Real-Time Object Detection (YOLO vs R-CNN): A Comparative Review for Efficient Vision Systems

Surya Prakash*, Shreya Maddheshiya, Shubham Kumar Jha, Shreya Tripathi, Sparsh Kumar, Tushar Sirohi, Tanish Bhati, Tarun Kumar
Department of Information Technology, Noida Institute of Engineering and Technology, Greater Noida, India
Email: *suryaprakash36016@gmail.com

***Abstract*—Real-time object detection has become a fundamental component of modern computer vision, enabling intelligent systems to perceive and interpret dynamic environments across a wide range of applications. From autonomous transportation and video surveillance to medical imaging, robotics, and smart manufacturing, the demand for accurate and computationally efficient detection models continues to increase. Among deep learning-based approaches, the Region-Based Convolutional Neural Network (R-CNN) family and the You Only Look Once (YOLO) family represent two influential paradigms that have significantly shaped the evolution of object detection research. While R-CNN-based methods emphasize detection accuracy through region proposal mechanisms, YOLO architectures adopt a unified detection strategy that prioritizes real-time performance and deployment efficiency.**

This review presents a comprehensive comparative analysis of YOLO and R-CNN frameworks by examining their architectural developments, detection strategies, computational characteristics, and practical applicability. The study systematically reviews the progression from R-CNN to Fast R-CNN, Faster R-CNN, and Mask R-CNN, alongside the evolution of YOLO from its initial design to recent lightweight and high-performance variants. Key evaluation aspects, including detection accuracy, inference speed, computational complexity, memory requirements, robustness to challenging scenarios, and suitability for edge devices, are critically discussed. Furthermore, the paper highlights representative application domains, identifies existing research gaps, and examines emerging trends such as transformer-assisted detection, model compression, multimodal learning, and edge intelligence. The comparative findings indicate that the choice between YOLO and R-CNN architectures depends largely on application-specific requirements, balancing precision, latency, and resource constraints. This review provides researchers and practitioners with an organized perspective on current advancements and future directions for designing efficient and scalable real-time vision systems.

***Keywords*—Real-time object detection, YOLO, R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN, deep learning, computer vision, efficient vision systems, edge intelligence.**

I Introduction

Object detection is a fundamental task in computer vision that aims to identify and localize objects of interest within digital images and video streams. Unlike image classification, which assigns a single label to an image, object detection simultaneously predicts the category and spatial location of multiple objects, making it indispensable for intelligent perception systems. Recent advances in deep learning have significantly improved detection accuracy and robustness, enabling practical deployment in autonomous driving, intelligent surveillance, industrial inspection, healthcare, robotics, precision agriculture, and smart city infrastructures [1, 2]. The increasing availability of large-scale annotated datasets and high-performance computing platforms has further accelerated the development of efficient detection algorithms capable of operating under real-world constraints.

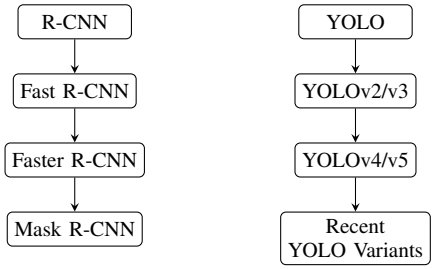

Fig. 1: Evolution of representative deep learning-based object detection frameworks.

The evolution of object detection has progressed from handcrafted feature-based methods to end-to-end deep neural networks. Early approaches relied on feature descriptors and sliding-window strategies, which often suffered from high computational cost and limited generalization. The introduction of the Region-Based Convolutional Neural Network (R-CNN) established a new paradigm by integrating deep convolutional features with region proposal mechanisms for accurate object localization [3]. Subsequent improvements, including Fast R-CNN and Faster R-CNN, reduced computational overhead through shared feature extraction and region proposal networks [4, 5]. The development of Mask R-CNN further extended this framework to instance segmentation tasks [6]. In parallel, the You Only Look Once (YOLO) family proposed a unified detection strategy that treats object detection as a single regression problem, substantially improving inference speed while maintaining competitive accuracy [7]. Continuous architectural refinements have resulted in lightweight and high-performance YOLO variants that support real-time applications on resource-constrained platforms.

Figure 1 summarizes the major milestones in the evolution of deep learning-based object detection. The figure illustrates the parallel advancement of the R-CNN and YOLO families, highlighting the transition from computationally intensive region-based frameworks to unified real-time detection architectures.

Despite remarkable progress, selecting an appropriate detection framework remains a challenging task because practical applications impose different requirements regarding accuracy, latency, computational complexity, memory consumption, and hardware availability. Region-based detectors generally achieve high localization precision but require greater computational resources, whereas single-stage detectors prioritize speed and efficient deployment. The rapid emergence of new architectures has created the need for a systematic comparison that examines their relative strengths, limitations, and application suitability under diverse operating conditions.

Motivated by these challenges, this review provides a structured comparative analysis of the YOLO and R-CNN families for efficient vision systems. The major contributions of this work are summarized as follows:

- A concise review of the architectural evolution of R-CNN and YOLO frameworks.
- A comparative discussion of detection accuracy, inference speed, computational requirements, and deployment characteristics.
- An overview of representative real-world applications and practical implementation considerations.
- Identification of existing research gaps and emerging directions for efficient real-time object detection.

The remainder of this paper is organized as follows. Section II describes the review methodology, followed by a detailed literature analysis of R-CNN and YOLO architectures. Subsequent sections discuss application domains, comparative performance characteristics, research gaps, future research opportunities, and concluding observations.

II Methodology

This review adopts a structured and reproducible methodology to provide a balanced comparison between the YOLO and R-CNN families of object detection frameworks. Rather than focusing on a single implementation, the study systematically examines the architectural evolution, computational characteristics, and application-specific performance of representative models reported in the literature. The adopted methodology follows widely accepted guidelines for evidence-based review studies, ensuring transparency in article selection and consistency in comparative analysis [8, 9]. The overall workflow employed in this study is illustrated in Figure 2.

Initially, a comprehensive literature search was conducted using major scientific databases, including IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and Google Scholar. The search process employed combinations of keywords such as “Object Detection,” “YOLO,” “R-CNN,” “Fast R-CNN,” “Faster R-CNN,” “Mask R-CNN,” “Real-Time Detection,” “Computer Vision,” and “Deep Learning.” Priority was given to highly cited foundational studies and recent contributions published between 2014 and 2025 to capture both historical developments and contemporary advances in efficient vision systems.

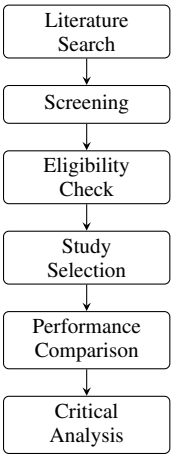

Fig. 2: Methodological framework adopted for the comparative review of YOLO and R-CNN object detection models.

To maintain the quality and relevance of the survey, explicit inclusion and exclusion criteria were adopted. Peer-reviewed journal articles, conference papers, benchmark studies, and influential survey papers directly related to YOLO and R-CNN architectures were included. Studies lacking experimental validation, duplicate publications, non-English articles, and works unrelated to real-time object detection were excluded. This filtering process ensured that the comparative discussion was based on reliable and practically significant research findings.

The selected studies were subsequently analyzed using a common set of evaluation metrics to facilitate objective comparison across different architectures. The primary metrics considered in this review include detection accuracy, measured through mean Average Precision (mAP); inference speed, represented by Frames Per Second (FPS); computational complexity; memory requirements; robustness to small and occluded objects; and suitability for deployment on edge and embedded platforms. These performance indicators are widely accepted in the computer vision community and provide a comprehensive assessment of detection efficiency under practical operating conditions [10, 11].

Figure 2 summarizes the review methodology adopted in this work, beginning with literature identification and ending with critical analysis and research gap identification.

The systematic implementation of these stages enables a consistent evaluation of the strengths and limitations of both detection paradigms while providing a reliable foundation for the subsequent literature review, comparative analysis, identification of research gaps, and discussion of future research opportunities.

III Literature Review

The remarkable progress achieved in deep learning-based object detection can largely be attributed to two major research directions, namely the region-based convolutional neural network family and the unified regression-based YOLO family. The former primarily focuses on improving localization

accuracy through candidate region generation, whereas the latter emphasizes real-time performance by directly predicting object classes and bounding boxes in a single inference stage. This section critically reviews the evolution of these architectures and highlights their major contributions to efficient vision systems.

A. R-CNN Family

The introduction of Region-Based Convolutional Neural Networks (R-CNN) marked a turning point in object detection research. Prior to deep learning, traditional detectors relied heavily on handcrafted feature descriptors and sliding-window techniques, which often exhibited limited robustness under varying imaging conditions. R-CNN addressed these limitations by integrating convolutional neural networks with selective search region proposals to extract high-level semantic features for object classification and localization [12].

The R-CNN framework first generates approximately 2000 candidate object regions using selective search, after which each proposal is independently processed by a deep convolutional network. Although this approach substantially improved detection accuracy on benchmark datasets such as Pascal VOC, the repeated feature extraction process resulted in considerable computational overhead. Furthermore, multi-stage training and large storage requirements restricted practical deployment in real-time applications. Nevertheless, R-CNN established the foundation for subsequent region-based detection frameworks by demonstrating the effectiveness of deep feature representations for object localization.

B. Fast R-CNN

To overcome the inefficiencies of R-CNN, Girshick introduced Fast R-CNN, which significantly reduced computational redundancy by performing convolutional feature extraction only once for the entire input image [13]. Instead of independently processing each region proposal, the model computes a shared feature map and applies a Region of Interest (RoI) pooling layer to generate fixed-length feature vectors corresponding to candidate regions.

This architectural modification simplified the training pipeline and substantially accelerated both training and inference. The introduction of a multitask loss function enabled simultaneous optimization of classification and bounding-box regression tasks, improving overall detection performance. However, Fast R-CNN still depended on selective search for region proposal generation, which remained computationally expensive and prevented true real-time detection.

C. Faster R-CNN

A major advancement was achieved with Faster R-CNN through the integration of the Region Proposal Network (RPN), which replaced external proposal generation methods with a fully learnable component [14]. The RPN shares convolutional features with the detection network, enabling candidate object regions to be generated efficiently within the same architecture.

TABLE I: Summary of Representative R-CNN Family Architectures

Model	Major Contribution	Main Limitation
R-CNN	CNN-based region classification	Slow inference
Fast R-CNN	Shared feature extraction	Selective search dependency
Faster R-CNN	Region Proposal Network	Two-stage complexity
Mask R-CNN	Instance segmentation	High computational cost

This unified framework significantly reduced inference time while maintaining high detection accuracy. The use of anchor boxes allowed the model to detect objects with varying scales and aspect ratios, improving robustness in complex visual environments. Faster R-CNN demonstrated state-of-the-art performance on benchmark datasets and became widely adopted in applications requiring precise object localization, including medical image analysis and industrial inspection. Despite these advantages, the two-stage detection pipeline remains computationally demanding compared with single-stage detectors.

D. Mask R-CNN

Building upon Faster R-CNN, He *et al.* introduced Mask R-CNN to simultaneously perform object detection and instance segmentation [15]. In addition to predicting class labels and bounding boxes, the architecture generates pixel-level segmentation masks for individual objects.

A key contribution of Mask R-CNN is the RoIAlign operation, which eliminates the quantization errors associated with RoI pooling and improves localization precision. The additional segmentation branch operates in parallel with classification and regression tasks without significantly affecting detection accuracy. Owing to its ability to capture fine-grained object boundaries, Mask R-CNN has found applications in medical diagnosis, autonomous navigation, biological imaging, and scene understanding. However, the increased architectural complexity introduces higher computational and memory requirements, limiting deployment on resource-constrained platforms.

Table I summarizes the principal characteristics of representative R-CNN-based architectures discussed in this review.

Table I indicates that the R-CNN family evolved by progressively reducing computational bottlenecks while improving localization accuracy and feature representation. Although these architectures remain highly effective for precision-oriented applications, their computational requirements motivated the development of unified single-stage detectors, particularly the YOLO family, which is discussed in the subsequent subsection.

E. YOLO Family

The introduction of the You Only Look Once (YOLO) framework established a fundamentally different approach to object detection by treating localization and classification as a single regression problem. Unlike region-based detectors that first generate candidate object proposals and subsequently classify them, YOLO directly predicts object categories and

bounding boxes from the entire image through a unified neural network. This design significantly reduces computational complexity and enables real-time inference, making YOLO particularly suitable for latency-sensitive applications such as autonomous driving, intelligent surveillance, robotics, and unmanned aerial vehicles [16].

The original YOLO architecture divided an input image into a fixed grid and assigned each grid cell the responsibility for predicting object locations and class probabilities. By processing the complete image in a single forward pass, YOLO achieved substantially higher detection speeds than contemporary two-stage detectors. Moreover, the use of global contextual information reduced false positive detections in background regions. Despite these advantages, the initial version encountered difficulties in accurately localizing small objects and densely packed targets because of its coarse spatial representation [16].

To overcome these limitations, YOLOv2 introduced several architectural improvements, including batch normalization, anchor boxes, high-resolution classifiers, and multi-scale training strategies [17]. The incorporation of anchor boxes enhanced localization capability for objects with varying dimensions, while multi-scale training improved robustness across different image resolutions. These modifications increased both detection accuracy and generalization performance without sacrificing inference speed.

A significant advancement was achieved with YOLOv3, which adopted the Darknet-53 backbone and introduced multi-scale feature prediction [18]. Feature maps extracted at different resolutions enabled more effective detection of objects with varying sizes, particularly small targets that posed challenges for earlier versions. Residual learning further improved gradient propagation and training stability, making YOLOv3 one of the most influential single-stage detection frameworks.

Subsequent developments focused on balancing accuracy, efficiency, and deployment flexibility. YOLOv4 integrated Cross Stage Partial (CSP) networks, weighted residual connections, self-adversarial training, and advanced data augmentation techniques to improve detection robustness under complex visual conditions [19]. The architecture demonstrated competitive performance on benchmark datasets while maintaining practical inference speed for real-world applications.

YOLOv5 further emphasized lightweight deployment and engineering optimization through efficient backbone structures, adaptive anchor computation, and simplified implementation pipelines. Although not formally introduced through a peer-reviewed conference publication, YOLOv5 gained considerable industrial adoption because of its ease of training, scalability, and compatibility with edge computing platforms. The architecture demonstrated that practical deployment considerations are increasingly important alongside benchmark accuracy.

F. Recent YOLO Variants

Recent YOLO variants continue to improve detection efficiency by incorporating advanced feature aggregation strategies, anchor-free detection mechanisms, transformer-inspired modules, and lightweight network designs. YOLOX eliminated anchor boxes to simplify label assignment and improve training efficiency, while PP-YOLO and related architectures integrated effective feature fusion techniques to enhance localization accuracy [20, 21].

The release of YOLOv6, YOLOv7, and YOLOv8 introduced further optimizations for industrial deployment and edge intelligence. YOLOv6 emphasized hardware-aware design for efficient inference on practical computing platforms. YOLOv7 proposed trainable bag-of-freebies strategies that improved accuracy without increasing inference cost, establishing state-of-the-art performance among real-time detectors [22]. YOLOv8 adopted an anchor-free architecture with enhanced feature extraction and simplified training procedures, improving flexibility across detection, segmentation, and tracking tasks.

Recent research trends have also explored transformer-assisted feature extraction, neural architecture search, knowledge distillation, model compression, and multimodal learning to improve detection under challenging scenarios involving occlusion, adverse weather, and crowded environments. Lightweight variants specifically target embedded devices and Internet of Things applications, where computational resources and energy consumption remain critical constraints. These developments indicate that modern object detection research increasingly seeks to balance detection precision with computational efficiency.

Table II summarizes the principal characteristics of representative YOLO architectures discussed in this survey.

TABLE II: Summary of Representative YOLO Architectures

Model	Major Contribution	Key Limitation
YOLO	Unified detection	Small object accuracy
YOLOv2	Anchor boxes and multi-scale	Dense scenes
YOLOv3	Multi-scale prediction	Moderate complexity
YOLOv4	CSP and advanced augmentation	Large model size
YOLOv5	Lightweight deployment	Industrial focus
YOLOv6–v8	Anchor-free and efficient design	Emerging benchmarks

As summarized in Table II, the evolution of the YOLO family demonstrates a consistent effort to improve detection accuracy while preserving real-time performance. Compared with region-based detectors, YOLO architectures provide an attractive balance between computational efficiency and predictive capability, particularly for applications requiring low latency and deployment on resource-constrained hardware. The complementary strengths of the YOLO and R-CNN families motivate the comparative analysis presented in subsequent sections of this review.

IV Comparative Analysis

The rapid evolution of deep learning-based object detection has established the YOLO and R-CNN families as two dominant paradigms, each optimized for different operational requirements. While region-based detectors primarily emphasize localization accuracy and feature representation, single-

stage YOLO architectures prioritize computational efficiency and real-time inference. Consequently, selecting an appropriate framework depends on application-specific constraints, including detection precision, latency, hardware resources, and deployment environment. This section presents a comparative analysis of these architectures based on four major performance indicators: accuracy, speed, computational cost, and scalability.

Detection accuracy remains one of the most important evaluation criteria for object detection systems. Region-based methods, particularly Faster R-CNN and Mask R-CNN, generally achieve superior localization precision because candidate object regions are carefully refined before classification [23, 24]. Their multi-stage detection pipeline enables effective handling of small, overlapping, and partially occluded objects. In contrast, YOLO models perform object localization and classification simultaneously through a unified network, leading to slightly lower precision in highly crowded scenes. However, recent YOLO variants have substantially reduced this performance gap through improved feature aggregation, anchor-free prediction mechanisms, and enhanced training strategies [25, 26].

Inference speed represents the principal advantage of the YOLO family. Since object detection is formulated as a single regression task, YOLO architectures require only one forward pass to produce final predictions, enabling real-time processing for video analytics and edge computing applications. Two-stage R-CNN detectors require region proposal generation followed by classification and bounding-box refinement, resulting in higher computational latency. Although Faster R-CNN significantly reduced the bottlenecks of earlier architectures through Region Proposal Networks, its inference speed remains lower than modern YOLO implementations [24, 25].

Computational cost is another critical factor affecting practical deployment. R-CNN-based architectures require substantial memory resources and computational power because of their multi-stage processing pipeline and complex feature extraction mechanisms. These requirements often necessitate dedicated GPU hardware for efficient execution. Conversely, YOLO models are designed with computational efficiency in mind and can be optimized for embedded devices, mobile platforms, and Internet of Things applications. Lightweight variants further reduce model size while maintaining competitive detection performance, making them attractive for resource-constrained environments.

Scalability has become increasingly important as modern vision systems operate across heterogeneous hardware platforms and diverse application scenarios. The modular design of the R-CNN family facilitates integration with advanced segmentation and feature extraction techniques but often increases architectural complexity. YOLO architectures demonstrate superior scalability because they can be adapted for object detection, segmentation, tracking, and multi-task learning while preserving real-time capabilities. Recent developments involving transformer modules, efficient backbone networks, and model compression techniques have further enhanced the adaptability of YOLO-based systems for large-scale industrial deployment [26, 27].

TABLE III: Comparative Analysis of YOLO and R-CNN Families

Parameter	YOLO Family	R-CNN Family
Detection Accuracy	High	Very High
Inference Speed	Very High	Moderate
Computational Cost	Low to Moderate	High
Memory Requirement	Moderate	High
Small Object Detection	Good	Excellent
Real-Time Capability	Excellent	Limited
Edge Deployment	Highly Suitable	Challenging
Scalability	High	Moderate
Architecture Complexity	Moderate	High
Typical Applications	Autonomous Driving, Robotics, Surveillance	Medical Imaging, Industrial Inspection, Precision Analysis

Table III summarizes the overall comparison between the YOLO and R-CNN families based on the major performance characteristics discussed above. The comparison indicates that R-CNN architectures remain preferable for applications requiring high localization precision, whereas YOLO models provide an effective balance between accuracy and computational efficiency for real-time intelligent vision systems.

As summarized in Table III, neither detection paradigm universally outperforms the other across all performance dimensions. The R-CNN family offers superior detection precision and robust feature learning for accuracy-critical tasks, whereas the YOLO family provides exceptional inference speed and computational efficiency for practical real-time deployment. Therefore, the selection of an object detection framework should be guided by application requirements, available computational resources, and operational constraints rather than relying solely on benchmark accuracy.

V Research Gaps

Despite the significant advancements achieved by both the YOLO and R-CNN families, several research challenges continue to limit their effectiveness in practical intelligent vision systems. Although region-based detectors provide high localization accuracy and single-stage detectors offer real-time performance, no existing architecture simultaneously achieves optimal accuracy, low latency, computational efficiency, and scalability across diverse application scenarios. These limitations indicate the need for further innovation in object detection methodologies.

One of the major research gaps involves the reliable detection of small, densely packed, and partially occluded objects. R-CNN-based architectures generally demonstrate superior localization capabilities but often incur substantial computational overhead, whereas YOLO models prioritize inference speed at the expense of detection precision in challenging environments. Similarly, adverse weather conditions, illumination variations, motion blur, and complex backgrounds continue to degrade detection performance for both paradigms, particularly in autonomous driving and intelligent surveillance applications [28, 29].

Another important challenge is the efficient deployment of deep object detection models on edge and embedded platforms. Modern architectures often require significant computational resources, memory capacity, and energy consumption, limiting their applicability in Internet of Things devices, mobile robots, and unmanned aerial vehicles. Although lightweight network designs and model compression techniques have shown promising results, achieving an effective balance between model complexity and detection accuracy remains an open research problem [30].

Furthermore, the majority of existing object detection frameworks rely heavily on large-scale annotated datasets for supervised training. The high cost of manual annotation and the limited availability of domain-specific datasets restrict the adaptability of current models to emerging application domains. Improving transfer learning, self-supervised learning, domain adaptation, and multimodal learning strategies could enhance model generalization while reducing dependence on extensive labeled data. In addition, standardized evaluation protocols for balancing accuracy, inference speed, robustness, and resource efficiency across heterogeneous hardware platforms require further investigation.

These research gaps suggest that future object detection systems should integrate efficient feature representation, adaptive learning mechanisms, and hardware-aware optimization strategies to develop scalable, robust, and real-time vision solutions capable of addressing practical deployment challenges.

VI Future Directions

The rapid advancement of deep learning and intelligent computing is expected to significantly influence the next generation of object detection systems. Future research should focus on developing hybrid architectures that combine the localization accuracy of region-based detectors with the computational efficiency of single-stage models. The integration of transformer-based feature extraction, attention mechanisms, and adaptive feature fusion strategies has shown considerable potential for improving detection performance in complex visual environments while maintaining real-time capability.

Another promising direction involves the design of lightweight and hardware-aware detection frameworks for edge computing platforms. Model compression, neural architecture search, knowledge distillation, and quantization techniques can reduce computational complexity without substantially affecting detection accuracy. These developments are particularly important for autonomous vehicles, mobile robotics, Internet of Things devices, and unmanned aerial systems, where computational resources and energy consumption are limited. Furthermore, self-supervised learning, domain adaptation, and multimodal learning approaches may reduce dependence on large annotated datasets while improving model robustness across diverse operating conditions.

Future object detection research should also emphasize explainability, robustness against adversarial attacks, and energy-efficient deployment to support trustworthy artificial intelligence applications. The convergence of efficient neural architectures with intelligent edge computing and adaptive learning strategies is expected to facilitate scalable and reliable vision systems capable of addressing increasingly complex real-world challenges.

VII Conclusion

This review presented a comprehensive comparative study of the YOLO and R-CNN families of deep learning-based object detection frameworks for efficient vision systems. The analysis demonstrated that region-based architectures, including Faster R-CNN and Mask R-CNN, provide high localization accuracy and robust feature representation, making them suitable for precision-critical applications. In contrast, YOLO-based detectors achieve exceptional inference speed and computational efficiency through unified single-stage prediction, enabling practical deployment in real-time intelligent systems.

The comparative investigation revealed that no single detection framework universally satisfies all performance requirements. Instead, the selection of an appropriate architecture depends on the intended application, available computational resources, latency constraints, and accuracy demands. Recent advances in lightweight network design, anchor-free detection, transformer-assisted feature extraction, and edge intelligence have substantially reduced the trade-off between speed and precision, creating new opportunities for scalable object detection solutions.

The identified research gaps and emerging technological trends indicate that future object detection systems should integrate efficient feature learning, adaptive optimization strategies, and hardware-aware implementations to achieve robust real-time performance. Overall, this review provides a consolidated perspective on the evolution, capabilities, and practical implications of YOLO and R-CNN architectures, offering useful insights for researchers and practitioners engaged in the development of next-generation intelligent vision systems.

References

[1] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, vol. 25, 2012, pp. 1097–1105.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[4] R. Girshick, “Fast R-CNN,” in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal

networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[6] K. He, G. Gkioxari, P. Doll'ar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2980–2988.
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
[8] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement," *PLoS Medicine*, vol. 6, no. 7, Art. no. e1000097, 2009.
[9] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University and Durham University Joint Report, EBSE-2007-01, 2007.
[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
[11] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. European Conf. Computer Vision (ECCV)*, 2014, pp. 740–755.
[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.
[13] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2015, pp. 1440–1448.
[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2980–2988.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
[17] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271.
[18] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv:1804.02767, 2018.
[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv:2004.10934, 2020.
[20] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," arXiv:2107.08430, 2021.
[21] X. Long, K. Deng, G. Wang, Y. Zhang, and Q. Dang, "PP-YOLO: An effective and efficient implementation of object detector," arXiv:2007.12099, 2020.
[22] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv:2207.02696, 2022.
[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2980–2988.
[25] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271.
[26] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv:2207.02696, 2022.
[27] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," arXiv:2107.08430, 2021.
[28] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7263–7271.
[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, 2017, pp. 2980–2988.
[30] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv:2207.02696, 2022.

