Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 3, No. 2, 2025

A Comparative Review of the Next-Generation YOLO Models: YOLOv10 and YOLO11

Feiyu Chen [1], Yingqian Zhang [2], Lei Fu [1], Rongru Hua [3,*], Qian Zhang [1], Shihao Bi [1]

[1] School of Mechanical Engineering, Sichuan University of Science and Engineering, China

[2] School of Civil Engineering, Sichuan University of Science and Engineering, China

[3] Sichuan Shengtuo Testing Technology Co. Ltd., China

Abstract: In recent years, the YOLO (You Only Look Once) series has remained a mainstream framework in object detection, continuously driving the balance between lightweight design and high accuracy. This paper focuses on two pivotal versions—YOLOv10 and YOLO11—and provides a systematic comparison and analysis in terms of architectural design, core modules, performance characteristics, and application scenarios. YOLOv10 introduces a unified end-to-end architecture that eliminates anchors and post-processing steps, thereby simplifying the detection pipeline and significantly improving deployment efficiency. In contrast, YOLO11 builds upon YOLOv8 by reconstructing its modules through the introduction of the C3k structure and an optimized feature fusion pathway, further enhancing detection accuracy and model representation capabilities. This review outlines the similarities and differences in structural philosophy and application orientation between the two models, summarizes their technological evolution, and explores the potential future directions of the YOLO series in multi-task integration, adaptive modeling, and lightweight deployment. The findings of this study aim to serve as a reference for the design and selection of object detection systems.

Keywords: Object detection; YOLO; YOLOv10; YOLO11.

1. Introduction

Object detection [1] is a fundamental task in computer vision, aiming to simultaneously classify and localize objects within images or video frames. In recent years, driven by advancements in deep neural networks, convolutional neural network (CNN) [2]-based object detection methods have evolved rapidly, achieving remarkable improvements in both accuracy and efficiency. Among them, the YOLO [3] (You Only Look Once) series stands out as one of the most representative one-stage detection frameworks. Owing to its end-to-end architecture, fast inference speed, and strong deployment adaptability, YOLO has been widely adopted in real-world applications and has become a mainstream approach for object detection tasks.

Since the introduction of YOLOv1 [4], the series has undergone continuous optimization and multiple iterations. YOLOv3 [5] incorporated the Darknet-53 [6] backbone to enhance feature representation capabilities; YOLOv4 [7] integrated various training strategies and module improvements; and YOLOv5 [8] further emphasized lightweight design and engineering practicality. Subsequently, YOLOv6 [9] and YOLOv7 [10] explored new directions in industrial deployment and feature fusion strategies, respectively. YOLOv8 [11] introduced a comprehensive module-level upgrade, featuring a redesigned backbone, the C2f [12] module, and an anchor-free mechanism, laying a solid foundation for future advancements. These developments reflect the series' ongoing efforts to balance speed, accuracy, and deployability.

Against this backdrop, YOLOv10 [13] and YOLOv11 [14] were both released in 2024, representing two distinct evolutionary paths within the YOLO framework. YOLOv10, proposed by a research institution, emphasizes unified modeling and end-to-end training. By eliminating anchor boxes, feature fusion modules, and post-processing steps such as non-maximum suppression [15] (NMS), it pursues architectural simplicity and task integration, aiming to advance object detection toward a purer end-to-end paradigm. A key innovation lies in its novel rethinking of the detection pipeline, unifying classification and regression tasks under a single loss function, thereby achieving theoretical completeness and a more robust optimization loop.

In contrast, YOLOv11, developed by the Ultralytics team, builds upon the YOLOv8 foundation and focuses on module-level optimization and inference efficiency. Its major improvements include the reconstruction of the backbone and neck using the newly introduced C3k and C3k2 [16] modules to replace the original C2f structure. It also refines the feature fusion path and detection head design. These enhancements enable a more lightweight model, improved feature extraction efficiency, and reduced inference latency, all without altering the overall modeling paradigm. YOLOv11 reflects a more engineering-oriented approach, emphasizing practicality and deployment-friendliness.

In summary, although both YOLOv10 and YOLOv11 belong to the YOLO family, they exhibit significant differences in design philosophy, structural approach, and optimization objectives. YOLOv10 represents a theoretical and paradigmatic rethinking of object detection, while YOLOv11 embodies continued refinement of engineering performance and practical adaptability. This paper provides a systematic comparison and analysis of their core principles, key modules, technical trajectories, and evolutionary background, aiming to offer valuable insights for future research and application development.

2. YOLOv10 Architecture and Principle Analysis

2.1. Design Motivation and Research Objectives

YOLOv10 was proposed to further advance object detection towards lightweight, efficient, and fully end-to-end models. Developed by the Meta Research team in 2024, its core philosophy lies in eliminating multi-stage modules commonly found in traditional detection pipelines—such as anchor generation and non-maximum suppression (NMS)—and simplifying the inference path through a unified detection structure. The goal is to reduce deployment complexity and enhance the performance of small models on resource-constrained devices.

Specifically, YOLOv10 focuses on two main objectives: (1) to construct a lightweight detection framework optimized for edge deployment, improving accuracy and inference speed on mobile devices, and (2) to bridge the gap between training and inference, achieving a closed-loop end-to-end architecture that reduces optimization bias and enhances detection robustness. While maintaining the real-time advantages typical of the YOLO series, the model introduces module-level reconstruction and task modeling strategies, offering a new perspective for the evolution of next-generation efficient detectors.

2.2. Network Architecture Analysis

YOLOv10 retains the classic three-part YOLO architecture of Backbone–Neck–Head, but its design significantly evolves towards unification and simplification to reduce deployment difficulty and improve overall efficiency. In the Backbone, YOLOv10 adopts a symmetric network design philosophy. By unifying channel configurations and standardizing downsampling ratios [17], it achieves a balance between information retention and computational complexity during feature extraction [18]. In lightweight model variants, some modules in the backbone are replaced with more compact convolution-activation blocks to further reduce model size and computational demand.

For the Neck, YOLOv10 abandons redundant paths and extensive lateral connections present in previous YOLO models. Instead, it uses a compact multi-scale feature fusion strategy, retaining only essential cross-level interactions. By controlling the depth and width of the fusion paths, the model avoids inference delays caused by information redundancy. This streamlined design also lays a solid foundation for efficient inference.

In the Head design, YOLOv10 introduces a unified detection head. This head eliminates both anchor-based and traditional anchor-free box generation strategies, directly predicting class probabilities and bounding box coordinates at each feature point, thereby simplifying the detection process. Additionally, the model removes the NMS step and instead performs final target selection using distributed prediction, confidence-based ranking, and soft filtering strategies, further enhancing training-inference consistency. This fully end-to-end detection design makes YOLOv10 not only easier to deploy but also more real-time and stable.

2.3. Core Module Introduction

YOLOv10's efficiency and deployment advantages largely stem from structural innovations and optimization strategies within its core modules. First, the backbone employs a symmetric channel design, using normalized network width and depth configurations to ensure good scalability and structural stability across different feature scales. This also reduces dimensional mismatches in cross-layer connections and facilitates automatic tensor matching and operator fusion during inference, improving execution efficiency.

The detection head utilizes a unified modeling strategy, merging classification and localization into a single prediction stream. It outputs the center offsets, size parameters, and class scores of objects directly, avoiding the redundancy introduced by anchor boxes. Compared with traditional anchor-free methods, YOLOv10's head improves spatial localization accuracy through dense sampling, while its simplified regression paradigm enhances bounding box fitting.

In terms of target assignment, YOLOv10 introduces a task-aligned one-to-one matching strategy. This approach considers both classification loss and localization error during assignment, enhancing semantic consistency and training stability in positive-negative sample selection. Compared with traditional IoU [19]-based matching or dynamic K matching, this strategy offers more stable convergence in early training stages and improves recall in dense object scenarios.

The most innovative aspect of YOLOv10 lies in its anchor-free, NMS-free inference path. The model replaces lengthy post-processing steps with a confidence-guided soft sorting strategy, allowing the entire detection process to complete in a single forward pass. This not only speeds up execution but also provides a highly generalizable detection paradigm across platforms.

2.4. Performance Analysis

According to official reports and results from open-source code, YOLOv10 demonstrates a high accuracy-efficiency trade-off on the COCO dataset. For example, YOLOv10-S achieves 38.9% mAP while maintaining a 40 FPS inference speed, significantly outperforming models of the same scale like YOLOv5s and YOLOv8n. In terms of parameter size, YOLOv10 submodels are slightly smaller than their YOLOv8 counterparts, and their FLOPs are more stable, indicating excellent computational efficiency.

Due to its simplified structure and unified modules, YOLOv10 also performs exceptionally well on edge devices such as the Jetson series, Raspberry Pi, and Android platforms. It offers fast inference, low memory consumption, and a smooth model conversion and optimization process, making it a strong foundation for large-scale deployment. Figure 1 shows the network architecture of YOLOv10.

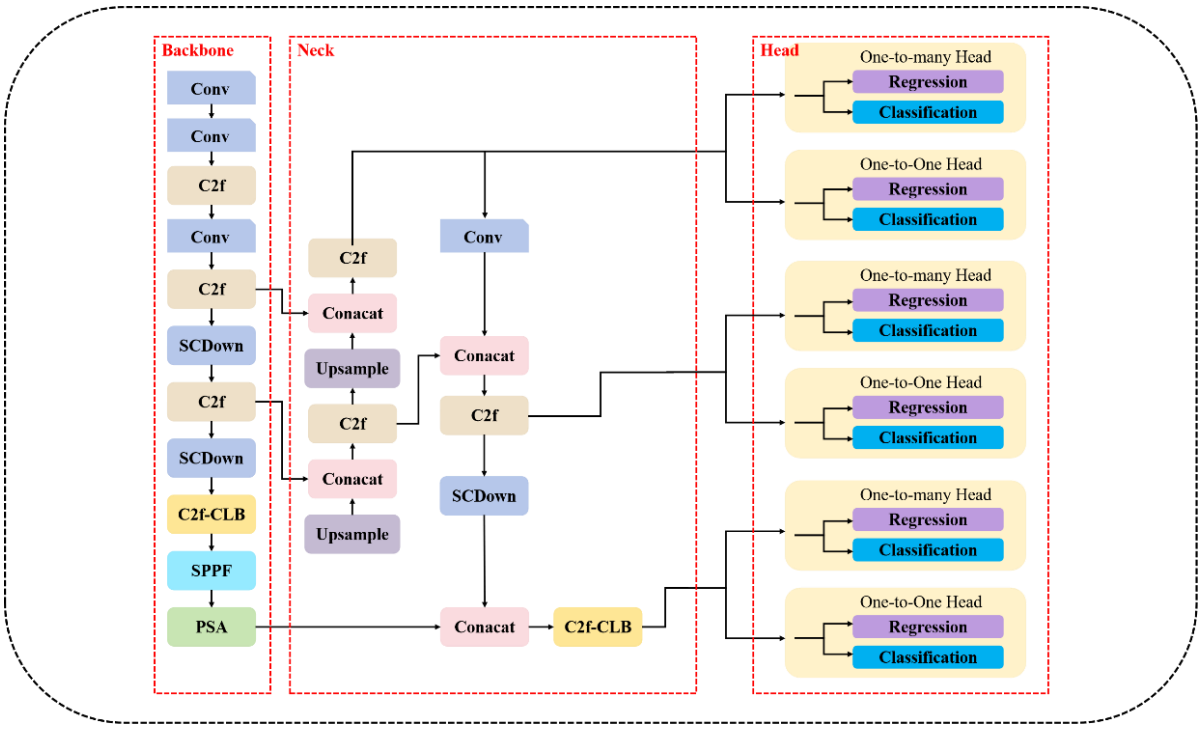

Figure 1. YOLOv10 network structure

3. Structural and Principle Analysis of YOLO11

3.1. Background and Optimization Objectives

YOLO11 is an evolutionary development based on the successful application of YOLOv8, with its core objective being to achieve a better balance between detection accuracy and deployment efficiency. As industrial application scenarios impose increasingly stringent requirements on multiple indicators such as model accuracy, latency, and size, YOLO11 continues the series' hallmark of efficient detection while further enhancing the expressive capacity and structural flexibility of the network.

The optimization of YOLO11 mainly focuses on three aspects: First, the backbone network incorporates new modules with stronger representational capacity to improve multi-scale feature modeling. Second, the Neck structure is redesigned to reconstruct the feature fusion paths and optimize the fusion mechanism, thereby enhancing semantic information propagation and spatial localization capability. Finally, the detection head further simplifies the Anchor-Free framework to improve the efficiency of positive-negative sample matching and reduce redundant processing in the end-to-end inference path. Through this series of improvements, YOLO11 aims to establish a new balance between lightweight deployment and high-accuracy applications, offering a more adaptive solution for complex object detection tasks.

3.2. Overview of Network Architecture

YOLO11 retains the overall architecture of backbone–neck–detection head from YOLOv8 but introduces targeted improvements in each sub-module. In the backbone, the C3k module replaces the original C2f structure. This module introduces deep convolutional pathways and cross-layer residual connections, which effectively enhance the information flow in the feature extraction process while maintaining low parameter count and computational complexity. The multi-branch structure of C3k introduces more nonlinear transformations while preserving lightweight characteristics, thereby improving the backbone's ability to represent complex-shaped objects.

In the Neck, feature fusion paths have been adjusted with the introduction of deeper cross-connections and direction-aware feature reconstruction mechanisms, enabling higher-level semantic information to be more effectively fed back to lower-level features, thus improving the detection of small objects. Additionally, by redesigning the number of concatenation paths and the scale hierarchy, YOLO11 significantly reduces intermediate feature memory usage while maintaining detection accuracy, thereby optimizing memory access efficiency during inference.

For the detection head, YOLO11 further improves YOLOv8's Anchor-Free design. With a simplified positive sample allocation mechanism and a multi-task joint prediction strategy, this detection head can achieve high-confidence bounding box regression using fewer candidate positions. Furthermore, with an optimized loss function weight allocation mechanism, YOLO11 significantly alleviates the training bias caused by regression or classification dominance, thereby improving the model's robustness in dense object and complex background scenarios.

3.3. Analysis of Core Modules

One of YOLO11's core innovative modules is the C3k structure, which extends and optimizes the C2f module from YOLOv8. While retaining the original lightweight path, it introduces stacked Bottleneck residual blocks in the main branch to enhance network depth and nonlinear representational power. After receiving input features, C3k divides channels into multiple sub-paths, each of which undergoes separate convolutional transformations before being concatenated and fused, thereby effectively enhancing inter-channel information interaction. C3k2, as a further evolved structure, adds a channel selection mechanism and spatial attention guidance, further enhancing the response to target regions and improving background suppression

capabilities.

In addition, YOLO11 modifies the path aggregation structure in the Neck by introducing deep semantic back-propagation paths and spatial alignment strategies. Unlike YOLOv8, which mainly relies on FPN/PAN configurations, YOLO11 adds direction-sensitive skip connections and layer-wise scale-matching convolutions in its fusion paths, making cross-scale information transfer more efficient. This optimization significantly improves receptive field completeness and detail recovery in scenarios with large variations in object size.

In the detection head, YOLO11 follows YOLOv8's Anchor-Free approach but further simplifies the structure to reduce matching complexity and inference redundancy. Thanks to an improved positive-negative sample filtering strategy, the model achieves fast convergence in early training stages while maintaining high recall in densely populated regions. The prediction head focuses on regressing object center offsets and sizes, and with the aid of a task-consistent loss function design, the model significantly improves in terms of stability and bounding box fitting accuracy.

3.4. Performance Evaluation

According to the official data released by the YOLO team, YOLO11 outperforms YOLOv10 and YOLOv8 across multiple public datasets. Different versions of the YOLO11 model (such as YOLO11n, YOLO11s, YOLO11m) have achieved better balance in terms of mAP improvement, model size control, and inference latency. For instance, YOLO11n further reduces model parameters to the single-digit MB level without compromising accuracy, enabling smooth deployment on edge devices such as Raspberry Pi and Jetson Nano.

In horizontal comparisons, YOLO11 not only maintains better accuracy and stability for medium and large object detection than YOLOv10, but also demonstrates improved generalization [20] capabilities in complex backgrounds and dense small-object scenarios. Moreover, inference latency evaluations on multiple hardware platforms show that, due to structural simplifications and module integration, YOLO11 achieves lower average latency under the same hardware conditions, providing greater deployment adaptability. Figure 2 shows the network architecture of YOLO11.

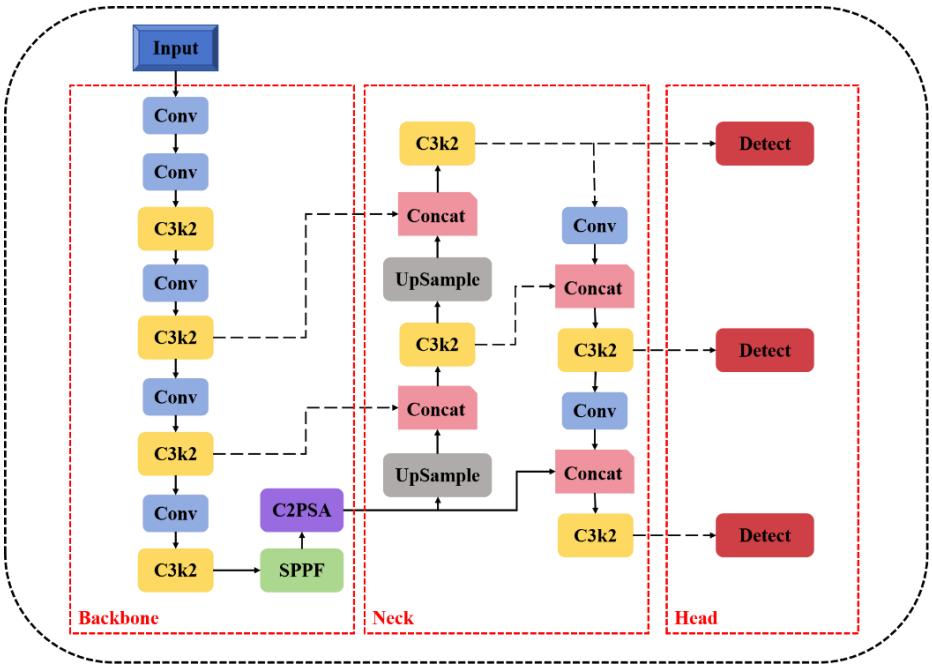

Figure 2. YOLO11 network structure

4. Comparative Analysis of YOLOv10 and YOLO11

4.1. Architectural Design Comparison

YOLOv10 and YOLO11 differ significantly in their architectural design philosophies. YOLOv10 emphasizes an end-to-end integrated detection pipeline, aiming to build a unified and streamlined object detection framework. In contrast, YOLO11 focuses on fine-grained modular evolution, continuing the modular structure of YOLOv8 by enhancing overall performance through structural replacements and feature flow reconstruction.

The network architecture of YOLOv10 reflects a high degree of uniformity. Its Backbone, Neck, and Head adopt a consistent design language with symmetrical and standardized construction, significantly reducing cross-module redundancy and coupling. This facilitates collaborative end-to-end training by the optimizer. Additionally, YOLOv10 eliminates traditional components such as anchor mechanisms and post-processing NMS, opting instead for integrated output encoding. This ensures consistency between the training and inference stages, simplifying the detection process and reducing deployment complexity, especially on edge devices.

In contrast, YOLO11 places more emphasis on expressive modular design and flexible component combinations. Its backbone introduces the enhanced C3k structure, which improves the depth and breadth of feature extraction through multi-branch information pathways. For the Neck, YOLO11 strengthens semantic back-propagation and feature alignment strategies, enabling richer multi-scale feature interaction. In the detection Head, YOLO11 retains the anchor-free paradigm but introduces task-adaptive designs in matching strategies and loss functions, enhancing both stability and generalization.

Their post-processing strategies also differ. YOLOv10 directly constrains outputs using a built-in training loss,

avoiding the need for explicit NMS. YOLO11, however, still incorporates NMS as a final selection module on top of its anchor-free prediction head, striking a balance between accuracy and speed. This makes YOLOv10 more suitable for end-to-end optimized deployment scenarios, while YOLO11 retains greater prediction conservativeness without compromising inference efficiency.

4.2. Performance Comparison

In terms of performance metrics, YOLO11 generally outperforms YOLOv10 across various public benchmarks, particularly excelling in small object detection and complex scene modeling. With mAP as a key indicator, YOLO11 variants (e.g., v11n, v11s) consistently achieve higher accuracy than their YOLOv10 counterparts, with average improvements of around 1.5% to 2.3% on datasets like COCO and VisDrone.

Regarding resource consumption, YOLOv10 benefits from its simplified architecture and unified parameter flow, resulting in lower model size and FLOPs, making it ideal for ultra-lightweight deployment. It offers faster response and better power efficiency on non-GPU or edge-limited devices. In contrast, YOLO11 introduces slightly higher model size and computation in exchange for accuracy improvements, but still maintains low inference latency on modern hardware platforms such as high-end GPUs and Jetson Xavier, keeping efficiency within acceptable bounds.

In deployment tests across various hardware platforms, YOLOv10's unified structure enables higher degrees of automated optimization through end-to-end deployment toolchains such as TensorRT and ONNX. YOLO11, while requiring more refined configuration in model conversion and quantization due to its complex modules, still maintains superior accuracy and robustness post-conversion, thanks to its expressive architecture.

4.3. Application Scenario Adaptability Comparison

In real-world applications, YOLOv10 and YOLO11 are each suited to different demands. YOLOv10, with its unified architecture and streamlined workflow, is better suited to industrial deployment scenarios that rely heavily on end-to-end optimization. It is particularly applicable in environments requiring strict structural constraints and high consistency in processing pipelines, such as intelligent edge devices and embedded vision systems. Its anchor-free and NMS-free modeling approach simplifies system design and shortens deployment and debugging cycles.

YOLO11, on the other hand, shows greater versatility in research and product development. Its flexible modular design allows for easy integration with various enhancement mechanisms—such as attention modules and wavelet-based feature processing—making it a strong foundation for architectural innovations in academic research. Moreover, its superior accuracy performance makes it well-suited for tasks demanding high robustness and precise object boundary localization, including industrial defect detection, multi-class object recognition in autonomous driving, and urban traffic surveillance.

In summary, YOLOv10 is more appropriate for scenarios with high requirements on system controllability and deployment consistency, while YOLO11 is better aligned with the pursuit of cutting-edge detection performance and flexible modular extensibility. These two models represent distinct development paths in the YOLO ecosystem: structural unification and modular refinement.

5. Conclusion and Outlook

As major evolutionary versions in the YOLO series, YOLOv10 and YOLO11 represent two distinct technological trajectories. YOLOv10 centers on end-to-end integration, removing anchors and post-processing to achieve structural uniformity and efficient inference—particularly suited to resource-constrained edge deployment. YOLO11, building upon YOLOv8, incorporates the C3k module and optimizes feature fusion and detection head structures, significantly improving detection accuracy and model expressiveness while maintaining deployment efficiency. It is better tailored to high-performance demands in general-purpose vision tasks.

The two models showcase respective strengths in architectural design, performance, and application adaptability, reflecting the trade-off between "unification" and "modular refinement" within the YOLO architecture. Future YOLO development is expected to further explore model lightweighting and multi-task fusion, enhancing adaptability and robustness in complex scenarios. YOLOv10 and YOLO11 together lay a solid foundation for continued research and offer complementary strengths in practical applications, delivering significant value for both academic and industrial contexts.

References

[1] Zhao Z Q, Zheng P, Xu S, et al. Object detection with deep learning: A review [J]. IEEE transactions on neural networks and learning systems, 2019, 30(11): 3212-3232.

[2] Wu J. Introduction to convolutional neural networks [J]. National Key Lab for Novel Software Technology. Nanjing University. China, 2017, 5(23): 495.

[3] Hussain M. YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection [J]. Machines, 2023, 11(7): 677.

[4] Terven J, Córdova-Esparza D M, Romero-González J A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas [J]. Machine learning and knowledge extraction, 2023, 5(4): 1680-1716.

[5] Zhao L, Li S. Object detection algorithm based on improved YOLOv3 [J]. Electronics, 2020, 9(3): 537.

[6] Yang L, Chen G, Ci W. Multiclass objects detection algorithm using DarkNet-53 and DenseNet for intelligent vehicles [J]. EURASIP Journal on Advances in Signal Processing, 2023, 2023(1): 85.

[7] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection [J]. arXiv preprint arXiv:2004.10934, 2020.

[8] Guo G, Zhang Z. Road damage detection algorithm for improved YOLOv5 [J]. Scientific reports, 2022, 12(1): 15523.

[9] Norkobil Saydirasulovich S, Abdusalomov A, Jamil M K, et al. A YOLOv6-based improved fire detection approach for smart city environments [J]. Sensors, 2023, 23(6): 3161.

[10] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors [C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7464-7475.

[11] Zhai X, Huang Z, Li T, et al. YOLO-Drone: an optimized YOLOv8 network for tiny UAV object detection [J]. Electronics, 2023, 12(17): 3664.

[12] Chen Y, Zhan S, Cao G, et al. C2f-Enhanced YOLOv5 for Lightweight Concrete Surface Crack Detection [C]//Proceedings of the 2023 International Conference on Advances in Artificial Intelligence and Applications. 2023: 60-64.

[13] Wang A, Chen H, Liu L, et al. Yolov10: Real-time end-to-end object detection [J]. Advances in Neural Information Processing Systems, 2024, 37: 107984-108011.

[14] He L, Zhou Y, Liu L, et al. Research on object detection and recognition in remote sensing images based on YOLOv11 [J]. Scientific Reports, 2025, 15(1): 14032.

[15] Hosang J, Benenson R, Schiele B. Learning non-maximum suppression [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4507-4515.

[16] Chen X, Jiang N, Yu Z, et al. Citrus leaf disease detection based on improved YOLO11 with C3K2 [C]//International Conference on Computer Graphics, Artificial Intelligence, and Data Processing (ICCAID 2024). SPIE, 2025, 13560: 746-751.

[17] Aflaki P, Hannuksela M M, Häkkinen J, et al. Impact of downsampling ratio in mixed-resolution stereoscopic video [C]//2010 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video. IEEE, 2010: 1-4.

[18] Mutlag W K, Ali S K, Aydam Z M, et al. Feature extraction methods: a review [C]//Journal of Physics: Conference Series. IOP Publishing, 2020, 1591(1): 012028.

[19] Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression [C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 12993-13000.

[20] Zhang C, Bengio S, Hardt M, et al. Understanding deep learning (still) requires rethinking generalization [J]. Communications of the ACM, 2021, 64(3): 107-115.