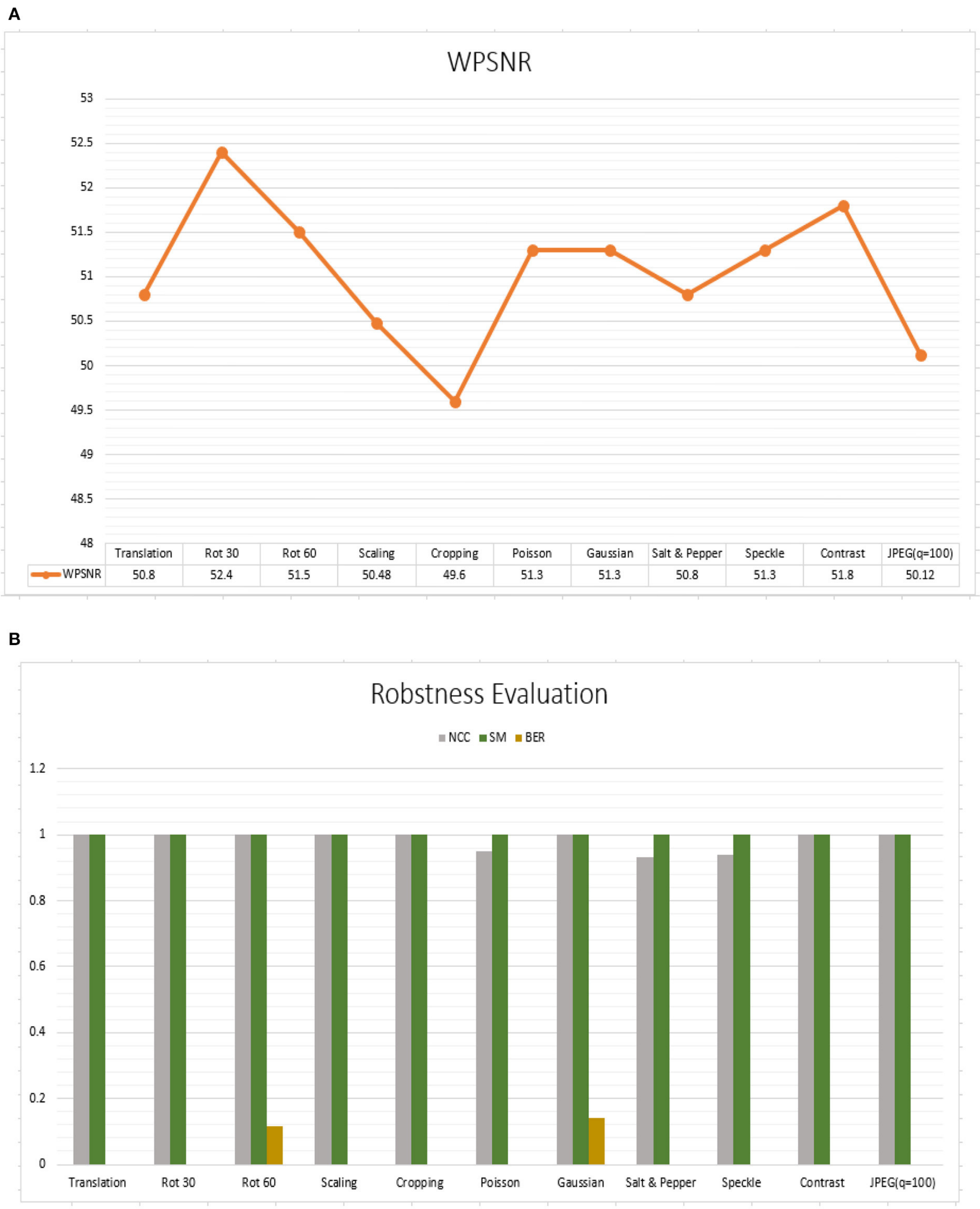


FIGURE 12
(A) Performance outcome of robustness (WPSNR), **(B)** Performance outcome of NCC (Normalized Cross-Correlation), SM (Similarity Measurement), BER (Bit Error Rate).

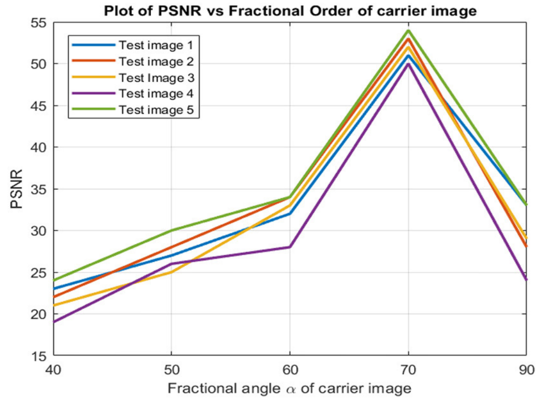


FIGURE 13
PSNR results of images with different fractional order α.

SM (Similarity Measurement), BER (Bit Error Rate) has been shown. The table demonstrates that the maximum WPSNR value is 52db which is good. NCC results are also excellent, except for Poisson, Salt & Pepper, and Speckle. NC > 0.93 means the original and extracted watermarks are similar. The SM (Similarity Measurement) also reports promising findings. Except for Rotation and Gaussian noise addition, the Bit Error Rate is less. The proposed algorithm is resilient against attacks.

The PSNR values acquired across various fractional orders have also been evaluated, and we found that, most often, the best PSNR was obtained almost at a rotation angle of 30°. In Figure 13, the PSNR value of the watermarked image gradually decreased on each side. The optimal embedding, according to this, occurs in the higher fractional orders. The optimal embedding fractional order had to be determined manually for each image, resulting in one of the drawbacks of the present work. This is an additional computing load, and research efforts may be directed toward developing more effective techniques for determining the appropriate fractional order.

Conclusion

In this paper, a novel informed watermarking technique for digital images has been proposed. The method uses the fractional Fourier transform and BSS to embed and extract the watermark. The embedded watermark is intentionally chosen to be visually recognizable to make the retrieval and identification process more conducive for typical end-users. The method also utilizes GA to optimize the embedding phase, ensuring that the watermark can be embedded in the target image with minimum distortions. Further work to provide RST invariance to the method only make the technique more robust. Currently, the usefulness of the Log polar transform is being explored to provide the necessary RST invariance property to the method. Additionally, more research must be conducted to optimize the process more robustly when faced with significant image cropping, in which case the watermark recovery is significantly hampered.

According to the results presented, the method works highly efficiently for the average case and is also very robust against many known signal processing attacks. Another important consideration while evaluating the algorithm's robustness is that the retrieval process depends stringently on correctly identifying the FrFT domain. For all other domains, the watermark stays hidden and thus does not lend itself to passive attacks or masquerades.

Data availability statement

The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Author contributions

RK: original draft writing, conceptualization, methodology, validation, and software. AM: conceptualization, methodology, validation, research, writing-review and editing, and supervision. All authors contributed to the article and approved the submitted version.

Acknowledgments

I would like to express my sincere gratitude and appreciation to BIT Mesra in Ranchi, India, for their financial support.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abraham, J., and Paul, V. (2017). "A dual domain digital image watermarking scheme," in *Conference of IEEE Trans.* 978-1-5090-6106. doi: 10.1109/ICICICT1.2017.8342541

Al-Qaheri, H., Mustafi, A., and Banerjee, S. (2010). Digital watermarking using ant colony optimization in fractional fourier domain. *J. Inf. Hiding Multimed. Signal Process.* 1, 179–189. Available online at: http://www.jihmsp.org/~jihmsp/2010/vol1/JIH-MSP-2010-03-003.pdf

Alvarez, V., Armario, J. A., Frau, M. D., Gudiel, F., Guemes, M. B., Martin, E., et al. (2018). GA based robust blind digital watermarking. *Electron. Notes Discrete Math.* 68, 149–154. doi: 10.1016/j.endm.2018.06.026

Belouchrani, A., Meraim, K. A., Cardoso, J. F., and Moulines, E. (1997). A blind source separation technique using second-order statistics. *IEEE Trans. Signal Process.* 45, 434–444. doi: 10.1109/78.554307

Bo, W., Cui, X. M., and Zhang, C. (2011). "Realization of digital image watermarking encryption algorithm using fractional Fourier transform," in *Strategic Technology (IFOST), 2011 6th International Forum on,* 2, 822–825. doi: 10.1109/IFOST.2011.6021147

Boland, F. M., Ruanaidh, J. J. K. O., and Dautzenberg, C. (1995). "Watermarking digital images for copyright protection image processing and its applications," in *4-6 July Conference Publication No.410.* doi: 10.1049/cp:19950674

Bultheel, A., and Sulbaran, H. E. M. (2004). Computation of the fractional fourier transform. *Appl. Comput. Harmon. Anal.* 16, 182–202. doi: 10.1016/j.acha.2004.02.001

Candan,. C., Kutay, M. A., and Ozaktas, H. M. (2000). The discrete fractional fourier transform. *IEEE Trans. Signal Process.* 48, 1329–1337. doi: 10.1109/78.839980

Choi, S., Cichocki, A., and Amari, S. (2002). Equivariant nonstationary source separation. *Neural Networks* 15, 121–130. doi: 10.1016/S0893-6080(01)00137-X

Cox, I., Miller, M., Bloom, J., Fredrich, J., and Kalker, T. (2008). *Digital Watermarking and Steganography, Second Edition.* Elsevier Science. doi: 10.1016/B978-012372585-1.50015-2

Djurovic, I., Stankovic, S., and Pitas, I. (2001). Digital watermarking in the fractional Fourier transform domain. *J. Netw. Comput. Appl.* 24, 167–173. doi: 10.1006/jnca.2000.0128

Fares, K., Amine, K., and Salah, E. (2020). A robust blind color image watermarking based on Fourier transform domain. *Int. J. Ligh Electron Optics* 208, 164562. doi: 10.1016/j.ijleo.2020.164562

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley Longman Publishing Co. USA.

Hernandez, J. R., Amado, M., and Gonzalez, F. P. (2000). DCT-domain watermarking techniques for still images: Detector performance analysis and a new structure. *Image Process. IEEE Trans.* 9, 55–68. doi: 10.1109/83.817598

Komatsu, N., and Tominaga, H. (1989). A proposal on digital watermark in document image communication and its application to realizing a signature. *Trans. Electron. Inform. Commun.* 73, 22–33. doi: 10.1002/ecja.4410730503

Kumar, M., and Rewani, R., Aman (2013). "Digital image watermarking using fractional fourier transform via image compression," in *IEEE International Conference on Computational Intelligence and Computing Research.* doi: 10.1109/ICCIC.2013.6724174

Kumari, R., and Mustafi, A. (2020). *Embedding image watermarks in carrier images using the fractional Fourier transform,* 25741–25751.

Kumari, R., and Mustafi, A. (2021). An optimized framework for digital watermarking based on multi-parameterized 2D-FrFT using PSO. *Optik Int. J. Light Electron.* Optics 248, 168077. doi: 10.1016/j.ijleo.2021.168077

Kumari, R., and Mustafi, A. (2022). "Denoising of images using fractional fourier transform," in *2022 2nd International Conference on Emerging Frontiers in Electrical and Electronic Technologies (ICEFEET),* 978-1-6654-8875-4. doi: 10.1109/ICEFEET51821.2022.9848244

Kutay, M. A., Ozaktas, H. M., Arikan, O., and Onural, L. (1997). Optimal filter in fractional fourier domains. *IEEE Trans. Signal Process.* 45, 1129–1143. doi: 10.1109/78.575688

Lang, J., and Zhan, Z. G. (2014). Blind digital watermarking method in the fractional Fourier transform domain. *Opt. Lasers Eng.* 53, 112–121. doi: 10.1016/j.optlaseng.2013.08.021

Lin, P. H. (2000). Robust transparent image watermarking system with spatial mechanisms. *J. System Softw.* 50, 107–116. doi: 10.1016/S0164-1212(99)00081-3

Lu, H., Shen, R., and Chung, F. L. (2003). Fragile watermarking scheme for image authentication. *Electron. Lett.* 39, 898–900. doi: 10.1049/el:20030589

Mustafi, A., and Ghorai, S. K. (2013). A novel blind source separation technique using fractional fourier transform for denoising medical images. Optik 124, 265–271. doi: 10.1016/j.ijleo.2011.11.052

Naheed, T., Usman, I., Khana, T. M., Dara, A. H., and Shafique, M. F. (2014). Intelligent reversible watermarking technique in medical images using GA and PSO. *Optik J.* 0030–4026. doi: 10.1016/j.ijleo.2013.10.124

Namias, V. (1980). The fractional order fourier transform and its application to quantum mechanics. *J. Inst. Math. Appl.* 25, 241–265. doi: 10.1093/imamat/25.3.241

Ozaktas, H. M., Zalevsky, Z., and Kutay, M. A. (2001). *The Fractional Fourier Transform with Applications in Optics and Signal Processing.* John Wiley and Sons Ltd. doi: 10.23919/ECC.2001.7076127

Sanchez, V. D. (2002). Frontiers of research in bss/ica. *Neurocomputing* 49, 7–23. doi: 10.1016/S.0925-2312(02)00533-7

Shieh, C. -S., Huang, H. -C., Wang, F. -H., and Pana, J. -S. (2004). Genetic watermarking based on transform-domain techniques. *Pattern Recog*nition 37, 555–565. doi: 10.1016/j.patcog.2003.07.003

Shih, F. Y. (2017). *Digital watermarking and steganography: Fundamentals and techniques. (2nd edition).* ISBN 13: 978-1-4987-3876-7 Hardback. doi: 10.1201/9781315121109

Silva, M., Martinez, B., Figueiredo, E., Costa, J. C. W. A., Yang, Y., and Mascarenas, D. (2020). Nonnegative matrix factorization-based blind source separation for full-field and high-resolution modal identification from video. *J. Sound Vibration.* 487, 115586. doi: 10.1016/j.jsv.2020.115586

Song, R., Zhang, S., Cheng, J., Li, C., and Chen, X. (2019). New insights on super-high resolution for video-baseds heart rate estimation with a semi-blind source separation method. *Comput. Biol. Med.* 116, 103535. doi: 10.1016/j.compbiomed.2019.103535

Wang, J., Peng, H., and Shi, P. (2011). An optimal image watermarking approach based on a multiobjective genetic algorithm.. *Inform. Sci.* 18, 5501–5514. doi: 10.1016/j.ins.2011.07.040

Woods, R. E., and Gonzalez, R. C. (2002). *Digital Image Processing. Pearson Education Asia. (3rd edition).*

Xia, X. G., Boncelet, C., and Arce, G. (1998). Wavelet transform based watermark for digital images. *Optics Express* 3, 497–511. doi: 10.1364/OE.3.000497

Xia, X. G., Boncelet, C. G., and Arce, G. R. (1997). "A multiresolution watermark for digital images," in *Proceedings of the International Conference on Image Processing.* (Santa Barbara, CA), 548–551. doi: 10.1109/ICIP.1997.647971

Yeredor, A. (2000). Blind separation of gaussian sources via second-order statistics with asymptotically optimal weighting. *IEEE Signal Process.* 7, 197–200. doi: 10.1109/97.847367

Zhang, X., Su, Q., Yuan, Z., and Liu, D. (2020). An efficient blind color image watermarking algorithm in spatial domain combining discrete Fourier transform. *Int. J. Light Electron Optics* 219, 165272. doi: 10.1016/j.ijleo.2020.165272

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 01 December 2022
DOI 10.3389/fnbot.2022.1082346

OPEN ACCESS

EDITED BY
Xin Ning,
Institute of Semiconductors (CAS), China

REVIEWED BY
Sahraoui Dhelim,
University College Dublin, Ireland
Nadeem Javaid,
COMSATS University, Islamabad Campus, Pakistan
Achyut Shankar,
Amity University, India

*CORRESPONDENCE
Yuchun Chang
cyc@dlut.edu.cn

RECEIVED 28 October 2022
ACCEPTED 14 November 2022
PUBLISHED 01 December 2022

CITATION
Dai J, Fu Y, Wang S and Chang Y (2022) Siamese hierarchical feature fusion transformer for efficient tracking. *Front. Neurorobot.* 16:1082346. doi: 10.3389/fnbot.2022.1082346

Siamese hierarchical feature fusion transformer for efficient tracking

Jiahai Dai[1], Yunhao Fu[1], Songxin Wang[2] and Yuchun Chang[1,3]*

[1]Department of Electronic Information Engineering, College of Electronic Science and Engineering, Jilin University, Changchun, China, [2]Department of Computer Science and Technology, College Science and Technology, Shanghai University of Finance and Economics, Shanghai, China, [3]Department of Electronic Science and Technology, School of Microelectronics, Dalian University of Technology, Dalian, China

Object tracking is a fundamental task in computer vision. Recent years, most of the tracking algorithms are based on deep networks. Trackers with deeper backbones are computationally expensive and can hardly meet the real-time requirements on edge platforms. Lightweight networks are widely used to tackle this issue, but the features extracted by a lightweight backbone are inadequate for discriminating the object from the background in complex scenarios, especially for small objects tracking task. In this paper, we adopted a lightweight backbone and extracted features from multiple levels. A hierarchical feature fusion transformer (HFFT) was designed to mine the interdependencies of multi-level features in a novel model—SiamHFFT. Therefore, our tracker can exploit comprehensive feature representations in an end-to-end manner, and the proposed model is capable of handling small target tracking in complex scenarios on a CPU at a rate of 29 FPS. Comprehensive experimental results on UAV123, UAV123@10fps, LaSOT, VOT2020, and GOT-10k benchmarks with multiple trackers demonstrate the effectiveness and efficiency of SiamHFFT. In particular, our SiamHFFT achieves good performance both in accuracy and speed, which has practical implications in terms of improving small object tracking performance in the real world.

KEYWORDS

visual tracking, hierarchical feature, transformer, lightweight backbone, real-time

Introduction

Visual tracking is an important task in computer vision that provides underlying technical support for more complex tasks; and is an essential procedure for advanced computer vision applications. Additionally, visual tracking has been widely used in various fields such as unmanned aerial vehicles (UAVs) (Cao et al., 2021), autonomous driving (Zhang and Processing, 2021), and video surveillance (Zhang G. et al., 2021). However, several challenges remain that hamper tracking performance, including edge computing devices and difficult external environments with occlusion, illumination variation, and background clutter.

Over the past few years, visual object tracking has made significant advancements based on the development of convolutional neural networks due to the breakthroughs that have been made to generate more powerful backbones, such as deeper networks (He et al., 2016; Chen B. et al., 2022), efficient network structure (Howard et al., 2017), attention mechanism (Hu et al., 2018). Inspired by the way of the human brain process the overload information (Wolfe and Horowitz, 2004), the attention mechanism is utilized to enhance the vital features and surpass the unnecessary information of the input feature. Due to the powerful feature representation ability, the attention mechanism becomes an important means to enhance the input features, such as channel attention (Hu et al., 2018), spatial attention (Wang F. et al., 2017; Wang N. et al., 2018), temporal attention (Hou et al., 2020), global attention (Zhang et al., 2020a), and self-attention mechanism (Wang et al., 2018). Among them, the self-attention based models, the transformer was initially designed for natural language processing (NLP) (Vaswani et al., 2017) task, where the attention mechanism is utilized to perform the machine translation tasks and achieved great improvements. Later, the pre-training model BERT (Devlin et al., 2018) achieve breakthrough progress in NLP tasks, further advance the development of the Transformer model. Since then, both academia and industry have set off a boom in the research and application of pre-trained models based on Transformer, and gradually extended from NLP to CV. For example, Vision Transformer (ViT) (Dosovitskiy et al., 2020), DETR (Carion et al., 2020), have surpassed previous SOTA in the fields of image classification, inspection, and video, respectively. Various variant models based on Transformer structure have been proposed, multi-task indicators in various fields have been continuously refreshed, and the deep learning community has entered a new era. Meanwhile, muti-level features fusion can effectively alleviate the deficiency of the transformer in handling the tracking of small objects.

Although transformer models provide enhancements in feature representation and result in promotion in terms of accuracy and robustness, trackers based on transformers have high computational costs that hinder them from meeting the real-time demands of tracking tasks on edge hardware devices, providing a disadvantage for the landing of the application. Therefore, how to balance the efficiency and efficacy of object trackers remains a significant challenge. Generally, discriminative feature representation is essential for tracking. Therefore, deeper backbones and online updaters are utilized in tracking frameworks, however these methods are computationally expensive leading to increased run time and budget. Typically, the lightweight backbone is also limited as it typically provides inadequate feature extraction, rendering the tracking model less robust for small objects or complex scenarios.

In this study, we employed a lightweight backbone network to avoid the efficiency loss caused by the computations of deep networks. To address the insufficient feature representations extracted by shallow networks, we extracted features from multiple levels of the backbone to enrich the feature representations. Furthermore, to leverage the advantages of transformers in global relationship modeling, we designed a hierarchical feature fusion module to integrate multi-level features comprehensively using multi-head attention mechanisms. The proposed Siamese hierarchical feature fusion transformer (SiamHFFT) tracker achieved robust performance in complex scenarios while maintaining real-time tracking speed on a CPU and it can be deployed on consumer CPUs. The main contributions of this study can be summarized as follows:

(1) We proposed a novel type of tracking network based on a Siamese architecture, which consisting of feature extraction, reshape module, Transformer-like feature fusion module, and head prediction modules.
(2) We designed a feature fusion transformer to exploit the hierarchical features in the Siamese tracking framework in an end-to-end manner, which is capable of advancing discriminability for small object tracking task.
(3) Comprehensive evaluations on five challenging benchmarks demonstrate the proposed tracker achieved promising results among state-of-the-art trackers. Besides, our tracker can run at a real-time speed. This efficient method can be deployed on resource-limited platforms.

The remainder of this paper is organized as follows. Section Related work describes related work on tracking networks and transformers. Section Method introduces the methodology used for implementing the proposed HFFT and network model. Section Experiments presents the results of experiments conducted to verify the proposed model. Finally, Section Conclusion contains our concluding remarks.

Related work

Siamese tracking

In recent years, Siamese-based networks have become a ubiquitous framework in the visual tracking field (Javed et al., 2021). Tracking an arbitrary object can be considered as learning similarity measure function learning problems. SiamFC (Bertinetto et al., 2016) introduced a correlation layer as a fusion tensor into the tracking framework for the first time, which pioneered the Siamese tracking procedure. Instead of directly estimating the target position according to the response map, SiamRPN (Li B. et al., 2018) attaches a region proposal extraction subnetwork (RPN) to the Siamese network and formulates the tracking as a one-shot detection task. Based on the results of classification and regression branches, SiamRPN achieves enhanced tracking accuracy. DaSiamRPN (Zhu et al.,

2018) uses a distractor-aware module to solve the problem of inaccurate tracking caused by the imbalance of positive and negative samples of the training set. C-RPN (Fan and Ling, 2019) and Cract (Fan and Ling, 2020) incorporate multiple stages into the Siamese tracking architecture to improve tracking accuracy. To address unreliable predicted fixed-ratio bounding boxes when a tracker drifts rapidly, an anchor-free mechanism was also introduced into the tracking task. To rectify the inaccurate bounding box estimation strategy of the anchor-based mechanism, Ocean (Zhang et al., 2020b) directly regresses the location of each point located in the ground truth. SiamBAN (Chen et al., 2020) adopts box adaptive heads to handle the classification and regression problem parallelly. SiamFC++ (Xu et al., 2020) and SiamCAR (Guo et al., 2020) draw on the FCOS architecture and add a branch to measure the accuracy of the classification results. Compared with anchor-based trackers, anchor-free-based trackers utilize fewer parameters and do not need prior information for the bounding box, these anchor-free-based trackers can achieve a real-time speed.

As feature representation plays a vital role in the tracking process (Marvasti-Zadeh et al., 2021), several works delicate to obtain discriminative features from different perspectives, such as adopting deeper or wider backbones, and using attention mechanisms to advance the feature representation. In the recent 3 years, the Transformer is capable of using global context information and preserving more semantic information. The introduction of the Transformer model in the tracking community boots the tracking accuracy to a great extent (Chen X. et al., 2021; Lin et al., 2021; Liu et al., 2021; Chen et al., 2022b; Mayer et al., 2022). However, the promotion of the accuracy of these trackers' increasingly complex models relies heavily on powerful GPUs, leading to the inability to deploy such models on edge devices, which hinders the further practical application of the models.

In this study, to optimize the trade-off between tracking accuracy and speed, we designed an efficient algorithm that employs a concise model consisting of a lightweight backbone network, a feature reshaping model, a feature fusion module, and a prediction head. Our model is capable of handling complex scenarios, and the proposed tracker can also achieve real-time speed on a CPU.

Transformer in vision tasks

As a new type of neural network, transformer shows superior performance in the field of AI applications (Han et al., 2022). Unlike the structure of CNNs and RNNs, Transformer adopts the self-attention mechanism, which has been proved to have strong feature representation ability and better parallel computing capability, making it more advantageous in several tasks.

The transformer model was first proposed by Vaswani et al. (2017) for application to natural language processing (NLP) tasks. In contrast to convolutional neural networks (CNNs) and recurrent neural networks (RNNs), self-attention facilitates both parallel computation and short maximum path lengths. Unlike earlier self-attention models based on RNNs for input representations (Lin Z. et al., 2017; Paulus et al., 2017), the attention mechanisms in transformer model are implemented with attention-based encoders and decoders instead of convolutional or recurrent layers.

Because transformers were originally designed for sequence-to-sequence learning on textual data and have exhibited good performance, their ability to integrate global information has been gradually unveiled and transformers have been extended to other modern deep learning applications such as image classification (Liu et al., 2020; Chen C. -F. R. et al., 2021; He et al., 2021), reinforcement learning (Parisotto et al., 2020; Chen L. et al., 2021), face alignment (Ning et al., 2020), object detection (Beal et al., 2020; Carion et al., 2020), image recognition (Dosovitskiy et al., 2020) and object tracking (Yan et al., 2019, 2021a; Cao et al., 2021; Lin et al., 2021; Zhang J. et al., 2021; Chen B. et al., 2022; Chen et al., 2022b; Mayer et al., 2022). Based on CNNs and transformers, the DERT (Carion et al., 2020) applies a transformer to object detection tasks. To improve upon previous CNN models, DERT eliminates post-processing steps that rely on manual priors such as non-maximum suppression (NMS) and anchor generators; and constructs a complete end-to-end detection framework. ViT (Dosovitskiy et al., 2020) mainly converts images into serialized data through token processing and introduces the concept of patches, where input images are divided into smaller patches and each patch is converted into a bidirectional encoder representation from transformers-like structure. Similar to the concept of patches in ViT, Swin Transformer (Liu et al., 2021) uses the concept of windows, but the calculations of different windows do not interfere with each other, hence, the computational complexity of the Swin Transformer is significantly reduced.

In the tracking community, transformers have achieved remarkable performance. STARK (Yan et al., 2021a) utilizes an end-to-end transformer tracking architecture based on spatiotemporal information. SwinTrack (Lin et al., 2021) incorporates a general position-encoding solution for feature extraction and feature fusion, enabling full interaction between the target object and search region during tracking process. TrTr (Zhao et al., 2021) used the transformer architecture to perform target classification and bounding box regression and designed a plug-in online update module for classification to further improve tracking performance. DTT (Yu et al., 2021) also feed these architectures to predict the location and the bounding box of the target. Cao et al. (2021) proposed an efficient and effective hierarchical feature transformer (HiFT) for aerial tracking. HCAT (Chen et al., 2022b) utilizes a novel feature sparsification module to reduce computational complexity and a hierarchical cross-attention transformer that employs a full cross-attention structure to improve efficiency and enhance representation ability. The hierarchical-based methods, both HiFT and HCAT show good tracking performance. However, transformer-based trackers lack robustness in small objects. In this paper, we propose a novel hierarchical feature fusion module based on a transformer to enable a tracker to achieve real-time speed while maintains good accuracy.

Feature aggregation network

Feature aggregation plays a vital role in the multi-level feature process, and is used to improve cross-scale feature interaction and multi-scale feature fusion, thereby enhancing the representation of features and enhancing network performance. Zhang G. et al. (2021) proposed a hierarchical aggregation transformer (HAT) framework consisting of transformer-based feature calibration (TFC) and deeply supervised aggregation (DSA) modules. The TFC module can merge and preserve semantic and detail information at multiple levels, and the DSA module aggregates the hierarchical features of the backbone with multi-granularity supervision. Feature pyramid networks (FPN) (Lin T.-Y. et al., 2017) introduce cross-scale feature interactions and achieve good results through the fusion of multiple layers. Qingyun et al. (2021) introduced a cross-modality fusion transformer, that makes full use of the complementarity between different modalities to improve the performance of features. However, the main challenge of a simple feature fusion strategy is how to fuse high-level semantic information and low-level detailed features. To address these issues, we propose an aggregation structure based on hierarchical transformers, which can fully mine the coherence among multi-level features at different scales, and achieve discriminative feature representation ability.

Method

Overview

In this section, we describe the proposed SiamHFFT model. As can be seen in Figure 1, our model follows a Siamese tracking framework. There are four key components in our model, namely the feature extraction module, reshape module, feature fusion module, and prediction head. During tracking, the feature extraction module extracts feature from the template and search region. The features of the two branches from the last three layers of the backbone are correlated separately, and the outputs are denoted as M_2, M_3, and M_4 in order. We then feed the correlated features into the reshaping module, which can transform the channel dimensions of the backbone features and flatten features in the spatial dimension. The feature fusion module is implemented by fusing features using our hierarchical feature fusion transformer (HFFT) and a self-attention module. Finally, we used the prediction head module to perform bounding box regression and binary classification on the enhanced features to generate tracking results.

Feature extraction and reshaping

Similar to most Siamese tracking networks, the proposed method uses template frame patch ($Z \in \mathbb{R}^{3\times 80\times 80}$) and search frame patch ($X \in \mathbb{R}^{3\times 320\times 320}$) as inputs. For the backbone, our method can use an arbitrary deep CNN such as ResNet, MobileNet (Sandler et al., 2018), AlexNet, or ShuffleNet V2 (Ma et al., 2018). In this study, because a deeper network is unsuitable for deployment with limited computing resources, we adopted ShuffleNetV2 as a backbone network. This network is utilized for both template and search branch feature extraction.

To obtain robust and discriminative feature representations, we incorporate detailed structural information into our visual representations by extracting hierarchical features with different scales and semantic information in stage two, three and four of feature extraction. We denote feature tokens from the template branch as $F_i(Z)$ and those from the search branch as $F_i(X)$, where i represents the stage number of feature extraction and $i \in \{2, 3, 4\}$.

Next, a convolution operation is performed on the feature maps from the multi stages correlation, which is defined as:

$$M_i = F_i(Z) * F_i(X), i = 2, 3, 4, \tag{1}$$

where $M_i \in \mathbb{R}^{C_i\times H_i\times W_i}$, and C, H, and W denote the channel, width, and height of the feature map respectively. Additionally, $C_i \in \{116, 232, 464\}$ and $*$ denotes the cross-correlation operator. Next, we use the reshaping module which consists of 1×1 convolutional kernels, to change the channel dimensions of the features from Equation (1). We then flatten the features in the spatial dimension because a unified channel can not only effectively reduce computing resource requirements, but is also an essential component for improving the performance of feature fusion. After these operations, we can obtain a reshaped feature map $M_i{}' \in \mathbb{R}^{W_iH_i\times C}$, where $C = 192$.

Feature fusion and prediction head

As illustrated in Figure 1, following the convolution and flattening operations in the reshaping module, the correlation features from different stages are unified in the channel dimension. To explore the interdependencies among multi-level features fully, we designed the HFFT, which is detailed in this section.

Multi-Head Attention (Vaswani et al., 2017): Generally, transformers have been successfully applied to enhance feature

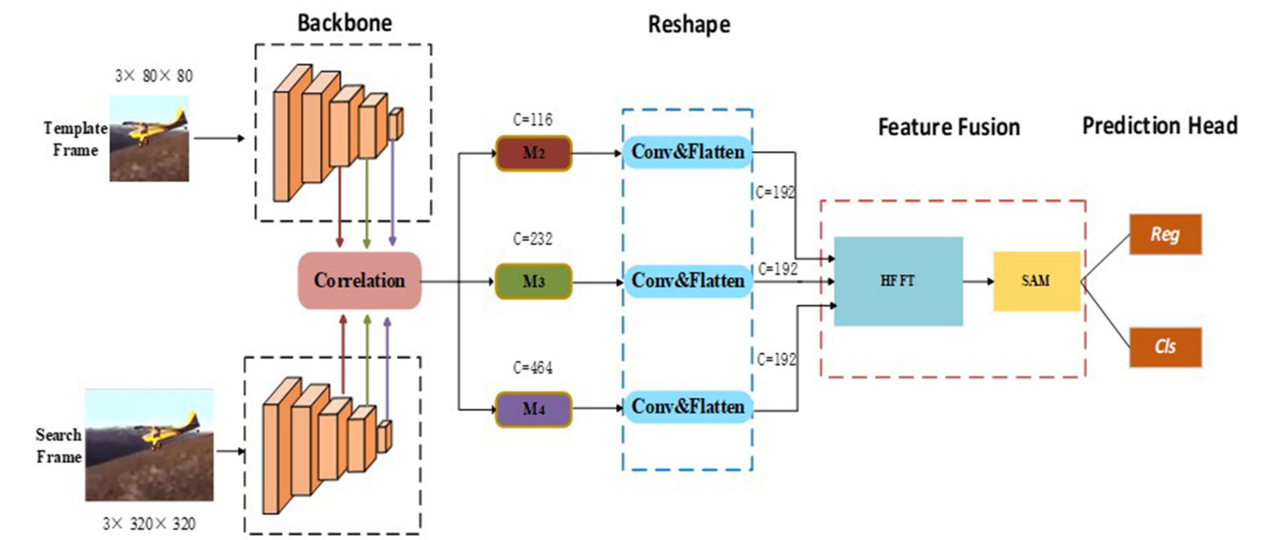


FIGURE 1
Architecture of the proposed SiamHFFT tracking framework. This framework contains four fundamental components: a feature extraction network, reshaping module, feature fusion module, and prediction head. The backbone network is used to extract hierarchical features. The reshaping module is designed to perform convolution operations and flatten features. The feature fusion transformer consists of the proposed HFFT module and a self-attention module (SAM). Finally, bounding boxes are estimated based on the regression and classification results.

representations in various bi-modal vision tasks. In the proposed feature fusion module, the attention mechanism is also a fundamental component. It is implemented using an attention function and operated on queries Q, keys K and values V using the scale dot-production method, which is defined as:

$$Attention(Q,K,V) = softmax(\frac{QK^{\top}}{\sqrt{C}})V \tag{2}$$

where C is the key dimensionality for normalizing the attention, and$\sqrt{C}$ is a scaling factor to avoid gradient vanishing in the loss function. Specifically, $Q = [q_1, \dots, q_N]^T \in \mathbb{R}^{N\times C}$ is the q input in Figure 2B, which denotes a collection of N features; similarly, K and V are the k and v inputs, respectively, which represent a collection of Mfeatures (i.e., $K, V \in \mathbb{R}^{M\times C}$). Notably, Q, K, V represent the mathematical implementation of the attention function and do not have practical meaning.

According to Vaswani et al. (2017), extending the attention function in Equation (2) to multiple heads is beneficial for enabling the mechanism to learn various attention distributions and enhancing its feature representation ability. This extension can be formulated as follows:

$$MultiHead(Q,K,V) = Concat(head_1, \dots head_h)W^o \tag{3}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V), i = 1, \dots h \tag{4}$$

where W_i^Q, W_i^K and $W_i^V \in \mathbb{R}^{C\times d_h}$, and $W^o \in \mathbb{R}^{C\times C}$. Here, h is the number of attention heads, which is defined as $d_h = \frac{C}{h}$. In this study, we adopted and $h = 6$ as default values.

Application to Dual-Input Tasks: The structure of a dual-input task is presented in Figure 2A, where Q, K, and V for normal NLP/vision tasks (Nguyen et al., 2020) share the same modality. In recent years, this mechanism has been extended to dual-inputs and applied to vision tasks (Chen X. et al., 2021; Chen et al., 2022a,b). However, the original attention mechanism cannot distinguish between the position information of different input feature sequences. The original mechanism only considers the absolute position and adds absolute positional encodings to inputs. It considers the attention from a source feature ϕ to a target feature θ as:

$$A_{\phi}(\theta) = MultiHead(\theta + P_{\theta}, \phi + P_{\phi}, \phi) \tag{5}$$

where P_{θ} and P_{ϕ} are the spatial positional encodings of features θ and ϕ, respectively. Spatial positional encoding is generated using a sine function. Equation (5) can be used not only as a single-direction attention enhancement, but also as a co-attention mechanism in which both directions are considered. Furthermore, self-attention from a feature to itself is also defined as a special case:

$$A_{\theta}(\theta) = MultiHead(\theta + P_{\theta}, \theta + P_{\theta}, \theta) \tag{6}$$

As shown in Figure 2A, following Equations (5) and (6), the designed transformer blocks are processed independently.

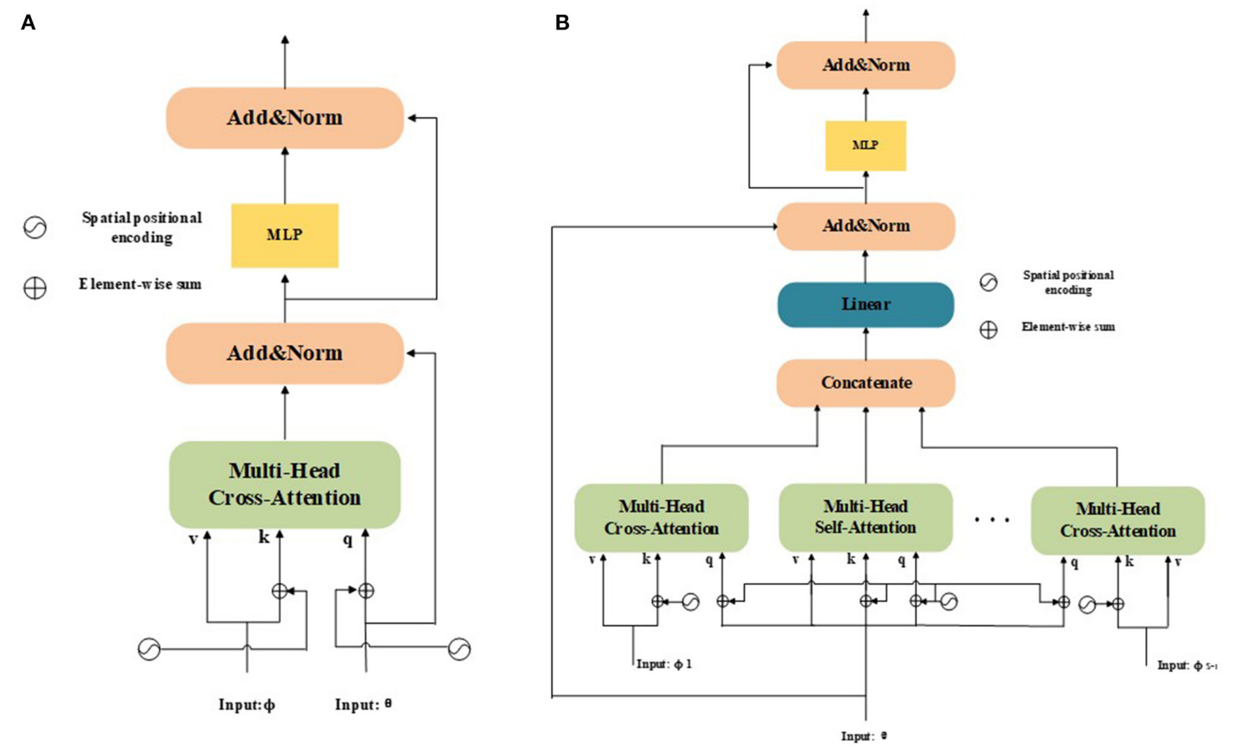


FIGURE 2
(A) Structure of a dual-input tasks; **(B)** Structure of a multi-input tasks. Unlike the original dual-input tasks, multi-input tasks can be used to learn the interdependencies of multi-level features and enhance the feature representation of the model in an end-to-end manner.

Therefore, the two modules can be used sequentially or in parallel. Additionally, a multilayer perceptron (MLP) module is used to enhance the fitting ability of the model. The MLP module is a fully connected network consisting of two linear projections with a Gaussian error linear unit (GELU) activation function between them, which can be denoted as:

$$MLP(\theta') = FC_2(GELU(FC_1(\theta'))) \tag{7}$$

Application to Multi-Input Tasks: To extend the attention mechanism to multiple inputs that are capable of handling multimodal vision tasks, pyramid structures, etc., we denote the total input number as S. The structure of a multi-input task is presented in Figure 2B. If we consider each possibility, there are a total of $S(S-1)$ source-target cases and S self-attention cases. Now, we denote the multiple inputs as $\{\theta, \phi_1, \ldots, \phi_{S-1}\}$, where the target $\theta \in \mathbb{R}^{N\times C}$ and source $\phi_i \in \mathbb{R}^{M\times C}$. Notably, θ and ϕ_i must have the same size as C. We then compute all the source-target cases as $\{A_{\phi_1}(\theta), \ldots, A_{\phi_{S-1}}(\theta)\}$. Next, we concatenate all source-to-target attention cases with self-attention $A_\theta(\theta)$, which can be formulated as:

$$\theta_{concat} = [A_\theta(\theta), A_{\phi_1}(\theta), \ldots, A_{\phi_{S-1}}(\theta)] \tag{8}$$

where $\theta_{concat} \in \mathbb{R}^{N\times SC}$. After concatenation, the dimensions of the enhanced features in the channel change to match the size SC of the original feature. To accelerate these calculations further, we apply a fully connected layer to reduce the channel dimensions to:

$$\theta_{concat}{}' = Linear[\theta_{concat}] \tag{9}$$

where $\theta_{concat}{}' \in \mathbb{R}^{N\times C}$. Through this process, we can obtain more discriminative features efficiently by aggregating features from different attention mechanisms.

HFFT: As is shown in Figure 2B, in our model, we make full use of the hierarchical features $M_i{}' \in \mathbb{R}^{W_iH_i\times C}$ $(i \in \{2, 3, 4\})$ and generate tracking-tailored features. To integrate low-level spatial information with high-level semantic information, we feed the reshaped features from the output of Equation (1), namely $M_2{}'$, $M_3{}'$, and $M_4{}'$, into the HFFT module, where $M_3{}'$ is used for target feature, $M_2{}'$ and $M_4{}'$ represent source features. The importance of different aspects feature information is assigned by applying the cross-attention operator to $M_2{}'$ and $M_4{}'$, which is beneficial for obtaining more discriminative features. We apply self-attention to $M_3{}'$, which can preserve the details of target information during tracking. Furthermore, positional information is encoded during the calculation process to enhance spatial information during the tracking process. The attention mechanisms are implemented using the operation of K, Q, V. Then, comprehensive features can be obtained by concatenating the outputs. Due to the complexity of a model increases with its input size, a fully connected layer is utilized to resize outputs. We also adopt residual connections around each sub-layer. Additionally, we use an MLP module to enhance the fitting ability of the model, and layer normalization (LN) is performed before the MLP and final output steps. The entire process of the HFFT can be expressed as:

$$M_{concat} = [A_{M_3{'}}(M_3{'}), A_{M_2{'}}(M_3{'}), A_{M_4{'}}(M_3{'})],$$

$$M_{concat}{'} = Linear[M_{concat}],$$

$$M_{out} = LN(M_{concat}{'} + M_3{'}),$$

$$X_{out} = LN(M_{out} + MLP(M_{out})) \tag{10}$$

SAM: The SAM is a feature enhancement module. The structure of the SAM is presented in Figure 3. The SAM adaptively integrates information from different feature maps using multi-head self-attention in the residual form. In the proposed model, the SAM take the out of Equation (10) X_{out} as input. The mathematical process of the SAM can be summarized as:

$$X_{out2} = LN(MultiHead(X_{out} + P_X, X_{out} + P_X, X_{out}) + X_{out}),$$

$$X_{SAM} = LN(MLP(X_{out2}) + X_{out2}) \tag{11}$$

Prediction Head: The enhanced features are reshaped back to the original feature size before being fed into the prediction head. The head network consists of two branches: a classification branch and bounding box regression branch. Each branch consists of a three-layer perceptron. The former is utilized to distinguish the target from the background, and the latter is used for estimating the location of the target by using a bounding box. Overall, the model is trained using a combination loss function formulated as:

$$L = \lambda_{cls}L_{cls} + \lambda_{giou}L_{giou} + \lambda_{loc}L_{loc} \tag{12}$$

where L_{cls}, L_{giou}, and L_{loc} represent the binary cross-entropy, GIOU loss, and L1-norm loss, respectively. λ_{cls}, λ_{giou}, and λ_{loc} are coefficients that balance the contributions of each type of losses.

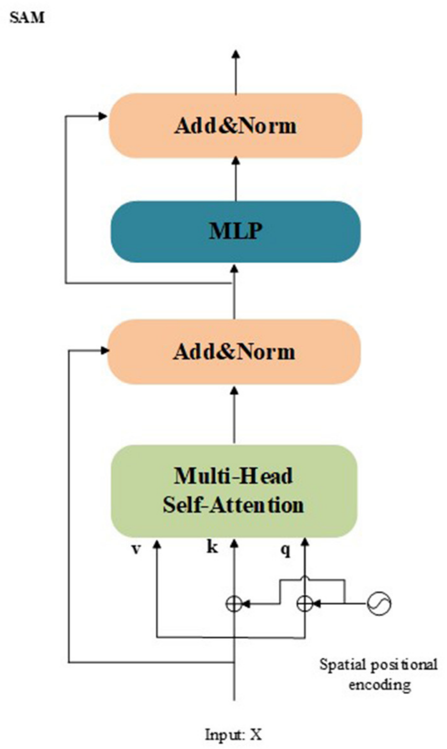


FIGURE 3
Architecture of the proposed SAM.

Experiments

This section presents the details of the experimental implementation of the proposed model. To validate the performance of the proposed tracker, we compared our method to state-of-the-art methods on four popular benchmarks. Additionally, ablation studies were conducted to analyse the effectiveness of key modules.

Implementation details

The tracking algorithm was implemented in Python based on PyTorch. The proposed model was trained on a PC with an Intel i7-11700k, 3.6 GHz CPU, 64 GB of RAM, and an NVIDIA 3080Ti RTX GPUs. The training splits of LaSOT (Fan et al., 2019), GOT-10k (Huang et al., 2019), COCO (Lin et al., 2014), and TrackingNet (Muller et al., 2018) were used to train the model. We randomly selected two image pairs from the same video sequences with a maximum gap of 100 frames to generate the search patches and template patches. The sizes of search patches were set to 320 × 320 × 3 and template patches were resized to sizes of 80 × 80 × 3. The parameters for the backbone network were initialized using ShuffleNetV2, which was pretrained on ImageNet. All models were trained for 150 epochs with a batch size of 32. Each epoch contained 60,000 sampling pairs. The coefficient parameters in Equation (12) were set to $\lambda_{cls} = 2$, $\lambda_{giou} = 2$, and $\lambda_{loc} = 5$. In the offline training phrase, the parameters of the model are optimized by ADAMW optimizer. The learning rates of the backbone network were set to le-5, and le-4 for the remaining parts.

Comparisions to state-of-the-art methods

We compared SiamHFFT to state-of-the-art trackers on four benchmarks: LaSOT, UAV123 (Mueller et al., 2016), UAV123@10fps, and VOT2020 (Kristan et al., 2020). The evaluation results are presented in the following paragraphs. It is worthy note that the performance (accuracy and success scores) of the comparision methods on these compared benchmarks are obtained by the public tracking results files, which are released by their authors.

Evaluation on LaSOT: LaSOT is a large-scale long-term tracking benchmark consisting of 1,400 sequences. We used test splits and the one pass evaluation (OPE) to evaluate the performances of the compared trackers. That is, initialize the tracking algorithm according to the target position given in the first frame of the video sequence, and then run the prediction of the target position and size in the whole video to obtain the tracking accuracy or success rate.

Figures 4, 5 report the plots of the precision and success scores of the comparision trackers, respectively. The precision score measures the center location error (CLE), which calculates the average Euclidean distance between the estimated bounding box and the ground truth. The CLE is calculated as follows:

$$CLE = \sqrt{(x_a - x_b)^2 + (x_a - x_b)^2} \tag{13}$$

As the CLEs of frame are obtained, the precision plots (Figure 4) show the percentage of frames in which the estimated CLE is lower than a certain threshold (usually set to 20 pixels) in the total frames of the video sequence.

The Success curve (Figure 5) refers to the percentage of the number of frames whose predicted overlap rate between the estimated bounding box and the ground truth is higher than the given threshold (usually set to 0.5) to the total number of frames in the video sequence. The overlap rate is calculated as follows:

$$S = \frac{|b_t \cap b_g|}{|b_t \cup b_t|} \tag{14}$$

where b_t denotes the estimated bounding box, b_g represents the ground truth bounding box, ∩ refers to intersection operator, ∪ stands for union operator, and || denotes the number of pixels in the resulted region.

The curves of the proposed SiamHFFT are depicted in green. Overall, our tracker ranks the third in precision, and achieves the second-best score in success, with 61% at the precision score and 62% success score. Compared with the trackers with deeper backbones, such as SiamCAR, SiamBAN, and SiamRPN++ (Li B. et al., 2019), our tracker exhibits competitive performance with a lighter structure. The DiMP achieves the best performance both in precision and success. Our SiamHFFT tracker outperforms other Siamese-based trackers, even with deeper backbones and dedicated-designed structures.

Evaluation on UAV123: UAV123 is an aerial tracking benchmark consisting of 123 videos containing small objects, target occlusions, out of view, and distractors. To validate the performance of our tracker, we evaluated the performances of our trackers and other state-of-the-art trackers, including SiamFC, ECO (Danelljan et al., 2017), ATOM (Danelljan et al., 2019), SiamAttn (Yu et al., 2020), SiamRPN++, SiamCAR, DiMP (Bhat et al., 2019), SiamBAN, and HiFT. Table 1 lists the results in terms of success, precision, and speed on GPU. Additionally, the backbones of the trackers are also reported for an intuitive comparision. The best performance for each criterion is indicated in red.

Among the trackers, those with deeper backbones, such as DiMP, ATOM, and SiamBAN, achieve better performance in term of both precision and success rate. SiamFC, HiFT, and the proposed SiamHFFT utilize lightweight backbone. SiamFC achieves the best performance in speed, but this naive network structure does not achieve satisfactory results in terms of precision and success rate. HiFT adopts a feature transformer to enhance feature representations. Compared to HiFT, our tracker exhibits a clear advantage in term of precision (82.8 vs. 78.7%) and success rate (62.5 vs. 58.9%), which validates the effectiveness of the proposed tracker. According to the last row in Table 1, all compared trackers can run in real-time on a GPU at an average speed of 68 FPS, proving that SiamHFFT maintains a suitable balance between performance and efficiency.

Figure 6 depicts the qualitative results by multiple algorithms on a subset of sequences in UAV123 benchmarks. We choose three sets of the challenging video sequences: Car18_1, Person21_1, and Group3_4_1. All of the three video sequences are shot by the camera of the UAV, the video frames undergo multiple challenges, for example scale variation, changes of different viewpoint, and so on. Generally, the given target appears in small size during the tracking process. The bounding boxes estimated by the trackers are marked in different colors to give an intuitive contrast. The bounding box of our SiamHFFT is shown in red, and it is obvious that our tracker can handle these complex scenarios well, especially for the small object tracking task.

UAV123@10fps: UAV123@10fps is a subset of UAV123 obtained by down-sampling the original videos with an image

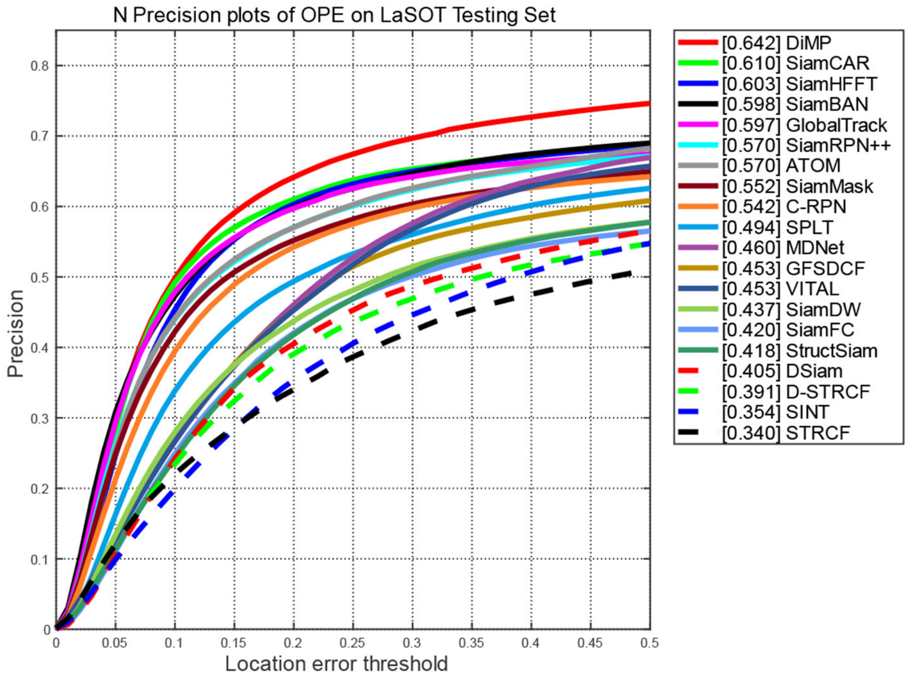


FIGURE 4
Precision scores of compared trackers on LaSOT.

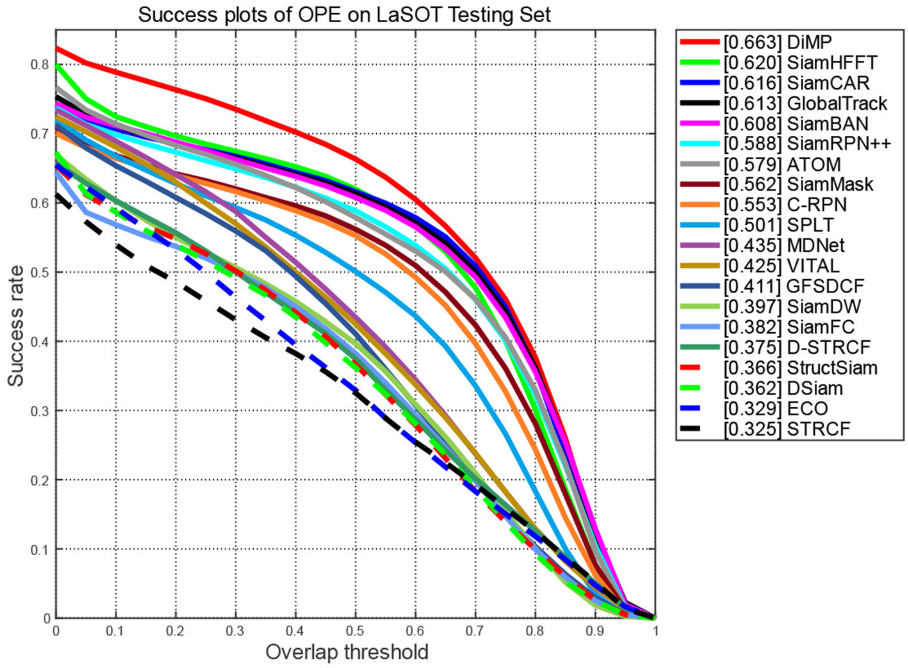


FIGURE 5
Success scores of compared trackers on LaSOT.

TABLE 1 Quantitative evaluation on UAV123 in term of precision (Prec.), success (Succ.) and GPU speed (FPS).

	SiamFC	ECO	ATOM	SiamAttn	SiamRPN++	SiamCAR	DiMP	SiamBAN	HiFT	SiamHFFT
Feat.	Alex	VGG	R18	R50	R50	R50	R50	R50	Alex	ShuffleNet
Prec.	72.5	75.2	83.7	84.5	76.9	76	84.9	83.3	78.7	82.9
Succ.	49.4	52.8	64.2	65	57.9	61.4	65.4	63.1	58.9	62.6
FPS	130	45	46	45	35	52	45	40	/	68

The best performance are shown in red.

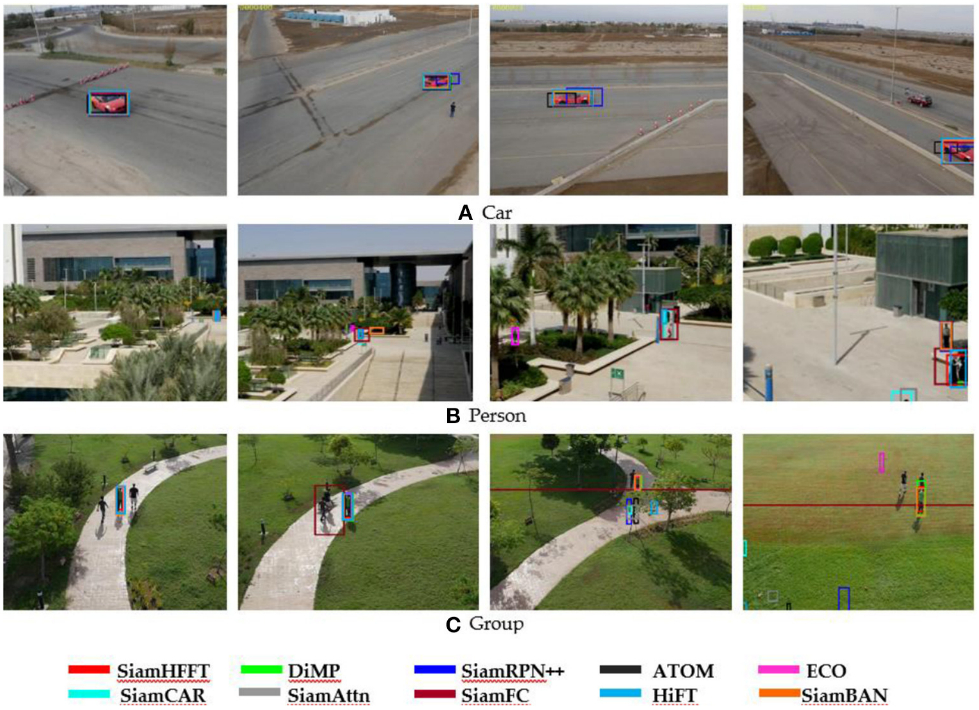


FIGURE 6
Qualitative experimental results in several challenging sequences on UAV123 dataset. **(A)** Video sequences of the Car, **(B)** video sequences of the Person, and **(C)** video sequences of the Group.

rate of 10 FPS. We use SiamFC, AutoTrack (Li et al., 2020), TADT (Li X. et al., 2019), MCCT (Wang et al., 2018), SiamRPN++, DeepSTRCF (Li F. et al., 2018), CCOT (Danelljan et al., 2016), ECO, and HIFT as comparisions. Among these trackers, AutoTrack, TADT, MCCT, CCOT, ECO and DeepSTRCF are correlation filter based trackers, which has a lightweight structure and less parameters than deep learning based trackers, and the model can be deployed on limited source device. Compared with UAV 123 benchmark, challenge in UAV123@10fps dataset are more abrupt and severe. The experimental results are listed in Table 2. Compared with the correlation filter based trackers, the deep trackers, HiFT and SiamRPN++ achieve higher precision and success scores, the performance of SiamFC is closer to these correlation based trackers, SiamFC utilize the AlexNet as the backbone, but the model does not further enhance the feature representation. Our SiamHFFT model yields the best precision (76.5%) and success rate (59.5%), which has an advantage over HiFT by 1.1, 2.1%, demonstrating the effectiveness of the HFFT module, and superior robustness capacity compared to other prevalent trackers.

Evaluation on VOT2020: We also test SiamHFFT on the VOT2020 benchmark against HCAT, LightTrack (Yan et al., 2021b), ATOM and DiMP. VOT2020 consists of 60 videos with mask annotations and adopts the expected average overlap (EAO) as the metric for evaluating the performance of the trackers, which is calculated by:

$$\bar{\phi}N_S = \frac{1}{N_S}\sum_{i=1}^{N_S}\phi N_S \tag{15}$$

TABLE 2 Overall evaluation on UAV123@10fps.

	SiamFC	AutoTrack	TADT	MCCT	SiamRPN++	DeepSTRCF	CCOT	ECO	HiFT	SiamHFFT
Prec.	67.8	67.6	68.4	68.1	74.0	68.0	70.4	70.9	75.4	76.5
Succ.	47.2	48.1	50.7	49.2	55.5	49.9	50.2	51.9	57.4	59.6

The best performance are shown in red.

TABLE 3 Evaluation on VOT2020.

	HCAT	LightTrack	ATOM	DiMP	SiamHFFT
EAO	0.276	0.242	0.271	0.274	0.231
Accuracy	0.455	0.422	0.462	0.457	0.459
Robustness	0.747	0.689	0.734	0.740	0.646

The best performance are shown in red.

where N_S denotes the length of the video sequences, ϕN_S denotes the average accuracy of a video sequence whose length is N_S. Finally, the EAO value can be obtained by calculating the average value of the video sequences of N_S length.

The experimental results are presented in Table 3. Our tracker achieves an EAO value of 0.231, robustness of 0.646, and accuracy of 0.459. The performance of SiamHFFT is comparable to that of the state-of-the-art models for each criterion.

Speed, FLOPs and params

To verify the efficiency of our tracker, we conducted a set of experiments on the GOT-10k benchmark, which is a large-scale tracking dataset consisting of more than 10,000 videos, covering a wide range of 560 types of common moving objects. Following the test protocols of GOT-10k, all of the evaluated trackers are trained with the same training data, and are tested with the same test data. We evaluated the performance of SiamHFFT against TransT, STARK, DiMP, SiamRPN++, ECO, ATOM, and LightTrack. Our SiamHFFT is conducted on PC while the data of other trackers on GOT-10k is obtained from Chen et al. (2022b). Both average overlap (AO) and speed were considered to evaluate the performance of the trackers. We visualize the AO performance with respect to the frames-per-seconds (FPS) tracking speed. The comparision results are presented in Figure 7. Each tracker is represented by a circle, and the radius of the circle r is calculated as follows:

$$r = k\frac{speed/Average(speed)}{AO} \tag{16}$$

where k denotes a scale factor, we set k=10. The higher value of r indicates the better performance. All trackers were tested on CPU platform, and real-time line (26 fps) performance is represented by a dotted line to measure the real-time capacity of the trackers, trackers locate on the right side of the line are considered to achieve the real-time performance. According to Figure 7, only SiamHFFT and LightTrack can meet the real-time requirement on the CPU. Among these comparision trackers, TransT utilized a modified ResNet50 as backbone and a transformer-based network to obtain discriminative features, and achieve the highest AO score, but it sacrifices the speed which runs a low speed on CPU. Similarly, STARK, DiMP, prDiMP, SiamRPN++ can only obtain satisfactory AO scores at the expense of speed. The correlation filter-based tracker, ECO, also adopts the deep features which does not achieve a satisfactory speed on CPU. Our tracker exhibits an average speed of 28 FPS on the CPU, not only reach the real-time requirement, but the area of the circle representing our method is the second large of all the trackers.

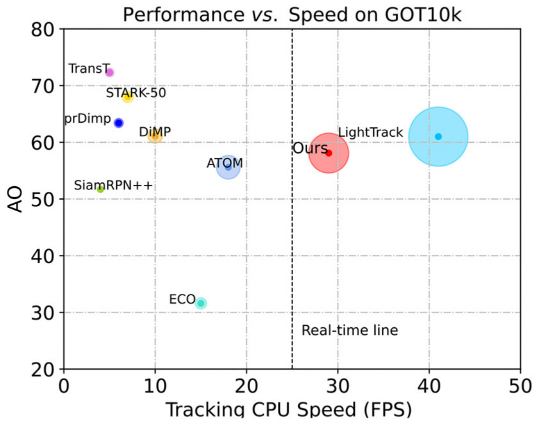


FIGURE 7
Speed and performance comparisions on GOT-10k. The horizontal axis represents model speed on a CPU and the vertical axis represents the AO score.

To validate the lightness of our model, we compared the floating-point operations (FLOPs) and Params of the model with STARK-S50 and SiamRPN++. FLOPs represent the theoretical

TABLE 4 Comparision about the FLOPs and params.

Trackers	FLOPs (G)	Params (M)
STARK-S50	10.5	23.3
SiamRPN++	48.9	54
SiamHFFT	0.6	4.4

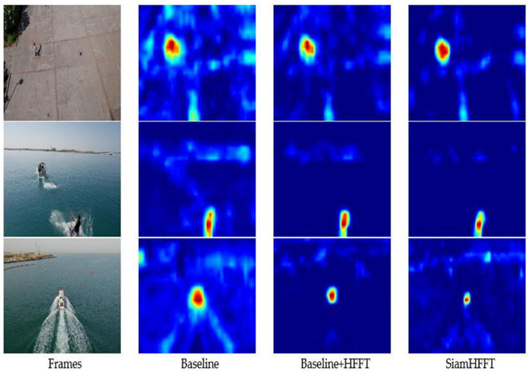


FIGURE 8
Visualization of the confidence maps of three trackers on several sequences from the UAV123 dataset. The response visualization results are an intuitive reflection of tracker performance.

calculation volume of the model, which means the number of calculations required for the inference. Params refer to the amount of the parameters in the model, which directly determines the size of the model and also directly affects the memory consumption when a model making inferences. The comparison results are illustrated in Table 4. It is worth note that our SiamHFFT tracker achieve a promising result over other trackers. The FLOPs and Parameters are 16× and 5× less than those of STARK-S50. This shows that our method can use fewer parameters and lower memory consumption, making it possible for deployments in the edge hardware environments.

Ablation studies

This section presents ablation studies conducted to verify the effectiveness of our framework. We selected several challenging frames from the UAV123 dataset and visualized the tracking results using heatmaps, as shown in Figure 8. The first column presents the given target which is highlighted with a red box, and the remaining columns present the visualized results of the predicted target prior to the current frame.

The second column presents the visualization results of the baseline, which only adopts ShuffleNetV2 as backbone with the reshaping module and the prediction head. The response area of the baseline is much larger than the original target size and has obscure edges affected by distractors in the frames.

TABLE 5 Experimental results on UAV 123 benchmark with different backbones.

	Baseline	Baseline+HFFT	SiamHFFT
AlexNet	73.6	77.2	78.9
ShuffleNetV2	74.1	81.6	82.8

The third column presents the visualization results of the baseline with the HFFT module. Compared with the baseline alone, the response area is smaller and clearer because the HFFT module enhances the critical semantic and spatial features of the target, enabling the model to generate more discriminative response maps. With the HFFT module, our tracker achieves significant improvement in tracking accuracy, which validates the effectiveness of the HFFT module for handling small objects.

The last column presents the response map generated by the proposed SiamHFFT, which adopts the entire operation module, backbone, reshaping module, HFFT module and the SAM, where the classification and regression head are utilized to estimate the location of a target. According to the visualization results of the response maps, our SiamHFFT model has clear advantages over other modified versions. The response areas are more precise and discriminative relative to the distractors.

We also test the performance on UAV123 benchmark with different backbones, we use the accuracy score to measure the performance variation. Experimental result is shown in Table 5, we choose two lightweight networks, AlexNet and ShuffleNetV2, to make a comparision. Similar to Figure 8, the effectiveness of the HFFT module is measured in a quantitative manner. The model adopts ShuffleNetV2 as backbone has better performance on all of the three criteria. The experiment results of Table 4 also demonstrate the effectiveness of the HFFT module.

Conclusion

In this paper, an HFFT tracking method based on a Siamese network was proposed. To integrate and optimize multi-level features, we designed a novel feature fusion transformer that can reinforce semantic information and spatial details during the tracking process. Additionally, based on our lightweight backbone, excessive computation for feature extraction is avoided, which accelerates object tracking speed. To validate the effectiveness of our trackers, extensive experiments were conducted on five benchmarks. Our method achieves excellent results on small target datasets such as UVA123 and UAV123@10fps, and also shows good performance on genetic public visual tracking datasets, such as LaSOT, VOT2020, and GOT-10k. Our method can potentially inspire further research on small object tracking, particularly for UAV tracking.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author/s.

Author contributions

Conceptualization, methodology, software, validation, formal analysis, data curation, and writing—original draft preparation: JD. Investigation: SW. Resources, writing—review and editing, supervision, and funding acquisition: YC. Visualization: YF. All authors have read and agreed to the published version of the manuscript.

Funding

YC is the leader of the funding for the research of National Science Foundation of China (No. 11975066), he received his Ph.D. degree from Jilin University, in 2002. He is a Professor with Jilin University, and also with Dalian University of Technology. His research interests include CMOS image sensor and digital signal processing of images.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2022.1082346/full#supplementary-material

References

Beal, J., Kim, E., Tzeng, E., Park, D. H., Zhai, A., and Kislyuk, D. J. (2020). Toward transformer-based object detection. *arXiv [Preprint]*. doi: 10.48550/arXiv.2012.09958

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. (2016). "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision* (New York, NY: Springer), 850–865. doi: 10.1007/978-3-319-48881-3_56

Bhat, G., Danelljan, M., Gool, L. V., and Timofte, R. (2019). "Learning discriminative model prediction for tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 6182–6191. doi: 10.1109/ICCV.2019.00628

Cao, Z., Fu, C., Ye, J., Li, B., and Li, Y. (2021). "HiFT: hierarchical feature transformer for aerial tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 15457–15466. doi: 10.1109/ICCV48922.2021.01517

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). "End-to-end object detection with transformers," in *European Conference on Computer Vision* (New York, NY: Springer), 213–229. doi: 10.1007/978-3-030-58452-8_13

Chen, B., Li, P., Bai, L., Qiao, L., Shen, Q., Li, B., et al. (2022). Backbone is all your need: a simplified architecture for visual object tracking. *arXiv preprint arXiv:2203.05328*. doi: 10.1007/978-3-031-20047-2_22

Chen, C. -F. R., Fan, Q., and Panda, R. (2021). "Crossvit: cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 357–366. doi: 10.1109/ICCV48922.2021.00041

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., et al. (2021). Decision transformer: reinforcement learning *via* sequence modeling. *Adv. Neural Inform. Process. Syst.* 34, 15084–15097.

Chen, X., Wang, D., Li, D., and Lu, H. J. (2022b). Efficient visual tracking via hierarchical cross-attention transformer. *arXiv [Preprint]*. doi: 10.48550/arXiv.2203.13537

Chen, X., Yan, B., Zhu, J., Wang, D., and Lu, H. J. (2022a). High-performance transformer tracking. *arXiv preprint arXiv:2203.13533* (New Orleans, LA: IEEE). doi: 10.1109/CVPR46437.2021.0080

Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., and Lu, H. (2021). "Transformer tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE), 8126–8135. doi: 10.1109/CVPR46437.2021.00803

Chen, Z., Zhong, B., Li, G., Zhang, S., and Ji, R. (2020). "Siamese box adaptive network for visual tracking", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6668–6677. doi: 10.1109/CVPR42600.2020.00670

Danelljan, M., Bhat, G., Khan, F. S., and Felsberg, M. (2019). "Atom: accurate tracking by overlap maximization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 4660–4669. doi: 10.1109/CVPR.2019.00479

Danelljan, M., Bhat, G., Shahbaz Khan, F., and Felsberg, M. (2017). "Eco: efficient convolution operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 6638–6646. doi: 10.1109/CVPR.2017.733

Danelljan, M., Robinson, A., Shahbaz Khan, F., and Felsberg, M. (2016). "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision* (New York, NY: Springer), 472–488. doi: 10.1007/978-3-319-46454-1_29

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. J. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv [Preprint]*. doi: 10.48550/arXiv.1810.04805

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16 × 16 words: transformers for image recognition at scale. *arXiv [Preprint]*. doi: 10.48550/arXiv.2010.11929

Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., et al. (2019). "Lasot: A high-quality benchmark for large-scale single object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 5374–5383. doi: 10.1109/CVPR.2019.00552

Fan, H., and Ling, H. (2019). "Siamese cascaded region proposal networks for real-time visual tracking," in *Proceedings of the IEEE/CVF Conference On Computer Vision and Pattern Recognition*, 7952–7961. doi: 10.1109/CVPR.2019.00814

Fan, H., and Ling, H. J. (2020). Cract: cascaded regression-align-classification for robust visual tracking. *arXiv preprint arXiv:2011.12483*. doi: 10.1109/IROS51168.2021.9636803

Guo, D., Wang, J., Cui, Y., Wang, Z., and Chen, S. (2020). "SiamCAR: siamese fully convolutional classification and regression for visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 6269–6277. doi: 10.1109/CVPR42600.2020.00630

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., et al. (2022). A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* 1. doi: 10.1109/TPAMI.2022.3152247

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 770–778. doi: 10.1109/CVPR.2016.90

He, X., Chen, Y., and Lin, Z. J. R. S. (2021). Spatial-spectral transformer for hyperspectral image classification. *Remote Sens*. 13, 498. doi: 10.3390/rs13030498

Hou, R., Ma, B., Chang, H., Gu, X., Shan, S., Chen, X. J. I. T., et al. (2020). IAUnet: global context-aware feature learning for person reidentification. *IEEE Trans Neural Netw Learn Syst*. 32, 4460–4474. doi: 10.1109/TNNLS.2020.3017939

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv [Preprint]*. doi: 10.48550/arXiv.1704.04861

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 7132–7141. doi: 10.1109/CVPR.2018.00745

Huang, L., Zhao, X., Huang, K. J., and Intelligence, M. (2019). Got-10k: a large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 1562–1577. doi: 10.1109/TPAMI.2019.2957464

Javed, S., Danelljan, M., Khan, F. S., Khan, M. H., Felsberg, M., and Matas, J. (2021). Visual object tracking with discriminative filters and siamese networks: a survey and outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* 1–20. doi: 10.1109/TPAMI.2022.3212594

Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.-K., et al. (2020). "The eighth visual object tracking VOT2020 challenge results," in *European Conference on Computer Vision* (New York, NY: Springer), 547–601.

Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., and Yan, J. (2019). "Siamrpn++: evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 4282–4291. doi: 10.1109/CVPR.2019.00441

Li, B., Yan, J., Wu, W., Zhu, Z., and Hu, X. (2018). "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 8971–8980. doi: 10.1109/CVPR.2018.00935

Li, F., Tian, C., Zuo, W., Zhang, L., and Yang, M.-H. (2018). "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 4904–4913. doi: 10.1109/CVPR.2018.00515

Li, X., Ma, C., Wu, B., He, Z., and Yang, M.-H. (2019). "Target-aware deep tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 1369–1378. doi: 10.1109/CVPR.2019.00146

Li, Y., Fu, C., Ding, F., Huang, Z., and Lu, G. (2020). "AutoTrack: towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 11923–11932. doi: 10.1109/CVPR42600.2020.01194

Lin, L., Fan, H., Xu, Y., and Ling, H. J. (2021). Swintrack: a simple and strong baseline for transformer tracking. *arXiv [Preprint]*. doi: 10.48550/arXiv.2112.00995

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 2117–2125. doi: 10.1109/CVPR.2017.106

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft coco: common objects in context," in *European Conference on Computer Vision* (New York, NY: Springer), 740–755. doi: 10.1007/978-3-319-10602-1_48

Lin, Z., Feng, M., Santos, C. N., Yu, M., Xiang, B., Zhou, B., et al. (2017). A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Liu, L., Hamilton, W., Long, G., Jiang, J., and Larochelle, H. J. (2020). A universal representation transformer layer for few-shot image classification. *arXiv preprint arXiv:2006.11702*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). "Swin transformer: hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 10012–10022. doi: 10.1109/ICCV48922.2021.00986

Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. (2018). "Shufflenet v2: practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich: Springer), 116–131. doi: 10.1007/978-3-030-01264-9_8

Marvasti-Zadeh, S. M., Cheng, L., Ghanei-Yakhdan, H., and Kasaei, S. (2021). Deep learning for visual tracking: a comprehensive survey. *IEEE Trans. Intell. Transp. Syst.* 23, 3943–3968. doi: 10.1109/TITS.2020.3046478

Mayer, C., Danelljan, M., Bhat, G., Paul, M., Paudel, D. P., Yu, F., et al. (2022). "Transforming model prediction for tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (New Orleans, LA: IEEE), 8731–8740. doi: 10.1109/CVPR52688.2022.00853

Mueller, M., Smith, N., and Ghanem, B. (2016). "A benchmark and simulator for uav tracking," in *European Conference on Computer Vision* New York, NY: Springer), 445–461. doi: 10.1007/978-3-319-46448-0_27

Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., and Ghanem, B. (2018). "Trackingnet: a large-scale dataset and benchmark for object tracking in the wild," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich: Springer), 300–317.

Nguyen, V.-Q., Suganuma, M., and Okatani, T. (2020). "Efficient attention mechanism for visual dialog that can handle all the interactions between multiple inputs," in *European Conference on Computer Vision* (New York, NY: Springer), 223–240. doi: 10.1007/978-3-030-58586-0_14

Ning, X., Duan, P., Li, W., and Zhang, S. J. I. S. P. L. (2020). Real-time 3D face alignment using an encoder-decoder network with an efficient deconvolution layer. *IEEE Signal Process. Lett.* 27, 1944–1948. doi: 10.1109/LSP.2020.3032277

Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., et al. (2020). "Stabilizing transformers for reinforcement learning," in *International Conference on Machine Learning*: *PMLR* (Vienna: ACM), 7487–7498.

Paulus, R., Xiong, C., and Socher, R. J. (2017). A deep reinforced model for abstractive summarization. *arXiv [Preprint]*. doi: 10.48550/arXiv.1705.04304

Qingyun, F., Dapeng, H., and Zhaokui, W. J. (2021). Cross-modality fusion transformer for multispectral object detection. *arXiv [Preprint]*. doi: 10.48550/arXiv.2111.00273

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 4510–4520. doi: 10.1109/CVPR.2018.00474

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Adv. Neural Inform. Process. Syst*. 30, 6000–6010.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., et al. (2017). "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 3156–3164. doi: 10.1109/CVPR.2017.683

Wang, N., Zhou, W., Tian, Q., Hong, R., Wang, M., and Li, H. (2018). "Multi-cue correlation filters for robust visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 4844–4853. doi: 10.1109/CVPR.2018.00509

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 7794–7803. doi: 10.1109/CVPR.2018.00813

Wolfe, J. M., and Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nat. Rev. Neurosci.* 5, 495–501. doi: 10.1038/nrn1411

Xu, Y., Wang, Z., Li, Z., Yuan, Y., and Yu, G. (2020). "Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proceedings of the AAAI Conference on Artificial Intelligence* (New York, NY: AAAI), 12549–12556. doi: 10.1609/aaai.v34i07.6944

Yan, B., Peng, H., Fu, J., Wang, D., and Lu, H. (2021a). "Learning spatio-temporal transformer for visual tracking," in *Proceedings of the IEEE/CVF*

International Conference on Computer Vision (Montreal, QC: IEEE), 10448–10457. doi: 10.1109/ICCV48922.2021.01028

Yan, B., Peng, H., Wu, K., Wang, D., Fu, J., and Lu, H. (2021b). "LightTrack: finding lightweight neural networks for object tracking *via* one-shot architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE), 15180–15189. doi: 10.1109/CVPR46437.2021.01493

Yan, B., Zhao, H., Wang, D., Lu, H., and Yang, X. (2019). "'Skimming-perusal'tracking: a framework for real-time and robust long-term tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 2385–2393. doi: 10.1109/ICCV.2019.00247

Yu, B., Tang, M., Zheng, L., Zhu, G., Wang, J., Feng, H., et al. (2021). "High-performance discriminative tracking with transformers," in: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 9856–9865. doi: 10.1109/ICCV48922.2021.00971

Yu, Y., Xiong, Y., Huang, W., and Scott, M. R. (2020). "Deformable siamese attention networks for visual object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 6728–6737. doi: 10.1109/CVPR42600.2020.00676

Zhang, G., Zhang, P., Qi, J., and Lu, H. (2021). "Hat: hierarchical aggregation transformers for person re-identification," in *Proceedings of the 29th ACM International Conference on Multimedia*, 516–525. doi: 10.1145/3474085.3475202

Zhang, J., Huang, B., Ye, Z., Kuang, L.-D., and Ning, X. J. S. R. (2021). Siamese anchor-free object tracking with multiscale spatial attentions. *Sci. Rep*. 11, 1–14. doi: 10.1038/s41598-021-02095-4

Zhang, W. J. M. S., and Processing, S. (2021). A robust lateral tracking control strategy for autonomous driving vehicles. *Mech. Syst. Signal Process*. 150, 107238. doi: 10.1016/j.ymssp.2020.107238

Zhang, Z., Lan, C., Zeng, W., Jin, X., and Chen, Z. (2020a). "Relation-aware global attention for person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 3186–3195. doi: 10.1109/CVPR42600.2020.00325

Zhang, Z., Peng, H., Fu, J., Li, B., and Hu, W. (2020b). "Ocean: object-aware anchor-free tracking," in *European Conference on Computer Vision* (New York, NY: Springer), 771–787. doi: 10.1007/978-3-030-58589-1_46

Zhao, M., Okada, K., and Inaba, M. J. (2021). Trtr: visual tracking with transformer. *arXiv [Preprint]*. doi: 10.48550/arXiv.2105.03817

Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., and Hu, W. (2018). "Distractor-aware siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich: Springer), 101–117. doi: 10.1007/978-3-030-01240-3_7

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 01 December 2022
DOI 10.3389/fnbot.2022.1036465

Physical-model guided self-distillation network for single image dehazing

Yunwei Lan, Zhigao Cui*, Yanzhao Su, Nian Wang, Aihua Li and Deshuai Han

Xi'an Research Institute of High Technology, Xi'an, China

Check for updates

OPEN ACCESS

EDITED BY
Xin Ning,
Institute of Semiconductors
(CAS), China

REVIEWED BY
Husnain Shahid,
Fudan University, China
Yuantao Chen,
Changsha University of Science and Technology, China

*CORRESPONDENCE
Zhigao Cui
cuizg10@126.com

RECEIVED 04 September 2022
ACCEPTED 07 October 2022
PUBLISHED 01 December 2022

CITATION
Lan Y, Cui Z, Su Y, Wang N, Li A and Han D (2022) Physical-model guided self-distillation network for single image dehazing.
Front. Neurorobot. 16:1036465.
doi: 10.3389/fnbot.2022.1036465

Motivation: Image dehazing, as a key prerequisite of high-level computer vision tasks, has gained extensive attention in recent years. Traditional model-based methods acquire dehazed images *via* the atmospheric scattering model, which dehazed favorably but often causes artifacts due to the error of parameter estimation. By contrast, recent model-free methods directly restore dehazed images by building an end-to-end network, which achieves better color fidelity. To improve the dehazing effect, we combine the complementary merits of these two categories and propose a physical-model guided self-distillation network for single image dehazing named PMGSDN.

Proposed method: First, we propose a novel attention guided feature extraction block (AGFEB) and build a deep feature extraction network by it. Second, we propose three early-exit branches and embed the dark channel prior information to the network to merge the merits of model-based methods and model-free methods, and then we adopt self-distillation to transfer the features from the deeper layers (perform as teacher) to shallow early-exit branches (perform as student) to improve the dehazing effect.

Results: For I-HAZE and O-HAZE datasets, better than the other methods, the proposed method achieves the best values of PSNR and SSIM being 17.41dB, 0.813, 18.48dB, and 0.802. Moreover, for real-world images, the proposed method also obtains high quality dehazed results.

Conclusion: Experimental results on both synthetic and real-world images demonstrate that the proposed PMGSDN can effectively dehaze images, resulting in dehazed results with clear textures and good color fidelity.

KEYWORDS

image dehazing, knowledge distillation, attention mechanism, deep learning, computer vision

Introduction

Images captured under haze condition have abnormal brightness and low contrast, which affects the further application in high-level computer vision tasks, such as image super-resolution (Chen et al., 2021a,b) and semantic segmentation. Hence, image dehazing, as a key prerequisite of high-level computer vision tasks, becomes a significant subject in recent years. Generally, the formation of hazy images can be modeled as Equation 1, atmospheric scattering model (also called as physical-model):

$$I(x) = J(x)\,t(x) + A\,(1 - t(x)) \tag{1}$$

where I represents images obtained under haze condition; J represents haze-free images; x represents the pixel location; A and t represent the atmospheric light and transmission map, respectively. Obviously, we cannot directly restore the haze-free images J from the given hazy images I since both the atmospheric light A and transmission map t are undetermined.

To address this problem, early methods use priors obtained from the statistical rule on haze-free images to estimate the atmospheric light and transmission map, then dehaze images *via* the atmospheric scattering model, including dark channel prior (DCP) (He et al., 2011), color-lines prior (CLP) (Fattal, 2014), color attenuation prior (CAP) (Zhu et al., 2015), and non-local dehazing (NLD) (Berman et al., 2016). These methods dehaze favorably in special scenes but tend to over enhance images since unilateral assumptions cannot fit in all situations. With the development of deep learning, some methods (Cai et al., 2016; Ren et al., 2016; Li et al., 2017; Zhang and Patel, 2018) adopt convolutional neural network (CNN) to estimate the atmospheric light and transmission map more accurately and obtain better dehazed images based on the atmospheric scattering model. However, the atmospheric scattering model is an ideal equation, which cannot sufficiently represent the formation of hazy images. Hence, these methods still cause some halos and color distortions.

To solve the problem, some end-to-end dehazing networks (Chen et al., 2019; Liu X. et al., 2019; Qu et al., 2019; Dong et al., 2020; Qin et al., 2020; Zhao et al., 2020) are proposed, which directly restore dehazed images by establishing the mapping between hazy and haze-free images instead of using the atmospheric scattering model. These model-free methods can produce dehazed images with better color fidelity. However, due to trained on synthetic datasets, these model-free methods can perform well on synthetic images but always acquire under-dehazed results when applied to real scenes since synthetic images cannot represent uneven haze distribution and complex illumination condition existing in real scenes. To this end, some novel end-to-end methods (Hong et al., 2020; Shao et al., 2020; Chen et al., 2021; Zhao et al., 2021) combine with model-based methods and achieve better dehazing effects in real scenes. However, these methods cannot exploit features from different depths to improve the guidance efficiency of extra knowledge.

According to the above analyses, we summarize that the existing model-based dehazing methods can effectively restore image texture details but tend to cause color changes and artifacts. By contrast, model-free dehazing methods directly obtain dehazed images with good color fidelity by supervised training. But the dehazing effect is often limited in natural scenes since the training samples are synthetic images. Thus, to improve the dehazing effect, we merge the merits of these two categories *via* self-distillation and propose a physical-model guided self-distillation network for single image dehazing. Moreover, we compare the dehazing effect of the above algorithms on a real-world image. The experimental results are shown in Figure 1. The model-based methods [DCP (He et al., 2011) and DCPDN (Zhang and Patel, 2018)] can restore dehazed images with discriminative textures but suffer from some color and illumination overenhancement. The model-free method MSBDN (Dong et al., 2020) can maintain color fidelity but acquire an under-dehazed image. Better than the other methods, the proposed PMGSDN combines the complementary merits of model-free methods and model-based methods, and obtains high quality dehazed results with natural color and rich details.

As shown in Figure 2, we first build a deep feature extraction network (DFEN) constructed with four attention guided feature extraction blocks (AGFEBs) to effectively extract features from different depths. Moreover, we add three early-exit branches to acquire intermediate dehazed images and optimize the network by a two-stage training strategy. In the first stage, we obtain the preliminary transmission map t_0 and atmospheric light A_0 by two early-exit branches and embed dark channel prior (DCP) into the network to acquire the preliminarily dehazed

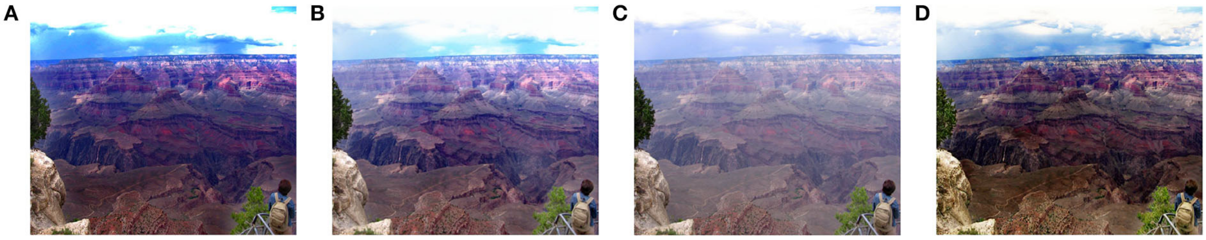


FIGURE 1
Comparative results on a real-world image. **(A)** High contrast result with some color distortion generated by DCP. **(B)** High contrast result with some illumination distortion generated by DCPDN. **(C)** Under-dehazed result with better color fidelity generated by MSBDN. **(D)** Our result, which combines their merits.

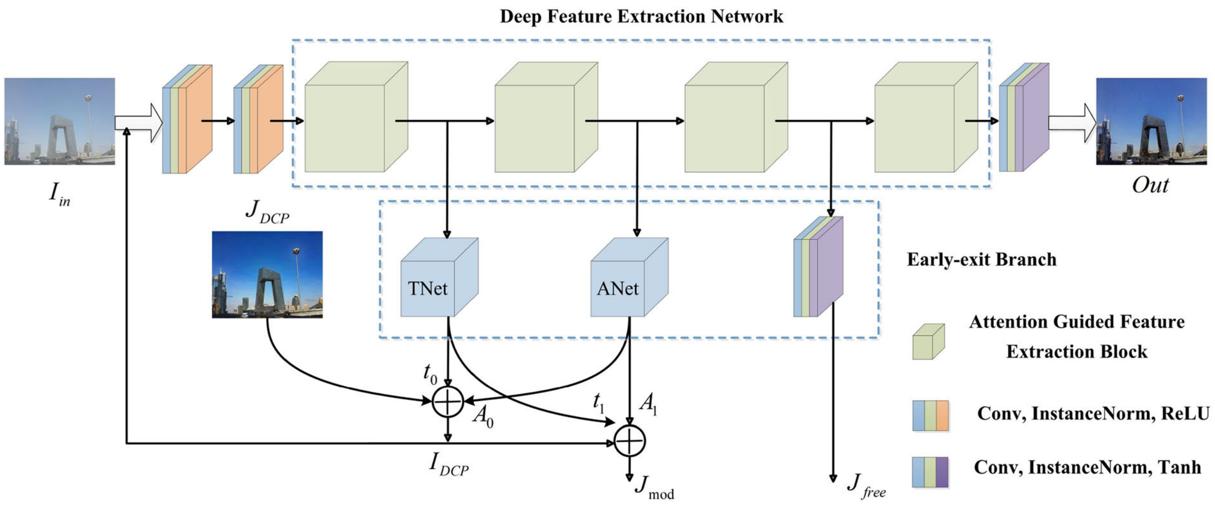


FIGURE 2
The general network structure of PMGSDN. TNet, the network adopted to estimate transmission map. ANet, the network adopted to estimate atmospheric light.

images J_{DCP} base on the hazy input I_{in}. Hence, reconstructed hazy images I_{DCP} can be obtained by substituting the J_{DCP}, A_0, and t_0 into the atmospheric scattering model. In the second stage, we feed the I_{DCP} into the network and obtain the final dehazed images *Out*, the intermediate model-free dehazed images J_{free}, and model-based dehazed images J_{mod} (produced by substituting the intermediate transmission map t_1, atmospheric light A_1, and the I_{DCP} into the model). Considering that these intermediate dehazed images have complementary advantages in terms of image contrast and color fidelity, we combine the merits of them by a one-stage knowledge distillation (see Figure 4), which transfers the knowledge from deeper layers (performs as a teacher) to shallow layers (performs as a student) within the network. We call this distillation strategy as self-distillation, which achieves the joint training and optimization of both teacher and students. For this article, the main contributions are as follows:

1. To improve the dehazing effect, we merge the merits of both model-based dehazing methods and model-free dehazing methods, and propose a physical-model guided self-distillation network for single image dehazing named PMGSDN.
2. In order to improve the feature extraction ability of the network for different depths, we propose an attention guided feature extraction blocks (AGFEB) to construct the deep feature extraction network.
3. To reduce the dependence of the student network on the pretrained teacher model and improve the efficiency of knowledge distillation, we propose a self-distillation strategy and embed the dark channel prior information to the network to further improve the dehazing effect.

Related work

Model-based methods

Model-based methods restore haze-free images using the atmospheric scattering model, where the estimation of transmission map and atmospheric light is a key prerequisite. Early model-based methods (also called prior-based methods) explore various priors concluded from the statistic rule to estimate transmission map and atmospheric light, and then dehaze images *via* the atmospheric scattering model. For example, the dark channel prior (DCP) (He et al., 2011) estimate transmission map based on the observation that clear images have low intensity in at least one of the RGB channels. The color-lines prior (CLP) (Fattal, 2014) constructs a model based on the color lines and estimates the transmission map using the lines' offset. Differently, the color attenuation prior (CAP) (Zhu et al., 2015) builds a linear model to estimate the scene depth and transmission map based on the difference between the brightness and saturation of hazy images. Another method no-local dehazing (NLD) (Berman et al., 2016) estimates the transmission map and acquires dehazed images *via* the hundreds of distinct colors. These prior-based methods achieve favorable dehazing effects but suffer from severe distortion and artifacts.

Recently, some methods estimate transmission map and atmospheric light more accurately by data driving and acquire dehazed images with fewer artifacts. For instance, Ren et al. propose a multi-scale convolution neural network (MSCNN) (Ren et al., 2016) to estimate the transmission map in a coarse-to-fine way. Another method DehazeNet (Cai et al., 2016) adopts Maxout units to effectively extract features and estimate the transmission map. Differently, AODNet (Li et al., 2017) combines these two parameters into one parameter to restore dehazed images. DCPDN (Zhang and Patel, 2018) embeds the atmospheric scattering model into CNN to estimate the atmospheric light and transmission map. These two methods estimate the transmission map and atmospheric light simultaneously and alleviate the cumulative error of two parameter estimations. However, due to the atmospheric scattering model being a simplified model, which cannot sufficiently represent the formation of hazy images, the above two model-based methods still suffer from color and illumination changes.

Model-free methods

Model-free methods directly restore dehazed images *via* an end-to-end network without using the atmospheric scattering model. Due to a huge gap between the features of hazy images and haze-free images, these methods usually increase the network scales and depths to enhance feature extraction ability. For example, the MSBDN (Dong et al., 2020) constructs a multi-scale boosting dehazing network to combine the features from different scales by a dense feature fusion module. FFA (Qin et al., 2020) effectively extracts features and restores dehazed images using a deep network constructed with feature attention blocks. Moreover, GridDehazeNet (Liu X. et al., 2019) and GCANet (Chen et al., 2019), respectively adopt attention mechanisms and gated fusion networks to effectively fuse multi-scale features. Differently, the EPDN (Qu et al., 2019) builds a generative adversarial network to improve the dehazing effect by the adversarial learning between a multi-scale generator and discriminator. Another dehazing method (Zhao et al., 2020) adopts the cycle generative adversarial network to alleviate the paired training constraint. These methods perform well on synthetic images but tend to fail to deal with real-world images due to being trained on synthetic datasets. To address this problem, DA (Shao et al., 2020) builds a bidirectional network to reduce the gap between real-word and synthetic images. PSD (Chen et al., 2021) adopts a committee consists of multi priors to guide the network training and acquire high contrast images but suffer from illumination changes, and RefineDNet (Zhao et al., 2021) embeds DCP and the atmospheric scattering model to reconstruct hazy images and then improves the model's generalization ability *via* unpaired adversarial training. Moreover, some methods also improve deep learning-based algorithms in other computer vision tasks by introducing additional knowledge. For example, Xia et al. (2022) improved the Kernel Correlation Filter algorithm to address the problem that the object tracking algorithm fails to track under the influence of occlusion conditions. Chen et al. (2021c) proposed an image completion algorithm based on an improved total variation minimization method.

Knowledge distillation

Knowledge distillation is first proposed by Hinton (Hinton et al., 2015) to compress the model by transferring the knowledge from a cumbersome teacher network to a compact student network. Recently, knowledge distillation is also applied to the model enhancement through improved learning strategy [including self-learning (Ji et al., 2021; Zheng and Peng, 2022) and mutual learning (Li et al., 2021)]. For example, Hong et al. (2020) applies knowledge distillation to heterogeneous task imitation and guides the student network training using the features extracted from the image reconstruction task. Liu Y. et al. (2019) adopts structure knowledge distillation to transfer the knowledge from a large network to a small semantic segmentation network since semantic segmentation is a structured prediction problem. These two distillation methods both start with a powerful but cumbersome teacher network (a pretrained network) and perform one-way knowledge transfer to a compact student network (a network to be trained). However, two shortcomings exist in them: a powerful teacher network is not always available; a two-stage training process is not efficient. Hence, online distillation and self-distillation are proposed to implement the joint training and optimization of both teacher and student (one-stage training process) by improved learning strategies. Typically, Li et al. (2021) builds a multi-branch network and acquires predicted heatmaps from each branch, which are then assembled (performs as a teacher) to teach each branch (performs as a student) in reverse. However, this method simply aggregates students to form an assembled teacher, which restrains the diversity of students and cannot exploit features from different depths of the network. Hence, we applied self-distillation (Zhang et al., 2021) into our PMGSDN to enhance the generalization ability in real scenes.

Proposed method

Overall structure

As shown in Figure 2, the PMGSDN contains three parts: preprocessing model, a deep feature extraction network, and early-exit branches. In the preprocessing model, we first adopt two 3 × 3 convolutions to preprocess the hazy input F_{in} and change its shape to 32 × 256 × 256, where each convolution is followed by an instance normalization and ReLU function. Moreover, these two convolutions have different parameter settings, the input channel, output channel, kernel size, stride, and padding of the first convolution are 3, 32, 3, 1, and 1, respectively, and the corresponding parameters of the second convolution are set to 32, 32, 3, 1, and 1.

Deep feature extraction network

To effectively extract features from different depths, we feed the preprocessed features into the deep feature extraction

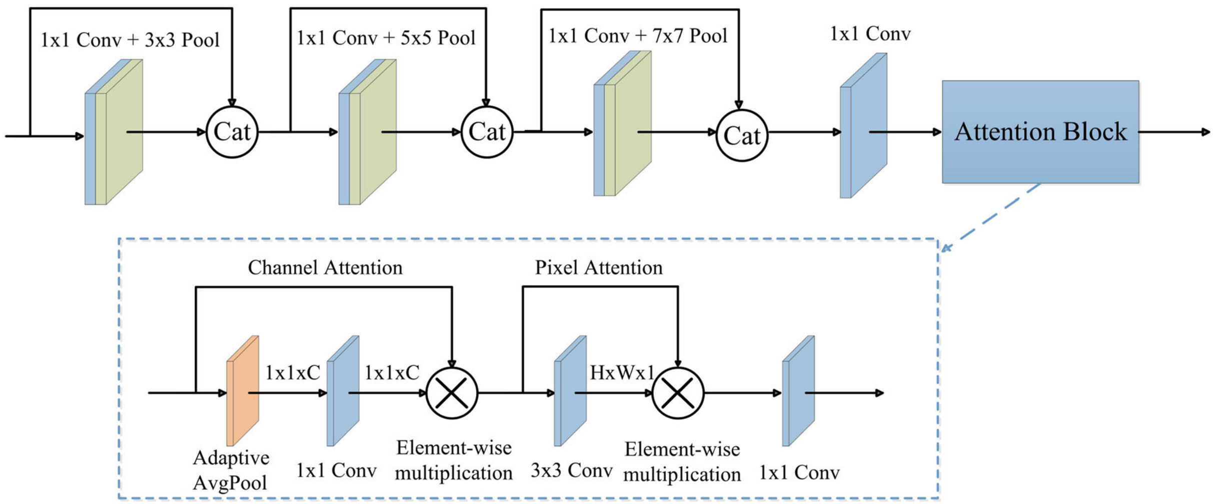


FIGURE 3
The structure of AGFEB. Cat, channel-wise concatenation.

network (DFEN) constructed with four attention guided feature extraction blocks (AGFEBs). After that, a convolution followed by an instance normalization and the Tanh function is utilized to produce the final dehazed images *Out*. The parameter settings of the convolution used here are set to 32, 3, 3, 1, and 1, respectively.

As shown in Figure 3, the proposed AGFEB first extracts features using four convolutions. These convolutions are all point-wise convolutions (1 × 1 convolution) (Zhang and Tao, 2020), where the first three convolutions with pooling layers form different receptive fields and the fourth convolution is utilized for dimension reduction. Note that we replace traditional convolutions with the kernel size of 3 × 3, 5 × 5, and 7 × 7 to point-wise convolutions with 3 × 3, 5 × 5, and 7 × 7 pooling layer, and thus the AGFEB is computationally efficient since no large convolutional kernel is used. Moreover, the first three convolutions combine the features of the current convolution with the features of the last one by channel-wise concatenation to obtain more abundant features. After that, we introduce an attention block consisting of channel attention, pixel attention, and a point-wise convolution to make the network pay more attention to improve feature representation. During the channel attention, an adaptive average pooling is firstly used to generate a channel vector with the shape of 1 × 1 × C and then a 1 × 1 convolution followed by a sigmoid function is utilized to produce channel attention maps, which are used to weigh these inputs *via* element-wise multiplication. After the channel attention, the enhanced features can concern different channel maps unequally and effectively alleviate the global color distortions. Different from the channel attention, the pixel attention first adopts a 3 × 3 convolution followed by a sigmoid function to generate spatial attention maps and then weights the input by element-wise multiplication to pay more attention to high frequency regions, such as textures and structures. Finally, we adopt the point-wise convolution to change the shape to 32 × 256 × 256 and get the output. The parameter settings of the proposed AGFEB are shown in Table 1.

Early-exit branches

To combine both model-based methods and model-free methods, we add three early-exit branches after each AGFEB. The first two branches are named as TNet and ANet to estimate the transmission map and atmospheric light respectively and then acquire the intermediate dehazed images by the atmospheric scattering model. The details of the TNet and ANet can be seen in article (Zhang and Patel, 2018). Moreover, the third branch is constructed with a convolution, an instance normalization, and the Tanh function, which directly acquires intermediate dehazed images in a model-free way, and the parameter settings of the convolution used here are set to 32, 3, 3, 1, and 1, respectively.

Forward prediction and self-distillation

To effectively combine the complementary merits of model-based methods and model-free dehazing methods, we divide the training process into two parts: forward prediction and self-distillation.

Forward prediction

As shown in Figure 2, we divide the forward prediction into two stages. In the first stage, we send the input hazy images I_{in} into the PMGSDN, and obtain the preliminary transmission

TABLE 1 The parameter settings of the proposed AGFEB.

	Conv1	Pool1	Conv2	Pool2	Conv3	Pool3	Conv4	Adaptive Avgpool	Conv5	Conv6	Conv7
Input channel	32	–	64	–	96	–	128	–	32	32	32
Output channel	32	–	32	–	32	–	32	–	32	1	32
Kernel size	1	3	1	5	1	7	1	1	1	3	1
Stride	1	1	1	1	1	1	1	–	1	1	1
Padding	0	1	0	2	0	3	0	–	0	1	0

The convolution and pooling used in AGFEB are expressed as Conv 1 to Conv 7 and Pool 1 to Pool 3 from left to right and top to bottom. Notice that the kernel size of the adaptive average pooling represents the target output size of the feature.

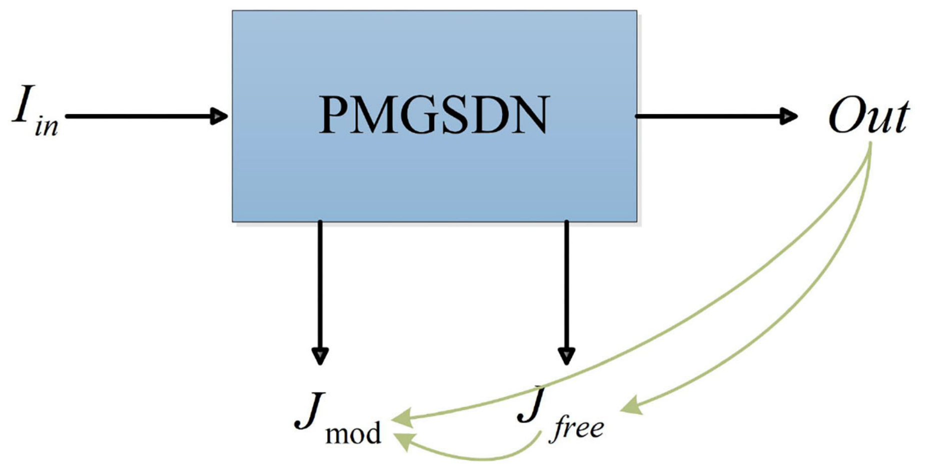


FIGURE 4
Self-distillation.

map t_0 and atmospheric light A_0 by the first two early-exit branches. Meanwhile, we embed dark channel prior (DCP) (He et al., 2011) into a network to acquire the preliminary dehazed images J_{DCP}. Hence, based on the atmospheric scattering model, reconstructed hazy images I_{DCP} can be produced, which can be expressed as Equation 2:

$$I_{DCP} = J_{DCP} t_0 + A_0 (1 - t_0) \tag{2}$$

Compared with the synthetic hazy images F_{in}, the reconstructed hazy images I_{DCP} are more similar to real-world hazy images since the DCP is a statistical rule based on the observation of haze-free images. Hence, in the second stage, we regard the reconstructed hazy images I_{DCP} as the input of PMGSDN and acquire the final dehazed images *Out* by the deep feature extraction network (DFEN). Similar to the first stage, the intermediate transmission map t_1 and atmospheric light A_1 are generated to acquire the model-based dehazed images J_{mod}. Differently, the model-free dehazed images J_{free} are generated simultaneously by the third early-exit branch.

Self-distillation

The intermediate dehazed images J_{mod} and J_{free} are generated by the features from different depths and have complementary advantages in terms of image contrast and color fidelity in local regions. Hence, we adopt a one-stage knowledge distillation called self-distillation to effectively combine the merits of them. As shown in Figure 4, we propose a self-distillation strategy, which builds extra distillation loss among intermediate model-based dehazed images J_{mod}, model-free dehazed images J_{free}, and the final dehazed images *Out*. In this way, the final dehazed images *Out* combine the features from different depths and improve the generalization ability of a model.

Loss function

Several experiments (Liu et al., 2020; Fu et al., 2021) have proven that the combination of pixel-wise and feature-wise loss can effectively improve training efficiency. Hence, the overall loss consists of reconstruct loss and distillation loss, which can be expressed as Equation 3:

$$L_{loss} = L_{rec} + L_{dist} \tag{3}$$

where L_{loss} represents the overall loss, L_{rec} represents the reconstruct loss, and L_{dist} represents the distillation loss.

Reconstruct loss

Previous work (Qin et al., 2020) has shown that pixel-wise loss can rapidly match the feature distribution between the dehazed images and ground truths. Different from L2 loss (mean square error), L1 loss (standard deviation error) can make the network training more stable. Moreover, as a feature-wise loss, the negative structural similarity loss (SSIM) (Wang et al., 2004) can effectively match the luminance, contrast, and structure between two images. Hence, we combine the L1 loss and the negative SSIM as reconstruct loss to train our network, which can be expressed as Eqaution 4:

$$L_{rec} = \sum_{i=1}^{3} \left(\|GT - J_i\|_1 - SSIM\left(GT, J_i\right) \right) \tag{4}$$

where L_{rec} represents the reconstruct loss and GT represents the ground truths. As shown in Figure 4, J_1, J_2, and J_3 represents the final dehazed images Out, intermediate model-based dehazed images J_{mod}, and the model-free dehazed images J_{free}, respectively.

Distillation loss

In our PMGSDN, the dehazed images obtained from deeper layers play a role of teacher and transfer the knowledge to the shallow early-exit branches (performs as a student) within the network. Hence, the Distillation loss L_{dist} can be expressed as Eqaution 5:

$$\begin{aligned} L_{dist} = {} & \left\|Out - J_{free}\right\|_1 + \left\|Out - J_{\text{mod}}\right\|_1 \\ & + \left\|J_{free} - J_{\text{mod}}\right\|_1 \end{aligned} \tag{5}$$

where $\|\cdot\|_1$ represents the L1 loss.

Training and inference

During the training, the deeper AGFEBs are regarded as the teacher and they are utilized to guide the training of shallow AGFEB (student) by a distillation loss among the final dehazed images Out, intermediate model-based dehazed images J_{mod}, and the model-free dehazed images J_{free}. After the training, the whole PMGSND is optimized by model-based methods and model-free methods, which makes the PMGSDN to combine their merits. During the inference process, all of the early-exit branches are dropped so they do not bring additional parameters and computation.

Moreover, to make our manuscript readable, we list out the training process of the proposed algorithm and add it to the manuscript as a pseudocode (Table 2).

TABLE 2 The proposed algorithm.

Training:	
Input:	Hazy input image I_{in}, Corresponding haze-free image (Ground Truth, GT), PMGSDN
Output:	The trained PMGSDN
Step 1	Start the training
Step 2	I_{in}, → PMGSDN get A_0, t_0, and, J_{DCP}
Step 3	A_0, t_0, and, J_{DCP} → atmospheric scattering model, get I_{DCP}
Step 4	I_{DCP} → PMGSDN, get A_1, t_1, J_{free}, and, Out
Step 5	A_1, t_1, and, I_{DCP} → atmospheric scattering model, get J_{mod}
Step 6	GT, Out, J_{mod}, and, J_{free} → Equation (4), get L_{rec}
Step 7	Out, J_{mod}, and, J_{free} → Equation (5), get L_{dist}
Step 8	L_{rec} and L_{dist} → Equation (6), get L_{loss}
Step 9	Back Propagation and update the PMGSDN
Step 10	Repeat the above steps until the end of the training
Inference:	
Input:	Hazy input image I_{in}, The trained PMGSDN
Output:	The final output Out

Experiments

To verify the effectiveness of the proposed PMGSDN, we quantitatively and qualitatively compare it with existing state-of-the-art methods, including DCP (He et al., 2011), DCPDN (Zhang and Patel, 2018), PSD (Chen et al., 2021), MSBDN (Dong et al., 2020), RefineD (Zhao et al., 2021), FFA (Qin et al., 2020), and DA (Shao et al., 2020). Moreover, we conduct an ablation study to verify the effectiveness of each part in PMGSDN.

Datasets

In this article, we adopt the Indoor Training Set (ITS) in RESIDE (Li B. et al., 2019) to train our network, which contains 13990 synthetic hazy images and the corresponding clear images. During the training of the network, we adopt the Synthetic Objective Testing Set (SOTS) indoor datasets as the validation set, which contains 500 synthetic hazy images and the corresponding clear images. For testing, we use three synthetic datasets [I-HAZE (Ancuti C. et al., 2018), O-HAZE (Ancuti C. O. et al., 2018), and HazeRD (Zhang et al., 2017)] to evaluate the performance of the PMGSDN. Among them, the I-HAZE and O-HAZE contain 35 pairs of indoor and 45 pairs of outdoor hazy images. The HazeRD includes 75 pairs of hazy images degraded by different levels of haze. Considering the discrepancy that exists between synthetic and real-world hazy images, we further adopt real-world images from paper (Fattal, 2014) and Unannotated Real Hazy Images (URHI) (Shao et al., 2020) to evaluate the dehazing effect in real scenes.

Implementation details

The proposed method is trained and tested in the Pytorch framework on a PC with the NIVIDIA GeForce RTX 3080 Ti. During the training, we resize input images to 256 × 256, set the batch size to 4, and train the network for 30 epochs. To effectively train the PMGSDN, we adopt the Adam optimizer with a default value for the attenuation coefficient to accelerate the training process ($\beta_1 = 0.9$, $\beta_2 = 0.999$). Moreover, we set the initial learning rate to 0.001 and reduce it by half every five epochs.

Comparisons with state-of-the-art methods

Results on synthetic datasets

Compared with indoor hazy images, outdoor hazy images have different scene depths and transmission maps. Hence, we pay more attention to the comparison results of outdoor images since the proposed PMGSDN is trained on indoor images. As shown in Figure 5, DCP effectively dehaze images but darken the results. Another model-based DCPDN estimates the transmission map and atmospheric light by CNN and generates better dehazed images but suffers from illumination distortion. By contrast, the model-free MSBDN restores dehazed images with better color fidelity but leads to a large amount of residual haze due to the over-fitting on training datasets. The FFA constructs a feature fusion attention network to effectively dehaze images but dims the brightness of results. Another method PSD can generate high contrast images but tend to overenhance the results due to simply guiding the pretrained network by priors. Compared with the above methods, the DA can restore dehazed images with satisfactory visual effect due to the use of domain adaption, and the RefineD restores dehazed images with vivid color but causes residual haze. Only our PMGSDN (see Figure 5I) acquires dehazed images with distinctive textures and abundant details, which verify the effectiveness of our method.

To further validate the performance of the proposed method, two metrics [peak signal-to-noise ratio (PSNR) and structural similarity (SSIM)] are adopted for quantitative comparison. As shown in Table 3, for I-HAZE, the DCP, DCPDN, and PSD perform poorly, which means that the abnormal illuminance and unwanted artifacts degrade the quality of dehazed images. By contrast, the end-to-end MSBDN and DA acquire a high value of PSNR and SSIM. Compared with other methods, the proposed PMGSDN achieves the highest value of these two metrics being 17.41 dB and 0.813, respectively. For O-HAZE, compared with the second-best method DA, the proposed PMGSDN improves the PSNR from 18.37 dB to 18.48 dB and improves the SSIM from 0.712 to 0.802, which validates its generalization ability. For HazeRD, the proposed PMGSDN achieves the PSNR and SSIM being 16.94dB and 0.867, which are slightly lower than that of RefineD.

Results on real-world datasets

Considering the discrepancy between synthetic and real-world hazy images, we further validate the performance of our method on real-world images in Unannotated Real Hazy Images (URHI). As shown in Figure 6, DCP can produce dehazed images with distinct textures but inevitably causes halos and color distortions, which degrade the visual effect of results. Another model-based method DCPDN improves the brightness and contrast of dehazed images but simultaneously introduces some color changes since the atmospheric scattering model is a simplified model. By contrast, the model-free methods can restore dehazed images with better color fidelity but fail to deal with dense haze due to the lacking of extra knowledge as guidance. For example, MSBDN cannot effectively dehaze images due to over-fitting in synthetic datasets. Due to the feature fusion mechanism, FFA can effectively remove the haze in the local area of the image. However, due to the insufficient generalization ability of this method, it still causes residual haze and color changes in some regions. By building a bidirectional network to reduce the gap between synthetic and real-world hazy images, DA dehazes most haze and restores high quality results. Unfortunately, the sky regions are still degraded. Moreover, PSD simply guides the pretrained network by using multi priors, and the results are degraded by a large amount of residual haze. Another method RefineD embeds the DCP into the network and restores high quality images. Better than the above methods, the proposed PMGSDN (see Figure 6I) acquires dehazed images with distinctive textures and vivid color, which verify that it sufficiently exploits the features from different depths by self-distillation and combines the merits of model-based and model-free methods.

To further validate the generalization ability of our PMGSDN, we compare these methods on real-world images (Fattal, 2014). As shown in Figure 7, the DCP still effectively restores the textures but causes obvious color distortion in some regions. Another model-based DCPDN dehazes most haze but suffers from illumination oversaturation. By contrast, MSBDN cannot dehaze effectively in the real scene due to the lacking

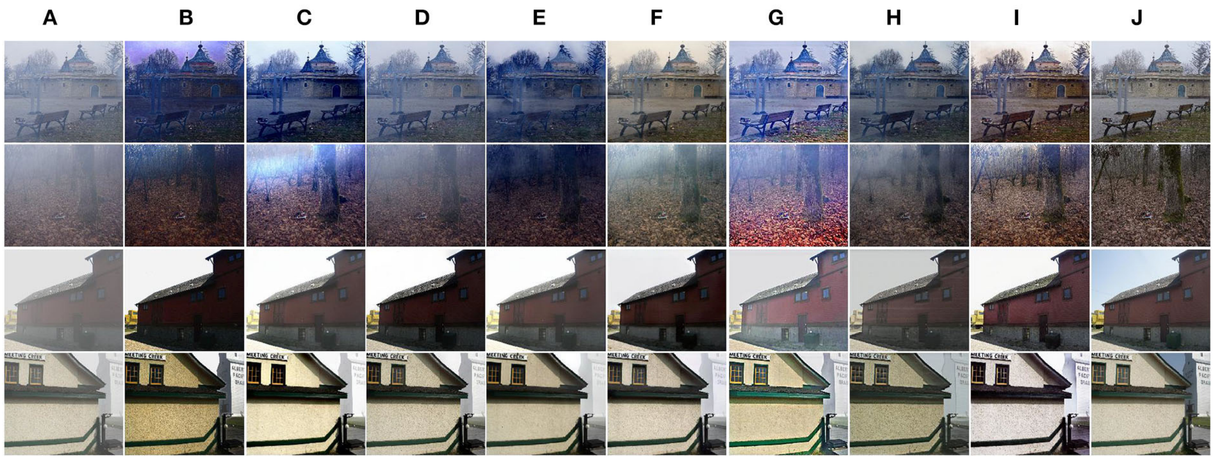


FIGURE 5
Qualitative comparisons on synthetic images from O-HAZE and HazeRD. The above two rows are images in O-HAZE and the others are images in HazeRD. **(A)** Haze. **(B)** DCP. **(C)** DCPDN. **(D)** MSBDN. **(E)** FFA. **(F)** DA. **(G)** PSD. **(H)** RefineD. **(I)** Ours. **(J)** GT.

TABLE 3 **Qualitative comparisons on I-HAZE, O-HAZE, and HazeRD.**

Datasets	Metric	DCP	DCPDN	MSBDN	FFA	DA	PSD	RefineD	Ours
I-Haze	PSNR	12.31 dB	14.27 dB	16.73 dB	13.10 dB	17.10 dB	12.92 dB	16.02 dB	17.41 dB
	SSIM	0.676	0.826	0.798	0.657	0.807	0.712	0.777	0.813
O-Haze	PSNR	14.94 dB	13.79 dB	18.08 dB	14.66 dB	18.37 dB	14.46 dB	17.71 dB	18.48 dB
	SSI	0.672	0.726	0.765	0.713	0.712	0.677	0.692	0.802
HazeRD	PSNR	13.26 dB	15.76 dB	15.23 dB	15.24 dB	16.88 dB	13.56 dB	17.81 dB	16.94 dB
	SSIM	0.795	0.781	0.839	0.745	0.818	0.742	0.850	0.867

Number in red and blue indicate the best and second-best results, respectively.

of knowledge guiding. Another model-free method FFA restores dehazed images with good color fidelity. However, this method neglects the generalization ability in the training process, which leads to the insufficient ability of the model. By contrast, DA removes most haze but suffers from slight color distortion. PSD suffers from illumination oversaturation and the sky regions contain some residual haze. Another method RefineD dehazes effectively and restores visually pleasing dehazed images. Better than the above methods, the proposed PMGSDN acquires high quality images with natural color and discriminative textures, which further shows that it conducts better generalization in real scenes.

In order to objectively evaluate the performance of the algorithm on real world datasets, we further select non-reference criteria that are widely used in image quality assessment for quantitative comparison. These criteria are Natural Image Quality Evaluator (NIQE) and Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE), which can be used to evaluate the effect of haze, color shifts, illumination changes, and other image degraded phenomena. Table 4 gives the quantitative comparison results on the real-world images from paper (Fattal, 2014) and URHI datasets. For images in paper (Fattal, 2014), the proposed method achieves the best values of NIQE (Mittal et al., 2013) and BRISQUE (Mittal et al., 2012) being 2.891 and 13.56, respectively. For URHI datasets, the proposed method also achieves good dehazing results, with NIQE and BRISQUE of 3.705 and 21.38, respectively.

Discussion

To verify the effectiveness of each part of the proposed PMGSDN, we conduct ablation studies to evaluate the performance of the following four key modules: the AGFEB, the guidance of preliminary dehazed images J_{DCP} generated by DCP, the guidance of intermediate dehazed images J_{mod} generated in a model-based way, and the guidance of intermediate dehazed images J_{free} generated in a model-free way. Hence, we construct the following variants: Variant A, the proposed method without the AGFEB, Variant B, the proposed method without the guidance of J_{DCP}, Variant

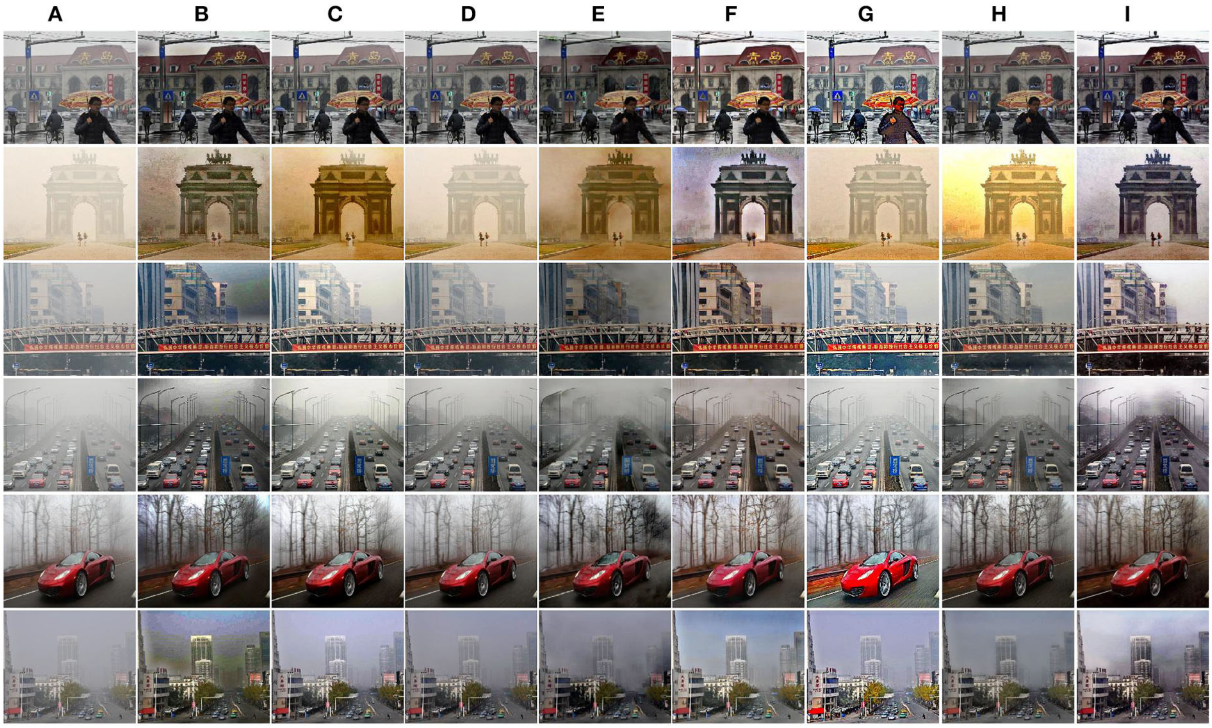


FIGURE 6
Qualitative comparisons on real-world images in URHI. **(A)** Hazy. **(B)** DCP. **(C)** DCPDN. **(D)** MSBDN. **(E)** FFA. **(F)** DA. **(G)** PSD. **(H)** RefineD. **(I)** Ours.

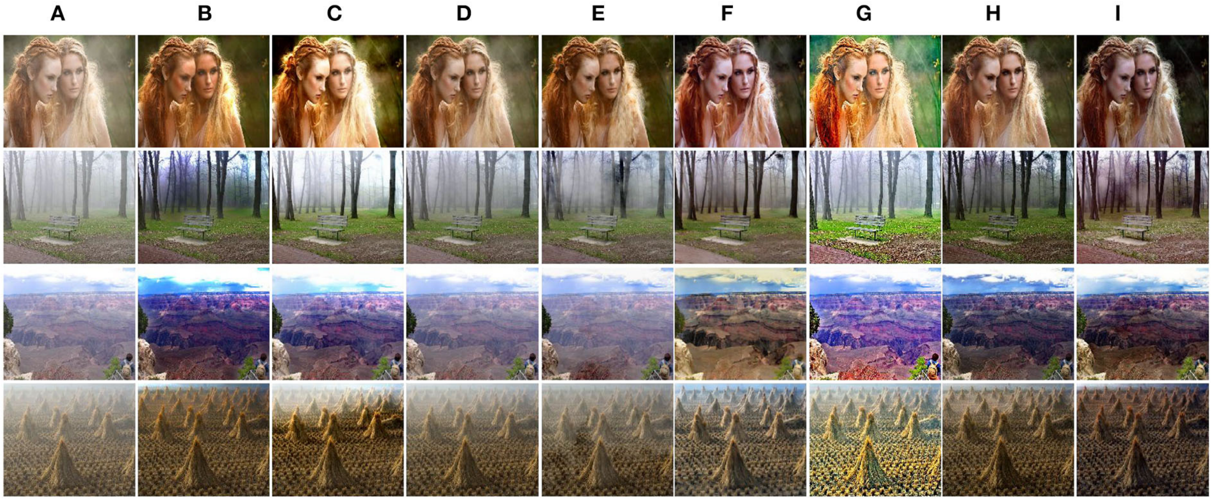


FIGURE 7
Qualitative comparisons on real-world images from Fattal (2014). **(A)** Hazy. **(B)** DCP. **(C)** DCPDN. **(D)** MSBDN. **(E)** FFA. **(F)** DA. **(G)** PSD. **(H)** RefineD. **(I)** Ours.

C, the proposed method without the guidance of J_{mod}, Variant D, the proposed method without the guidance of J_{free}, and Variant E, the proposed PMGSDN. We train these variants on ITS for 30 epochs and test them on I-HAZE and O-HAZE to evaluate the performance of each variant. As shown in Table 5, the proposed method achieves

TABLE 4 Quantitative comparison results on the images in paper (Fattal, 2014) and URHI datasets.

Datasets	Metric	Haze	DCP	DCPDN	MSBDN	FFA	DA	PSD	RefineD	Ours
Images in paeper (Fattal, 2014)	NIQE	3.783	3.521	4.201	4.003	3.671	4.499	3.835	3.047	2.891
	BRISQUE	18.96	13.74	18.97	15.36	16.88	14.47	16.59	14.70	13.56
URHI	NIQE	4.715	3.982	4.058	4.605	3.707	4.388	3.822	3.511	3.705
	BRISQUE	33.73	27.62	27.89	27.36	27.53	21.79	24.26	22.64	21.38

The numbers in red, blue indicate the first and second-best results, respectively. Lower values of NIQE and BRISQUE represent better performance.

TABLE 5 Results of ablation study.

	Variant A	Variant B	Variant C	Variant D	Variant E
IHAZE	15.85 dB	16.05 dB	16.72 dB	17.27 dB	17.41 dB
	0.728	0.719	0.738	0.759	0.813
OHAZE	16.24 dB	16.51 dB	16.33 dB	17.09 dB	18.48 dB
	0.702	0.647	0.692	0.697	0.802

superior performance with PSNR and SSIM both on I-HAZE and O-HAZE, which validates that each part contributes to the PMGSDN.

Conclusion

In this article, we propose a physical-model guided self-distillation network for single image dehazing named PMGSDN. First, we extract abundant features by the deep feature extraction network and acquire two intermediate dehazed images based on the model-based methods and model-free methods, respectively. Second, we embed the dark channel prior information to the network to combine the merits of both model-based methods and model-free methods to improve the dehazing effect. Finally, we adopt self-distillation strategy to improve the dehazing effect. For I-HAZE and O-HAZE datasets, the proposed method achieves the highest values of PSNR and SSIM being 17.41dB, 0.813, 18.48dB, and 0.802, respectively. For real-world images in URHI datasets, the proposed method also achieves the best value of BRISQUE being 21.38. The experimental results on both synthetic and real-world images show that the proposed PMGSDN dehazes more effectively and causes less distortions when compared with the state-of-the-art methods.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This research was funded by the National Natural Science Foundation of China, grant number 61501470.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Ancuti, C., Ancuti, C. O., Timofte, R., and De Vleeschouwer, C. (2018). "I-HAZE: a dehazing benchmark with real hazy and haze-free indoor images," in *Advanced Concepts for Intelligent Vision Systems*, eds J. Blanc-Talon, D. Helbert, W. Philips, D. Popescu, and P. Scheunders, Vol. 11182 (New York, NY: Springer International Publishing), 620–631. doi: 10.1007/978-3-030-01449-0_52

Ancuti, C. O., Ancuti, C., Timofte, R., and De Vleeschouwer, C. (2018). "O-HAZE: a dehazing benchmark with real hazy and haze-free outdoor images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Salt Lake City, UT: IEEE), 867–8678. doi: 10.1109/CVPRW.2018.00119

Berman, D., Treibitz, T., and Avidan, S. (2016). "Non-local image dehazing," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV: IEEE). doi: 10.1109/CVPR.2016.185

Cai, B., Xu, X., Jia, K., Qing, C., and Tao, D. (2016). DehazeNet: an end-to-end system for single image haze removal. *IEEE Trans. Image Process.* 25, 5187–5198. doi: 10.1109/TIP.2016.2598681

Chen, D., He, M., Fan, Q., Liao, J., Zhang, L., Hou, D., et al. (2019). "Gated context aggregation network for image dehazing and deraining," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (Waikoloa Village, HI: IEEE), 1375–1383. doi: 10.1109/WACV.2019.00151

Chen, Y., Liu, L., Phonevilay, V., Gu, K., Xia, R., Xie, J., et al. (2021a). Image super, resolution reconstruction based on feature map attention mechanism. *Appl. Intell.* 51, 4367–4380. doi: 10.1007/s10489-020-02116-1

Chen, Y., Phonevilay, V., Tao, J., Chen, X., Xia, R., Zhang, Q., et al. (2021b). The face image super-resolution algorithm based on combined representation learning. *Multimed. Tools Appl.* 80, 30839–30861. doi: 10.1007/s11042-020-09969-1

Chen, Y., Zhang, H., Liu, L., Tao, J., Zhang, Q., Yang, K., et al. (2021c). Research on image inpainting algorithm of improved total variation minimization method. *J. Ambient Intell. Humaniz. Comput*. doi: 10.1007/s12652-020-02778-2

Chen, Z., Wang, Y., Yang, Y., and Liu, D. (2021). "PSD: principled synthetic-to-real dehazing guided by physical priors," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE), 7176–7185. doi: 10.1109/CVPR46437.2021.00710

Dong, H., Pan, J., Xiang, L., Hu, Z., Zhang, X., Wang, F., et al. (2020). "Multi-scale boosted dehazing network with dense feature fusion," in *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (Seattle, WA: IEEE), 2154–2164. doi: 10.1109/CVPR42600.2020.00223

Fattal, R. (2014). Dehazing using color-lines. *ACM Trans. Graph.* 34, 1–14. doi: 10.1145/2651362

Fu, M., Liu, H., Yu, Y., Chen, J., and Wang, K. (2021). "DW-GAN-A discrete wavelet transform GAN for nonhomogeneous dehazing," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Nashville, TN: IEEE), 203–212. doi: 10.1109/CVPRW53098.2021.00029

He, K., Sun, J., and Tang, X. (2011). Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 2341–2353. doi: 10.1109/TPAMI.2010.168

Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. *arXiv [Preprint]*. (2015). arXiv: 1503.02531. doi: 10.48550/arXiv.1503.02531

Hong, M., Xie, Y., Li, C., and Qu, Y. (2020). "Distilling image dehazing with heterogeneous task imitation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA: IEEE), 3459–3468. doi: 10.1109/CVPR42600.2020.00352

Ji, M., Shin, S., Hwang, S., Park, G., and Moon, I.-C. (2021). "Refine myself by teaching myself: feature refinement via self-knowledge distillation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Nashville, TN: IEEE), 10659–10668. doi: 10.1109/CVPR46437.2021.01052

Li, B., Peng, X., Wang, Z., Xu, J., and Feng, D. (2017). "AOD-Net: all-in-one dehazing network," in *2017 IEEE International Conference on Computer Vision (ICCV)* (Venice: IEEE), 4780–4788. doi: 10.1109/ICCV.2017.511

Li, B., Ren, W., Fu, D., Tao, D., Feng, D., Zeng, W., et al. (2019). Benchmarking single-image dehazing and beyond. *IEEE Trans. Image Process.* 28, 492–505. doi: 10.1109/TIP.2018.2867951

Li, Z., Ye, J., Song, M., Huang, Y., and Pan, Z. (2021). "Online knowledge distillation for efficient pose estimation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (Montreal, QC: IEEE), 11720–11730. doi: 10.1109/ICCV48922.2021.01153

Liu, J., Wu, H., Xie, Y., Qu, Y., and Ma, L. (2020). "Trident dehazing network," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (Seattle, WA: IEEE), 1732–1741. doi: 10.1109/CVPRW50498.2020.00223

Liu, X., Ma, Y., Shi, Z., and Chen, J. (2019). "GridDehazeNet: attention-based multi-scale network for image dehazing," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (Seoul: IEEE), 7313−7322. doi: 10.1109/ICCV.2019.00741

Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, Z., et al. (2019). "Structured knowledge distillation for semantic segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, CA: IEEE), 2599–2608. doi: 10.1109/CVPR.2019.00271

Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.* 21, 4695–4708. doi: 10.1109/TIP.2012.2214050

Mittal, A., Soundararajan, R., and Bovik, A. C. (2013). Making a "completely blind" image quality analyzer. *IEEE Signal Process. Lett.* 20, 209–212. doi: 10.1109/LSP.2012.2227726

Qin, X., Wang, Z., Bai, Y., Xie, X., and Jia, H. (2020). FFA-Net: feature fusion attention network for single image dehazing. *Proc. AAAI Conf. Artif. Intell.* 34, 11908–11915. doi: 10.1609/aaai.v34i07.6865

Qu, Y., Chen, Y., Huang, J., and Xie, Y. (2019). "Enhanced Pix2pix dehazing network," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE), 8152–8160. doi: 10.1109/CVPR.2019.00835

Ren, W., Si, L., Hua, Z., Pan, J., and Yang, M. H. (2016). "Single image dehazing via multi-scale convolutional neural networks," in *European Conference on Computer Vision* (Cham). doi: 10.1007/978-3-319-46475-6_10

Shao, Y., Li, L., Ren, W., Gao, C., and Sang, N. (2020). "Domain adaptation for image dehazing," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE), 2805–2814. doi: 10.1109/CVPR42600.2020.00288

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13, 600–612. doi: 10.1109/TIP.2003.819861

Xia, R., Chen, Y., and Ren, B. (2022). Improved anti-occlusion object tracking algorithm using Unscented Rauch-Tung-Striebel smoother and kernel correlation filter. *J. King Saud Univ - Comput. Inf. Sci.* 34, 6008–6018. doi: 10.1016/j.jksuci.2022.02.004

Zhang, H., and Patel, V. M. (2018). "Densely connected pyramid dehazing network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 3194–3203. doi: 10.1109/CVPR.2018.00337

Zhang, J., and Tao, D. (2020). FAMED-Net: a fast and accurate multi-scale end-to-end dehazing network. *IEEE Trans. Image Process.* 29, 72–84. doi: 10.1109/TIP.2019.2922837

Zhang, L., Bao, C., and Ma, K. (2021). Self-distillation: towards, efficient, and compact neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 1–1. doi: 10.1109/TPAMI.2021.3067100

Zhang, Y., Ding, L., and Sharma, G. (2017). "HazeRD: an outdoor scene dataset and benchmark for single image dehazing," in *2017 IEEE International Conference on Image Processing (ICIP)* (Beijing: IEEE), 3205–3209. doi: 10.1109/ICIP.2017.8296874

Zhao, C.-Y., Jia, R.-S., Liu, Q.-M., Sun, H.-M., and Sun, H.-B. (2020). Image dehazing method via a cycle generative adversarial network. *IET Image Process.* 14, 4240–4247. doi: 10.1049/iet-ipr.2020.0928

Zhao, S., Zhang, L., Shen, Y., and Zhou, Y. (2021). RefineDNet: a weakly supervised refinement framework for single image dehazing. *IEEE Trans. Image Process.* 30, 3391–3404. doi: 10.1109/TIP.2021.3060873

Zheng, Z., and Peng, X. (2022). "Self-guidance: improve deep neural network generalization via knowledge distillation," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (IEEE), 3451–3460. doi: 10.1109/WACV51458.2022.00351

Zhu, Q., Mai, J., and Shao, L. A. (2015). Fast single image haze removal algorithm using color attenuation prior. *IEEE Trans. Image Process.* 24, 3522–3533. doi: 10.1109/TIP.2015.2446191

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 10 January 2023
DOI 10.3389/fnbot.2022.1082550

OPEN ACCESS

EDITED BY
Praveen Kumar Donta,
Vienna University of Technology,
Austria

REVIEWED BY
Rambabu Pemula,
Raghu Engineering College, India
P. Suman Prakash,
G. Pullaiah College of Engineering and
Technology (GPCET), India

*CORRESPONDENCE
Yong Liu
✉ liuy1602@njust.edu.cn

RECEIVED 28 October 2022
ACCEPTED 19 December 2022
PUBLISHED 10 January 2023

CITATION
Xue T, Zheng D, Yan J and Liu Y (2023)
A multi-scale robotic tool grasping method for robot state segmentation masks.
Front. Neurorobot. 16:1082550.
doi: 10.3389/fnbot.2022.1082550

A multi-scale robotic tool grasping method for robot state segmentation masks

Tao Xue, Deshuai Zheng, Jin Yan and Yong Liu*

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, Jiangsu, China

As robots begin to collaborate with humans in their daily work spaces, they need to have a deeper understanding of the tasks of using tools. In response to the problem of using tools in collaboration between humans and robots, we propose a modular system based on collaborative tasks. The first part of the system is designed to find task-related operating areas, and a multi-layer instance segmentation network is used to find the tools needed for the task, and classify the object itself based on the state of the robot in the collaborative task. Thus, we generate the state semantic region with the "leader-assistant" state. In the second part, in order to predict the optimal grasp and handover configuration, a multi-scale grasping network (MGR-Net) based on the mask of state semantic area is proposed, it can better adapt to the change of the receptive field caused by the state semantic region. Compared with the traditional method, our method has higher accuracy. The whole system also achieves good results on untrained real-world tool dataset we constructed. To further verify the effectiveness of our generated grasp representations, A robot platform based on Sawyer is used to prove the high performance of our system.

KEYWORDS

human-robot collaboration, instance segmentation, robotic grasping, grasp detection, robotic grasp platform

1. Introduction

With the increasingly serious aging of the population, how to provide effective homecare for the growing elderly population has ushered in new challenges and changes, especially the COVID-19 epidemic, which makes the need for homecare for the elderly extremely urgent. In order to prevent the elderly from using tools incorrectly and to ensure the safety of tools when using them, we effortlessly draw on our understanding of the functions that tools and their parts provide. Using vision, we can identify the function of the part, so we can find the right tool part for our operation. As robots like PR2, Asimo, and Baxter begin to collaborate with humans in homecare industry, they will also need us to have a more detailed understanding of the tools involved in the task.

When completing tasks through human-robot collaboration, robots are designed to provide more assistance to humans, rather than let the robot perform all tasks autonomously. There are two reasons for this. Firstly, the type and level of knowledge and the training required for robots to complete tasks on their own are difficult to establish and collect. Secondly, despite the significant progress made in robotics such as manipulation (Kroemer et al., 2015; Fu et al., 2016), robots are still far from having the fine manipulation capabilities required for tasks such as furniture assembly (for example, using a screwdriver on small screws). Therefore, we hope that the robot can choose the behavior suitable for the robot, while letting the human worker perform the action more suitable for the human. For example, robots may provide supportive or transmit behaviors, such as stabilizing components or bringing heavy components required for assembly (Mangin et al., 2017), while human workers can perform operations that require more adaptability to tasks, such as screwing screws. Therefore, in the task of using various tools through human-robot collaboration, how to understand the task requirements and assign them to different states of robots and humans to grasp tools is a very critical issue.

Brahmbhatt et al. (2019) used thermal camera to study human grasping contacts on 50 household objects textured with contact maps for two tasks. Fang et al. (2020) developed a learning-based approach for task-oriented grasping in simulation with reinforcement learning. Liu et al. (2020) considered a broad sense of context and proposed a data-driven approach to learn suitable semantic grasps. These methods are able to solve the problem of understanding task requirements related to grasp tools through pixel-level enlightening segmentation of a small group of known object categories (Do et al., 2018). However, for collaborative tasks, there is still a lack of consideration for different states that lead to different tool grasping representation. In order to realize the understanding of tools according to different state definitions of robots, we constructed a tool classification dataset used to analyze the different states played by robots when grasping various tools.

We recruited some volunteers to take on different states in grasping the tools in the dataset. And we recorded the grasping areas corresponding to different states and counted these positions. We borrowed the idea of region classification and proposed the state semantics (grasp and handover) region, that is, different states often make people grasp different position of tools. Based on the knowledge of this region, we define two types of robot states: active operator and assisting passer, corresponding to the previous semantics "grasp" and "handover."

The main contributions of our work mainly include the following four points:

1. We proposed a modular system for multi-states tool grasping task under human-robot interaction, which can realize the collaborative grasping and interaction of humans and robots based on tasks.
2. A multi-layer instance segmentation network is proposed to complete the classification of operating areas for task-related tools. Therefore, in different tasks, we can find the most suitable grasping area for humans or robots in different states.
3. Considering that grasping based on the local semantic region of the tool will increase the variation range of the receptive field, we propose a multi-scale grasping convolutional network MGR-Net based on state semantics to improve the prediction accuracy of the network.
4. We collected real-world tool images through "realsense" camera as a test set, and the experimental results show that our method performs well on untrained real-world tool images. Furthermore, we used robotic platform based on Sawyer to validate our grasping representation.

The other chapters of this article are arranged as follows. In Section 2, we briefly review related literature. In Section 3, we detail the proposed grasping framework based on semantic state area. In Section 4, Our experimental results are presented. Finally, we conclude this work in Section 5.

2. Related work

Learning to use an item as a tool requires an understanding of what it helps to achieve, the properties of the tool that make it useful, and how the tool must be manipulated in order to achieve the goal. In order to further meet the operational requirements of our robots based on different states, the tool grasping tasks under different states can be divided into the following three aspects:

1. Detection of tools related to different tasks.
2. Research on the properties of the tool itself.
3. Robotic grasping detection of tools.

2.1. Task-related tool detection

The earliest classification of tasks is mostly to find corresponding task objects in multiple objects. With the great power of machine learning in classification, researchers find that novel objects grasp detection can be classified into two parts, which is graspable or ungraspable. SVM has been widely used in grasp feature classification (Fischinger et al., 2015; Ten Pas and Platt, 2018). Ten Pas and Platt (2018) used knowledge of the geometry of a good grasp to improve detection. Through sampling lots of hand configuration as the input features, they used the notion of an antipodal grasp to classify these grasp hypotheses. Deep learning methods are also been applied in grasp detection. Lenz et al. (2015) presented a two-step cascaded system with two deep networks and ran at 13.5 s per frame with an accuracy of 93.7%.

In order to better identify task-related tools among multiple types of tools and avoid the interference of irrelevant tools, instance segmentation methods are introduced to achieve more accurate tool detection accuracy. Top-down methods (He et al., 2017; Chen et al., 2020) solve the problem from the perspective of object detection. For example, first detecting an object, then segmenting it in the box. Recently, the anchor-free object detectors were used by some researchers and got good results (Tian et al., 2019). Bottom-up methods (Liu et al., 2017; Gao et al., 2019) view the task as a label-then-cluster problem. These method learn the per-pixel embeddings and then cluster them into groups. The latest direct method (SOLO) (Wang et al., 2020a) no longer relies on box detection or embedding learning, and deals with instance segmentation directly. Wang et al. (2020b) appreciate the basic concept of SOLO and further explore the direct instance segmentation solutions.

2.2. Tool attribute classification

The above methods can identify objects of known classes very well. However, in the case of using a spoon, the robot needs to know which part of the spoon to grasp and which part to hold the soup. Work on grasp affordances tends to focus on robust interactions between objects and the autonomous agent. It is typically limited to a single affordance per object. Moreover, affordance labels tend to be assigned arbitrarily instead of through data-driven techniques to collect human-acceptable interactions about grasping. Krüger et al. (2011) focus on relating abstractions of sensory-motor processes with object structures [e.g., object-action complexes (OACs)], and capture the interaction between objects and associated actions through an object affordance. Others use purely visual input to learn affordances to relate objects and actions through deep learning or supervised learning techniques (Hart et al., 2015). Chu et al. (2019) presented a novel framework to predict the affordance of objects *via* semantic segmentation.

It is worth considering that in the interactive use of tools, robots not only need to find the task-related tools and operating areas, but also clarify the state of the robot at this time, whether it is the "leader" or the "assistant" of the task. However, the previous classification of tool attributes at this time is not sufficient to meet this goal, they only consider the case where the robot is a single operator. In order to solve this problem, based on the attributes generated by the classification of tool functions, we focus on the grasping operation during interactive tasks. Through data-driven technology, the functional attributes of the tool are combined with the state of the robot to find the optimal grasping area of the tool for the robot under different states.

2.3. Robotic grasping detection

Deep learning has been a hot topic of research since the advent of ImageNet success and the use of GPU's and other fast computational techniques. Also, the availability of affordable RGB-D sensors enabled the use of deep learning techniques to learn the features of objects directly from image data. Recent experimentations using deep neural networks (Schmidt et al., 2018; Zeng et al., 2018) proved that they were quite efficient when calculating stable grasp configurations. Guo et al. (2017) fused tactile and visual data to train hybrid deep architectures. Mahler et al. (2017) trained a Grasp Quality Convolutional Neural Network (GQ-CNN) with only synthetic data from Dex-Net 2.0 grasp planner dataset. Levine et al. (2018) presented a method for learning hand-eye coordination for robotic grasping from monocular images. They use a CNN for grasp success prediction, and a continuous servoing mechanism used this network to continuously control the manipulator. Antanas et al. (2019) proposed a probabilistic logic framework that is said to improve the grasping capability of a robot with the help of semantic object parts. This framework combines high-level reasoning with low-level grasping. The high-level reasoning leverages symbolic world knowledge through comprising object-task affordances, categories, and task-based information while low-level reasoning depends on visual shape features.

Most of these grasp synthesis approaches are enabled by representing the grasp as an oriented rectangle in the image (Dong et al., 2021). Kumra et al. (2020) used an improved version of grasp representation, complemented by a novel convolutional network, which improves the accuracy of robotic grasping. Depierre et al. (2021) introduced a new loss function, which associates the regression of the grab parameters with the score of the grabability. Dong et al. (2022) used the transformer network as an encoder to obtain global context information. Shukla et al. (2022) proposed GI-NNet model based on inception module, it can achieve better results under limited data sets, but it is less adaptable to big data. These grasping methods tend to focus on the tool itself, ignoring the impact of different tasks on grasping. Especially in human-computer interaction tasks, different states prompt the robot to grasp different parts of the tool. In order to solve the problem of robot grasping under human-computer interaction, we modified the grasping representation of the tool based on the different state semantic regions of the tool. Through an improved grasping neural network, the accuracy of grasping detection is improved.

3. Method

In this human-robot collaboration work, we consider the operating area of the tool when people are in the two different states of leader and assistant. And let our network learn this selection rule, so that when the robot assists the human or

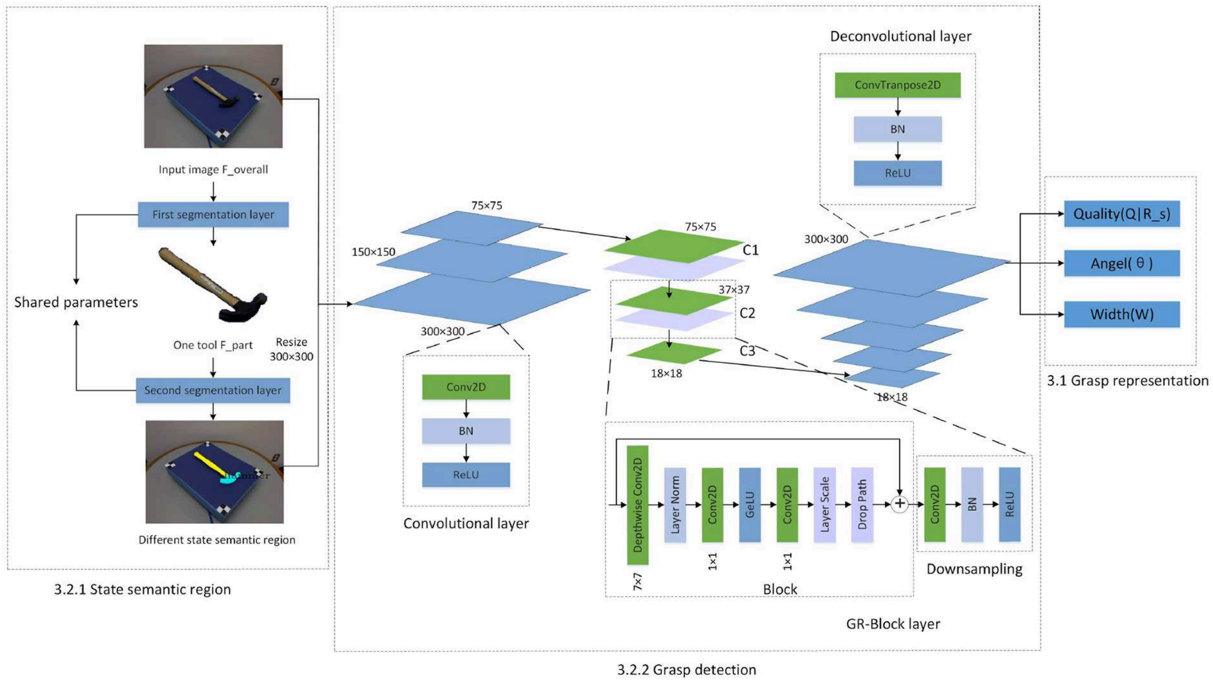


FIGURE 1
Our MGR-Net based on state semantic regions.

the robot operates under the guidance of the human, it can find the relevant task position as much as possible. In this paper, in order to study how to generate the robot grasp detection problem under different states, the following state semantic region classification and grasping detection framework of collaborative task are proposed, as shown in the Figure 1.

Our grasping detection network mainly consists of two parts. First, finding the task-related state semantic region of object. Second, finding the most suitable grasp configuration for robots or humans based on different state semantic regions.

3.1. Grasp representation

In this work, we define the robot grasping detection problem as predicting unknown objects from the n-channel image of the scene and assigning states based on the task according to the provided task description, so as to carry out the corresponding grasping and execute it on the robot. Instead of the five-dimensional grip representation used in Kumra and Kanan (2017), we use an improved version similar to the grasp representation proposed by Morrison et al. (2020). Considering that the optimal grasping configuration of the robot will change in different state states, we incorporate the attribute of the state semantic area into the robot frame, and change the grasping posture to be expressed as:

$$G = (P, \theta, W, Q|R_s) \tag{1}$$

Among them, $P = (x, y, z)$ is the center position of the tool tip, θ is the rotation of the tool around the z-axis, W is the required width of the tool, R_s represents the state semantic area, and $Q|R_s$ represents the grasp score of the corresponding state area.

The grasp quality score Q is the grasp quality of each point in the image, and is expressed as a fractional value between 0 and 1, with values closer to 1 indicating a greater chance of successful grasping. θ represents a measure of the amount of angular rotation at each point required to grasp the object of interest, expressed as a value in the range $[\frac{-\pi}{2}, \frac{\pi}{2}]$. W is the desired width, expressed as a measure of uniform depth, and expressed as a value in the range $[0, W_{max}]$ pixels. W_{max} is the maximum width of the gripper.

3.2. Grasp detection network

3.2.1. State semantic region

We input image $F_{overall}$ to the first layer of tool segmentation network. Through the generated mask, we construct the input image F_{part} of the second layer of state semantic segmentation

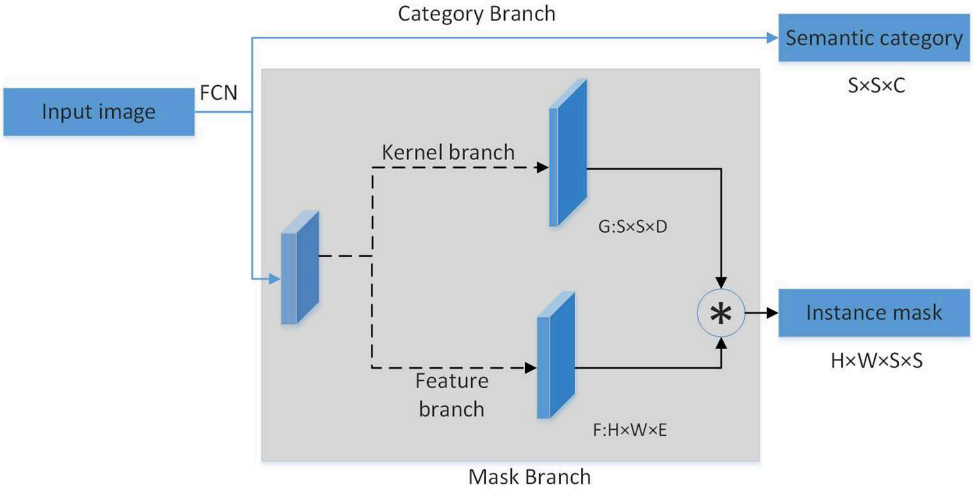


FIGURE 2
State semantic region segmentation.

network. Based on the state that the robot assumes in the task, the second layer finally generates semantic regions related to the robot state. More descriptions of the tool datasets will be introduced in Section 4.1. The modules in the segmentation layer are shown in Figure 2.

Two segmentation layers is designed to achieve different purposes. The first layer of the overall segmentation layer finds out the mask of the task-related object in the multi-object environment, which includes two branches: (1) Category Branch is responsible for predicting the semantic category of the object. (2) Mask Branch is responsible for predicting the mask region of the object. The second layer further divides the task object based on the state to obtain the state semantic area of the object. The state semantic area mainly contained in this layer is the "grasp" area as the state of leader and the "handover" area as the state of assistant. The difference between this layer and the first layer is: (1) Category Branch is responsible for predicting the state semantic category of the task area of the object. (2) Mask Branch is responsible for predicting the mask of the semantic area of different states of the object. Each layer uses FPN behind the backbone network to cope with the size. After each layer of FPN, the above two parallel branches are connected to predict the category and position. The number of grids in each branch is correspondingly different. Small examples correspond to more grids.

Category Branch is responsible for predicting the semantic category of each task area of the object. Each grid predicts the category S×S×C. The mask branch is decomposed into mask kernel branch and mask feature branch, which correspond to the learning of the convolution kernel and the learning of features, respectively. The output of the two branches is finally combined into the output of the entire mask branch. For each grid, the kernel branch predicts the D-dimensional output, which represents the predicted weight of the convolution kernel, and D is the number of parameters. So for the number of grids of S×S, the output is S×S×D. Mask feature branch is used to learn the expression of features. Its input is the features of different levels extracted by backbone+FPN, and the output is the mask feature of H×W×E, denoted by F.

3.2.2. Grasp detection

Feature output is similar to Kumra et al. (2020), and also contains three different prediction maps ($Q|R$, angle, width) represented by the grasping posture, as shown in the Figure 1. But the difference is that since our grasping posture contains the content of the state assignment area, our grasping score is also closely related to the character area.

The input image and the state semantic region mask corresponding to the task are sent to the convolutional layer together. The convolutional layer consists of conv2d layer, batch normalization (BN) layer and relu layer. The output of the convolutional layer is fed to 3 GB-Block layers (C1–C3), the first two GR-Block layer contains a Block and Downsampling, as shown in the Figure 1. We designed this Block from Liu et al. (2022). Three conv2d layers are used in Block with different kernel functions, and Layer Norm replaces Batch Norm for better effect. Since we focus on the semantic area above the object rather than the object itself, the change in the size of the object will increase the difficulty of detection. We use three Block of different sizes to obtain different receptive fields to improve the detection accuracy. A downsampling module is to connect two Block of different sizes, as shown in the Figure 1. After that, in order to more easily interpret and preserve the spatial

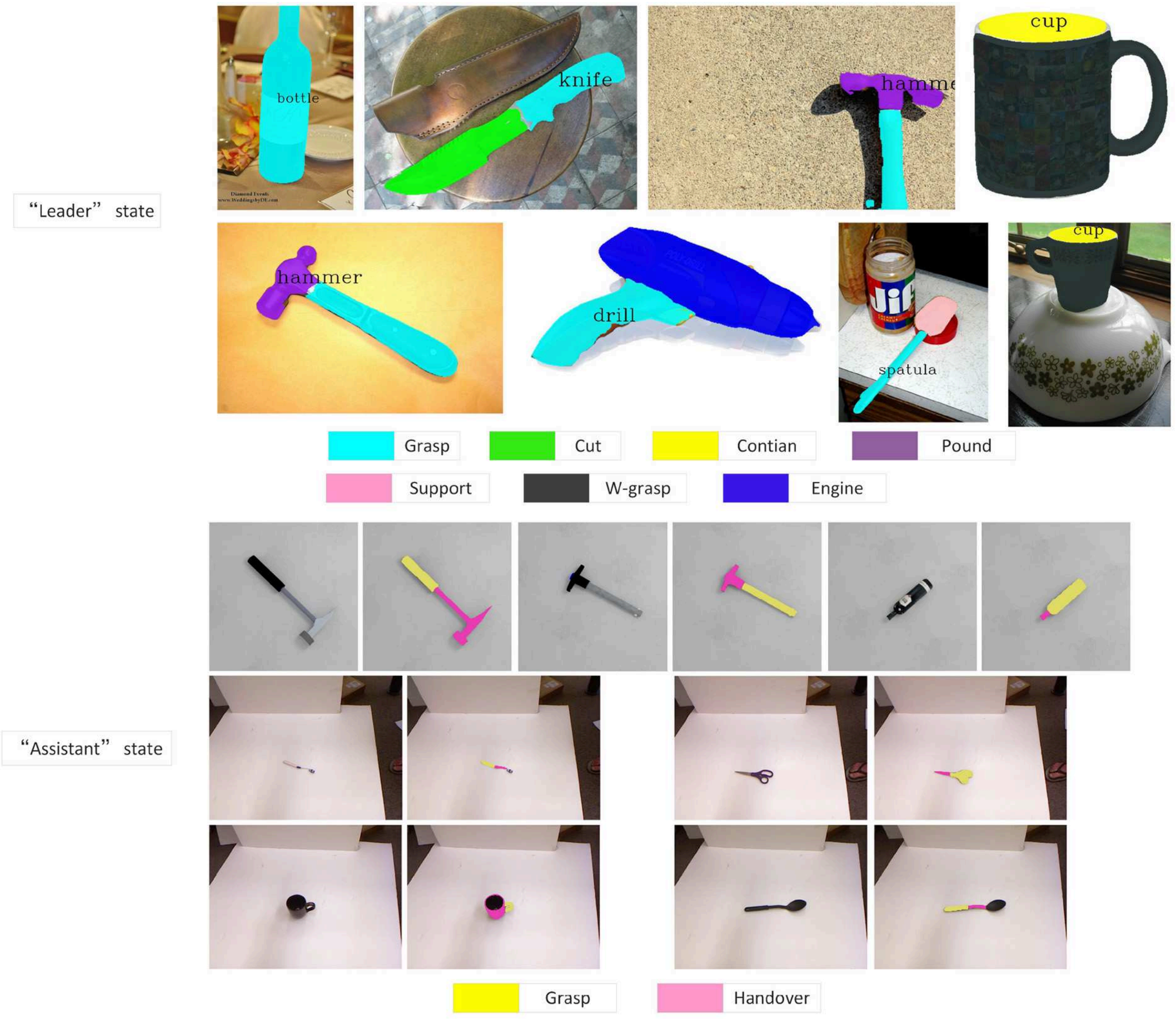


FIGURE 3
Segmentation results based on “leader” and “assistant” state.

characteristics of the image after the convolution operation, we use five deconvolutional layers to upsample the image. Therefore, we get the same size image at the output as the input. Grasp representation is generated as network output from the deconvolutional layer.

3.2.3. Loss function

For each input image p, combined with the local attribute region image p_k generated by its different state semantic regions M, our grasping network is optimized by the following loss function:

$$loss(G_k, \hat{G_k}) = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{2}$$

where s_i is given by:

$$s_i = \begin{cases} 0.5 \cdot (\hat{G}_{ki} - f(G_{ki}))^2, & if|\hat{G}_{ki} - f(G_{ki})| < 1 \\ |\hat{G}_{ki} - f(G_{ki})| - 0.5 & otherwise \end{cases} \tag{3}$$

G_k is the grasp generated by the network corresponding to p_k and $\hat{G_k}$ is the ground truth grasp.

4. Experiment

We implemented our detection network in PyTorch and the computer configuration used in the experiment is intel core I7-8700 CPU and NVIDIA 2080ti GPU. The following experimental part mainly contains three pieces.

TABLE 1 Performance on IIT-AFF dataset.

	DeepLab (Chen et al., 2017)	Affordance-net (Do et al., 2018)	RAN-ResNet50 (Zhao et al., 2020)	Our method
Contain	68.84	79.61	80.20	87.10
Cut	55.23	75.68	78.04	72.80
Display	61.00	77.81	79.14	91.20
Engine	63.05	77.50	81.22	85.50
Grasp#1	54.31	68.48	71.59	82.60
Hit	58.43	70.75	88.52	91.00
Pound	54.25	69.57	76.91	81.90
Support	54.28	69.81	80.12	78.90
Grasp#2	–	–	79.27	88.86
Handover#2	–	–	77.96	80.08

4.1. Dataset

In order to meet the image input required by our network, we constructed a dataset of collaboration tools. We selected 6,000 tool images from IIT-AFF Dataset (Nguyen et al., 2017), UMD Dataset (Myers et al., 2014), Cornell Grasp Dataset and Jacquard Grasping Dataset (Depierre et al., 2018). We resize the images in the tool dataset to the same size. This tool dataset is used for two networks. One is mainly used for the classification of the object task area. At this time, 90% of the images in the dataset are used as the training set, and the rest are the test set. Another use is tool grasp detection based on the robot's state. The training set at this time comes from the jacquard part of the tool dataset, there are 4,000 images, and the remaining jacquard images are used as the test set together with other parts of the dataset. The extended version of Cornell Grasp Dataset comprises of 1,035 RGB-D images with a resolution of 640 × 480 pixels of 240 different real objects with 5,110 positive and 2,909 negative grasps. The annotated ground truth consists of several grasp rectangles representing grasping possibilities per object. The Jacquard Grasping Dataset is built on a subset of ShapeNet which is a large CAD models dataset. It consists of 54 k RGB-D images and annotations of successful grasping positions based on grasp attempts performed in a simulated environment. In total, it has 1.1 M grasp examples.

4.2. Task area

In this section, we mainly discuss the results of semantic region classification. Different states are given to the robot according to the task, and the robot has a more specific functional area classification for the tool. As shown in Figure 3, when the robot acts as the "leader," the tools are classified according to their affordance. Such classification enables the robot to grasp more accurately, and avoids damage to the object or the gripper caused by the wrong grasping position. When the robot acts as an "assistant," it always expects the human to grasp the most suitable position for grasping. Therefore, the robot needs to avoid this grasping area as much as possible and find a suitable area for handover. Through the delivery of the robot, human can always grasp the tool most efficiently and safely. For example, when passing scissors, such classification can avoid being accidentally injured by scissors due to people's carelessness.

To further test the effectiveness of our two-layer segmentation network, we compare it with other methods on the IIF-AFF Dataset, as shown in the Table 1. Among them, grasp#2 and handover#2 represent the classification results when the robot is "assistant." It can be seen that our network still has high accuracy.

4.3. Grasp detection metric

In order to better compare our results with the results of previous researchers, we refer to the comparison scale in Jiang et al. (2011) and make some optimizations. Since our grasp is aimed at a smaller task area, we set the iou value between ground truth grasp rectangle and the predicted grasp rectangle to two types: (1) The iou value is >25% for rough grasping. (2) The iou value is >50% for stable and accurate grasping. In addition, The offset between the grasp orientation of the predicted grasp rectangle and the ground truth rectangle is <30°.

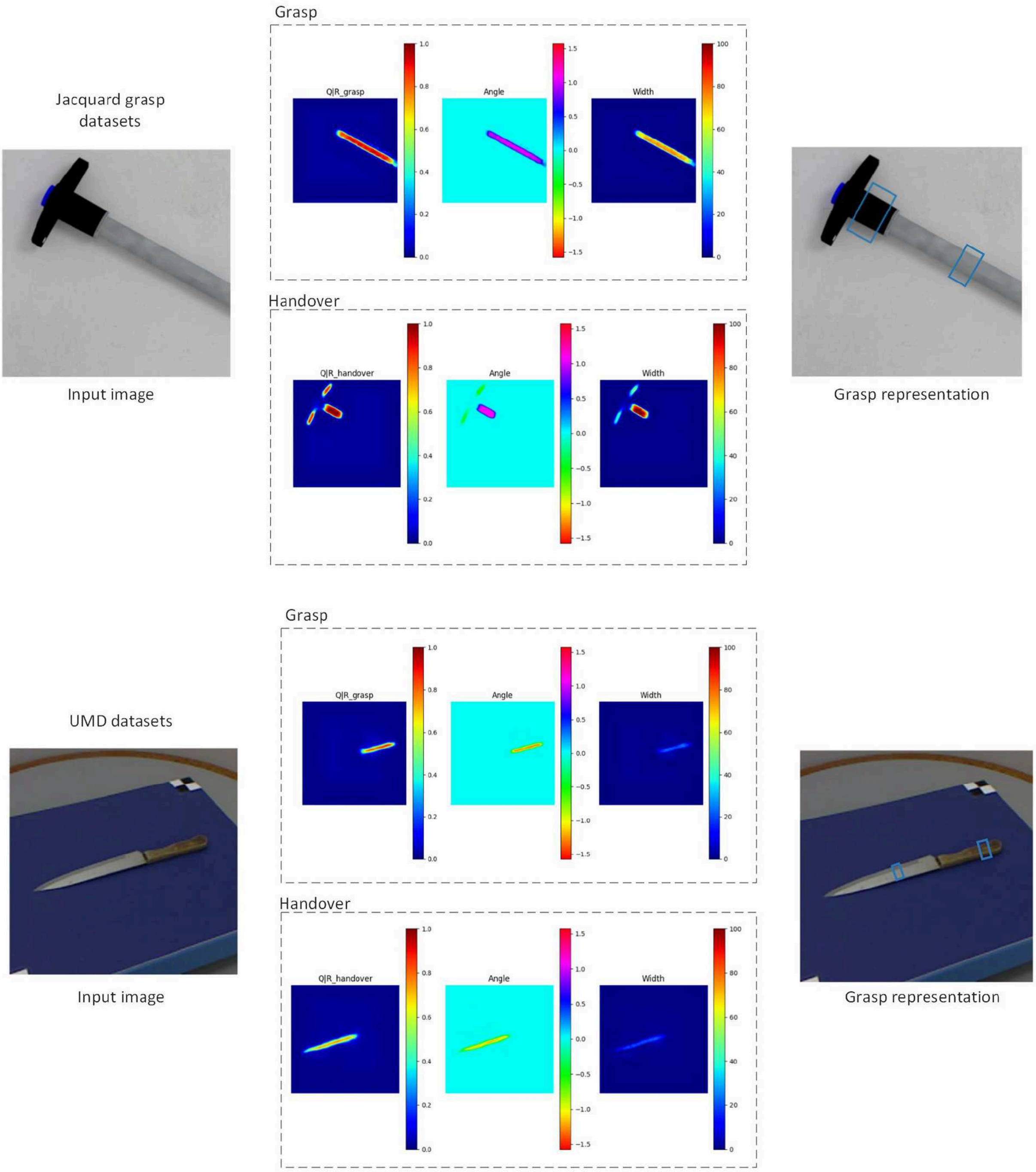


FIGURE 4
Qualitative results on different datasets.

4.4. Grasp detection

We discuss the results of our experiments here. We evaluate MGR-Net on our tools datasets, and demonstrate that our model is able to adapt to various types of tool objects. In addition, our method can not only grasp the whole object, but also understand the robot operation information contained in the task and grasp a certain area of the tool, so as to help people safely grasp the target tool. Figure 4 shows the qualitative results obtained on previously unseen tools.

The Table 2 shows the changes in the overall grasp due to the improvement of the network module. After obtaining the grasping representation of the tool through our detection network. Based on the robot platform, we use Sawyer robot to verify the grasping representation. Since the coordinate relationship between the camera and the robot is known, we transform the grasp representation from the image space to the robot coordinate system. Figure 5 shows the process of our verification through Sawyer robot, where Figures 5A, D are the result graphs generated by our capture of the detection network. After the camera space is converted to the robot space, Sawyer reaches the designated position and closes the gripper, as shown in Figures 5B, E. Figures 5C, F lift the object upward to prove whether our grasp is successful or not. We used 20 unseen real tools. Each test object contains five different positions and directions and the grasp accuracy is 92%. The experiment proves the effectiveness of our method.

TABLE 2 Ablation study.

Network structure	Accuracy (25%)	Accuracy (50%)
Residual block	0.95	0.83
Only block	0.95	0.84
GR-block	0.96	0.87

4.5. Comparison of different approaches

Considering that the traditional method does not involve the content of the state task area, we regard the entire object as an area with a grasp attribute, that is, the mask is the entire tool. We compared the accuracy of our network with the results of previous experiments on the Jacquard dataset (as shown in Table 3). It can be seen that the more accurate what needs to be captured, the more obvious the superiority of our method is. To further test the effectiveness of our grasping network, we tested it on a dataset of tools constructed by ourselves. Tool images are captured by a realsense camera. It is worth mentioning that our training set does not contain images from our homemade

TABLE 3 We compared our grasp network with other work.

References	Accuracy (25%)	Accuracy (50%)
Depierre et al. (2018)	0.74	–
Zhou et al. (2018)	0.92	–
Kumra et al. (2020)	0.94	0.72
Depierre et al. (2021)	0.86	–
Shukla et al. (2022)	0.90	0.69
Ours	0.95	0.77

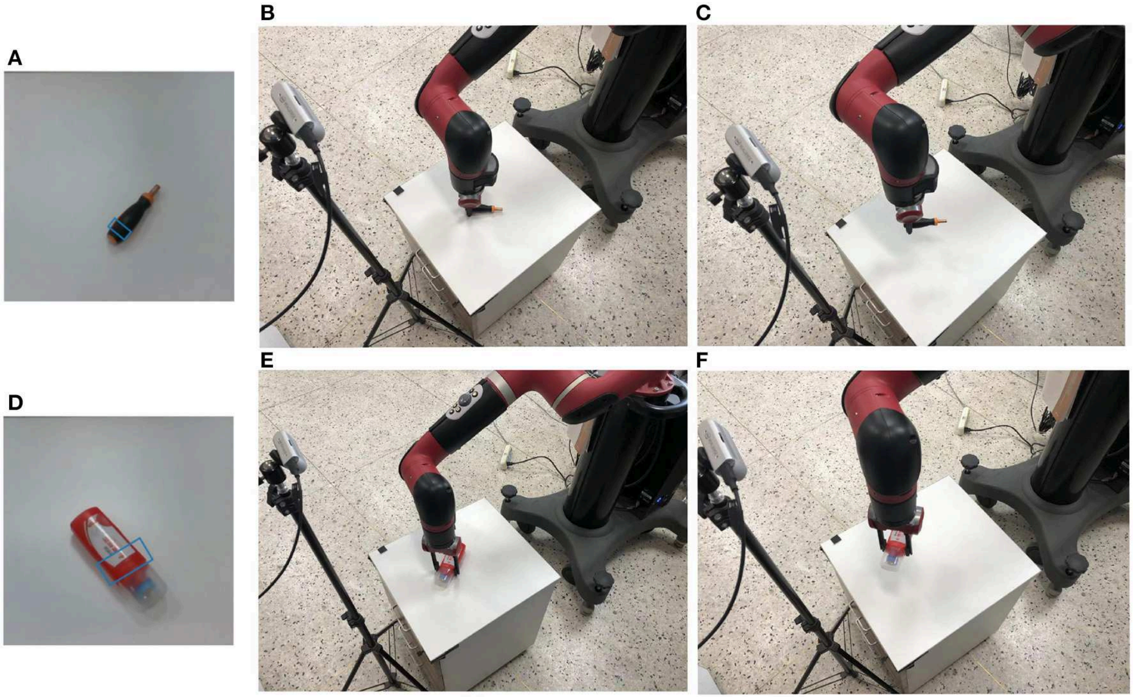


FIGURE 5
Verification through robot platform. **(A, D)** The results of grasp detection. **(B, E)** The robot grasping tools. **(C, F)** The robot lifting tools to indicate whether the grasping is successful or not.

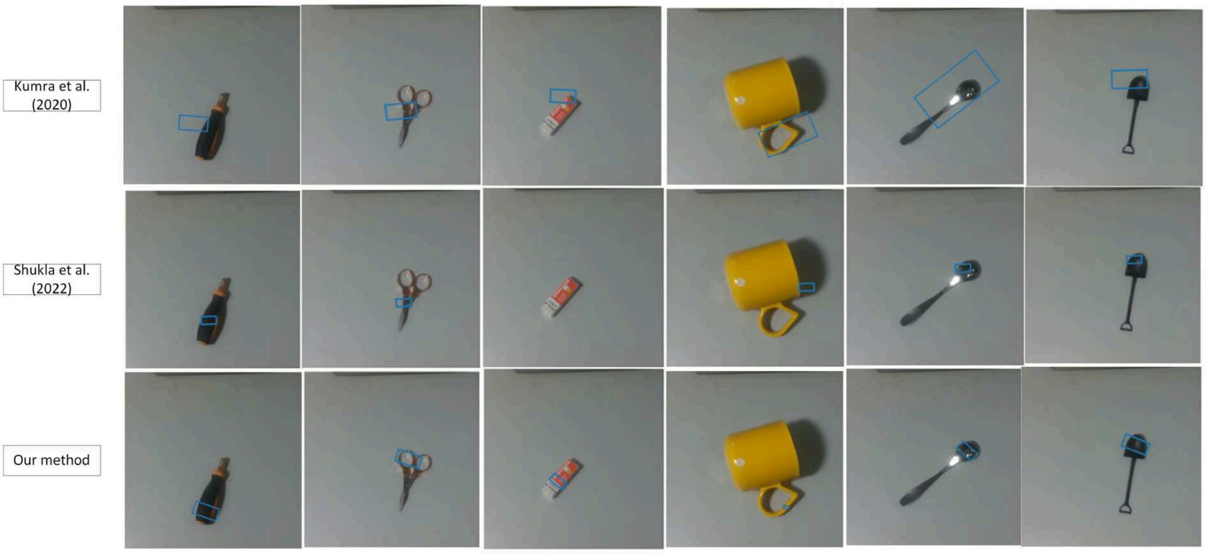


FIGURE 6
Untrained single tool images.

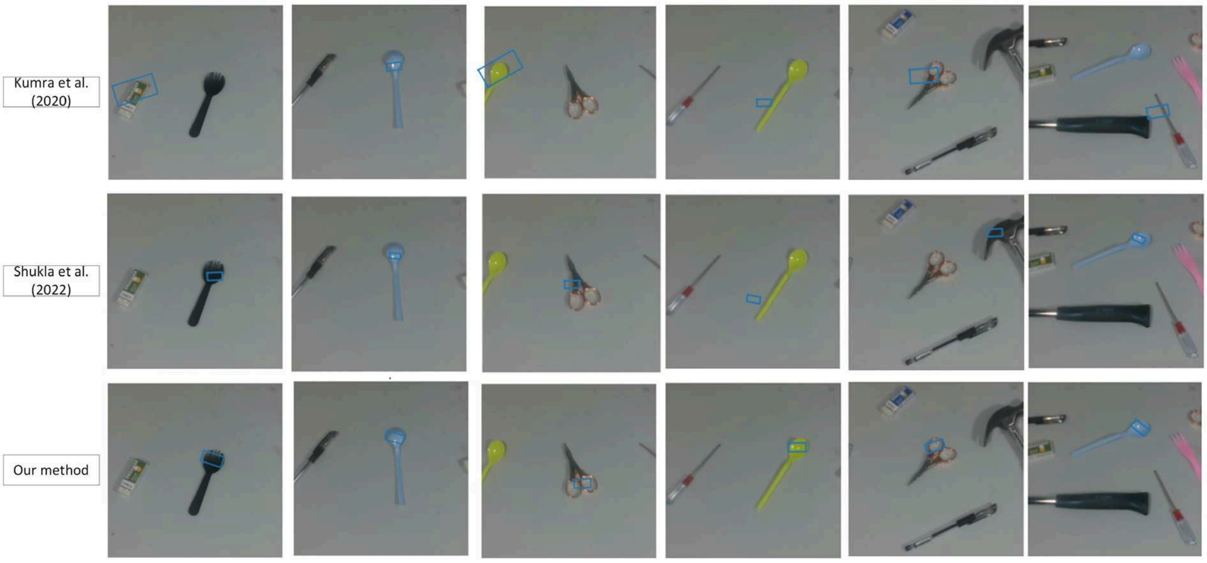


FIGURE 7
Untrained multi-tools images.

dataset. We have compared with Kumra et al. (2020) and Shukla et al. (2022), as shown in Figures 6, 7. It can be seen from the Figure 6 that in the untrained real images with uneven lighting, our method can more accurately find the grasp configuration of objects, and adopt a suitable size of the grasp box. For example, when grasping a cup, a small frame is generated at the handle of the cup to avoid the collision between the gripper and the rest of the cup. Figure 7 shows the strong anti-interference ability of our method and proves the necessity of generating object mask.

5. Conclusion

We presented a modular solution for tool usage issues in the context of human-robot interaction. A multi-layer instance segmentation network helps robots understand the regional attributes and semantics of objects under different states. Based on the state assigned to the robot based on the task, it is able to grasp or handover novel objects using our convolutional neural network MGR-Net that uses *n*-channel input data to generate images that can be used to infer grasp rectangles for each pixel in an image.

We validate our proposed system on our robotics platform. The results demonstrate that our system can perform accurate grasps for previously unseen objects with different state, even our method is able to adapt to changes in lighting conditions to a certain extent.

We hope to extend our solution to more complex object environments, such as where tools overlap and occlude each other. Besides, combining multiple visual angles to improve the success rate of grasping should also be considered in our later work.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/supplementary material.

Author contributions

TX proposed the method and designed experiments to verify the method, and then wrote this article. DZ and JY assisted in the experiment. YL reviewed and improved the manuscript. All authors contributed to the article and approved the submitted version.

Funding

This work was supported in part by National Natural Science Fund of China (Grant No. 61473155) and Primary Research & Development Plan of Jiangsu Province (Grant No. BE2017301).

Acknowledgments

Appreciations are given to the editors and reviewers of the journal.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Antanas, L., Moreno, P., Neumann, M., de Figueiredo, R. P., Kersting, K., Santos-Victor, J., et al. (2019). Semantic and geometric reasoning for robotic grasping: a probabilistic logic approach. *Auton. Robots* 43, 1393–1418. doi: 10.1007/s10514-018-9784-8

Brahmbhatt, S., Ham, C., Kemp, C. C., and Hays, J. (2019). "ContactDB: analyzing and predicting grasp contact via thermal imaging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8709–8719. doi: 10.1109/CVPR.2019.00891

Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., and Yan, Y. (2020). "Blendmask: top-down meets bottom-up for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8573–8581. doi: 10.1109/CVPR42600.2020.00860

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 834–848. doi: 10.1109/TPAMI.2017.2699184

Chu, F.-J., Xu, R., and Vela, P. A. (2019). Learning affordance segmentation for real-world robotic manipulation via synthetic images. *IEEE Robot. Autom. Lett.* 4, 1140–1147. doi: 10.1109/LRA.2019.2894439

Depierre, A., Dellandréa, E., and Chen, L. (2018). "Jacquard: a large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3511–3516. doi: 10.1109/IROS.2018.8593950

Depierre, A., Dellandréa, E., and Chen, L. (2021). "Scoring graspability based on grasp regression for better grasp prediction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 4370–4376. doi: 10.1109/ICRA48506.2021.9561198

Do, T.-T., Nguyen, A., and Reid, I. (2018). "AffordanceNet: an end-to-end deep learning approach for object affordance detection," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 5882–5889. doi: 10.1109/ICRA.2018.8460902

Dong, M., Bai, Y., Wei, S., and Yu, X. (2022). "Robotic grasp detection based on transformer," in *International Conference on Intelligent Robotics and Applications* (Springer), 437–448. doi: 10.1007/978-3-031-13841-6_40

Dong, M., Wei, S., Yu, X., and Yin, J. (2021). Mask-GD segmentation based robotic grasp detection. *Comput. Commun.* 178, 124–130. doi: 10.1016/j.comcom.2021.07.012

Fang, K., Zhu, Y., Garg, A., Kurenkov, A., Mehta, V., Fei-Fei, L., et al. (2020). Learning task-oriented grasping for tool manipulation from simulated self-supervision. *Int. J. Robot. Res.* 39, 202–216. doi: 10.1177/0278364919872545

Fischinger, D., Weiss, A., and Vincze, M. (2015). Learning grasps with topographic features. *Int. J. Robot. Res*. 34, 1167–1194. doi: 10.1177/0278364915577105

Fu, J., Levine, S., and Abbeel, P. (2016). "One-shot learning of manipulation skills with online dynamics adaptation and neural network priors," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4019–4026. doi: 10.1109/IROS.2016.7759592

Gao, N., Shan, Y., Wang, Y., Zhao, X., Yu, Y., Yang, M., et al. (2019). "SSAP: single-shot instance segmentation with affinity pyramid," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 642–651. doi: 10.1109/ICCV.2019.00073

Guo, D., Sun, F., Liu, H., Kong, T., Fang, B., and Xi, N. (2017). "A hybrid deep architecture for robotic grasp detection," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 1609–1614. doi: 10.1109/ICRA.2017.7989191

Hart, S., Dinh, P., and Hambuchen, K. (2015). "The affordance template ROS package for robot task programming," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 6227–6234. doi: 10.1109/ICRA.2015.7140073

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969. doi: 10.1109/ICCV.2017.322

Jiang, Y., Moseson, S., and Saxena, A. (2011). "Efficient grasping from RGBD images: learning using a new rectangle representation," in *2011 IEEE International Conference on Robotics and Automation*, 3304–3311. doi: 10.1109/ICRA.2011.5980145

Kroemer, O., Daniel, C., Neumann, G., Van Hoof, H., and Peters, J. (2015). "Towards learning hierarchical skills for multi-phase manipulation tasks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 1503–1510. doi: 10.1109/ICRA.2015.7139389

Krüger, N., Geib, C., Piater, J., Petrick, R., Steedman, M., Wörgötter, F., et al. (2011). Object-action complexes: Grounded abstractions of sensory-motor processes. *Robot. Auton. Syst*. 59, 740–757. doi: 10.1016/j.robot.2011.05.009

Kumra, S., Joshi, S., and Sahin, F. (2020). "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9626–9633. doi: 10.1109/IROS45743.2020.9340777

Kumra, S., and Kanan, C. (2017). "Robotic grasp detection using deep convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 769–776. doi: 10.1109/IROS.2017.8202237

Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *Int. J. Robot. Res*. 34, 705–724. doi: 10.1177/0278364914549607

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res*. 37, 421–436. doi: 10.1177/0278364917710318

Liu, S., Jia, J., Fidler, S., and Urtasun, R. (2017). "SGN: sequential grouping networks for instance segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 3496–3504. doi: 10.1109/ICCV.2017.378

Liu, W., Daruna, A., and Chernova, S. (2020). "Cage: context-aware grasping engine," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2550–2556. doi: 10.1109/ICRA40945.2020.9197289

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. (2022). "A convNet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11976–11986. doi: 10.1109/CVPR52688.2022.01167

Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., et al. (2017). Dex-Net 2.0: deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*. doi: 10.15607/RSS.2017.XIII.058

Mangin, O., Roncone, A., and Scassellati, B. (2017). How to be helpful? Implementing supportive behaviors for human-robot collaboration. *arXiv preprint arXiv:1710.11194*.

Morrison, D., Corke, P., and Leitner, J. (2020). Learning robust, real-time, reactive robotic grasping. *Int. J. Robot. Res*. 39, 183–201. doi: 10.1177/0278364919859066

Myers, A., Kanazawa, A., Fermuller, C., and Aloimonos, Y. (2014). "Affordance of object parts from geometric features," in *Workshop on Vision meets Cognition, CVPR, Vol. 9*. doi: 10.1109/ICRA.2015.7139369

Nguyen, A., Kanoulas, D., Caldwell, D. G., and Tsagarakis, N. G. (2017). "Object-based affordances detection with convolutional neural networks and dense conditional random fields," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5908–5915. doi: 10.1109/IROS.2017.8206484

Schmidt, P., Vahrenkamp, N., Wächter, M., and Asfour, T. (2018). "Grasping of unknown objects using deep convolutional neural networks based on depth images," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6831–6838. doi: 10.1109/ICRA.2018.8463204

Shukla, P., Pramanik, N., Mehta, D., and Nandi, G. (2022). Generative model based robotic grasp pose prediction with limited dataset. *Appl. Intell*. 52, 9952–9966. doi: 10.1007/s10489-021-03011-z

Ten Pas, A., and Platt, R. (2018). "Using geometry to detect grasp poses in 3D point clouds," in *Robotics Research* (Springer), 307–324. doi: 10.1007/978-3-319-51532-8_19

Tian, Z., Shen, C., Chen, H., and He, T. (2019). "FCOS: fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9627–9636. doi: 10.1109/ICCV.2019.00972

Wang, X., Kong, T., Shen, C., Jiang, Y., and Li, L. (2020a). "Solo: segmenting objects by locations," in *European Conference on Computer Vision* (Springer), 649–665. doi: 10.1007/978-3-030-58523-5_38

Wang, X., Zhang, R., Kong, T., Li, L., and Shen, C. (2020b). Solov2: dynamic and fast instance segmentation. *Adv. Neural Inform. Process. Syst*. 33, 17721–17732.

Zeng, A., Song, S., Yu, K.-T., Donlon, E., Hogan, F. R., Bauza, M., et al. (2018). "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 3750–3757. doi: 10.1109/ICRA.2018.8461044

Zhao, X., Cao, Y., and Kang, Y. (2020). Object affordance detection with relationship-aware network. *Neural Comput. Appl*. 32, 14321–14333. doi: 10.1007/s00521-019-04336-0

Zhou, X., Lan, X., Zhang, H., Tian, Z., Zhang, Y., and Zheng, N. (2018). "Fully convolutional grasp detection network with oriented anchor box," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7223–7230. doi: 10.1109/IROS.2018.8594116

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 12 January 2023
DOI 10.3389/fnbot.2022.1111621

OPEN ACCESS

EDITED BY
Praveen Kumar Donta,
Vienna University of
Technology, Austria

REVIEWED BY
Lidong Yang,
Inner Mongolia University of Science
and Technology, China
Shubham Mahajan,
Shri Mata Vaishno Devi University, India

*CORRESPONDENCE
Xiaoman Lu
✉ luxiaoman@stumail.neu.edu.cn

RECEIVED 29 November 2022
ACCEPTED 20 December 2022
PUBLISHED 12 January 2023

CITATION
Lu X, Lu R, Zhao W and Ma E (2023) Facial image inpainting for big data using an effective attention mechanism and a convolutional neural network. *Front. Neurorobot.* 16:1111621. doi: 10.3389/fnbot.2022.1111621

Facial image inpainting for big data using an effective attention mechanism and a convolutional neural network

Xiaoman Lu*, Ran Lu, Wenhao Zhao and Erbin Ma

Department of Mathematics, College of Science, Northeastern University, Shenyang, Liaoning, China

Big data facial image is an important identity information for people. However, facial image inpainting using existing deep learning methods has some problems such as insufficient feature mining and incomplete semantic expression, leading to output image artifacts or fuzzy textures. Therefore, it is of practical significance to study how to effectively restore an incomplete facial image. In this study, we proposed a facial image inpainting method using a multistage generative adversarial network (GAN) and the global attention mechanism (GAM). For the overall network structure, we used the GAN as the main body, then we established skip connections to optimize the network structure, and used the encoder–decoder structure to better capture the semantic information of the missing part of a facial image. A local refinement network has been proposed to enhance the local restoration effect and to weaken the influence of unsatisfactory results. Moreover, GAM is added to the network to magnify the interactive features of the global dimension while reducing information dispersion, which is more suitable for restoring human facial information. Comparative experiments on CelebA and CelebA-HQ big datasets show that the proposed method generates realistic inpainting results in both regular and irregular masks and achieves peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), as well as other evaluation indicators that illustrate the performance and efficiency of the proposed model.

KEYWORDS

big data artificial intelligence (AI), deep learning algorithm, deep learning-based facial image inpainting, generative adversarial network, convolutional neural networks

1. Introduction

With the rapid development of computer vision technology, digital images (Singh and Goel, 2020) have become the mainstream of facial image acquisition. Normally, people usually rely on electronic devices to obtain facial images; however, watermark occlusion, smear, part of the area missing, and other problems often appear in the transmission process of digital images (Baeza et al., 2009), preservation (Meyers and Scott, 1994), and post-processing (Shen and Kuo, 1997), damaging the quality of the facial image and resulting in poor visual feeling (Parmar, 2011). To solve the abovementioned problems, related scholars began to study these kinds of problems and proposed a series of novel inpainting approaches.

Image inpainting is a very challenging task in image processing (Elharrouss et al., 2020), and its purpose is to restore and complete the missing or defaced image part. A new image needs to be inferred and constructed according to the contextual information of the damaged image and the overall image structure (Jin et al., 2021). The restored image should have clear textures and natural boundary pixels and conform to human visual perception. Compared to other image inpainting tasks, some similar image blocks cannot be found in other facial image areas (Yang et al., 2020). For example, it is difficult to infer a reasonable nose image based on the surrounding areas when the nose part is missing, which may lead to an imbalance proportion of facial images. For this problem, it is necessary to reconstruct images that satisfy human visual perception according to a large amount of prior information and contextual semantics (Yeh et al., 2017).

Before deep learning methods were proposed, there were two kinds of theoretical research in image inpainting, including partial differential equations- and texture-based methods. Bertalmio et al. (2000) used partial differential equations to diffuse neighborhood pixel information to the missing area using an isograd direction field. For images with small missing areas, satisfactory results can be achieved. However, it is not ideal for images with large missing areas because this method does not consider the semantic information of the image context. Efros and Leung (1999) first proposed the generation of patch blocks with similar textures using the extracted texture information of the missing regions and then the use of the generated patch blocks to fill in the missing regions. The disadvantage of this method is that, although the missing area is filled, the filled area is compact overall from the content level but not from the pixel level. In other words, the repair result is not smooth enough, with many traces of artificial processing. Criminisi et al. (2004) established a block image restoration method based on texture synthesis. In this method, a pixel randomly selected on the image's missing area boundary is taken as the center to choose a certain size image texture block, which is then used to repair the missing area. This image inpainting method can fill in more appropriate texture information for the missing areas, but because the contextual semantic information of an image is ignored and contextual semantics of the repaired image becomes incoherent, the complex facial image inpainting task cannot be completed.

Deep-learning-based facial image inpainting technology (Qin et al., 2021) is more suitable for a variety of restoration scenarios than traditional image restoration methods. The feature distribution dataset learned by a neural network is more suitable for facial image restoration with a large missing area and random damage. Not only are the texture details accurate but also are the contours harmonized, and the facial image conforms to the contextual semantics (Wei et al., 2019). After ongoing in-depth research by relevant scholars, deep learning-based image repair methods have produced a number of results.

Pathak et al. (2016) used a context encoder to complete an image repair task, which was the first image inpainting method based on a generative adversarial network (GAN). The generator is divided into an encoder and a decoder (Sun et al., 2018). The encoder is responsible for compressing and extracting feature information from an incomplete image, and the decoder is responsible for restoring an input-compressed feature to the image. In this method, the context encoder can achieve a good repair effect, but the generation antagonism losses adopted by the context encoder considers only local information of an incomplete region and not the overall semantic coherence of the image. Iizuka et al. (2017) adopted a global–local double discriminator to improve the context encoder. A local discriminator was applied to the repair result of an incomplete area, and a global discriminator was applied to the overall repair result. This design ensured not only the accuracy of the repair area but also the integrity of the final result. However, the prediction results of this method are still inaccurate when the large area facial image is missing. Yang et al. (2017) proposed the use of content and texture generation networks to complete the image repair task. The content generation network is responsible for inferring the semantics and global structure of an image, while the texture generation network is responsible for generating high-frequency details of an image. Compared to previous methods, this method solves the problem of high-resolution image repair. Yan et al. (2018) added a shift-connection layer on the basis of the U-Net network. In this method, pixel information from known regions is transferred to the corresponding missing regions to assist the image repair generated in the process of guided loss minimization, which encodes and decodes the distance between the distribution and the true distribution. However, due to the shortcomings of a simple structure of the algorithm, it is not effective in restoring facial images, which have problems such as blurred edges.

Although GANs are widely used in the field of image inpainting, they still rely too much on the self-generation ability of generative networks and have many problems to solve. For example, when the texture structure of a facial image is more complex, it is easy to appear fuzzy, semantic incoherence and other phenomena. When the local feature of the facial image is not clear, the information stored in the model is too large and network training is prone to information overload.

To solve these problems, based on the normalization of the feature layer output in the GAN and the guiding role of the attention mechanism in image detail inpainting, this study proposes a facial image inpainting method using a multistage GAN based on a global attention mechanism (GAM) named CLGN, where a generative network can accelerate the training speed and improve training stability through feature layer output normalization. By using step coiling instead of up-sampling and full-connection layers, convolution can play a good role in extracting image features. Meanwhile, GAM (Liu et al., 2021) was introduced to enhance the guiding role of

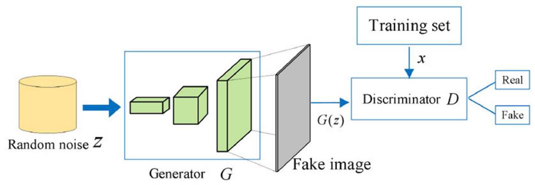


FIGURE 1
The model of generative adversarial networks (GANs).

important features during the image inpainting process. In addition, a U-Net skip-connection (Ronneberger et al., 2015) was introduced between the encoder and the decoder to reduce information loss due to down-sampling and to optimize texture consistency. The loss function is used as an important factor to measure the generated image quality and loss (Gao and Fang, 2011), weighted reconstruction loss, perceptual loss, style loss, and total variation (TV) loss, which were combined to optimize the total loss of the generated network for model training.

Our study provides the following contributions:

- Building a multistage (crude-local-global) generative network CLGN to capture feature information from receptive fields of different sizes and enhance presentation capabilities.
- Adding GAM to magnify the interactive features of the global dimension while reducing information dispersion, which is more suitable for restoring human facial information.
- The proposed CLGN produces photorealistic and plausible inpainting results on two datasets, CelebA and CelebA-HQ. The remainder of this paper is organized as follows: Section 2 introduces the relevant theories used in our proposed method. Section 3 shows the observation and motivation, the network architecture, and loss functions. Section 4 focuses on comparative and ablation experiments of our methods. Section 5 concludes and discusses future research.

2. Related theory

2.1. Generative adversarial networks

A generative adversarial network was proposed by Goodfellow et al. (2014). In recent years, GANs have been extensively studied in combination with other machine learning algorithms in some specific applications, such as semi-supervised learning (Odena, 2016), transfer learning (Cho et al., 2017), and reinforcement learning (Wang et al., 2020), and are widely used in image inpainting. GAN has made a considerable breakthrough in image inpainting by producing realistic images. The core idea of GAN comes the "two-player zero-sum game" in game theory (Ge et al., 2018), in which networks are optimized by cheating each other between generators and discriminators, resulting in a Nash equilibrium. The GAN consists of a generative network G and a discriminant network D, and its structure is shown in Figure 1.

By learning the probability distribution mapping P_{data} of the real data, the generative network G is expected to output content $G(z)$ close to the real data. The discriminant network D needs to identify the source of the input data as much as possible, i.e., classify x and $G(z)$. When the discriminant network D cannot distinguish data sources, network performance is optimal. Its objective function is as follows:

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}}\left[\log(D(x))\right] + E_{z\sim P_g}\left[\log(1-D(G(z)))\right],$$

where G represents a generative network, D represents a discriminant network, $E(\bullet)$ represents the mathematical expectation, V represents the objective function, x represents the sample, z represents random noise, and P_{data} represents the distribution of the real sample.

2.2. Visual geometry group network

The visual geometry group network (VGGNet) was proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group at the University of Oxford (Simonyan and Zisserman, 2015). An outstanding contribution of VGGNet is to demonstrate that small convolutions can effectively improve performance by increasing network depth. VGG expertly inherits the mantle of Alexnet while also exhibiting the characteristics of a deeper network layer.

The structure of VGGNet is shown in Figure 2 (Noh et al., 2015) and consists of five convolutional layers, three fully connected layers, and softmax output layers. These layers are separated by max-pooling (maximization pool), and the activation units of all hidden layers adopt the ReLU function. VGG uses multiple convolution layers with smaller convolution kernels (3 × 3) to replace one convolution layer with a larger convolution kernel. On the one hand, parameters can be reduced. On the other hand, it is equivalent to perform more non-linear mapping, which can increase the network's ability to fit and express.

2.3. Global attention mechanism

In recent years, attention mechanisms have been widely used in many applications (Zn et al., 2021). The convolutional block attention module (CBAM) (Woo et al., 2018) sequentially places

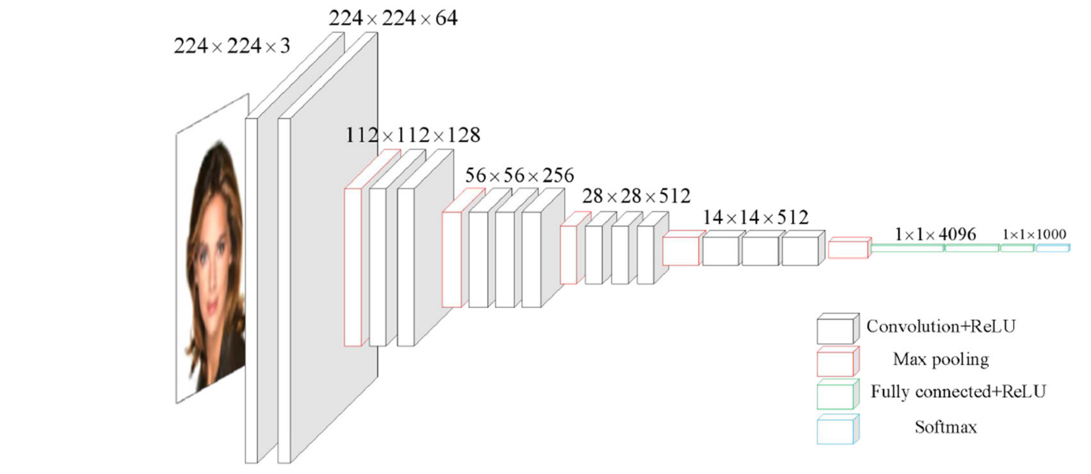


FIGURE 2
The structure of a VGG16 module. The face images are adapted from the celeba-HQ dataset, which comes from Karras et al. (2017).

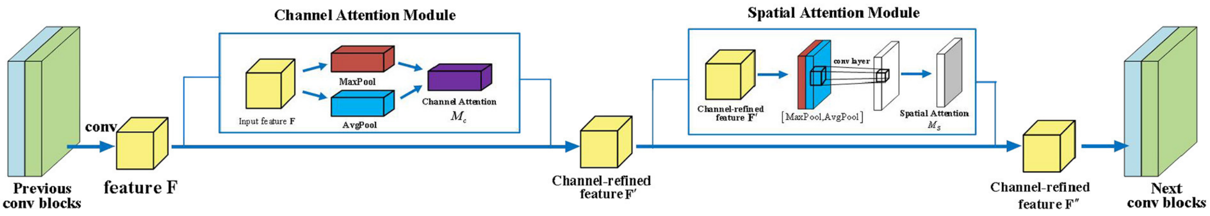


FIGURE 3
The model of convolutional block attention module (CBAM).

the channel and spatial attention operation, while the bottleneck attention module (BAM) (Park et al., 2018) does it in parallel. However, both of them ignore channel-spatial interactions and consequently lose cross-dimensional information.

Therefore, GAM that boosts network performance by keeping the amount of information to a minimum and zooming in on the global interaction representation has been proposed. GAM (Liu et al., 2021) is a simple yet effective attention module that reserves information to magnify the "global" cross-dimensional interactions. The GAM adopts the sequential channel-spatial attention mechanism from CBAM (Woo et al., 2018), which is an elementary yet practical attention module for feed-forward convolutional neural networks. CBAM can be regarded as a dynamic selection process for inputting important information into an image, which significantly improves the performance level of many computer vision tasks and plays an important part in image inpainting with complex image structures.

The internal structure of CBAM is shown in Figure 3. We set the intermediate feature map $F \in \mathbb{R}^{C\times H\times W}$ as input. CBAM deduces an attention map in two separate dimensions, channel and space, which are shown as a one-dimensional (1D) channel attention map $M_c \in \mathbb{R}^{C\times 1\times 1}$ and a two-dimensional (2D) spatial attention map $M_s \in \mathbb{R}^{1\times H\times W}$. In conclusion, the general process of the attention module can be represented as:

$$\begin{cases} \mathbf{F}' = \mathrm{M_c}(\mathbf{F}) \otimes \mathbf{F}, \\ \mathbf{F}'' = \mathrm{M}_s(\mathbf{F}') \otimes \mathbf{F}', \end{cases}$$

where ⊗ indicates an element-wise multiplication and F'' is named as the final refined output. The detailed operations for each module are described as follows.

For the channel attention module, first, we applied both average pool and max pool operations to gather spatial information, producing two disparate spatial context descriptors: $\mathbf{F}^c_{avg}$ and $\mathbf{F}^c_{max}$ representing the max-pooling features. Then, we sent two descriptors to a shared network to produce a channel attention map $M_c \in \mathbb{R}^{C\times 1\times 1}$.

To cut down parameter overhead, we set the hidden activation size $\mathbb{R}^{C/r\times 1\times 1}$, where r is the reduction ratio.

Finally, the output feature vectors are conflated by element-wise summation after we applied the shared network to each descriptor. In conclusion, the channel attention is represented by:

$$\begin{aligned} \mathrm{M}_c(\mathbf{F}) &= \sigma(MLP(AvgPool(\mathbf{F})) + MLP(MaxPool(\mathbf{F}))) \\ &= \sigma(\mathbf{W}_1(\mathbf{W}_0(\mathbf{F}^c_{avg})) + \mathbf{W}_1(\mathbf{W}_0(\mathbf{F}^c_{\max}))), \end{aligned}$$

where σ indicates the sigmoid function, $W_0 \in \mathbb{R}^{C/r \times C}$, and $W_1 \in \mathbb{R}^{C \times C/r}$.

For the spatial attention module, firstly, we introduce the average and maximum pools on the canal axis and connect them to establish an adequate feature description so that spatial attention can be calculated. In cascaded feature descriptors, we used the convolution layers to create a spatial attention pattern $M_s\,(F) \in R^{H \times W}$ in which positions of emphasis or suppression can be encoded. In particular, we created two 2D maps: $F^s_{avg} \in \mathbb{R}^{1 \times H \times W}$ and $F^s_{max} \in \mathbb{R}^{1 \times H \times W}$, which reflect the average characteristics of swimming pools in the canal and their maximum characteristics. Then, a 2D spatial attention table is output after being connected to the standard convolution layer and convoluted at its end. Spatial attention is calculated as follows:

$$\begin{aligned} \mathrm{M}_s\,(F) &= \sigma(f^{7 \times 7}([AvgPool(\mathbf{F}); MaxPool(\mathbf{F})])) \\ &= \sigma(f^{7 \times 7}([\mathbf{F}^s_{avg}; \mathbf{F}^s_{\max}])), \end{aligned}$$

where σ represents the sigmoid function and $f^{7 \times 7}$ denotes a convolution operation with the filter size of 7 × 7.

3. Proposed method

3.1. Observation and motivation

Traditional image inpainting methods are based on texture extension (Bertalmio et al., 2000) or similar block matching (Criminisi et al., 2004). These methods do not repair some damaged images with large missing areas and complex structures of missing areas. Especially, in facial image inpainting, the big challenge is how to ensure the overall consistency of the inpainting results and restore the missing details and textural features.

In this study, we put forward a facial image inpainting method using an attention-based multistage GAN followed by a crude-local-global framework. Considering that missing areas of different sizes can be solved, the proposed network contains a three-stage network for image inpainting to combine the networks with different receptive fields. The network structure and the corresponding loss functions are described in Section 3.2.

3.2. Network architecture

3.2.1. Crude inpainting network

Our crude inpainting network Net_C employs an encoder–decoder framework with the addition of a skip connection, consisting of eight down- and up-sampling operations. We used long skip connections to transmit information from the encoder to the decoder to restore information lost during down-sampling. The receptive field resolution is 766×766 and is nearly three times larger than the input image resolution with a size of 256 × 256.

For a convolutional neural network, a large receptive field is helpful to the whole image inpainting. At the input end, the network receives an input image I_{in} and a binary mask M, which describe the missing areas. Note that the missing pixel is equal to 1 and the valid pixel is equal to 0. Meanwhile, at the output end, the network exports an inpainting image I^C_{out}.

To weaken the blur effect and improve the restoration effect of inpainting images, a patch-based discriminator with spectral normalization was also applied. The inputs for the discriminator were a ground truth image and the inpainting image I^C_{out}, while the output was a 2D feature map where the shape is $\mathbb{R}^{32 \times 32}$. The function of the discriminator is to determine whether each element in the feature map is true or false.

3.2.2. Local development network

To further optimize the local refinement, we designed a surface-deep network called the local refinement network Net_L, which includes two down-sampling operations, four residual blocks, and two up-sampling operation, as shown in the middle row of Figure 4.

Due to its surface nature, this network has a small receptive field with the size of 109 × 109 for each output neuron. The local area of the above-mentioned rough inpainting results was then processed in a sliding window manner. Because of this design, some missing areas, such as the local structures and the textures, can be properly repaired by the surrounding local image information. Moreover, this process is not affected by the long distances and content not being filled. In addition, more residual blocks are introduced into this network, which can gradually make the receptive field larger and significantly reduce the model generalization error.

3.2.3. Global attention-based network

After the local refinement network process, some unresolved visual artifacts are properly removed with the help of surrounding local areas. Nevertheless, some missing areas (e.g., facial features such as the eyes or the mouth that are easily mismatched) still need to be better refined when capturing information from the corresponding large surrounding areas. In view of this fact, a global attention-based network is established,

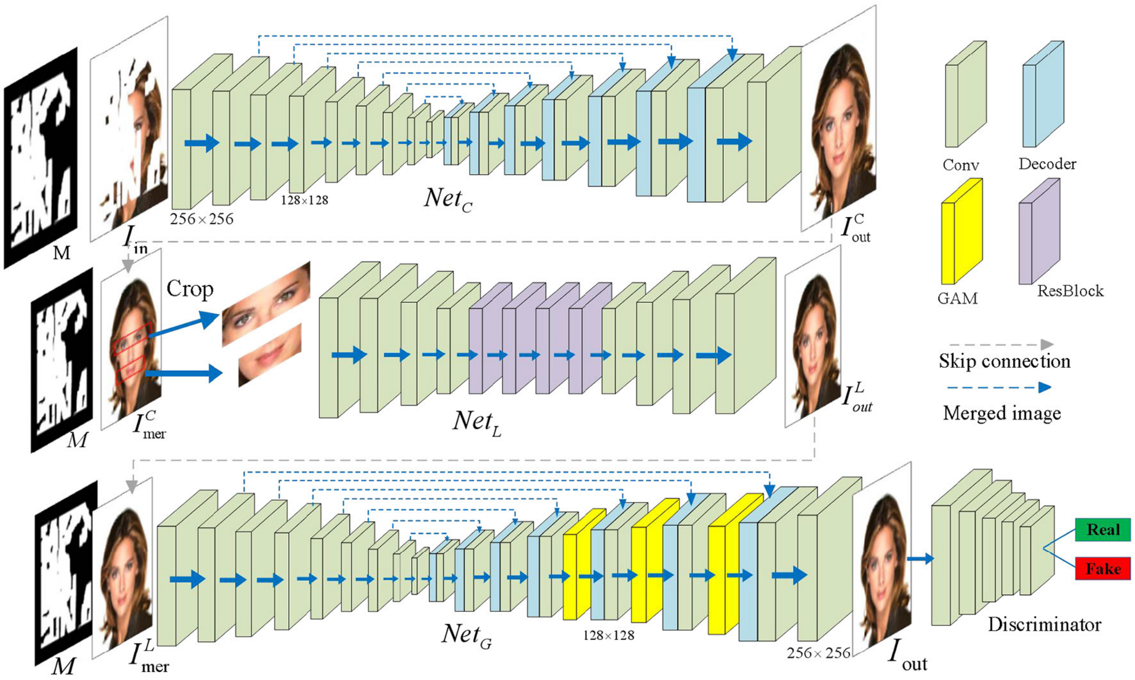


FIGURE 4
The network architecture of our proposed method CLGN. The purple block in the local development network indicates a two-layer residual block (He et al., 2016). The three yellow blocks represent the GAM attention modules with resolutions of 16 × 16, 32 × 32, and 64 × 64, respectively. The green blocks represent the convolutional layer and the blue blocks represent the decoder. The face images are adapted from the celeba-HQ dataset, which comes from Karras et al. (2017).

which can expand the scope of access to information for a neuron in two ways, i.e., the attention mechanism and a large receptive field.

Considering that a crude inpainting network has enough receptive fields to cover the whole image area, we exploited the basic structure of GAM. Based on this, three CBAMs are added in front of the decoder, aiming to attain a global attention-based network Net_G (see the three yellow blocks in the third row of Figure 4). Moreover, considering that the local development network can already provide relatively correct image restoration results, there is a major trend for a novel network Net_G based on the attention mechanism to become more stable and robust. Some existing studies (Yu et al., 2018, 2019) used the attention mechanisms to calculate the correlation between contextual information and the missing areas. In this study, a lightweight and powerful GAM attention module along two separate dimensions (i.e., channel and spatial) was used. A feature map $F \in \mathbb{R}^{C\times HW}$is given, and the affinity $s_{i,j} \in \mathbb{R}^{HW\times HW}$ of F_i and F_j is computed by:

$$s_{i,j} = \frac{exp(\widehat{s}_{i,j})}{\sum_k exp(\widehat{s}_{i,j})}, \widehat{s}_{i,j} =< \frac{F_i}{||F_i||}, \frac{F_j}{||F_j||} > .$$

Note that the weighted average version F is $\widetilde{F} = F * S \in \mathbb{R}^{C\times HW}$ in terms of matrix multiplication.

In the end, we connected F and $\widetilde{F}$. Then, we introduced a 1 × 1 convolutional layer to maintain the number of inchoative channels F.

3.3. Loss functions

3.3.1. Reconstruction loss

In terms of pixel-level supervision, we used weighted l_1 loss as the reconstruction loss to measure the distance between the ground truth I_{gt} and the generated image I_{out}, let:

$$\mathcal{L}_{valid}{}^C = \frac{1}{sum(1_A - M)}||(I^C_{out} - I_{gt}) \odot (1_A - M)||_1,$$
$$\mathcal{L}_{hole}{}^C = \frac{1}{sum(M)}||(I^C_{out} - I_{gt}) \odot M||_1,$$

where 1_A means the indicator function, $I_{gt}$is the ground truth image, $\odot$ is the element-wise product operation, and $sum(M)$ is the number of non-zero elements in M. Then, the pixel-wise reconstruction loss is formulated as:

$$\mathcal{L}^C_r = \mathcal{L}^C_{valid} + \lambda_h \cdot \mathcal{L}^C_{hole}.$$

In addition, the first training target of the local refinement network (Net_L) is the weighted reconstruction loss $\mathcal{L}^L_r$, which is the same as Equation (7) except for replacing I_{out}^{c} with I_{out}^{L} in Equation (6).

3.3.2. Adversarial loss

In this study, we used the least square loss function for GAN loss. Least square loss (Mao et al., 2017) not only enhances stability during the training process but also develops generator performance with the aid of more gradients. Then, we define the corresponding loss functions for the crude inpainting network and discriminator as:

$$\begin{aligned} \mathrm{I}_{\mathrm{mer}}^{\mathrm{C}} &= \mathrm{I}_{in} \odot (1_A - \mathrm{M}) + \mathrm{I}_{out}^{\mathrm{C}} \odot \mathrm{M}, \\ \mathrm{L}_{\mathrm{G}}^{\mathrm{C}} &= E_{\mathrm{Imer}} \sim p_{\mathrm{I}mer}(\mathrm{Imer}) \left[(D(I_{mer}^{C}) - 1)^2 \right], \\ \mathcal{L}_D &= \frac{1}{2} E_{\mathrm{I}-pdata(I)} [(D(I_{gt}) - 1)^2] \\ &\quad + \frac{1}{2} E_{\mathrm{I}_{mer} \sim p_{\mathrm{I}_{mer}}(\mathrm{I}_{mer})} [(D(I_{mer}^{c}))^2], \end{aligned}$$

where 1_A represents the indicator function, E means mathematical expectation, $I_{\mathrm{mer}}^{\mathrm{C}}$ is the merged image, and I_{gt} is the ground truth image.

3.3.3. Total variation loss

In signal processing, TV denoising is a noise-removal process (Liu et al., 2018). It is based on the principle that signals with excessive and possibly spurious detail have high TV, that is, the integral of the absolute image gradient is high. Following Liu et al. (2018), we used TV loss as a smoothing penalty. The formula is as follows:

$$\begin{aligned} \mathcal{L}_{\mathrm{tv}}^{\mathrm{L}} &= ||I_{\mathrm{mer}}^{\mathrm{L}}(i, j+1) - I_{\mathrm{mer}}^{\mathrm{L}}(i, j)||_1 \\ &\quad + ||I_{\mathrm{mer}}^{\mathrm{L}}(i+1, j) - I_{\mathrm{mer}}^{\mathrm{L}}(i, j)||_1. \end{aligned}$$

where the calculation process is precisely the same as that of $I_{\mathrm{mer}}^{\mathrm{C}}$, i.e., Equation (8).

3.3.4. Perceptual loss

To better renovate the structural and textual information, we apply the perceptual loss (Johnson et al., 2016) based on VGG-16 (Simonyan and Zisserman, 2015), which is trained in ImageNet beforehand. Unlike the pixel-level reconstruction loss and TV loss mentioned above, which are done in pixel space, the perceptual loss is calculated in feature space. Furthermore, perceptual loss is shown by:

$$\mathcal{L}_{per}^{\mathrm{L}} = \sum_i \begin{array}{l} ||\mathcal{F}_i(I_{\mathrm{out}}^{\mathrm{C}}) - \mathcal{F}_i(I_{\mathrm{gt}})||_1 \\ +||\mathcal{F}_i(I_{\mathrm{mer}}^{\mathrm{L}}) - \mathcal{F}_i(I_{\mathrm{gt}})||_1 \end{array},$$

where is the feature map of the ith layer in the VGG-16 network (Simonyan and Zisserman, 2015), which is pretrained, $i \in \{5, 10, 17\}$.

3.3.5. Style loss

Style loss represents the difference in the Gram matrix between the features of the synthesized image and the features of the style image, ensuring that the style of the generated image matches the style image. Here, we define style loss as follows:

$$\mathcal{L}_{sty}^{\mathrm{L}} = \sum_i \begin{array}{l} ||\mathcal{G}_i(I_{\mathrm{out}}^{\mathrm{L}}) - \mathcal{G}_{ii}(I_{\mathrm{gt}})||_1 \\ +||\mathcal{G}_i(I_{\mathrm{mer}}^{\mathrm{L}}) - \mathcal{G}_{ii}(I_{\mathrm{gt}})||_1 \end{array},$$

where $\mathcal{G}_i(\cdot) = \mathcal{F}_i(\cdot)\mathcal{F}_i(\cdot)^T$ is the Gram matrix.

3.3.6. Style loss

For a crude inpainting network, we summarized the total loss of Net_C:$\mathcal{L}_C = \mathcal{L}_{valid}^{C} + \lambda_h.\mathcal{L}_{hole}^{C} + \lambda_g \cdot \mathcal{L}_G^{C}$. It should be noted that we set $\lambda_h = 6$ and $\lambda_g = 0.1$ in all experiments.

For the local development network, the target for the local refinement network Net_L is defined as:

$$\begin{aligned} \mathcal{L}_L &= \mathcal{L}_{valid}^{L} + \lambda_h \cdot \mathcal{L}_{hole}^{L} + \lambda_{TV} \cdot \mathcal{L}_{TV}^{L} + \lambda_{per} \cdot \mathcal{L}_{per}^{L} \\ &\quad + \lambda_{sty} \cdot \mathcal{L}_{sty}^{L} \end{aligned}$$

In our experiments, we discovered that weight losses in Liu et al. (2018) were correspondingly balanced in the order of magnitude, so the weight setting was adopted. We set $\lambda_h = 6$,$\lambda_{tv} = 0.1$,$\lambda_{per} = 0.05$, and $\lambda_{sty} = 120$ in a special way.

For a GAM attention-based global refinement network, we found that the training target $\mathcal{L}_G$ of Net_G is almost consistent with $\mathcal{L}_L$ of Net_L, and we only need to replace I_{out}^{L} with I_{out}^{G} in the corresponding positions of $\mathcal{L}_L$.

In this connection, the novel inpainting network CLGN is trained using an "end-to-end" method, and the overall CLGN output becomes the final image inpainting result. The sum total of three subnetworks and a discriminator is the final training loss, i.e., $\mathcal{L}_C + \mathcal{L}_L + \mathcal{L}_G + \mathcal{L}_D$.

4. Experiments

4.1. Experimental settings

4.1.1. Experimental platform and parameters

For network training, the hardware platform is an AMD EPYC 7302 16-Core Processor CPU, a single GeForce RTX 3090 (31GB), and the software platform is PyTorch1.3.0. During training, each image and mask were resized to 256 × 256 by bicubic interpolation, and there are no data arguments. The

Adam optimizer is used with an initial learning rate of 0.0002 for the first 100 epochs and later decays the learning rate to 0 for the next 100 epochs to fine-tune the model. In addition, the first-order momentum was set as $\beta_1 = 0.5$ and the second-order momentum was set as $\beta_2 = 0.999$.

4.1.2. Data sets

The proposed method is evaluated on two datasets of CelebA (Liu et al., 2015) and CelebA-HQ (Karras et al., 2017). The CelebA (Liu et al., 2015) face dataset is an open dataset from the Chinese University of Hong Kong, which contains 202,599 facial images of 10,177 celebrity identities, and all of them are well-labeled. It is a very useful dataset for face-related training. We randomly selected 40,000 of these faces for our experiment. The 40,000 images are divided into a training set of 36,000 images and a test set of 4,000 images. The CelebA-HQ (Karras et al., 2017) dataset is a high-quality version of CelebA. It is a celebrity face attribute dataset containing 30,000 face images. We randomly select 27,000 images as the training sample and 3,000 images as the testing sample.

To train our network, we used irregular masks based on the quick draw irregular mask data set (QD-IMD) (Iskakov, 2021). Moreover, when testing the network, the irregular mask data provided by Liu et al. (2018) was used to assess our training result. Note that the irregular mask set includes 12,000 masks, which were divided into six categories with different coverage rates, i.e., $(0.01, 0.1]$,$(0.1, 0.2]$,$(0.2, 0.3]$,$(0.3, 0.4]$,$(0.4, 0.5]$$(0.5, 0.6]$.

4.2. Performance comparison

To show the inpainting performance of the proposed method, we first introduced our evaluation indicators and then compared quantitative measurements, visual comparisons, and subjective evaluations separately.

The following six mainstream image inpainting methods are used to compare with the proposed network: CA (Yu et al., 2018): A model trained in two stages of coarse and fine precision, which used a contextual attention mechanism in a fine precision network in the form of two codecs in series. GMCNN (Wang et al., 2018): A generative multicolumn neural network architecture in the form of three codecs in parallel. MEDFE (Liu et al., 2020): A mutual encoder–decoder CNN with feature equalizations for joint recovery of architecture and texture. RFR (Li et al., 2020): An advanced image inpainting method in feature space with recurrent feature reasoning and knowledge-continued attention. MADF (Zhu et al., 2021): A U-shaped framework with mask-aware dynamic filtering for image inpainting with a point-wise normalization. LG-net (Quan et al., 2022): A multilayer network architecture for image inpainting to combine networks with different receptive fields, considering the complexity of missing regions.

4.2.1. Evaluation methods

To objectively evaluate the inpainting performance of different inpainting methods, the following objective indicators are used to evaluate the inpainting quality under the same experimental conditions:

l_1 loss function (Gao and Fang, 2011): By calculating the sum of the absolute difference between the inpainting image and the original image, the similarity between the two images at the pixel level can be evaluated.

$$l_1 = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y_i}\right|.$$

Peak signal-to-noise ratio (PSNR) (Hore and Ziou, 2010): It is defined by the maximum possible pixel value Z and mean square error (MSE) between images.

$$\begin{cases} MSE &= \frac{1}{n}\sum\limits_{i=1}^{n}\left(y_i - \widehat{y_i}\right)^2 \\ PSNR &= 10\log_{10}\left(\frac{Z^2}{MSE}\right) \end{cases},$$

where Z is equal to 255. The value of PSNR is usually between 20 and 40. The higher the value, the better the quality.

Structural similarity (SSIM) (Wang et al., 2004): This index compares the SSIM between images based on a comparison of the brightness and contrast characteristics of the images, and it can be shown by:

$$SSIM\left(y_i, \widehat{y_i}\right) = \frac{\left(2\mu_{y_i}\mu_{\widehat{y_i}} + C_1\right)\left(\sigma_{y_i\widehat{y_i}} + C_2\right)}{\left({\mu_{y_i}}^2 + {\mu_{\widehat{y_i}}}^2 + C_1\right)\left({\sigma_{y_i}}^2 + {\sigma_{\widehat{y_i}}}^2 + C_2\right)},$$

where μ and σ represent the mean and variance of image pixels, respectively.

Frechet inception distance (FID) (Heusel et al., 2017): It is a performance index for calculating the distance between a real image and a modified image feature vector. The lower the FID score, the better the image quality generated, and the higher the similarity to the original image.

4.2.2. Quantitative comparison results

For quantitative evaluation, l_1 loss function (Gao and Fang, 2011), PSNR (Hore and Ziou, 2010), SSIM (Wang et al., 2004), and FID (Heusel et al., 2017) are evaluation metrics. The results are shown in Tables 1, 2.

Tables 1, 2 compare the parameters of the seven methods used in the CelebA and CelebA-HQ data sets under four

TABLE 1 Quantitative comparisons of ours with the other six methods in CelebA.

	Masks	CA	GMCNN	MEDFE	RFR	LG-net	Ours
$\ell 1(\%)^{\dagger}$	1–10%	1.77	1.54	1.43	1.57	0.44	0.39
	20–30%	5.28	3.01	3.72	3.74	2.45	2.19
	40–50%	7.92	4.63	7.64	6.51	5.31	5.11
	AVG	4.99	3.06	4.26	3.94	2.73	2.56
PSNR‡	1–10%	33.12	36.29	36.21	37.26	40.72	42.25
	20–30%	24.07	28.33	27.85	29.14	30.67	31.66
	40–50%	21.11	26.08	23.50	25.23	26.09	26.56
	AVG	26.10	30.23	29.19	30.54	32.49	33.49
SSIM‡	1–10%	0.971	0.977	0.990	0.990	0.995	0.996
	20–30%	0.901	0.928	0.945	0.952	0.962	0.986
	40–50%	0.853	0.895	0.844	0.899	0.911	0.913
	AVG	0.908	0.933	0.926	0.947	0.956	0.965
FID†	1–10%	2.14	0.82	0.79	0.85	0.40	0.41
	20–30%	6.82	2.26	3.21	2.73	2.11	2.07
	40–50%	12.39	4.51	7.19	5.22	4.60	5.02
	AVG	7.11	2.53	3.73	2.93	2.37	2.50

‡Higher is better.

†Lower is better.

TABLE 2 Quantitative comparisons of ours with the other six methods in CelebA-HQ.

	Masks	CA	GMCNN	MEDFE	RFR	LG-net	Ours
$\ell 1(\%)^{\dagger}$	1–10%	1.86	1.14	1.02	1.59	0.46	0.39
	20–30%	5.33	3.05	3.68	3.58	2.38	2.11
	40–50%	7.84	4.51	7.65	6.44	5.27	5.03
	AVG	5.01	2.90	4.12	3.87	2.70	2.51
PSNR‡	1–10%	32.66	35.96	36.13	36.39	40.04	41.53
	20–30%	23.94	28.52	27.75	29.07	30.54	31.33
	40–50%	21.98	25.89	23.47	25.09	26.01	26.55
	AVG	26.19	30.12	29.12	30.18	32.19	33.14
SSIM‡	1–10%	0.971	0.984	0.990	0.991	0.995	0.997
	20–30%	0.903	0.933	0.943	0.957	0.968	0.987
	40–50%	0.853	0.897	0.865	0.902	0.917	0.921
	AVG	0.909	0.938	0.932	0.950	0.960	0.968
FID†	1–10%	2.06	0.85	0.84	0.86	0.39	0.37
	20–30%	6.97	2.24	3.17	2.67	2.08	2.11
	40–50%	12.42	4.56	7.12	5.21	4.61	4.47
	AVG	7.15	2.55	3.17	2.91	2.36	2.31

‡Higher is better.

†Lower is better.

different indexes. The smaller the l_1 and FID index values, the better the quality of figures, and the larger the PSNR and SSIM index values, the better the quality of figures. Through quantitative analysis, we can see that, under different coverage and indicators, our method generally outperforms the others. Only when the GMCNN image inpainting method deals with facial images with large area coverage (more than 40% coverage), some evaluation indexes are better than our method. A possible reason is that the jump connection between the residual blocks in our network pays too much attention to the shallow feature information of the image and neglects the processing of the global semantics. In addition, our performance on PSNR and SSIM assessment was significantly better than the other methods, showing that the facial image recovered by our method was of high quality and had a high SSIM with the original image.

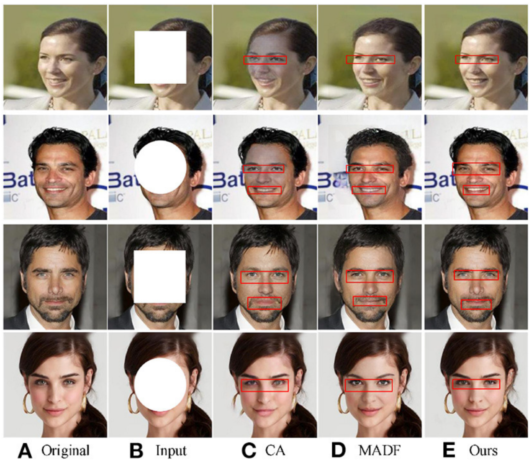


FIGURE 5
Visual comparison of different image inpainting methods on CelebA-HQ and ParisView datasets with regular masks. Obvious differences on the faces are highlighted by red boxes. The face images are adapted from the celeba-HQ dataset, which comes from Karras et al. (2017).

4.2.3. Visual comparison

To better illustrate the inpainting effect, we compared the visual results from different image inpainting methods. As shown in Figure 5, the results of three different methods under the regular mask of CelebA are shown in the first and second lines, while the results of CelebA-HQ datasets are shown in the third and fourth lines. The hole size of the square mask was set as 128 × 128, and the radii of the circle mask was set as 64. From Figure 5, we found that facial images with rectangular masks restored by CA (Yu et al., 2018) and MADF (Zhu et al., 2021) tend to be fuzzy, and problems such as chromatic aberration and excessive discontinuity appear at the edges of the restored areas in the lips and eyeballs. However, facial images restored by the proposed method have clear facial features and good color consistency, making it difficult to distinguish the original image

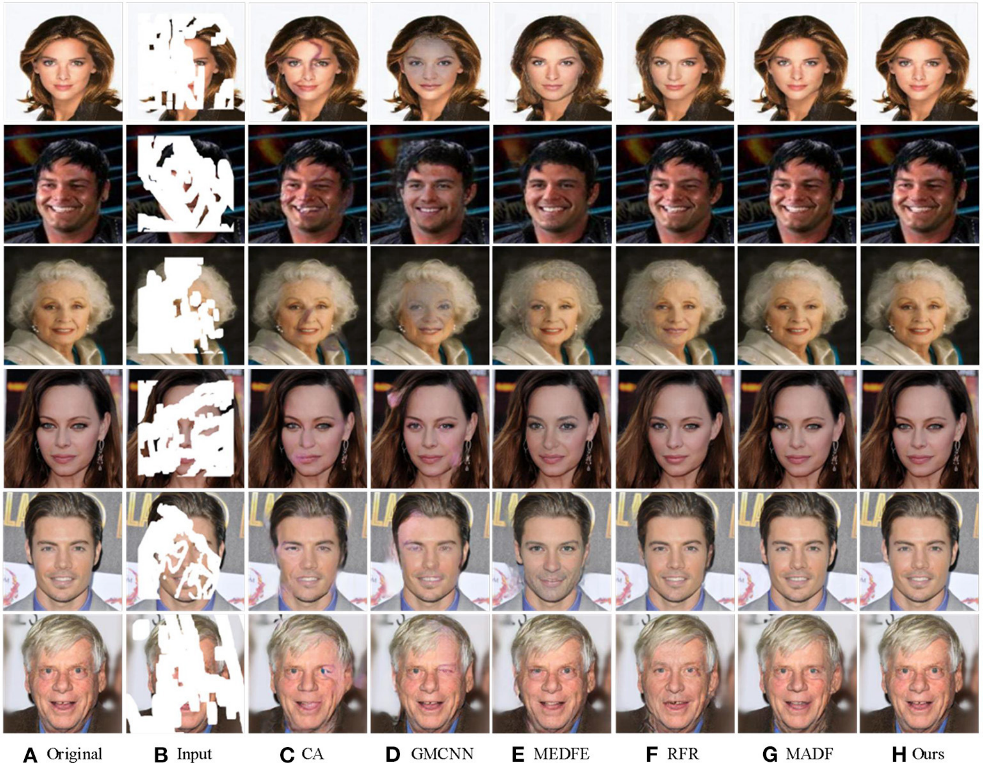


FIGURE 6
The comparison of different image inpainting methods on CelebA-HQ and ParisView with irregular masks. The face images are adapted from the celeba-HQ dataset, which comes from Karras et al. (2017).

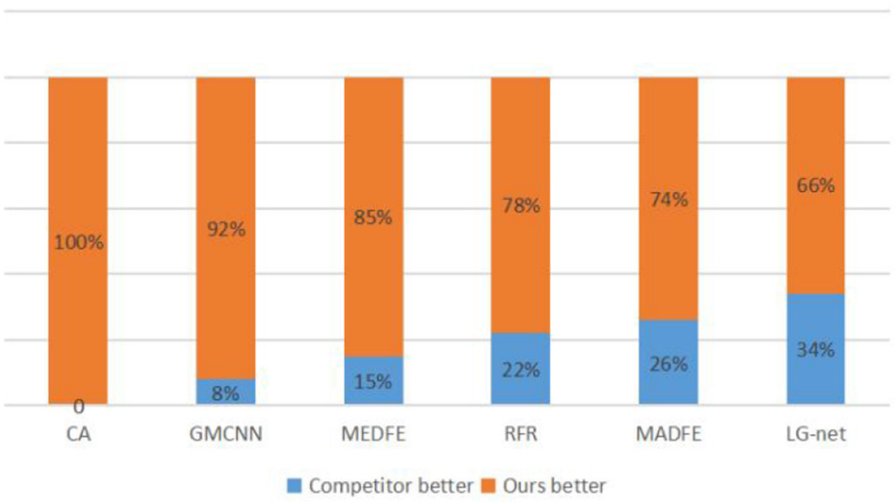


FIGURE 7
Statistical results from a user study on the CelebA data set. The value shows the percentage of each method chosen as the better one.

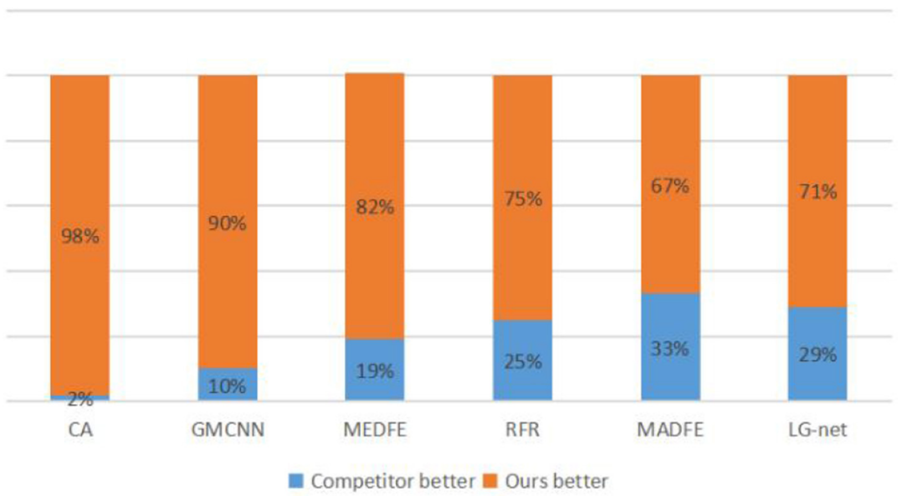


FIGURE 8
Statistical results of a user study on the CelebA-HQ data set. The value shows the percentage of each method chosen as the better one.

from the restored image with the naked eye. All these verify the effectiveness of the proposed method.

To further verify the inpainting effect of our method, we compared the inpainting performance of our method with other competitors on irregular masks.

The corresponding results are shown in Figure 6. The output images from the six different image inpainting methods of CelebA are shown in the first three lines, while the results from the CelebA-HQ dataset are shown in the last three lines.

Compared to the results generated with CA (Yu et al., 2018) and GMCNN (Wang et al., 2018), CLGN eliminated the phenomenon of blurring and distortion in the repair region, and the generation results was smoother and achieved a perfect transition from the damaged region to the undamaged region. MEDFE (Liu et al., 2020) and RFR (Li et al., 2020) offer excellent inpainting performance when the area to be repaired is small. However, for a large area of masks, they showed a wavy visual blur of water, which affects the overall observation effect of the inpainting image. Compared with MADF (Zhu et al., 2021), our GAM attention module-based method is more robust and stable depending on the good results of the local refinement network.

4.2.4. User study

Because the evaluation metrics are not exactly fit human perception, we performed a user study on the Google Forms platform to further compare the visual quality of our method with six other mainstream image inpainting methods. For comparison, we randomly selected 10 pairs of CelebA (Liu et al., 2015) and CelebA-HQ (Karras et al., 2017), where each pair contains two inpainting images, one by a comparable method and another by our method. Note that the input images are covered by the same masked region. Then, we invited 24 volunteers for choosing the more natural and realistic images from each pair. In the end, we totally collected 2,880 votes. From Figures 7, 8, it can be concluded that our method is significantly more likely to be chosen than the other six methods, indicating that the visual quality of inpainting images of our method is superior.

4.3. Ablation studies

To verify the effectiveness of the loss function and a multistage network in our proposed method, ablation studies were performed on CelebA (Liu et al., 2015) and CelebA-HQ (Karras et al., 2017). The ablation experiment in this study as divided into three parts, which analyze the weighted loss network design, and GAM, respectively.

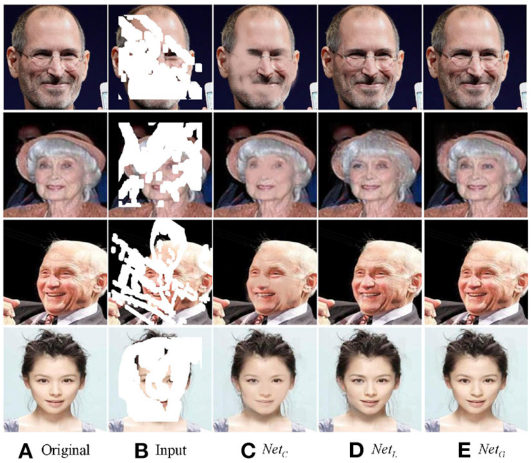


FIGURE 9
The output images of three subnetworks.

4.3.1. Network design

There are three subnetworks in our method: crude inpainting work Net_C, local development network Net_L, and a global attention-based network Net_G. By comparing different variants of CLGN, the effectiveness of our network design can be verified and evaluated. Figure 9 shows the visual comparison, and Table 3 presents the corresponding numerical results. Note that we used incomplete images with one central square hole size of 128 × 128.

By comparing the results of "Net_C" (C), "$Net_C + Net_L$" ($C + L$), "$Net_C + Net_G$" ($C + G$), "$Net_C + Net_L + Net_G$" ($C + L + G$) in Table 4, we conclude that our proposed a multistage network, especially the global attention-based network, has a great effect on the inpainting results. This is probably because different types of networks can handle different types of visual artifacts. Hence, the more types and number of networks, the better the image processing effect.

In addition, we analyzed our proposed method by comparing the inpainting results of three subnetworks and drew a conclusion from Figure 9 that the visual quality of the output images is getting better.

As shown in the first row of Figure 9, since the role of Net_C is to repair the image initially, the output image has a small range of blur. Moreover, Net_L removes local blur, especially those in the face by using the local information and Net_G, finally, recovers complete semantic information and the image is restored to a maximum extent.

TABLE 4 The ablation of attention mechanism on CelebA.

Strategy	PSNR	SSIM	FID	LPIPS
w/o GAM	21.43	0.892	11.28	0.203
w/one CBAM	21.68	0.907	7.90	0.169
w/one GAM	21.79	0.923	8.12	0.178
w/three CBAM	22.98	0.976	7.13	0.126

TABLE 3 The ablation of network design on CelebA and CelebA-HQ data sets.

CelebA data set					CelebA-HQ data set				
Network	C	C + L	C + G	C + L + G	Network	C	C + L	C + G	C + L + G
ℓ1(%)†	7.16	6.98	7.03	6.82	ℓ1(%)†	4.29	4.54	4.57	4.42
PSNR‡	23.01	23.05	23.08	23.14	PSNR‡	26.03	26.37	26.35	26.48
SSIM‡	0.948	0.971	0.969	0.980	SSIM‡	0.948	0.977	0.974	0.988

‡ Higher is better.
† Lower is better.

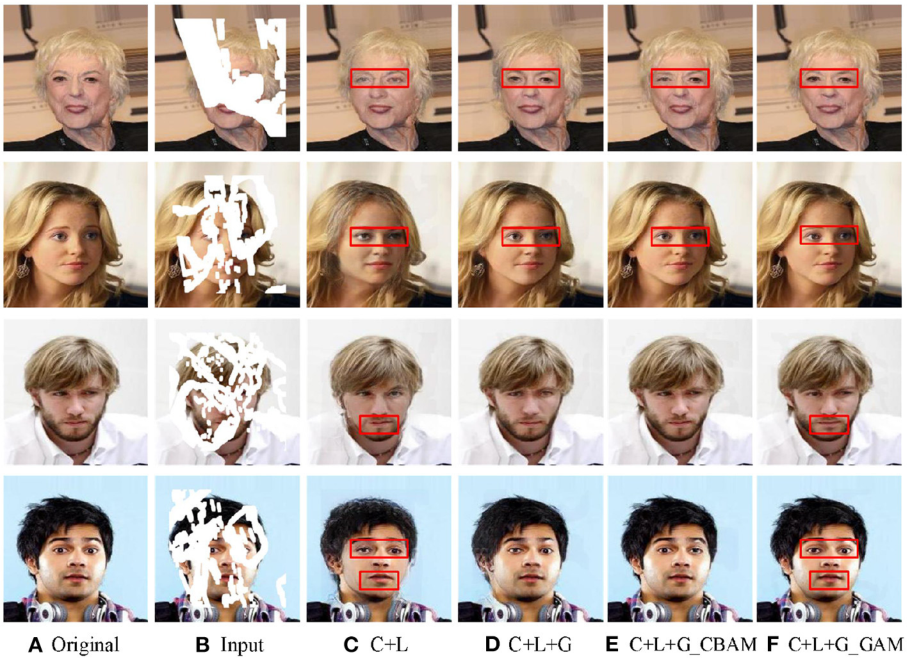


FIGURE 10
The outputs of different network frameworks with different attention mechanisms. Here "*C*" indicates a crude inpainting network, "*L*" means a local refinement network, and "*G*" means a global refinement network without any attention mechanism. "G_CBAM" means a global refinement network based on CBAM, while "G_GAM" means a GAM-based global refinement network. Obvious differences on the faces are highlighted by red boxes. The face images are adapted from the celeba-HQ dataset, which comes from Karras et al. (2017).

TABLE 5 The ablation of loss functions on the CelebA data set.

Strategy	PSNR	SSIM	FID	LPIPS
w/o reconstruction loss	23.01	0.958	7.62	0.128
w/o adversarial loss	22.93	0.931	7.27	0.134
w/o TV loss	23.04	0.943	7.31	0.151
w/o perceptual loss	22.73	0.907	7.18	0.133
w/o style loss	22.98	0.972	7.23	0.136
All	23.14	0.980	7.11	0.124

4.3.2. Attention mechanism

To study the key role of GAM in the network, we conducted an ablation experiment on it. We attempted the following situations: remove the attention mechanism, place CBAM, and deploy GAM. The experimental results are presented in Table 4. From Table 4, it can be concluded that FID is greatly affected. while others are only a little affected by the attention mechanism. Moreover, compared to CBAM, GAM has an excellent effect on facial image inpainting.

Next, we analyzed and compared the visual results of the different networks in Figure 10. From Figure 10, the results of "Net_C+Net_L" can only roughly repair the whole image, but there are artifacts or mismatches in the eyes, the mouth, and other parts. In contrast, attention mechanism-based network is more coordinated in global semantics and has a high similarity with the original image.

4.3.3. Loss functions

In our study, we introduced five loss functions, namely reconstruction loss, adversarial loss, TV loss, perceptual loss, and style loss. Then, we conducted ablation experiments on the CelebA-HQ (Karras et al., 2017) dataset by removing these five loss functions from the network and analyzing the PSNR, SSIM, FID, and learned perceptual image patch similarity (LPIPS) values of the inpainting images. Note that we used incomplete images with one center square hole size of 128 × 128. From Table 5, it can be concluded that reconstruction loss plays the most critical role in performance optimization and that perceptual loss and style loss have the least impact on the performance of image inpainting.

5. Conclusions and future works

Facial image inpainting technology has practical significance in many fields. In this study, we proposed a multistage GAN (CLGN) for GAM-based facial image inpainting. This method combined the normalization of feature layer output in a deep convolutional GAN with the guidance of GAM to improve the robustness and accuracy of image detail recovery. As human faces have a common structure with different features such as the nose, the mouth, and the eyes, a multistage (crude-local-global) network can play the complete restoration role in distinct parts. Moreover, a skip connection was introduced using an encoder-decoder network to compensate for the loss of features due to down-sampling. The proposed method was compared with several inpainting methods on CelebA (Liu et al., 2015) and CelebA-HQ (Karras et al., 2017), and it had better performance than the mainstream traditional image inpainting method in both qualitative and quantitative analyses.

However, from the perspective of inpainting results, although our methods can predict a reasonable result according to the incomplete image, there are still some inevitable differences in color and texture details compared to the actual values. The guidance of structural information ensures its overall structure to some extent, but it is difficult to approach the true value for high-level semantic repairs such as the human eyes and mouths. From the perspective of the training process, large datasets can ensure that network training fits the model better, but at the same time, the long training time of the network becomes a thorny problem. Therefore, in subsequent study, we should focus on the facial image inpainting of higher semantics, which can ensure the credibility of the results and bring them closer to their actual value. At the same time, when designing a network for large data sets, network performance should be guaranteed and network training time should be minimized.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

XL: conception and design of study and drafting the manuscript. RL: writing—reviewing and editing. WZ: data curation and revising the manuscript critically for important intellectual content. EM: analysis and/or interpretation of data and software. All authors contributed to the article and approved the submitted version.

Funding

Project 202210145053 Supported by National Training Program of Innovation and Entrepreneurship for Undergraduates.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Baeza, I., Verdoy, J. A., Villanueva-Oller, J., and Villanueva, R. J. (2009). ROI-based procedures for progressive transmission of digital images: a comparison. *Math. Comput. Model.* 50, 849–859. doi: 10.1016/j.mcm.2009.05.014

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY: ACM Press), 417–424. doi: 10.1145/344779.344972

Cho, H., Lim, S., Choi, G., and Min, H. (2017). Neural stain-style transfer learning using gan for histopathological images. *arXiv [Preprint]*. arXiv: 1710.08543.

Criminisi, A., Pérez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.* 13, 1200–1212. doi: 10.1109/TIP.2004.833105

Efros, A. A., and Leung, T. K. (1999). "Texture synthesis by non-parametric sampling," in *Proceedings of the Seventh IEEE International Conference on Computer Vision* (Piscataway, NJ: IEEE), 1033–1038.

Elharrouss, O., Almaadeed, N., Al-Maadeed, S., and Akbari, Y. (2020). Image inpainting: a review. *Neural Process. Lett.* 51, 2007–2028. doi: 10.1007/s11063-019-10163-0

Gao, X., and Fang, Y. (2011). A note on the generalized degrees of freedom under the L1 loss function. *J. Stat. Plan. Inference* 141, 677–686. doi: 10.1016/j.jspi.2010.07.006

Ge, H., Xia, Y., Chen, X., Berry, R., and Wu, Y. (2018). "Fictitious gan: training gans with historical models," in *Proceedings of the European Conference on Computer Vision* (Munich: Berlin Springer). doi: 10.1007/978-3-030-01246-5_8

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial networks. *Commun. ACM* 63, 139–144. doi: 10.1145/3422622

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Piscataway, NJ; Las Vegas, NV: IEEE), 770–778. doi: 10.1109/CVPR.2016.90

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inform. Process. Syst.* 30, 12. doi: 10.48550/arXiv.1706.08500

Hore, A., and Ziou, D. (2010). "Image quality metrics: PSNR vs. SSIM," in *Proceedings of the 20th International Conference on Pattern Recognition, ICPR 2010* (Istanbul: IEEE Computer Society). doi: 10.1109/ICPR.2010.579

Iizuka, S., Simo-Serra, E., and Ishikawa, H. (2017). Globally and locally consistent image completion. *ACM Trans. Graph.* 36, 1–14. doi: 10.1145/3072959.3073659

Iskakov, K. (2021). *Qd-imd: Quick Draw Irregular Maskdataset*. Available online at: https://github.com/karfly/qd-imd (accessed July 29, 2021).

Jin, J., Hu, X., He, K., Peng, T., Liu, J., and Yang, J. (2021). "Progressive semantic reasoning for image inpainting," in *Companion Proceedings of the Web Conference 2021* (New York, NY: Association for Computing Machinery). doi: 10.1145/3442442.3451142

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). "Perceptual losses for real-timestyle transfer and super-resolution," in *European Conference on Computer Vision* (Cham: Springer), 694–711. doi: 10.1007/978-3-319-46475-6_43

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of GANS for improved quality, stability, and variation. *arXiv [Preprint]*. arXiv: 1710.10196. doi: 10.48550/arXiv.1710.10196

Li, J., Wang, N., Zhang, L., Du, B., and Tao, D. (2020). "Recurrent feature reasoning for image inpainting," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Piscataway, NJ; Seattle, WA: IEEE). doi: 10.1109/CVPR42600.2020.00778

Liu, G., Reda, F. A., Shih, K. J., Wang, T.-C., Tao, A., and Catanzaro, B. (2018). "Image inpainting for irregular holes using partial convolutions," in *European Conference on Computer Vision*. p. 85–100.

Liu, H., Jiang, B., Song, Y., Huang, W., and Yang, C. (2020). "Rethinking image inpainting via a mutual encoder-decoder with feature equalizations," in *European Conference on Computer Vision* (Cham: Springer). doi: 10.1007/978-3-030-58536-5_43

Liu, Y., Shao, Z., and Hoffmann, N. (2021). Global attention mechanism: retain information to enhance channel-spatial interactions. arXiv preprint arXiv:2112.05561.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). "Deep learning face attributes in the wild," in *Proceedings of the IEEE International Conference on Computer Vision* (Munich: Berlin Springer). doi: 10.1109/ICCV.2015.425

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. (2017). "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision* (Piscataway, NJ; Venice: IEEE). doi: 10.1109/ICCV.2017.304

Meyers, F. J., and Scott, J. W. (1994). *Sensor System Having Nonuniformity Suppression with Image Preservation*. Los Angeles, CA: Hughes Aircraft Company.

Noh, H., Hong, S., and Han, B. (2015). *Learning Deconvolution Network for Semantic Segmentation IEEE*. Piscataway, NJ; Santiago: IEEE. doi: 10.1109/ICCV.2015.178

Odena, A. (2016). Semi-supervised learning with generative adversarial networks. *arXiv [Preprint]*. arXiv: 1606.01583. doi: 10.48550/arXiv.1606.01583

Park, J., Woo, S., Lee, J. Y., and Kweon, I. S. (2018). *BAM: Bottleneck Attention Module*. Newcastle upon tyne: BMVA.

Parmar, C. (2011). *Automatic image inpainting for the facial images of monuments* (Dissertation). Dhirubhai Ambani Institute of Information and Communication Technology.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). "Context encoders: feature learning by inpainting," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition* (Piscataway, NJ; Las Vegas, NV: IEEE). doi: 10.1109/CVPR.2016.278

Qin, Z., Zeng, Q., Zong, Y., and Xu, F. (2021). Image inpainting based on deep learning: a review. *Displays* 69, 102028. doi: 10.1016/j.displa.2021.102028

Quan, W., Zhang, R., Zhang, Y., Li, Z., Wang, J., and Yan, D. M. (2022). Image inpainting with local and global refinement. *IEEE Trans. Image Process*. 31, 2405–2420. doi: 10.1109/TIP.2022.3152624

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (New York, NY: Springer International Publishing). doi: 10.1007/978-3-319-24574-4_28

Shen, M. Y., and Kuo, C. (1997). Review of image postprocessing techniques for compression artifact removal. *J. Vis. Commun. Image Represent*. 9, 2–14.

Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *arXiv [Preprint]*. arXiv: 1409.1556v6. doi: 10.48550/arXiv.1409.1556

Singh, G., and Goel, A. K. (2020). "Face detection and recognition system using digital image processing," in *Proceedings of the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications* (Bengaluru: IEEE). doi: 10.1109/ICIMIA48430.2020.9074838

Sun, Q., Ma, L., Oh, S. J., Van Gool, L., Schiele, B., and Fritz, M. (2018). "Natural and effective obfuscation by head inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Piscataway, NJ; Salt Lake City, UT: IEEE). doi: 10.1109/CVPR.2018.00530

Wang, Q., Ji, Y., Hao, Y., and Cao, J. (2020). GRL: Knowledge graph completion with GAN-based reinforcement learning. *Knowl. Based Syst*. 209, 106421. doi: 10.1016/j.knosys.2020.106421

Wang, Y., Tao, X., Qi, X., Shen, X., and Jia, J. (2018). Image inpainting *via* generative multi-column convolutional neural networks. *Adv. Neural Inform. Process. Syst*. 31, 10. doi: 10.48550/arXiv.1810.08771

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* 13, 600–612. doi: 10.1109/TIP.2003.819861

Wei, J., Lu, G., Liu, H., and Yan, J. (2019). Facial image inpainting with deep generative model and patch search using region weight. *IEEE Access* 7, 67456–67468. doi: 10.1109/ACCESS.2019.2919169

Woo, S., Park, J., Lee, J. Y., and Kweon, I. S. (2018). "CBAM: convolutional block attention module," in *The European Conference on Computer Vision* (Munich: Berlin Springer). doi: 10.1007/978-3-030-01234-2_1

Yan, Z., Li, X., Li, M., Zuo, W., and Shan, S. (2018). *Shift-Net: Image Inpainting Via Deep Feature Rearrangement*. Munich: Berlin Springer. doi: 10.1007/978-3-030-01264-9_1

Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., and Li, H. (2017). "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI; Piscataway, NJ: IEEE). doi: 10.1109/CVPR.2017.434

Yang, J., Liu, J., and Wu, J. (2020). Facial image privacy protection based on principal components of adversarial segmented image blocks. *IEEE Access* 8, 103385–103394. doi: 10.1109/ACCESS.2020.2999449

Yeh, R. A., Chen, C., Yian Lim, T., Schwing, A. G., Hasegawa-Johnson, M., and Do, M. N. (2017). "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI; Piscataway, NJ: IEEE). doi: 10.1109/CVPR.2017.728

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2018). "Generative image inpainting with contextual attention," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Piscataway, NJ; Salt Lake City, UT: IEEE). doi: 10.1109/CVPR.2018.00577

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2019). "Free-form image inpainting with gated convolution," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision* (IEEE), 4470–4479. doi: 10.1109/ICCV.2019.00457

Zhu, M., He, D., Li, X., Li, C., Li, F., Liu, X., et al. (2021). "Image inpainting by end-to-end cascaded refinement with mask awareness," in *IEEE Transactions on Image Processing* (Piscataway, NJ: IEEE). doi: 10.1109/TIP.2021.3076310

Zn, A., Gz, A., and Hui, Y. B. (2021). A review on the attention mechanism of deep learning. *Neurocomputing* 452, 48–62. doi: 10.1016/j.neucom.2021.03.091

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 19 January 2023
DOI 10.3389/fnbot.2023.1120560

Check for updates

OPEN ACCESS

EDITED BY
Xin Ning,
Institute of Semiconductors (CAS), China

REVIEWED BY
Gabriel Gomes,
State University of Campinas, Brazil
Cheng Tong,
Saigon University, Vietnam
Dazhong Shu,
Sanya University, China

*CORRESPONDENCE
Yue Teng
✉ 689666@swansea.ac.uk

RECEIVED 10 December 2022
ACCEPTED 02 January 2023
PUBLISHED 19 January 2023

CITATION
Teng Y, Bao J, Li Y and Ye H (2023) Effect analysis of neural network robot system in music relaxation training to alleviate adverse reactions of chemotherapy in patients with breast cancer.
Front. Neurorobot. 17:1120560.
doi: 10.3389/fnbot.2023.1120560

Effect analysis of neural network robot system in music relaxation training to alleviate adverse reactions of chemotherapy in patients with breast cancer

Yue Teng[1]*, Jinlei Bao[2], Yinfeng Li[3] and Haichun Ye[4]

[1]College of Human and Health Sciences, Swansea University, Swansea, United Kingdom, [2]College of Nursing, Shandong Xiehe University, Jinan, Shandong, China, [3]School of Medicine, Sichuan Cancer Hospital and Institute, Sichuan Cancer Centre, University of Electronic Science and Technology of China, Chengdu, China, [4]Department of Geriatric Services and Management, School of Humanities Education, Ningxia Vocational Technical College of Industry and Commerce, Yinchuan, China

Music therapy is a common method to relieve anxiety and pain in cancer patients after surgery in recent years, but due to the lack of technical and algorithmic support, this therapy is not particularly stable and the therapeutic effect is not good. In this study, a neural network robotic system based on breast cancer patients was designed to analyze the effect of music relaxation training on alleviating adverse reactions after chemotherapy in breast cancer patients. Firstly, this paper introduces the necessity of neural network robot system research under the background of music therapy, and then summarizes the positive effect of music relaxation therapy on alleviating adverse reactions after chemotherapy in breast cancer patients, finally, uses neural network robot system to construct music therapy system. The experimental results show that the new music therapy proposed in this study has a good effect in alleviating the adverse reactions of breast cancer patients after chemotherapy, and the cure rate is increased by 7.84%. The research results of this paper provide reference for the next development of neural network robot system in the medical field.

KEYWORDS

breast cancer patients, alleviate adverse reactions, music therapy, neural network robot system, breast cancer

1. Introduction

Breast cancer patients have to give up or interrupt chemotherapy because they cannot tolerate the adverse reactions brought by chemotherapy, which ultimately affects the therapeutic effect. The purpose of this manuscript is to use music relaxation training to alleviate adverse reactions during chemotherapy, promote muscle and nerve relaxation of patients, reduce anxiety and pain of patients, and provide reference value for improving the quality of life of patients and formulating cure measures.

Alleviating the adverse reactions of patients during treatment is the focus of treatment work, and also the research focus of the medical community. Fu Yali evaluated the adverse effects of multi-target tyrosine kinase inhibitors in the treatment of gastrointestinal tumors. Finally, it was concluded that multi target tyrosine kinase inhibitors have a good effect in the treatment of mild or moderate adverse reactions (Fu et al., 2018). Ji Jing analyzed the effect of high-quality nursing on relieving adverse reactions of liver cancer treatment. Finally, high-quality nursing intervention in the treatment of liver cancer can reduce the incidence of surgical pain and postoperative adverse reactions (Jing, 2020). Deshmukh Vineeta applied the nano particle system to the adverse reactions during cancer chemotherapy. Practice showed that this system cannot only explore the delivery effect of drugs, but also reduce the systemic toxicity of patients (Vineeta, 2021). Yang Qian explored the role of chemical photothermal combination therapy in inhibiting adverse reactions of cancer chemotherapy by constructing a porous nano particle system. The research showed that this therapeutic approach showed a strong anti-cancer therapeutic effect (Qian et al., 2018). Giavina-Bianchi Pedro reviewed the common adverse reactions caused by cancer chemotherapy drugs - hypersensitivity reactions. It was concluded that rapid drug desensitization can minimize allergic reaction when hypersensitivity occurs (Giavina-Bianchi et al., 2017). Demaria Marco believed that many genotoxic chemotherapy would produce adverse reactions, and pointed out that aging cells would cause some side effects of chemotherapy, and provide a new target for reducing the toxicity of anti-cancer therapy (Marco, 2017). Singh Kanchanlata discussed the effects of different antioxidants and their analogues on adverse reactions during chemotherapy. Comprehensive data showed that antioxidant supplementation during chemotherapy can improve the therapeutic effect and improve the quality of life of patients (Singh et al., 2018). These researches on relieving adverse reactions of patients are still of reference value, but they have not been applied to music relaxation training.

With the continuous updating of therapeutic techniques, music relaxation training (music therapy) has achieved good results in recent years. Sandler Hubertus analyzed the role of compact disc music in the treatment of patients with depression or anxiety disorder, and finally found that some excellent music can induce patients to have a relaxed state and subjective well-being (Hubertus, 2017). Ghezeljeh T. Najafi studied the effects of massage and music on pain intensity and anxiety intensity of burn patients. The research results showed that music, massage and the combination of these two interventions are effective in reducing pain and anxiety intensity (Najafi Ghezeljeh et al., 2017). Nelson Kirsten analyzed the effect of introducing music to adolescents before surgery to reduce pain and anxiety. Practice showed that during the operation, the pain and anxiety of teenagers have been significantly reduced (Nelson et al., 2017). Liao Juan summarized the current situation of the application of the five elements music therapy in the depression psychology of cancer patients. The study found that, as a simple and effective intervention, this treatment method reduced the anxiety and depression of cancer patients, and provided an effective way for better self-management in cancer treatment (Liao et al., 2018). Bradt Joke applied music relaxation training to dental treatment, and finally found that the anti-anxiety effect of music can reduce the anxiety state and pain of patients to a certain extent (Bradt and Teague, 2018). Pradopo Seno combined aromatherapy and music therapy for dental treatment of pediatric patients. Practice showed that aromatherapy and music therapy can divert children's attention and reduce their anxiety (Pradop et al., 2017). Golino Amanda J studied the effect of active music therapy intervention on physiological parameters of patients in intensive care unit. The study found that the sleep quality of patients was significantly improved after music intervention (Golino et al., 2019). The above application of music therapy in the medical field is more detailed, but it does not involve the alleviation of adverse reactions of chemotherapy in breast cancer patients.

During the treatment of breast cancer, many patients are troubled by the adverse reactions caused by chemotherapy all the time, and what people can do now is to alleviate these adverse reactions. Conventional treatment methods have certain defects, and it is inevitable that patients' psychological and physiological problems would not be taken into account. Music relaxation training is used to alleviate adverse reactions of patients, which is conducive to reducing anxiety and pain. In this manuscript, neural network robot system is applied to music therapy to improve the relief effect of music relaxation training and change the postoperative quality of life of patients. Experimental results show that the music therapy method proposed in this manuscript has a good effect in alleviating the adverse reactions of breast cancer patients during chemotherapy, reducing the anxiety and pain of patients.

2. Music therapy in the context of neural network robot system

In recent years, robot system has become an important research and application field. Many robots are used to complete complex and even dangerous tasks. With the progress of computer science and technology such as sensor technology, medical imaging technology, artificial neural network technology, and modern information processing, robot technology has been applied to the medical field, resulting in medical robots. Medical robots are mainly used for analyzing patients' conditions, assisting in surgery, therapeutic training, rehabilitation treatment, etc. Because medical data needs to be stored in large quantities, medical robots must perform a large amount of calculations to complete complex tasks, and neural networks can calculate and integrate large amounts of data, which is why neural network robot systems are gradually entering the medical field.

With the increasing emphasis on music therapy in the medical community, its development space has been greatly improved. As shown in **Figure 1**, music therapy is a relatively complex discipline, which includes not only medicine, but also psychology and neural network robot system. AI is a very broad science, which consists of different fields, such as machine learning, computer vision, etc. In general, one of the main goals of AI research is to enable machines to be competent for some complex tasks that usually require human intelligence. Under the background of neural network robot system, music therapy is used for various diseases, such as mental disease, nervous system, cardiovascular disease, etc. As a new therapeutic approach, music therapy is not mature enough, and problems such as untimely monitoring of the treatment process and difficult evaluation of the treatment effect often occur. The emergence of neural network robot system has brought a development opportunity for the innovation of treatment methods. The application of neural network and robot

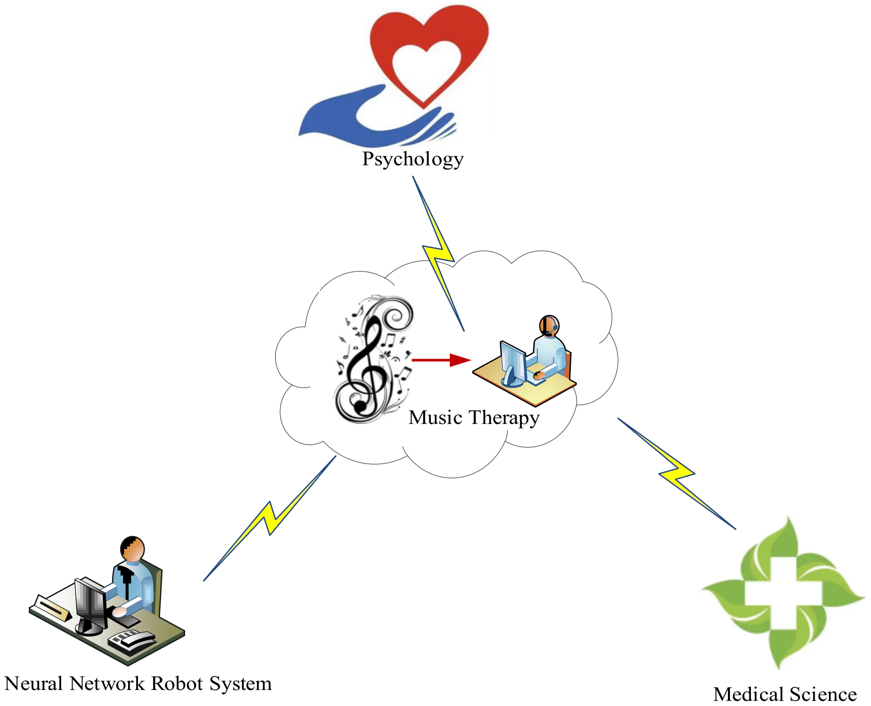


FIGURE 1
Scope of music therapy.

technology to music therapy can further improve the monitoring effect and the accuracy of information acquisition. In addition, relevant evaluation models and music recommendation systems can also be established by using relevant machine learning algorithms.

3. Necessity of music relaxation training

Relaxation training can offset the negative effects of physiological and psychological stress, restore the balance and coordination of human body, psychology and spirit, help individuals to face the challenges of life in a healthier way, and make the human body's involuntary reactions, such as heartbeat, respiration, blood pressure, and adrenaline secretion under autonomous control. At present, relaxation training has been widely used in patients with bronchial asthma, coronary heart disease, hypertension, diabetes, cancer, surgery, childbirth and chemotherapy, and has achieved relatively positive results. During chemotherapy for cancer patients, relaxation training can improve their anxiety, depression and other negative emotions by reducing nerve stimulation. When patients have a positive and good attitude, it would increase their resistance, and would also have a certain role in reducing the spread of cancer cells. To sum up, breast cancer patients need chemotherapy after surgery, which would cause psychological and physiological discomfort. Music relaxation training can reduce the stress response of tumor patients, regulate the emotion of tumor patients, reduce anxiety and depression, improve physical symptoms, alleviate pain, and enhance immune function.

4. Positive role of music relaxation training in alleviating the adverse reactions of chemotherapy in breast cancer patients

4.1. Alleviating adverse reactions of chemotherapy

The patient has just undergone the treatment of breast cancer surgery and is still in a state of serious psychological and physiological stress. The ongoing chemotherapy would further weaken the immune system of the body, reduce the body's resistance, and produce various forms of adverse reactions. Playing some gentle music during relaxation training can create a comfortable and calm treatment environment for patients, eliminate the negative impact of physical and mental pressure, restore physical and mental balance and harmony, and help patients better cope with the pain caused by cancer. In this way, the patient's heart rate, respiration, blood pressure, adrenaline secretion, etc., would become stable, and the adverse reactions during chemotherapy would also be reduced.

4.2. Reducing the anxiety of patients during chemotherapy

Anxiety is a complex emotional state of fear of adverse consequences. A certain amount of anxiety helps to improve the psychological tension of the body and enhance the adaptability to stressors. If it is too strong, it would weaken this ability, and if it is too weak, it would lack an objective evaluation of threatening

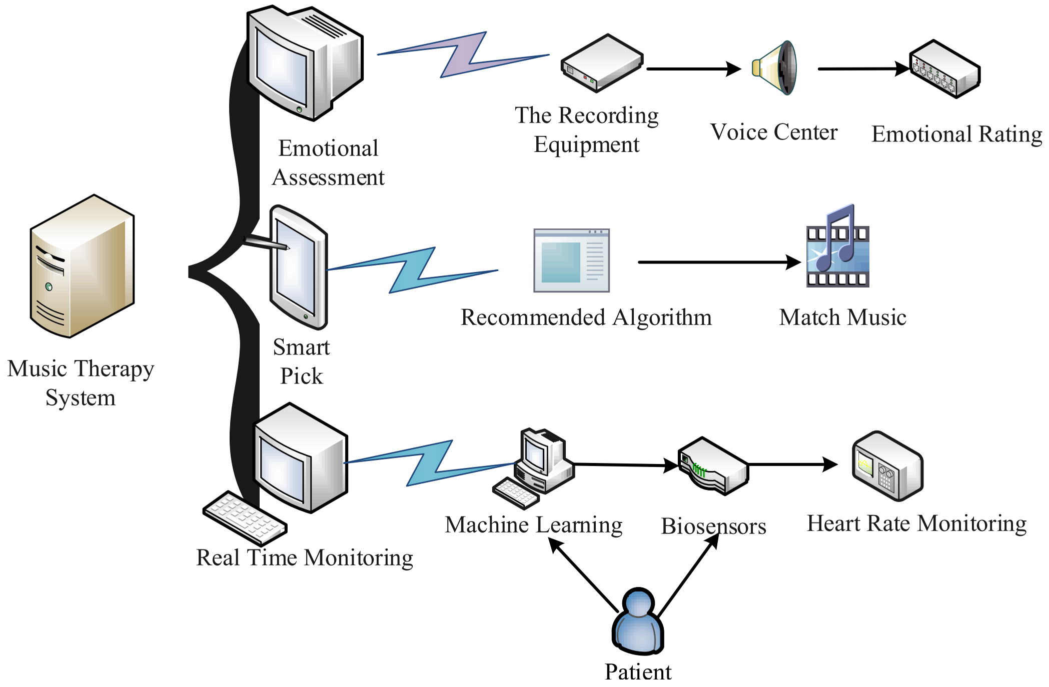


FIGURE 2
Overall framework of music therapy system.

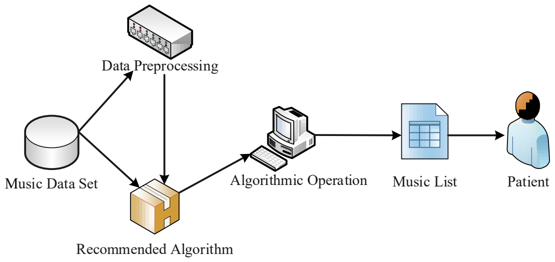


FIGURE 3
Application process of recommendation algorithm in music recommendation.

situations. Relaxation training combined with music can make the whole body of the patient enter a relaxed state, and keep the heart, brain and other important organs in a stable state. Anxiety would decrease with the gradual relaxation of the body functions. The mood of the patient would become better, and the anxiety state would become weaker, which is conducive to improving the treatment effect during chemotherapy.

5. Methods and techniques of music relaxation training

5.1. Music therapy system

Combined with the neural network robot system, this manuscript proposes a new music therapy system. As shown in Figure 2, the specific functions of the system are divided into three parts: emotional evaluation, intelligent selection and real-time monitoring.

Affective assessment is the emotional assessment of breast cancer patients in the form of voice. The voice recorder or other intelligent voice equipment is used to collect the patient's words, and then stored in the voice data center. Finally, the emotional state of the patient is divided into three levels: relaxed, normal, and excited by voice analysis equipment.

5.1.1. Intelligent selection

According to the evaluation results, different types of music are matched for patients with different emotional states through music recommendation algorithms, that is, the first stage of music therapy.

5.1.2. Real time monitoring

This part mainly uses machine learning technology and physiological sensors to monitor the heart rate indicators of breast cancer patients in real time, and then analyzes the current psychological status of patients according to specific indicators. When the patient's heart rate index changes constantly, the index information would be transmitted to the computer in time, and then the computer would judge the heart rate index in the form of machine learning.

5.2. Establishment of music therapy music library

As the emotional state is divided into three levels: relaxed, normal and excited, the music therapy music library is also mainly composed of relaxed, normal and excited music. In addition, digital signal processing technology is used to describe the characteristics of different types of music, quantify the characteristics, and classify the types of music. The music library selects thousands of different styles of music works, and then extracts sound features from their music

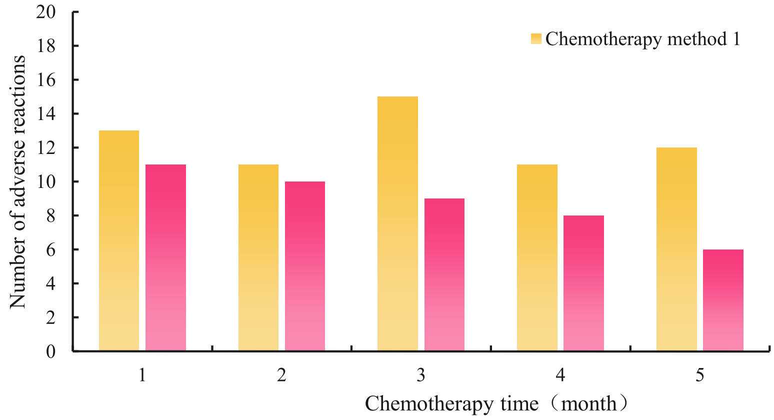


FIGURE 4
Number of adverse reactions of patients under two chemotherapy methods.

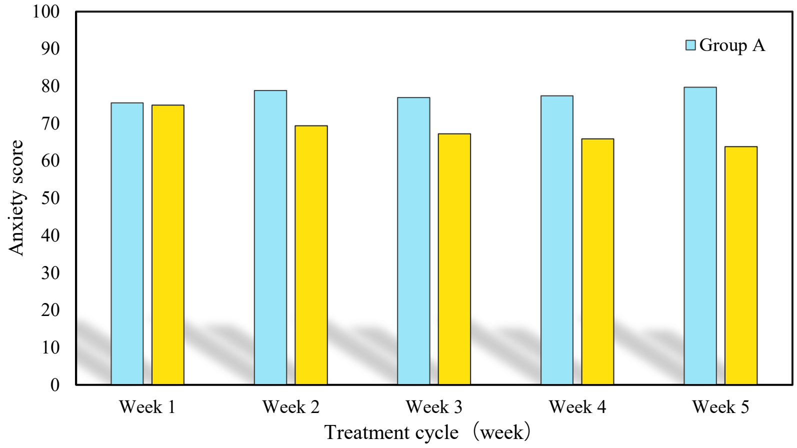


FIGURE 5
Anxiety score of breast cancer patients within 5 weeks under two treatment methods.

signals. Special diagnosis extraction includes melody speed, rhythm, frequency, length of time, melody, timbre, lyrics features, etc.

5.3. Feedback regulation system of music therapy based on biosensor

Biosensors are sensitive devices to biological reactions, which can convert the concentration of biological reactions into electrical signals (Sopan and Patil, 2022). The biosensor has the advantages of good selectivity, high sensitivity, fast analysis speed and low cost, and it can conduct online continuous monitoring in complex systems. This system uses the biosensor module that can be used for ECG monitoring to collect the heart rate information of breast cancer patients in real time. Heart rate information can reflect the psychological state of patients in real time. When patients are excited, normal and relaxed, their heart rate signals would change to varying degrees. The feedback regulation system would have a set threshold value, and then automatically adjust the music type and level according to the threshold value and heart rate signal value. The specific adjustment mechanism is that when the heart rate signal is greater than the threshold value, the system would play highly soothing music; when the heart rate signal is lower than the threshold value, the system would play gently soothing music to adjust the patient’s emotional state; when the heart rate signal value is between two thresholds, the system would play moderate soothing music.

5.4. Music recommendation algorithm for alleviating adverse reactions of chemotherapy in breast cancer patients

In order to make the recommendation effect more comprehensive, the music recommendation algorithm proposed in this manuscript to alleviate the adverse reactions of chemotherapy in breast cancer patients includes content-based recommendation algorithm, collaborative filtering based recommendation algorithm, Item based Collaborative Filtering (Item-CF) recommendation algorithm and recommendation algorithm based on the weighted fusion of content and collaborative filtering. The application process of the recommendation algorithm is shown in **Figure 3**. First, data preprocessing and recommendation algorithm processing are performed on the music dataset, and then the corresponding song

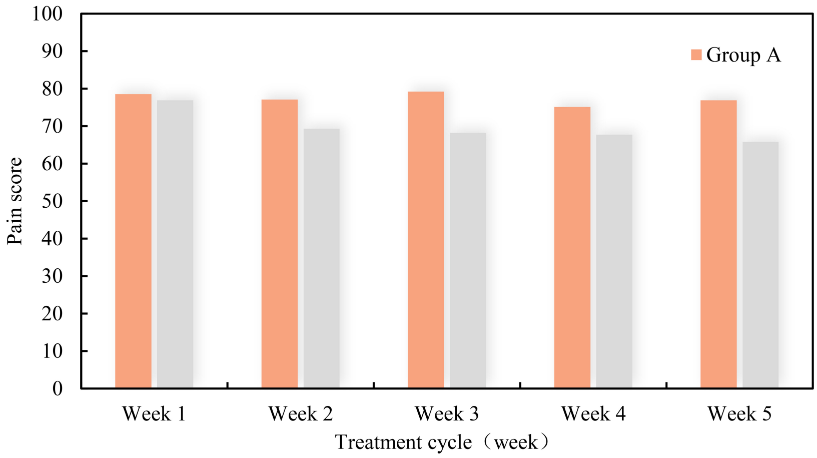


FIGURE 6
Pain score of breast cancer patients within 5 weeks under two treatment methods.

list would be generated after the operation, and finally recommended to breast cancer patients.

5.4.1. Content based recommendation algorithm

According to the text information of music content, the preference matrix of breast cancer patients is obtained by using the Term Frequency - Inverse Document Frequency (TF-IDF) method. Assuming that the set of given music is $O = \{O_1, O_2, O_3, \cdots, O_q\}$, the characteristic (key) phrase is $I = \{x_1, x_2, x_3, \cdots x_m\}$, and O_y represents the fourth music, the formula of word frequency is:

$$G_{TF}(x, y) = \frac{f(x, y)}{\sum_{k \in y} f_{k,y}} \tag{1}$$

Among them, $f(x, y)$ is the number of occurrences of word x in music y; $\sum_{k \in y} f_{k,y}$ is the total number of occurrences of all words in music y; $k \in y$ is the number of occurrences of words in music y. The anti document frequency formula is:

$$G_{IDF}(x) = \log \frac{Q}{q(x)} \tag{2}$$

Among them, Q refers to the number of all music, and $q(x)$ refers to the number of music that has appeared in the feature word x of q.

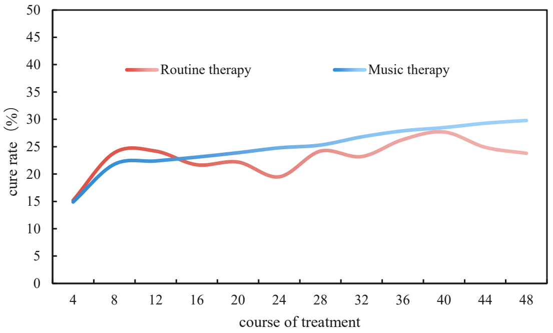


FIGURE 7
Cure rate under two treatment methods.

The combined TF-IDF weight of feature word x in music y is calculated as:

$$G_{TF-IDF}(x, y) = G_{TF}(x, y) \times G_{IDF}(x) \tag{3}$$

Among them, $G_{TF-IDF}(x, y)$ represents the word corresponding to the characteristic word x in the y music. It is normalized:

$$Z_{yx} = \frac{G_{TF-IDF}(x, y)}{\sqrt{\sum_{x=1}^{|I|} G_{\mathrm{TF}-IDF}(x, y)^2}} \tag{4}$$

Among them, Z_{yx} refers to the normalization of the x word of the y music, so the music preference matrix of breast cancer patients can be obtained:

$$G_u = \begin{pmatrix} Z_{11} & Z_{12} & \cdots & Z_{1q} \\ Z_{21} & Z_{22} & \cdots & Z_{2q} \\ \vdots & \vdots & & \vdots \\ Z_{q1} & Z_{q2} & \cdots & Z_{qq} \end{pmatrix} \tag{5}$$

5.4.2. Collaborative filtering recommendation algorithm

Collaborative filtering algorithm is a more famous and commonly used recommendation algorithm, which is based on the mining of user historical behavior data to find user preferences, and predict the products that users may like to recommend. It must first find "similar breast cancer patients", and then look for "similar items that breast cancer patients like". First, cosine similarity is used:

$$Z_{uv} = \frac{|Q(u) \bigcap Q(v)|}{\sqrt{|Q(u)|\,|Q(v)|}} \tag{6}$$

Formula (6) is improved, including:

$$Z_{uv} = \frac{\sum_{a \in Q(u) \bigcap Q(v)} \frac{1}{\lg(1+|Q(a)|)}}{\sqrt{|Q(u)|\,|Q(v)|}} \tag{7}$$

The reciprocal part of the molecule in Formula (7) is used to punish the popular music in the common preference list of breast cancer patients u and v, and reduce the impact of popular songs on the similarity of breast cancer patients. The formula of breast cancer

TABLE 1 Satisfaction of four categories of personnel with new music therapy.

	Very satisfied		Satisfied		Dissatisfied	
	Number of people	Proportion (%)	Number of people	Proportion (%)	Number of people	Proportion (%)
Patient	89	59.3	43	28.7	18	12
Family members	85	57.7	41	27.3	24	16
Therapist	78	52	47	31.3	25	16.7
Hospital leaders	69	46	52	34.7	29	19.3

According to the data in this table, most of the four types of people were very satisfied with the new music therapy, and very few were dissatisfied, which means that the new music therapy is recognized by everyone.

patients' preference for songs is:

$$G\left(u,y\right)=\sum_{v\in S(u,K)\bigcap Q(y)}Z_{uv}R_{vy} \tag{8}$$

Among them, $G\left(u,y\right)$ refers to the preference degree of breast cancer patients u for song y; $S\left(u,K\right)$ refers to the first K breast cancer patients with the most similar interest to breast cancer patients u; $Q\left(y\right)$ refers to the collection of breast cancer patients' behavior history for song y; Z_{uv} refers to the preference similarity between breast cancer patients u and v; R_{vy} refers to the preference score matrix of breast cancer patients for song j.

5.4.3. Item-CF based recommendation algorithm

Item-CF recommendation algorithm is to find similar items through interest items and recommend similar items to breast cancer patients. The similarity between songs is:

$$Z_{hy}=\frac{\left|Q\left(h\right)\bigcap Q\left(y\right)\right|}{\left|Q\left(h\right)\right|} \tag{9}$$

Among them, $\left|Q\left(h\right)\right|$ indicates the number of songs h loved by multiple breast cancer patients, and the molecule indicates the number of songs h and y loved by multiple breast cancer patients. After punishing popular music, the similarity is calculated as:

$$Z_{hy}=\frac{\left|Q\left(h\right)\right|\bigcap\left|Q\left(y\right)\right|}{\sqrt{\left|Q\left(h\right)\right|\left|Q\left(y\right)\right|}} \tag{10}$$

Formula (10) reduces the weight of song y and reduces the possibility that any song is similar to a popular song.

The Item-CF recommendation algorithm is to first establish an inverted list of breast cancer patients to songs to get the corresponding relationship between breast cancer patients and songs; secondly, a co-occurrence matrix is constructed through the inverted list, and the similarity between the two music is calculated according to Formula (10) to obtain the similarity matrix between each music; finally, the preference of breast cancer patients for songs is calculated. The formula for calculating the preference of breast cancer patients for songs is:

$$G\left(u,y\right)=\sum_{y\in S\left(h,K\right)\bigcap Q(u)}Z_{hy}R_{uy} \tag{11}$$

Among them, $Q\left(u\right)$ refers to the collection of songs loved by breast cancer patients u; $S\left(h,K\right)$ refers to the collection of the top K songs most similar to song h; Z_{hy} refers to the similarity between music h and music y; R_{uy} refers to the preference score of breast cancer patients u for music y.

D. Music recommendation algorithm based on weighted fusion of content and collaborative filtering.

The algorithm preference formula is:

$$G=\beta P_{u}+\left(1-\beta\right)P\left(u,y\right),\beta\in R,0\leq\beta\leq1 \tag{12}$$

Among them, β refers to the weight of breast cancer patients' preference matrix, and $(1-\beta)$ refers to the weight of breast cancer patients' preference for songs in the collaborative filtering algorithm. The lower the value of β, the greater the impact of similar preferences between breast cancer patients or items on recommendations. With the increase of the value of β, the impact of music content on recommendations is also increasing. The first K values are taken, and K types of music lists are obtained, which are recommended to corresponding breast cancer patients u.

6. Evaluation of experimental results of new music therapy in alleviating adverse reactions of chemotherapy in breast cancer patients

In order to better alleviate the adverse reactions of chemotherapy in breast cancer patients, this manuscript applies the neural network robot system to music training, forming a new music therapy. The following experiments are designed for the practical effect of the new music therapy. This manuscript first investigates the number of adverse reactions of breast cancer patients during chemotherapy in a large tumor hospital under the application of new music therapy. Among them, chemotherapy method 1 represents conventional treatment techniques, and chemotherapy method 2 represents new music therapy. The specific time is divided into 1, 2, 3, 4, and 5 months. The investigation results are shown in **Figure 4**.

It can be seen from the histogram in **Figure 4** that during the 5-month chemotherapy period, the number of adverse reactions of breast cancer patients using chemotherapy method 1 exceeded 10 times per month, and did not decrease gradually with the increase of chemotherapy time. In the application of chemotherapy method 2, although the number of adverse reactions was 10 or more 1 and 2 months later, the number of adverse reactions in the following 3 months was significantly reduced, and the overall trend was gradually reduced with the increase of chemotherapy time.

Anxiety and pain are the most common physical phenomena in breast cancer patients during chemotherapy. If the treatment can reduce the anxiety and pain of patients, then the treatment is undoubtedly successful. In order to test whether the new music therapy can reduce the anxiety and pain of patients, the anxiety scores and pain scores of breast cancer patients within 5 weeks were investigated using the new music therapy and conventional therapy. Among them, patients in Group A applied conventional therapy, while patients in Group B applied new music therapy. The higher the score, the stronger the sense of anxiety and pain. The specific findings are shown in **Figures 5**, **6**.

In the above histogram, the first represents the score of anxiety within 5 weeks, and the second represents the score of pain within 5 weeks. From the weekly score, with the increase of treatment time, the anxiety and pain of patients in Group A did not decrease significantly, and the score basically remained between 75 and 80. In contrast, although the anxiety and pain of patients in Group B were more than 70 points in the first week, they gradually decreased from the second week, and the scores were lower than 70 points. Obviously, with the increase of treatment time, the anxiety and pain of patients in Group B decreased, which indicates that the application of new music therapy is effective.

As a common malignant tumor, breast cancer also has a certain cure rate. This manuscript proposes that the main purpose of the new music therapy is to alleviate the adverse reactions of chemotherapy in breast cancer patients, but whether it can make a contribution to the cure rate of breast cancer still needs practice. In the case of music therapy and conventional therapy, the cure rate of breast cancer patients in a large tumor hospital within 48 courses of treatment was investigated. Every four courses of treatment were a stage. The investigation results are shown in **Figure 7**.

It can be concluded from the trend of the curve that the cure rate of breast cancer patients under conventional treatment did not increase gradually with the increase of the course of treatment, and the cure rate at each stage fluctuated greatly, but the overall cure rate was below 25%. The cure rate of breast cancer patients under music therapy was lower than that of conventional therapy in 4, 8, and 12 courses of treatment, because the application of new treatment methods is not very suitable. From the 16th course of treatment, the cure rate showed a steady rise, but the overall cure rate was also below 30%. In contrast, the overall cure rate of music therapy was significantly higher than that of conventional therapy, about 7.84% higher.

It is common for breast cancer patients to have adverse reactions during chemotherapy, and the relevant treatment methods must be able to alleviate them. The above surveys showed the effect of the new music therapy. In order to have a more comprehensive and objective understanding of the practical results of this treatment method, this manuscript investigated the satisfaction of patients, family members, therapists and hospital leaders in a large cancer hospital with the new music therapy. The specific samples were 150 patients, 150 family members, 150 therapists and 150 hospital leaders. The degree of satisfaction was divided into three levels: very satisfied, satisfied and dissatisfied. The survey results are shown in **Table 1**.

7. Conclusion

Breast cancer is a common malignant tumor disease among many cancers, and many patients are troubled by adverse reactions during chemotherapy. During routine chemotherapy, anxiety and pain have always been the "nightmare" that patients are difficult to get rid of. With the progress of science and technology and the continuous updating of therapeutic techniques, music therapy has gradually entered the field of chemotherapy. In this manuscript, the neural network robot system is applied to music relaxation training, forming a new type of music therapy, and it is used to alleviate the adverse reactions of chemotherapy in breast cancer patients. The research showed that the new music therapy is effective, and it also provided reference value for the development of disease diagnosis and examination technology in the future.

Data availability statement

The original contributions presented in this study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

YT and HY: writing – original draft preparation. YL and HY: editing data curation and supervision. All authors contributed to the article and approved the submitted version.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bradt, J., and Teague, A. (2018). Music interventions for dental anxiety. *Oral Dis.* 24, 300–306. doi: 10.1111/odi.12615

Fu, Y., Wei, X., Lin, L., Xu, W., and Liang, J. (2018). Adverse reactions of sorafenib, sunitinib, and imatinib in treating digestive system tumors. *Thorac. Cancer* 9, 542–547. doi: 10.1111/1759-7714.12608

Giavina-Bianchi, P., Patil, S., and Banerji, A. (2017). Immediate hypersensitivity reaction to chemotherapeutic agents. *J. Allergy Clin. Immunol.* 5, 593–599.

Golino, A., Leone, R., Gollenberg, A., Christopher, C., Stanger, D., Davis, T., et al. (2019). Impact of an active music therapy intervention on intensive care patients. *Am. J. Crit. Care* 28, 48–55. doi: 10.4037/ajcc2019792

Hubertus, S. (2017). Subjective experience of relaxation–induced by vibroacoustic stimulation by a Body Monochord or CD music–a randomised, controlled study in patients with psychosomatic disorders. *Nord. J. Music Ther.* 26, 79–98. doi: 10.1371/journal.pone.0170411

Jing, J. (2020). A study on the efficacy of high-quality nursing on alleviating adverse reactions and cancer pain, and its effect on QOL of patients with liver cancer after interventional surgery. *Int. J. Clin. Exp. Med.* 13, 925–932.

Liao, J., Wu, Y., Zhao, Y., Zhao, Y., Zhang, X., Zhao, N., et al. (2018). Progressive muscle relaxation combined with Chinese medicine five-element music on depression for cancer patients: A randomized controlled trial. *Chin. J. Integr. Med.* 24, 343–347. doi: 10.1007/s11655-017-2956-0

Marco, D. (2017). Cellular senescence promotes adverse effects of chemotherapy and cancer relapse cellular senescence and chemotherapy. *Cancer Discov.* 7, 165–176. doi: 10.1158/2159-8290.CD-16-0241

Najafi Ghezeljeh, T., Mohades Ardebili, F., and Rafii, F. (2017). The effects of massage and music on pain, anxiety and relaxation in burn patients: Randomized controlled clinical trial. *Burns* 43, 1034–1043. doi: 10.1016/j.burns.2017.01.011

Nelson, K., Adamek, M., and Kleiber, C. (2017). Relaxation training and postoperative music therapy for adolescents undergoing spinal fusion surgery. *Pain Manag. Nurs.* 18, 16–23. doi: 10.1016/j.pmn.2016.10.005

Pradop, S., Sinaredi, B. R., and Januarisca, B. V. (2017). Pandan Leaves (*Pandanus Amaryllifolius*) aromatherapy and relaxation music to reduce dental anxiety of pediatric patients. *J. Int. Dent. Med. Res.* 10, 933–937.

Qian, Y., Peng, J., Xiao, Y., Li, W., Tan, L., Xu, X., et al. (2018). Porous Au@ Pt nanoparticles: Therapeutic platform for tumor chemo-photothermal co-therapy and alleviating doxorubicin-induced oxidative damage. *ACS Appl. Mater. Interfaces* 10, 150–164. doi: 10.1021/acsami.7b14705

Singh, K., Bhori, M., Kasu, Y., Bhat, G., and Marar, T. (2018). Antioxidants as precision weapons in war against cancer chemotherapy induced toxicity–Exploring the armoury of obscurity. *Saudi Pharm. J.* 26, 177–190. doi: 10.1016/j.jsps.2017.12.013

Sopan, N., and Patil, P. (2022). Nanoarchitectured bioconjugates and bioreceptors mediated surface plasmon resonance biosensor for in vitro diagnosis of Alzheimer's disease: Development and future prospects. *Crit. Rev. Anal. Chem.* 52, 1139–1169. doi: 10.1080/10408347.2020.1864716

Vineeta, D. (2021). Effectiveness of ayurvedic treatment in alleviating side-effects of radiotherapy in patients suffering from oropharyngeal cancer and its relationship with im-provement in immune status of the host. *Open Access J. Sci. Technol.* 9, 16–20.

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 02 March 2023
DOI 10.3389/fnbot.2023.1132679

OPEN ACCESS

EDITED BY
Xiao Bai,
Beihang University, China

REVIEWED BY
Jianchu Lin,
Huaiyin Institute of Technology, China
Zhang Hengmin,
University of Macau, China
Zhilei Chai,
Jiangnan University, China

*CORRESPONDENCE
Chunhe Song
✉ songchunhe@sia.cn
Peng Zeng
✉ zp@sia.cn

RECEIVED 27 December 2022
ACCEPTED 13 February 2023
PUBLISHED 02 March 2023

CITATION
Wu T, Song C and Zeng P (2023) Model pruning based on filter similarity for edge device deployment. *Front. Neurorobot.* 17:1132679. doi: 10.3389/fnbot.2023.1132679

Model pruning based on filter similarity for edge device deployment

Tingting Wu[1,2,3,4], Chunhe Song[1,2,3]* and Peng Zeng[1,2,3]*

[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, [2]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang, China, [3]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, China, [4]University of Chinese Academy of Sciences, Beijing, China

Filter pruning is widely used for inference acceleration and compatibility with off-the-shelf hardware devices. Some filter pruning methods have proposed various criteria to approximate the importance of filters, and then sort the filters globally or locally to prune the redundant parameters. However, the current criterion-based methods have problems: (1) parameters with smaller criterion values for extracting edge features are easily ignored, and (2) there is a strong correlation between different criteria, resulting in similar pruning structures. In this article, we propose a novel simple but effective pruning method based on filter similarity, which is used to evaluate the similarity between filters instead of the importance of a single filter. The proposed method first calculates the similarity of the filters pairwise in one convolutional layer and then obtains the similarity distribution. Finally, the filters with high similarity to others are deleted from the distribution or set to zero. In addition, the proposed algorithm does not need to specify the pruning rate for each layer, and only needs to set the desired FLOPs or parameter reduction to obtain the final compression model. We also provide iterative pruning strategies for hard pruning and soft pruning to satisfy the tradeoff requirements of accuracy and memory in different scenarios. Extensive experiments on various representative benchmark datasets across different network architectures demonstrate the effectiveness of our proposed method. For example, on CIFAR10, the proposed algorithm achieves 61.1% FLOPs reduction by removing 58.3% of the parameters, with no loss in Top-1 accuracy on ResNet-56; and reduces 53.05% FLOPs on ResNet-50 with only 0.29% Top-1 accuracy degradation on ILSVRC-2012.

KEYWORDS

network acceleration, filter pruning, edge intelligence, network compression, convolutional neural networks

1. Introduction

Deep neural networks(DNNs) have become one of the most widely used algorithms in image classification (Krizhevsky et al., 2012), object detection (Ren et al., 2015), video analysis (Graves et al., 2013), and other fields with far surpassing accuracy than traditional algorithms. However, the high computing power and memory requirements of DNNs make it difficult for edge devices to deploy them with low latency, low power consumption, and high precision (Uddin and Nilsson, 2020; Veeramanikandan et al., 2020; Zhang et al., 2020; Fortino et al., 2021). To address this problem, various methods have been proposed for network compression and inference acceleration, including lightweight architecture design (Howard et al., 2017; Zhang X. et al., 2018), network pruning (LeCun et al., 1990; Hassibi and Stork, 1993; Li et al., 2016), weight quantization (Courbariaux et al., 2015; Hubara et al., 2017), matrix factorization (Denton et al., 2014), and knowledge distillation

(Hinton et al., 2015; Gou et al., 2021). Quantization compresses the model by reducing the size of the weights or activations. Matrix factorization is to approximate the large number of redundant filters of a layer using a linear combination of fewer filters. And knowledge distillation trains another simple network by using the output of a pre-trained complex network as a supervisory signal. Among them, network pruning compresses the existing network to reduce the requirements for space and computing power, to achieve real-time operation on portable devices. According to the granularity of pruning, network pruning methods can be divided into structured and unstructured pruning. Unstructured pruning requires specialized hardware and software for effective reasoning, and random connections will lead to poor cache locality and memory jump access, which makes acceleration very limited. Among structured pruning methods, filter pruning has received widespread attention because of its advantages of being directly compatible with current general-purpose hardware and highly efficient basic linear algebra subprogram (BLAS) libraries. The research in this paper belongs to the category of structured pruning, that is, the pruning granularity is at the level of convolution kernels.

Formally, for a CNN with weights of W and L convolutional layers, and N_i filters in each layer, determining which filter needs to be pruned is a combinatorial optimization problem, that can be expressed as follows (Zhou et al., 2019):

$$\begin{cases} \min_{\mathcal{M}} \mathcal{C}(\mathcal{D}; \mathcal{M} \circ W) \\ \min_{\mathcal{M}} \sum_{i=1}^{L} \|\mathcal{M}_i\|_1 \end{cases} \tag{1}$$

where $\mathcal{M}$ is the mask of the filter, and $\mathcal{C}$ is the cost function of the CNN on dataset $\mathcal{D}$. If there is a subset of convolution kernels such that the network can be pruned without performance degradation, it will be required to perform $2\sum_{i=1}^{L} N_i$ search and evaluation steps. For the current large network structure, this is an NP-hard problem, which is difficult to accurately solve by searching all possible subsets.

Among the simplest methods is the greedy method, or saliency-based method, which sorts weights by importance. The core problem is how to measure the importance of the filters. Recently a variety of filter pruning methods have been proposed to design more effective pruning guidelines. Hu et al. (2016) proposed using the average percentage of zero values (APoZ) to measure the importance of the activation value, which is defined as the proportion of zeroes in the activation values. Li et al. (2016) put forward a hypothesis based on the absolute value: the smaller the $l_1 - norm$ of the filter is, the less its influence on the final result. Molchanov et al. (2016) utilized the absolute value of the first-order term in the expansion of the objective function relative to the activation function as the criterion for pruning. Liu et al. (2017) introduced a channel scaling factor to the BN layer, added l_1 regularization to make it sparse, and then pruned the filters with a smaller scaling factor. He et al. (2019) developed a pruning method based on a geometric median to remove redundant filters.

Although the above works have achieved notable achievements, there are still many limitations: (1) Due to the different distributions of the values of the convolution kernels in different layers, the abovementioned pruning methods based on global or local criteria for sorting filters may ignore filters with smaller values in the sorting but extract edge features. Huang et al. (2020) compared different pruning standards and found that they have strong similarities, and that the importance of the obtained filters is almost the same, resulting in similar pruning structures. (2) Recent work (Liu et al., 2018) shows that the pruning structure is the key to determining the performance of the pruning model rather than the inheritance weight. Manually setting the pruning rate of each convolutional layer is equivalent to redesigning the network structure completely, and improper pruning rate settings will result in insufficient pruning or excessive pruning. In addition, for large networks, it is very expensive to accurately calculate the importance of the filters and set the pruning rate of each layer. (3) For special network structures such as residual blocks, most works only prune the channels of the middle layer of the block, which limits the space available for pruning. (4) The pruning process and the large number of fine-tuning required to restore the pruning performance lead to an excessively long pruning cycle, which is also the direction that needs to be optimized at present.

This paper focuses on the above problems and aims to improve the network performance under the same compression ratio. Therefore, we propose a channel pruning framework based on filter similarity, and optimize the pruning redundancy criterion, pruning strategy, pruning structure and pruning process, as shown in Figure 1. Specifically, in the pruning criteria, different from previous works which used precise rules to sort filters, we consider the problem from another perspective, focusing on the correlation of filters in one layer, and propose that two filters with high similarity extract similar features, and the extracted features can replace each other. In the pruning strategy, we do not need to specify the pruning rate of each layer, and automatically determine the pruning rate of each layer after determining the filter to be deleted according to the redundancy condition. In the pruning structure, we propose fine-grained pruning for special structures, in which the input and output channels of each block are calculated according to the redundancy condition constraints and then pruned in units of groups, thus increasing the reliability selection space for pruning channels. In addition, in the pruning process, for the situation that a lot of fine tuning is needed in the existing works, we perform a small amount of fine-tuning after each pruning of the whole network, which improves the efficiency of pruning. To summarize, our main contributions are as follows:

- We propose a novel method for estimating filter redundancy based on filter similarity, which does not rely on precise criteria to evaluate the importance of filters.
- The algorithm adaptively obtains the pruning rate of the layers according to the redundancy degree of each layer, which is difficult to determine in previous methods.
- The algorithm optimizes the channel pruning strategy of the special network structure, allowing the input and output channels of the residual block to be removed, further increasing the pruning space.
- The algorithm prunes the filters of the entire network at one time, and adopts two different pruning processes, hard pruning and soft pruning, which greatly reduces the large amount of fine-tuning caused by layer-by-layer pruning.

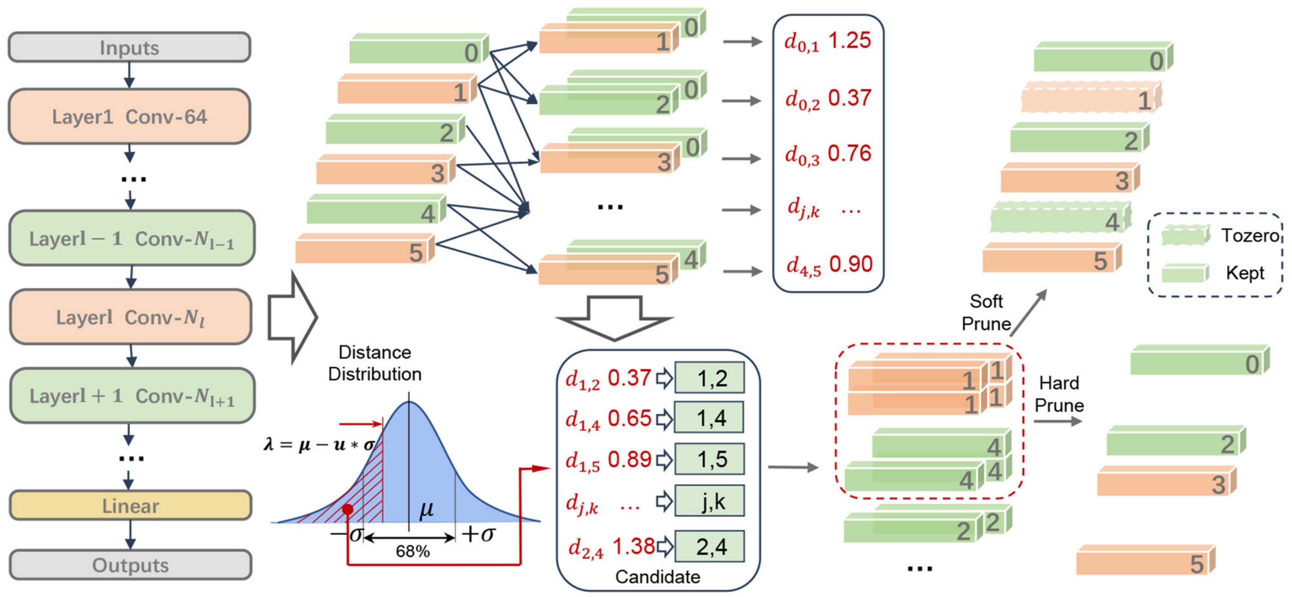


FIGURE 1
The pruning diagram of a convolutional layer. First, we compare all the filters in pairs, and then we count the distance set obtained, and put the filters corresponding to the distance value below the soft threshold into the set to be pruned. Finally, the filters with a higher frequency are considered redundant. The two methods of soft pruning and hard pruning are used to deal with redundant filters.

2. Related work

The typical work of network pruning is weight pruning and filter pruning. Weight pruning prunes individual parameter in the network to obtain a sparse weight matrix. Different from weight pruning, filter pruning removes the entire filter according to a certain measure. Filter pruning significantly reduces storage usage and decreases the computational cost of online inference. The key to filter pruning is the selection of filters, which should yield the highest compression ratio with the lowest compromise in accuracy. Based on the design of the filter importance criterion, we empirically divide the filter pruning into the following categories.

2.1. Based on magnitude

The simplest heuristic is to evaluate importance according to the absolute value of the parameter (or feature output) and then prune the part below the threshold by the greedy method, which is called amplitude-based weight pruning. Li et al. (2016) proposed using the absolute value of the weight as a measure of its importance (Zhang H. et al., 2018; Zhang et al., 2022). For structured pruning, group LASSO is often used to obtain structured sparse weights, such as in Liu et al. (2015) and Wen et al. (2016). Liu et al. (2017) introduced a channel scaling factor in the BN layer and pruned the corresponding weights with small scaling factors. In addition, the importance evaluation can also focus on the activation value. Hu et al. (2016) proposed using the average percentage of zero value (APoZ) to measure the importance of the activation value.

2.2. Based on loss function

The assumption based on absolute value judgment is that the smaller the absolute value of a parameter is, the smaller the influence on the final result. We call this the "smaller-norm/less-important" criterion, but this assumption is not necessarily true (as discussed in Ye et al., 2018). Another method is to consider the impact of parameter pruning on loss. LeCun et al. (1990) and Hassibi and Stork (1993) proposed the OBD and OBS methods, respectively, which measure the importance of weights in a network based on the second derivative of the loss function relative to the weight (the Hessian matrix for the weight vector). The method of Molchanov et al. (2016) was also based on Taylor expansion, but it utilized the absolute value of the first-order term in the expansion of the objective function relative to the activation function as the criterion for pruning. This avoids the calculation of second-order terms (i.e., the Hessian matrix). Lee et al. (2018) regarded the absolute value of the derivative of the normalized objective function with respect to the parameter as a measure of importance.

2.3. Based on the reconstructability of the feature output

The third method is to consider the impact on the rebuildability of the feature output, that is, minimizing the reconstruction error of the pruned network for the feature output. Typically, methods such as those of Luo et al. (2017) and He et al. (2017) identify channels that need to be pruned by minimizing feature reconstruction errors. Yu et al. (2018) proposed the NISP algorithm by minimizing the reconstruction error of the penultimate layer of the network, and back-propagating the importance information to the front to determine the channel to be pruned. Zhuang et al. (2018) proposed the DCP method. On the one hand, additional discriminative perception loss is added to the middle layer (to strengthen the discriminative ability of the middle layer), and on the other hand, the loss function of the error is also considered. The gradient information of the two losses is synthesized for the parameters, and the channels that need to be pruned are determined.

2.4. Other criteria

There are also other criteria based on the weights of the importance of ranking. He et al. (2019) proposed a filter pruning via geometric median (FPGM) method, the basic idea of which was to remove redundant parameters based on the geometric median. Lin et al. (2020) developed a method that was mathematically formulated to prune filters with low-rank feature maps. Wang et al. (2021) statistically modeled the network pruning problem in a redundancy reduction perspective and finded that pruning in the layer with the most structural redundancy outperforms pruning the least important filters across all layers. Cai et al. (2022) utilized a variant of the pruning mask as a prior gradient mask to guide fine-tuning. The disadvantage of the greedy algorithm is that it can only find local optimal solutions and ignores the relationship between the parameters. Some studies have aimed to consider the interrelationships among parameters to find a better global solution. Peng et al. (2019) proposed the CCP method, which considers the dependence between channels and formalizes the channel selection problem as a constrained quadratic programming problem. Wang et al. (2018) and Zhuo et al. (2018) used spectral clustering and subspace clustering to explore the relevant information in the channels and feature maps, respectively. With the development of AutoML research, such as AMC (He et al., 2018b), RNP (Lin et al., 2017), and N2N learning (Ashok et al., 2017), these tasks are all attempts to automate part of the pruning process.

3. Methodology

In this section, we introduce in detail the pruning algorithm based on the similarity of filters. The algorithm uses the similarity between the convolution filters in the convolutional layer to obtain network compression recommendations.

3.1. Motivation

Unlike current views of parameter importance-based pruning, we show that the removal of any one of the channels will not significantly impair the representational power of the network as long as there are two sufficiently similar channels. We derive theoretical support to justify the reasonability of our similarity-based pruning approach. Assuming that the neural network has L convolutional layers, N_l and N_{l+1} represent the number of input channels and output channels of the l_{th} layer convolution layer, respectively. $F^{(l,i)}$ represents the i_{th} filter of the l_{th} layer, and the corresponding input feature map can be expressed as $\mathcal{F}^{(l,i)} \in \mathbb{R}^{H\times W\times B}$, where H, W, B represent the height and width of the feature maps, and the batch size, respectively. The tensor of the connections of the l_{th} and $l+1_{th}$ layers can be parameterized by $\mathcal{W} \in \mathbb{R}^{N_l\times N_{l+1}\times K\times K}, 1 \leq l \leq L$.

Considering two consecutive convolutional layers and using non-linear activation $h(\bullet)$ after each linear convolution, then:

$$\mathcal{F}^{(l+1,n_{l+1})} = \sum_{n_l\in\{1,\ldots,N_l\}} h\left(\mathcal{F}^{(l,n_l)}\right) * \mathcal{W}^{(n_l,n_{l+1})} \tag{2}$$

where $\mathcal{W}^{(n_l,n_{l+1})} \in \mathbb{R}^{K\times K}$ is the n_l-dimensional weight of the n_{l+1}-th convolution kernel, corresponding to the n_{l+1}-th input feature map. We explore and analyze the loss of representational power brought about by removing one of two similar feature channels and its filter. Suppose that $\mathcal{F}^{(l,i)}$ and $\mathcal{F}^{(l,j)}$ are two similar channels, deleting the $\mathcal{F}^{(l,i)}$, then for the pruned $\mathcal{F}_p^{(l+1,n_{l+1})}$ we have:

$$\begin{aligned}\mathcal{F}_p^{(l+1,n_{l+1})} &= h\left(\mathcal{F}^{(l,j)}\right) * \left(\mathcal{W}^{(i,n_{l+1})} + \mathcal{W}^{(j,n_{l+1})}\right) \\ &\quad + \sum_{n_l\neq i,j} h\left(\mathcal{F}^{(l,n_l)}\right) * \mathcal{W}^{(n_l,n_{l+1})}\end{aligned} \tag{3}$$

We use mean squared error (MSE) to quantify the loss of the two feature maps before and after pruning:

$$\begin{aligned}&\mathcal{L}\left(\mathcal{F}^{(l+1,n_{l+1})}, \mathcal{F}_p^{(l+1,n_{l+1})}\right) \\ &= \left(H_{l+1} \times W_{l+1} \times B\right)^{-1} \times \left\|\mathcal{F}^{(l+1,n_{l+1})} - \mathcal{F}_p^{(l+1,n_{l+1})}\right\|_2^2 \\ &= \frac{1}{a_{l+1}} \left\|\left(h\left(\mathcal{F}^{(l,i)}\right) - h\left(\mathcal{F}^{(l,j)}\right)\right) * \mathcal{W}^{(i,n_{l+1})}\right\|_2^2\end{aligned} \tag{4}$$

where $a_{l+1} = H_{l+1} \times W_{l+1} \times B$. For each feature map $\mathcal{F}_p^{(l+1,n_{l+1})}$ in the $l+1$-th convolutional layer, the loss caused by removing the feature map $\mathcal{F}^{(l,i)}$ from the l-th convolutional layer, as defined in Equation (4), admits the following upper bound:

$$\mathcal{L}\left(\mathcal{F}^{(l+1,n_{l+1})}, \mathcal{F}_p^{(l+1,n_{l+1})}\right) \leq \varepsilon \times \min_{j\in\{1,\ldots,N_l\}} \mathcal{L}\left(\mathcal{F}^{(l,i)}, \mathcal{F}^{(l,j)}\right) \tag{5}$$

where $\varepsilon = \frac{a_l}{a_{l+1}} K^2 \left\|\mathcal{W}^{(i,n_{l+1})}\right\|_2^2$ and K^2 corresponds to the size of each filter $\mathcal{W}^{(n_l,n_{l+1})}$. Detailed derivation can be found in Appendix. We can conclude from Equation (5) that $\mathcal{E}$ is determined by the size of the feature maps, the $L2$-norm of the convolution kernel and its weights. In experiments, $\mathcal{E}$ is usually a value of the order of 10^{-2}, which means that the loss of removing one of the similar channels is negligible, as long as there are sufficiently similar channels to replace it.

In practice, our goal is to find similar channels and remove one of them. However, computing the similarity of channels directly has two apparent limitations. First, the activations of feature maps are affected differently by different batches of data. Second, calculating the similarity between all channels is inefficient for current large CNN architectures. To solve these issues, we use the convolution kernel as a unit for comparison. It can be seen from Equation (2) that when the input feature maps are the same, the feature maps obtained by similar convolution kernels are also identical, and the parameters of the kernels are not affected by the data batch.

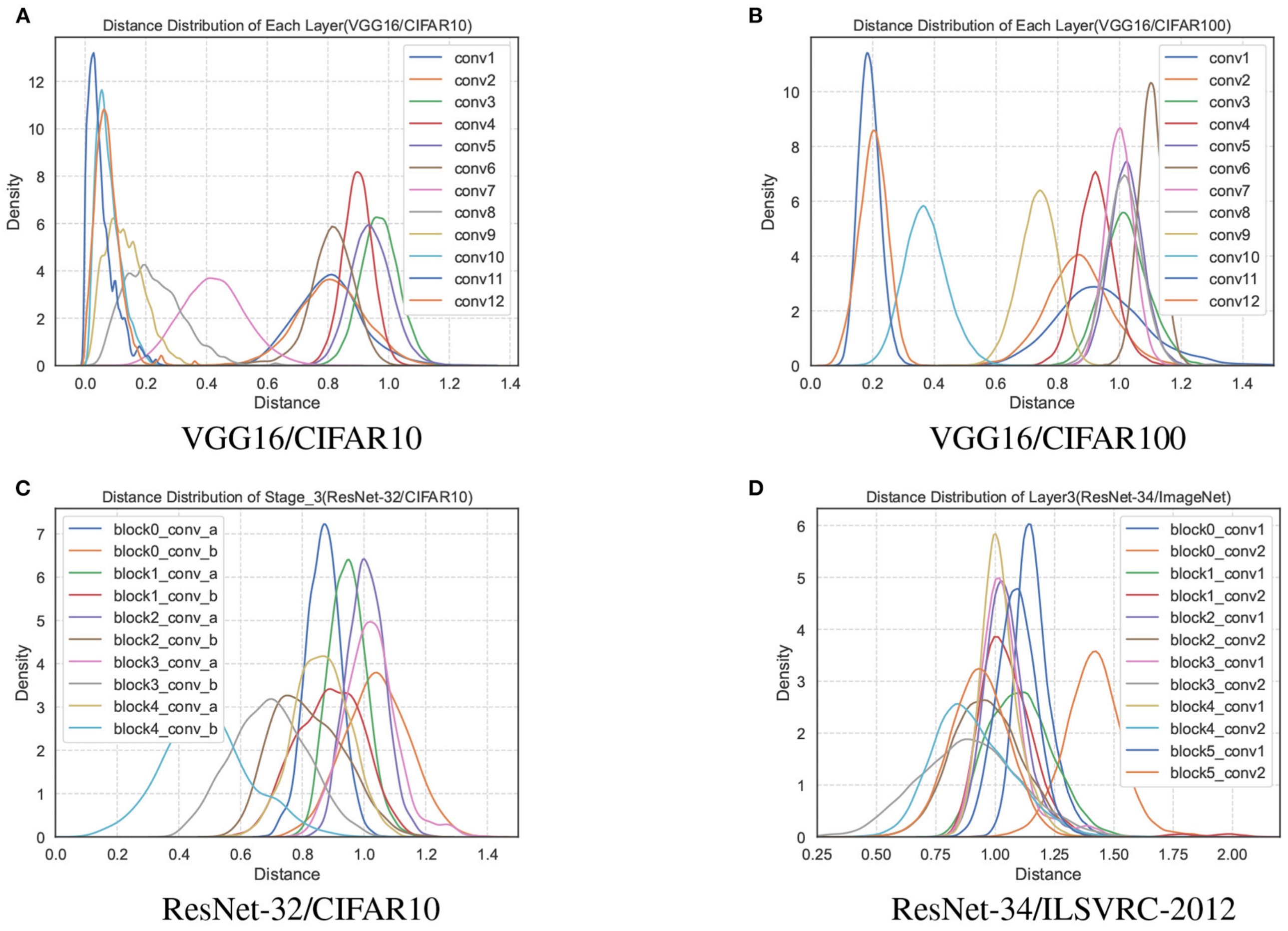


FIGURE 2
The distance distributions of each layer of the network model parameters trained by VGG16 on the CIFAR10 and CIFAR100 datasets are shown in **(A, B)**. **(C)** The distance distribution of all convolutional layers in the third stage of ResNet-32/CIFAR10. **(D)** The third layer of ResNet-34/ILSVRC-2012.

Intuitively, we quantify the similarity of two kernels by Euclidean distance, which is more commonly used in the analyses that need to reflect a difference in dimensions. In addition, Euclidean distance measures the distance between points in multidimensional space and can remember the absolute difference of characteristics. Therefore, for the l_{th} convolutional layer:

$$\begin{aligned} D^{(l)} &= dist\left(F_{l,j}, F_{l,k}\right), 0 \le j \le N_{l+1}, j \le k \le N_{l+1} \\ &= \left\{ \begin{matrix} d_{0,1} & d_{0,2} & \cdots & d_{0,N_{it1}-1} \\ & d_{1,2} & \cdots & d_{1,N_{it+1}-1} \\ & & & \vdots \\ & & \ddots & d_{j,k} \\ & & & \vdots \\ & & & d_{N_{i+1}-2,N_{i+1}-1} \end{matrix} \right\} \end{aligned} \qquad (6)$$

where

$$d_{j,k} = \sqrt{\sum_{n=1}^{N_l} \sum_{k_1}^{K} \sum_{k_2=1}^{K} \left| \mathcal{W}^j\left(n, k_1, k_2\right) - \mathcal{W}^k\left(n, k_1, k_2\right) \right|^2} \qquad (7)$$

$\mathcal{W}^j(n, k_1, k_2)$ is each weight in the filter $F^{(l,j)}$. For the l_{th} convolution layer, we obtain a set of distances $D^{(l)}$, which contains the distances between the j_{th} filter and all other filters. The smaller the distance is, the more significant the similarity between the two filters, indicating that the filter has extracted features similar to those of other filters.

We remove the repeated distance with the same subscript in $D^{(l)}$, and perform statistical analysis on all values in the set. Statistics show an interesting phenomenon that the distance distribution of each layer is an approximately Gaussian distribution in the trained network, as shown in Figure 2. The distance sets $D^{(l)}$ of different layers in the network are distributed differently, and the mean value even differs by an order of magnitude. However, the distance distribution between the filters has partial jitters since the convolutional layers, such as conv1 and conv2 of the VGG16, are affected by the input data.

3.2. Filter pruning based on similarity

After the distance distribution of each convolutional layer is obtained, how great can the distance between the two filters be determined to be similar? One of the methods is to get a minimum distance value $min[D^{(l)}]$ each time, that is, to remove one filter each time until the set requirement is reached. That is inefficient and laborious for network structures with thousands of convolution kernels. To obtain a set of redundant filters simultaneously, we first need to set a threshold λ, and a pair of filters corresponding to a distance less than this threshold are judged to be similar. Since the distance distribution of each convolutional layer is different, simply specifying the threshold of each layer will bring more hyperparameter problems. How can a reasonable threshold be set for each layer more efficiently with fewer hyperparameters?

Inspired by the empirical rule (3σ) of a Gaussian distribution, the probability of falling within $[\mu - \sigma, \mu + \sigma]$ is 0.68:

$$P^{(i)}(\mu - \sigma \leq x \leq \mu + \sigma) = 0.68, x \in D^{(l)} \tag{8}$$

we set a scaling factor α such that $\lambda = \mu - \alpha\sigma \in (-\infty, \mu]$, and then $\alpha \in [0, +\infty)$,

$$\begin{aligned} P^{(l)}\left(d_{j,k}^{(l)} \leq \lambda\right) &= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\lambda} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx \\ &\xlongequal{t=\frac{x-\mu}{\sigma}} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{\lambda-\mu}{\sigma}} \exp\left(-\frac{t^2}{2}\right) dt \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\alpha} \exp\left(-\frac{t^2}{2}\right) dt \\ &= \Phi(-\alpha) = 1 - \Phi(\alpha) \end{aligned} \tag{9}$$

where $\Phi(\bullet)$ is the distribution function of the standard normal distribution, it can be obtained from checking the Standard normal distribution table: when $\alpha = 0, \lambda = \mu, P = 0.5$; and $\alpha \to +\infty, \lambda \to -\infty, P \to 0$. If $d_{j,k}^{(l)} < \mu - \alpha^*\sigma$, the filters corresponding to $d_{j,k}^{(l)}$ in the shaded part of Figure 1 are judged to be similar, and then $d_{j,k}^{(l)}$ is selected as the candidate set $D_{select}^{(l)}$:

$$\begin{gathered} d_{j,k}^{(l)} \in D_{select}^{(l)} \\ j,k \in F_{select}^{(l)} \end{gathered} \tag{10}$$

$F_{select}^{(l)}$ is the set of indexes of the corresponding filters in $D_{select}^{(l)}$. We use a hyperparameter α to get equal-probability candidate sets in different layers for different distance distributions in each layer.

It can be seen in the experiment that a filter satisfies similar conditions simultaneously with multiple filters, but how can we determine the final deleted filters in the candidate set. For the l_{th} layer, we count the number of times of the j_{th} appears in $F_{select}^{(l)}$, denoted by $C_j^{(l)}$. Under extreme circumstances, if $d_{j,k}^{(l)} < \lambda \left(0 \leq k \leq N_{l+1} - 1, k \neq j\right)$ holds for the distance between the j_{th} filter and all other filters, then $C_j^{(l)} = N_{l+1} - 1$. We use the proportional factor $r \in [0, 1]$ to represent the frequency of the j_{th} filter,

$$r = \frac{C_j^{(l)}}{N_{l+1} - 1} \tag{11}$$

If $C_j^{(l)} > r^*(N_{l+1} - 1)$, then $j \in F_{pruned}^{(l)}$, $F_{pruned}^{(l)}$ is the set of final pruning filters. The above algorithm obtains a set of redundant filters for one convolution layer in the network structure, and the schematic diagram of the pruning process of each layer is shown in Figure 1.

3.3. Compression recipes

In addition to the judgment method of network parameter redundancy, the pruning strategy, implementation and network structure are also essential factors that affect the compression performance. As the pruning rate increases, network performance loss increases, and the redundant judgment of parameters is also prone to deviation when the network parameters deviate from the optimal point. Previous work uses layer-by-layer pruning and fine-tuning strategies or retraining to reduce the judgment error caused by performance loss and iterates this process until the target compression rate is achieved. However, when the iteration parameter setting is small and the target compression rate is significant, the pruning period will greatly increase, and the training time cost will be very high. Therefore, we prune all layers at once instead of layer-by-layer pruning and fine-tuning, significantly reducing the pruning cost. After complete pruning, the computation and parameter quantity of the whole network are calculated. If the set pruning requirements are met, the pruning is completed; otherwise, the redundant filters will continue to be searched for further pruning on the network structure of the last pruning until the set pruning requirements are met (computational cost reduction or parameter reduction), as shown in Algorithm 1.

In the implementation of pruning, He et al. (2018a) proposed not to directly delete the pruned parameters in the pruning process, which increases the fault tolerance of judgment. Many current works are based on soft pruning implementations, and for a fair comparison, we propose an iteration pruning strategy based on soft pruning. In the experiment, it is found that although the filters set to zero in the previous iteration are not deleted, they will not change in the subsequent fine-tuning no matter how the network

```
Require: Training dataset $\mathcal{D}$; the model with $\mathcal{W}$, and
    each layer with $\mathcal{W}^{(l)} \in \mathbb{R}^{N_l \times N_{l+1} \times K \times K}, 1 \leq l \leq L$; FLOPs
    or params pruning rate: $rate = rate_{FLOPs}/rate_{params}$.
Ensure: The pruned model $\mathcal{W}_{(\tau)}$
$\mathcal{W} \leftarrow train(\mathcal{W}, \mathcal{D})$
while $pruned\_rate = 0$ to $rate$ do
  for $i = 1$ to $L$ do
    for $j, k = 0$ to $N_{i+1} - 1$ do
      $D_{j,k}^{(i)} = dist(F_{i,j}, F_{i,k})$
      if $D_{j,k}^{(i)} < \mu - \alpha^*\sigma$ then
        $D_{j,k}^{(i)} \in D_{select}^{(i)}$
        $j, k \in F_{select}^{(i)}$
      end if
      if $C_j^{(i)} > r^*N_{i+1} - 1$ then
        $j \in F_{pruned}^{(i)}$
      end if
      $F^{(i)} \leftarrow F^{(i)} - F_{pruned}^{(i)}$
    end for
  end for
  $\mathcal{W} \leftarrow update\_params(\mathcal{W}, \mathcal{D})$
end while
$\mathcal{W} \leftarrow finetune(\mathcal{W}, \mathcal{D})$
```

Algorithm 1. Iterative pruning algorithm.

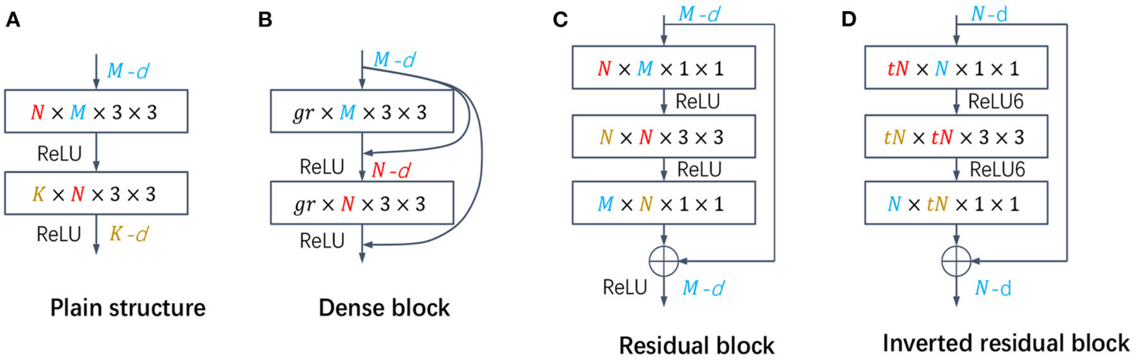


FIGURE 3
(A) The plain structure in VGG, and the number of output channel pruning in each layer is directly calculated by the algorithm. **(B)** The dense block, the number of output channels of each layer is fixed (*gr*), and the number of input channels is calculated by the algorithm. **(C, D)** Residual block and inverted residual block, respectively. The pruning rate of all layers in the block is the same, and the number of input channels and output channels of each block is guaranteed to be the same.

is updated, which affects the distance calculation and redundant judgment. To solve this problem, we set a mask for each filter, the pruned filters are 0, and the others are 1, and the mask is updated by the algorithm in real-time in each iteration. When calculating the distance between the filters in one layer, the distance will be multiplied by the mask value corresponding to the two filters at the same time,

$$\begin{aligned} d_{j,k}^{(l)} &= dist\left(F_{l,j}, F_{l,k}\right) * mask_j * mask_k \\ &= \begin{cases} 0, & mask_j = 0 \ or \ mask_k = 0 \\ dist\left(F_{l,j}, F_{l,k}\right), & mask_j \neq 0 \ and \ mask_k \neq 0 \end{cases} \end{aligned} \qquad (12)$$

In the distance set $D^{(l)}$, the distance $d_{j,k}^{(l)}$ between a filter with a mask of zero and any other filter is zero. Before the next step of obtaining the distance statistics, the algorithm ignores a value of zero for $d_{j,k}^{(l)}$, which is equivalent to allowing only the unpruned filters to participate in the subsequent pruning.

In pruning structure, some networks with special structures, such as ResNet and DenseNet, improve the efficiency and performance, but also make pruning more challenging. Only pruning the middle layer in the block is currently the most used strategy, but the filters between blocks are not easily pruned due to excessive constraints. We propose a more flexible pruning strategy, which is pruned in units of blocks, increasing the selection space of pruned filters under the guarantee rules. First, we calculate the pruning rate of the middle layers of all blocks in a group according to the filter redundancy determination algorithm proposed in the previous section, and then take the minimum value as the group's pruning rate $rate_{group}$. And then, the number of filters $card(F_{pruned}^{(l)})$ to be pruned at any l_{th} layer in the group can be obtained:

$$card\left(F_{pruned}^{(l)}\right) = rate_{group} * N_{l+1} \qquad (13)$$

For the l_{th} layer, $F_{selected}^{(l)}$ can be obtained by Equation (10), and the number of occurrences $C_j^{(l)}$ of the j_{th} filter in $F_{selected}^{(l)}$ can be sorted. The final pruned filters $F_{pruned}^{(l)}$ intercept the top $card(F_{pruned}^{(l)})$ filters from $F_{selected}^{(l)}$. The specific pruning mode of the different structures is shown in Figure 3.

The algorithm calculates the redundant filters of the whole network at one time instead of layer-by-layer, and then prunes or sets them to zero. The FLOPs and parameters reduction for the entire network is calculated after one iteration. If the set pruning rate is reached, the pruning is completed; otherwise, the parameters are updated to find more similar filters for further pruning. Then pruning is performed again until the set pruning rate is reached. After all pruning is completed, only a small amount of fine-tuning is required, as shown in Algorithm 1. In addition, we compare the current works with our proposed method from the aspects of criteria, whether to manually set the pruning rate of each layer, whether to process the residual structure, and the pruning method, as shown in Table 1. The proposed method optimizes and improves the pruning criterion, pruning rate setting, special structure processing, and pruning method.

4. Experiments

We evaluate the effectiveness of our algorithm on CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), and ILSVRC-2012 (Russakovsky et al., 2015) datasets using representative CNN architectures: VGGNet (Simonyan and Zisserman, 2014), ResNet (He et al., 2016), and DenseNet. CIFAR10 contains 50,000 training images and 10,000 testing images of size 32 × 32, which are categorized into 10 different classes. CIFAR100 is similar to CIFAR-10 but has 100 classes. ImageNet contains 1.28 million training images and 50 k validation images of 1,000 classes. VGGNet and ResNet represent two typical network structures with single branch and multiple branches respectively, and DenseNet prunes the input channels.

We calculate the size and computational complexity of the network through the number of network parameters and floating point operations (FLOPs) for one forward propagation. For the l_{th} convolutional layer,

$$\begin{aligned} FLOPs &= HW\left(C_{in}K^2 + 1\right)C_{out} \\ params &= \left(C_{in}K^2 + 1\right)C_{out} \end{aligned} \qquad (14)$$

TABLE 1 Comparison of our proposed method with current works.

Method	Criterion	Manually specify the pruning rate?	Residual structure processing?	Pruning method
L1	L1-norm	Yes	No	Hrad
Taylor	Taylor expansion	Yes	No	Hrad
ThiNet	Reconstruction error	Yes	No	Hrad
SFP	L2-norm	Yes	No	Soft
FPGM	Geometric median	Yes	No	Soft
HRank	Feature maps' average rank	Yes	No	Hrad
SRR-GR	L1-norm	No	No	Hrad
PGMPF	L2-norm	Yes	No	Soft
Ours	Filter similarity	No	Yes	Hrad/Soft

H and W are the length and width of the input feature map, respectively, and C_{in}, C_{out} are the number of input and output channels of the l_{th} convolutional layer, which correspond to the number of filters N_l and N_{l+1}.

We evaluate the performance of the convolution kernel pruning method by using the method of parameter quantity or the drop rate of computation, and different performance indicators can be used according to the requirements of different scenarios:

$$\begin{aligned} rate_{FLOPs} &= 1 - \frac{FLOPs_{original}}{FLOPs_{compressed}} \\ rate_{params} &= 1 - \frac{params_{original}}{params_{compressed}} \end{aligned} \tag{15}$$

Different pruning methods use pre-trained models or self-trained models as the baseline network. Due to the different training parameters (e.g., different learning rates, training times, data augmentations, etc.) and different experimental frameworks (TensorFlow, PyTorch, etc.), the Top-1 and Top-5 accuracies of the baseline network reported in the original papers are different. To make a fair comparison, we evaluate the effectiveness of pruning using the drop rate of Top-1 and Top-5 accuracy on the test set, which is the accuracy difference between the baseline network and the compressed network. Under the same compression rate, the smaller the difference, the better the pruning effect. All the comparison results in this paper are directly quoted from the original paper of the related method or the official code reproduction. All experiments are implemented on four NVIDIA TITAN Xp GPUs using PyTorch.

4.1. Results on the CIFAR-10/100 datasets

We analyze the performance on the CIFAR datasets with VGG16, DenseNet-40, and ResNet-32/56/110. All the networks are trained using SGD with Nesterov momentum (Sutskever et al., 2013) 0.9, a weight decay parameter of 10^{-4}, and an initial learning rate of 0.1. The learning rate is set to 0.001 when updating parameters or fine-tuning. For VGG16 and DenseNet-40, the baseline network is trained for 300 epochs with a batch size of 256. And for ResNet, the baseline network is trained for 200 epochs with a batch size of 256.

4.1.1. CIFAR10

We make a comparison with methods using hard pruning strategies, such as L1 (Li et al., 2016), the method of Molchanov et al. (2016), and with some current soft pruning methods, such as SFP (He et al., 2018a), FPGM (He et al., 2019), and HRank (Lin et al., 2020), and SRR-GR (Wang et al., 2021). In the VGG16/DenseNet experiment, α is set to 1, r is set to 0.35. And in ResNet, α is set to 1, r is set to 0.3. We adopt $rate_{FLOPs}$ as constraints and report $rate_{params}$ at the same time.

Results on CIFAR10 dataset are shown in Table 2. It can be observed that our proposed algorithm outperforms other methods under different networks and with similar or even higher compression ratios. In VGG16 with a plain structure, the performance of the similarity-based redundancy determination method far exceeds the other pre-defined determination methods, which indicates that the similarity-based determination method can effectively identify redundant parameters. On pruning strategy, soft pruning and hard pruning have little difference in performance under the same FLOPs pruning rate constraint. For example, at a pruning rate of 42.5%, the pruning performance of soft pruning is even worse than hard pruning. Moreover, there is little difference in performance between the evaluation criteria at a low pruning rate, but as the pruning rate increases, the judgment criteria have a more significant impact on the pruning performance.

In ResNet, the processing of the pruning structure and the pruning strategy also have an impact on the compression performance in addition to the criterion. The performance of hard pruning for L1 and ours is slightly worse than that of the soft pruning strategy. SFP uses the pruning principle with a small absolute value and does not prune the channels between the residual blocks, thus the performance is the worst. FPGM and HRank employ more effective criteria and a lot of fine-tuning, and the performance is improved. We achieve superior compression performance over existing work using a

TABLE 2 Comparison of the results of different network structures on the CIFAR10 dataset.

Model	Method	Prune	Top-1 (↓) (%)	FLOPs (↓) (%)	Params (↓) (%)
VGG16	L1 (Li et al., 2016)	✓	0.15	34.20	64.00
	Ours	✓	**0.17**	**42.47**	43.95
	L1 (Li et al., 2016)	✓	3.66	83.51	83.46
	(Molchanov et al., 2016)	✓	2.78	78.03	84.56
	Ours	✓	**1.74**	**81.62**	**82.33**
	FPGM (He et al., 2019)	✗	0.34	34.20	64.0
	Ours	✗	**0.17**	**42.48**	43.96
	HRank (Lin et al., 2020)	✗	2.73	76.50	92.0
	Ours	✗	**1.56**	**79.68**	81.64
	Ours	✗	**1.93**	**88.99**	**92.70**
ResNet-32	L1 (Li et al., 2016)	✓	11.81	43.76	44.72
	Ours	✓	**0.31**	43.47	43.61
	SFP (He et al., 2018a)	✗	0.55	41.50	–
	FPGM (He et al., 2019)	✗	0.70	53.2	–
	Ours	✗	**−0.29**	50.36	**55.71**
ResNet-56	L1 (Li et al., 2016)	✓	1.75	27.60	–
	SFP (He et al., 2018a)	✗	1.33	52.60	–
	FPGM (He et al., 2019)	✗	0.66	52.60	–
	HRank (Lin et al., 2020)	✗	0.09	50.00	42.40
	SRR-GR (Wang et al., 2021)	✗	−0.37	53.8	–
	Ours	✗	**−0.64**	**61.10**	**58.31**
ResNet-110	L1 (Li et al., 2016)	✓	0.61	38.60	–
	Ours	✓	1.65	**60.70**	**60.80**
	SFP (He et al., 2018a)	✗	0.30	40.80	–
	FPGM (He et al., 2019)	✗	−0.05	52.30	–
	HRank (Lin et al., 2020)	✗	0.85	68.60	42.40
	Ours	✗	**0.53**	**71.69**	**76.06**
DenseNet-40	HRank (Lin et al., 2020)	✗	0.57	40.80	36.5
	Ours	✗	**0.37**	**45.24**	**41.04**
	HRank (Lin et al., 2020)	✗	1.13	61.00	53.80
	Ours	✗	**0.90**	**62.22**	**62.02**

The "✓" indicates hard-pruning and "✗" indicates soft-pruning. The "(↓)" denotes the drop between baseline and the pruned model. A negative value in "Top-1(↓)(%)" indicates an improve model accuracy over the baseline model. The "-" denotes results are not reported in original papers. Other tables follow the same convention. The bold values indicate that experimental results are better than other methods.

similarity-based determination method and fewer fine-tuning epochs with the same soft-tuning implementation strategy. For DenseNet, where the input channels need to be pruned, we more effectively identify the redundant input channels while achieving excellent compression performance. Overall, soft pruning achieves higher pruning rates with similar accuracy than hard pruning. The criterion has a greater impact on the plain structure, in which the number of channels between layers is not constrained. The pruning performance of models with unique structures is affected by the judging criteria and the pruning strategy.

4.1.2. CIFAR100

The results on the CIFAR100 dataset are shown in Table 3. Compared to the CIFAR10 dataset, CIFAR100 is more challenging for pruning due to more categories. We compare with L1 (Li et al., 2016), the method of Molchanov et al. (2016), SFP (He et al., 2018a), FPGM (He et al., 2019), and PGMPF (Cai et al., 2022) on VGG16 and ResNet32/56/110. In the VGG16 experiment, α is set to 1, r is set to 0.35, and in ResNet, α is set to 1, r is set to 0.3. All the data in the table are obtained under the same number of fine-tuning according to the public code. The parameters not given in the table are because the code or the paper does not give the

TABLE 3 Comparison of pruned ResNet on CIFAR100.

Depth	Method	Prune	Top-1 (↓) (%)	Top-5 (↓) acc (%)	FLOPs (↓) (%)	Params (↓) (%)
VGG16	L1 (Li et al., 2016)	✓	2.24	1.27	50.44	50.23
	(Molchanov et al., 2016)	✓	2.36	1.42	40.25	47.36
	Ours	✓	**1.69**	**1.72**	**51.99**	**68.79**
	FPGM (He et al., 2019)	✗	2.06	1.73	48.93	–
	PGMPF (Cai et al., 2022)	✗	0.35	–	48.20	–
	Ours	✗	**0.34**	**1.25**	**52.80**	**62.97**
ResNet-32	L1 (Li et al., 2016)	✓	18.37	11.47	43.76	44.16
	Ours	✓	**2.74**	**1.73**	43.45	43.38
	SFP (He et al., 2018a)	✗	2.21	1.12	53.16	–
	FPGM (He et al., 2019)	✗	0.16	-0.63	53.16	–
	Ours	✗	**−0.59**	−0.07	50.51	**53.25**
ResNet-56	SFP (He et al., 2018a)	✗	1.05	−0.16	63.16	–
	FPGM (He et al., 2019)	✗	1.33	−0.10	63.16	–
	PGMPF (Cai et al., 2022)	✗	2.71	–	52.6	–
	Ours	✗	**0.71**	1.03	**64.98**	**61.45**
ResNet-110	**Ours**	✗	0.98	0.65	59.23	56.70

The bold values indicate that experimental results are better than other methods.

specific calculation process. It can be observed that our method still outperforms other existing methods when reaching similar or higher pruning rates. Compared with the CIFAR10 dataset, the gap between different judgment criteria methods is more prominent, and even the accuracy gain brought by the increased number of fine-tuning still cannot compensate for the performance loss of the network due to inaccurate pruning. For example, SFP reduces the accuracy by 2.21% under half the FLOPs compression on ResNet32. FPGM still has an accuracy loss of 0.16% with the increased number of fine-tuning. However, the accuracy of our method has not decreased but increased, which can reflect the differences between different evaluation criteria. At the same time, the network is more sensitive to pruning on larger datasets, and the redundancy of the network does not increase with the depth of the network, which brings more difficulty to the judgment of parameter redundancy. For ResNet110, while the pruning rate is reduced compared to ResNet56, the network performance also drops significantly.

4.2. Results on ILSVRC-2012

In the experiments, we use ResNet-18/34/50 to demonstrate the proposed pruning performance on a large-scale dataset, ILSVRC-2012 (Russakovsky et al., 2015). All the baseline networks are obtained by training 100 epochs with a batch size of 256. We follow the same parameter settings as [16] and [56], where the hyperparameter α is set to 1 and r is set to 0.35. We compare the proposed method with ThiNet (Luo et al., 2018), FPGM (He et al., 2019), MIL (Dong et al., 2017), PFEC (Li et al., 2016), CP (He et al., 2017), SFP (He et al., 2018a), HRank (Lin et al., 2020), PGMPF (Cai et al., 2022), and SRR-GR (Wang et al., 2021) and present the results in Table 4. All the results of the other methods in the table are directly from their reports in the literature. For ResNet with different depths, the hard pruning and soft pruning strategies are tested to make a fair comparison with other methods of different implementations. From the previous experiments on the CIFAR10/100 datasets, we conclude that the network performance is more sensitive to pruning in underfitted network structures. For ResNet18/34, our algorithm achieves the same FLOPs drop rate under the hard pruning strategy and achieves a smaller Top-1 accuracy drop rate than other methods using soft pruning strategies; in soft pruning, a better performance is still obtained with more pruned FLOPs than other methods. For ResNet50, the performance of the pruning algorithms is not very different, but our algorithm still achieves a better performance. For example, it reduces the computation by nearly half (53.05%), while the Top-1 accuracy loss is only 0.29%. Similarly, the final performance of the soft pruning strategy is still significantly better than that of the hard pruning strategy.

4.3. Ablation study

4.3.1. Influence of hyperparameters

There are two hyperparameters α and r in the algorithm proposed in this paper. These two parameters together determine the pruning rate of each layer. From the introduction of the algorithm in Section 3, we only need to specify a set of α and r values for each network structure, to avoid manually specifying the pruning rate of each layer in the network. Next, we will discuss how to select the hyperparameters in the experiment and how their values affect the pruning rate. To explore the relationship more clearly, we choose to use the VGG16 to experiment on the CIFAR100 datasets.

TABLE 4 Comparison of pruned ResNet on ILSVRC-2012.

Model/ Data	Method	P.F.	Base top-1 acc(%)	Pruned top-1 acc(%)	Top-1 (↓)(%)	Base top-5 acc(%)	Pruned top-5 acc(%)	Top-5 (↓)(%)	FLOPs (↓)(%)
ResNet18	MIL (Dong et al., 2017)	✓	69.98	66.33	3.65	86.94	89.24	2.30	34.6
	Ours	✓	**70.48**	**68.58**	**1.90**	**89.60**	**88.44**	**1.16**	**50.1**
	SFP (He et al., 2018a)	✗	70.28	67.10	3.18	89.63	87.78	1.85	41.8
	FPGM (He et al., 2019)	✗	70.28	67.81	2.47	89.63	88.11	1.52	41.8
	PGMPF (Cai et al., 2022)	✗	70.23	66.67	3.56	89.51	87.36	2.15	53.5
	Ours	✗	**70.48**	**68.96**	**1.52**	**89.60**	**88.55**	**1.05**	**52.85**
ResNet34	MIL (Dong et al., 2017)	✓	73.42	72.99	0.43	91.36	91.19	0.17	24.8
	PFEC (Li et al., 2016)	✓	73.23	72.17	1.06	-	-	-	24.2
	Ours	✓	**73.90**	**72.30**	**1.60**	**91.59**	**90.79**	**0.80**	**53.1**
	SFP (He et al., 2018a)	✗	73.92	71.83	2.09	91.62	90.33	1.29	41.1
	FPGM (He et al., 2019)	✗	73.92	72.11	1.81	91.62	90.69	0.93	41.1
	PGMPF (Cai et al., 2022)	✗	73.27	70.64	2.63	91.43	89.87	1.56	52.7
	Ours	✗	**73.90**	**72.80**	**1.10**	**91.59**	**91.04**	**0.55**	**52.07**
ResNet50	ThiNet (Luo et al., 2018)	✓	75.30	74.03	1.27	92.20	92.11	0.09	36.79
	CP (He et al., 2017)	✓	-	-	-	92.20	90.80	1.40	50.0
	Ours	✓	**75.82**	**74.74**	**1.08**	**92.95**	**92.28**	**0.67**	**40.78**
	SFP (He et al., 2018a)	✗	76.15	74.61	1.54	92.87	92.06	0.81	41.8
	FPGM (He et al., 2019)	✗	76.15	75.03	1.12	92.87	92.40	0.47	42.2
	HRank (Lin et al., 2020)	✗	76.15	74.98	1.17	92.87	92.33	0.54	43.76
	SRR-GR (Wang et al., 2021)	✗	76.13	75.76	0.37	92.86	92.60	0.19	44.10
	PGMPF (Cai et al., 2022)	✗	76.01	75.11	0.90	92.93	92.41	0.52	53.5
	Ours	✗	**75.82**	**75.53**	**0.29**	**92.95**	**92.83**	**0.12**	**53.05**

The bold values indicate that experimental results are better than other methods.

For different α values, the algorithm can obtain different candidate sets. This value determines how large the distance value of two filters should be if they will be selected to be pruned. The larger α is, the more filters are finally pruned. For a fixed value of r, the pruning rate of different layers obtained by different α is shown in Figure 4A. For different r values, different sets of final pruned channels can be obtained. For a convolution kernel, r determines how many other convolution kernels it is similar to, and it is regarded as a redundant convolution kernel. The larger r is, the fewer pruned filters are obtained. For a fixed α, the pruning rate of different layers obtained by different r values are shown in Figure 4B.

It can be inferred from the above figures that the values of r and α are correlated roughly linearly with the final pruning rate. These two hyperparameters together determine the pruning rate of each layer. According to the rules obtained from the experiments in the figure, we can take the appropriate r and α for different networks in later experiments. The algorithm does not need to precisely specify the exact values of r and α. Excellent experimental performance can be obtained when α is between 0.8 and 1.1 and r between 0.25 and 0.4, and the settings of α and r have a certain influence on the number of iterations. Once they are set, there is no need to specify the pruning rate of each layer, and the algorithm directly derives the filters that need to be pruned for each layer.

4.3.2. Pruning rate change during iteration

The proposed algorithm determines the pruning rate of each layer adaptively without manual specification. After setting the FLOPs or parameters constraints, the algorithm automatically prunes the redundancies in each layer and calculates the FLOPs and parameters after each iteration. After several iterations, the set target is reached, and pruning is completed, thereby avoiding layer-by-layer pruning and much fine-tuning. As shown in Figure 4, pruning becomes increasingly difficult with increasing numbers of iterations, and the network performance becomes increasingly sensitive to pruning. In the last few iterations, only a small number of filters are pruned, which results in a significant decrease in accuracy. For different datasets, the redundancy of each convolutional layer for the same network structure is different. On the CIFAR10 datasets, the redundancy of the first few convolutional layers is higher, and the pruning rate is between 50 and 80%. However, the pruning rates of the first few layers on the CIFAR100 datasets are all below 40%.

4.3.3. Feature map visualization and actual speedup

To verify whether the filters identified by our proposed algorithm are truly redundant, we visualize the first layer of

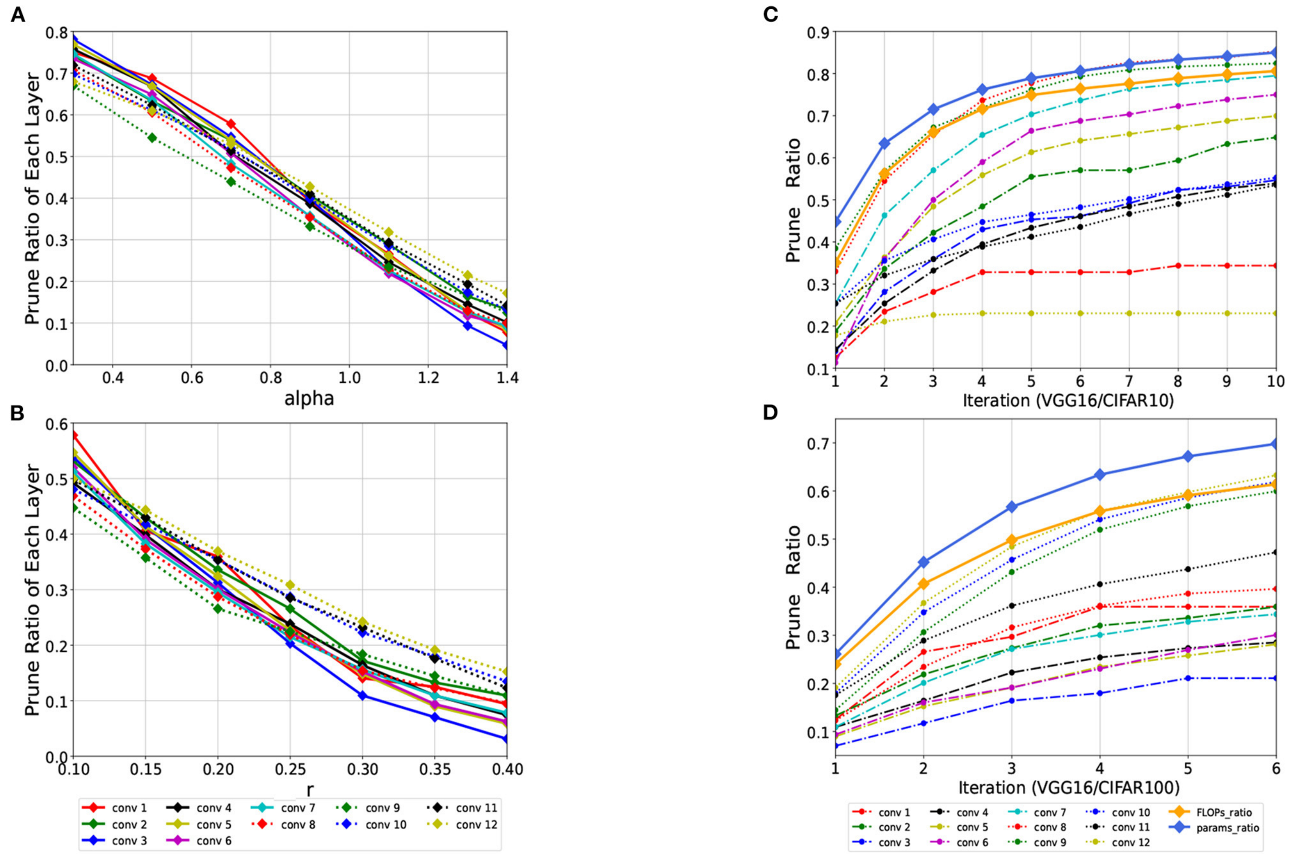


FIGURE 4
(A) The pruning rate of each convolutional layer with different values of α for the VGG16 on the CIFAR100 datasets at $r = 0.35$. **(B)** The pruning rate of different layers with different values of r when $\alpha = 1$. **(C, D)** The pruning process of the VGG16 on the CIFAR10/100 datasets when $rate_{FLOPs}$ is set to 0.8 and 0.6, respectively. On CIFAR10, the algorithm meets the pruning requirement after 10 iterations, while CIFAR100 exceeds 6 iterations. The pruning rate of each layer and the change process of $rate_{FLOPs}$ and $rate_{params}$ are also shown in the figure.

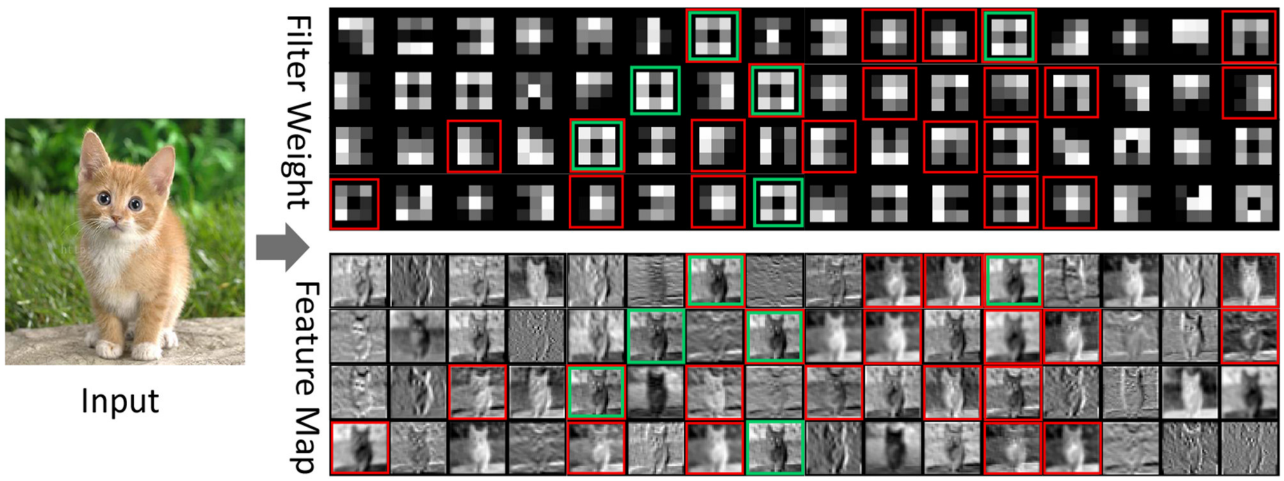


FIGURE 5
Visualization of the weights and feature maps of the first convolutional layer of the VGG16 on the CIFAR100 datasets. The first convolutional layer has 64 filters, and the filters with red bounding boxes are to be pruned. The green boxes are similar channels artificially determined according to the feature map and the convolution kernel, while the green boxes and red boxes are the channels identified and pruned by the algorithm.

TABLE 5 Speedups of compressed network models on different datasets.

Model/Dataset		Original time (ms)	Pruned time (ms)	Speedup
CIFAR10	VGG16 (11.01%)	16.92	4.69	3.61 ×
	ResNet-32 (49.64%)	6.78	3.83	1.77 ×
	ResNet-56 (38.90%)	9.60	4.15	2.31 ×
	ResNet-110 (23.94%)	15.12	4.28	3.53 ×
	DenseNet-40 (37.98%)	23.26	9.78	2.38 ×
ImageNet	ResNet18 (47.15%)	45.15	23.39	1.93 ×
	ResNet34 (47.93%)	73.82	45.38	1.63 ×
	ResNet50 (46.95%)	165.31	97.61	1.69 ×

the convolution kernel and the corresponding feature map of the VGG16 on the CIFAR100 datasets. The part marked in red in the figure contains the pruned filters and the corresponding feature maps. We analyze the filters and the corresponding feature maps and find that there are multiple similar filters in the same convolution layer, and their corresponding feature maps are also quite similar. For example, comparing their weights and feature maps, the filters (7, 12, 22, 24, 37, 56) all extract the overall outline of the cat. Our algorithm prunes the filters (7, 12, 24, 37) and keeps the other two similar filters (22, 56), as shown in Figure 5.

We evaluate the actual speedup of our proposed method on the intelligent edge accelerator Jeston nano, as shown in Table 5. Since previous works used different GPUs and libraries, and pruned models are not readily available, we only report the inference time and speedup of the original model and the pruned model using our proposed method. It can be seen from the table that on edge devices, the inference speed of our proposed compression model is faster than that of the original model, but the actual speedup ratio cannot reach the theoretical reduction of calculation. The actual acceleration ratio of VGG is much smaller than the theoretical acceleration ratio, while the acceleration ratio of ResNet and DenseNet is comparable to the theoretical acceleration ratio. We believe that the gap between theoretical and actual speedup is mainly caused by the cache effect and memory accessing pattern in GPU, which is affected by the hardware itself, the network architecture, and Pytorch library implementation.

5. Conclusion

In this article, we propose a novel strategy for judging the redundancy of filters based on similarity. To obtain the redundant filters, we analyze the similarity distribution law for filters in a convolution layer, and obtain a compact network by pruning the redundant filters with certain strategies. A large number of experiments proved the effectiveness and flexibility of the method under the same experimental parameters and the performance does not depend on a large number of fine-tunings.

Although the pruning method we proposed does not need to specify the pruning rate of each layer, it still relies on the values of two hyperparameters. If different hyperparameters are specified for each layer according to the redundancy of each layer, the network will be further compressed. We plan to combine this method with reinforcement learning to automatically adjust the required parameters and improve the performance to a higher level. We performed simple statistical tests on similar filters to provide a basis for further pruning, which is far insufficient for complex CNNs. We will further analyze the filters' similarity data and combine the visual analysis of each layer to provide guidance for pruning.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

Author contributions

TW: writing—original draft. CS and PZ: writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by National Key R&D Program of China (2017YFA0700300), Nature Science Foundation of China (62273337), Nature Science Foundation of Liaoning Province (2021-MS-030), and Independent Project of State Key Laboratory of Robotics (2022-Z03).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Ashok, A., Rhinehart, N., Beainy, F., and Kitani, K. M. (2017). N2n learning: Network to network compression *via* policy gradient reinforcement learning. *arXiv preprint arXiv:1709.06030*.doi: 10.48550/arXiv.1709.06030

Cai, L., An, Z., Yang, C., Yan, Y., and Xu, Y. (2022). "Prior gradient mask guided pruning-aware fine-tuning," in *Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 1* (Vancouver, CA). doi: 10.1609/aaai.v36i1.19888

Courbariaux, M., Bengio, Y., and David, J.-P. (2015). "Binaryconnect: training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems* (Montreal, QC), 3123–3131.

Denton, E. L., Zaremba, W., Bruna, J., LeCun, Y., and Fergus, R. (2014). "Exploiting linear structure within convolutional networks for efficient evaluation," in *Advances in Neural Information Processing Systems* (Montreal, QC), 1269–1277.

Dong, X., Huang, J., Yang, Y., and Yan, S. (2017). "More is less: a more complicated network with less inference complexity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 5840–5848. doi: 10.1109/CVPR.2017.205

Fortino, G., Zhou, M., Hassan, M. M., Pathan, M., and Karnouskos, S. (2021). Pushing artificial intelligence to the edge: emerging trends, issues and challenges. *Eng. Appl. Artif. Intell.* 103:104298. doi: 10.1016/j.engappai.2021.104298

Gou, J., Yu, B., Maybank, S. J., and Tao, D. (2021). Knowledge distillation: a survey. *Int. J. Comput. Vis.* 129, 1789–1819. doi: 10.1007/s11263-021-01453-z

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (Vancouver, CA), 6645–6649. doi: 10.1109/ICASSP.2013.6638947

Hassibi, B., and Stork, D. G. (1993). "Second order derivatives for network pruning: optimal brain surgeon," in *Advances in Neural Information Processing Systems* (San Francisco), 164–171.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 770–778. doi: 10.1109/CVPR.2016.90

He, Y., Kang, G., Dong, X., Fu, Y., and Yang, Y. (2018a). Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*. doi: 10.24963/ijcai.2018/309

He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J., and Han, S. (2018b). "AMC: Automl for model compression and acceleration on mobile devices," in *Proceedings of the European Conference on Computer Vision* (Munich), 784–800. doi: 10.1007/978-3-030-01234-2_48

He, Y., Liu, P., Wang, Z., Hu, Z., and Yang, Y. (2019). "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 4340–4349. doi: 10.1109/CVPR.2019.00447

He, Y., Zhang, X., and Sun, J. (2017). "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 1389–1397. doi: 10.1109/ICCV.2017.155

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*. doi: 10.48550/arXiv.1503.02531

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*. doi: 10.48550/arXiv.1704.04861

Hu, H., Peng, R., Tai, Y.-W., and Tang, C.-K. (2016). Network trimming: a data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*. doi: 10.48550/arXiv.1607.03250

Huang, Z., Wang, X., and Luo, P. (2020). Convolution-weight-distribution assumption: rethinking the criteria of channel pruning. *arXiv preprint arXiv:2004.11627*. doi: 10.48550/arXiv.2004.11627

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. (2017). Quantized neural networks: training neural networks with low precision weights and activations. *J. Mach. Learn. Res.* 18, 6869–6898. doi: 10.48550/arXiv.1609.07061

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inform. Process. Syst.* 25, 1097–1105. doi: 10.1145/3065386

LeCun, Y., Denker, J. S., and Solla, S. A. (1990). "Optimal brain damage," in *Advances in Neural Information Processing Systems* (Denver, CO), 598–605.

Lee, N., Ajanthan, T., and Torr, P. H. (2018). SNIP: single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*. doi: 10.48550/arXiv.1810.02340

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. (2016). Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*. doi: 10.48550/arXiv.1608.08710

Lin, J., Rao, Y., Lu, J., and Zhou, J. (2017). "Runtime neural pruning," in *Advances in Neural Information Processing Systems* (Long Beach, CA), 2181–2191.

Lin, M., Ji, R., Wang, Y., Zhang, Y., Zhang, B., Tian, Y., et al. (2020). "Hrank: filter pruning using high-rank feature map," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle), 1529–1538. doi: 10.1109/CVPR42600.2020.00160

Liu, B., Wang, M., Foroosh, H., Tappen, M., and Pensky, M. (2015). "Sparse convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 806–814.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. (2017). "Learning efficient convolutional networks through network slimming," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 2736–2744. doi: 10.1109/ICCV.2017.298

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. (2018). Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*. doi: 10.48550/arXiv.1810.05270

Luo, J.-H., Wu, J., and Lin, W. (2017). "Thinet: a filter level pruning method for deep neural network compression," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice), 5058–5066. doi: 10.1109/ICCV.2017.541

Luo, J.-H., Zhang, H., Zhou, H.-Y., Xie, C.-W., Wu, J., and Lin, W. (2018). Thinet: pruning CNN filters for a thinner net. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 2525–2538. doi: 10.1109/TPAMI.2018.2858232

Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. (2016). Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*. doi: 10.48550/arXiv.1611.06440

Peng, H., Wu, J., Chen, S., and Huang, J. (2019). "Collaborative channel pruning for deep networks," in *International Conference on Machine Learning* (Long Beach, CA), 5113–5122.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*. doi: 10.1109/TPAMI.2016.2577031

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 211–252. doi: 10.1007/s11263-015-0816-y

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. doi: 10.48550/arXiv.1409.1556

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning* (Atlanta, GA), 1139–1147.

Uddin, M. Z., and Nilsson, E. G. (2020). Emotion recognition using speech and neural structured learning to facilitate edge intelligence. *Eng. Appl. Artif. Intell.* 94:103775. doi: 10.1016/j.engappai.2020.103775

Veeramanikandan, Sankaranarayanan, S., Rodrigues, J. J., Sugumaran, V., and Kozlov, S. (2020). Data flow and distributed deep neural network based low latency IoT-edge computation model for big data environment. *Eng. Appl. Artif. Intell.* 94:103785. doi: 10.1016/j.engappai.2020.103785

Wang, D., Zhou, L., Zhang, X., Bai, X., and Zhou, J. (2018). Exploring linear relationship in feature map subspace for convnets compression. *arXiv preprint arXiv:1803.05729*. doi: 10.48550/arXiv.1803.05729

Wang, Z., Li, C., and Wang, X. (2021). "Convolutional neural network pruning with structural redundancy reduction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (virtual), 14913–14922. doi: 10.1109/CVPR46437.2021.01467

Wen, W., Wu, C., Wang, Y., Chen, Y., and Li, H. (2016). "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems* (Barcelona), 2074–2082.

Ye, J., Lu, X., Lin, Z., and Wang, J. Z. (2018). Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers. *arXiv preprint arXiv:1802.00124*. doi: 10.48550/arXiv.1802.00124

Yu, R., Li, A., Chen, C.-F., Lai, J.-H., Morariu, V. I., Han, X., et al. (2018). "NISP: pruning networks using neuron importance score propagation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 9194–9203. doi: 10.1109/CVPR.2018.00958

Zhang, H., Qian, F., Zhang, B., Du, W., Qian, J., and Yang, J. (2022). Incorporating linear regression problems into an adaptive framework with feasible optimizations. *IEEE Trans. Multim.* doi: 10.1109/TMM.2022.3171088

Zhang, H., Yang, J., Shang, F., Gong, C., and Zhang, Z. (2018). LRR for subspace segmentation via tractable schatten-*p* norm minimization and factorization. *IEEE Trans. Cybern.* 49, 1722–1734. doi: 10.1109/TCYB.2018.2811764

Zhang, P., Zhang, A., and Xu, G. (2020). Optimized task distribution based on task requirements and time delay in edge computing environments. *Eng. Appl. Artif. Intell.* 94:103774. doi: 10.1016/j.engappai.2020.103774

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). "ShuffleNet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT), 6848–6856. doi: 10.1109/CVPR.2018.00716

Zhou, Y., Yen, G. G., and Yi, Z. (2019). A knee-guided evolutionary algorithm for compressing deep neural networks. *IEEE Trans. Cybern*. 51, 1626–1638. doi: 10.1109/TCYB.2019.2928174

Zhuang, Z., Tan, M., Zhuang, B., Liu, J., Guo, Y., Wu, Q., et al. (2018). "Discrimination-aware channel pruning for deep neural networks," in *Advances in Neural Information Processing Systems* (Montréal, QC), 875–886.

Zhuo, H., Qian, X., Fu, Y., Yang, H., and Xue, X. (2018). SCSP: spectral clustering filter pruning with soft self-adaption manners. *arXiv preprint arXiv:1806.05320*.

Appendix

Proof of the equation 5

For the i_{th} feature map $\mathcal{F}^{(}l,i)$ of the l_{th} layer, let $\delta_{i,j}^{(l)} = h\left(\mathcal{F}^{(l,i)}\right) - h\left(\mathcal{F}^{(l,j)}\right)$ and $\delta_{i,j}^{(l)}\left(p_1,p_2\right)$ denote the image block centered at (p_1,p_2), then we have:

$$\begin{aligned}
&\mathcal{L}\left(\mathcal{F}^{(l+1,n_{l+1})},\mathcal{F}_p^{(l+1,n_{l+1})}\right)\\
&=\frac{1}{a_{l+1}}\left\|\left(h\left(\mathcal{F}^{(l,i)}\right)-h\left(\mathcal{F}^{(l,j)}\right)\right)*\mathcal{W}^{(i,n_{l+1})}\right\|_2^2\\
&=\frac{1}{a_{l+1}}\sum_{\substack{p_1\in\{1,\cdots,H_{l+1}\}\\ p_2\in\{1,\cdots,l_{l+1}\}}}\left(\delta_{i,j}^{(l)}\left(p_1,p_2\right)*\mathcal{W}^{(i,n_{l+1})}\right)^2\\
&=\frac{1}{a_{l+1}}\sum_{\substack{p_1\in\{1,\cdots,l_{l+1}\}\\ p_2\in\{1,\cdots,l_{l+1}\}}}\left|\left\langle\delta_{i,j}^{(l)}\left(p_1,p_2\right),\mathcal{W}^{(i,n_{l+1})}\right\rangle\right|^2
\end{aligned}\tag{16}$$

Applying Cauchy-Schwarz inequality, then:

$$\begin{aligned}
&\mathcal{L}\left(\mathcal{F}^{(l+1,n_{l+1})},\mathcal{F}_p^{(l+1,n_{l+1})}\right)\\
&\leq\frac{1}{a_{l+1}}\sum_{\substack{p_1\in\{1,\cdots,H_{l+1}\}\\ p_2\in\{1,\cdots,H_{l+1}\}}}\left\|\delta_{i,j}^{(l)}\left(p_1,p_2\right)\right\|_2^2\left\|\mathcal{W}^{(i,n_{l+1})}\right\|_2^2 k\\
&=\frac{1}{a_{l+1}}\left\|\mathcal{W}^{(i,n_{l+1})}\right\|_2^2\sum_{\substack{p_1\in\{1,\cdots,H_{l+1}\}\\ p_2\in\{1,\cdots,H_{l+1}\}}}\left\|\delta_{i,j}^{(l)}\left(p_1,p_2\right)\right\|_2^2
\end{aligned}\tag{17}$$

Actually $\delta_{i,j}^{(l)}$ appears at most K^2 times in the convolution operation, except for the border. For activation functions commonly used in CNN such as ReLU or sigmoid, $\max_{x\in\mathbb{R}}\left(\frac{dh(x)}{dx}\right) \leq 1$ and $\min_{x\in\mathbb{R}}\left(\frac{dh(x)}{dx}\right) \geq 0$, and then:

$$\begin{aligned}
&\mathcal{L}\left(\mathcal{F}^{(l+1,n_{l+1})},\mathcal{F}_p^{(l+1,n_{l+1})}\right)\\
&\leq\frac{1}{a_{l+1}}K^2\left\|\mathcal{W}^{(i,n_{l+1})}\right\|_2^2\left\|\delta_{i,j}^{(l)}\right\|_2^2\\
&=\frac{1}{a_{l+1}}K^2\left\|\mathcal{W}^{(i,n_{l+1})}\right\|_2^2\left\|h\left(\mathcal{F}^{(l,i)}\right)-h\left(\mathcal{F}^{(l,j)}\right)\right\|_2^2\\
&\leq\frac{1}{a_{l+1}}K^2\left\|\mathcal{W}^{(i,n_{l+1})}\right\|_2^2\left\|\mathcal{F}^{(l,i)}-\mathcal{F}^{(l,j)}\right\|_2^2\\
&=\frac{a_l}{a_{l+1}}K^2\left\|\mathcal{W}^{(i,n_{l+1})}\right\|_2^2\mathcal{L}\left(\mathcal{F}^{(l,i)},\mathcal{F}^{(l,j)}\right)
\end{aligned}\tag{18}$$

Since $\mathcal{F}^{(l,i)}$ is an arbitrary channel of the l_{th} layer, we can further narrow the upper bound:

$$\begin{aligned}
&\mathcal{L}\left(\mathcal{F}^{(l+1,n_{l+1})},\mathcal{F}_p^{(l+1,n_{l+1})}\right)\\
&\leq\frac{a_l}{a_{l+1}}K^2\left\|\mathcal{W}^{(i,n_{l+1})}\right\|_2^2\min{}_{j\in\{1,\ldots,N_l\}}\mathcal{L}\left(\mathcal{F}^{(l,i)},\mathcal{F}^{(l,j)}\right)\\
&=\varepsilon\times\min_{j\in\{1,\ldots,N_l\}}\mathcal{L}\left(\mathcal{F}^{(l,i)},\mathcal{F}^{(l,j)}\right)
\end{aligned}\tag{19}$$

where $\varepsilon = \frac{a_l}{a_{l+1}}K^2\left\|\mathcal{W}^{(i,n_{l+1})}\right\|_2^2$ and $a_l = H_l \times W_l \times B$.

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 20 March 2023
DOI 10.3389/fnbot.2023.1148545

OPEN ACCESS

EDITED BY
Xin Ning,
Institute of Semiconductors (CAS), China

REVIEWED BY
Alwin Poulose,
Indian Institute of Science Education
and Research, India
S. Khan,
Kohat University of Science and Technology,
Pakistan
A. Jorge Morais,
Universidade Aberta, Portugal

*CORRESPONDENCE
Xiaobing Chen
✉ chenxiaobing@hyit.edu.cn

RECEIVED 20 January 2023
ACCEPTED 13 February 2023
PUBLISHED 20 March 2023

CITATION
Lin J, Xie X, Wu W, Xu S, Liu C, Hudoyberdi T and Chen X (2023) Model transfer from 2D to 3D study for boxing pose estimation. *Front. Neurorobot.* 17:1148545. doi: 10.3389/fnbot.2023.1148545

Model transfer from 2D to 3D study for boxing pose estimation

Jianchu Lin, Xiaolong Xie, Wangping Wu, Shengpeng Xu, Chunyan Liu, Toshboev Hudoyberdi and Xiaobing Chen*

Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huai'an, China

Introduction: Boxing as a sport is growing on Chinese campuses, resulting in a coaching shortage. The human pose estimation technology can be employed to estimate boxing poses and teach interns to relieve the shortage. Currently, 3D cameras can provide more depth information than 2D cameras. It can potentially improve the estimation. However, the input channels are inconsistent between 2D and 3D images, and there is a lack of detailed analysis about the key point location, which indicates the network design for improving the human pose estimation technology.

Method: Therefore, a model transfer with channel patching was implemented to solve the problems of channel inconsistency. The differences between the key points were analyzed. Three popular and highly structured 2D models of OpenPose (OP), stacked Hourglass (HG), and High Resolution (HR) networks were employed. Ways of reusing RGB channels were investigated to fill up the depth channel. Then, their performances were investigated to find out the limitations of each network structure.

Results and discussion: The results show that model transfer learning by the mean way of RGB channels patching the lacking channel can improve the average accuracies of pose key points from 1 to 20% than without transfer. 3D accuracies are 0.3 to 0.5% higher than 2D baselines. The stacked structure of the network shows better on hip and knee points than the parallel structure, although the parallel design shows much better on the residue points. As a result, the model transfer can practically fulfill boxing pose estimation from 2D to 3D.

KEYWORDS

boxing robot, computer vision, human pose estimation, 3D model transfer, negative transfer

1. Introduction

Boxing as a strenuous exercise is gradually being accepted by the general public in China. It has been promoted in many universities and has relevant professional courses (Xu, 2018; Li L., 2019; Li X., 2019; Logan et al., 2019). It can improve citizens' and students' physical and mental health (Tjønndal, 2019), and even enhances the self-protection abilities of women (Hu, 2018; Fuerniss and Jacobs, 2020). However, this results in a new problem of a coach shortage. Many researchers have tried to employ computer vision and robot technology to solve the shortage problem of coaches (Huang et al., 2019; Li et al., 2021; Lin et al., 2021, 2022; Mendez et al., 2022). The human pose estimation technology can predict boxing elements for better teaching, which can reduce reliance on coaches and increase entertainment in boxing training.

Currently, the 3D camera can provide more depth information than the traditional 2D RGB camera. This advantage can help in the advancement of many tasks of image processing, such as MRI images (Chen et al., 2019; Wu G. et al., 2022), robots (Song et al., 2020), 3D faces (Ning et al., 2020; Wu H. et al., 2022), and so on. 3D human pose estimation becomes a cutting-edge and interesting direction. Many researchers have attempted to reconstruct a 3D human pose estimation with 2D or 3D cameras. Since 2D estimation has been researched comprehensively and wholistically (Wang et al., 2021), it will be crucial to determine whether 2D estimation is compatible with 3D estimation. Adapting the existing 2D models to the application with 3D cameras and studying the advantages of these models is essential to boxing applications and promoting the technology of human pose estimation.

In the field of artificial intelligence, there are two main ways for human pose estimation: bottom-up and top-down (Xiao et al., 2018; Wang et al., 2021). For instance, the top-down method detects each person first, and then directly detects the key points of each person. It is a two-stage method. Most research is based on 2D imagery and shows brightness design and theoretical structures that achieve SOTA results, such as Hourglass (HG) models (Newell et al., 2016; Xiao et al., 2018; Hua et al., 2020; Xu and Takano, 2021), and High Resolution (HR) networks (Sun et al., 2019; Yu et al., 2021; Xu et al., 2022). In contrast, the bottom-up method recognizes the limbs of people at the beginning and groups these limbs for each person, such as in the OpenPose(OP) models (Cao et al., 2017) and Hourglass(HG) models (Nie et al., 2018). Three mainstream models of the OP, HG, and HR networks are suitable for our boxing application. However, the problem of channel inconsistency directly affects the transfer of a 2D model to a 3D image. The basic popular methods need to be investigated deeply, and it is important to reveal their performance differences in detail for better improvement.

Model transfer technology is employed to help improve the application of human pose estimation by transferring their models and parameters. The performance of estimation of boxing poses is evaluated on RGBD image. The main contributions of this paper are:

- The depth channel is patched by different strategies when data input is inconsistent, which illustrates that the negative transfer can happen in this step, and it implies that the machine learning method can further improve the strategy.
- A detailed analysis of human pose estimation technology reveals the advantages and disadvantages of mainstream models used in boxing pose estimation, indicating the new improving direction of this technology.
- The model transfer from 2D to 3D images is studied for boxing practice, which shows that 2D models can be compatible with the 3D inputs of 3D cameras.

In this manner, the three mainstream models of the OP, HG, and HR networks are studied. The following sections are mainly divided into three parts: (1) Related work. Research work about human pose estimation is presented and analyzed. The important structures of neural networks are discussed; (2) Method. The model-transfer technology is employed to study the transfer of relative top-down and bottom-up models, respectively. 2D inputs are transferred to adaptive 3D inputs. This section also describes different ways for model transfer. (3) Results and discussion. The previously mentioned approaches are carried out after model transfer, and 3D and 2D transfer results are analyzed in detail. Three basic methods are discussed to analyze their existing problems.

2. Related work

2.1. Top-down way

Newell et al. (2016) proposed the HG method, which expanded the ResNet structure to realize the extraction of pose information. To improve the joint position regression, Xiao et al. (2018) added a few deconvolutional layers over the last convolution stage in the ResNet, which generated heatmaps from deep and low-resolution features. Considering Xiao's architecture, Moon et al. (2019) designed a PoseFix network to refine the estimation, which applies to a model-agnostic pose refinement method. Hua et al. (2020) took a multipath affinage way to improve HG networks. Furthermore, Graph stacked HG network was developed by Xu and Takano (2021). It has an HG shape consisting of a chain of convolution and up-convolution layers followed by a regression part for generating a 3D pose. However, this estimation is based on 2D image inputs.

Chen et al. (2018) proposed a cascaded pyramid network (CPN) for human pose estimation. It has GloableNet and RefineNet as two parts, and each layer was parallel to exchange information. But for a better exchange of information between different scale features, Sun et al. (2019) further proposed a high-resolution (HR) network method for information exchange at the base of a huge pyramid structure. HigherHRNet was proposed (Cheng et al., 2020) to use the high-resolution feature pyramid for prediction by a 1 × 1 convolution to heatmaps based on the HR network. It can solve the scale variation challenge in bottom-up multi-person pose estimation. To reduce the parameters and improve speed, Yu et al. (2021) refined the HR network called the Lite-HR network, which applies shuffle blocks to the HR network. The accuracy got a slight drop. Xu et al. (2022) applied the transformer model to human pose estimation. This simple structure can be the backbone to extract features for the HR network. This improvement is based on longer training, and it has challenges for patch embedding when there is less information around key points.

2.2. Bottom-up way

The OP network model (Cao et al., 2017) based on the affinity field could simultaneously locate multiple people and get wild applications (Nakai et al., 2018; Viswakumar et al., 2019; Chen et al., 2020; Nakano et al., 2020). The key points are detected in each joint class and grouped into limbs between each joint point. With a part association field (PAF) restriction, the limbs are gathered at a minimum cost. Nie et al. (2018) proposed a pose partition network (PPN) to detect joints and regression for multi-persons, which is based on the HG network. Since a PAF

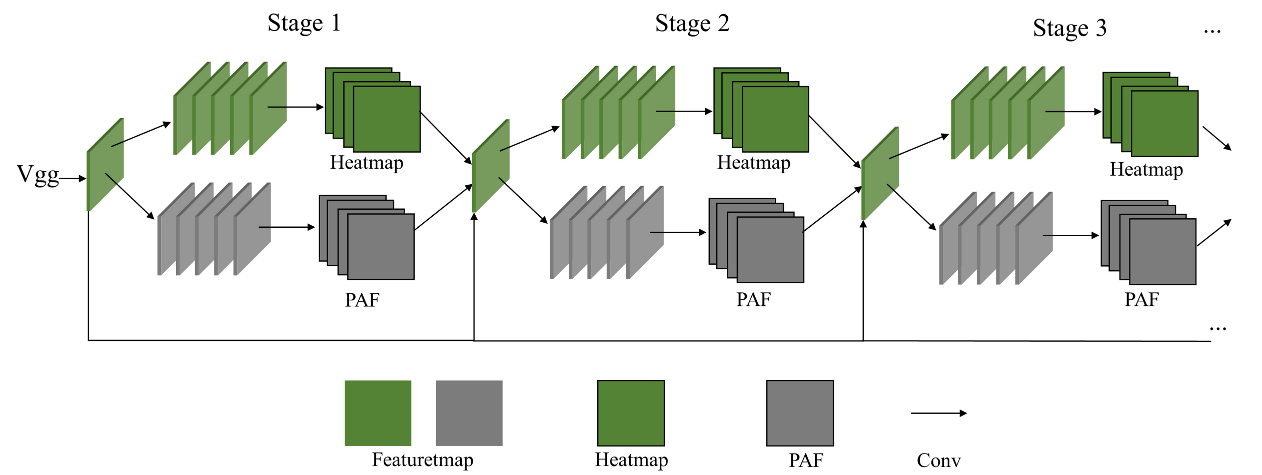


FIGURE 1
OP network model with a parallel steam structure of heatmaps and PAFs.

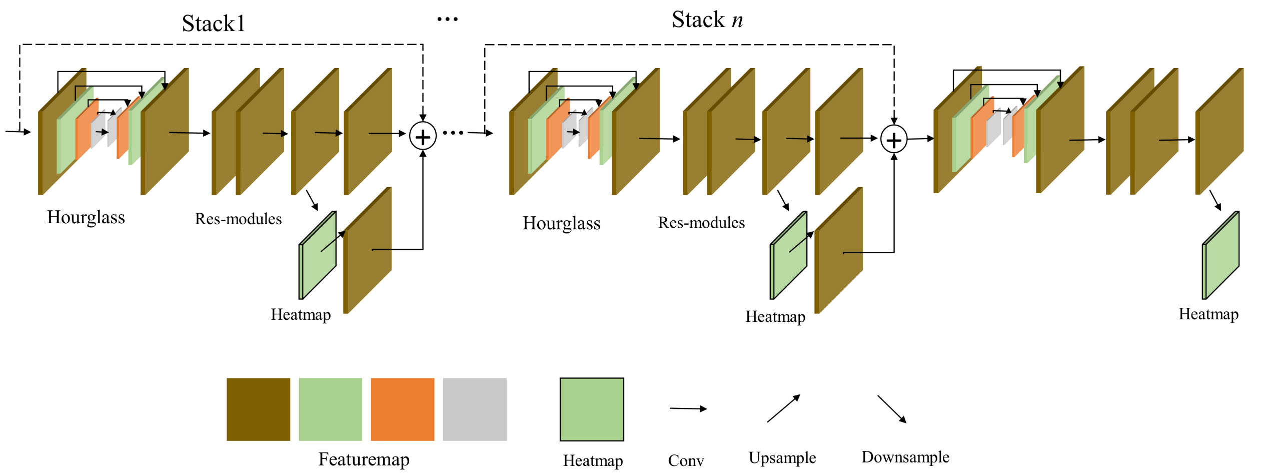


FIGURE 2
Stacked HG network with local and global contexts.

was used to associate body parts with each other, a part intensity field (PIF) was proposed to localize body parts (Kreiss et al., 2019) and help form full human poses. Osokin (2018) proposed a lightweight OP network, which only remained in one refinement stage and replaced the VGG network with the MobileNet in the backbone. Wu et al. (2021) proposed a rapid OP network for astronaut operation attitude detection. They changed the original two-branch structure to a single-branch structure, which improved the calculation speed. Geng et al. (2021) proposed disentangled key point regression (DEKR), which uses a multi-branch structure for separate regressions to get the key points in the bottom-up paradigm. For multi-person pose estimation, Jin et al. (2020) reformulated the task of multi-person pose estimation as a graph-clustering problem. The OP networks rely on the backbone network for feature extraction.

According to the above analysis, studies based on the OP, HG, and HR networks are very extensive, and the HR and HG networks can be used both in the top-down and bottom-up way. The study of these three methods can be better at comprehensively finding problems in our boxing sport application.

3. Materials and methods

3.1. Device and dataset

The image data collecting tool was a 3D Intel Realsense D455 camera with a 640 × 480 resolution. It was placed approximately 185 cm above the ground, with a depression angle of 10 degrees around. RGBD data was collected in various indoor environments such as classrooms and research labs. Five basic boxing poses, including punch, swing, hook, backward, and side slide were recorded in left and right ways. More than 100 students contributed to the collection. After data cleaning, 280 images were selected to form a dataset for this research. The dataset was randomly divided into three parts: a training set (120 images), a validation set (40

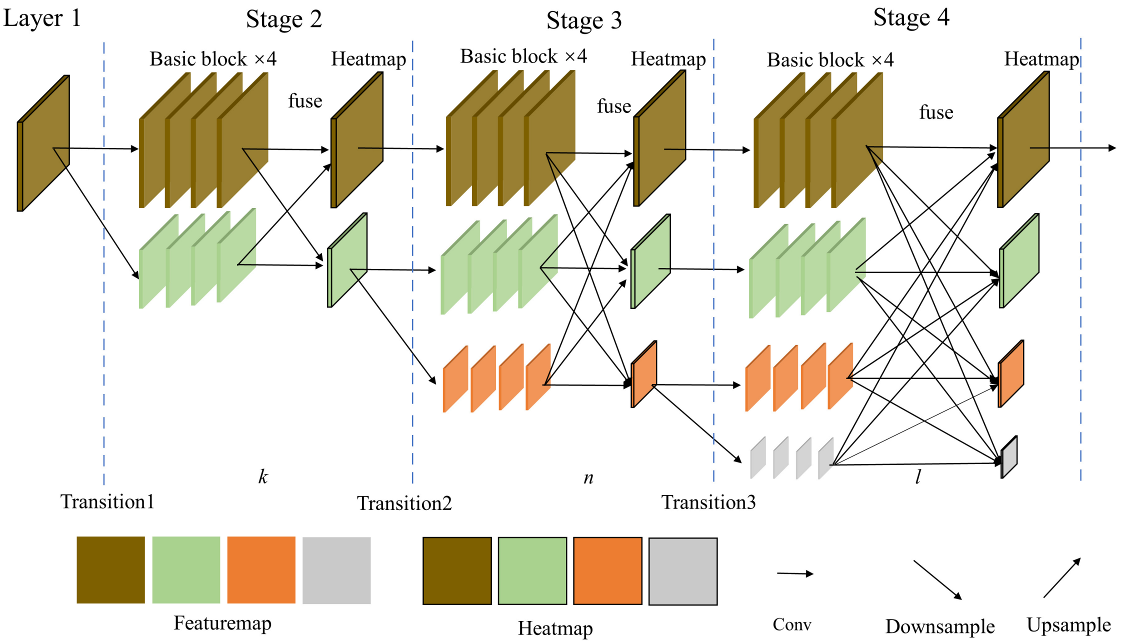


FIGURE 3
HR model with an inverse pyramid parallel structure and information exchange fuse.

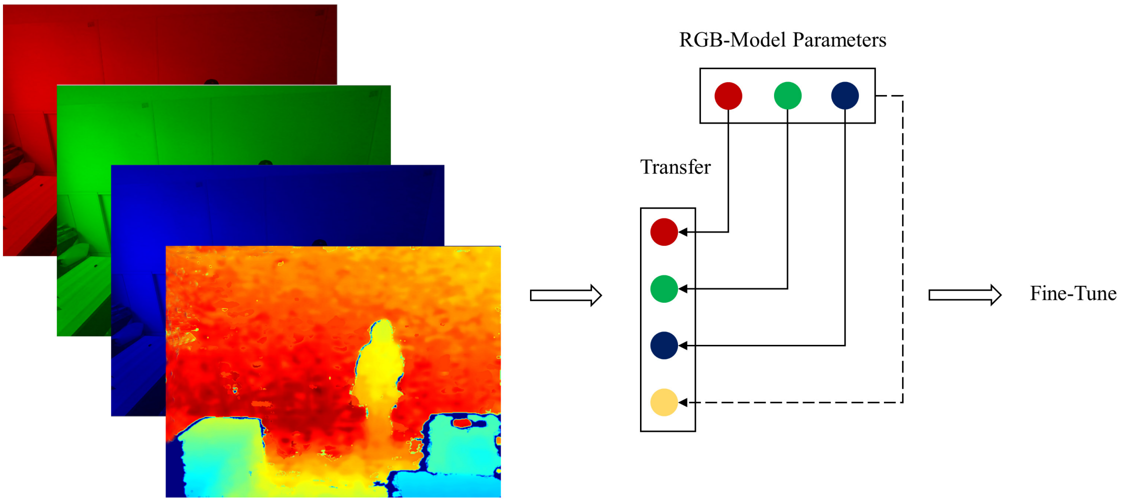


FIGURE 4
The RGB-D model transfer based on the RGB model.

images), and a test set (120 images). These three sets didn't have the same person.

3.2. Model transfer learning

After testing, three SOTA models of the OP, HG, and HR networks were studied. They were transferred as source models since these three basic models have been researched extensively and achieve each best performance. All the models are 2D inputs. The boxing pose estimation was the target learning task. The 640 × 480 boxing image is estimated directly because three models are studied under the same size of the input, feature maps, and heatmaps.

3.2.1. OP network model

There are two primary parallel branches in the OP network. One branch is trained to predict the heatmap of human pose key points, and the other branch is trained to predict the PAF that can help organize the components of body limbs in a bottom-up way. The model repeats the basic branches several times as stages as displayed in Figure 1. This design can be easy for multi-person estimation as it estimates every person's key point at one computation, but the heatmap branch is simple for

TABLE 1 The average accuracy of different key points.

Network	Key point	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total average
OP (%)	Without	82.3	89.9	80.2	72.6	71.7	91.4	92.8	83.0
	2D	**98.9**	**95.2**	**91.7**	**91.3**	**88.2**	**97.1**	**97.4**	**95.3**
	3D-R	98.6	96.2	89.7	88.9	85.7	95.8	97.4	93.2
	3D-G	97.9	94.7	85.2	86.3	86.7	95.3	97.6	92.0
	3D-B	97.3	96.4	86.8	90.6	85	97.2	97.4	93.0
	3D-Mean	**99.4**	**96.9**	**92.2**	**92.1**	**92.1**	**97.5**	**97.9**	**95.8**
HG (%)	Without	95.8	89.9	80.2	72.6	87.6	90.2	94.6	90.7
	2D	**99.7**	**95.2**	**91.7**	**91.3**	**92.9**	**99.2**	**97.5**	**96.1**
	3D-R	99.8	96.2	89.7	88.9	93.4	98.0	97.2	96.3
	3D-G	99.6	94.7	85.2	86.3	92.5	98.7	96.8	95.9
	3D-B	99.6	96.4	86.8	90.6	92.3	98.6	96.9	96
	3D-Mean	**99.8**	**96.9**	**92.2**	**92.1**	**93.8**	**99.5**	**98.9**	**96.5**
HR (%)	Without	94.5	98.7	91.6	81.2	79.5	97.5	98.7	91.7
	2D	**99.8**	**98.6**	**94.0**	**95.9**	**90.5**	**97.9**	**99.8**	**96.6**
	3D-R	99.5	97.5	85.0	91.7	85.4	93.1	96.1	92.6
	3D-G	99.6	97.9	91.4	93.6	86.5	97.2	97.4	96.9
	3D-B	96.2	98.3	85.0	93.5	86.4	93.5	96.7	92.8
	3D-Mean	**99.8**	**98.6**	**94.6**	**96.8**	**91.8**	**97.9**	**99.8**	**97.0**

Bold values represent the best results of 2D and 3D model transfer.

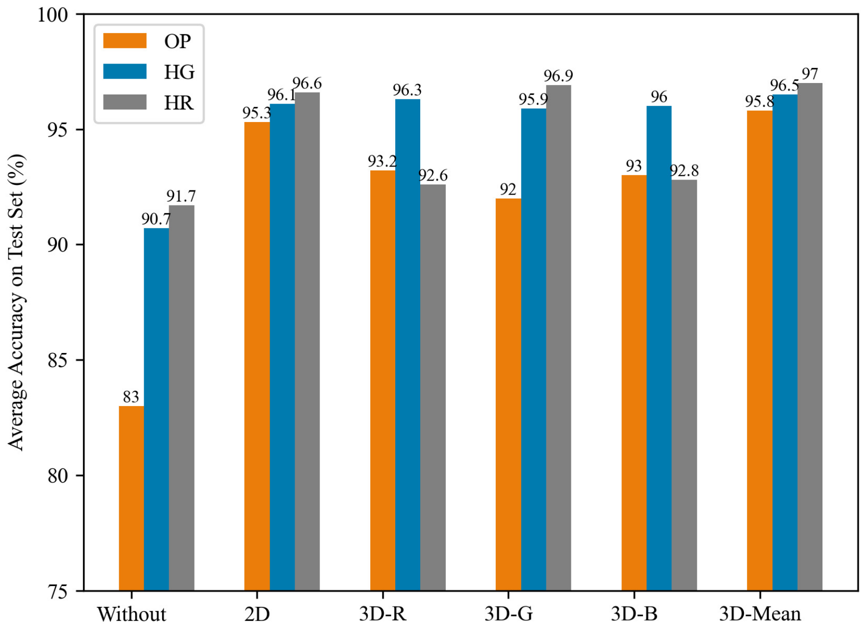


FIGURE 5
The average results of different transfer way by OP, HG, and HR models.

extracting complex features and structures since it only depends on convolution layers.

3.2.2. Stacked HG network model

In Figure 2, the HG network is inspired by the pyramid structure to deal with the local and global context. In each stack, there is a pyramid structure integrated inside, and the heatmaps are generated to predict key points, and each stack is repeated to group a complex network. It can be seen that the learning ability is improved by its pyramid structure. The stack is very similar to the OP stage, so it can be used in both top-down and bottom-up ways, but there is less information exchange for each stack. This may cause limited learning in the local context.

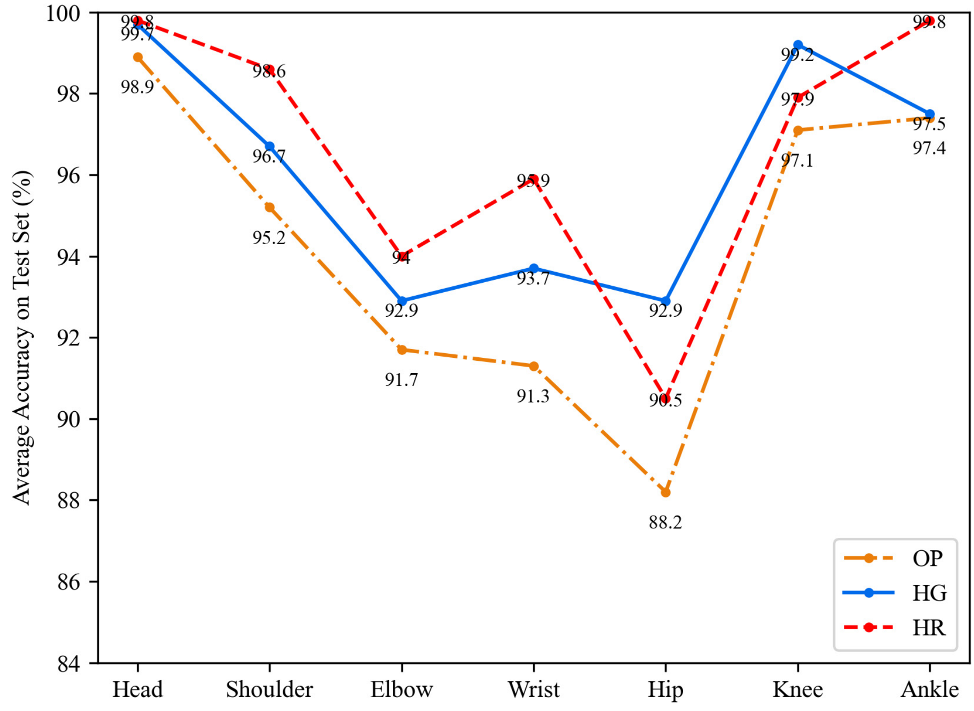


FIGURE 6
The average accuracy on each kind of key point in a 2D way.

3.2.3. HR network model

As shown in Figure 3, the HR network has an inverse pyramid parallel structure compared to the HG network. It can be seen that there are 4 parallel channels in different image scales, which can get local and global information. Besides, this separates 3 or 4 stages to make the model better exchange information between different scale feature maps. The parallel branches in each stage are usually repeated a few times to make a better extraction. Therefore, features can be combined in multiple ways. Compared with the above two networks, the HR network in each stage doesn't keep the same size, which means it may have less learning ability for symmetry structures.

The above three source models ($Model_{2D}$) were transferred as shown in Figure 4. The input data in a source learning task can be described as$X_S = \{x_1^n, x_2^n, x_3^n, \cdots, x_k^n\}$, n is the input data dimension, and k is the instance number. In the target learning task, the input data is$X_T = \{x_1^{n+1}, x_2^{n+1}, x_3^{n+1}, \cdots, x_k^{n+1}\}$, and the instance number m is far less than k. Therefore, the posterior distribution of the source domain $P_S(y|x^n)$ needs to change to the target domain posterior distribution $P_T(y|x^{n+1})$. The fine-tuning method can be used to adapt the source posterior distribution to the target domain. To solve the problem of lacking depth channel in source models, an additional channel of the parameter was patched based on RGB channels as in formula (1):

$$\mathrm{Model}_{3\mathrm{D}} = Model_{2\mathrm{D}}\,(R, G, B) + Param_{\mathrm{channel}} \qquad (1)$$

Where the channel parameters can be chosen from R, G, and B channels or the mean combination of these three channels. When transferring parameters from source models, the different channel effects should be examined, and the best way to improve the depth channel effects on predicting posture points should be determined. Boxing pose data were used to fine-tune the models to generate new models.

4. Results and discussion

The performance of transferred models was studied in four ways: (1) the average accuracy was obtained about boxing pose key point positions, and the corresponding accuracy of each model after the transfer of different channels was compared to baselines of 2D transfer; (2) the impact of fine-tuning instance amount on model transfer improvement; (3) the Flops and parameter number of each model were shown for evaluating model complex, and average cost times of models per image were compared; (4) a direct comparison of the pose estimation of boxing basic actions among different models, along with pose estimation display.

4.1. The average prediction accuracies of key points

There are seven distinct critical points for estimating human poses including the head, shoulder, elbow, wrist, hip, knee, and ankle. To keep the comparison of points consistent, the neck point of OP does not show here. Table 1 shows the average accuracy of 10 times repeat on each point recorded without fine-tuning, 2D transfer with fine-tuning, and 3D transfer with a different kind of channel. The source models of the OP, HR, and HG networks were pretrained and released publicly by their developers.

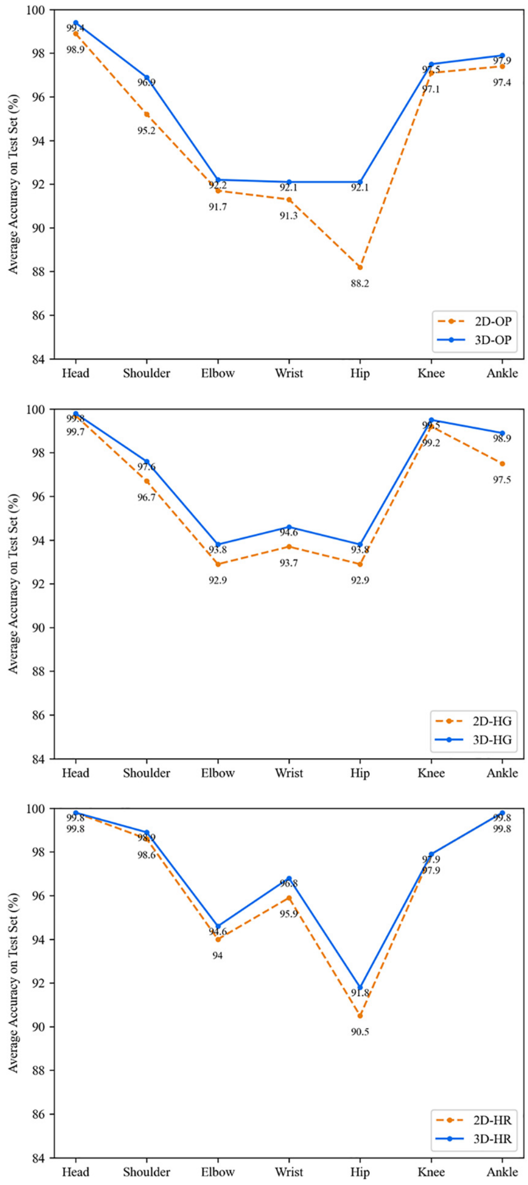


FIGURE 7
The detailed performance of different networks comparing 2D and mean of 3D.

Table 1 shows that three models are listed in each column, and in each row, the average accuracy of each key point is compared. The 2D rows can be chosen as baselines for 3D transfer. In 3D transfers, the R, G, and B channels were examined, respectively. In addition, the mean of the three channels was also examined.

Each network in Table 1 includes 6 different methods of fine-tuning; 2D, and 3D are displayed in Figure 5. There are 6 groups for three networks. The accuracies of the first group are lower than those of the second group after finetuning in 2D transfer, and the HR network achieves the highest average accuracy of 96.6%. The HG network achieves a similar performance at 96.1%, while the OP network achieves 95.3%, which is increased by 12.3%. After fine-tuning, both networks perform slightly better than the OP network. When in 3D transfer, the third to sixth groups in Figure 5 show different channel parameters that are chosen or combined to patch the lacking depth channel. The R and B channels affect the OP and HG networks the most, whereas the G channel affects the HR network the most, and the R and B channels even can cause a negative transfer on the HR network (from 96.6 to 92.8%). The mean strategy of R, G, and B channels get the best estimation than in a single one-channel way. This situation shows that the three networks learn different patterns in different channels. The OP and HG network extracts features from three channels equally, while the HR network gets features from the G channel, which is highly related to depth information. Features from B and R channels are less related to depth information.

Figure 6 shows the detailed results of 2D. The HR network can get the best estimation on the head, shoulder, elbow, wrist, and ankle points. However, the HG network estimates hip and knee points better than the HG network. Both the OP and HR networks are worse in hip point estimation. This phenomenon might be caused by the lower learning ability of the OP network and the lower symmetric ability of the HR network than the HG network since these two kinds of points have fewer texture features in the image. The HG model has a lower feature extraction than the HR network.

Finally, compared with the second group of 2D transfer, the mean 3D group average accuracies are all higher than that of the 2D group as shown in Figure 7. The OP network is improved by 3.3% on hip points, which is higher than other networks. This may be caused because of the previous imbalance training by authors. In Figure 8, the 3D transfer also shows a similar result as the 2D transfer. The HR network performs better on the head, shoulder, elbow, wrist, and ankle points, but the HG network performs better on hip and knee points. It means that the HR network has a deficiency on the hip and knee points when there is less texture information around. The depth channel shows less help to the estimation. This may be the bottleneck of transfer learning when lacking depth training data.

4.2. The fine-tuning effect of different training data set size

The average accuracy curves of the OP, HR, and HG networks are drawn under the different training dataset sizes from 40 to 120, which increases by 20 each step.

As shown in Figure 9, the horizontal axis indicates the amount of training dataset size. The vertical axis indicates the average accuracy on the test set. The results show the HR model still performs better than others. But when the data size is small, from 40 to 80, it is almost the same as the HG network. With an increase in the training dataset size, the HR network becomes better than the HG network. It means the HR network might get much better results when the dataset size becomes large. The OP network performance increases a bit slower than the other two networks and it tends to be plain.

4.3. The FLOPs and average cost time

The Floating-Point Operations (FLOPs) and average cost time on the test set are shown in Table 2. The same resolution images

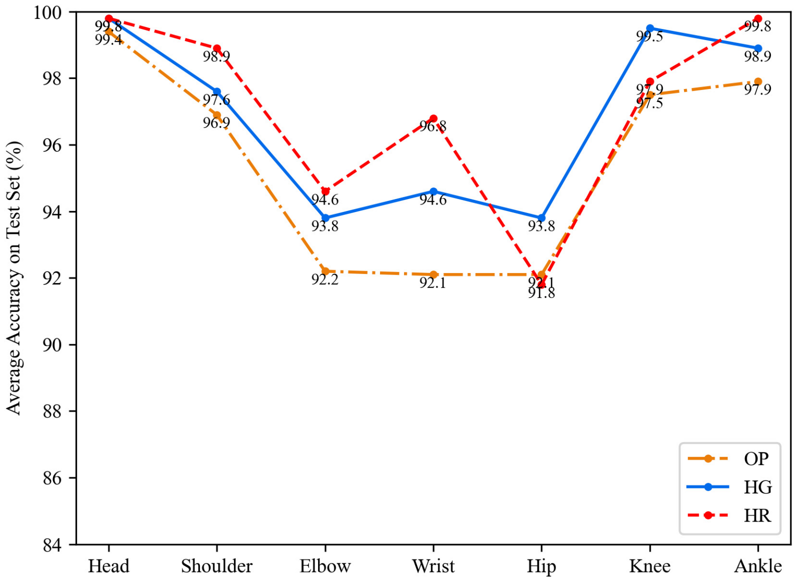


FIGURE 8
The average accuracy on each key point in the mean 3D way.

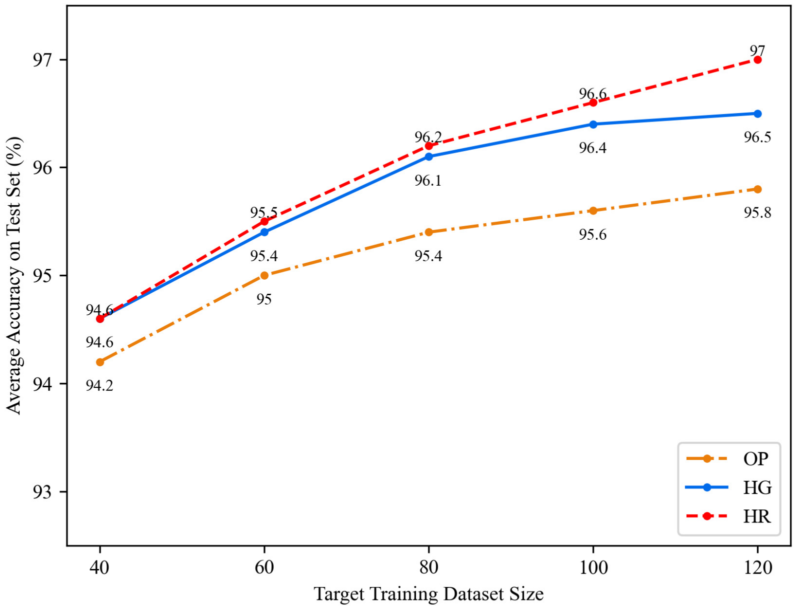


FIGURE 9
The total average accuracy of different networks after model transfer.

TABLE 2 FLOPs and average time cost on each model.

Method	Input size	#Params	GFLOPs	Average cost time (s)
OP	640 × 480	52.3M	308.5	0.57
HG	640 × 480	53.1M	359.8	0.68
HR	640 × 480	28.5 M	48.08	0.34

were fed into the three networks. The parameter number of each kind of network is displayed. It can be seen that the numbers of HG and OP's parameters and average cost times are almost equal, and they are both nearly twice that of HR. As the parameters are reduced by half, the GFLOPs can be reduced largely.

4.4. Comparison of the pose estimation on boxing basic movements

Five postures of punch, swing, hook, backward, and side sliding are estimated in both left and right ways as displayed in **Figure 10**. The figure shows two different scenes, and the results are listed in a sequence of without fine-tuning, after the 3D transfer, and ground

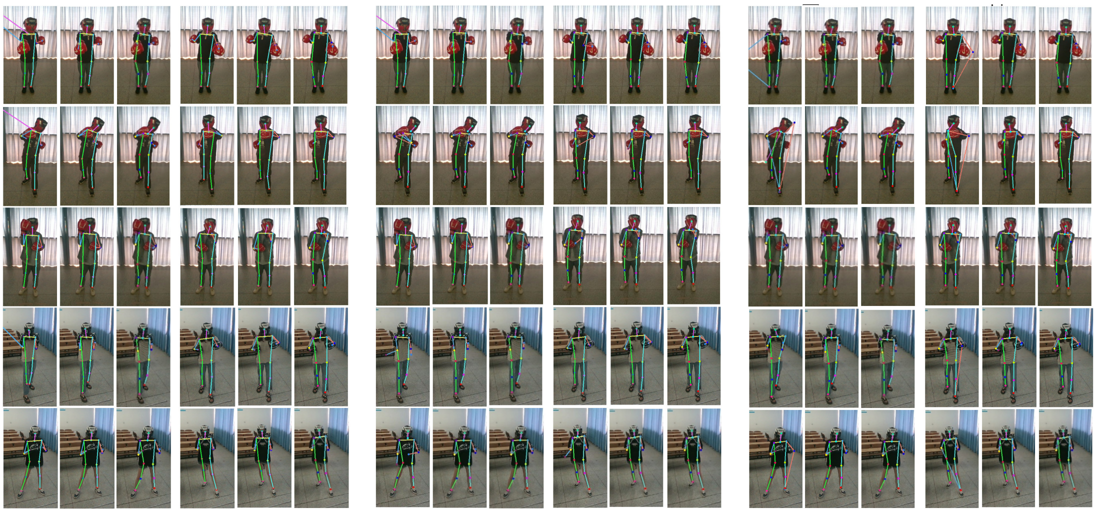


FIGURE 10
The pose estimation of the OP network **(left)**, HG network **(middle)**, and HR network **(right)** after model transfer.

truth. It also displays the results of the OP, HG, and HR networks, respectively.

In the first column of each figure, there are many errors in five poses. The head and wrist points are wrongly located. The estimation results are littered randomly since the left and right results of both the OP and HG networks have a big difference. This phenomenon may be caused by the background. The edge of curtains or chairs might be like the human edge. The HR network's left and right results are more symmetric than other methods. Besides, the obscuring from the boxing helmet and the camera's inconsistent view also result in locating the wrong place such as the ankle, hip, and elbow.

The second columns of the left and right poses show the estimations are improved. It is much closer to the third column of ground truth. So, fine-tuning can correct the errors from background interference, obscuring, and view inconsistency. In addition, it can be seen that the HR network's estimation of hip and knee joints is compelled in a line, which is quite different from the HG's estimation. That means the HR network can be further improved.

5. Conclusion

With the popularization of boxing in China, the lack of coaches and amusement impedes the promotion of this sport. The research on intelligent humanoid boxing robots becomes hotter, and the problem of insufficient coaches can be solved. Through the application of human pose estimation technology, the actions of boxing athletes can be analyzed, guided, and taught. The inconsistent inputs between the current image-based 2D human pose estimation technology and the 3D data of RGBD prevent our study because of the shortage of boxing data. The model transfer method is adopted to improve the technology application by patching the lack of channel. Three SOTA models of this technology were studied and transferred for experiments. Different strategies of transfer were examined to patch the lack of depth channel. The results show that the mean combination of RGB channel parameters is suitable to patch the depth channel. This strategy can improve models' estimation performance stability. In addition, model transfer learning can efficiently reduce the dependence on collecting new data. The three SOTA models of the OP, HR, and HG networks exhibit competitive ability, and each model achieves a better performance after a mixture of depth channel information. Based on this research, the technical problems existing in the application of boxing can be revealed further, such as the HR network needing to improve the estimation of hip and knee joints and integrating these basic models into a small platform for the different kinds of applications. A machine learning method to optimize this combination can be researched further. Nonetheless, transfer learning with the channel patching method has been successfully studied for boxing pose estimation, and the 2D model performance can be improved by a 3D camera. Data can be collected to enhance the model's application.

Data availability statement

The original contributions presented in this study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

JL supervised the projects, proposed the research plan for this article, and dealt with the revision of the algorithm. XC supervised the analysis of models and analyzed the theory for networks. XX and TH were in charge of data collection and analysis, coding, article writing, translation, and discussion. WW, SX, and CL implemented the experiments, analysis of results, and code revising. All authors contributed to the article and approved the submitted version.

Funding

This work is supported by the program of Entrepreneurship and Innovation Ph.D. in Jiangsu Province (JSSCBS20211175), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (21KJB520002), and the School Ph.D. Talent Funding (Z301B2055).

Acknowledgments

Thanks to the students who participated in data collection and patiently completed the boxing postures.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, HI, 7291–7299. doi: 10.1109/CVPR.2017.143

Chen, S., Ma, K., and Zheng, Y. (2019). Med3d: Transfer learning for 3d medical image analysis. *arXiv* [Preprint]. arXiv:1904.00625.

Chen, W., Jiang, Z., Guo, H., and Ni, X. (2020). Fall detection based on key points of human-skeleton using OpenPose. *Symmetry* 12:744. doi: 10.3390/sym12050744

Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., and Sun, J. (2018). "Cascaded pyramid network for multi-person pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, UT, 7103–7112. doi: 10.1109/CVPR.2018.00742

Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T. S., and Zhang, L. (2020). "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Seattle, WA, 5386–5395. doi: 10.1109/CVPR42600.2020.00543

Fuerniss, K., and Jacobs, J. M. (2020). We are strong: Strategies for fostering body empowerment in a boxing program for middle school girls. *J. Sport Psychol. Act.* 11, 45–56. doi: 10.1080/21520704.2019.1693456

Geng, Z., Sun, K., Xiao, B., Zhang, Z., and Wang, J. (2021). "Bottom-up human pose estimation via disentangled keypoint regression," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Manhattan, NY, 14676–14686. doi: 10.1109/CVPR46437.2021.01444

Hu, X. (2018). Analysis on the development status of professional boxing in China. *Adv. Phys. Sci.* 3, 91–95. doi: 10.12677/APS.2018.63016

Hua, G., Li, L., and Liu, S. (2020). Multipath affinage stacked-hourglass networks for human pose estimation. *Front. Comput. Sci.* 14:144701. doi: 10.1007/s11704-019-8266-2

Huang, X., Wang, G., Ren, P., and Hu, Y. (2019). Study on the dynamic system of the desktop boxing robot. *J. Mach. Des.* 36, 32–36.

Jin, S., Liu, W., Xie, E., Wang, W., Qian, C., Ouyang, W., et al. (2020). "Differentiable hierarchical graph grouping for multi-person pose estimation," in *Proceedings of the European conference on computer vision*, *Online*, 718–734. doi: 10.1007/978-3-030-58571-6_42

Kreiss, S., Bertoni, L., and Alahi, A. (2019). "Pifpaf: Composite fields for human pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Manhattan, NY, 11977–11986. doi: 10.1109/CVPR.2019.01225

Li, L. (2019). Research on problems and countermeasures in the development of boxing into the campus. *Box. Fight* 4:134.

Li, X. (2019). Feasibility study on opening boxing in physical education course of primary and secondary schools. *Wushu Stud.* 4, 70–72.

Li, Y., Wang, P., Fang, Y., and Wang, D. (2021). Design and implementation of a boxing robot based on fuzzy control. *J. Phys.* 1303:012065. doi: 10.1088/1742-6596/1303/1/012065

Lin, J. C., Gu, Z., Amir, A. M., Chen, X., Ashim, K., and Shi, K. (2021). "A fast humanoid robot arm for boxing based on servo motors," in *Proceedings of the IEEE international Conference on high-performance big data and intelligent systems*, Macau, 252–255. doi: 10.1109/HPBDIS53214.2021.9658471

Lin, J. C., Xie, X., Wu, W., Xu, S., Liu, C., and Hudoyberdi, T. (2022). "Human pose estimation for boxing based on model transfer learning," in *Proceedings of the IEEE international conference on high-performance big data and intelligent systems*, Tianjin, 333–336. doi: 10.1109/HDIS56859.2022.9991696

Logan, K., Cuff, S., LaBella, C. R., Brooks, M. A., Canty, G., Diamond, A. B., et al. (2019). Organized sports for children, preadolescents, and adolescents. *Pediatrics* 143:e20190997. doi: 10.1542/peds.2019-0997

Mendez, S. L. A., Ng, H. Y., Lim, Z. Y., Lu, Y.-J., and Han, P.-H. (2022). "MovableBag: Substitutional robot for enhancing immersive boxing training with encountered-type haptic," in *Proceedings of the SIGGRAPH Asia 2022 XR*, (New York, NY: ACM), 1–2. doi: 10.1145/3550472.3558406

Moon, G., Chang, J. Y., and Lee, K. M. (2019). "Posefix: Model-agnostic general human pose refinement network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Long Beach, CA, 7773–7781. doi: 10.1109/CVPR.2019.00796

Nakai, M., Tsunoda, Y., Hayashi, H., and Murakoshi, H. (2018). "Prediction of basketball free throw shooting by OpenPose," in *Proceedings of the JSAI international symposium on artificial intelligence*, New York, NY, 435–446. doi: 10.1007/978-3-030-31605-1_31

Nakano, N., Sakura, T., Ueda, K., Omura, L., Kimura, A., Iino, Y., et al. (2020). Evaluation of 3D markerless motion capture accuracy using OpenPose with multiple video cameras. *Front. Sports Act. Living* 2:50. doi: 10.3389/fspor.2020.00050

Newell, A., Yang, K., and Deng, J. (2016). "Stacked hourglass networks for human pose estimation," in *Proceedings of the European conference on computer vision*, (Cham: Springer), 483–499. doi: 10.1007/978-3-319-46484-8_29

Nie, X., Feng, J., Xing, J., and Yan, S. (2018). "Pose partition networks for multi-person pose estimation," in *Proceedings of the European conference on computer vision*, Munich, 684–699. doi: 10.1007/978-3-030-01228-1_42

Ning, X., Duan, P., Li, W., and Zhang, S. (2020). Real-time 3D face alignment using an encoder-decoder network with an efficient deconvolution layer. *IEEE Signal Process. Lett.* 27, 1944–1948. doi: 10.1109/LSP.2020.3032277

Osokin, D. (2018). Real-time 2d multi-person pose estimation on CPU: Lightweight OpenPose. *arXiv* [Preprint]. arXiv:1811.12004. doi: 10.5220/0007555407440748

Song, S., Zeng, A., Lee, J., and Funkhouser, T. (2020). Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robot. Autom. Lett.* 5, 4978–4985. doi: 10.1109/LRA.2020.3004787

Sun, K., Xiao, B., Liu, D., and Wang, J. (2019). "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Long Beach, CA, 5693–5703. doi: 10.1109/CVPR.2019.00584

Tjønndal, A. (2019). "Girls are not made of glass!": Barriers experienced by women in Norwegian olympic boxing. *Sociol. Sport J.* 36, 87–96. doi: 10.1123/ssj.2017-0130

Viswakumar, A., Rajagopalan, V., Ray, T., and Parimi, C. (2019). "Human gait analysis using OpenPose," in *Proceedings of the IEEE international conference on image information processing*, Shimla, 310–314. doi: 10.1109/ICIIP47207.2019.8985781

Wang, J., Tan, S., Zhen, X., Xu, S., Zheng, F., He, Z., et al. (2021). Deep 3D human pose estimation: A review. *Comput. Vis. Image Understand.* 210:103225. doi: 10.1016/j.cviu.2021.103225

Wu, E. Q., Tang, Z. R., Xiong, P., Wei, C. F., Song, A., and Zhu, L. M. (2021). ROpenPose: A rapider openpose model for astronaut operation attitude detection. *IEEE Trans. Ind. Electron.* 69, 1043–1052. doi: 10.1109/TIE.2020.3048285

Wu, G., He, F., Zhou, Y., Jing, Y., Ning, X., Wang, C., et al. (2022). ACGAN: Age-compensated makeup transfer based on homologous continuity generative adversarial network model. *IET Comput. Vis.* 1–12. doi: 10.1049/cvi2.12138

Wu, H., Luo, J., Lu, X., and Zeng, Y. (2022). 3D transfer learning network for classification of Alzheimer's disease with MRI. *Int. J. Mach. Learn. Cybern.* 13, 1997–2011. doi: 10.1007/s13042-021-01501-7

Xiao, B., Wu, H., and Wei, Y. (2018). "Simple baselines for human pose estimation and tracking," in *Proceedings of the European conference on computer vision*, Munich, 466–481. doi: 10.1007/978-3-030-01231-1_29

Xu, Q. (2018). Exploration on the construction of boxing culture in Chinese colleges and universities. *Sports World* 9, 105–106.

Xu, T., and Takano, W. (2021). "Graph stacked hourglass networks for 3d human pose estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Manhattan, NY, 16105–16114. doi: 10.1109/CVPR46437.2021.01584

Xu, Y., Zhang, J., Zhang, Q., and Tao, D. (2022). ViTPose: Simple vision transformer baselines for human pose estimation. *arXiv* [Preprint]. arXiv:2204.12484.

Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L., Sang, N., et al. (2021). "Lite-hrnet: A lightweight high-resolution network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, *Online*, 10440–10450. doi: 10.1109/CVPR46437.2021.01030

frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 21 March 2023
DOI 10.3389/fnbot.2023.1155038

OPEN ACCESS

EDITED BY
Xin Ning,
Institute of Semiconductors (CAS), China

REVIEWED BY
Baoli Lu,
University of Portsmouth, United Kingdom
Arghya Datta,
Amazon, United States
Aditya Kumar Sahu,
Amrita Vishwa Vidyapeetham University, India

*CORRESPONDENCE
Tangxian Chen
✉ 114830073@qq.com

RECEIVED 31 January 2023
ACCEPTED 03 March 2023
PUBLISHED 21 March 2023

CITATION
Zhong J, Chen T and Yi L (2023) Face expression recognition based on NGO-BILSTM model. *Front. Neurorobot.* 17:1155038.
doi: 10.3389/fnbot.2023.1155038

Face expression recognition based on NGO-BILSTM model

Jiarui Zhong, Tangxian Chen* and Liuhan Yi

College of Electrical Engineering and New Energy, China Three Gorges University, Yichang, China

Introduction: Facial expression recognition has always been a hot topic in computer vision and artificial intelligence. In recent years, deep learning models have achieved good results in accurately recognizing facial expressions. BILSTM network is such a model. However, the BILSTM network's performance depends largely on its hyperparameters, which is a challenge for optimization.

Methods: In this paper, a Northern Goshawk optimization (NGO) algorithm is proposed to optimize the hyperparameters of BILSTM network for facial expression recognition. The proposed methods were evaluated and compared with other methods on the FER2013, FERplus and RAF-DB datasets, taking into account factors such as cultural background, race and gender.

Results: The results show that the recognition accuracy of the model on FER2013 and FERPlus data sets is much higher than that of the traditional VGG16 network. The recognition accuracy is 89.72% on the RAF-DB dataset, which is 5.45, 9.63, 7.36, and 3.18% higher than that of the proposed facial expression recognition algorithms DLP-CNN, gACNN, pACNN, and LDL-ALSG in recent 2 years, respectively.

Discussion: In conclusion, NGO algorithm effectively optimized the hyperparameters of BILSTM network, improved the performance of facial expression recognition, and provided a new method for the hyperparameter optimization of BILSTM network for facial expression recognition.

KEYWORDS

northern goshawk algorithm, NGO-BILSTM model, face recognition, facial expression, hyperparameter optimization

1. Introduction

The change of facial expression can reflect the change of human emotions and psychology, which plays an indispensable role in daily life (Li and Deng, 2020; Revina and Emmanuel, 2021). Human beings express their emotions mainly through language, voice tone, body movements and facial expressions, and facial expressions contain a large amount of effective information, which can convey the real emotions of human hearts and are more accurate than the information conveyed by language expressions (Li et al., 2018; Minaee et al., 2021; Yang et al., 2021).

In recent years, a variety of AI devices have come into the public eye, and Artificial intelligence algorithms are developing rapidly (Prajapati et al., 2021; Ramachandran and Rajagopal, 2022; Ravinder et al.). The public hopes that computers can understand and express their own emotions through facial expression recognition like humans do, and also give correct feedback according to users' emotional needs (Li et al., 2020; Zhang, 2020).

Exploring face expression recognition technology can provide technical support for artificial intelligence emotional expression. The literature (Han et al., 2022) proposes a new HRL model, which uses the universal matching measure to dynamically display the discriminant learning constraint features in facial expression recognition, and develops an example demonstration. The literature (Gao et al., 2020) proposes a face recognition method based on plural data enhancement, which uses the information provided by the original face image for feature extraction and then fuses the original image with the new feature image to obtain a synthetic plural image that can perform face image recognition under non-ideal conditions. The literature (Gurukumar et al., 2021) plans the facial expression recognition model with the help of artificial intelligence techniques, which mainly includes the steps of data acquisition, face detection, optimal feature extraction and emotion recognition, and uses the optimal scale-invariant feature transform for face expression feature extraction and hybrid metaheuristic algorithm for optimizing the key points that give unique information, which in turn performs facial expression recognition.

In addition, in literature (Zhao et al., 2020), a new deep neural network is constructed to deeply encode the face region and a new face alignment algorithm is proposed. The Literature (Lu, 2021) proposes a multi-angle facial expression recognition method, which is based on generative adversarial network for feature mapping and CNN for classification and learning. The literature (Cao et al., 2021) proposed a method of facial expression recognition by Fourier frequency transform, and obtained the correct facial expression information by adjusting the frequency band of the wrong expression. The Literature (Liao and Gu, 2020) proposes a face recognition method based on subspace extended sparse, which uses subspace extended sparse representation classifier for recognition.

In the latest research on facial expression recognition, the Literature (Li et al., 2017) propose a new DLP-CNN (Deep Locality-Preserving CNN) method, which aims to enhance the discriminative power of deep features by preserving the locality closeness while maximizing the inter-class scatters. In the Literature (Li et al., 2018), a convolutional neural network (CNN) with an attentional mechanism (ACNN) is proposed for facial expression recognition in the field, which can perceive the obscured area of the face and focus on the most discriminating unobscured area. Two versions of ACNN have been introduced: the patch-based ACNN (pACNN) and the global-local ACNN (gACNN). pACNN only focuses on partial facial patches. gACNN combines a local representation at the patch level with a global representation at the image level. The Literature (Chen et al., 2020) propose a novel approach named Label Distribution Learning on Auxiliary Label Space Graphs (LDL-ALSG) that leverages the topological information of the labels from related but more distinct tasks, such as action unit recognition and facial landmark detection. The underlying assumption is that facial images should have similar expression distributions to their neighbors in the label space of action unit recognition and facial landmark detection.

In order to explore the application of NGO-BILSTM model in facial expression recognition, this paper analyzes the debate in three parts. The first part introduces the basic principles of NGO algorithm. In the second part, LSTM neural network is introduced, pointing out that LSTM can not obtain reverse information, and a BILSTM neural network combining forward LSTM and reverse LSTM is introduced. The network can form two independent networks with opposite data flows, and can simultaneously process data with positive and negative flows. Then, according to the NGO algorithm, the hyperparameters of BILSTM were optimized to build the NGO-BILSTM model for facial expression recognition. The third part is the result analysis. According to the constructed NGO-BILSTM facial expression recognition model and the accuracy evaluation index of confusion matrix, three facial expression data sets of FER2013, FERPlus and RAF-DB are provided to evaluate the performance of this model. The validity and feasibility of the proposed model in face expression recognition are measured by the accuracy.

2. NGO-BILSTM model

Based on the NGO algorithm and BILSTM neural network, combining the advantages of the two algorithms, this paper builds a NGO-BILSTM model to provide a better technical basis for artificial intelligence emotional expression.

2.1. NGO algorithm

2.1.1. Initializing the NGO algorithm

In the NGO algorithm, the population number and location of the northern goshawk can be represented by the following population matrix, namely:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{m\times N} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}_{m\times N}$$

In the NGO algorithm, the objective function value of the northern eagle population is represented by a vector, i.e.:

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{1\times N} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{1\times N}$$

2.1.2. NGO's prey identification

In the first stage, the prey selection and aggressive behavior of the northern goshawk is represented by the following mathematical formula:

$$P_i = X_k \tag{1}$$

$$x_{i,j}^{new,P1} = \begin{cases} x_{i,j} + r(p_{i,j} - Ix_{i,j}), F_{p_i} < F_i \\ x_{i,j} + r(x_{i,j} - p_{i,j}), F_{p_i} \geq F_i \end{cases} \tag{2}$$

$$X_i = \begin{cases} X_i^{new,P1}, F_i^{new,P1} < F_i \\ X_i \, , F_i^{new,P1} \geq F_i \end{cases} \tag{3}$$

Where P_i is the prey position, F_{p_i} is the objective function value of P_i, k is a random integer in the range of $[1, N]$. $x_{i,j}^{new,P1}$ is the new position of the jth dimension of the ith northern goshawk, $F_i^{new,P1}$ is the objective function value of the ith northern goshawk based on the first stage update. r belongs to $[0, 1]$, I is 1 or 2.

2.1.3. NGO chase and escape

In the second stage, prey escape and the northern goshawk chasing prey are represented by the following mathematical formula:

$$X_{i,j}^{new,P2} = x_{i,j} + R(2r-1)x_{i,j} \tag{4}$$

$$R = 0.02\left(1-\frac{t}{T}\right) \tag{5}$$

$$X_i = \begin{cases} X_i^{new,P2}, F_i^{new,P2} < F_i \\ X_i \, , F_i^{new,P2} \geq F_i \end{cases} \tag{6}$$

Where t is the current iteration number, is the maximum iteration number (Dehghani et al., 2021).

2.2. BILSTM neural network algorithm

2.2.1. LSTM neural network

LSTM is more efficient because the long-term memory network retains important in-formation for long-term memory and forgets other information to some extent, and sequential data processing is more efficient than recurrent neural networks. The neuron structure of LSTM is shown in Figure 1 (Bao et al., 2021; Zhou et al., 2022).

LSTM and RNN explore the dependencies between sequence elements through internal state transfer. However, LSTM introduces a gating mechanism to solve the short-comings of the RNN gradient update. The gating link of LSTM is divided into forgetting, input and output, and the state unit is introduced to regulate the operation of the whole network.

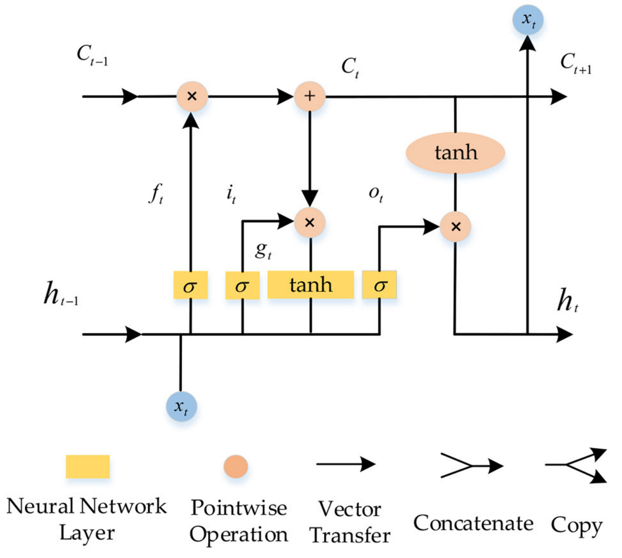


FIGURE 1
Neuronal structure of the LSTM.

2.2.1.1. Oblivious gating

f_t is the forgetting gate, which has the role of determining the degree of retention of the incoming information at the previous moment. The forgetting gate is obtained by linearly transforming the input at moment t with the output at moment $t-1$ and then applying an activation function, which is calculated as:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{7}$$

Where σ denotes the sigmoid activation function, W_f and U_f denote the weight matrix of the forgetting gate, x_t is the input, h_{t-1} is the implicit layer output, b_f denotes the bias value of the forgetting gate.

2.2.1.2. Input gate

i_t is the input gate, it is mainly to decide the retention Chengdu of the information input at t moments. The input gate is calculated in a similar way to the forgetting gate, and its expression is:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{8}$$

Where W_i and U_i denote the input gate weight matrix, b_i denotes the input gate bias value.

g_t is the input state, which is obtained from the implied layer output at moment $t-1$ and the input at moment t by applying a tanh function through a linear transformation, whose expression is:

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \tag{9}$$

Where W_g and U_g denote the temporary cell state weight matrix, denotes the temporary cell state bias value.

2.2.1.3. State unit

The state unit of the LSTM is mainly used to update the internal state of the LSTM at the previous time to the internal state at this time. The formula calculates the internal state at this moment:

$$C_t = C_{t-1} \cdot f_t + g_t \cdot i_t \tag{10}$$

2.2.1.4. Output gate

o_t is the output gate, the output control of the output gate depends on the degree of the state unit, which is calculated as:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{11}$$

Where W_o and U_o denote the weight matrix of the output gate, b_o denotes the bias value of the output gate.

The implied state output h_t at the final moment, which is determined by both the internal state and the output gate, is calculated as:

$$h_t = o_t \cdot \tanh(C_t) \tag{12}$$

2.2.2. BILSTM neural network

BILSTM neural network is proposed based on the LSTM network. BILSTM is composed of forward LSTM and reverse LSTM. The forward LSTM processes input data in the forward direction, while the reverse LSTM processes input data in the reverse direction. After processing, the output of the two LSTMS is joined together, namely, the output of BILSTM. BILSTM can transfer between past and future implied layer states and perform a feedback neural network, which can well uncover the implied connections between time series data. The BILSTM network can find the intrinsic links between the current moment data and the past and future data, which can improve the model testing accuracy and the data utilization efficiency.

Structurally, compared with the one-way LSTM network, the BILSTM neural network is a two-way cyclic structure with forward and backward propagation. In terms of temporal structure, the flow of LSTM data is from the past to the future. In contrast, the flow of BILSTM data is added to the flow of data that will come to the past on top of the flow from the past to the future. The implied layers in the past and the implied layers in the future are independent of each other, so BILSTM can better explore the temporal characteristics of the data. BILSTM structure diagram is shown in Figure 2 (Gong et al., 2021; Hou and Zhu, 2021).

The above figure indicates no interaction between the positive and negative implicit layers, forming two independent networks with opposite data flow directions, which can handle models with both positive and negative flow directions.

The forward LSTM network computational expression is:

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}) \tag{13}$$

The inverse LSTM network computational expression is:

$$\vec{h}_t = LSTM(x_t, \vec{h}_{t-1}) \tag{14}$$

2.3. Face expression recognition model based on NGO-BILSTM

Based on the algorithmic advantages of the NGO algorithm and BILSTM neural network introduced in the previous section, this section combines the research characteristics of face recognition technology and constructs a facial expression recognition model based on NGO-BILSTM.

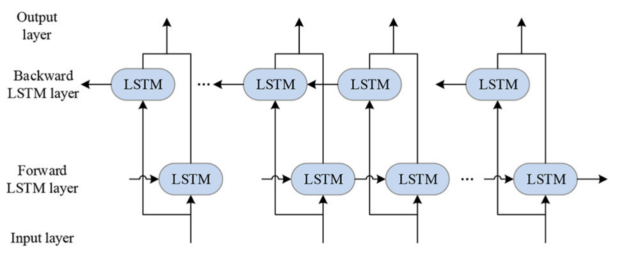


FIGURE 2
BILSTM structure diagram.

2.3.1. Fairness in facial expression recognition

Facial expression recognition technology has become increasingly prevalent in recent years, with applications ranging from security and surveillance to emotion detection in marketing and healthcare. However, concerns about the fairness and accuracy of these systems have also emerged, particularly with respect to their impact on marginalized groups (Buolamwini and Gebru, 2018; Datta and Joshua Swamidass, 2021).

The issue of fairness in facial expression recognition technology is rooted in the fact that these systems are often trained on biased data sets, which can result in the perpetuation of existing societal biases. For example, if a data set used to train a facial recognition system is predominantly composed of images of white individuals, the system may perform poorly when trying to recognize the facial expressions of individuals with darker skin tones. This can result in inaccurate and unfair outcomes, such as misidentifying individuals of color as potential threats or suspects.

Another issue with facial expression recognition technology is that it may not be able to accurately detect or recognize expressions in individuals from certain cultures or backgrounds. For instance, some cultures may have different facial expressions for emotions such as happiness or sadness, and a facial recognition system trained on a data set with limited cultural diversity may struggle to accurately detect or recognize these expressions.

There are ongoing efforts to address the issue of fairness in facial expression recognition technology. One approach is to use more diverse and representative data sets to train these systems, in order to mitigate the impact of biases in the training data. Additionally, there have been calls for greater transparency and accountability in the development and deployment of these systems, including the use of third-party audits and evaluations to ensure that they are accurate and fair for all individuals, regardless of their race, ethnicity, gender, or other factors.

Overall, the issue of fairness in facial expression recognition technology is complex and multifaceted, and requires ongoing attention and effort to address. By working to ensure that these systems are accurate, transparent, and equitable for all individuals, we can help to mitigate the potential harms of biased and unfair technology, and create a more just and equitable society for all.

2.3.2. Face expression data pre-processing

The BILSTM network is used to process the input image and crop out the face region, after which the face region is aligned to reduce the interference of noise. Since the training of the network requires a large amount of data, some datasets with a small number of images, such as the SFEW dataset, cannot be supported for training. Therefore, during the training process, data enhancement, such as random cropping, rotation angle, flipping, etc., is needed for the input data. The data enhancement techniques can effectively enrich the diversity of images in many datasets.

The images obtained after the preprocessing is completed are fed into the BILSTM network for feature extraction. During the training, NGO algorithms were used to optimize the hyperparameters. The input image is passed through the implicit layer for feature extraction and output feature map for feature recognition and classification, and finally, the expression category and recognition accuracy corresponding to the input image are output. The trained network model is loaded and tested using the test set in the network testing phase. The data can be enhanced in the testing phase to improve the robustness of the network model. For the facial expression recognition task, the recognition accuracy is usually used as a criterion to measure a good or bad network model, and the higher the recognition accuracy, the better the performance of the network model and the better the recognition classification.

As fairness is crucial in facial expression recognition, the selection of facial expression data sets should consider whether the data sets include factors such as race, gender, skin color and cultural background. In order to ensure the fairness of face recognition, three data sets FER2013, FERPlus and RAF-DB are selected in this paper. The three data sets are described below:

2.3.2.1. FER2013 dataset

FER2013 is a dataset containing 35,887 images of facial expressions labeled with seven basic emotions (anger, disgust, fear, happiness, sadness, surprise, and neutral). The data set contains facial images of people from different countries and regions, so it covers people of different races, genders, ages and cultural backgrounds. The data distribution of this dataset is shown in Table 1.

2.3.2.2. FERPlus dataset

FERPlus is an expanded version of FER2013 that includes images from FER2013, but for each image, provides more accurate emotional labels, including "uncertain" labels. FERPlus's emotional tags were collected by collecting human tagger tags on Amazon Mechanical Turk and applying model-based methods to filter and clean up. The data distribution of the FERPlus dataset is shown in Table 2.

2.3.2.3. RAF-DB dataset

The RAF-DB dataset contains about 30,000 images of various expressions downloaded from the Web. All the images in the RAFDB dataset differ between subjects in multiple aspects, such as masking of the face, lighting conditions, age, head posture, ethnic skin color, and racial gender. Each faces facial expression image was independently labeled by ∼ 30 to 40 trained coders, so the RAF-DB dataset is rich in images not only in terms of number but also in terms of expression images in various states. The data distribution of the RAF-DB dataset is shown in Table 3.

2.3.3. Optimization of BILSTM network parameters based on the NGO algorithm

The BILSTM network has more parameters, and the parameters that have a greater impact on the facial expression recognition results are the number of LSTM hidden layers, batch size, learning rate, number of iterations, and the parameter selection of Adam optimizer. Before training, the parameters of BILSTM that have a great influence are optimized by using the northern hawk optimization algorithm, and the optimal parameters of the BILSTM network are obtained, and the parameter settings after optimization are shown in Table 4.

2.3.4. NGO-BILSTM model training process

The NGO-BILSTM face expression recognition model is trained by pre-processing the face expression dataset and the BILSTM network after the optimization algorithm of the northern hawk, and the model training process uses the neural network backpropagation algorithm. The flow chart of the facial expression recognition model based on NGO-BILSTM is shown in Figure 3.

Firstly, the face expression dataset is divided into the training set and a test set, and the training set is normalized to the data. Secondly, the combined output values of forward and backward LSTM neurons are calculated according to the forward propagation algorithm. The error terms of the LSTM output layer are calculated according to the loss function, and then back propagated to the forward and backward LSTM implicit layers. The error terms of each LSTM neuron at the end of the implicit layer are calculated. Finally, the gradient of each weight is calculated based on the corresponding error term (Li et al., 2022), and the gradient descent-based optimization algorithm of the Adam optimizer is used to perform the weight update of the LSTM.

The Adam optimizer is an adaptive learning mechanism improved based on SGD architecture, which has the advantage of

TABLE 1 Distribution of expression categories in the FER2013 dataset.

Expression category	Happy	Sadness	Fear	Surprise	Disgust	Anger	Neutral	Total
Training set	3,274	4,286	4,154	3,274	415	4,021	5,032	24,456
Public test set	405	610	532	405	62	489	631	3,134
Private test set	398	647	504	398	64	457	612	3,080

TABLE 2 Distribution of expression categories in the FERPlus dataset.

Expression category	Happy	Sadness	Fear	Surprise	Anger	Disgust	Disdain	Neutrality	Total
Training set	7,246	2,961	501	3,014	1,998	106	112	8,482	24,420
Public test set	854	326	60	405	279	26	18	1,198	3,166
Private test set	892	378	78	394	265	15	18	1,087	3,127

TABLE 3 Distribution of expression categories in the RAF-DB dataset.

Expresion category	Happy	Sadness	Fear	Suprise	Disgust	Anger	Neutral	Total
Training set	4,768	1,991	283	1,288	721	708	2,520	12,279
Test set	1,188	482	78	732	163	165	183	2,991

TABLE 4 Parameter setting after optimization search.

Parameters	Value
LSTM implied layers	128
Batch size	4
Learning rate	0.0005
Number of iterations	300
Adam	$\beta_1 = 0.99, \beta_2 = 0.999$

low dependence on the adaptive learning rate and the assignment of hyperparameters. The loss function is used to measure the difference between the predicted value and the real value. The smaller the value, the better the robustness of the model.

By setting the minimum value of the loss function as the optimization objective, the Adam optimizer updates the BILSTM weights continuously until the optimal face expression recognition model is obtained.

3. Analysis of facial expression recognition results based on the NGO-BILSTM model

Based on the previous design of the facial expression recognition process of NGO-BILSTM, this chapter mainly develops the experimental analysis to clarify the application of the NGO-BILSTM model in facial expression recognition.

3.1. Experimental preparation

3.1.1. Experimental environment and data set

The experimental environment configuration of this chapter for facial expression recognition results is shown in Table 5.

In this chapter, face expression recognition experiments will be conducted on the FER2013 dataset, FERPlus dataset and RAF-DB dataset. The cross-entropy loss is used to optimize the network together with the Adam. The initial learning rate is set to 0.001, the momentum is set to 0.99, and 300 iterations are performed on both the FER2013 dataset, the FERPlus dataset and RAF-DB dataset, and the batch size is set to 15.

3.1.2. Evaluation indicators of the NGO-BILSTM model

In the classification task of machine learning, we often use many metrics to measure the model's performance, such as ROC curve, PSI, recall, accuracy, F1 value, AUC value and confusion matrix. In this paper, we choose the confusion matrix to evaluate the performance of the NGO-BILSTM-based face expression recognition model.

The confusion matrix is the error matrix from which the recognition accuracy can be calculated. The confusion matrix in the classification task is used to reflect the probability that one of the total samples is predicted to be the remaining other samples, and its matrix size is generally $n \times n$, n is the number of categories, and the confusion matrix is shown in Table 6.

Accuracy is the ratio of the classification model's accurate prediction of a certain category of a given test set or the correct proportion of the whole sample predicted by the classification model, which is calculated by the formula:

$$Accuaracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

3.2. Comparison and analysis of experimental results

To verify the effectiveness of the proposed model for facial expression recognition, this section tests and compares three types of face expression datasets, namely, the FER2013 dataset, FERPlus dataset and RAF-DB dataset, to validate the application of this paper's NGO-BILSTM model for facial expression recognition.

3.2.1. FER2013 dataset

To verify the effectiveness of the facial expression recognition model proposed in this paper, the recognition accuracy of the NGO-BILSTM face recognition model constructed in this paper is compared with the traditional VGG16 network on the FER2013 dataset. The confusion matrix of face recognition using the two methods is used as the experimental results, and the comparison results are shown in Figure 4.

The average recognition accuracy of this model in the FER2013 dataset is 51.29%, and the recognition accuracy of the VGG16 network is 45.14%. Compared with the VGG16 network, the recognition accuracy of the proposed NGO-BILSTM face recognition model has improved by 6.15%. The facial expression of "happy" is recognized very well, and the accuracy of "fear" is enhanced by 24% compared with the VGG16 network. The accuracy of "disgust" and "sadness" is still very low, although the accuracy of these two expression categories is slightly improved compared with the VGG16 network because the number of images of these two expression categories in the FER2013 dataset is small, and the network cannot This is because the number of pictures of these two categories of facial expressions in the FER2013 dataset is small, and the network is not fully trained for these two categories

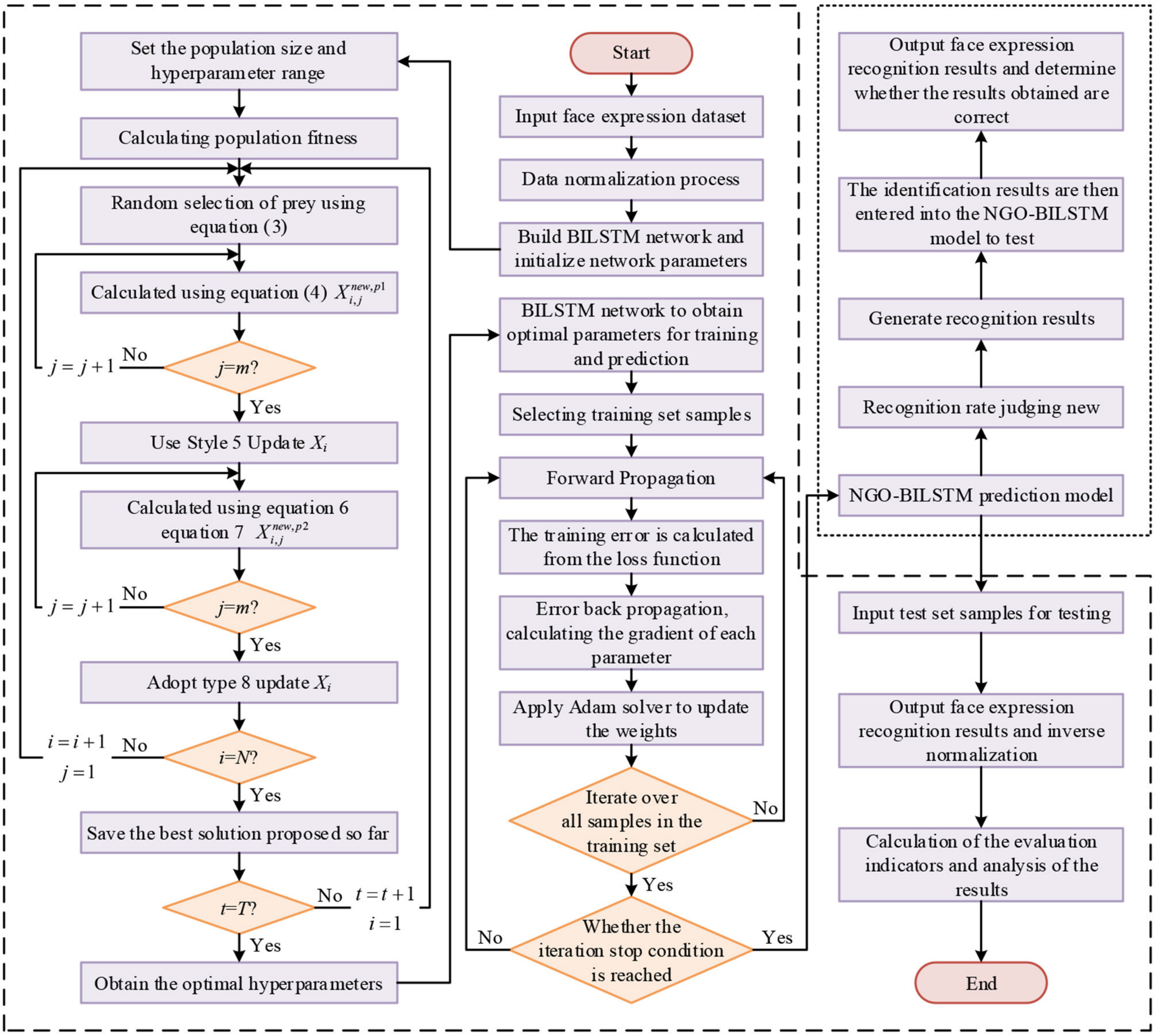

Set the population size and hyperparameter range
Calculating population fitness
Random selection of prey using equation (3)
Calculated using equation (4) $X_{i,j}^{new,p1}$
$j=j+1$
No
j=m?
Yes
Use Style 5 Update X_i
Calculated using equation 6 equation 7 $X_{i,j}^{new,p2}$
$j=j+1$
No
j=m?
Yes
Adopt type 8 update X_i
$i=i+1$
$j=1$
No
i=N?
Yes
Save the best solution proposed so far
t=T?
No
$t=t+1$
$i=1$
Yes
Obtain the optimal hyperparameters
Start
Input face expression dataset
Data normalization process
Build BILSTM network and initialize network parameters
BILSTM network to obtain optimal parameters for training and prediction
Selecting training set samples
Forward Propagation
The training error is calculated from the loss function
Error back propagation, calculating the gradient of each parameter
Apply Adam solver to update the weights
Iterate over all samples in the training set
No
Yes
Whether the iteration stop condition is reached
No
Yes
Output face expression recognition results and determine whether the results obtained are correct
The identification results are then entered into the NGO-BILSTM model to test
Generate recognition results
Recognition rate judging new
NGO-BILSTM prediction model
Input test set samples for testing
Output face expression recognition results and inverse normalization
Calculation of the evaluation indicators and analysis of the results
End

FIGURE 3
Flowchart of facial expression recognition based on NGO-BILSTM.

TABLE 5 Experimental environment configuration.

Environment configuration	Parameter configuration
Operating systems	Windows 10 professional edition
GPU	Intel(R) Core(TM) i3-10100
Memory	8.00 GB
Accelerated libraries	CUDA 9.0
Programming languages	Python 3.8

of facial expressions, so the accuracy of these two categories is relatively low.

This also shows that the model proposed in this paper has a high recognition accuracy and verifies the reliability of the proposed model.

3.2.2. FERPlus dataset

To further verify the effectiveness of the method proposed in this chapter, the recognition accuracy of the NGO-BILSTM face recognition model constructed in this paper is compared with that of the traditional VGG16 network on the FERPlus dataset. The confusion matrix of face recognition using the two methods is used as the experimental results, and the comparison results are shown in Figure 5.

The average recognition accuracy of this model on the FERPlus dataset is 78.75%, and the recognition accuracy of the VGG16 network is 66.63%. Compared with the VGG16 network, the recognition accuracy of the NGO-BILSTM face recognition model proposed in this paper has improved by 12.12%. The accuracy of all the eight expression categories is improved, and the accuracy of "neutral" face expression recognition is the highest, reaching 96%, while the accuracy of "fear" face expression recognition is the

TABLE 6 Confusion matrix.

True value	Predicted value	
	Positive example	Negative example
Positive example	TP	FN
Negative example	FP	TN

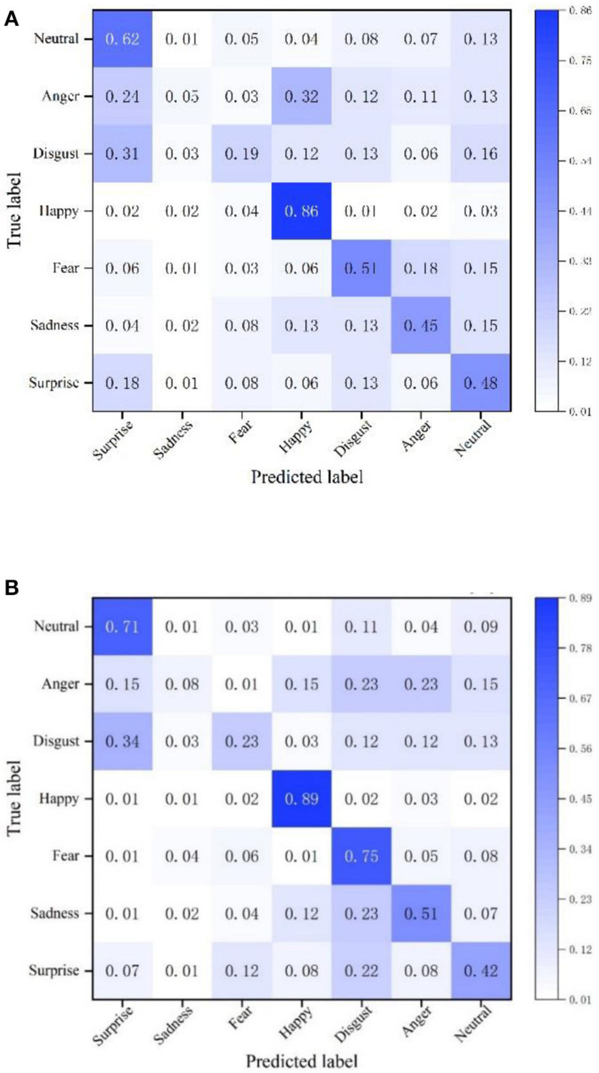


FIGURE 4
Confusion matrix on the FER2013 dataset. **(A)** Confusion matrix of VGG16 network. **(B)** The confusion matrix of the model in this paper.

highest, increasing by 26%. This shows that the proposed NGO-BILSTM-based face expression recognition model has better face expression recognition results on the FERPlus dataset.

3.2.3. RAF-DB dataset

To further verify the effectiveness of the method proposed in this chapter, the recognition accuracy of the NGO-BILSTM face recognition model constructed in this paper is compared with that of DLP-CNN, GACNN, PACNN, and LDL-ALSG on the RAF-DB dataset, and the comparison results are shown in Table 7.

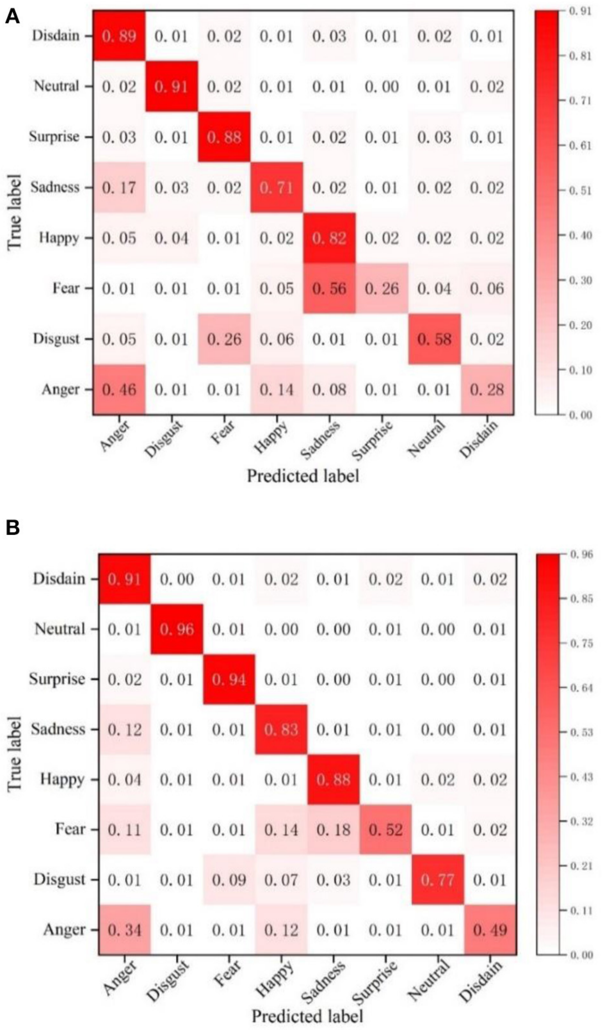


FIGURE 5
Confusion matrix on the FERPlus dataset. **(A)** Confusion matrix of VGG16 network. **(B)** The confusion matrix of the model in this paper.

The recognition accuracy of the NGO-BILSTM face expression recognition model proposed in this paper is 89.72% on the RAF-DB dataset, which is 5.45, 9.63, 7.36, and 3.18% higher than those of the four methods DLP-CNN, gACNN, pACNN, and LDL-ALSG on the RAF-DB dataset, respectively. This indicates that the facial expression recognition model based on NGO-BILSTM in this paper has higher recognition accuracy and verifies the reliability of the proposed model in this paper.

4. Conclusion

To explore the application of the NGO-BILSTM model in facial expression recognition, this paper constructs the NGO-BILSTM face expression recognition model based on the NGO

TABLE 7 Accuracy of different models on the RAF-DB dataset.

Network model	Accuracy rate (%)
DLP-CNN	84.27
GACNN	80.09
PACNN	82.36
LDL-ALSG	86.54
NGO-BILSTM	89.72

algorithm and BILSTM neural network and uses the loss function with Adam optimizer for weight update. For the effectiveness of the model in this paper, the three face expression datasets of FER2013, FERPlus and RAF-DB are evaluated by the accuracy of the confusion matrix, and the experimental results are as follows:

(1) The average recognition accuracy of this paper's model on the FER2013 dataset is 51.29%, and the recognition accuracy of the VGG16 network is 45.14%. Compared with the VGG16 network, the recognition accuracy of the model proposed in this paper is improved by 6.15%.
(2) The average recognition accuracy of the NGO-BILSTM model proposed in this paper on the FERPlus dataset is 78.75%, and the recognition accuracy of the VGG16 network is 66.63%. Compared with the VGG16 network, the recognition accuracy of the proposed model in this paper is improved by 12.12%.
(3) The identification accuracy of the NGO-BILSTM model proposed in this paper is 89.72% on the RAF-DB dataset, which is 5.45, 9.63, 7.36, and 3.18% higher than the recognition accuracy of the four methods DLP-CNN, gACNN, pACNN, and LDL-ALSG on the RAF-DB dataset, respectively.

This shows that the NGO-BILSTM-based facial expression recognition model proposed in this paper has high recognition accuracy and can be effectively used in facial expression recognition applications.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

JZ and TC contributed to conception and design of the study. JZ and LY organized the database, performed the statistical analysis, and wrote sections of the manuscript. JZ wrote the first draft of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This work was supported by National Natural Science Foundation of China Project No. 61603212.

Acknowledgments

This paper is under the guidance of the teacher and the help of the students to complete. I would like to express my heartfelt thanks to them for their careful help and guidance. During this period of time, they gave me great help, which enabled me to have an overall grasp of the whole idea of the research, and patiently helped me solve many practical problems, so that I have a great harvest. At the same time, they made a lot of constructive comments throughout the development process and solved some professional problems for me. I not only learned a lot of professional knowledge from them, but also felt that they are cautious and conscientious in their work and approachable in life. In addition, their rigorous academic attitude and selfless work spirit are worthy of my learning.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Bao, Z., Wei, Q., Zhou, T., Jiang, X., and Watanabe, T. (2021). Predicting stock high price using forecast error with recurrent neural network. *Appl. Math. Nonlinear Sci.* 6:283–292. doi: 10.2478/amns.2021.2.00009

Buolamwini, J., and Gebru, T. (2018). "Gender shades: intersectional accuracy disparities in commercial gender classification." in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency, Vol. 81 of Proceedings of Machine Learning Research*, eds Sorelle A. Friedler and Christo Wilson (New York, NY, USA), 77–91.

Cao, T., Liu, C., Chen, J., and Gao, L. (2021). Nonfrontal and asymmetrical facial expression recognition through half-face frontalization and pyramid fourier frequency conversion. *IEEE Access.* 9, 17127–17138. doi: 10.1109/ACCESS.2021.3052500

Chen, S., Wang, J., Chen, Y., Shi, Z., Geng, X., and Rui, Y. (2020). "Label distribution learning on auxiliary label space graphs for facial expression recognition." in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, 13984–13993. doi: 10.1109/CVPR42600.2020.01400

Datta, A., and Joshua Swamidass, S. (2021). Fair-Net: a network architecture for reducing performance disparity between identifiable sub-populations. *arXiv preprint arXiv:210600720*. (2021). doi: 10.5220/0010877400003116

Dehghani, M., Hubálovský, Š., and Trojovský, P. (2021). Northern goshawk optimization: a new swarm-based algorithm for solving optimization problems. *IEEE Access*. 9, 162059–162080. doi: 10.1109/ACCESS.2021.3133286

Gao, J., Li, L., and Guo, B. A. (2020). New extendface representation method for face recognition. *Neural Process. Lett.* 51, 473–486. doi: 10.1007/s11063-019-10100-1

Gong, L., Zhang, X., Chen, T., and Zhang, L. (2021). Recognition of disease genetic information from unstructured text data based on BiLSTM-CRF for molecular mechanisms. *Secur. Commun. Netw*. 2021, 1–8. doi: 10.1155/2021/6635027

Gurukumar, L., Harinatha, R. G., and Giri, P. (2021). Optimized scale-invariant feature transform with local tri-directional patterns for facial expression recognition with deep learning model. *Comput. J*. 65:2506–2527.

Han, J., Du, L., Ye, X., Zhang, L., and Feng, J. (2022). The devil is in the face: exploiting harmonious representations for facial expression recognition. *Neurocomputing*. 486, 104–113. doi: 10.1016/j.neucom.2022.02.054

Hou, J., and Zhu, A. (2021). Fake online review recognition algorithm and optimisation research based on deep learning. *Appl. Math. Nonlinear Sci*. 7, 861–874. doi: 10.2478/amns.2021.2.00170

Li, G., Jian, X., Wen, Z., and AlSultan, J. (2022). Algorithm of overfitting avoidance in CNN based on maximum pooled and weight decay. *Appl. Math. Nonlinear Sci.* 7:965–974. doi: 10.2478/amns.2022.1.00011

Li, K., Jin, Y., Akram, M. W., Han, R., and Chen, J. (2020). Facial expression recognition with convolutional neural networks *via* a new face cropping and rotation strategy. *Vis. Comput.* 36, 391–404. doi: 10.1007/s00371-019-01627-4

Li, S., and Deng, W. (2020). Deep facial expression recognition: a survey. *IEEE Trans. Affect. Comput*. 13:1195–1215. doi: 10.1109/TAFFC.2020.2981446

Li, S., Deng, W., and Du, J. P., (2017). Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. *2017 Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 2852–2861. doi: 10.1109/CVPR.2017.277

Li, Y, Zeng, J, Shan, S, Chen, X. (2018). Occlusion aware facial expression recognition using CNN with attention mechanism. *IEEE Trans. Image Process.* 28, 2439–2450. doi: 10.1109/TIP.2018.2886767

Liao, M., and Gu, X. (2020). Face recognition approach by subspace extended sparse representation and discriminative feature learning. *Neurocomputing.* 373, 35–49. doi: 10.1016/j.neucom.2019.09.025

Lu, L. (2021). Multingle face expression recognition based on generative adversarial networks. *Comput. Intell.*

Minaee, S., Minaei, M., and Abdolrashidi, A. (2021). Deep-emotion: facial expression recognition using attentional convolutional network. *Sensors.* 21, 3046. doi: 10.3390/s21093046

Prajapati, H. B., Vyas, A. S., and Dabhi, V. K. (2021). Concise CNN model for face expression recognition. *Intell. Decis. Technol.* 15, 179–187. doi: 10.3233/IDT-190181

Ramachandran, B., and Rajagopal, S. D. (2022). 3D face expression recognition with ensemble deep learning exploring congruent features among expressions. *Comput. Intell.* 38, 345–365. doi: 10.1111/coin.12498

Ravinder, M., Malik, K., Hassaballah, M., Tariq, U., Javed, K., Ghoneimy, M., et al. (2022). An approach for gesture recognition based on a lightweight convolutional neural network. *Int. J. Artif. Intell. Tools*. doi: 10.1142./S0218213023400146

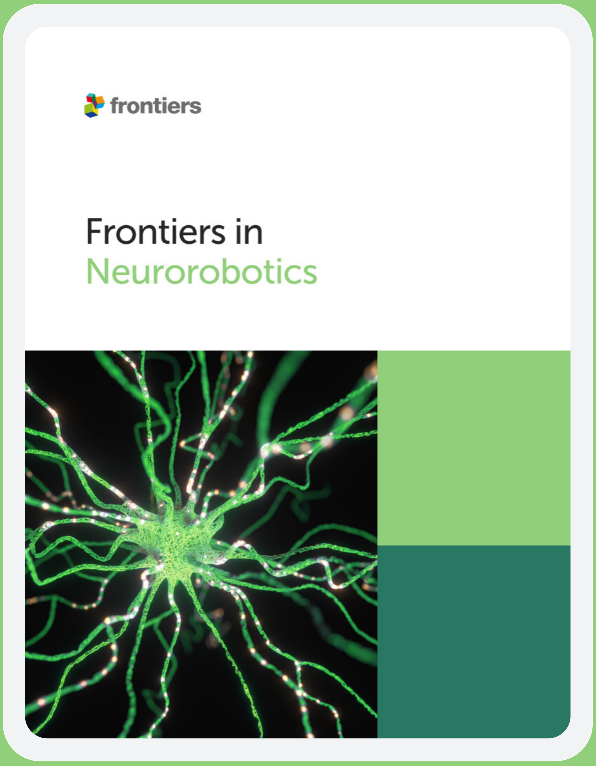
Revina, I. M., and Emmanuel, W. R. S. (2021). A survey on human face expression recognition techniques. *J. King Saud Univ. - Comput. Inf. Sci.* 33, 619–628. doi: 10.1016/j.jksuci.2018.09.002

Yang, H., Zhu, K., Huang, D., Li, H., Wang, Y., and Chen, L. (2021). Intensity enhancement *via* GAN for multimodal face expression recognition. *Neurocomputing.* 454. doi: 10.1016/j.neucom.2021.05.022

Zhang, F, Zhang, T, Mao, Q, and Xu, C. (2020). A unified deep model for joint facial expression recognition, face synthesis, and face alignment. *IEEE Trans. Image Process.* 29, 6574–6589. doi: 10.1109/TIP.2020.2991549

Zhao, F., Li, J., Zhang, L., Li, Z., Na, S-G. (2020). Multi-view face recognition using deep neural networks. *Future Gener. Comput. Syst*. 111, 375–380. doi: 10.1016/j.future.2020.05.002

Zhou, Y., Lin, Q., and Xiao, D., (2022). Application of LSTM-LightGBM nonlinear combined model to power load forecasting. *J. Phys.: Conf. Ser*. doi: 10.1088/1742-6596/2294/1/012035



frontiers

Frontiers in
Neurorobotics

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact