TYPE Original Research
PUBLISHED 23 March 2023
DOI 10.3389/fphy.2023.1153637

Lightweight neural architectures to improve COVID-19 identification

Mohammad Mehedi Hassan[1]*, Salman A. AlQahtani[1], Abdulhameed Alelaiwi[1] and João P. Papa[2]

[1]College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia, [2]Department of Computing, São Paulo State University, Bauru, Brazil

Check for updates

OPEN ACCESS

EDITED BY
Zhihan Lv,
Uppsala University, Sweden

REVIEWED BY
Zhihui Lu,
Fudan University, China
Akshi Kumar,
Manchester Metropolitan University,
United Kingdom

*CORRESPONDENCE
Mohammad Mehedi Hassan,
mmhassan@ksu.edu.sa

SPECIALTY SECTION
This article was submitted to Medical Physics and Imaging,
a section of the journal
Frontiers in Physics

RECEIVED 29 January 2023
ACCEPTED 13 March 2023
PUBLISHED 23 March 2023

CITATION
Hassan MM, AlQahtani SA, Alelaiwi A and Papa JP (2023), Lightweight neural architectures to improve COVID-19 identification.
Front. Phys. 11:1153637.
doi: 10.3389/fphy.2023.1153637

The COVID-19 pandemic has had a global impact, transforming how we manage infectious diseases and interact socially. Researchers from various fields have worked tirelessly to develop vaccines on an unprecedented scale, while different countries have developed various sanitary protocols to deal with more contagious variants. Machine learning-assisted diagnosis has emerged as a powerful tool that can help health professionals deliver faster and more accurate outcomes. However, medical systems that rely on deep learning often require extensive data, which may be impractical for real-world applications. This paper compares lightweight neural architectures for COVID-19 identification using chest X-rays, highlighting the strengths and weaknesses of each approach. Additionally, a web tool has been developed that accepts chest computer tomography images and outputs the probability of COVID-19 infection along with a heatmap of the regions used by the intelligent system to make this determination. The experiments indicate that most lightweight architectures considered in the study can identify COVID-19 correctly, but further investigation is necessary. Lightweight neural architectures show promise in computer-aided COVID-19 diagnosis using chest X-rays, but they did not reach accuracy rates above 88%, which is necessary for medical applications. These findings suggest that additional research is necessary to improve the accuracy of lightweight models and make them practical for real-world use.

KEYWORDS
COVID-19, convolutional neural networks, deep learning, heatmap analyses, web tool

1 Introduction

The COVID-19 pandemic has significantly impacted our world, and it has required rapid adaptation to new ways of social interaction. The widespread use of facial masks, barrier face coverings, gloves, and other sanitary practices have been introduced on an unprecedented scale. Although it appears relatively controlled, COVID-19 is still prevalent, and new outbreaks are continually being reported. The good news is the rapid development of vaccines, but there are also concerns regarding their efficacy.

In the effort to better understand the internal mechanisms of intelligent systems and to provide faster COVID-19 diagnosis, machine learning-aided diagnosis has emerged. There is a vast body of literature on this topic, with many exciting works. For example, Hassan et al.Hassan et al. [1] introduced Deep Taylor Decomposition Montavon et al. [2] to explain COVID-19 recognition from chest X-ray images. The experiments included the well-known Composite Layer-wise Propagation Samek et al. [3] and Single Taylor Decomposition for comparison purposes concerning selectivity, continuity, and input perturbation. Santos et al.Santos et al. [4] used Efficient Nets Tan and Le [5], specifically EfficientNet-B6, to improve COVID-19 identification. The authors showed improvements using a min-max normalization, outperforming COVID-Net.

Although deep learning techniques have shown promising results, they require a large amount of data Hu et al. [6]; Ohata et al. [7]; de Mesquita et al. [8]; Parah et al. [9]. Convolutional Neural Networks (CNNs) LeCun et al. [10] are crucial in this area, but deeper architectures come at a high computational cost, which can be unfeasible for real-world applications. Hence, researchers have been pursuing lightweight architectures to address this shortcoming Gumaei et al. [11]; Hassan et al. [12]; Alsahli et al. [13]. These lightweight architectures, despite having lower performance, are more efficient and can be embedded in sensors and portable devices Hassan et al. [14].

Polsinelli et al.Polsinelli et al. [15] proposed a lightweight CNN based on Squeeze Net Iandola et al. [16] for efficient discrimination between COVID-19, pneumonia, and healthy individuals using chest computed tomography (CT) images. Their work outperformed the original SqueezeNet and could analyze thousands of images daily, even with limited hardware resources. Zhang et al. Zhang et al. [17] employed MobileNetV2 and transfer learning to address the shortage of computational power in some environments. They fine-tuned the ImageNet-trained backbone on a CT dataset provided by Kaggle and showed that their proposed system can analyze and diagnose a chest CT image in 1.06 s on a computer without a dedicated graphics processing unit (GPU).

Paluru et al.Paluru et al. [18] proposed a lightweight CNN-based strategy called Anam-Net to segment anomalies in chest CT images affected by COVID-19. The approach has 7.8 times fewer parameters than U-Net Ronneberger et al. [19], making it suitable for mobile devices or resource-constrained platforms. State-of-the-art results were reported in terms of the Dice similarity score, with Anam-Net deployed in a Raspberry Pi equipped with an NVIDIA Jetson Xavier Agx. In another study, Iyer et al. Iyer et al. [20] compared four CNN models, including U-Net, SegNet, HR Net, and VGG Net, to segment CT images from patients affected by COVID-19. Using their dataset with more than 3,000 images, the authors found that HR Net was the most effective model, achieving an accuracy of 96.24% and a Dice score of 0.91. The work highlighted that lightweight models are adequate for infectious tissue segmentation on CT slices.

Huang and Liao Huang and Liao [21] proposed LightEfficientNetV2, which aimed to improve accuracy and speed up the training step with fewer parameters. The approach used chest CT images to distinguish between COVID-19, pneumonia, and healthy individuals, and promising accuracies were obtained in distinct datasets. In addition, Zhao et al. Zhao et al. [22] presented LCOV-Net, a lightweight 3D CNN for image segmentation of COVID-19 pneumonia lesions from CT volumes. An attention-based convolutional block played a significant role in the approach, and it consisted of a spatiotemporal separable convolution module to reduce parameters and a feature calibration to improve learning. The proposed approach was tested on a dataset with 130 patients affected by COVID-19 for lesion segmentation, with a reported speed up of 27.93% compared to a standard 3D U-Net. The central core of the proposed approach was a 3D UNet.

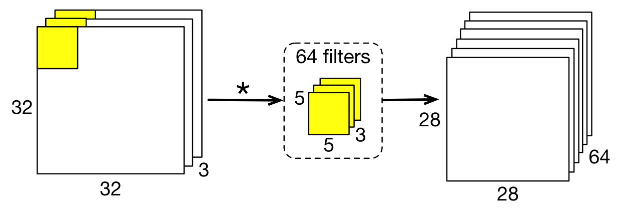


FIGURE 1
Input image with 32 × 32 pixels and three channels convolved with 64 filters of size 5 × 3 × 3, ending up in 5 × 5 × 3 × 32 × 32 × 64 = 4, 915, 200 multiplication operations. The input channels are convolved with each filter separately and further integrated.

Security issues are a major concern when dealing with data from different organizations. Heiradi et al. Heidari et al. [23] presented an approach based on blockchain and lightweight CNNs to address COVID-19 detection. Given data from different hospitals, the authors trained a global model while maintaining the confidentiality of the institutions. State-of-the-art results were reported. Kamal et al. Kamal et al. [24] emphasized the importance of using the Internet of Things (IoT) framework to combat COVID-19. Digital telehealth has expanded the frontiers of artificial intelligence and remote diagnosis, but it has also brought new issues such as the need for larger bandwidth and better security. The authors suggested that lightweight security and intelligent systems are a few solutions that should be explored to prevent the further spread of COVID-19.

The literature lacks a more substantial comparison among lightweight neural architectures for COVID-19 identification, as most works focus on lesion segmentation. Therefore, our main scientific contribution is to address this gap and provide a web tool that allows users to upload CT images from the chest and receive two outputs: 1) the probability of being affected by COVID-19 concerning different models (the user can pick one), and 2) a heatmap with the regions used by the intelligent tool to make its decision. Although CT scans are more informative than X-rays, the latter is cheaper and can be used in remote locations.

In summary, this manuscript has two main contributions:

- A comprehensive comparison of lightweight neural architectures for COVID-19 identification; and
- A web tool for analyzing chest CT images.

The remainder of the manuscript is organized as follows. Section 2 provides a brief background on the neural architectures used in this work, and Section 3 discusses the methodology, evaluation measures, dataset, and the neural architectures considered. Section 4 presents the experiments and the web tool, while Section 5 summarizes the conclusions and outlines potential future work.

2 Theoretical background

This section presents a brief theoretical background of the neural architectures used in the manuscript.

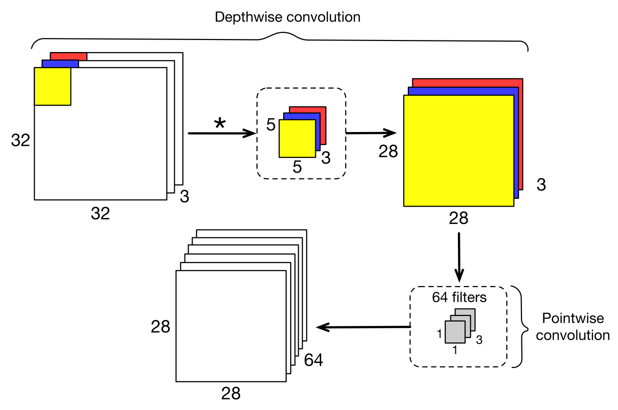


FIGURE 2
The input 32 × 32 × 3 image is convolved with a 5 × 5 × 3 kernel, ending up in a 28 × 28 × 3 output (76,800 multiplication operations). Further, the feature maps are convolved with 64 filters of size 1 × 1 × 3, ending up in 150,528 multiplication operations, summing 76, 800 + 150, 528 = 227, 328 operations.

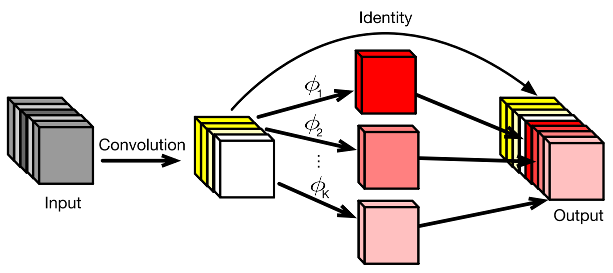


FIGURE 4
The idea behind the ghost features: cheap and linear operations generate redundant feature maps instead of convolutional modules.

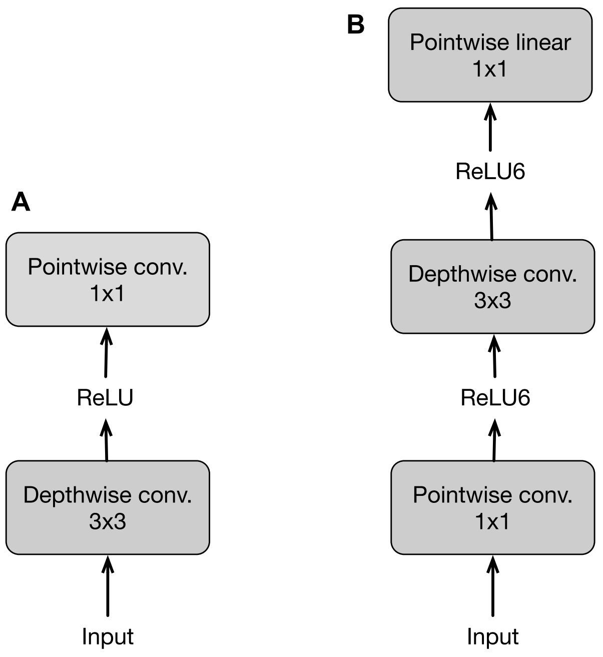


FIGURE 3
The MobileNetV1 **(A)** and MobileNetV2 **(B)** architectures: ReLU6 is used in MobileNetV2 instead of the standard ReLU.

2.1 MobileNet

Convolutional layers play a crucial role in deep neural networks, but they can become computationally expensive due to the large number of matrix multiplications involved. To address this, MobileNet Howard et al. [25] was introduced as a CNN architecture that is typically faster than conventional CNNs. This is achieved by using a convolutional layer called depthwise separable convolution. The main difference between standard (two-dimensional) and depthwise convolutions is that the former is performed over all input channels, while the latter treats each channel separately. Figure 1 illustrates the concept of a standard convolution.

The concept of depthwise separable convolution starts with the idea of *spatial separable convolutions*, which involves performing two sequential one-dimensional convolutions instead of a single two-dimensional one (as illustrated in Figure 2). However, the primary issue with this approach is that not all kernels can be separated into two smaller ones. An alternative approach is depthwise separable convolution, which works with kernels that cannot be factored into smaller kernels. The idea behind depthwise separable convolution is to perform two different convolutions on the input kernel: depthwise convolution and pointwise convolution. In the depthwise convolution, a kernel operates over each channel separately, addressing the depth dimension (i.e., the number of channels). To aggregate each feature map, a pointwise convolution is then performed. MobileNet Howard et al. [25] is a CNN architecture that uses this approach and is usually faster than conventional CNNs.

Other versions of MobileNet are also available, such as MobileNetV2 Sandler et al. [26] and MobileNetV3 Howard et al. [27]. MobileNetV2 introduces two new features to its previous version: 1) *linear bottlenecks* between layers and 2) *shortcut connections* between the bottlenecks. The motivation behind MobileNetV2 is to address the issue of non-linearities encoded by ReLU, which can cause the input channel to collapse by assigning zero values to the activation units' outputs. The authors in Sandler et al. [26] demonstrated experimentally that incorporating linear bottleneck layers within the convolutional blocks can prevent non-linearities from destroying too much information. Additionally, MobileNetV2 employs residual connections between the bottlenecks, called inverted residuals. Figure 3 illustrates the architectures of both MobileNetV1 and MobileNetV2.

The MobileNetV3 architecture is built on the same core architecture used in MobileNetV2 but introduces some new features. The architecture uses *squeeze-and-excite* blocks and *swish* activation functions in the residual layers. The authors of Howard et al. [27] demonstrated that MobileNetV3 outperforms many existing models in various tasks, achieving state-of-the-art results.

2.2 GhostNet

Han et al.Han et al. [28] proposed GhostNet, which aims to reduce redundancy in feature maps produced by deep neural

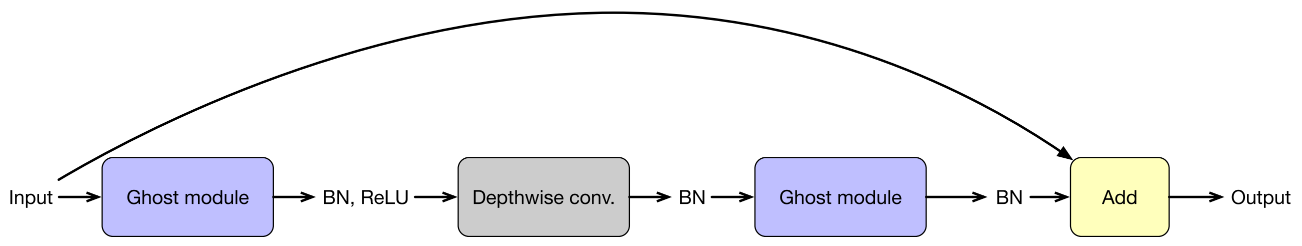


FIGURE 5
Ghost bottlenecks for stride value greater or equal to 2.

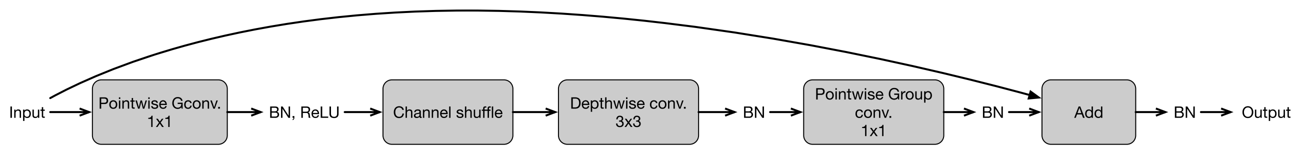


FIGURE 6
ShuffleNet unit with no stride (i.e., stride = 1).

networks. The authors argued that while redundancy is important, many feature maps are similar or even identical to each other. They refer to these similar feature maps as "ghosts" of each other. The authors further noted that generating such redundant maps with convolutional modules is unnecessary, as it requires a large amount of computation and parameters. Figure 4 illustrates the idea behind GhostNet.

GhostNet generates these redundant feature maps, called *ghost features*, using inexpensive linear operations that operate on each channel with lower computational cost than standard convolutional operations. The authors of Han et al. [28] also introduced the Ghost bottleneck, specially designed for small-sized networks. Similar to a ResNet residual block He et al. [29], the proposed bottleneck layer consists of two stacked Ghost modules: 1) the first acts as an expansion layer, increasing the number of channels, and 2) the second reduces the number of channels to match the shortcut path that connects the inputs and outputs of these Ghost modules. Batch normalization and ReLU are applied after each layer, except that ReLU is not employed after the second Ghost module. Figure 5 illustrates the Ghost bottleneck.

2.3 ShuffleNet

ShuffleNet was proposed by Zhang et al. Zhang et al. [30], and it employs two main operations, namely, *group convolution* and *channel shuffle*. Group convolution allows data from different channels to strengthen the representation, whereas channel shuffle facilitates the efficient implementation of group convolution. Additionally, channel shuffle is differentiable, which means it can be embedded into different architectures for end-to-end training.

A *ShuffleNet unit* comprises a pointwise group convolution, followed by a channel shuffle operation. The second pointwise operation aims to recover the channel dimension and match the shortcut path. Batch normalization and ReLU are also employed. Figure 6 illustrates a ShuffleNet unit with no stride.

3 Methodology

This section describes the dataset used in the experiments, the lightweight architectures, the experimental setting, and the measures used for evaluation purposes.

3.1 Dataset

We used the "COVID-19 Radiography Dataset"[1], which comprises 15,153 chest X-ray images divided into three classes: 1) 1,345 images positive to viral pneumonia, 2) 3,616 images positive to COVID-19, and 3) 10,192 images from healthy people. Figure 7 depicts some examples from the dataset.

3.2 Neural architectures

We considered five deep neural architectures that use different mechanisms for training purposes:

- MobileNetV1 Howard et al. [25]: It receives 224 × 224 RGB images that are forwarded to a 3 × 3 convolutional layer with

1 Available at https://www.kaggle.com/tawsifurrahman/covid19-radiography-database.

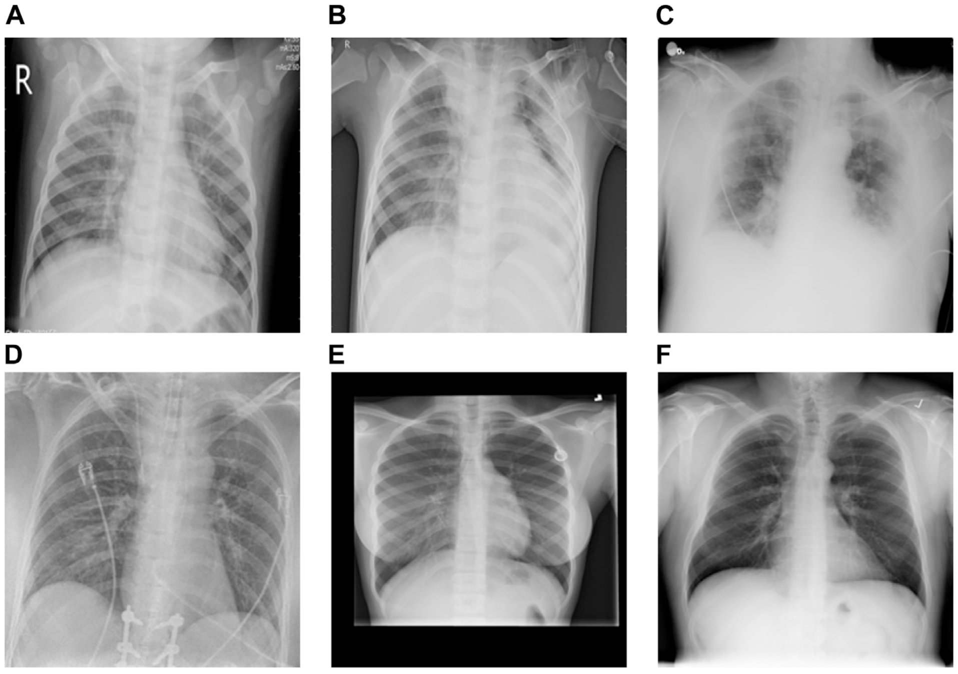


FIGURE 7
Some samples from the dataset: **(A,B)** images positive to viral pneumonia, **(C,D)** images positive to COVID-19, and **(E,F)** images from healthy people.

stride fixed to 2 pixels that outputs 32 feature maps; further, three blocks composed of a depthwise convolution with a 3 × 3 × 32 kernel, stride set to 1, pointwise convolution with a 1 × 1 × 32 × 64 kernel (stride set to 1), a depthwise convolution with 3 × 3 × 32 kernel, stride set to 2, and another pointwise convolution with a 1 × 1 × 32 × 64 kernel (stride set to 1); 5 convolutional blocks composed of a depthwise convolution with a 3 × 3 × 512 kernel (stride set to 1), followed by a pointwise convolution with a 1 × 1 × 512 × 512 kernel (stride set to 1); two blocks of a depthwise convolutional with a 3 × 3 × 512 kernel (stride set to 2) followed by a pointwise convolution with a 1 × 1 × 512 × 1, 024 kernel (stride set to 1); average pooling with a 7 × 7 kernel (stride set to 1); and a softmax layer with 1,280 inputs and 1,000 outputs, for it has been designed to be trained on ImageNet dataset.

- MobileNetV2 Sandler et al. [26]: It receives 224 × 224 RGB images that are forwarded to a 3 × 3 convolutional layer with stride fixed to 2 pixels that outputs 32 feature maps; further, 7 bottleneck layers are applied for a subsequently 3 × 3 convolution with stride set to 1 pixel; average pooling with a 7 × 7 window is employed for the further application of a softmax layer with 1,280 inputs and 1,000 outputs, for it has been designed to be trained on ImageNet dataset.
- MobileNetV3[2] Howard et al. [27]: It receives 224 × 224 RGB images that are forwarded to a 3 × 3 convolutional layer with stride fixed to 2 pixels that outputs 16 feature maps; further, three blocks composed of bottleneck with a 3 × 3 kernel, 8 blocks composed of bottleneck with a 5 × 5 kernel, a pointwise convolution (stride set to 1); average pooling with a 7 × 7 kernel (stride set to 1), two blocks composed of a pointwise convolution (stride set to 1) with no batch normalization; and a softmax layer with 1,024 inputs and 1,000 outputs, for it has been designed to be trained on ImageNet dataset.
- GhostNet Han et al. [28]: It receives 224 × 224 RGB images that are forwarded by a first convolutional layer of size 3 × 3; the following 16 modules stand for Ghost bottlenecks for the further application of a 1 × 1-sized convolutional operation, then an average pooling with a window of size 7 × 7 and another 1 × 1-sized convolutional operation are performed; a fully connected layer receives 1,280 features and outputs 1,000 neurons since it has been designed to be trained on ImageNet dataset; the last layer stands for a softmax layer.
- MobileNetV1 Howard et al. [25]: It receives 224 × 224 RGB images that are forwarded to a 3 × 3 convolutional layer with stride fixed to 2 pixels, outputting 32 feature maps. Then, three blocks are applied, each consisting of a depthwise convolution with a 3 × 3 × 32 kernel, a stride set to 1, followed by a pointwise convolution with a 1 × 1 × 32 × 64 kernel and stride set to 1. Afterward, five convolutional blocks are applied, each consisting of a depthwise convolution with a 3 × 3 × 512 kernel (stride set to 1), followed by a pointwise convolution with a 1 × 1 × 512 × 512 kernel (stride set to 1). Finally, two blocks of a depthwise convolutional layer with a 3 × 3 × 512 kernel (stride set to 2) followed by a pointwise convolutional layer with a 1 × 1 × 512 × 1, 024 kernel (stride set to 1) are

2 We considered the MobileNetV3-Small in this manuscript.

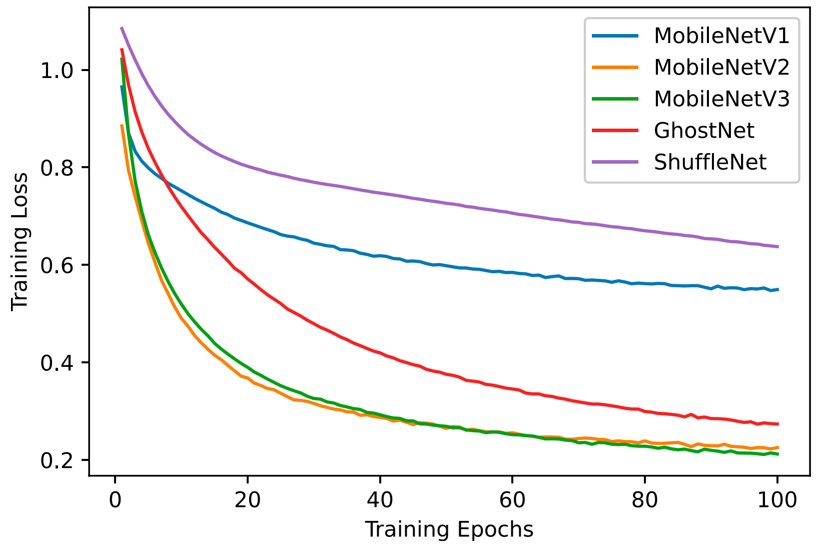


FIGURE 8
Training step loss.

TABLE 1 Training time over the COVID-19 Radiography Dataset.

Model	Training time	Times faster
MobileNetV1	89 m 58 s	2.3×
MobileNetV2	112 m 47 s	2.9×
MobileNetV3	37 m 57 s	-
GhostNet	78 m 49 s	2.0×
ShuffleNet	42 m 39 s	1.12×

applied, along with average pooling with a 7 × 7 kernel (stride set to 1) and a softmax layer with 1,280 inputs and 1,000 outputs. It was designed to be trained on the ImageNet dataset.

- MobileNetV2 Sandler et al. [26]: It receives 224 × 224 RGB images that are forwarded to a 3 × 3 convolutional layer with stride fixed to 2 pixels, outputting 32 feature maps. Then, seven bottleneck layers are applied for subsequent 3 × 3 convolutions with stride set to 1 pixel. Next, average pooling with a 7 × 7 window is employed, followed by a softmax layer with 1,280 inputs and 1,000 outputs. It was designed to be trained on the ImageNet dataset.
- MobileNetV3[3]Howard et al. [27]: It receives 224 × 224 RGB images that are forwarded to a 3 × 3 convolutional layer with stride fixed to 2 pixels, outputting 16 feature maps. Then, three blocks of bottlenecks with a 3 × 3 kernel are applied, followed by eight blocks of bottlenecks with a 5 × 5 kernel, and a pointwise convolution (stride set to 1). Next, average pooling with a 7 × 7 kernel (stride set to 1) is applied, followed by two blocks of pointwise convolution (stride set to 1) with no batch normalization. Finally, a softmax layer with 1,024 inputs and 1,000 outputs is applied. It was designed to be trained on the ImageNet dataset.
- Ghost Net Han et al. [28]: It receives 224 × 224 RGB images that are forwarded by a first convolutional layer of size 3 × 3. The following 16 modules are Ghost bottlenecks, which involve a 1 × 1-sized convolutional operation, average pooling with a 7 × 7 window, and another 1 × 1-sized convolutional operation. A fully connected layer receives 1,280 features and outputs 1,000 neurons since it was designed to be trained on ImageNet dataset; the last layer stands for a softmax layer.
- ShuffleNet Zhang et al. [30]: It receives 224 × 224 RGB images that are passed through a 3 × 3 convolutional layer with stride fixed to 2 pixels, resulting in 24 feature maps. Then a 56 × 56 max-pooling is applied. The network consists of a stack of ShuffleNet units grouped into three stages, where the first building block in each stage uses a stride of 2 and the second uses a stride of 1. The final layers include global pooling with a 7 × 7 kernel, a fully connected layer with 1,000 output neurons, and a softmax layer with 1,000 outputs. This architecture is designed to be trained on the ImageNet dataset.

3.3 Experimental setting

Since deep neural architectures are being used, data augmentation is performed on the training dataset to double its size by horizontally flipping every training image. This transformation does not affect the natural appearance of the images since we are dealing with chest X-ray data.

Out of the 15,153 images, 12,122 are used to compose the training set (80%), and the remaining 3,031 images are used as the test set. Although our goal is not to outperform state-of-the-art approaches in terms of COVID-19 identification, we understand

3 We considered the MobileNetV3-Small in this manuscript.

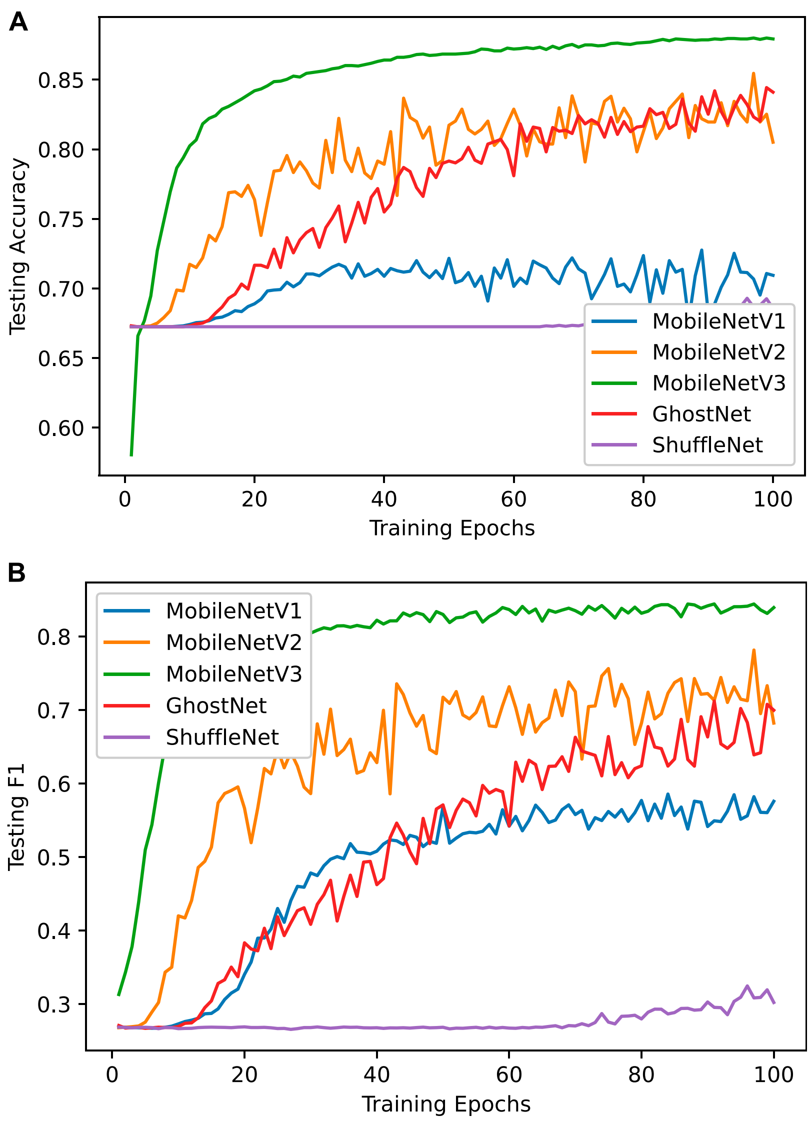


FIGURE 9
Testing set validation over different epochs: **(A)** accuracy and **(B)** F1-Score.

TABLE 2 Accuracy and F1-Score values concerning all models over the testing set.

Model	Accuracy	F1-score
MobileNetV1	0.7275	0.5744
MobileNetV2	0.8545	0.7814
MobileNetV3	0.8799	0.8441
GhostNet	0.8443	0.7076
ShuffleNet	0.6929	0.3246

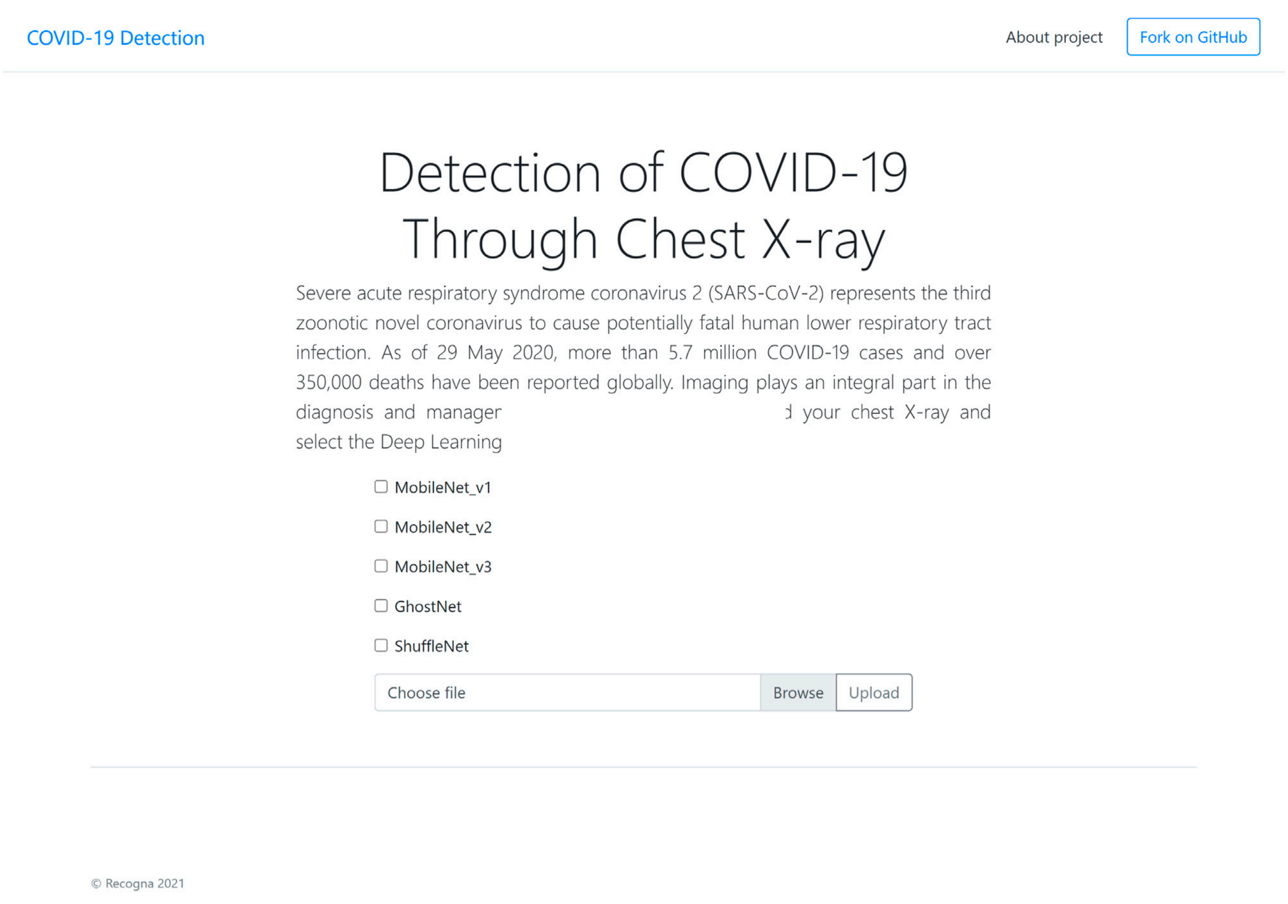
that larger training sets would be helpful in building more consistent models that could improve their accuracy.

All architectures mentioned in Section 3.2 were first trained on ImageNet before being fine-tuned on the COVID-19 Radiography Dataset for 100 epochs[4]. Notice that this last step was only performed on the fully connected layers. Mini-batches of size 120 were used, with cross-entropy as the loss function, Adam optimizer Kingma and Ba [31], and a learning rate of 3×10^{-5}.

3.4 Quantitative analysis

The lightweight architectures evaluated in this manuscript will be assessed using two standard measures: accuracy and F1-Score. We opted to use F1-Score since the dataset is highly unbalanced, with the vast majority of samples belonging to healthy individuals.

4 We used the official implementation provided by PyTorch Paszke et al. [33].


COVID-19 Detection

Detection of COVID-19 Through Chest X-ray

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) represents the third zoonotic novel coronavirus to cause potentially fatal human lower respiratory tract infection. As of 29 May 2020, more than 5.7 million COVID-19 cases and over 350,000 deaths have been reported globally. Imaging plays an integral part in the diagnosis and manager ɹ your chest X-ray and select the Deep Learning

FIGURE 10
Primary screen of the web tool to identify COVID-19 from chest X-ray images.

4 Experiments

This section presents the experiments comparing the lightweight models for COVID-19 identification and, later, screenshots of the developed web tool.

4.1 Comparison among light architectures

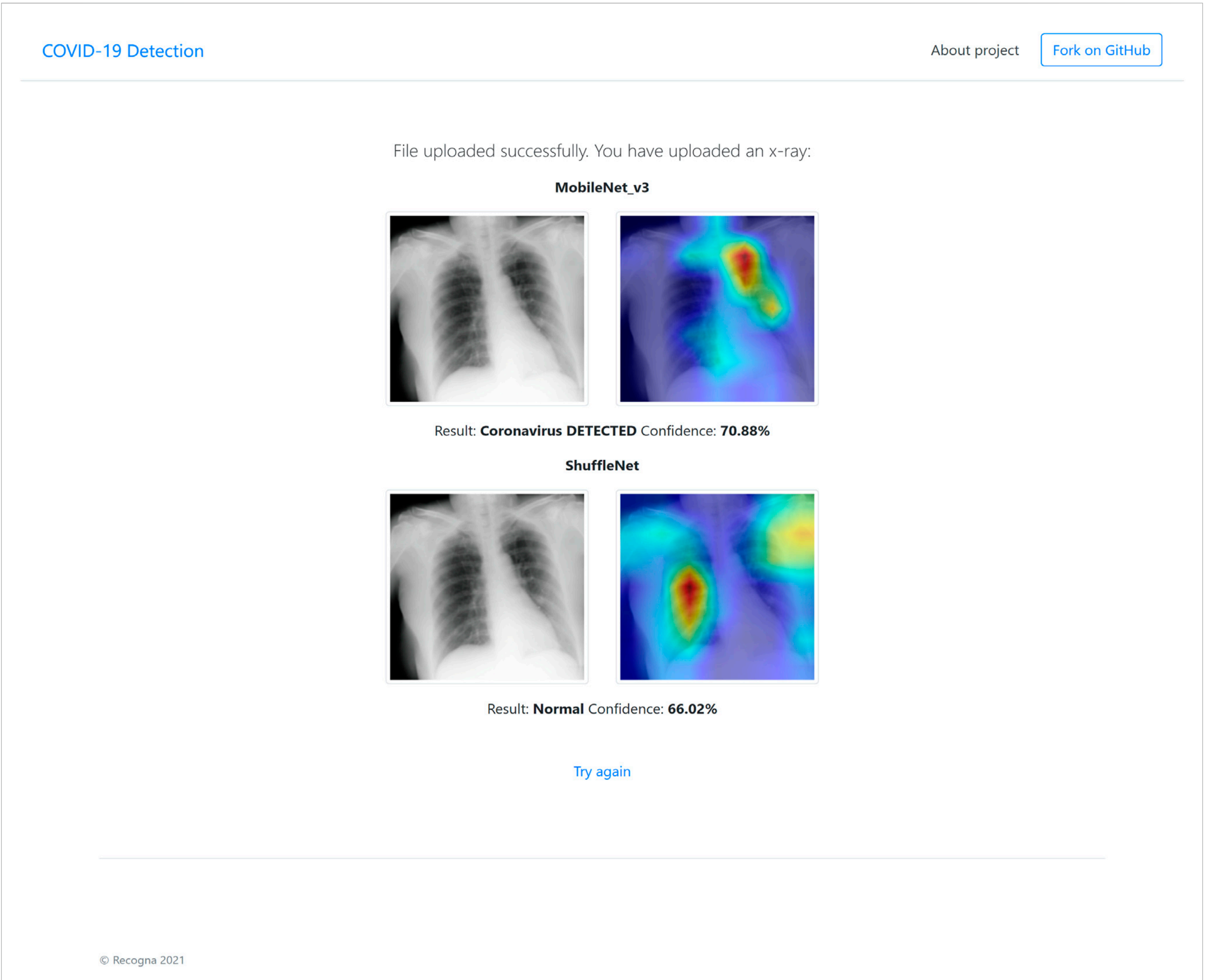
We begin by examining the behavior of the training loss, as shown in Figure 8. We observe that MobileNetV2 and MobileNetV3 had the fastest learning, closely followed by GhostNet. This suggests that the improvements made to MobileNetV1 have had a significant impact. ShuffleNet, while being the second most efficient for training, was the slowest to learn. It uses group convolutions to capture more complex relationships between filters and features, which do not appear to be effective in this particular domain. We believe that COVID-19 images have specific regions containing important information and do not vary significantly from other datasets with natural images.

All models started with a relatively high loss in the first epochs, but 20 iterations were sufficient to significantly reduce the error for MobileNetV2, MobileNetV3, and GhostNet. As previously noted, ShuffleNet did not produce satisfactory results. However, we believe that other data augmentation methods, such as adding noise, could increase dataset variability and benefit ShuffleNet. This could be a topic for future work, as our primary objective here was to evaluate the backbones themselves and not different data augmentation methods.

The following crucial information regarding the efficiency of the lightweight models concerns the training step. Table 1 presents the computational load required to train each model, reflecting the neural backbone's complexity and the number of parameters. MobileNetV3 was the fastest model, followed by ShuffleNet. We can observe three groups of models regarding their efficiency: 1) the top group comprising MobileNetV3 and ShuffleNet, 2) the second group composed of GhostNet and MobileNetV1, and 3) the less efficient group consisting of MobileNetV2. MobileNetV2 includes linear bottleneck layers and inverted residuals, which add an extra computational cost to its counterparts. More layers, even if linear, represent more parameters to learn. However, MobileNetV2 is one of the best for training, as shown in Figure 8.

The third column in Table 1 shows the efficiency gain of all techniques compared to the fastest one, i.e., MobileNetV3, which was almost three times faster than its previous version, i.e., MobileNetV2. Moreover, it was slightly more efficient than ShuffleNet (1.2×) but had better learning capacity (Figure 8). Considering both the training loss and computational time, MobileNetV3 emerged as the best option.

One last piece of information we shall consider is the effectiveness of the testing set. As mentioned in Section 3.4, we


COVID-19 Detection

File uploaded successfully. You have uploaded an x-ray:

MobileNet_v3

Result: **Coronavirus DETECTED** Confidence: **70.88%**

ShuffleNet

Result: **Normal** Confidence: **66.02%**

FIGURE 11
Outcome screen with the outputs given by each neural model selected in the previous step. A heatmap is also provided.

employed the standard accuracy and F1-Score to evaluate all backbones quantitatively. Both measures are used in two distinct ways: 1) mean accuracies over the testing using 100 epochs (the same number of epochs used for training), and 2) the accuracy with the best model selected in the previous step. Keeping track of the accuracies over the test set gives us an idea of whether the model was overfitting during training. We expect better recognition rates as the number of epochs increases, though not being a rule.

Figures 9A, B illustrate the accuracy and F1-Score values over the testing set, respectively. MobileNetV3 exhibited the expected behavior, with accuracy and F1-Scores increasing as the number of epochs increases. The same occurred for GhostNet and MobileNetV2, although to a lesser extent. ShuffleNet displayed the same behavior observed earlier, with the worst results. Its accuracy did not change over time (Figure 9A).

Table 2 presents the mean accuracy and F1-Score values considering all models over the testing set. Following the same behavior, the most accurate model was MobileNetV3, followed by its previous version MobileNetV2. The MobileNetV3 obtained the best results in all experiments, i.e., learning speed, training time, and effectiveness over the test set. The outcomes also highlighted the GhostNet as a reasonable architecture with a good trade-off between recognition rate and computational training complexity.

4.2 Web tool

This section briefly presents a web tool that allows users to upload a human chest X-ray image and receive its probability of belonging to one of the following classes: 1) viral pneumonia, 2) COVID-19, and 3) healthy[5]. Figure 10 illustrates the main home page, where users can choose from up to five pre-trained lightweight neural models (the same ones considered in this paper).

Figure 11 depicts the results of the web tool, showing the probabilities computed by each neural model and a heatmap highlighting the key regions used by each model to make its decision.

5 The tool is available at http://xraycovid.space.

The web tool has been deployed using Flask as the web framework, Apache as the server, PyTorch for the machine learning models, and Grad-CAM Selvaraju et al. [32] to generate the heatmaps.

5 Conclusion and future works

This manuscript compared five lightweight neural architectures to cope with computer-assisted COVID-19 recognition: MobileNetV1, MobileNetV2, MobileNetV3, GhostNet, and ShuffleNet. Experiments were conducted in three rounds: 1) training learning convergence, 2) training running time, and 3) efficiency over the testing set. Among all models, MobileNetV3 was the best concerning all rounds mentioned above, i.e., it has been the most effective for both training and testing steps and the fastest for training purposes.

Another contribution was a web tool that allows anyone to input chest X-ray images, choose one or more neural backbones, and get their probabilistic outputs together with a heat map. We expect the tool will serve anyone interested in having his images analyzed. We are aware that other image modalities, e.g., CT and lung ultrasound, are acknowledged to be more accurate than X-rays to detect COVID-19. However, X-ray images are more abundant and cheaper, with more datasets available. Therefore, in future works, we intend to use more datasets like CT and lung ultrasound and include more lightweight neural backbones.

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

MH, SA, AA, and JP contributed to conception and design of the study. MH organized the database. MH, SA, AA, and JP performed the statistical analysis. MH and JP wrote the first draft of the manuscript. MH, SA, AA, and JP wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This work was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (5-21-01-001-0031).

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

References

1. Hassan MM, AlQahtani SA, Alelaiwi A, Papa JP. Explaining COVID-19 diagnosis with Taylor decompositions. *Neural Comput Appl* (2022) 1–14. doi:10.1007/s00521-022-08021-7

2. Montavon G, Lapuschkin S, Binder A, Samek W, Müller K-R. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition* (2017) 65: 211–22. doi:10.1016/j.patcog.2016.11.008

3. Samek W, Binder A, Lapuschkin S, Müller K-R. Understanding and comparing deep neural networks for age and gender classification. In: 2017 IEEE International Conference on Computer Vision Workshops; 22-29 October 2017; Venice, Italy (2017). p. 1629–38.

4. Santos CFG, Passos LA, de Santana MC, Papa JP. Normalizing images is good to improve computer-assisted COVID-19 diagnosis. In: U Kose, D Gupta, VHC de Albuquerque, A Khanna, editors. *Data science for COVID-19*. Academic Press (2021). p. 51–62.

5. Tan M, Le QV. Efficientnet: Rethinking model scaling for convolutional neural networks. In: K Chaudhuri R Salakhutdinov, editors. Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA. PMLR (2019). p. 6105–14. 97 of Proceedings of Machine Learning Research.

6. Hu Q, Gois FNB, Costa R, Zhang L, Yin L, Magaia N, et al. Explainable artificial intelligence-based edge fuzzy images for Covid-19 detection and identification. *Appl Soft Comput* (2022) 123:108966. doi:10.1016/j.asoc.2022.108966

7. Ohata EF, Bezerra GM, das Chagas JVS, Neto AVL, Albuquerque AB, De Albuquerque VHC, et al. Automatic detection of Covid-19 infection using chest x-ray images through transfer learning. *IEEE/CAA J Automatica Sinica* (2020) 8: 239–48. doi:10.1109/jas.2020.1003393

8. de Mesquita VA, Cortez PC, Ribeiro AB, de Albuquerque VHC. A novel method for lung nodule detection in computed tomography scans based on boolean equations and vector of filters techniques. *Comput Electr Eng* (2022) 100:107911. doi:10.1016/j.compeleceng.2022.107911

9. Parah SA, Kaw JA, Bellavista P, Loan NA, Bhat GM, Muhammad K, et al. Efficient security and authentication for edge-based internet of medical things. *IEEE Internet Things J* (2020) 8:15652–62. doi:10.1109/jiot.2020.3038009

10. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* (2015) 521:436–44. doi:10.1038/nature14539

11. Gumaei A, Ismail WN, Hassan MR, Hassan MM, Mohamed E, Alelaiwi A, et al. A decision-level fusion method for Covid-19 patient health prediction. *Big Data Res* (2022) 27:100287. doi:10.1016/j.bdr.2021.100287

12. Hassan MR, Ismail WN, Chowdhury A, Hossain S, Huda S, Hassan MM. A framework of genetic algorithm-based cnn on multi-access edge computing for automated detection of Covid-19. *The J Supercomputing* (2022) 78:10250–74. doi:10.1007/s11227-021-04222-4

13. Alsahli MA, Alsanad A, Hassan MM, Gumaei A. Privacy preservation of user identity in contact tracing for Covid-19-like pandemics using edge computing. *IEEE Access* (2021) 9:125065–79. doi:10.1109/access.2021.3110762

14. Hassan MR, Hassan MM, Altaf M, Yeasar MS, Hossain MI, Fatema K, et al. B5g-enabled distributed artificial intelligence on edges for Covid-19 pandemic outbreak prediction. *Ieee Netw* (2021) 35:48–55. doi:10.1109/mnet.011.2000713

15. Polsinelli M, Cinque L, Placidi G. A light cnn for detecting Covid-19 from ct scans of the chest. *Pattern Recognition Lett* (2020) 140:95–100. doi:10.1016/j.patrec.2020.10.001

16. Iandola FN, Moskewicz MW, Ashraf K, Han S, Dally WJ, Keutzer K. *Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size* (2016). CoRR abs/1602.07360.

17. Zhang X, Zhou J, Sun W, Jha SK. A lightweight cnn based on transfer learning for Covid-19 diagnosis. *Comput Mater Continua* (2022) 72:1123–37. doi:10.32604/cmc.2022.024589

18. Paluru N, Dayal A, Jenssen HB, Sakinis T, Cenkeramaddi LR, Prakash J, et al. Anam-net: Anamorphic depth embedding-based lightweight cnn for segmentation of anomalies in Covid-19 chest ct images. *IEEE Trans Neural Networks Learn Syst* (2021) 32:932–46. doi:10.1109/tnnls.2021.3054746

19. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: N Navab, J Hornegger, WM Wells, AF Frangi, editors. *Medical image computing and computer-assisted intervention – miccai 2015*. Cham: Springer International Publishing (2015). p. 234–41.

20. Iyer TJ, Raj ANJ, Ghildiyal S, Nersisson R. Performance analysis of lightweight cnn models to segment infectious lung tissues of Covid-19 cases from tomographic images. *PeerJ Comp Sci* (2021) 7:e368–20. doi:10.7717/peerj-cs.368

21. Huang M-L, Liao Y-C. A lightweight cnn-based network on Covid-19 detection using x-ray and ct images. *Comput Biol Med* (2022) 146:105604. doi:10.1016/j.compbiomed.2022.105604

22. Zhao Q, Wang H, Wang G. Lcov-net: A lightweight neural network for Covid-19 pneumonia lesion segmentation from 3d ct images. In: 2021 IEEE 18th International Symposium on Biomedical Imaging; 13-16 April 2021; Nice, France (2021). p. 42–5.

23. Heidari A, Toumaj S, Navimipour NJ, Unal M. A privacy-aware method for Covid-19 detection in chest ct images using lightweight deep conventional neural network and blockchain. *Comput Biol Med* (2022) 145:105461. doi:10.1016/j.compbiomed.2022.105461

24. Kamal M, Aljohani AJ, Alanazi E, Albogamy FR. 5G and blockchain enabled lightweight solutions for containing COVID-19. *Security Commun Networks* (2022) 2022:1–12. doi:10.1155/2022/3370408

25. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. *Mobilenets: Efficient convolutional neural networks for mobile vision applications* (2017). doi:10.48550/ARXIV.1704.04861

26. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L. Mobilenetv2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 18-23 June 2018; Salt Lake City, UT, USA. Los Alamitos, CA, USA: IEEE Computer Society (2018). p. 4510–20.

27. Howard A, Sandler M, Chen B, Wang W, Chen L, Tan M, et al. Searching for mobilenetv3. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 27 October 2019 - 02 November 2019; Seoul, Korea (South). Los Alamitos, CA, USA: IEEE Computer Society (2019). p. 1314–24.

28. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C. Ghostnet: More features from cheap operations. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 13-19 June 2020; Seattle, WA, USA. Los Alamitos, CA, USA: IEEE Computer Society (2020). p. 1577–86.

29. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 27-30 June 2016; Las Vegas, NV, USA (2016). p. 770–8.

30. Zhang X, Zhou X, Lin M, Sun J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 18-23 June 2018; Salt Lake City, UT, USA (2018). p. 6848–56.

31. Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Y Bengio Y LeCun, editors. *3rd international conference on learning representations*. ICLR (2015).

32. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV); 22-29 October 2017; Venice, Italy (2017). p. 618–26. doi:10.1109/ICCV.2017.74

33. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. In: H Wallach, H Larochelle, A Beygelzimer, F d' Alché-Buc, E Fox, R Garnett, editors. *Advances in neural information processing systems 32*. Curran Associates, Inc. (2019). p. 8024–35.