frontiers | Frontiers in Computer Science

TYPE Review
PUBLISHED 09 April 2025
DOI 10.3389/fcomp.2025.1437664

OPEN ACCESS

EDITED BY
Sokratis Makrogiannis,
Delaware State University, United States

REVIEWED BY
Aili Wang,
Harbin University of Science and Technology, China
Manoj Kumar,
University of Wollongong in Dubai, United Arab Emirates

*CORRESPONDENCE
Teerayut Horanont
✉ teerayut@siit.tu.ac.th

RECEIVED 24 May 2024
ACCEPTED 03 March 2025
PUBLISHED 09 April 2025

CITATION
Lamichhane BR, Srijuntongsiri G and Horanont T (2025) CNN based 2D object detection techniques: a review. *Front. Comput. Sci.* 7:1437664. doi: 10.3389/fcomp.2025.1437664

CNN based 2D object detection techniques: a review

Badri Raj Lamichhane, Gun Srijuntongsiri and Teerayut Horanont*

School of Information, Computer and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Bangkok, Thailand

Significant advancements in object detection have transformed our understanding of everyday applications. These developments have been successfully deployed in real-world scenarios, such as vision surveillance systems and autonomous vehicles. Object recognition technologies have evolved from traditional methods to sophisticated, modern approaches. Contemporary object detection systems, leveraging high accuracy and promising results, can identify objects of interest in images and videos. The ability of Convolutional Neural Networks (CNNs) to emulate human-like vision has garnered considerable attention. This study provides a comprehensive review and evaluation of CNN-based object detection techniques, emphasizing the advancements in deep learning that have significantly improved model performance. It analyzes 1-stage, 2-stage, and hybrid approaches for object recognition, localization, classification, and identification, focusing on CNN architecture, backbone design, and loss functions. The findings reveal that while 2-stage and hybrid methods achieve superior accuracy and detection precision, 1-stage methods offer faster processing and lower computational complexity, making them advantageous in specific real-time applications.

KEYWORDS

CNN, hybrid, object detection, 1-stage, 2-stage, review

1 Introduction to object detection

Access to information occurs through a diverse array of channels, encompassing both traditional and digital sources. Traditional sources include newspapers, magazines, television, radio, books, libraries, and billboards, while digital sources comprise websites, blogs, social media platforms, mobile applications, streaming services, and search engines. When individuals encounter visual stimuli such as advertisements or traffic signs, the ability to accurately identify the objects depicted and extract pertinent information is crucial. Effective information extraction guides individuals along appropriate pathways and mitigates the risks associated with confusion or misinformation that may lead to erroneous conclusions. Consequently, meticulous and precise extraction of information from images is of paramount importance in ensuring informed decision-making (Ardia et al., 2020).

Image detection represents an advanced computational technology that processes visual data to identify and locate specific objects within images. This methodology differs from image classification, which categorizes entire images without delineating object locations. Image detection focuses on recognizing the presence and spatial positioning of objects, often utilizing bounding boxes to indicate their locations within a given frame. The significance of image detection spans multiple domains due to its ability to automate and enhance processes that previously relied on human visual assessment. For instance,

image detection improves operational efficiency in manufacturing quality control by enabling the rapid and accurate inspection of numerous components. In autonomous driving applications, it is essential for detecting pedestrians, traffic signs, and other vehicles, thereby ensuring safety on roadways (Dong et al., 2018). In healthcare settings, image detection is critical for identifying tumors or abnormalities in medical imaging, leading to improved diagnostic accuracy and timely interventions. Furthermore, surveillance systems leverage image detection technologies to monitor environments for unauthorized access or suspicious activities, thereby enhancing security measures. Therefore, image detection provides vital insights that enable systems to respond appropriately to their visual contexts, thereby highlighting its crucial role in contemporary technological applications and societal functions (Hammoudeh et al., 2022). This technology uses advanced machine learning and deep learning algorithms to improve safety across various domains by accurately detecting objects and their environments (Guan, 2017).

1.1 Object identification in image

For a comprehensive understanding of visual data, the classification of images and object detection methodologies constitute critical paradigms in computer vision. The precise identification and spatial localization of objects within digital images or video streams enable a nuanced understanding of content across diverse computational applications. This fundamental interpretive framework encompasses multifaceted computational processes, including but not limited to object trajectory analysis, pose estimation, instance-level object segmentation, and sophisticated inventory management strategies (Dong et al., 2018; Hammoudeh et al., 2022; Dundar et al., 2016). Traditional object detection methodologies are characterized by their ability to operate without necessary historical training data, rendering them predominantly unsupervised. Seminal approaches such as the Viola-Jones algorithm (Viola and Jones, 2001; Li et al., 2012), the Scale-Invariant Feature Transform (SIFT) (Lowe, 1999), and histogram-based techniques (Freeman and Roth, 1995; Dalal and Triggs, 2005) exemplify this methodological category. Contemporary research, however, demonstrates the exponential efficacy of supervised learning paradigms leveraging sophisticated deep learning architectures, which have become predominant in real-world computational scenarios. Within this context, machine learning and advanced artificial intelligence techniques are strategically deployed to comprehend and interpret visual information (Hammoudeh et al., 2022). These sophisticated computational tools enable precise object localization and identification, finding critical applications in multidimensional domains such as intelligent traffic monitoring systems, comprehensive surveillance and navigation frameworks, and advanced biometric recognition technologies in smartphones and autonomous vehicular systems (Guan, 2017; Tang et al., 2023; He et al., 2015a). Figure 1 provides a schematic representation of the intricate object detection and classification methodological landscape.

The paradigmatic evolution of object detection methodologies is significantly driven by the sophisticated integration of advanced deep learning architectures with supervised learning algorithmic frameworks. This intricate symbiosis represents a pivotal mechanism for optimizing object detection methodologies, substantially augmenting the computational capacity to precisely identify and spatially localize objects within digital imagery and video streams. Contemporary deep learning neural networks, exemplified by Convolutional Neural Networks (CNN) (Sun, 2024), Region-based Convolutional Neural Networks (R-CNN) (Girshick et al., 2014a), You Only Look Once (YOLO) (Redmon et al., 2016), and Residual Networks (ResNet) (He et al., 2016b,a), have made transformative contributions to the field of computer vision. These advanced computational models demonstrate exceptional performance by strategically integrating multi-scale feature representations (Zhang et al., 2019) and iteratively refining candidate object bounding box delineations (Yao et al., 2022) during object identification processes. Neural network algorithmic frameworks, which systematically build upon established architectural paradigms (Dundar et al., 2016) and advanced learning systems, have initiated a global transformation in computational object detection capabilities (Hammoudeh et al., 2022; Guo et al., 2019). Despite these remarkable advancements, significant computational challenges remain in fully recognizing objects across heterogeneous imaging contexts, which include varying illumination conditions, diminutive object dimensions, partial occlusions, diverse viewing angles, complex poses, and varying spatial configurations. Consequently, the scholarly discourse has increasingly focused on object localization methodologies, with researchers actively seeking innovative solutions to address these inherent computational complexities (Guan, 2017). The primary scholarly objective is to achieve unprecedented accuracy in object identification through cutting-edge computational tools. To enable a comprehensive understanding, the conventional object detection framework can be systematically outlined in three fundamental computational stages: the strategic identification of salient informative regions, sophisticated feature extraction mechanisms, and subsequent probabilistic classification processes. This modular computational architecture significantly enhances object recognition capabilities by employing a rigorous, multi-stage approach to identifying and taxonomically classifying objects within digital imagery (Girshick et al., 2014b).

The computational process of object detection involves a sophisticated, multi-staged methodological framework, with each stage playing a crucial role in the precise identification and spatial localization of objects in digital imagery. Informative region selection represents the first computational phase, strategically focusing on identifying spatially salient regions with a high probabilistic likelihood of containing target objects. This critical stage is enabled through advanced computational methodologies such as selective search algorithms and Region Proposal Networks (RPNs), which generate candidate regions through a comprehensive analysis of image chromatic intensities, textural characteristics, and spatial configurations (Girshick et al., 2014a,b). Bounding box representations are systematically employed to outline these potential object zones, providing a precise cartographic representation of the object's spatial positioning within the digital image. The subsequent phase, feature extraction, involves the sophisticated retrieval and transformation of relevant computational data from the selected bounding box

FIGURE 1
Object detection and classification.

regions. Convolutional Neural Networks (CNNs) function as computational transformative mechanisms, converting raw image features into standardized representational matrices that enable the extraction of intricate spatial and hierarchical characteristics essential for effective object detection. Established computational techniques, including Scale-Invariant Feature Transform (SIFT) (Lowe, 1999), HOG (Dalal and Triggs, 2005), and Haar-like feature extraction methodologies (Cristianini and Shawe-Taylor, 2000), are strategically deployed during this computational phase to enhance feature representation and discriminative capabilities. The conclusive stage, Classification, involves the probabilistic assignment of taxonomical class labels to candidate object regions predicated upon the extracted computational attributes. This process entails identifying specific object categories such as anthropomorphic entities, vehicular structures, arboreal organisms, and urban infrastructural elements. The classification mechanism is realized through sophisticated classifiers embedded within fully connected neural network architectures, which leverage the extracted computational characteristics as input to determine the most probable taxonomical designation for each detected object. Canonical classification algorithms, including Support Vector Machines (SVM) (Cristianini and Shawe-Taylor, 2000; Awad et al., 2015), AdaBoost ensemble learning frameworks (Freund et al., 1999), and Deformable Part-based Model (DPM) networks (Viola and Jones, 2001), represent pivotal computational paradigms employed in this sophisticated classificatory process. These meticulously orchestrated computational stages collectively constitute a comprehensive and robust methodological framework for executing sophisticated object detection across diverse computational and visual analysis applications (Chhabra et al., 2024).

This observation highlights the critical need for improvement to increase the effectiveness of real-time object detection systems. Solutions tailored for hardware compatibility rely on discriminant feature descriptors that involve minimal computational overhead and shallow, easily trainable architectures. This is achieved by adopting a pragmatic methodology grounded in reality. However, these techniques may become less reliable when recognizing and predicting essential items is crucial. Finding the right balance between accuracy and efficiency remains essential for successfully using these methods. Deep learning approaches have seen significant advancements through a result-driven focus. In this study, we focus on object recognition methods using CNNs, which are renowned for their ability to replicate human visual intelligence. We examine one-stage, two-stage, and hybrid approaches to image recognition, localization, classification, and identification to gain a better understanding of the methodologies used by CNN-based object detection systems. We illustrate the benefits of two-stage and hybrid methods regarding accuracy and detection precision while acknowledging the effectiveness of one-stage methods concerning processing speed and computational simplicity. This analysis considers the architecture, backbone structure, and performance metrics of these approaches, emphasizing the need to strike a balance between accuracy, efficiency, and resource usage. Our review aims to facilitate informed decisions when designing and implementing CNN-based object identification systems (Zhao et al., 2024; Aggarwal and Kumar, 2021).

This study focuses on a brief discussion of object detection techniques based on CNN. It begins with key milestones illustrating the developmental process, then dives into several deep-learning object detection techniques utilizing a variety of benchmark datasets. The study explores the hierarchical growth of CNN-based detection strategies, focusing on the architecture of deep-learning CNN models for both one-stage and two-stage object detection. Additionally, it compares approaches based on computational cost, time efficiency, accuracy, algorithmic adaptability, and significance within and across detection stages for both CNN-based generic and salient object detection architectures. The “Challenges and Future Opportunities” section discusses potential ways to overcome the existing obstacles in object detection, while the ’Conclusion and Future Works’ section summarizes the study’s conclusions and provides guidance for future research directions aimed at advancing CNN-based object detection methodologies.

2 Key milestones in object detection development

Detecting objects in images is a crucial step in the transition from hand-crafted templates to advanced deep learning models. Initially, there was a template-matching approach where image patches were compared to predefined templates. Subsequently, the

idea of designing manual features emerged, such as edges, colors, and textures, which were developed for object identification. After this period, the use of statistical methods for object recognition gained popularity, which proved valuable in certain applications as well (Zou et al., 2023). HOG (Dalal and Triggs, 2005), and SIFT (Lowe, 1999) were prominent during this time. HOG divides an image into blocks to calculate gradients, subsequently combining these blocks with adjacent ones to produce gradient orientation histograms that capture light and maintain invariance over broader areas. While this method is effective for low-cost geometric modifications and varying lighting, it is less effective in identifying small objects or multiple objects within the same image. On the other hand, SIFT examines an object's surrounding areas and spatial context, using edge detection or Laplacian filtering to identify unique key points. The construction of SIFT descriptors relies on histogram computation, with the Gaussian window defining core regions, while key-point matching is performed using Euclidean distance. SIFT provides robustness by carefully selecting key points that generate descriptors, but it is susceptible to issues such as occlusion and background clutter. Therefore, it must be used with caution.

The early manual methods that relied on designed elements such as colors and textures were constrained and rigid; therefore, improvements were required. Furthermore, obstacles to accurate object identification within images include overfitting, which arises from issues with training data for algorithms, such as large datasets and computational resources mentioned by Chen et al. (2017). Deep learning became more famous for overcoming these limitations after 2006 (Zou et al., 2023; Elgendy, 2020), as it fully harnesses the extensive learning capacity of a network structure. Thus, after 2010, there was a revolution in deep learning methods, marking a pivotal shift toward robust convolutional neural networks (CNNs). CNNs utilize deep learning-enabled features to learn complex patterns from massive data, delivering significant accuracy directly. The techniques employed, such as AlexNet (Krizhevsky et al., 2017), GoogleNet (Szegedy et al., 2016), and VGGNet (Simonyan and Zisserman, 2014), in the two-stage detectors, such as R-CNN (Girshick et al., 2014a), Fast R-CNN (Girshick, 2015), and Faster R-CNN (Ren et al., 2015), were enhanced to improve accuracy and performance, making them possible to work in real-time applications. The milestone chart shown in Figure 2 presents the development year alongside the corresponding architecture name. As deep learning approaches expand their applicability in the real world, they diversify techniques that address real challenges without being constrained by their previous limitations. These diverse fields include autonomous vehicles, robotics, medical imaging, security, and more. Advanced deep learning methods revolutionize object detection techniques, ushering in a new era of possibilities. However, some challenges remain to unlock its full potential (Elgendy, 2020).

3 Understanding deep neural networks

Deep Neural Networks (DNNs) are computational models inspired by the human brain, characterized by multiple interconnected layers that excel at capturing complex data patterns. These hidden layers enable hierarchical learning of intricate relationships within data, making DNNs powerful tools for addressing diverse and challenging tasks. Neurons in these layers dynamically adjust weights and biases during training to improve feature abstraction. Discrepancies between actual and predicted values are minimized through gradient descent optimization, ensuring improved performance (Krizhevsky et al., 2012).

Various types of DNNs have been developed for specific applications: Feedforward Neural Networks (FNN) are commonly utilized for identification and recognition tasks (Ben Braiek and Khomh, 2023); Convolutional Neural Networks (CNN) excel at image processing tasks (Sun, 2024); Recurrent Neural Networks (RNNs) are primarily used for time series data (Girshick et al., 2014a); Long Short-Term Memory (LSTM) networks address issues related to vanishing gradients in longer sequences (Hochreiter and Schmidhuber, 1997); and Transformer Networks have gained prominence in natural language processing tasks.

CNNs specifically target image recognition and processing. In this context, models are trained using labeled datasets, enabling them to effectively extract relevant information from test images. The feature map technique highlights detected features and serves as input for subsequent layers that progressively build a hierarchical image representation. The fully connected layer then integrates features from earlier layers, mapping them to specific classes and playing a crucial role in image classification (Elgendy, 2020; Zou et al., 2023).

Region-based CNNs (R-CNN) (Girshick et al., 2014a) combine the strengths of CNNs with region-based approaches, significantly enhancing object detection accuracy. Initially, R-CNN segments an image into multiple proposals that may contain objects. A CNN then processes these regions for feature extraction and classification, further refining the final classification through this process. Key variants include Fast R-CNN (Girshick, 2015), which improves processing speed by sharing convolutional features across regions, and Faster R-CNN (Ren et al., 2018), which directly generates proposals to enhance both speed and accuracy. Additionally, Mask R-CNN (He et al., 2017) extends this framework to predict object masks for segmentation tasks. LSTM networks (Hochreiter and Schmidhuber, 1997) enhance learning dependencies within sequential data, thereby improving performance on time series tasks through a network-controlling mechanism that regulates information flow (Amjoud and Amrouch, 2023).

In the realm of object detection, both generic object detection (Girshick et al., 2014b) and salient object detection (Liu et al., 2015; Vig et al., 2014) methodologies aim to identify and understand objects within images. Generic object detection relies on deep learning models to meticulously detect objects of varying sizes when trained on labeled datasets; this includes applications such as pedestrian and traffic sign recognition in autonomous vehicles. Conversely, salient object detection mimics human attention by prioritizing visually distinct objects based on contrast and spatial arrangement attributes. This approach enhances tasks such as robust vision, image compression, and segmentation by emphasizing captivating elements within an image. Implementing these complex algorithms requires extensive training on large

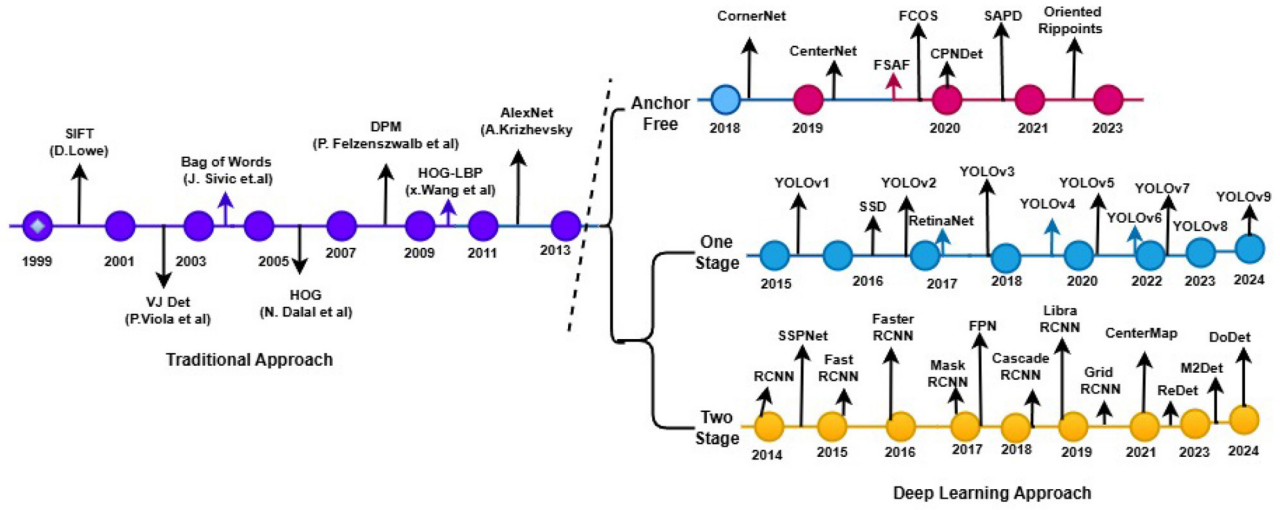

FIGURE 2
Time-line of 2D focused object detection techniques.

datasets to continually improve object detection capabilities. Consequently, deep learning methodologies have revolutionized image editing techniques and significantly impacted various fields reliant on image classification and object detection (Zhao et al., 2019).

4 Exploring the architecture and functionality of CNN

CNNs represent a powerful deep learning architecture commonly used in various domains, including artificial intelligence, natural language processing, computer vision, and autonomous vehicles. As a dependable foundation for image recognition and analysis, CNNs utilize a structured layer arrangement that collaboratively processes and extracts meaningful information from input data, particularly images. As illustrated in Figure 3, the architecture encompasses several key components: the input layer, pooling layers, convolutional layers, and fully connected layers, each playing a role in extracting abstract features from the image while facilitating sophisticated data interpretation and analysis. This intricate interplay of layers enhances the model's capability to identify complex patterns, positioning CNNs at the forefront of technological advancements in visual recognition tasks.

4.1 Input layer

The input layer is the gateway that receives raw data, establishing the foundation for subsequent network processing. The input data can encompass various categories based on availability and specific requirements. This may include image data in a 3D map format with pixel values indicating width and height, time series data such as stock market values in a 2D format corresponding to time steps, or textual data fed into the network for desired outputs. Pre-processing steps, including normalization, are crucial for preparing the data to ensure it aligns with the network's processing capabilities. The primary role of the input layer is to enable meaningful insights derived from the transfer of data during the input stage.

4.2 Convolutional layers

The convolutional layer is a fundamental component where most computations occur. It employs small grids, filters, or kernels to detect specific patterns such as lines, curves, or shapes within the receptive field. This multi-layered architecture of convolutional layers progressively interprets the visual information embedded in raw image data. To detect complex patterns and objects, each successive layer extracts feature maps that inform the deeper layers of the network.

4.3 Pooling layers

Pooling layers generate summary statistics for adjacent layers by downsampling data while retaining essential information. This process enhances object detection capabilities by providing invariance to rotations and translations. Additionally, pooling reduces memory consumption, manages computational costs and weights, and mitigates overfitting. The most common pooling methods include max pooling and average pooling. In max pooling, the highest value within a specified region of the input feature is selected as the output for that region, thereby emphasizing prominent features. Conversely, average pooling calculates the average value from a specific region of the input feature map to produce a smoother representation of features within that region. This approach aids in locating objects in images while considering their overall appearance.

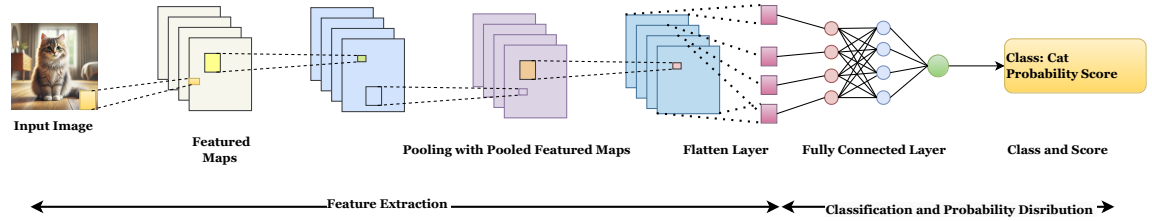

FIGURE 3
Convolution neural network basic architecture (Buckner, 2019).

4.4 Activation layers

Activation layers are critical components that enable CNNs to learn non-linear transformations of complex patterns for object detection tasks. These layers capture subtle relationships between features, leading to advanced models with enhanced generalization capabilities. As illustrated in Table 1, popular activation functions include ReLU (Fukushima, 1975), Tanh (Hereman and Malfliet, 2005), Leaky ReLU (Bai, 2022), ELU (Clevert et al., 2015), Sigmoid (Ramachandran et al., 2017), and SELU (Zhu et al., 2023), each with unique characteristics (Nwankpa et al., 2018). By leveraging diverse activation functions, CNNs can effectively address more challenging object detection tasks while retaining resilience and adaptability. These activation layers are essential for enhancing the network's ability to recognize intricate patterns and generate accurate predictions, ultimately improving performance in object detection by modeling and interpreting complex relationships within the data.

4.5 Fully connected layers

Fully Connected Layers (FCL) are integral components of CNNs, designed to connect neurons across different layers. Comprising neurons, weights, and biases, these dense layers serve as essential mechanisms that transform extracted information into a format that can be meaningfully interpreted. The FCL facilitates complex information sharing by integrating features according to the specific nature of the task, whether classification or regression. Ultimately, a single neuron representing the expected output emerges as the final result of a fully connected layer. The structure and functionality of FCLs are illustrated in Figure 3. Fully connected layers enhance the network's ability to comprehend intricate patterns by acting as a bridge between feature extraction and decision-making. The close interconnectivity of neurons within these layers enables CNNs to excel in object detection tasks, effectively synthesizing information from various features to inform predictions (Alzubaidi et al., 2021).

4.6 Architectural backbone network

The backbone architecture of a neural network is its fundamental structure, forming the basis for models, particularly in deep learning applications utilized for tasks such as image processing. The core of CNNs is composed of layers designed for hierarchical feature extraction. These layers may include pooling, normalization, and convolutions. The backbone architecture captures precise representations of incoming data as it moves through the system, enhancing the network's capability to understand and handle complex information. CNNs employ several well-known backbone networks, including AlexNet (Krizhevsky et al., 2017), VGGNet (Simonyan and Zisserman, 2014), ResNet (Residual Networks) (Choi et al., 2018), InceptionNet(GoogLeNet) (Szegedy et al., 2016), MobileNet (Sandler et al., 2018), and DenseNet (Huang et al., 2017).

4.7 VGGNet architecture

Simonyan and Zisserman (2014) proposed the VGG architecture by significantly enhancing traditional CNN models. This refined design achieved an impressive top-5 accuracy of 92.7% on the widely recognized ImageNet benchmark dataset, demonstrating its effectiveness in large-scale image classification tasks. A general diagram is shown in Figure 4. A key innovation of the VGG architecture is the consistent use of 3x3 convolutional filters throughout the network, reducing the overall parameter count and ensuring architectural simplicity and coherence while maintaining the ability to capture intricate features. The authors presented two variants of this architecture, namely VGG16 and VGG19, which comprise 16 and 19 layers of deep neural networks, respectively (Nash et al., 2018).

4.8 InceptionNet (GoogLeNet)

A groundbreaking deep learning architecture, InceptionNet, also known as GoogLeNet, was introduced by Szegedy et al. (2016). This architecture addressed a critical bottleneck in traditional models by allowing images of varying resolutions to be fed directly into the network without extensive preprocessing. Designed with computational efficiency in mind, InceptionNet achieves superior performance in image classification tasks while optimizing resource utilization. A defining feature of this architecture is the introduction of inception modules, which integrate multi-scale convolutions within a single layer and concatenate their outputs. This approach facilitates the effective capture of local and global features, enhancing the network's ability to learn complex

TABLE 1 Comparative analysis of activation functions in deep learning architectures.

Activation function	Mathematical representation	Benefits	Limitations	Optimal use cases
ReLU (Fukushima, 1975)	$f(x) = \max(0, x)$	Computationally efficient, Alleviates vanishing gradient issues	Susceptible to Dying ReLU phenomenon	Widely adopted in various architectures
Sigmoid (Ramachandran et al., 2017)	$f(x) = \frac{1}{1+e^{-x}}$	Produces binary-outputs (0 and 1)	Vulnerable to vanishing gradients, Saturation effects	Primarily for binary classification tasks
Tanh (Hereman and Malfliet, 2005)	$f(x) = \tanh(x)$	Outputs centered between -1 and 1, Superior gradient behavior compared to Sigmoid	Saturation effects at extremes	Viable alternative to Sigmoid function
Leaky ReLU (Bai, 2022)	$f(x) = \max(\lambda x, x)$ (λ is a small positive constant)	Mitigates Dying ReLU issue, Reduces vanishing gradient risk	More intricate than standard ReLU	Effective in preventing neuron inactivity
ELU (Clevert et al., 2015)	$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha(e^x - 1) & \text{otherwise} \end{cases}$ (α is the leakiness parameter)	Produces smoother outputs compared to ReLU, Prevents Dying ReLU issue	Increased complexity relative to ReLU	Suitable for deep neural networks requiring robustness
SELU (Zhu et al., 2023)	$f(x) = \lambda \cdot \alpha \cdot e^x - \alpha$ (λ and α are constants)	Self-normalizing properties, Scaled variant of ELU function	More complex implementation than ELU	High gain scenarios (requires careful tuning)
Swish (Ramachandran et al., 2017)	$f(x) = x \cdot \text{sigmoid}(\beta \cdot x)$ (β is a tunable hyperparameter)	Facilitates smooth gradient propagation, Non-monotonic behavior enhances expressiveness	More computationally intensive than ReLU alternatives	Balances performance and efficiency in various applications
Mish (Ramachandran et al., 2017)	$f(x) = x \cdot \tanh(\ln(1 + e^x))$	Provides smooth outputs akin to ReLU, Enhances training stability through improved gradients	Complex computational overhead compared to simpler functions	Optimal for scenarios requiring superior optimization performance

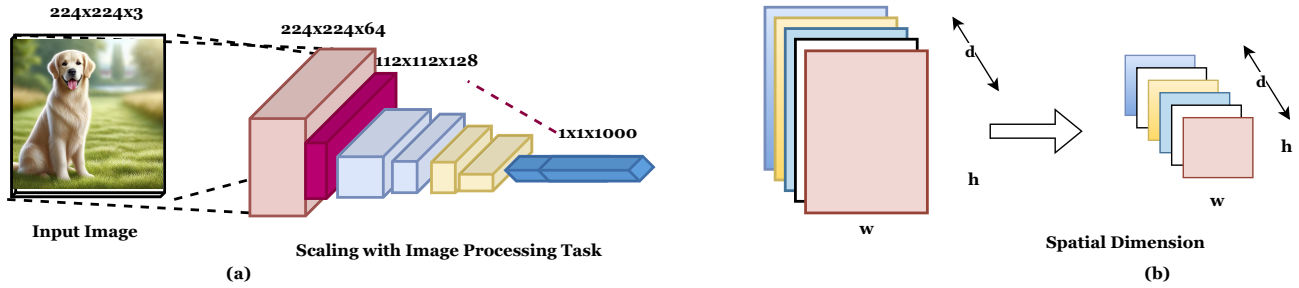

FIGURE 4
Key components of deep learning architectures: **(a)** the VGG architecture (Grimaldi et al., 2018) and **(b)** Global Average Pooling (Zhang et al., 2020). These architectures are fundamental to the design and implementation of deep neural networks in computer vision and image recognition, balancing readability with the technical details essential for understanding their significance.

representations. Furthermore, compared to conventional deep neural networks, InceptionNet significantly reduces the number of parameters while maintaining state-of-the-art accuracy, making it both innovative and efficient.

4.9 ResNet

The Residual Network (ResNet) architecture revolutionized deep learning by addressing the challenges associated with training complex neural networks. ResNet incorporates residual connections, also known as skip connections, which allow the direct flow of information and gradients between layers. This innovative approach effectively mitigates the vanishing gradient problem, a common issue in deep networks, and facilitates training exceptionally deep architectures comprising hundreds or even thousands of layers. ResNet models, such as ResNet-50, ResNet-101, and ResNet-152, are available in varying depths, with the numbers indicating the total layers in the network. These architectures have demonstrated state-of-the-art performance across various computer vision tasks, including image classification, object detection, and segmentation, establishing ResNet as a foundational model in deep learning research (Choi et al., 2018).

4.10 Output layer

The output layer is the final decision-making component in object detection with CNNs, providing results after thorough data

processing and analysis. It adapts to the specific requirements of regression or classification tasks. In classification scenarios, for instance, the output layer may consist of individual neurons corresponding to different classes, estimating the probability that an input belongs to each category. Conversely, in regression tasks, a single neuron may represent the predicted value or utilize an activation function to convey the learned prediction, with evaluation metrics such as mean squared error or absolute error assessing accuracy. As a critical element of the neural network architecture, the output layer embodies the system's capability to accurately detect and classify objects based on predefined criteria, ensuring optimal performance in object detection tasks and reflecting the complexity of the problems addressed.

5 Generic object detection techniques

Generic object detection techniques are recognized for their adaptability and versatility in the realm of CNN-based object detection. These methods can detect and classify a wide range of objects in images, including those that do not fit into predefined categories. They effectively identify and classify objects in complex visual environments by generating bounding boxes that outline object locations and provide confidence scores for their detection. A key feature of generic object detection is its ability to function without prior knowledge of specific object categories, relying instead on universal attributes such as color, shape, texture, and edge patterns to facilitate detection. The field includes various methodologies tailored to different requirements and scenarios, integrating advanced strategies to address varying challenges. These techniques achieve accurate and comprehensive object detection through innovative algorithms and robust feature extraction processes, even in intricate and unstructured visual data. As the field evolves, researchers continue to explore novel approaches to feature extraction and detection strategies, enhancing the efficiency, accuracy, and applicability of generic object detection techniques in real-world applications.

The One-Stage framework represents a regression-based approach designed to prioritize speed by predicting object attributes directly, eliminating the need for a separate region proposal step. This architecture aims to simultaneously predict object locations and bounding boxes in a single forward pass through the network, making it particularly suitable for real-time applications. However, the challenge of accurately detecting objects in a single pass often hinders its ability to achieve the same level of precision as more complex frameworks. Prominent implementations of the One-Stage framework include Single Shot Multibox Detectors (SSD) (Erhan et al., 2014; Van de Sande et al., 2011), You Only Look Once (YOLO) (Redmon et al., 2016), AttentionNet (Yoo et al., 2015), G-CNN (Najibi et al., 2016), and Differentiable Single Shot Detector (DSSD) (Fu et al., 2017). These models have garnered widespread popularity due to their ability to streamline object detection tasks and deliver rapid inference times, making them ideal for scenarios where computational efficiency is critical. Despite their speed advantages, substantial research is underway to enhance their accuracy while maintaining efficiency. As a result, the One-Stage framework remains an active and significant area of exploration in CNN-based object detection, offering both practical applications and opportunities for further innovation.

The Two-Stage framework employs a region-based approach, operating in two distinct stages, making it particularly effective for accurately handling objects with complex shapes and varying poses. In the first stage, the framework identifies potential object regions within the image. The second stage refines the bounding boxes and classifies the objects within these proposed regions. Compared to the One-Stage framework, the Two-Stage approach consistently achieves higher accuracy, although it comes at the cost of increased computational complexity and longer inference times. Prominent implementations of the Two-Stage framework include R-CNN (Girshick et al., 2014a), SPPnet (He et al., 2015a), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2015), R-FCN (Dai et al., 2016), and Mask R-CNN (He et al., 2017). The evolution of this framework has been marked by several key innovations. R-CNN introduced the concept of a region proposal phase, enabling the detection of potential object locations, while SPPnet incorporated spatial pyramid pooling to effectively handle objects of varying scales. Faster R-CNN enhanced this approach by introducing the Region Proposal Network (RPN), which generates region proposals more efficiently. Fast R-CNN improved upon R-CNN by sharing convolutional features across proposals, which significantly reduces computational overhead. R-FCN introduced position-sensitive score maps, allowing for more precise localization and classification, while Mask R-CNN extended Faster R-CNN by adding instance segmentation capabilities. Despite its superior accuracy, the Two-Stage framework's computational demands and extended processing time render it less suitable for real-time applications. However, it remains a pivotal area of research in object detection using CNNs. Ongoing advancements aim to balance the framework's high accuracy with the need for improved computational efficiency, ensuring its continued relevance in academic and applied settings (Shah and Tembhurne, 2023). The simple classification diagram is shown in Figure 5.

The goal of the **hybrid approach** is to balance speed, computational complexity, and accuracy by combining elements of both one-stage and two-stage frameworks. This strategy enables the creation of object detection systems that leverage the advantages of both methods, maximizing their respective strengths. NAS-FPN (Ghiasi et al., 2019) is a well-known hybrid approach that uses multi-scale representations to enhance object detection at various scales. Other implementations of the hybrid approach use the focal loss technique to address class imbalance issues commonly encountered in object detection applications, such as Mask R-CNN with attention and YOLO with FPN. Several R-CNN variants also adopt a hybrid approach to enhance performance, integrating components of both one-stage and two-stage frameworks. The capability of hybrid approaches to capitalize on the benefits of both frameworks while addressing their limitations has contributed to their increasing popularity in recent years.

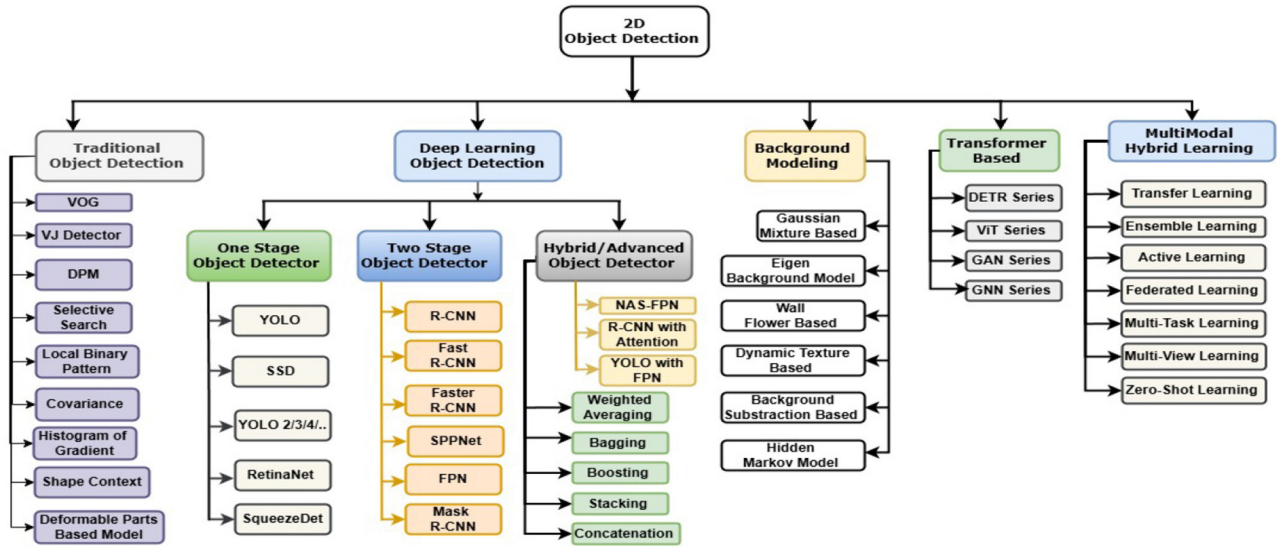

FIGURE 5
Classification of object detection techniques. The chart offers a comprehensive overview of various object detection techniques in 2D computer vision, detailing the different algorithms and models within each category.

5.1 Comprehensive review of one-stage (regression based) networks object detection model

The objects in the image can be recognized quickly and efficiently using single-stage object detection techniques. Single-stage detection methods, such as SSD (Elgendy, 2020) and the YOLO series, predict an object's approximate bounding box in a single neural network run, enabling quick and effective object recognition. Although this comes at the expense of lower accuracy rates, these methods demonstrate excellent reliability compared to two-stage detectors. Typically, greater accuracy is achieved in identifying larger items compared to smaller and closely spaced objects. Later iterations of YOLO (Redmon et al., 2016), such as YOLOv2 (Redmon and Farhadi, 2017), YOLOv3 (Redmon and Farhadi, 2018), and others, have successfully incorporated deeper neural networks to enhance performance. This modification has yielded more effective results while maintaining strong accuracy and low processing complexity. An in-depth examination of the YOLO and SSD architectures, along with their comparative outcomes across various datasets, will illuminate their advantages and disadvantages. These investigations contribute to a better understanding of the practical performance of these one-stage detection methods. Single-stage detection methods hold promise in applications requiring fast response times as they provide a trade-off between speed and accuracy. Further improvements in these methods will lead to even higher efficiency and accuracy.

5.2 You Only Look Once (YOLO)

Redmon et al. (2016) proposed a unique and improved single-stage detector called YOLO for object detection and image verification. YOLO enhances object recognition by combining high-level feature mapping with a reliable evaluation of various item categories, resulting in precise predictions represented in bounding boxes, as shown in Figure 6a. This innovative design, shown in Figure 6b, divides the input image into SxS cells using a grid-based method, with each cell providing bounding box features and essential confidence scores. The probability of an object's presence and the accuracy of the bounding box location are carefully combined to yield the confidence score. Pr (project)=1 indicates the target object's presence, while Pr(object)=0 indicates its absence. YOLO ensures substantial accuracy in object localization evaluations using the Intersection over Union (IoU) metric, a crucial measure of alignment between the actual and predicted boxes. The method of calculating confidence involves a rigorous multiplication of dimensions (x, y, w, h), which represent the size and position of the bounding box. This illustrates YOLO's commitment to precision and effectiveness in recognizing object tasks (Redmon et al., 2016).

$$Pr(\text{Class}_i|\text{Object}) \times Pr(\text{Object}|\text{IOU}_{\text{pred}}^{\text{truth}}) = Pr(\text{Class}_i) \times IOU_{\text{pred}}^{\text{truth}} \quad (1)$$

5.3 YOLOv2

Redmon and Farhadi (2017) developed YOLOv2 in 2017 based on the foundation of YOLOv1. The authors aim to enhance both the speed and accuracy of object detection in a real-time system. The concept of anchor boxes and predefined boxes of varying sizes and aspect ratios is introduced to better estimate the location of objects in the image. Additionally, it randomly resizes input images during the training phase to improve the network's capacity to detect objects at different scales. In contrast to YOLOv1, which

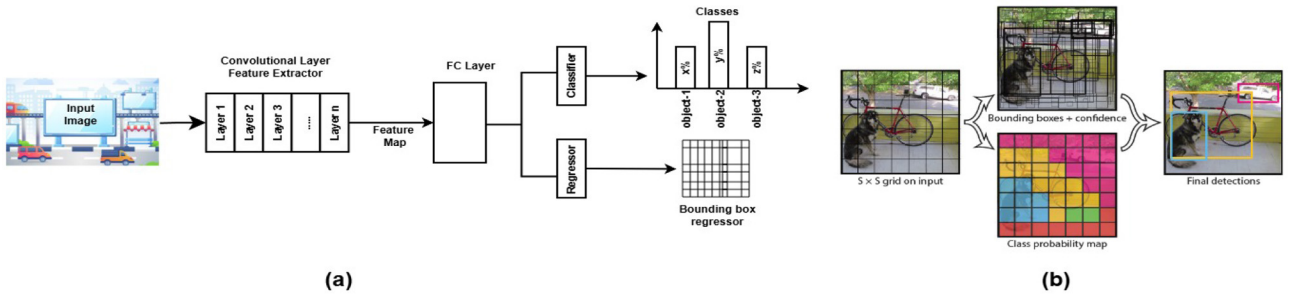

FIGURE 6
General block diagram of the one-stage (regression based) networks object detection model. It provides how detections are formed by merging bounding box predictions with class probabilities. **(a)** Highlights the core modules of single-stage approach and **(b)** data flow of theYOLO V1 object detection framework (Redmon et al., 2016).

relies solely on down-sampling to bolster the high-resolution classifier for small object detection, YOLOv2 utilizes multiple layers to transmit high-resolution features to the detector.

5.4 YOLOv3

With the release of YOLOv3 (Redmon and Farhadi, 2018), real-time object identification has significantly advanced, showcasing impressive improvements in both speed and accuracy. To improve feature extraction performance and mitigate deterioration in deeper neural networks, a robust Darknet-53 architecture is implemented, consisting of 53 convolutional layers with residual connections. The multi-class probabilistic classifier, featuring independent classes and pyramidal forecasts, integrated into YOLOv3's innovative design, revolutionizes object recognition precision. It improves bounding box creation by applying distance penalties using an aggregated intersection over union technique, further improving the model's ability to accurately locate objects within images. Due to its speed and accuracy, YOLOv3 is one of the top choices for real-time object detection applications, such as detection systems and automated vehicles. It can identify objects at large scales and aspect ratios with minimal processing overhead, making it an effective tool for resource-constrained object detection tasks. Its versatile detection capabilities and economical approach enhance its usability.

5.5 YOLOv4 and higher version

Several significant enhancements are included in YOLOv4 (Bochkovskiy et al., 2020) compared to the earlier version for improving object detection performance. First, CSPDarknet53, a more effective backbone, extracts rich features while maintaining a lower computational load. Second, a smoother activation function ensures improved gradient flow and training stability. Thirdly, by focusing on key points and combining data from different sizes, the Spatial Attention Module and Path Aggregation Network improve feature representation. In addition, different anchor boxes and optimized loss functions address variations in object size and enhance localization; a focus method prioritizes high-confidence objects during inference, improving real-time performance. Figure 7 is the representation of DC-SPP in YOLOv4. It highlights its use of spatial pyramid pooling with dilated convolutions to enhance receptive fields and capture multi-scale features for robust object detection. The main enhancements of YOLOv5's (Jocher et al., 2021) small object detection are responsible for its success. Initially, prioritizing areas with high feature values, the layer-focusing technique enables the model to efficiently allocate processing power and provide sharper, more accurate detection, especially for smaller objects. Second, the model's ability to identify and locate small objects, which may appear dim or fuzzy at a single scale, is improved by the "Multi-Scale Feature Fusion" technique, which effectively merges information from various feature maps generated at multiple scales. These improvements highlight YOLOv5's dedication to addressing the difficulties related to recognizing small objects in the object detection domain, significantly improving its overall accuracy, particularly in small object identification. Figure 8a presents the PP-YOLO object detection network, visually illustrating its architecture and showcasing key components such as the backbone, feature pyramid, detection head, and post-processing steps for efficient object detection. Figure 8b presents the architecture of the YOLOv5 object detection model.

The new versions of the object detection model, YOLOv6 (Li et al., 2022) and YOLOv7 (Wang et al., 2023), not only predict objects but also estimate their poses. One of the key features of YOLOv6 and YOLOv7 is their ability to forecast an object's presence and pose. This means that the models provide detailed information about the detected objects, making them essential for applications that require a thorough understanding of the spatial orientation of objects within an image. In particular, YOLOv7 incorporates advanced techniques such as position encoding, level smoothing, and data augmentation. These improvements result in more accurate and versatile object identification systems by improving the models' ability to manage real-world data, reducing noise, and enhancing spatial understanding.

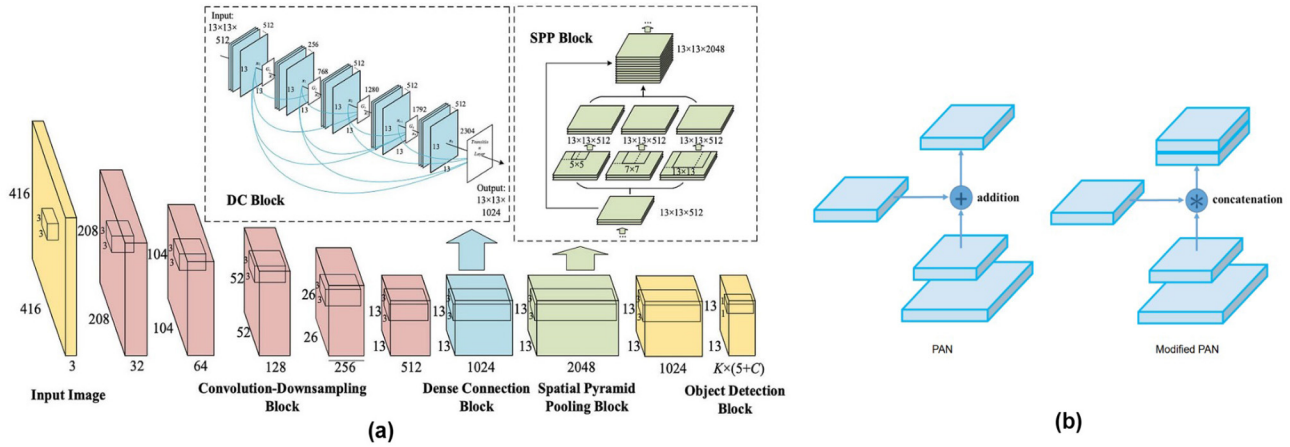

FIGURE 7
Architectural components of the YOLOv4 object detection model **(a)** General representation of DC-SPP used in YOLOv4 Model (Huang et al., 2020) and **(b)** Path aggregation network of YOLOv4 (Bochkovskiy et al., 2020).

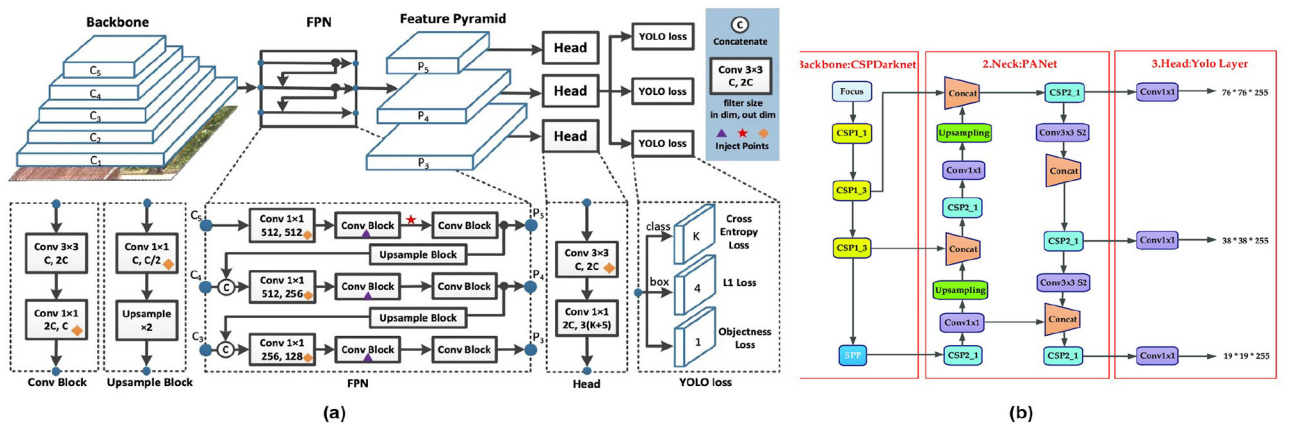

FIGURE 8
Architecture of the YOLOv5 and higher object detection model **(a)** Graphical depiction of the PP-YOLO object detection network (Long et al., 2020) and **(b)** Frameworks description of YOLOv5 (Jocher et al., 2021).

YOLOv8 (Reis et al., 2023) primarily focuses on pose estimation through image segmentation. Higher versions of YOLO represent a continual development process, offering improved accuracy, better identification of small objects, enhanced pose detection, and more accurate detection of cropped images. The table below summarizes the different versions of YOLO, the architecture used, the techniques implemented during development by the respective authors, and their performance evaluated on various standard datasets. Detection accuracy, computational time, and complexity with resources are the key factors distinguishing the different YOLO versions.

YOLOv9 (Wang and Liao, 2024) introduced feedback initialization, attention-based modules, and improved feature pyramids, enhancing the detection of small and distant objects while optimizing multi-scale feature learning for faster, more robust inference. Building on this foundation, YOLOv10 (Wang et al., 2024) incorporated dynamic task prioritization and transformer-based feature extraction, improving the management of complex object interactions and strengthening robustness in challenging scenarios such as occlusions. YOLOv11 (Jocher and Qiu, 2024) further advanced the detection pipeline by implementing cross-domain learning, refining loss functions for better localization, and applying knowledge distillation techniques, enabling efficient training with limited data. Additionally, YOLOv11 achieved state-of-the-art performance with reduced computational overhead, making it highly effective for edge and real-time applications. Collectively, these developments illustrate a trajectory of innovation focused on enhanced feature extraction, robustness, and computational efficiency, positioning YOLOv11 as a versatile model for diverse detection tasks. Table 2 provides a comprehensive overview of various YOLO versions, detailing their advancements in one-stage object detection techniques. It highlights key aspects such as backbone architecture, loss functions, datasets used, and

TABLE 2 Understanding of different YOLO versions of 1-stage object detection techniques.

YOLO version	Researcher	Backbone architecture	Loss function	Used dataset	Accuracy (mAP)	Detection head	References
YOLOv1	Redmon et al. (2016)	GoogleNet	Square Error	VOC 2012 VOC 2007	57.90% 63.40%	Connected Layers, Linear regression	Redmon et al., 2016
YOLOv2	Redmon et al. (2017)	Darknet-19	Logistic regression	COCO dataset VOC 2012 (0.5 IoU threshold) VOC 2007	44.00% 78.60% 78.80%	Multi scale prediction, Connected Layers	Redmon and Farhadi, 2017
YOLOv3	Redmon et al. (2018)	Darknet-53	Binary cross Entropy	COCO VOC 2007 COCO dataset	80.50% 57.90% 33.00%	FPN	Redmon and Farhadi, 2018
YOLOv4	Bochkovskiy et al. (2020)	CSPDarknet-53	Consolidated IoU	COCO	40–61% 62.8% (Ap-50 at over 96 FPS)	SPP PANet	Bochkovskiy et al., 2020
YOLOv5	Glenn Jocher et al. Ultralytics (2021)	CSPDarknet-53	CIOU	COCO	55.8% (YOLO-v5s) 62.4% (YOLO-v5m) 65.4% (YOLO-v5l)	SPPF, CSP-PAN	Jocher et al., 2021
YOLOv6	Li et al. at Meituan (2022)	Efficient REP	CIOU	COCO	43.5% (YOLO-v6-S) 49.7% (YOLO-v6-M) 51.7% (YOLO-v6-L-ReLU)	REPPAN	Li et al., 2022
YOLOv7	Wang et al. (2023)	Extended ELAN	CIOU	COCO	52.8% (YOLO-v7-tiny) 69.7% (YOLO-v7) 71.1% (YOLO-v7-X) 70-84.5%	PAN and SPPCSPC	Wang et al., 2023
YOLOv8	Reis et al. (2023)	modified CSPDarknet53	CIOU with weight loss calculation	COCO and VOC	53.98% (on COCO)	PAN and SPP	Reis et al., 2023
YOLOv9	Wang and Liao (2024)	GELAN	PGI	COCO	Improved accuracy to SOTA	Anchor-free mechanism	Wang and Liao, 2024
YOLOv10	Wang et al. (2024)	Enhanced version of CSPNet	Combines elements of classification loss, localization loss, and objectness loss	COCO	Improved accuracy to SOTA	Adaptive anchor assignment and dynamic label assignment	Wang et al., 2024

10.3389/fcomp.2025.1437664

accuracy metrics across different iterations from YOLOv1 to YOLOv10.

5.5.1 Single Shot MultiBox Detector (SSD)

Liu et al. (2016) introduced the SSD concept in 2015, utilizing a CNN as the backbone architecture for object detection. SSD enables fast object classification and localization in a single forward path. It employs a set of predefined boxes called "anchor boxes" with different sizes and aspect ratios to detect objects at different locations in an image. The network attempts to predict offsets and adjusts these anchor boxes to accurately fix detected objects according to their size and location using the deep learning techniques. For each object class, the anchor box receives a confidence score to indicate the likelihood of the presence of an object within the box. SSD architecture is divided into two major parts: firstly, the backbone model, a pre-trained classification model, which is a feature map extractor, and secondly, the SSD head, which is moved to the top of the backbone model. This SSD head will provide the bounding box as output over any detected object, resulting in a fast and efficient object detection model. Compared to YOLO, where the object detection method is used to run on different layers at different scales, SS and D run only on the top layers. Similarly, relying on the COCO7 dataset, the tiny SSD has performed better with reliability than the tiny SSD (Womg et al., 2018).

5.5.2 RetinaNet

Lin et al. (2017b) revolutionized object detection by introducing a groundbreaking concept that enhances both accuracy and efficiency through a novel loss function. Rather than using the traditional cross-entropy loss, they proposed the Focal Loss function, which is specifically designed to address the challenges associated with class imbalance during training (Lin et al., 2017b). This innovation allows the single-stage RetinaNet object detector to achieve exceptional accuracy, particularly for small and densely packed objects in images. The model employs a robust backbone network architecture along with two specialized subnetworks, which operate seamlessly at multiple scales to detect objects with precision. The backbone processes input images of varying sizes to compute convolutional feature maps, while the subnetworks manage object classification: one is embedded within the backbone for feature extraction, and the other focuses on the bounding boxes of detected objects. Collectively, these components synergistically improve detection performance within this single-stage framework.

RetinaNet incorporates two pivotal upgrades: Focal Loss and Feature Pyramid Networks (FPN), which redefine its capabilities. Focal loss mitigates the impact of class imbalance caused by the prevalence of background classes or numerous anchor boxes, effectively diminishing the loss contributions of easy-to-classify samples while focusing on more complex cases. Meanwhile, by leveraging a multi-scale feature extraction strategy, FPN enables RetinaNet to excel across varying object scales. Constructing an image pyramid captures critical features at different layers, allowing for precise detection of objects regardless of size. However, the convolutional process within the CNN architecture naturally reduces feature map sizes at deeper layers, forming a hierarchical, pyramid-like structure that is ideal for multi-scale detection. Figure 9 provides a detailed illustration of the RetinaNet architecture, showcasing its innovative design.

In addition to YOLO, SSD, and RetinaNet, other similar one-stage architectures are used in object detection. Popular examples include SqueezeDet (Wu et al., 2017), DSSD (Fu et al., 2017), DenseNet (Huang et al., 2017), and CornerNet (Law and Deng, 2018). SqueezeNet enhances accuracy in large object detections with its fire module backbone architecture but is limited on mobile platforms. Deconvolutional layer SSD (DSSD) is more efficient for dense and smaller objects in images, utilizing multi-scale predictions for higher accuracy results. However, it consumes significant memory and has slower performance. DenseNet SSD incorporates the feature reuse concept within the SSD framework, serving as a balance between accuracy and resource utilization. CornerNet represents a unique style of object detection, identifying objects through keypoint estimation techniques that can accommodate rotated objects in images; nonetheless, its computational complexity is excessively high. PAA-SSD is the latest model that focuses on one specific type of system, integrating with various SSD model-based platforms to improve accuracy through the probabilistic anchor assignment technique. The results depend on the dataset used and the chosen backbone architecture. Additionally, it may increase the computational complexity of training based on the chosen platform, necessitating careful assignment. Table 3 illustrates different one-stage object detection techniques along with a comparative analysis. The variation in datasets for various object detections, taking into account not only size but also purpose, is considered for the results obtained.

5.6 Comprehensive review of two-stage (region based) networks object detection model

Two-stage or region-based deep learning approaches rank among the most prominent models for achieving accurate object detection in images. These methods excel by employing a two-step process. The Region Proposal Network (RPN) focuses solely on areas containing objects, thereby avoiding an exhaustive search across the entire image. Unlike brute-force methods, RPNs enhance detection accuracy by training on data relevant to object-specific regions, facilitating precise and efficient classification. This method is especially beneficial for real-time applications, as RPNs identify potential object regions, allowing the classification stage to refine bounding boxes for accurate localization. Both stages present opportunities for customization, with advanced network architectures designed to meet the specific requirements of RPN and classification. This section summarizes and analyzes popular two-stage object detection models, comparing them across factors such as speed, accuracy, computational complexity, and advancements proposed by various researchers.

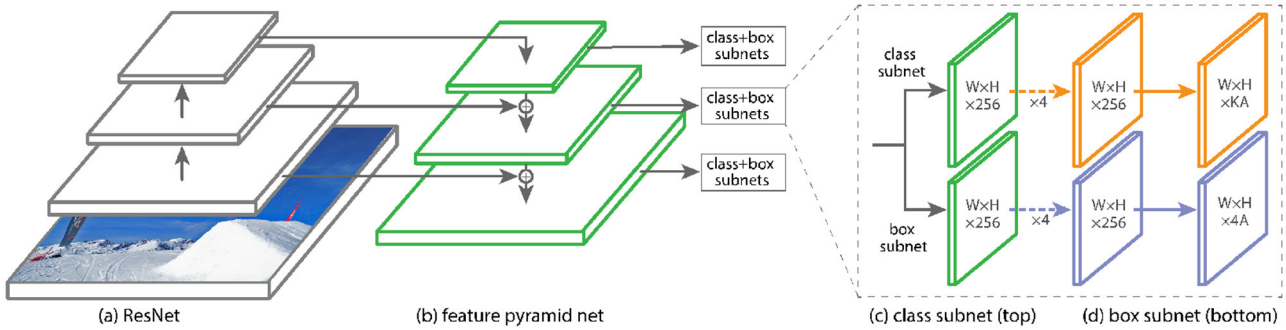

FIGURE 9
Object detection using RetinaNet (Lin et al., 2017b).

TABLE 3 Understanding of Different 1-Stage Object Detection Techniques beside YOLO.

Model	Researcher	Backbone	Key Parameter	Dataset	Input Size	References
SSD	Liu et al.	VGG-16	Simple, 1-stage	VOC 2007	300x300	Liu et al., 2016
SqueezeDet	Wu et al.	Fire modules	Better accuracy	COCO 2017	Various	Wu et al., 2017
RetinaNet	Lin et al.	Resnet-50	Improved accuracy	COCO 2017	800x800	Lin et al., 2017b
DSSD	Fu et al.	VGG-16	Small Objects prediction	VOC 2007	300x300	Fu et al., 2017
DenseNet	Huang et al.	DenseNet	Higher accuracy	VOC 2007	300x300	Huang et al., 2017
MobileNet	Sandler et al.	MobileNet	Lightweight	VOC 2007	300x300	Sandler et al., 2018
Mobiledets	Xiong et al.	MobileNet	Fast and real-time	COCO	300x300	Xiong et al., 2021
CornerNet	Law and Deng	Hourglass network	Key-point detection	COCO 2017	Various	Law and Deng, 2018
NAS-FPN	Ghiasi et al.	Neural Architecture	Optimized FPN	COCO 2017	Various	Ghiasi et al., 2019

5.6.1 Region-based Convolutional Neural Network

In 2014, Girshick et al. (2014b) introduced a seminal approach to object detection by incorporating CNNs to enhance detection accuracy and improve bounding box quality through deep feature extraction. This method achieved a significant milestone, attaining a mean Average Precision (mAP) of 53.4%, a remarkable improvement over contemporaneous models. The model was trained on the PASCAL VOC 2012 benchmark dataset, setting a new standard for object detection tasks. A simplified representation of the RCNN process is illustrated in Figure 10. The RCNN process comprises two primary stages: region proposal and feature extraction with classification. In the region proposal stage, the entire image is scanned using a selective search algorithm, which evaluates features such as color, texture, position, and location to generate candidate regions likely to contain objects. These candidate regions are resized to conform to the input dimensions the applied CNN requires.

In the feature extraction and classification stage, the resized regions are processed using a pre-trained CNN model to extract high-level features such as color, shapes, textures, and edges. The extracted features are then input into two distinct support vector machines (SVMs): one for object classification and the other for bounding box refinement. The classification SVM predicts the object class (e.g., car, airplane, chair, person, cat), while the bounding box refinement SVM fine-tunes the proposed bounding box to ensure better localization of the detected object. The SVM assigns a score to each class through non-maximum suppression while maintaining the Intersection over Union (IoU) below a predefined threshold, further enhancing detection precision.

RCNN pioneered object detection by leveraging deep neural networks to extract hierarchical features from images, capturing multi-scale information across layers for precise detection. The model classifies objects and generates bounding boxes around detected regions. However, RCNN also has notable limitations. The fully connected layers in the CNN necessitate resizing images to a fixed size of 277 × 277, which increases computational overhead. The selective search algorithm generates thousands of potential regions, resulting in inefficiencies and high computation time. Additionally, processing these regions individually leads to redundant computations, and the SVM-based classification introduces bottlenecks that hinder speed and optimization. These challenges limit RCNN's performance in real-time applications, complex image backgrounds, and small object detection.

To address these issues, several advancements have been proposed. For instance, Zhang et al. (2015) tackled inaccurate localization in RCNN by introducing three key improvements: (1) Bayesian optimization to refine bounding boxes by evaluating classification scores and localization accuracy; (2) structured loss to penalize inaccuracies in predicted bounding boxes; and (3) class-specific CNNs to enhance accuracy for diverse object categories (Zhang et al., 2015). Furthermore, adopting superpixel classification can refine object boundaries and improve efficiency in handling complex scenes and small objects. However, careful

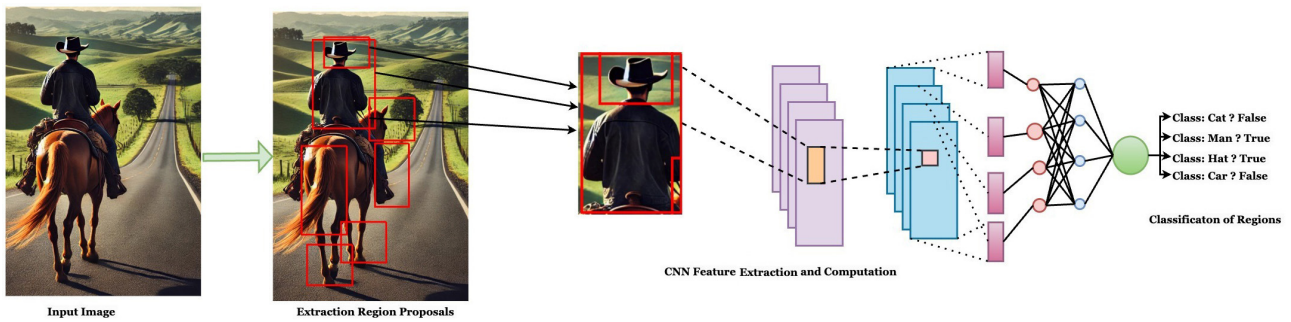

FIGURE 10
Region-based convolutional neural network (Girshick et al., 2014b).

implementation is needed to mitigate potential segmentation inaccuracies.

The fixed filter sizes used in CNN training, along with challenges such as object rotation, deformation, and pose variation, were addressed by Ouyang et al. using deformation-constrained pooling layers. Their approach used guided deformable filters to adaptively adjust shape and size, predicting offset values for local object alignment while applying geometric penalties to promote meaningful deformations. This method, integrated into the DeepID-Net, demonstrated improved accuracy on the ISVRC 2014 dataset (Ouyang et al., 2015).

The limitations of anchor-based bounding boxes, which affect small object detection and computational efficiency, were mitigated by the DeepBox anchor-free design proposed by Zhang et al. This method detects objects without predefined anchors, facilitating better localization of small objects with flexible bounding box shapes and orientations, although it remains sensitive to hyperparameter tuning. Additionally, Pinheiro et al. (2016) introduced SharpMask for refining critical regions, providing superior performance in managing overlapping objects and complex scenes, albeit at the expense of increased computational requirements due to the attention mechanism. Understanding these advancements and their trade-offs provides valuable insights into the suitability of various RCNN-based models for specific tasks. Models such as SPPNet (He et al., 2015a) and Fast RCNN (Girshick, 2015) build upon the foundation of RCNN, delivering significant improvements in efficiency and performance, which will be explored in subsequent sections.

5.6.2 Spatial Pyramid Pooling Network (SPPNet)

In the context of RCNN, the CNN model requires input images to be of a fixed size, creating challenges when handling images of varying dimensions. This necessitates resizing, which can lead to information loss, reduced accuracy, and increased computational overhead during the scaling process. The Spatial Pyramid Pooling Network (SPPNet) was introduced to address this limitation, enabling the processing of variable-sized input images. SPPNet employs a spatial pyramid pooling mechanism that divides the input image into pyramids of subregions, extracts features from each subregion, and pools them into a fixed-size representation that is independent of the original image dimensions. This approach enhances the model's flexibility and scale invariance. Furthermore, SPPNet allows feature extraction from multiple convolutional layers at different resolutions, facilitating improved object localization and mitigating the resolution reduction issue inherent in RCNN. By integrating multi-scale feature extraction and enabling efficient handling of variable image sizes, SPPNet significantly improves object detection models' flexibility, scalability, and robustness. It is particularly effective in managing diverse image sizes and complex backgrounds, making it a valuable advancement over traditional RCNN methods. Figure 11 illustrates pivotal architectures in object detection. Fast R-CNN enhances region-based convolutional networks by integrating RoI pooling and shared convolutional features, while SPPNet introduces spatial pyramid pooling to efficiently handle input images of varying sizes (Kaur and Singh, 2023).

Lazebnik et al. (2006) introduced the groundbreaking concept of SPM for object detection, which captures the spatial information of an image by dividing it into multiple subregions and extracting features at various scales. This innovative approach enables the representation of spatial relationships within the image, enhancing feature extraction and localization. Building on this concept, SPPNet incorporates several key advancements. SPPNet efficiently handles images of various sizes, providing scale-invariant detection capabilities. Leveraging multi-scale features extracted from different subregions significantly improves localization accuracy and enhances overall detection performance. These features collectively make SPPNet a robust and accurate model for object detection, particularly in scenarios involving diverse image sizes and complex spatial structures.

5.6.3 Fast R-CNN

The primary limitation of R-CNN lies in its slow processing speed and high computational cost, primarily due to its dependence on selective search for region proposals. Addressing this issue, Girshick (2015) introduced Fast R-CNN, a model that significantly enhances detection speed while maintaining high accuracy. In Fast R-CNN, the RPN directly generates region proposals from image features, eliminating the inefficiency of exhaustive region searches and reducing computational overhead. This approach accelerates

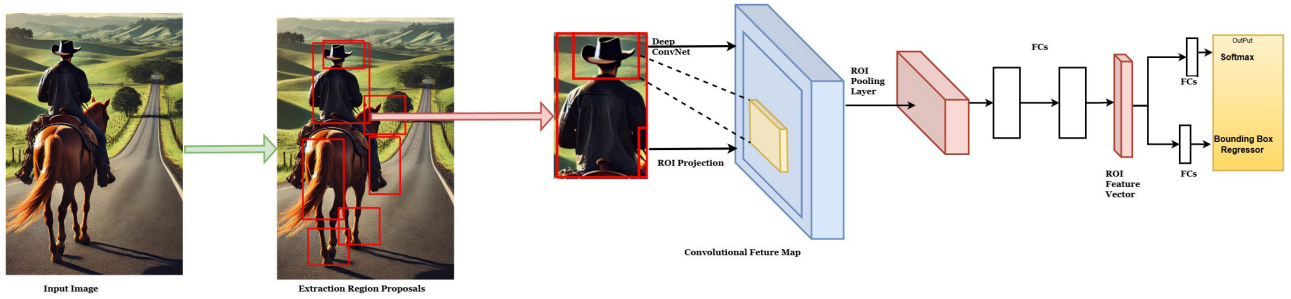

FIGURE 11
(a) Fast R-CNN architecture (Girshick, 2015) with RoI pooling for object detection. **(b)** Spatial Pyramid Pooling Network (SPPNet)(He et al., 2015a) for fixed-length feature representation, enabling multi-scale feature extraction.

the process and ensures accurate object detection, as presented in Figure 11a.

Fast R-CNN employs a multi-task learning strategy, jointly training the RPN and the classifier to optimize region proposal generation and object detection in a unified pipeline. This integrated framework improves accuracy compared to traditional pipeline-based methods. To further enhance efficiency, Fast R-CNN leverages pre-trained CNN models, such as VGG or ResNet, which are trained on large-scale datasets such as ImageNet. These pre-trained networks capture hierarchical features, ranging from low-level patterns to high-level abstractions, enabling precise analysis of specific regions. The challenge of processing variable-sized regions of interest in fully connected layers, prevalent in R-CNN, is addressed in Fast R-CNN by introducing ROI pooling. ROI pooling divides each ROI into fixed-size subgrids, extracting uniformly sized features for the fully connected layers, thus ensuring consistency and improving detection accuracy.

For region proposal generation, ROI pooling utilizes features extracted from the RPN, avoiding external algorithms such as selective search. The softmax layer classifies objects within the image by predicting the probability of each object class, resulting in a $K + 1$-dimensional vector for K object classes, with the additional dimension representing the background category. The class with the highest probability is assigned to the detected object. The bounding box regression branch also employs linear regression to refine the predicted bounding box coordinates. Offset values, derived from ROI-pooled features, are added to the initial ROI coordinates to improve bounding box precision, ensuring accurate localization of objects. Fast R-CNN represents a significant advancement over its predecessor, achieving superior speed and accuracy in object detection tasks (Girshick, 2015).

The multi-task loss L of Fast R-CNN is jointly expressed with the two output layers, specifically training for classification and bounding box regression for each labeled ROI. For the trained model, the discrete probability distribution p, computed by a softmax over $K+1$ categories per ROI from a fully connected layer, is defined by

$$p = (p_1 + p_2 + \ldots + p_n) \tag{2}$$

the output bounding box regression offset is given by

$$t^k = (t_x^k, t_y^k, t_w^k, t_h^k). \tag{3}$$

Where K is the object class indexed by k. The Iverson bracket indicator function $[u \geq 1]$ is employed to omit all background RoIs (Girshick et al., 2014b).

$$L(p,\ u,\ t^u,\ v) = L_{\text{cls}}(p,\ u) + \lambda[u \geq 1]L_{1\text{oc}}(t^u,\ v) \tag{4}$$

in which $L_{\text{cls}}(p,\ u)$ is log loss for true class u.

The second task loss, L_{LOC}. The bounding-box regression targets for the class $(u, v) = (v_x, v_y, v_w, v_h)$ and the predicted tuple $\text{tu} = (tu_x, tu_y, tu_w, tu_h)$. For the bounding box regression, we use the loss

$$L_{1\text{oc}}(t^u,\ v) = \sum_{i\in\{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i), \tag{5}$$

in which

$$\text{smooth}L_1(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \tag{6}$$

Here the L1 loss is less sensitive to the outliers than the L2 loss used in R-CNN and SSPNet.

5.6.4 Faster R-CNN

Although Fast R-CNN improves detection speed and accuracy compared to its predecessor, it still has limitations in terms of optimization and efficiency. Fast R-CNN relies on pre-trained feature extraction and external ROI pooling mechanisms that utilize fixed-size features and a softmax bounding box classifier. This dependence on external algorithms for region proposals can introduce inefficiencies, including slower processing and potential inaccuracies. Additionally, the requirement for separate stages in the training process decreases overall efficiency. Faster R-CNN was introduced to address these issues by integrating RPN with the CNN architecture into a unified framework. This design eliminates the dependence on external algorithms, resulting in significantly higher speeds than Fast R-CNN. The model trains the entire pipeline jointly, enhancing efficiency, accuracy, and effectiveness. Furthermore, Faster R-CNN is better equipped to handle diverse

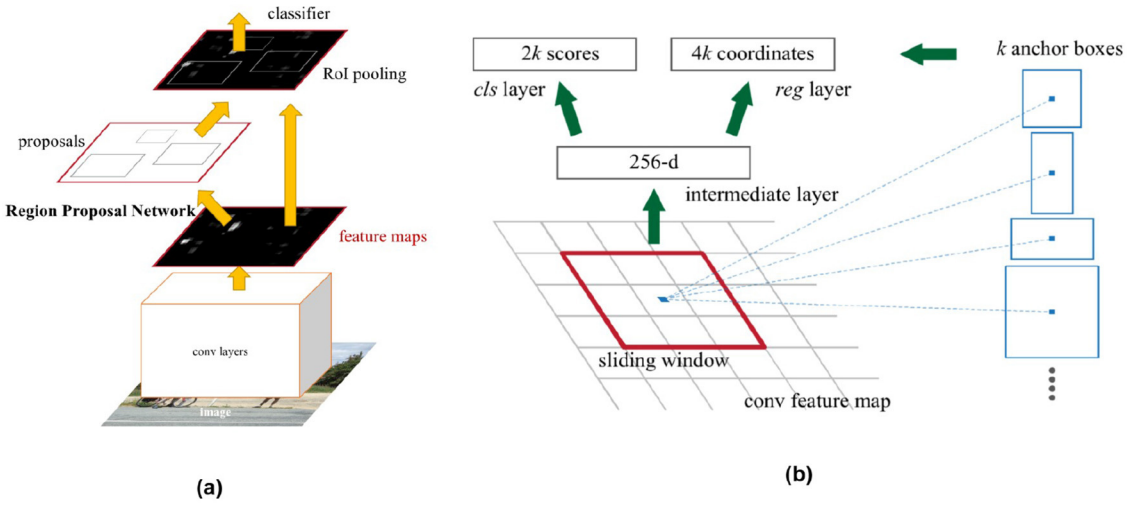

FIGURE 12
(a) Faster R-CNN (Ren et al., 2016) and **(b)** Region Proposal Network (RPN) (Ren et al., 2016). The diagram effectively illustrates the architecture of the Faster R-CNN model, highlighting the interactions of RPN and other essential components.

datasets, improving its performance in real-time applications (Ren et al., 2015).

In the architecture of Faster R-CNN, the RPN slides a small spatial window over the feature map, connecting to an $n \times n$ spatial region. For instance, with VGG16, a low-dimensional vector of size 512 is extracted within the sliding window and passed to two fully connected (FC) layers: one for box classification (cls) and another for box regression (reg). This architecture incorporates an $n \times n$ convolutional layer connected to two 1×1 convolutional layers, as depicted in the corresponding (Figure 12).

Bounding box regression is achieved by refining the proposals in relation to the reference boxes. The model utilizes anchors of three different scales and three aspect ratios, which improve detection for objects of various sizes and shapes. The loss function in Faster R-CNN is similar to that of Fast R-CNN, maintaining a balance between classification accuracy and bounding box regression.

The loss function is given by,

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (7)$$

Where

Pi: Predicted probability of anchor box i containing an object (foreground).

ti: Predicted bounding box coordinates (4 values: x, y, width, height) for anchor box i.

p*i: Ground truth label for anchor box i (1 for foreground, 0 for background).

t*i: Ground truth bounding box coordinates for the object associated with positive (foreground) anchor box i.

Ncls: Number of anchor boxes per image in the mini-batch during training.

Nreg: Number of positive (foreground) anchor boxes in the image.

λ: Hyperparameter balancing the importance of classification and regression tasks.

$L_{\text{cls}}(p_i, p_i^*)$: Classification loss for anchor box i, often binary cross-entropy.

$L_{\text{reg}}(t_i, t_i^*)$: Regression loss for the predicted bounding box of anchor box i, often Smooth L1 loss.

5.6.5 Mask R-CNN

Mask R-CNN is a robust deep learning framework for object detection and instance segmentation. During object detection, it identifies and localizes objects within an image while incorporating instance-level context, which enables precise recognition of what the objects are and their locations. In the segmentation phase, Mask R-CNN goes beyond bounding boxes to create pixel-level masks for individual objects, providing superior precision. It can accurately segment various objects, such as cars, cats, bicycles, or billboards, even in challenging conditions such as partial occlusion or shadowed regions.

Mask R-CNN addresses key limitations of Faster R-CNN, such as the inability to segment individual objects within the same class or differentiate between multiple instances (e.g., distinguishing people in a crowd). It also reduces the computational overhead of storing intermediate features, thereby improving efficiency. By incorporating fine-tuning mechanisms, Mask R-CNN enhances detection accuracy and optimization, making it a robust solution for tasks such as autonomous navigation. Additionally, its end-to-end network training provides better optimization and performance compared to Fast and Faster R-CNN, establishing it as a versatile and reliable tool for image analysis.

Introduced by He et al. (2017), Mask R-CNN enhances the capabilities of Faster R-CNN by adding instance segmentation at the pixel level. Its innovation lies in integrating a mask prediction branch alongside the bounding box classifier, addressing the spatial limitations of Faster R-CNN's bilinear interpolation with a novel sampling technique that preserves spatial information.

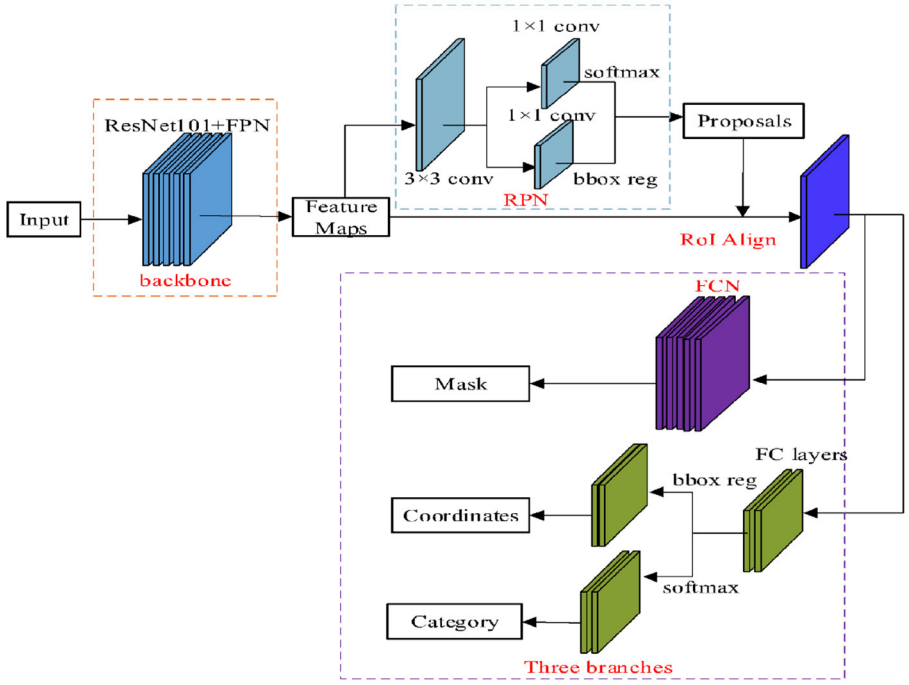

FIGURE 13
Architecture of mask R-CNN for instance object detection (He et al., 2017).

By combining region proposal and classification, this unified network architecture improves training efficiency by eliminating intermediate feature storage and optimizing the entire network through end-to-end learning. This architecture is depicted in Figure 13.

The core of Mask R-CNN is built upon Faster R-CNN, incorporating additional components such as ROI Align, shared feature pooling, and a mask prediction branch. Key architectural elements include the backbone feature extractor, RPN, shared pooling layers, detection and segmentation branches, and a multi-task loss function for joint optimization. Table 4 summarizes the performance of various models, with segmentation showing notable results. The findings demonstrate significant improvements in AP across different backbone architectures, highlighting the effectiveness of Mask R-CNN in this domain.

The foundational architecture of Mask R-CNN is Faster R-CNN. The in-depth architectural components of Mask R-CNN include the backbone feature extractor, RPN, shared feature pooling, detection, segmentation, and the calculation of the training and loss functions. ROI Align, shared features, and end-to-end training are additional components of Mask R-CNN compared to Faster R-CNN.

The backbone extractor, typically ResNet (50/101) or a VGG variant, captures complex features to enhance detection accuracy. Feature maps derived from this backbone provide rich semantic information about the input image. Anchors within the RPN are adapted to the objects' shape and size, and convolutional layers predict the presence of objects and bounding boxes. Non-maximum suppression (NMS) ensures efficient processing by suppressing redundant regions and selecting high-confidence proposals. Shared feature pooling, specifically ROI Align, preserves spatial information while resizing features for consistent mask prediction. For each ROI, the classification branch predicts object classes using fully connected layers and a softmax activation function, while the bounding box regression branch refines localization. The mask branch generates binary masks for ROIs, and skip connections enhance the network's ability to capture object shapes and extents. A multi-task loss function optimizes classification, bounding box regression, and mask prediction simultaneously, enabling robust performance through end-to-end training. Despite its computational complexity and high hardware requirements, Mask R-CNN remains a state-of-the-art tool for computer vision applications (He et al., 2017).

5.6.6 Feature Pyramid Network

The concept of Feature Pyramid Networks (FPN) was developed by researchers (Lin et al., 2017a) to address the challenges associated with traditional object detection methodologies. Specifically, FPN aims to resolve two primary issues: the loss of spatial information due to down-sampling and the limited semantic information that can hinder accurate object detection. Furthermore, FPN partially mitigates the limitations inherent in Mask R-CNN, which employs traditional CNNs that often experience reduced spatial resolution, complicating the precise localization of small objects within images. FPN integrates bottom-up and top-down pathways to produce multi-scale feature maps while maintaining semantic information. This architecture enhances detection capabilities for small objects across various resolutions and sizes. The semantic gap in feature maps derived from different levels in Mask R-CNN can significantly degrade detection accuracy, particularly in cluttered scenes. To address

TABLE 4 Mask R-CNN performance for instance segmentation (He et al., 2017).

Model	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50
Mask RCNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask RCNN	ResNet-101-FPN	35.7	58	37.8	15.5	38.1	52.4
Mask RCNN	ResNetXt-101-FPN	37.1	60	39.4	16.9	39.9	53.5

this, FPN employs lateral connections that bridge this gap by injecting high-level semantic information from deeper layers into the feature maps. This ensures accurate object identification along with location and class information (Chhabra et al., 2023).

A notable advantage of FPN is its ability to reuse features computed within the CNN backbone, which minimizes computational overhead. This resource-efficient design enables the construction of multi-scale features without creating pyramids from scratch, thereby enhancing computational efficiency. As a result, FPN demonstrates improved accuracy while maintaining low computational complexity across diverse applications. It excels at detecting small objects of varying sizes with high accuracy and efficiency. During the detection process, FPN effectively integrates both bottom-up and top-down approaches. The bottom-up pathway captures fine spatial details with high semantic value using existing convolutional networks, although it may lack semantic richness and exhibit lower resolution. Conversely, the top-down pathway begins with the deepest feature map and progressively upsamples it while merging it with shallower maps through lateral connections. This synthesis results in a comprehensive representation that combines high-level semantic information with preserved low-level spatial details. Figure 14 presents how features are extracted at multiple scales, resulting in a feature pyramid that captures information at different levels of detail and abstraction.

5.6.7 CentripetalNet

CentripetalNet demonstrates higher prediction accuracy than the bounding box approach in FPN. It achieves fine-grained localization of potential objects within an image and delivers superior performance in challenging scenarios, such as dense or crowded scenes and partially visible objects. The architecture of CentripetalNet, illustrated in Figure 15, leverages key points for object detection. Kivee (Dong et al., 2020) developed CentripetalNet to pursue high-quality keypoint pairs for object detection, addressing issues related to inaccurate keypoint matching and limited feature integration, which often result in the loss of spatial context and essential information for effective object detection.

This recent object detection approach relies on key points instead of bounding boxes, predicting primary objects based on the location and relationships of corner key points. Initially, the model predicts the corner key points associated with each object and utilizes a shift vector, known as the centripetal shift, to guide these points toward the object's center. To pair corresponding key points within the same object, it employs predicted shift values in a process called shift matching, which is particularly useful when the points are initially scattered. Corner pooling extracts features from the area surrounding each corner point with sufficient precision to represent them as detected objects. Finally, deformable convolutions are employed to refine the exact shape of the object in real-time. Table 5 provides a comparative evaluation of object detection performance on the MS-COCO test-dev dataset, focusing on various detection methodologies and their performance metrics. Key indicators, including Average Precision (AP), AP at different Intersection over Union (IoU) thresholds (AP_{50}, AP_{75}), and performance across small, medium, and large object scales, are presented. The findings indicate that multi-scale approaches, particularly those employing Centernet511 and CetripetalNet, exhibit enhanced performance across all assessed metrics, highlighting their efficacy in object detection tasks.

5.6.8 Dual-path aggregation network (D2Det)

Cao et al. (2020) introduced an aggregation network for object detection that significantly enhances accuracy while maintaining high processing speed, making it suitable for real-time applications. D2Det addresses the limitations of traditional methods by employing a dual-path aggregator that integrates high spatial detail from low-resolution features with rich semantic information from high-resolution features. This design balances the trade-off between accuracy and classification efficiency. D2Det selectively applies deformable convolutions at specific stages to improve feature learning, optimizing the balance between system performance and computational efficiency. Furthermore, lightweight layers ensure faster processing speeds, making the architecture highly suitable for real-time tasks. The simplified yet advanced design of D2Det has led to its adoption in various real-time domains, demonstrating robust performance and scalability.

5.6.9 TridentNet

Li et al. (2019) developed TridentNet, a multi-branch architecture featuring a three-branch structure aimed at addressing scale variations in object detection. Each branch processes distinct field parameters to specialize in detecting objects of various sizes. The low-resolution branch captures fine-grained details of small objects, the middle-resolution branch balances detail and semantic information for medium-sized objects, and the high-resolution branch focuses on the semantic representation of large objects.

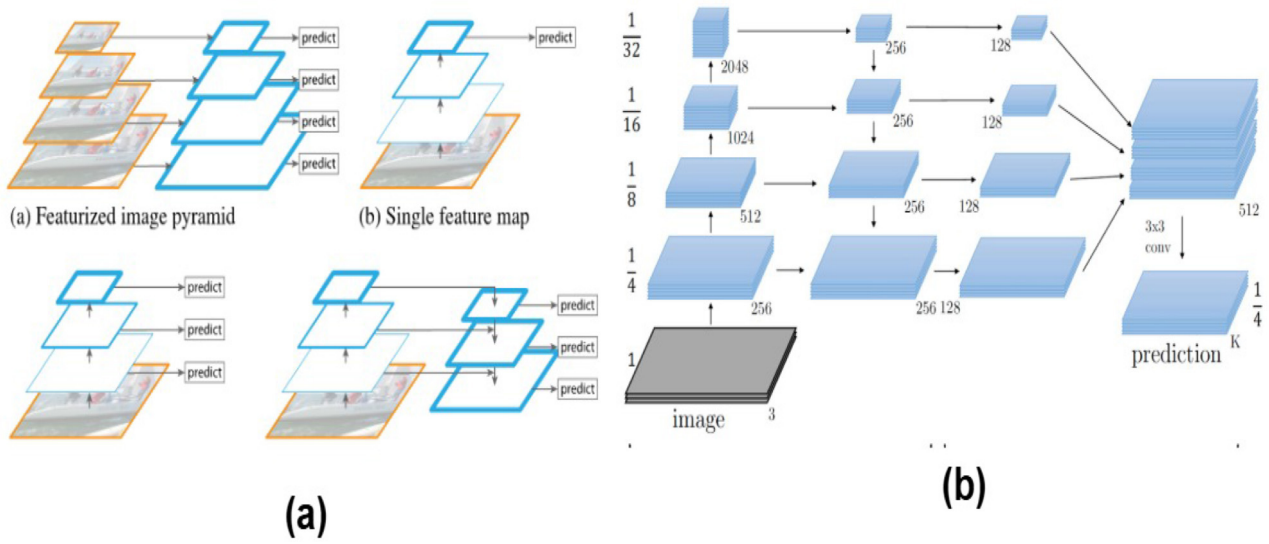

FIGURE 14
(a) Featured image pyramids. Lin et al. (2017a) fundamental idea behind FPNs. **(b)** Feature Pyramid Network. Lin et al. (2017a) that shows how the network takes an input image and generates a single feature map for key concepts and architectural details of FPNs.

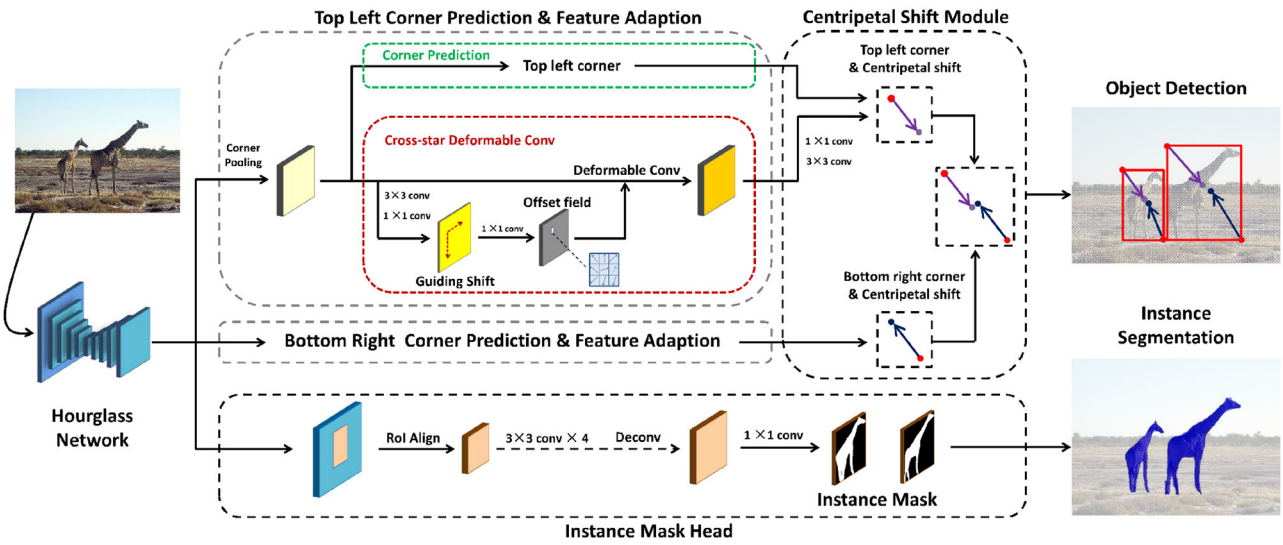

FIGURE 15
Architecture of CentripetalNet for instance object detection (Dong et al., 2020).

During the training phase, TridentNet segments the image based on the size of the objects, with each branch functioning on its corresponding scale. To achieve an efficient design with low computational complexity, TridentNet shares weights among the branches, setting it apart from conventional multi-scale approaches (Alzubaidi et al., 2021). This architecture provides improved accuracy and effectiveness for detecting objects across diverse scales, as demonstrated in its performance comparison with other architectures in Table 6.

The methods discussed above emphasize multitasking, multi-scaling, and contextual detection to manage objects of varying sizes and complexities in images. In multitask learning, they detect objects using bounding boxes of regular shapes, classify the detected objects, and estimate key points, particularly corner points, while assessing object depth. Multi-scale representations facilitate the detection of objects at different scales by extracting and integrating features from various resolutions within an image. Contextual modeling focuses on understanding the relationships between objects and their surrounding backgrounds, enabling the differentiation of overlapping objects through spatial information. This approach enhances scene comprehension, leading to more accurate object detection in complex environments. Addressing these aspects improves detection accuracy, particularly in challenging scenarios.

TABLE 5 Comparison of object detection performance on the MS-COCO test-dev dataset for various methods, highlighting metrics such as Average Precision (AP), AP_{50}, AP_{75}, and performance across small (AP_S), medium (AP_M), and large (AP_L) object scales for single-scale and multi-scale evaluations (Dong et al., 2020).

Methods	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ExtremeNet (single-scale)	Hourglass -104	40.2	55.5	43.2	20.4	43.2	53.1
CornerNet511(multi-scale)	Hourglass -104	42.1	57.8	45.3	20.8	44.8	56.7
ExtremeNet (multi-scale)	Hourglass -104	43.7	60.5	47	24.1	46.9	57.6
Centernet511(single-scale)	Hourglass -104	43.7					
Centernet511(multi-scale)	Hourglass -104	**47**	**64.5**	**50.7**	**28.9**	**49.9**	**58.9**
CetripetalNet w./o mask (single-scale)	Hourglass -104	45.8	63	49.3	25	48.2	58.7
CetripetalNet w./o mask (multi-scale)	Hourglass -104	47.8	65	51.5	28.9	50.2	59.4
CetripetalNet (single-scale)	Hourglass -104	46.1	63.1	49.7	25.3	48.7	59.2
CetripetalNet (Multi-scale)	Hourglass -104	**48**	**65.1**	**51.8**	**29**	**50.4**	**59.9**
Instance segmentation performance comparison							
CetripetalNet (single-scale)	Hourglass -104	38.8	**60.4**	41.7	19.7	41.3	**51.3**
CetripetalNet (Multi-scale)	Hourglass -104	**40.2**	62.3	**43.1**	**22.5**	**42.6**	52.1

The bold values (with highest score) indicate the best performance.

TABLE 6 Two-stage object detection architectures summary.

Architecture	Authors (Year)	Region proposal	Bounding box prediction	Loss function	References
R-CNN	Girshick et al. (2014)	Selective Search	SVM classification	SVM hinge loss	Girshick et al., 2014a
Fast R-CNN	Girshick et al. (2015)	RPN (CNN-based)	Shared CNN layers with region-specific features	Multi-task learning with SGD: Smooth L1 loss, softmax loss	Girshick, 2015
Faster R-CNN	Ren et al. (2015)	RPN (CNN-based)	Shared CNN layers with RoI Pooling	Multi-task learning with SGD: Smooth L1 loss, softmax loss	Ren et al., 2015
Mask R-CNN	He et al. (2017)	RPN (CNN-based)	Shared CNN layers with RoI Pooling and mask prediction branch	Multi-task learning with SGD: Smooth L1 loss, softmax loss, binary cross-entropy loss	He et al., 2017
Cascade R-CNN	Cai and Nuno (2018)	RPN (CNN-based)	Multi-stage refinement with residual connections	Multi-stage learning with SGD: Smooth L1 loss, softmax loss	Cai and Nuno, 2018
RetinaNet	Lin et al. (2017)	FPAN-based anchor generation	FPN-based multi-level prediction with focal loss	Focal loss, multi-task learning with SGD: Smooth L1 loss, softmax loss	Lin et al., 2017b
PolarMask	Xie et al. (2020)	RPN (CNN-based)	Shared CNN layers with RoI Pooling and mask refinement branch	Multi-task learning with Adam: Smooth L1 loss, softmax loss, binary cross-entropy loss	Xie et al., 2020
NAS-FPN	Ghiasi et al. (2019)	RPN (CNN-based)	Shared CNN layers with NAS-designed FPN	Multi-task learning with SGD: NAS-optimized loss function	Ghiasi et al., 2019
Deformable DETR	Zhu et al. (2021)	Transformer-based proposal generation	Set Transformer-based bounding box prediction	Hungarian loss	Zhu et al., 2021

5.7 Hybrid approach of object detection model

In addition, algorithms such as Cascade R-CNN (Cai and Nuno, 2018) utilize a cascaded framework based on Faster R-CNN, merging the strengths of both stages. In the first stage, it generates feature maps for proposal generation and coarse classification, while the second stage refines these feature maps to improve accuracy. This combination of features enhances detection precision and achieves superior performance on benchmarks. Similarly, Mask R-CNN, which includes a cascade head, integrates both stage concepts by utilizing a cascading structure for mask prediction, thus refining bounding boxes and improving prediction accuracy. This method excels in instance segmentation by delivering accurate mask predictions. Similarly, Libra R-CNN (Pang et al., 2019) utilizes a hybrid approach that alternates between YOLOv2 (one-stage) and Faster R-CNN (two-stage) depending on the confidence score. YOLOv2 manages initial predictions, while Faster R-CNN provides further refinement, balancing speed and accuracy. Another notable method is RetinaNet-RegNet

citepxu2022regnet, a hybrid object detection technique that incorporates the RegNet backbone into the RetinaNet architecture. This method improves the detection of small objects while ensuring robust performance in multi-scale and multi-class object detection. Table 7 summarizes various hybrid approaches in object detection algorithms, highlighting their region proposal methods, use of softmax, and loss functions.

5.8 CornerNet and CornerNet lite

CornerNet (Law and Deng, 2018) and CornerNet-Lite (Law et al., 2019) are complex object detection algorithms designed for irregularly shaped objects. These architectures utilize a keypoint-based approach to predict objects rather than relying on anchor boxes. This design enhances robustness to object orientation by being rotation-invariant and achieves higher accuracy while eliminating the hyperparameter tuning associated with anchor boxes. CornerNet employs a single-stage architecture consisting of three key steps: corner heatmap prediction to identify the corner points of objects throughout the image, embedding prediction to locate key points with associated class information, and box refinement to finalize bounding boxes along with class probabilities. CornerNet-Lite addresses the computational limitations of CornerNet while enhancing accuracy. It introduces two key innovations: CornerNet Saccade, which reduces unnecessary computations through an attention mechanism focused on key points, and CornerNet Squeeze, which ensures efficient feature extraction compatible with the backbone architecture. These advancements make CornerNet-Lite suitable for real-time applications.

5.9 Datasets

Datasets are crucial in testing and training object detection models, enabling researchers to create more accurate and adaptable algorithms. These datasets feature detailed annotations, such as segmentation masks and bounding boxes, which enable precise object localization and classification. They encompass a variety of object classes and contexts, making them suitable for numerous computer vision applications. The availability of such datasets has significantly advanced object detection, resulting in the development of more sophisticated and reliable detection models.

5.9.1 General purpose datasets

The COCO dataset comprises over 200,000 images captured in diverse environments, encompassing various objects and scenarios. Its primary goal is to enhance object recognition and segmentation models by providing a comprehensive benchmark and encouraging algorithms to manage diverse categories and complex scenarios. COCO includes detailed annotations for object instances, segmentation masks, and key points. These annotations are invaluable for training models to identify and differentiate objects, particularly in situations involving partial occlusion or intricate shapes. Furthermore, these annotations are especially beneficial for developing models that perform reliably in real-world scenarios.

The Pascal VOC dataset, available in two editions (2007 and 2012), consists of over 20,000 images spanning various object categories and backgrounds. Pascal VOC has been a cornerstone in the development and evaluation of object detection models, serving as a benchmark for early detection methods. Its annotated bounding boxes support tasks such as object recognition and localization. These annotations are useful for assessing the performance of detection models in realistic settings where objects may be partially obscured or exhibit complex geometries.

ImageNet, one of the largest image datasets, contains over 14 million meticulously labeled images across numerous object categories. It serves as the foundation for training and evaluating large-scale object detection and recognition algorithms. ImageNet has played a crucial role in advancing deep learning in computer vision, providing extensive resources for the development of innovative algorithms. Its detailed annotations allow models to learn and identify a wide range of objects with high accuracy, significantly enhancing their performance and robustness across numerous applications. Table 8 summarizes popular object detection datasets, detailing their training, validation, and testing statistics, including the number of images and objects in each dataset.

5.9.2 Domain-specific dataset

Domain-specific datasets, which provide specialized data tailored to particular application domains, are vital for advancing object detection research. These curated datasets enable practitioners and researchers to develop highly accurate and effective object detection models by addressing the unique requirements and challenges of specific industries or scenarios. They include targeted annotations, diverse object classes, and relevant contextual information, making them essential for training and evaluating object detection algorithms in real-world environments. Domains such as autonomous driving, medical imaging, retail, and agriculture benefit significantly from this customized approach, which enhances model performance within specific domains and fosters innovation in specialized object detection research.

The KITTI Vision Benchmark Suite is a prominent dataset for autonomous driving. It offers annotations for objects such as cars, pedestrians, and bicycles alongside images and LiDAR data from diverse scenarios. Similarly, BDD100K is another extensive dataset for autonomous driving, featuring detailed object labels and a wide range of driving conditions. NuScenes, designed for urban scene understanding, provides large-scale object annotations across complex urban landscapes.

In medical imaging, specialized datasets address tasks such as organ segmentation and tumor detection using modalities such as computed tomography (CT) scans and X-rays. Notable examples include datasets developed under the Medical Image Computing and Computer-Assisted Intervention Society (MICCAI). These datasets have significantly contributed to research and development in medical imaging analysis, leading to advancements in diagnostic accuracy and improved treatment planning.

Table 9 provides a comparative analysis of various object detection models evaluated on the Microsoft COCO dataset,

TABLE 7 Hybrid approaches of object detection algorithms summary.

Algorithm	Researchers (Year)	Region proposal	Notes	Softmax incorporated	Loss function	References
Cascade R-CNN	Zhaowei Cai, Tsung-Yi Lin, Wei Wei, Songtao Xu (2018)	Two-stage: uses Faster R-CNN for region proposals in the first stage. Subsequent stages refine proposals based on previous predictions.	True hybrid, combining one-stage (initial proposals) and two-stage (refinement).	Yes	Multi-stage learning with SGD: Smooth L1 loss for bounding box regression, softmax loss for classification	Cai and Nuno, 2018
Mask R-CNN with Cascaded Head	Kailin He, Georgia Gkioxari, Piotr Dollar, Ross Girshick (2017)	Two-stage: Uses standard Mask R-CNN for region proposals (no cascade for proposals). Cascaded structure applies only for the mask prediction branch.	Not necessarily a hybrid in the context of region proposals.	Yes	Multi-stage learning with SGD: Smooth L1 loss for bounding box regression, softmax loss for classification, binary cross-entropy loss for mask prediction	He et al., 2017
Libra R-CNN	Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, Dahua Lin (2019)	Hybrid: Chooses between YOLOv2 (one-stage) and Faster R-CNN (two-stage) proposals based on confidence scores.	True hybrid, dynamically using both types of proposals.	Yes	Multi-task learning with SGD: Focal loss for classification, Smooth L1 loss for bounding box regression	Pang et al., 2019
RegNet	Jing Xu, Yu Pan, Xinglin Pan, Steven Hoi (2021)	One-stage: No explicit region proposals. Uses anchor boxes for candidate object locations.	Not a hybrid in terms of region proposals.	Yes	Multi-task learning with SGD: Focal loss for classification, smooth L1 loss for bounding box regression	Xu et al., 2021

TABLE 8 Popular object detection datasets and their statistics (Zou et al., 2023).

Dataset	Train		Validation		TrainVal		Test	
	Image	Objects	Image	Objects	Image	Objects	Image	Objects
VOC-2007	2,501	6,301	2,510	6,307	5,011	12,608	4,952	14,976
VOC-2007	5,717	13,609	5,823	13,841	11,540	27,450	10,991	-
ILSVRC-2014	456,567	478,807	20,121	55,502	476,688	534,309	40,152	-
ILSVRC-2017	456,567	478,807	20,121	55,502	476,688	534,309	65,500	-
MS-COCO-2015	82,783	604,907	40,504	291,875	123,287	896,782	81,434	-
MS-COCO-2017	118,287	860,001	5,000	36,781	123,287	896,782	40,670	-
Objects 365-2019	600,000	9,623,000	38,000	479,000	628,000	10,102,000	100,000	170,000
OID-2020	1,743,042	14,610,229	41,620	303,980	1,784,662	14,914,209	125,436	937,327

detailing key performance metrics such as AP and AP at different intersections over union thresholds. The results highlight advancements in object detection technologies, demonstrating that newer models outperform their predecessors across multiple performance metrics.

6 Salient object detection

Salient object detection, also called visual saliency detection, is a domain of computer vision dedicated to identifying the most significant or visually distinctive regions in an image. These regions often correspond to areas that naturally capture human attention, similar to how our eyes instinctively focus on specific elements within a scene. By leveraging advanced algorithms, salient object detection identifies these visually unique areas, facilitating applications such as image understanding, object recognition, and scene analysis. This capability enhances tasks such as content-based image editing, image retrieval, segmentation, and cropping, making it a vital area of computer vision research and application development.

TABLE 9 Comparison of Object Detection Models in Microsoft COCO (Zhao et al., 2019).

Models	Backbone architecture	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Fast R-CNN (Girshick, 2015)	ResNet	20.5	39.9	19.4	4.1	20.0	35.8
ION (Bell et al., 2016)	-	23.6	43.2	23.6	6.4	24.1	38.2
OHEM+FRCN (Shrivastava et al., 2016)	VGG16	22.6	42.5	22.2	5.0	23.7	34.6
Faster R-CNN (Ren et al., 2015)	ResNet	24.2	45.3	23.5	7.7	26.4	37.1
YOLOv2 (Redmon and Farhadi, 2017)	Darknet	21.6	44.0	19.2	5.0	22.4	35.5
SSD300 (Liu et al., 2016)	VGG16	23.2	41.2	23.4	5.3	23.2	39.6
SSD512 (Liu et al., 2016)	ResNet-50	26.8	46.5	27.8	9.0	28.9	41.9
R-FCN (Dai et al., 2016)	ResNet101	29.2	51.5	-	10.8	32.8	45.0
R-FCN (multi-scale training) (Dai et al., 2016)	ResNet101	29.9	51.9	-	10.4	32.4	43.3
FPN (Lin et al., 2017a)	ResNet101	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN (He et al., 2017)	ResNet101+FPN	38.2	60.3	41.7	20.1	41.1	50.2
Mask R-CNN (He et al., 2017)	ResNeXt101+FPN	39.8	62.3	43.4	22.1	43.2	51.2
DSSD513 (Fu et al., 2017)	ResNet101	33.2	53.3	35.2	13.0	35.4	51.1

Identifying salient objects can be likened to solving a complex mystery. The bottom-up (BU) approach (Tu et al., 2016) acts like a meticulous investigator, analyzing local features such as edges and spatial information. However, its limited perspective often results in low-contrast and blurry "saliency maps," resembling vague shadows rather than well-defined objects. Conversely, the top-down (TD) approach (Yang and Yang, 2016) functions as a strategic analyst, utilizing prior knowledge about object types to refine the saliency map and emphasize the object's key features. For instance, in semantic segmentation tasks, where individual pixels are classified, the TD approach enhances the clarity and accuracy of BU-detected details, ensuring that the proper structure and boundaries of the object are effectively captured (Gao et al., 2009).

6.1 Deep learning for salient object detection

CNNs are pivotal in high-level and multi-scale feature representation within salient object detection. These architectures have proven effective in various computer vision tasks, including edge detection, object recognition, and semantic segmentation. Eleonora (Vig et al., 2014) pioneered a data-driven approach, leveraging deep networks with diverse layers and parameters to maximize feature extraction. Similarly, Kümmerer et al. (2014) introduced Deep Gaze, which utilized AlexNet to create a high-dimensional feature space for saliency mapping, addressing challenges posed by limited training data. Extending this idea, Huang et al. (2015) fine-tuned pre-trained object recognition deep networks using saliency evaluation metrics such as Similarity and KL-Divergence. Numerous strategies have since been developed to enhance the integration of local and global visual cues for salient object detection. For instance, Wang et al. employed two separate deep CNNs to capture both local and global features, while Cholakkal et al. (2018) proposed a weakly supervised system that fuses top-down and bottom-up saliency maps, refining them through multi-scale superpixel averaging. Additionally, Zhao et al. (2015) designed a multi-context deep learning framework using superpixel segmentation to combine local and global contextual modeling.

Efforts to incorporate context modeling and semantic information into salient object detection have also shown promising results. For example, Li et al. (2016) proposed a multi-task deep saliency model that creates intrinsic connections between saliency detection and semantic segmentation. In contrast, He et al. (2015b) introduced SuperCNN, a superpixel-based CNN aimed at enhancing performance.

The integration of multi-scale feature maps has proven crucial for improving detection accuracy. Liu et al. (2015) utilized CNNs for fixation prediction by jointly learning visual saliency components, while Wang et al. (2015) introduced RegionNet, which preserves edges and incorporates multi-scale contextual modeling for salient object detection. The evolution of deep learning techniques in salient object detection demonstrates a continuous trajectory toward more accurate and efficient solutions, solidifying its importance in computer vision research (Gao et al., 2009).

6.2 Benchmark datasets and evaluation metrics

ECSSD (Yan et al., 2013), HKU-IS (Li and Yu, 2016b), PASCALS (Li et al., 2014), and SOD (Movahedi and Elder, 2010) are widely recognized benchmark datasets for evaluating the performance of salient object detection methods. ECSSD contains over 4,000 challenging images characterized by low contrast and

TABLE 10 Comparison between the state-of-art methods in salient object detection.

Dataset	Metrics	CHM (Li et al., 2013)	RC (Cheng et al., 2014)	DRFI (Jiang et al., 2013)	MC (Zhao et al., 2015)	MDF (Li and Yu, 2016b)	DSR (Tang et al., 2016)	DCL (Li and Yu, 2016a)	ELD (Lee et al., 2016)	NLDF (Luo et al., 2017)	DSSC (Hou et al., 2017)
PASCALS	MAE	0.222	0.225	0.221	0.147	0.145	0.128	0.108	0.121	0.099	0.080
ECSSD	MAE	0.195	0.187	0.166	0.107	**0.108**	**0.037**	0.071	0.098	0.063	0.052
HKU-IS	MAE	**0.058**	**0.165**	**0.143**	**0.098**	0.129	0.040	**0.048**	**0.071**	**0.048**	**0.039**
SOD	MAE	0.249	0.242	0.215	0.184	0.155	-	0.126	0.154	0.143	0.118

The bold values (lowest score) indicate the best performance.

multiple salient objects, while HKU-IS comprises over 1,000 semantically rich and complex natural images. PASCALS originates from the validation set of the PASCAL VOC 2010 segmentation dataset, consisting of 850 natural images. In comparison, the SOD dataset includes 300 images, each featuring multiple salient objects. Adhering to the standardized training and validation splits proposed by Jiang et al. (2013) ensures a rigorous and consistent evaluation of methodologies.

Saliency map evaluation primarily relies on two metrics: Mean Absolute Error (MAE) and F-measure. The F-measure quantifies saliency map quality through precision and recall, computed based on the intersection of the generated binary mask B with a ground truth Z. These datasets collectively cover diverse image attributes and complexities, enabling a comprehensive assessment of salient object detection techniques.

$$F_\beta = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}} \tag{8}$$

where β^2 is set to 0.3 to emphasize how crucial the precision value is. Using the following formula, the MAE score is calculated.

The Mean Absolute Error (MAE) is calculated using the following equation:

$$\text{MAE} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left| \hat{S}(i,j) - \hat{Z}(i,j) \right| \tag{9}$$

It represents the average absolute difference between the predicted values $\widehat{S}$ denote the value at position (i,j) in matrix S, and $\widehat{Z}$ represents the ground truth value at the corresponding position in the matrices. The Mean Absolute Error is calculated by taking the absolute difference between each corresponding pair of values in the matrices, summing up these absolute differences, and then dividing by the total number of elements (H×W) in the matrices, where the W and H are the width and height of the salient area.

This research looks at some salient feature object detection methods, such as deep learning-focused and classical methods. Notable for their exceptional performance are the Context-aware Hierarchical Model (CHM) (Li et al., 2013), Region Contrast (RC) (Cheng et al., 2014), and Discriminative Region Feature Integration (DRFI) (Jiang et al., 2013). CNN is the foundation of other methods, including Multi-Contextual (MC) (Zhao et al., 2015), Multi-level Deep Feature Integration (MDF) (Li and Yu, 2016b), Deep contrast learning (DCL) (Li and Yu, 2016a) Edge-Loss with Diverse-thresholding (ELD) (Lee et al., 2016), Non-Local Deep Features (NLDF) (Luo et al., 2017), and Deep Scale Selection and Classification (DSSC) (Hou et al., 2017).

In general, CNN-based techniques outperform classical approaches; the Table 10 shows the evaluation metrics, F-measure, and Mean Absolute Error (MAE). In particular, MC and MDF make better saliency forecasts by utilizing data from both local and global settings. ELD provides additional information by taking advantage of low-level custom features. LEGS employs generic region recommendations for the first salient regions, which might not be sufficient. Future directions for improvement are suggested by integrating semantic segmentation and recurrent networks in DSR and MT. Multi-scale representations and superpixel segmentation are necessary for DCL, NLDF, and DSSC to produce highly salient regions and smooth boundaries. Among these, DCL, NLDF, and DSSC show the best performance on all four datasets; scale-to-scale short connection modeling allows DSSC to show the best performance.

Most CNN-based techniques require using superpixel segmentation to simulate visual saliency along area boundaries because CNN primarily provides salient information in small regions. Measuring local conspicuity requires extracting multi-scale deep CNN features. Strengthening local connections between several CNN layers and incorporating complementing data from local and global contexts is considered vital.

7 Challenges and future opportunities

The use of CNN examines potential advancements in object detection. This study highlights the importance of enhancing object identification methods to strike a balance between speed and accuracy. Two-stage and hybrid detection systems provide greater precision at the cost of increased computational complexity, while one-stage alternatives prioritize quicker data processing with a certain level of accuracy. Future research will creatively address this trade-off by developing systems that are both precise and efficient.

However, there are several challenges in object detection, such as occlusion, where items may be hidden by other objects, leading to inaccurate detection. Additionally, the less noticeable characteristics of small or distant objects complicate recognition. Moreover, object detection algorithms encounter difficulties in situations with overlapping objects and issues related

to illumination and viewpoint. Furthermore, the limited availability of data and the lack of annotated data hinder effective model training. Additionally, the incorporation of multi-modal detection affects the overall performance of object identification systems. However, the resolution and image processing techniques impose restrictions on this integration.

It is critical to ensure the stability and dependability of object-detecting systems in various situations. Models can be made more general using domain adaptation and transfer learning strategies, which will help them function well in novel environments. Combining many modalities of information, including textual or temporal signals, might increase the accuracy of complicated scene interpretation and improve contextual understanding. To balance this trade-off and create object identification algorithms that are both extremely precise and computationally economical, researchers frequently employ multi-task loss functions to penalize misclassifications and localization errors.

8 Conclusion

The rapid evolution of object detection algorithms marks a transformative era in image and pattern recognition, enabling groundbreaking advancements in visual perception and interaction. This study has comprehensively reviewed object detection methodologies, ranging from single-stage to two-stage and hybrid approaches. While single-stage methods excel in speed and computational efficiency, two-stage and hybrid approaches demonstrate superior accuracy and detection precision, making them highly suitable for real-world applications. By analyzing architectural frameworks, backbone structures, and loss functions, this study emphasizes the critical importance of iterative improvement to address the growing demands of modern technological applications. The developments discussed pave the way for revolutionary advancements in domains such as autonomous vehicles, surveillance systems, and broader image recognition tasks, fundamentally reshaping how humans interact with the visual environment.

Future research must prioritize the integration of multimodal data by combining textual, contextual, and visual signals to improve robustness and contextual sensitivity in object detection models. This multidisciplinary approach holds promise for innovative applications in multimedia analysis, augmented reality, and human-computer interaction. Furthermore, scalable and parallelizable object detection systems are essential for meeting the growing demand for real-time processing of large image and video datasets. Advances in distributed computing, edge computing, and hardware acceleration will be crucial for deploying these systems in resource-constrained environments.

Equally important is the need to address object detection technologies' social and ethical implications. Privacy, bias, and fairness concerns must be rigorously examined to ensure responsible and equitable deployment across diverse societal contexts. Future research should strive to develop frameworks and policies that safeguard these principles, fostering the ethical adoption of object detection systems. By aligning technological innovation with ethical accountability, the field can ensure its advancements serve humanity responsibly while unlocking unprecedented opportunities for creative and practical applications.

Author contributions

BL: Conceptualization, Investigation, Methodology, Resources, Writing – original draft, Writing – review & editing. GS: Formal analysis, Investigation, Project administration, Supervision, Writing – original draft. TH: Conceptualization, Methodology, Project administration, Supervision, Validation, Funding acquisition, Writing – original draft.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. The research was financially supported by Center of Excellence in Digital Earth and Emerging Technology (CoE:DEET), Thammasat University, Thailand.

Acknowledgments

BL acknowledges the Faculty-Quota Scholarship awarded by SIIT, Thammasat University. The authors would like to thank the Advance Geospatial Technology Research Unit, SIIT, and the Center of Excellence in Digital Earth and Emerging Technology (CoE: DEET), Thammasat University, for providing a technical environment and supporting information. We gratefully acknowledge the financial support provided by the Thammasat University Research Fund. The authors acknowledge the assistance of ChatGPT developed by OpenAI, for helping to refine the content and generate some figures for this work.

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

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2025.1437664/full#supplementary-material

References

Aggarwal, A., and Kumar, M. (2021). Image surface texture analysis and classification using deep learning. *Multimed. Tools Appl.* 80, 1289–1309. doi: 10.1007/s11042-020-09520-2

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., et al. (2021). Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *J. Big Data* 8, 1–74. doi: 10.1186/s40537-021-00444-8

Amjoud, A. B., and Amrouch, M. (2023). Object detection using deep learning, cnns and vision transformers: a review. *IEEE Access* 11, 35479–35516. doi: 10.1109/ACCESS.2023.3266093

Ardia, D., Ringel, E., Ekstrand, V. S., and Fox, A. (2020). "Addressing the decline of local news, rise of platforms, and spread of mis-and disinformation online," in *UNC Center for Media Law and Policy*.

Awad, M., Khanna, R., Awad, M., and Khanna, R. (2015). "Support vector machines for classification," in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, 39–66.

Bai, Y. (2022). "Relu-function and derived function review," in *SHS Web of Conferences* (Les Ulis: EDP Sciences), 02006.

Bell, S., Zitnick, C. L., Bala, K., and Girshick, R. (2016). "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 2874–2883.

Ben Braiek, H., and Khomh, F. (2023). Testing feedforward neural networks training programs. *ACM Trans. Softw. Eng. Methodol.* 32, 1–61. doi: 10.1145/3529318

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv* preprint arXiv:2004.10934. doi: 10.48550/arXiv.2004.10934

Buckner, C. (2019). Deep learning: a philosophical introduction. *Philosophy Comp.* 14e12625. doi: 10.1111/phc3.12625

Cai, Z., and Nuno, N. (2018). "Cascade R-CNN: delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 6154–6162. doi: 10.1109/CVPR.2018.00644

Cao, J., Cholakkal, H., Anwer, R. M., Khan, F. S., Pang, Y., and Shao, L. (2020). "D2det: Towards high quality object detection and instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 11485–11494.

Chen, G., Choi, W., Yu, X., Han, T., and Chandraker, M. (2017). "Learning efficient object detection models with knowledge distillation," in *Advances in Neural Information Processing Systems*, 30.

Cheng, M.-M., Mitra, N. J., Huang, X., Torr, P. H., and Hu, S.-M. (2014). Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 569–582. doi: 10.1109/TPAMI.2014.2345401

Chhabra, M., Ravulakollu, K. K., Kumar, M., Sharma, A., and Nayyar, A. (2023). Improving automated latent fingerprint detection and segmentation using deep convolutional neural network. *Neural Comp. Appl.* 35, 6471–6497. doi: 10.1007/s00521-022-07894-y

Chhabra, M., Sharan, B., Elbarachi, M., and Kumar, M. (2024). "Intelligent waste classification approach based on improved multi-layered convolutional neural network," in *Multimedia Tools and Applications, Vol. 83*, 84095–84120. doi: 10.1007/s11042-024-18939-w

Choi, H., Ryu, S., and Kim, H. (2018). "Short-term load forecasting based on resnet and lstm," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)* (Aalborg: IEEE), 1–6.

Cholakkal, H., Johnson, J., and Rajan, D. (2018). Backtracking spatial pyramid pooling-based image classifier for weakly supervised top-down salient object detection. *IEEE Trans. Image Proc.* 27, 6064–6078. doi: 10.1109/TIP.2018.2864891

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (ELUS). *arXiv* preprint arXiv:1511.07289. doi: 10.48550/arXiv.1511.07289

Cristianini, N., and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge: Cambridge University Press.

Dai, J., Li, Y., He, K., and Sun, J. (2016). "R-FCN: object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 29.

Dalal, N., and Triggs, B. (2005). "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (San Diego, CA: IEEE), 886–893.

Dong, X., Zheng, L., Ma, F., Yang, Y., and Meng, D. (2018). Few-example object detection with model communication. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 1641–1654. doi: 10.1109/TPAMI.2018.2844853

Dong, Z., Li, G., Liao, Y., Wang, F., Ren, P., and Qian, C. (2020). "Centripetalnet: pursuing high-quality keypoint pairs for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10519–10528. doi: 10.1109/CVPR42600.2020.01053

Dundar, A., Jin, J., Martini, B., and Culurciello, E. (2016). Embedded streaming deep neural networks accelerator with applications. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 1572–1583. doi: 10.1109/TNNLS.2016.2545298

Elgendy, M. (2020). *Deep Learning for Vision Systems*. New York: Simon and Schuster.

Erhan, D., Szegedy, C., Toshev, A., and Anguelov, D. (2014). "Scalable object detection using deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 2147–2154.

Freeman, W. T., and Roth, M. (1995). "Orientation histograms for hand gesture recognition," in *International Workshop on Automatic Face and Gesture Recognition* (Princeton, NJ: Citeseer), 296–301.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *J.-Jap. Soc. Artif. Intellig.* 14:1612.

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., and Berg, A. C. (2017). DSSD: Deconvolutional single shot detector. *arXiv* preprint arXiv:1701.06659. doi: 10.48550/arXiv.1701.06659

Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biol. Cybern.* 20, 121–136. doi: 10.1007/BF00342633

Gao, D., Han, S., and Vasconcelos, N. (2009). Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 989–1005. doi: 10.1109/TPAMI.2009.27

Ghiasi, G., Lin, T.-Y., and Le, Q. V. (2019). "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (Long Beach, CA: IEEE), 7036–7045. Available online at: https://arxiv.org/abs/1904.07392

Girshick, R. (2015). "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision* (Santiago: IEEE), 1440–1448.

Girshick, R., Donahue, J., Darrell, T., Berkeley, U., and Malik, J. (2014a). "R-CNN: region-based convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and pattern Recognition (CVPR)*, 2–9.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014b). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 580–587. Available online at: https://arxiv.org/abs/1311.2524

Grimaldi, M., Tenace, V., and Calimera, A. (2018). Layer-wise compressive training for convolutional neural networks. *Future Internet* 11(1):7. doi: 10.3390/fi11010007

Guan, L. (2017). *Multimedia Image and Video Processing*. Boca Raton, FL: CRC Press.

Guo, C., Fan, B., Gu, J., Zhang, Q., Xiang, S., Prinet, V., et al. (2019). "Progressive sparse local attention for video object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 3909–3918.

Hammoudeh, M. A. A., Alsaykhan, M., Alsalameh, R., and Althwaibi, N. (2022). Computer vision: a review of detecting objects in videos-challenges and techniques. *Int. J. Online Biomed. Eng.* 18:27577. doi: 10.3991/ijoe.v18i01.27577

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 2961–2969. doi: 10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., and Sun, J. (2015a). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 1904–1916. doi: 10.1109/TPAMI.2015.2389824

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

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). "Identity mappings in deep residual networks," in *Computer Vision-ECCV 2016: 14th European Conference* (Amsterdam: Springer), 630–645.

He, S., Lau, R. W., Liu, W., Huang, Z., and Yang, Q. (2015b). Supercnn: a superpixelwise convolutional neural network for salient object detection. *Int. J. Comput. Vis.* 115, 330–344. doi: 10.1007/s11263-015-0822-0

Hereman, W., and Malfliet, W. (2005). "The tanh method: a tool to solve nonlinear partial differential equations with symbolic software," in *Proceedings 9th World Multi-Conference on Systemics, Cybernetics and Informatics* (Orlando, FL: IEEE), 165–168.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hou, Q., Cheng, M.-M., Hu, X., Borji, A., Tu, Z., and Torr, P. H. (2017). "Deeply supervised salient object detection with short connections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 3203–3212.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 4700–4708.

Huang, X., Shen, C., Boix, X., and Zhao, Q. (2015). Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. *In Proceedings of the IEEE international conference on computer vision, pages* 262–270. doi: 10.1109/ICCV.2015.38

Huang, Z., Wang, J., Fu, X., Yu, T., Guo, Y., and Wang, R. (2020). Dc-spp-yolo: dense connection and spatial pyramid pooling based yolo for object detection. *Inf. Sci.* 522, 241–258. doi: 10.1016/j.ins.2020.02.067

Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., and Li, S. (2013). "Salient object detection: a discriminative regional feature integration approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 9 Portland, OR: IEEE), 2083–2090.

Jocher, G., and Qiu, J. (2024). *Ultralytics Yolo11*. Available online at: https://github.com/ultralytics/yolov5

Jocher, G., Stoken, A., Borovec, J., Chaurasia, A., Changyu, L., Hogan, A., et al. (2021). *Ultralytics/yolov5: v5. 0-yolov5-p6 1280 Models, AWS, Supervise. LY and Youtube Integrations*. Geneva: Zenodo.

Kaur, R., and Singh, S. (2023). A comprehensive review of object detection with deep learning. *Digit. Signal Process.* 132:103812. doi: 10.1016/j.dsp.2022.103812

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 25. Available online at: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Kümmerer, M., Theis, L., and Bethge, M. (2014). Deep gaze I: Boosting saliency prediction with feature maps trained on imagenet. *arXiv* preprint arXiv:1411.1045. doi: 10.48550/arXiv.1411.1045

Law, H., and Deng, J. (2018). "Cornernet: detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Cham: Springer), 734–750.

Law, H., Teng, Y., Russakovsky, O., and Deng, J. (2019). Cornernet-lite: efficient keypoint based object detection. *arXiv* preprint arXiv:1904.08900. doi: 10.48550/arXiv.1904.08900

Lazebnik, S., Schmid, C., and Ponce, J. (2006). "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (New York, NY: IEEE), 2169–2178.

Lee, G., Tai, Y.-W., and Kim, J. (2016). "Deep saliency with encoded low level distance map and high level features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 660–668.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., et al. (2022). Yolov6: A single-stage object detection framework for industrial applications. *arXiv* preprint arXiv:2209.02976. doi: 10.48550/arXiv.2209.02976

Li, G., and Yu, Y. (2016a). "Deep contrast learning for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV,: IEEE), 478–487.

Li, G., and Yu, Y. (2016b). Visual saliency detection based on multiscale deep cnn features. *IEEE trans. Image Proc.* 25, 5012–5024. doi: 10.1109/TIP.2016.2602079

Li, Q., Niaz, U., and Merialdo, B. (2012). "An improved algorithm on viola-jones object detector," in *2012 10th International Workshop on Content-Based Multimedia Indexing (CBMI)* (Annecy: IEEE), 1–6.

Li, X., Li, Y., Shen, C., Dick, A., and Van Den Hengel, A. (2013). "Contextual hypergraph modeling for salient object detection," in *Proceedings of the IEEE International Conference on Computer Vision* (Sydney: IEEE), 3328–3335.

Li, X., Zhao, L., Wei, L., Yang, M.-H., Wu, F., Zhuang, Y., et al. (2016). Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Trans. Image Proc.* 25, 3919–3930. doi: 10.1109/TIP.2016.2579306

Li, Y., Chen, Y., Wang, N., and Zhang, Z. (2019). "Scale-aware trident networks for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 6054–6063.

Li, Y., Hou, X., Koch, C., Rehg, J. M., and Yuille, A. L. (2014). "The secrets of salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE) 280–287.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE) 2117–2125. Available online at: https://arxiv.org/abs/1612.03144

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 2980–2988. Available online at: https://arxiv.org/abs/1708.02002

Liu, N., Han, J., Zhang, D., Wen, S., and Liu, T. (2015). "Predicting eye fixations using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston: IEEE), 362–370.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "SSD: Single shot multibox detector," in *Computer Vision-ECCV 2016: 14th European Conference* (Amsterdam: Springer), 21–37.

Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., et al. (2020). Pp-yolo: An effective and efficient implementation of object detector. *arXiv* preprint arXiv:2007.12099. doi: 10.48550/arXiv.2007.12099

Lowe, D. G. (1999). "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision* (Kerkyra: IEEE), 1150–1157.

Luo, Z., Mishra, A., Achkar, A., Eichel, J., Li, S., and Jodoin, P.-M. (2017). "Non-local deep features for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 6609–6617.

Movahedi, V., and Elder, J. H. (2010). "Design and perceptual validation of performance measures for salient object segmentation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops* (San Francisco, CA: IEEE), 49–56.

Najibi, M., Rastegari, M., and Davis, L. S. (2016). "G-CNN: an iterative grid based object detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 2369–2377.

Nash, W., Drummond, T., and Birbilis, N. (2018). A review of deep learning in the study of materials degradation. *NPJ Mater Degrad.* 2:37. doi: 10.1038/s41529-018-0058-x

Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv* preprint arXiv:1811.03378. doi: 10.48550/arXiv.1811.03378

Ouyang, W., Wang, X., Zeng, X., Qiu, S., Luo, P., Tian, Y., et al. (2015). "Deepid-net: Deformable deep convolutional neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA: IEEE), 2403–2412.

Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., and Lin, D. (2019). "Libra R-CNN: towards balanced learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 821–830. Available online at: https://arxiv.org/abs/1904.02701

Pinheiro, P. O., Lin, T.-Y., Collobert, R., and Dollár, P. (2016). "Learning to refine object segments," in *Computer Vision-ECCV 2016: 14th European Conference* (Amsterdam: Springer), 75–91.

Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv* [preprint] arXiv:1710.05941. doi: 10.48550/arXiv.1710.05941

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition* (Las Vegas, NV: IEEE), 779–788.

Redmon, J., and Farhadi, A. (2017). "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 7263–7271. doi: 10.1109/CVPR.2017.690

Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv* [preprint] arXiv:1804.02767. doi: 10.48550/arXiv.1804.02767

Reis, D., Kupec, J., Hong, J., and Daoudi, A. (2023). Real-time flying object detection with yolov8. *arXiv* [preprint] arXiv:2305.09972. doi: 10.48550/arXiv.2305.09972

Ren, S., He, K., Girshick, R., and Sun, J. (2015). "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 28.

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. arXiv [preprint] arXiv:1506.01497. doi: 10.1109/TPAMI.2016.2577031

Ren, Y., Zhu, C., and Xiao, S. (2018). Small object detection in optical remote sensing images via modified faster r-cnn. *Appl. Sci.* 8:813. doi: 10.3390/app8050813

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.

Shah, S., and Tembhurne, J. (2023). Object detection using convolutional neural networks and transformer-based models: a review. *J. Elect. Syst. Inform. Technol.* 10:54. doi: 10.1186/s43067-023-00123-z

Shrivastava, A., Gupta, A., and Girshick, R. (2016). "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 761–769. doi: 10.1109/CVPR.2016.89

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

Sun, Z. (2024). *Parallelization of Hybrid Multi-Objective Evolutionary Algorithm on Multi-Core Architectures*. Available online at: https://mspace.lib.umanitoba.ca/items/01bac995-c4a2-4f19-9ac7-a13e5fd1bbfc

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 2818–2826.

Tang, G., Ni, J., Zhao, Y., Gu, Y., and Cao, W. (2023). A survey of object detection for uavs based on deep learning. *Remote Sens*. 16:149. doi: 10.3390/rs16010149

Tang, Y., Wu, X., and Bu, W. (2016). "Deeply-supervised recurrent convolutional neural network for saliency detection," in *MM '16: Proceedings of the 24th ACM international conference on Multimedia* (New York, NY: IEEE), 397–401. doi: 10.1145/2964284.2967250

Tu, W.-C., He, S., Yang, Q., and Chien, S.-Y. (2016). "Real-time salient object detection with a minimum spanning tree," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 2334–2342.

Van de Sande, K. E., Uijlings, J. R., Gevers, T., and Smeulders, A. W. (2011). "Segmentation as selective search for object recognition," in *2011 International Conference on Computer Vision* (Barcelona: IEEE), 1879–1886.

Vig, E., Dorr, M., and Cox, D. (2014). "Large-scale optimization of hierarchical features for saliency prediction in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH: IEEE), 2798–2805.

Viola, P., and Jones, M. (2001). "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (Kauai, HI: IEEE).

Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., et al. (2024). Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv*:2405.14458. doi: 10.48550/arXiv.2405.14458

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023). "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Vancouver, BC: IEEE), 7464–7475.

Wang, C.-Y., and Liao, H.-Y. M. (2024). *Yolov9: Learning What You Want to Learn Using Programmable Gradient Information*. Available online at: https://arxiv.org/abs/2402.13616

Wang, L., Lu, H., Ruan, X., and Yang, M.-H. (2015). "Deep networks for saliency detection via local estimation and global search," in *Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition* (Boston, MA: IEEE), 3183–3192.

Womg, A., Shafiee, M. J., Li, F., and Chwyl, B. (2018). "Tiny SSD: a tiny single-shot detection deep convolutional neural network for real-time embedded object detection," in *2018 15th Conference on computer and robot vision (CRV)* (Toronto, ON: IEEE), 95–101.

Wu, B., Iandola, F., Jin, P. H., and Keutzer, K. (2017). "Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (Honolulu, HI: IEEE), 129–137. Available online at: https://arxiv.org/pdf/1612.01051

Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., et al. (2020). "Polarmask: Single shot instance segmentation with polar representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 12193–12202.

Xiong, Y., Liu, H., Gupta, S., Akin, B., Bender, G., Wang, Y., et al. (2021). "Mobiledets: Searching for object detection architectures for mobile accelerators," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Nashville, TN: IEEE) 3825–3834. doi: 10.1109/CVPR46437.2021.00382

Xu, J., Pan, Y., Pan, X., Hoi, S., Yi, Z., and Xu, Z. (2021). RegNet: self-regulated network for image classification. *arXiv [Preprint]*. arXiv: 2101.00590. doi: 10.48550/arXiv.2101.00590

Yan, Q., Xu, L., Shi, J., and Jia, J. (2013). "Hierarchical saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Portland, OR: IEEE) 1155–1162. doi: 10.1109/CVPR.2013.153

Yang, J., and Yang, M.-H. (2016). Top-down visual saliency via joint crf and dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 576–588. doi: 10.1109/TPAMI.2016.2547384

Yao, Y., Cheng, G., Wang, G., Li, S., Zhou, P., Xie, X., et al. (2022). On improving bounding box representations for oriented object detection. *IEEE Trans. Geosci. Remote Sens*. 61, 1–11. doi: 10.1109/TGRS.2022.3231340

Yoo, D., Park, S., Lee, J.-Y., Paek, A. S., and So Kweon, I. (2015). "Attentionnet: Aggregating weak directions for accurate object detection," in *Proceedings of the IEEE International Conference on Computer Vision* (Santiago: IEEE), 2659–2667.

Zhang, S., Wen, L., Lei, Z., and Li, S. Z. (2020). Refinedet++: Single-shot refinement neural network for object detection. *IEEE Trans. Circuits Syst. Video Technol*. 31, 674–687. doi: 10.1109/TCSVT.2020.2986402

Zhang, W., Jiao, L., Liu, X., and Liu, J. (2019). "Multi-scale feature fusion network for object detection in vhr optical remote sensing images," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium* (Yokohama: IEEE), 330–333.

Zhang, Y., Sohn, K., Villegas, R., Pan, G., and Lee, H. (2015). "Improving object detection with deep convolutional networks via bayesian optimization and structured prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA: IEEE), 249–258.

Zhao, R., Ouyang, W., Li, H., and Wang, X. (2015). "Saliency detection by multi-context deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA: IEEE), 1265–1274.

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., and Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artif. Intellig. Rev*. 57:99. doi: 10.1007/s10462-024-10721-6

Zhao, Z.-Q., Zheng, P., Xu, S., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst*. 30, 3212–3232. doi: 10.1109/TNNLS.2018.2876865

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2021). Deformable DETR: deformable transformers for end-to-end object detection. *arXiv [Preprint]*. arXiv:2010.04159. doi: 10.48550/arXiv.2010.04159

Zhu, Z., Zhou, Y., Dong, Y., and Zhong, Z. (2023). PWLU: Learning specialized activation functions with the piecewise linear unit. *IEEE Trans. Pattern Analy. Mach. Intellig*. 45, 12269–12286. doi: 10.1109/TPAMI.2023.3286109

Zou, Z., Chen, K., Shi, Z., Guo, Y., and Ye, J. (2023). Object detection in 20 years: a survey. *Proc. IEEE* 111, 257–276. doi: 10.1109/JPROC.2023.3238524