WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 6, Number 1, Year 2025
DOI: https://doi.org/10.62051/ijcsit.v6n1.12
Journal homepage: https://wepub.org/index.php/IJCSIT/index

From YOLOv5 to YOLOv8: Structural Innovations and Performance Improvements

Feiyu Chen [1], Yingqian Zhang [2], Lei Fu [1], Hui Xie [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

With the rapid advancement of object detection technology, the YOLO series has become ubiquitous across diverse computer vision applications owing to its efficiency and real-time capabilities. This paper delivers a systematic comparative analysis of YOLOv5 and YOLOv8, with an emphasis on their innovations and distinctions in network architecture, training mechanisms, inference optimizations, and detection performance. Relative to YOLOv5, YOLOv8 introduces substantial structural enhancements, notably the lightweight C2f feature extraction module and an anchor-free detection head, alongside state-of-the-art data augmentation strategies and novel loss functions that collectively boost both accuracy and inference speed. Furthermore, YOLOv8 advances inference optimization by supporting more flexible model export formats and acceleration pipelines, thereby facilitating deployment on mobile and edge devices. Through this comparison, we trace the technological evolution from YOLOv5 to YOLOv8 and project future trends in object detection research—particularly the integration of emerging techniques to further elevate model efficiency and performance. Our findings underscore the enduring potential of the YOLO series to drive progress in object detection methodologies.

KEYWORDS

Object detection; YOLO; YOLOv5; YOLOv8

1. INTRODUCTION

In recent years, Object Detection (OD) [1], as a core task in computer vision, has been widely used in many fields such as intelligent security, automatic driving, and industrial quality inspection. Along with the development of deep learning, the detection algorithms based on convolutional neural network (CNN) [2] have made significant progress, especially the single-stage target detector represented by the YOLO (You Only Look Once) [3] series, which has attracted widespread attention due to its high-speed, lightweight, and balanced accuracy.

Since YOLOv1 was first proposed in 2016, the YOLO series has been continuously iterated and optimized, and several versions of YOLOv2 [4], YOLOv3 [5], and YOLOv4 [6] have been launched. Among them, YOLOv5 [7], which was open sourced by the Ultralytics team in 2020, has become one of the most widely used versions, and has become the preferred choice for many real-world projects due to its modularized structural design, rich model sizing (n, s, m, l, x), and friendly training deployment support. However, YOLOv5 is not an official version, and its nomenclature as well as the anchor-based detection mechanism have sparked an ongoing discussion in the academic community.

With the increase of computing power and the introduction of new technologies, Ultralytics officially released YOLOv8 [8] in 2023. As a major update of the YOLO series, YOLOv8 has been completely restructured compared to YOLOv5. It abandons the traditional anchor-based mechanism and adopts a more flexible anchor-free detection head, while introducing an improved C2f feature extraction module, leading to significant improvements in detection accuracy and model generalization. In addition, YOLOv8 natively supports multi-task detection (e.g., image segmentation, pose estimation) and mainstream deployment formats such as ONNX and TensorRT, which further broadens its application space in edge computing and industrial scenarios.

Although YOLOv8 shows better performance than YOLOv5 in many aspects, there are still significant differences between the two in terms of structural design, detection mechanism, training strategy, and actual deployment. To address this situation, this paper provides a systematic review and comparative analysis of YOLOv5 and YOLOv8, comprehensively analyzes their architectural evolution, key technical features and performance differences.

2. OVERVIEW OF YOLOV5

YOLOv5 is a single-stage object detector released by the Ultralytics team in 2020. Although it is not a direct continuation of the original YOLO series, its high inference speed, flexible scalability, and excellent detection performance have led to its rapid adoption in both industry and academia. Compared with earlier YOLO versions, YOLOv5 introduces numerous innovations in network architecture, training methodology, and inference optimization, achieving a well-balanced trade-off among accuracy, speed, and resource consumption.

2.1. Network Architecture

The overall architecture of YOLOv5 follows the canonical three-stage paradigm of feature extraction [9] → feature aggregation [10] → detection prediction, implemented via modular Backbone, Neck, and Head components to enable efficient end-to-end detection.

2.1.1. Backbone

For feature extraction, YOLOv5 employs CSPDarknet53, a Cross-Stage Partial (CSP) [11] variant of Darknet53 [12]. By introducing a partial-split mechanism within each residual block, CSPDarknet53 reduces parameter count and computational cost while enhancing representational capacity. This design not only improves computational efficiency but also mitigates gradient vanishing, thereby boosting the learning of deep features [13].

2.1.2. Neck

The Neck module integrates a Path Aggregation Network (PANet) [14] to strengthen multi-scale feature fusion. PANet augments shallow features via a bottom-up path and combines this with both upsampling and downsampling operations, yielding rich semantic and fine-grained [15] representations that feed into the detection head.

2.1.3. Head

YOLOv5's detection head retains an anchor-based mechanism, performing bounding-box regression and object classification using predefined anchors. Unlike traditional shared-head designs, YOLOv5 adopts a decoupled head [16], in which separate branches handle classification and regression tasks. This decoupling reduces inter-task interference, enhances training stability, and notably improves performance on small objects.

2.2. Key Techniques

Beyond architectural refinements, YOLOv5 incorporates a suite of training and inference optimizations that further elevate overall performance and practicality.

2.2.1. Data Augmentation

During training, YOLOv5 applies advanced augmentation strategies such as Mosaic and MixUp. Mosaic stitches four random images into a single composite, vastly increasing sample diversity and scale variation, thereby strengthening robustness to complex backgrounds. MixUp blends two images and their labels, which mitigates over-fitting and enhances generalization [17].

2.2.2. Loss Function

For bounding-box regression, YOLOv5 leverages the Complete Intersection over Union (CIoU) [18] loss. CIoU extends traditional IoU [19] by accounting for center-point distance and aspect-ratio consistency, accelerating convergence and improving localization accuracy—especially in scenarios with large pose or scale variation.

2.2.3. Training Strategy

YOLOv5 employs adaptive learning-rate schedules such as Cosine Annealing and the OneCycle policy to boost training efficiency and convergence. Coupled with transfer learning—fine-tuning from pre-trained weights—these strategies substantially shorten training time on small datasets and improve model stability and final accuracy, facilitating rapid deployment across diverse applications.

2.3. Model Variants and Applications

To meet varying resource constraints and detection requirements, YOLOv5 offers five model variants—v5n (Nano), v5s (Small), v5m (Medium), v5l (Large), and v5x (Extra-large). Each trades off parameter count, FLOPs, and accuracy to suit different deployment scenarios. Extensive evaluations on COCO and VOC datasets demonstrate YOLOv5's outstanding balance of inference speed and detection precision. As a result, it has become a foundational baseline in autonomous driving, industrial inspection, video surveillance, and other real-time vision systems. The YOLOv5 network structure is shown in Figure 1.

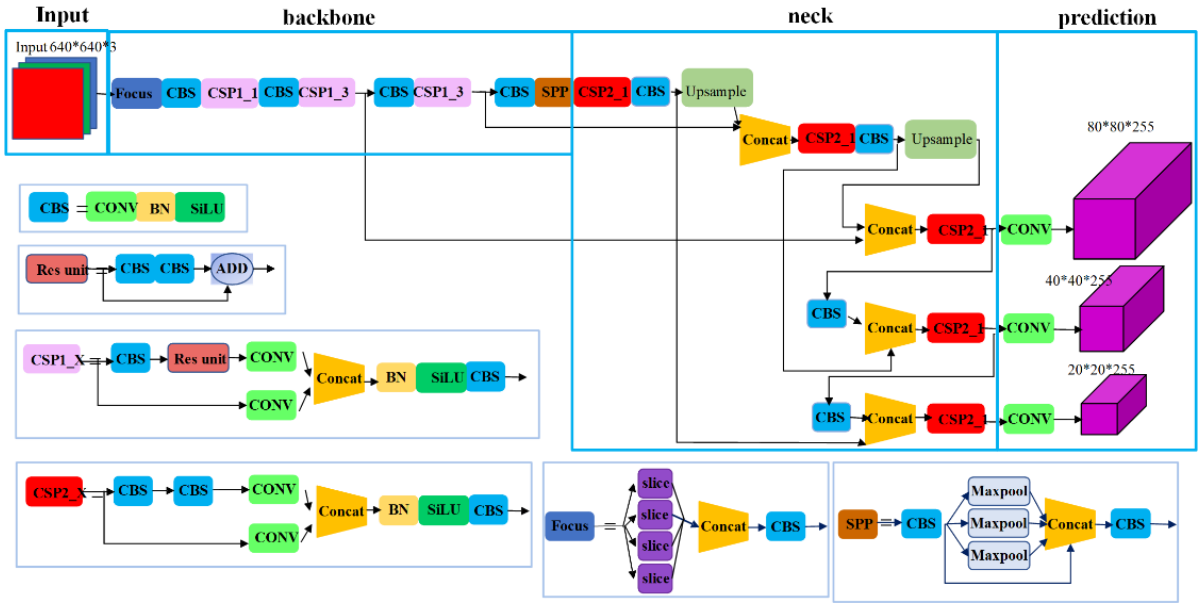

Figure 1. YOLOv5 network structure

3. OVERVIEW OF YOLOV8

YOLOv8, released by Ultralytics in 2023, represents a significant advancement in algorithmic design and engineering implementation within the YOLO series. As a fully Python-implemented framework, YOLOv8 systematically innovates across network architecture, training strategies, inference optimization, and model extensibility, while maintaining efficient detection performance. These improvements further enhance the model's generalization capabilities and deployment flexibility, establishing YOLOv8 as a highly performant and user-friendly detection baseline widely adopted in practical applications.

3.1. Network Architecture

YOLOv8 continues the three-stage design paradigm of Backbone–Neck–Head but introduces substantial optimizations at the modular level to strengthen feature representation [20] and detection accuracy.

3.1.1. Backbone

The Backbone of YOLOv8 adopts a newly designed lightweight structure, discarding the CSPDarknet backbone traditionally used in anchor-based designs. Instead, it introduces a customized Cross-Stage Partial Fusion (C2f) [21] module. Building on the strengths of CSP, the C2f module enables more flexible feature fusion, thereby improving parameter efficiency and feature extraction capabilities. Additionally, the Backbone employs the SiLU (Sigmoid Linear Unit) [22] activation function by default, maintaining strong nonlinear representation while enhancing training stability.

3.1.2. Neck

For feature aggregation, YOLOv8 retains a lightweight PAN-FPN (Path Aggregation Network combined with Feature Pyramid Network) [23] structure, optimizing feature transmission during upsampling and downsampling processes. By introducing improved feature fusion strategies, the Neck module of YOLOv8 more effectively integrates multi-scale semantic information, enhancing the detection of small objects and objects in complex backgrounds.

3.1.3. Head

One of the most notable changes in YOLOv8 is the introduction of an anchor-free detection head. Abandoning traditional anchor-based mechanisms, YOLOv8 adopts a direct end-to-end prediction approach, regressing center offsets, object dimensions, and class probabilities. This anchor-free design not only simplifies the training and inference pipelines but also improves adaptability across different object scales, significantly enhancing the detection of small objects and rare classes.

3.2. Key Techniques

YOLOv8 also incorporates a range of advanced techniques in training and inference optimization, leading to further improvements in overall model performance.

3.2.1. Data Augmentation

YOLOv8 refines its data augmentation strategies by retaining classic methods such as Mosaic and MixUp while introducing stricter image normalization and size consistency handling. The addition of the Copy-Paste augmentation technique, which pastes objects from one image into another, notably improves training outcomes for small-sample classes, enhancing the model's robustness.

3.2.2. Loss Function

YOLOv8 employs an improved Distribution Focal Loss (DFL) alongside Varifocal Loss to more precisely model regression errors and classification uncertainty. Particularly under the anchor-free

paradigm, the distributed regression strategy effectively mitigates class imbalance issues, accelerates convergence, and achieves higher precision in bounding box localization.

3.2.3. Training Strategies

YOLOv8 incorporates Exponential Moving Average (EMA) parameter smoothing and adopts optimized learning rate schedules such as Cosine Annealing with Warm Restarts, significantly improving training stability and final model accuracy. Furthermore, standardized Batch Normalization strategies and Automatic Mixed Precision (AMP) training are integrated to enhance training efficiency, especially on large-scale datasets.

3.3. Model Variants and Applications

To meet diverse computational and performance requirements, YOLOv8 offers multiple model variants, ranging from YOLOv8n (Nano), YOLOv8s (Small), YOLOv8m (Medium), YOLOv8l (Large), to YOLOv8x (Extra-large). Each variant balances parameters, inference speed, and detection accuracy differently, allowing for flexible deployment across mobile devices, edge computing platforms, and server environments. In practical applications, YOLOv8 consistently demonstrates superior performance across large-scale datasets such as COCO and Objects365. It particularly excels in small object detection, complex background recognition, and cross-domain transfer tasks, further solidifying its status as a leading next-generation general-purpose object detection framework. The network architecture of YOLOv8 is illustrated in Figure 2.

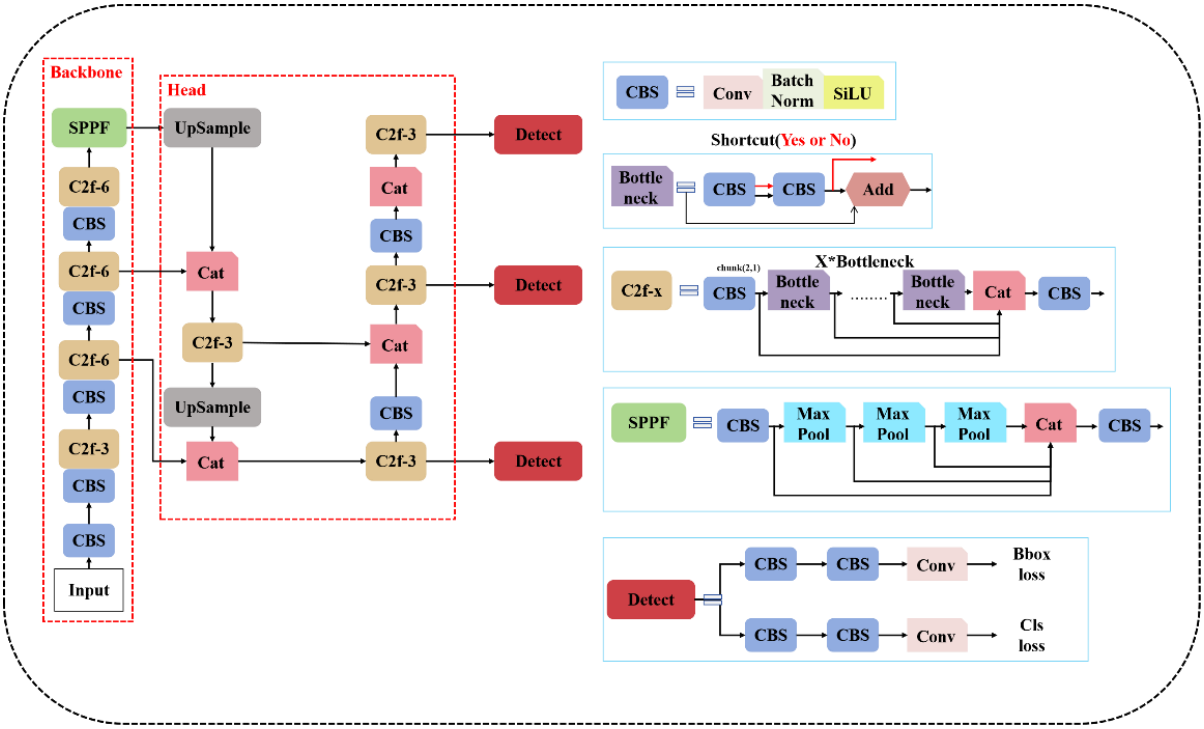

Figure 2. YOLOv8 network structure

4. COMPARATIVE ANALYSIS BETWEEN YOLOV5 AND YOLOV8

To better illustrate the evolutionary advancements in structural innovation and performance improvement throughout the YOLO series, this section systematically compares YOLOv5 and YOLOv8 across aspects such as network architecture, training mechanisms, inference optimization, and detection performance.

4.1. Network Architecture Comparison

Both YOLOv5 and YOLOv8 adopt the three-stage design of Backbone, Neck, and Head. However, substantial differences exist in module implementation and connectivity. YOLOv5 utilizes CSPDarknet in its Backbone, emphasizing cross-stage residual connections to reduce computational redundancy. In contrast, YOLOv8 replaces CSPDarknet with the more concise and efficient C2f module, achieving more flexible feature fusion and better gradient flow.

For the Neck, YOLOv5 primarily employs the standard PAN-FPN structure, whereas YOLOv8 builds upon PAN-FPN with further lightweight optimization and enhanced feature integration.

In the Head, the difference is particularly pronounced: YOLOv5 uses anchor-based detection heads requiring predefined anchor settings, whereas YOLOv8 adopts anchor-free detection, directly regressing object centers and dimensions. This shift significantly simplifies the detection pipeline and improves adaptability to small objects and long-tail categories.

4.2. Training Mechanism Comparison

In terms of training mechanisms, both YOLOv5 and YOLOv8 emphasize data augmentation and optimizer design. YOLOv5 uses Mosaic and MixUp for augmentation and applies the CIoU loss function for bounding box regression. YOLOv8 enhances these strategies by introducing Copy-Paste augmentation to better handle small-sample classes, while replacing the loss functions with Distribution Focal Loss and Varifocal Loss for finer modeling of regression errors and classification uncertainty.

Regarding optimization, YOLOv5 relies on SGD with cosine scheduling, while YOLOv8 generally employs EMA smoothing and mixed precision training to further improve training stability and efficiency.

4.3. Inference Optimization Comparison

During inference, YOLOv5 supports Automatic Mixed Precision (AMP) acceleration and model export via ONNX/TensorRT for deployment. Building upon this, YOLOv8 offers even greater flexibility and efficiency, with direct support for TensorRT and OpenVINO exports. Additionally, due to the anchor-free architectural changes, YOLOv8 greatly reduces preprocessing and postprocessing overhead during inference, resulting in improved inference speed and deployment convenience.

4.4. Detection Performance Comparison

In terms of detection performance, YOLOv8 consistently outperforms YOLOv5 on several public benchmarks such as COCO. Particularly in small object detection, complex background segmentation, and cross-category transfer tasks, YOLOv8 benefits from its anchor-free detection head and more sophisticated feature fusion strategies. Across equivalent model scales (e.g., YOLOv8n vs. YOLOv5n, YOLOv8s vs. YOLOv5s), YOLOv8 models achieve higher mAP scores while maintaining or reducing parameter counts and computational loads. Furthermore, YOLOv8 reduces inference latency, providing superior solutions for deployment on mobile and edge devices.

5. CONCLUSION AND FUTURE OUTLOOK

This paper provides a comprehensive comparison between YOLOv5 and YOLOv8 in terms of architectural design, training mechanisms, inference optimization, and detection performance. Compared to YOLOv5, YOLOv8 achieves significant improvements in both accuracy and speed by introducing the lightweight C2f module, an anchor-free detection head, advanced data augmentation

strategies and loss functions, as well as more efficient inference and deployment solutions. The comparison clearly illustrates the evolutionary trajectory of the YOLO series in structural innovation and performance enhancement, offering valuable insights for the field of object detection.

Looking ahead, there remains considerable room for further optimization within the YOLO series. Future developments are expected to focus on how to compress model size and improve inference efficiency without sacrificing detection accuracy, as well as how to incorporate emerging technologies such as Transformers and adaptive inference mechanisms. With ongoing advancements in hardware platforms and algorithmic theory, the YOLO series is poised to demonstrate even greater potential in real-time detection and resource-constrained deployment scenarios.

REFERENCES

[1] Zou Z, Chen K, Shi Z, et al. Object detection in 20 years: A survey [J]. Proceedings of the IEEE, 2023, 111(3): 257-276.

[2] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks [J]. Pattern recognition, 2018, 77: 354-377.

[3] Jiang P, Ergu D, Liu F, et al. A Review of Yolo algorithm developments [J]. Procedia computer science, 2022, 199: 1066-1073.

[4] Sang J, Wu Z, Guo P, et al. An improved YOLOv2 for vehicle detection [J]. Sensors, 2018, 18(12): 4272.

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

[6] Dewi C, Chen R C, Jiang X, et al. Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V4 [J]. Multimedia Tools and Applications, 2022, 81(26): 37821-37845.

[7] Zhang Y, Guo Z, Wu J, et al. Real-time vehicle detection based on improved yolo v5 [J]. Sustainability, 2022, 14(19): 12274.

[8] Sohan M, Sai Ram T, Rami Reddy C V. A review on yolov8 and its advancements [C]//International Conference on Data Intelligence and Cognitive Informatics. Springer, Singapore, 2024: 529-545.

[9] 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.

[10] Li H, Xiong P, Fan H, et al. Dfanet: Deep feature aggregation for real-time semantic segmentation [C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 9522-9531.

[11] Zhao J, Zhang Z, Ren J, et al. Dual Cross-Stage Partial Learning for Detecting Objects in Dehazed Images [C]//2024 IEEE International Conference on Data Mining (ICDM). IEEE, 2024: 629-638.

[12] Wang H, Zhang F, Wang L. Fruit classification model based on improved Darknet53 convolutional neural network [C]//2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS). IEEE, 2020: 881-884.

[13] Ma J, Jiang X, Fan A, et al. Image matching from handcrafted to deep features: A survey [J]. International Journal of Computer Vision, 2021, 129(1): 23-79.

[14] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759-8768.

[15] He J, Chen J N, Liu S, et al. Transfg: A transformer architecture for fine-grained recognition [C]//Proceedings of the AAAI conference on artificial intelligence. 2022, 36(1): 852-860.

[16] Qiu M, Huang L, Tang B H. Bridge detection method for HSRRSIs based on YOLOv5 with a decoupled head [J]. International Journal of Digital Earth, 2023, 16(1): 113-129.

[17] 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.

[18] Wang X, Song J. ICIoU: Improved loss based on complete intersection over union for bounding box regression [J]. IEEE Access, 2021, 9: 105686-105695.

[19] Zhou D, Fang J, Song X, et al. Iou loss for 2d/3d object detection [C]//2019 international conference on 3D vision (3DV). IEEE, 2019: 85-94.

[20] Xu Y, Mo T, Feng Q, et al. Deep learning of feature representation with multiple instance learning for medical image analysis [C]//2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2014: 1626-1630.

[21] Zhao J, Zhang Z, Ren J, et al. Dual Cross-Stage Partial Learning for Detecting Objects in Dehazed Images [C]//2024 IEEE International Conference on Data Mining (ICDM). IEEE, 2024: 629-638.

[22] Elfwing S, Uchibe E, Doya K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning [J]. Neural networks, 2018, 107: 3-11.

[23] Guo F, Wang Y, Qian Y. Real-time dense traffic detection using lightweight backbone and improved path aggregation feature pyramid network [J]. Journal of Industrial Information Integration, 2023, 31: 100427.