International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

YOLOv8: A Comprehensive Review of Architecture, Advancements, and Applications in Real-Time Object Detection

Kiran Sahu[1], Borra Rupa Sri[2], Trisha Kumari Sahu[3], Govindu Usha[4]

[1, 2, 3, 4]*Department of Computer Applications & Aditya University, Surampalem, Kakinada, A. P, India*

Abstract - Recent advancements in deep learning have significantly improved object detection systems, with the YOLO (You Only Look Once) family emerging as a dominant paradigm for real-time detection. YOLOv8, developed by Ultralytics, introduces architectural refinements such as anchor-free detection, decoupled heads, and multi-task capabilities. This paper presents a comprehensive review of YOLOv8, including its architecture, innovations, performance benchmarks, and applications. Furthermore, the study compares YOLOv8 with prior YOLO versions and other detection frameworks while highlighting limitations and future research directions.

Key Words: object detection, computer vision, YOLOv8, YOLO, deep learning, decoupled head, convolutional neural networks (CNNs), feature pyramid networks (FPN), image recognition, edge computing

1. INTRODUCTION

Object detection is a core problem in computer vision involving classification and localization of objects in images. Traditional methods such as R-CNN relied on region proposals, whereas YOLO reframed detection as a single-stage regression problem [1].

YOLOv8 represents the latest advancement in this lineage, offering improved efficiency, scalability, and usability for real-world deployment.

2. LITERATURE REVIEW

Object detection has been a central problem in computer vision, evolving significantly with the advent of deep learning. Early approaches relied on region-based convolutional neural networks such as **R-CNN**, **Fast R-CNN** [25], and **Faster R-CNN** [8], which introduced region proposal mechanisms for accurate localization. Although these methods achieved high accuracy, they suffered from high computational complexity, limiting their real-time applicability.

To address these limitations, single-stage detectors such as **SSD** [9] were proposed, enabling faster detection by eliminating the region proposal stage. A major breakthrough came with the introduction of **YOLO** by Joseph Redmon et al. [1], which reformulated object detection as a single regression problem. This approach significantly improved detection speed while maintaining competitive accuracy.

Subsequent versions of YOLO introduced progressive improvements. **YOLOv2** [2] incorporated batch normalization and anchor boxes, enhancing localization accuracy. **YOLOv3** [3] introduced multi-scale detection using feature pyramid structures inspired by **Feature Pyramid Network** [10]. Later, **YOLOv4** [4] and **YOLOv5** [5] integrated advanced training strategies, including **CSPNet** backbones, data augmentation techniques such as **MixUp** [33] and **CutMix** [34], and improved loss functions like **Complete IoU Loss** [35]. These advancements significantly improved both accuracy and efficiency.

The introduction of **YOLOv7** [6] further enhanced performance by optimizing training procedures and network architecture. Building upon this foundation, **YOLOv8** [7] represents the latest advancement, introducing anchor-free detection inspired by **FCOS** [15], decoupled detection heads, and multi-task learning capabilities. These innovations improve both detection accuracy and flexibility across various tasks such as segmentation and pose estimation.

Parallel to YOLO's evolution, several backbone architectures have contributed to improved feature extraction. Models such as **ResNet** [11], **DenseNet** [12], **AlexNet** [23], and **VGGNet** [24] have significantly influenced modern detection frameworks. Lightweight architectures like **ShuffleNet** [16] further enable deployment on edge devices.

Feature aggregation techniques have also evolved to enhance detection performance. Methods such as **PANet** [38] and **NAS-FPN** [30] improve multi-scale feature representation. Additionally, transformer-based models introduced by Ashish Vaswani et al. [13] have influenced recent object detection architectures by enabling global context modeling.

Anchor-free detection methods, including **CenterNet** [18] and **CornerNet** [17], have gained popularity due to their simplicity and improved localization accuracy. The **YOLOX** [19] further demonstrates the effectiveness of anchor-free approaches, influencing the design of YOLOv8.

Recent advancements also include improved detection frameworks such as **EfficientDet** [14], which balances accuracy and efficiency through compound scaling. Other models like **Cascade R-CNN** [28], **Hybrid Task Cascade** [29], and **Dynamic R-CNN** [31] focus on improving detection quality through multi-stage refinement.

Benchmark datasets have played a critical role in evaluating object detection models. Datasets such as **PASCAL VOC** [21], **MS COCO** [22], and **Cityscapes** [40] provide standardized evaluation frameworks for comparing model performance.

Recent studies [41]–[44] highlight the practical applications of YOLOv8 in real-time systems, including surveillance, autonomous driving, healthcare, and industrial automation. These works emphasize YOLOv8's superior trade-off between speed and accuracy, making it suitable for deployment in resource-constrained environments.

In summary, the literature demonstrates a clear progression from traditional region-based detectors to efficient real-time models, with the YOLO family leading this evolution. The integration of anchor-free detection, advanced feature extraction, and optimized training strategies in YOLOv8 represents a significant milestone in object detection research, while ongoing developments continue to push the boundaries of performance and applicability.

3. BACKGROUND AND EVOLUTION OF YOLO

The YOLO family has evolved as follows:

- YOLOv1 → Real-time detection framework [1]
- YOLOv2 → Batch normalization, anchor boxes [2]
- YOLOv3 → Multi-scale detection [3]
- YOLOv4/v5 → Improved training and CSPNet backbone [4][5]
- YOLOv7 → Enhanced efficiency and accuracy [6]
- YOLOv8 → Anchor-free, decoupled architecture [7]

YOLOv8 builds upon these advancements by simplifying design and improving detection performance.

4. YOLOv8 ARCHITECTURE

YOLOv8 follows a modular architecture consisting of three primary components:

4.1 Overall Architecture Diagram

4.2 Backbone

- Uses **CSP-based architecture (CSPDarknet variant)**.
- Extracts hierarchical features efficiently.
- Improves gradient flow and reduces redundancy.

4.3 Neck

- Uses **PAN-FPN or C2f module** for feature aggregation.
- Combines multi-scale features to improve small object detection.

4.4 Detection Head

- **Anchor-free mechanism** replaces anchor-based prediction.
- Uses **decoupled head**:
  - Classification branch
  - Regression branch

This improves convergence and detection accuracy.

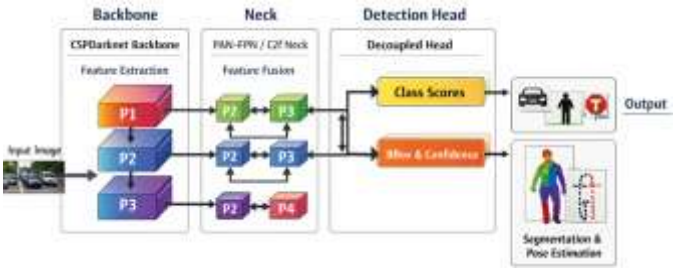

Fig-1: YOLOv8 Architecture

5. KEY INNOVATIONS IN YOLOv8

YOLOv8 introduces several architectural and training innovations that significantly enhance both **accuracy and efficiency** compared to earlier YOLO versions. These improvements make it highly suitable for real-time and multi-domain applications.

5.1 Anchor-Free Detection

Traditional object detection models, including earlier YOLO versions, rely on **anchor boxes**—predefined bounding boxes of various sizes and aspect ratios—to predict object locations. However, anchor-based methods introduce several challenges:

- Complex hyperparameter tuning (number, size, and ratios of anchors)
- Increased computational overhead
- Difficulty in generalizing across datasets with varying object scales

YOLOv8 adopts an **anchor-free detection mechanism**, where objects are detected by directly predicting their center points and bounding box dimensions.

Advantages:

- **Simplified training pipeline** (no need for anchor design)
- **Reduced computational complexity**
- **Improved generalization** across diverse datasets
- Better performance on **small and irregularly shaped objects**

This design aligns YOLOv8 with modern detectors such as FCOS and CenterNet, which have demonstrated strong performance without anchor dependency.

5.2 Decoupled Head Architecture

In earlier YOLO models, classification (what the object is) and localization (where the object is) were handled jointly in a **shared detection head**, which often led to suboptimal optimization.

YOLOv8 introduces a **decoupled head**, where:

- One branch is dedicated to **classification**
- Another branch focuses on **bounding box regression**

Benefits:

- Reduces conflict between classification and localization tasks
- Enables **task-specific feature learning**
- Improves **convergence speed during training**
- Enhances overall detection accuracy

This separation allows the network to independently optimize each objective, leading to better performance, especially in complex scenes.

5.3 Multi-Task Learning Capability

A major advancement in YOLOv8 is its **unified multi-task framework**, enabling it to perform multiple vision tasks within a single architecture.

Supported Tasks:

- **Object Detection** → Identifying and localizing objects
- **Instance Segmentation** → Pixel-level object boundaries
- **Image Classification** → Assigning labels to images
- **Pose Estimation** → Detecting human keypoints

Advantages:

- Reduces the need for separate models for each task
- Enables **shared feature representations**, improving efficiency
- Simplifies deployment in real-world systems
- Supports **transfer learning across tasks**

This multi-task capability makes YOLOv8 highly versatile for applications such as autonomous systems, healthcare imaging, and surveillance.

5.4 Advanced Training Techniques

YOLOv8 integrates several modern training strategies to improve robustness and generalization.

1. Data Augmentation Techniques

- **Mosaic Augmentation**: Combines multiple images into one, improving detection of small objects and enhancing context diversity
- **MixUp Augmentation**: Blends images and labels, reducing overfitting and improving generalization

2. Improved Loss Functions

YOLOv8 employs advanced **IoU-based loss functions** (e.g., CIoU, DIoU), which consider:

- Overlap area
- Distance between box centers
- Aspect ratio consistency

Benefits:

- More precise bounding box regression
- Faster convergence
- Improved localization accuracy

3. Optimization Enhancements

- Better learning rate scheduling
- Efficient label assignment strategies
- Improved normalization techniques

Together, these techniques significantly enhance the model's **robustness, accuracy, and training efficiency**.

6. PERFORMANCE EVALUATION

YOLOv8 has been extensively evaluated on standard benchmarks such as the **COCO dataset**, demonstrating strong performance in terms of **mean Average Precision (mAP)** and inference speed (FPS).

6.1 Model Variants and Performance

Table-1: YOLOv8 Model Variants and Performance

Model	mAP	Speed	Use Case
YOLOv8n (Nano)	Moderate	Very Fast	Edge devices, mobile applications
YOLOv8s (Small)	Good	Fast	Real-time applications
YOLOv8m (Medium)	High	Moderate	Balanced performance
YOLOv8l/x (Large/Extra)	Very High	Slower	High-accuracy tasks

6.2 Performance Analysis

1. Speed vs Accuracy Trade-off

YOLOv8 provides a flexible range of models:

- **Lightweight models (v8n, v8s)** prioritize speed and are suitable for **edge devices**.
- **Larger models (v8l, v8x)** prioritize accuracy for **high-performance computing environments**.

This scalability allows practitioners to choose models based on **hardware constraints and application requirements**.

2. Benchmark Performance (COCO Dataset)

YOLOv8 demonstrates:

- High **mAP scores**, indicating strong detection accuracy.
- Competitive **FPS**, enabling real-time processing.
- Improved **small object detection** compared to earlier YOLO versions.

3. Comparison with Previous YOLO Versions

Compared to YOLOv5 and YOLOv7, YOLOv8 shows:

- Better **generalization capability** due to anchor-free design.
- Improved **training stability**.
- Enhanced **multi-task support**.
- More efficient **feature extraction and representation**.

4. Real-World Performance

YOLOv8 performs well in:

- Dynamic environments (e.g., traffic scenes).
- Low-latency systems (e.g., surveillance).
- Resource-constrained devices (e.g., mobile, IoT).

6.3 Key Observations

- Achieves an excellent **balance between speed and accuracy**.
- Scales efficiently across different hardware platforms.
- Outperforms earlier YOLO models in many real-time detection tasks.

Suitable for both **research and industrial applications**.

7. APPLICATIONS

7.1 Autonomous Driving

YOLOv8 plays a critical role in modern intelligent transportation systems by enabling **real-time object detection with low latency**, which is essential for safety-critical decisions. It is widely used for detecting **pedestrians, vehicles, traffic signs, and lane boundaries**. Its anchor-free architecture improves detection accuracy for small and fast-moving objects, which are common in urban driving scenarios. Additionally, YOLOv8 can be integrated with sensor fusion frameworks combining LiDAR, radar, and camera inputs to enhance robustness under challenging conditions such as fog, rain, or night driving.

7.2 Healthcare

In healthcare, YOLOv8 is increasingly adopted for **medical image analysis**, particularly in detecting abnormalities such as tumors, lesions, and fractures. Its high precision and speed allow for **real-time diagnostic assistance** in radiology workflows. YOLOv8 is also used in applications such as:

- Cancer detection in MRI and CT scans
- Cell counting in microscopic images
- Surgical instrument detection in robotic surgery

The model's ability to generalize across datasets makes it valuable in **AI-assisted diagnosis**, though domain-specific training is often required.

7.3 Surveillance and Security

YOLOv8 enables **intelligent video surveillance systems** by detecting suspicious activities, unauthorized access, and crowd anomalies in real time. It supports:

- Face and person detection
- Intrusion detection in restricted areas
- Behavior analysis in public spaces

Its fast inference speed allows deployment in **edge-based surveillance systems**, reducing dependency on centralized servers and improving privacy.

7.4 Agriculture

In precision agriculture, YOLOv8 is used for **crop monitoring and disease detection**, helping farmers improve productivity and reduce losses. Applications include:

- Detection of plant diseases from leaf images.
- Pest identification and classification.
- Fruit counting and yield estimation.

By integrating with drones and IoT devices, YOLOv8 enables **automated large-scale field monitoring**, supporting sustainable farming practices.

7.5 Industrial Automation

YOLOv8 is widely applied in **smart manufacturing environments** for quality control and defect detection. Its real-time capabilities enable:

- Detection of surface defects in products
- Assembly line monitoring
- Object tracking in robotic systems

It improves efficiency by reducing human intervention and ensuring consistent quality standards in production lines.

8. COMPARISON WITH OTHER MODELS

Table-2: Comparative Analysis of Object Detection Models

Model	Type	Speed	Accuracy	Key

				Characteristics
Faster R-CNN	Two-stage	Low	Very High	Region proposal-based, high precision but slow
SSD	One-stage	Medium	Medium	Faster than R-CNN, struggles with small objects
YOLOv7	One-stage	High	High	Optimized architecture, strong performance
YOLOv8	One-stage	Very High	Very High	Anchor-free, efficient, multi-task support

Discussion

Compared to traditional two-stage detectors like Faster R-CNN, YOLOv8 significantly reduces computational overhead by eliminating the region proposal stage, resulting in **faster inference speeds**. While SSD offers a balance between speed and accuracy, it often underperforms in detecting small objects.

YOLOv7 introduced several architectural optimizations, but YOLOv8 further enhances performance through:

- Anchor-free detection mechanism
- Improved feature pyramid networks
- Better loss functions and training strategies

Overall, YOLOv8 achieves a **superior trade-off between speed and accuracy**, making it suitable for real-time applications across various domains.

9. LIMITATIONS

Despite its advantages, YOLOv8 has several limitations:

1. **Dependency on Large Labeled Datasets:** High performance requires extensive annotated datasets, which are costly and time-consuming to create.
2. **Performance in Dense and Occluded Scenes:** Detection accuracy decreases in crowded environments where objects overlap significantly.
3. **Computational Complexity for Larger Models:** Although optimized, larger YOLOv8 variants still require high GPU resources, limiting deployment on low-power devices.
4. **Limited Interpretability:** Like most deep learning models, YOLOv8 operates as a "black box," making it difficult to interpret decision-making processes.
5. **Generalization Issues:** Performance may degrade when applied to unseen domains without proper fine-tuning.

10. FUTURE RESEARCH DIRECTIONS

1. **Transformer-Based YOLO Architectures:** Integrating transformer mechanisms can improve global context understanding and long-range dependencies.
2. **Lightweight Models for Edge Deployment:** Developing compact versions of YOLOv8 for mobile and embedded systems is essential for real-time edge applications.
3. **Explainable AI (XAI) Integration:** Enhancing model transparency through explainability techniques will increase trust in critical applications such as healthcare.
4. **Open-Vocabulary Object Detection:** Future models should detect unseen object classes using natural language descriptions, improving adaptability.
5. **3D Object Detection and Multi-Modal Learning:** Extending YOLOv8 to 3D environments using LiDAR and depth data will enhance performance in robotics and autonomous systems.
6. **Self-Supervised and Few-Shot Learning:** Reducing dependency on labeled data by leveraging unlabeled datasets is a promising direction.

11. CONCLUSION

YOLOv8 represents a major step forward in real-time object detection, combining efficiency, flexibility, and high accuracy. Its anchor-free design and multi-task capabilities make it suitable for a wide range of applications. Despite limitations, YOLOv8 sets a strong foundation for future research in computer vision.

REFERENCES

1. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. CVPR*, 2016.
2. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proc. CVPR*, 2017.
3. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.
4. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020.
5. G. Jocher et al., "YOLOv5," Ultralytics, 2020.
6. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies," 2022.
7. Ultralytics, "YOLOv8: Next-Generation Object Detection," 2023.
8. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN," in *Proc. NeurIPS*, 2015.
9. W. Liu et al., "SSD: Single Shot MultiBox Detector," in *Proc. ECCV*, 2016.
10. T.-Y. Lin et al., "Feature Pyramid Networks for Object Detection," in *Proc. CVPR*, 2017.
11. K. He et al., "Deep Residual Learning for Image Recognition," in *Proc. CVPR*, 2016.
12. G. Huang et al., "Densely Connected Convolutional Networks," in *Proc. CVPR*, 2017.
13. A. Vaswani et al., "Attention Is All You Need," in *Proc. NeurIPS*, 2017.
14. M. Tan et al., "EfficientDet: Scalable and Efficient Object Detection," in *Proc. CVPR*, 2020.
15. Z. Tian et al., "FCOS: Fully Convolutional One-Stage Object Detection," in *Proc. ICCV*, 2019.
16. X. Zhang et al., "ShuffleNet: Efficient CNN for Mobile Devices," in *Proc. CVPR*, 2018.
17. H. Law and J. Deng, "CornerNet," in *Proc. ECCV*, 2018.
18. X. Zhou et al., "CenterNet," in *Proc. ICCV*, 2019.
19. Z. Ge et al., "YOLOX: Exceeding YOLO Series in 2021," 2021.
20. C. Wang et al., "Scaled-YOLOv4," in *Proc. CVPR*, 2021.
21. M. Everingham et al., "The PASCAL VOC Challenge," *IJCV*, 2010.
22. T.-Y. Lin et al., "Microsoft COCO Dataset," in *Proc. ECCV*, 2014.
23. A. Krizhevsky et al., "ImageNet Classification with Deep CNNs," in *Proc. NeurIPS*, 2012.
24. K. Simonyan and A. Zisserman, "Very Deep CNNs (VGG)," 2015.
25. R. Girshick, "Fast R-CNN," in *Proc. ICCV*, 2015.
26. J. Dai et al., "R-FCN," in *Proc. NeurIPS*, 2016.
27. S. Zhang et al., "RefineDet," in *Proc. CVPR*, 2018.
28. Z. Cai and N. Vasconcelos, "Cascade R-CNN," in *Proc. CVPR*, 2018.
29. K. Chen et al., "Hybrid Task Cascade," in *Proc. CVPR*, 2019.
30. X. Dai et al., "NAS-FPN," in *Proc. CVPR*, 2019.
31. Z. Cai et al., "Dynamic R-CNN," in *Proc. ECCV*, 2020.
32. Y. Li et al., "Generalized Focal Loss," in *Proc. NeurIPS*, 2020.
33. H. Zhang et al., "mixup: Beyond Empirical Risk Minimization," 2018.

34. T. DeVries and G. Taylor, “CutMix,” in *Proc. ICCV*, 2019.
35. Z. Zheng et al., “CIoU Loss,” in *Proc. AAAI*, 2020.
36. R. Girshick et al., “Mask R-CNN,” in *Proc. ICCV*, 2017.
37. K. He et al., “Spatial Pyramid Pooling,” in *Proc. ECCV*, 2014.
38. S. Liu et al., “Path Aggregation Network,” in *Proc. CVPR*, 2018.
39. Y. Chen et al., “DeepLab,” in *Proc. ECCV*, 2018.
40. M. Cordts et al., “Cityscapes Dataset,” in *Proc. CVPR*, 2016.
41. Y. Yaseen, “YOLOv8 Analysis,” 2024.
42. D. Reis et al., “Real-Time YOLOv8 Detection,” 2023.
43. R. Sapkota et al., “Evolution of YOLO Models,” 2025.
44. Ultralytics Blog, “YOLOv8 Insights,” 2023.

BIOGRAPHIES (Optional not mandatory)

1’st Author Photo

Description about the author1 (in 5-6 lines)