Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Enhancing YOLOv8n for Improved Small Object Detection on Custom Datasets

Narimane Wafaa Krolkral[1], Kamel Mohamed Faraoun[2], Chahreddine Medjahed[3]
[1]EEDIS Laboratory, Djillali Liabes University, Algeria
[2]Computer Science Department, EEDIS Laboratory, Djillali Liabes University, Algeria
[3]Computer Science Department, Hassiba Benbouali Chlef University, Algeria

ARTICLE INFO

Received: 24 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Deep learning has achieved remarkable performance in object detection, with YOLO (You Only Look Once) standing out for its speed and accuracy. In this paper, we present an improved detection model based on the YOLOv8 architecture, evaluated on two large-scale datasets. Our method introduces a new detection scale (P2), enhancing small object detection by capturing finer features. Additional modifications include advanced upsampling, feature concatenation, and the integration of the C2f module into the model's head, improving multi-scale fusion and overall accuracy. On a single-class dataset (9,215 images), our model achieves a mAP at 50 of 98.9% from scratch and 98.5% with pre-training, with precision and recall up to 97% and 96.2%, respectively. On a multi-class dataset with seven categories, it reaches a mAP at 50:95 of 78.1% with pre-training, and up to 94.5% precision and 89.6% recall. The model regularly surpasses YOLOv8n, YOLOv10n, and YOLOv11n across both datasets, exhibiting notable accuracy, robustness, and scalability, with a computational cost of 12.6 GFLOPs.

Keywords: Object Detection; Deep Learning; YOLOv8n; Ultralytics YOLO.

INTRODUCTION

Object detection, one of the essential computer vision tasks, helps to find and describe the exact places of the objects in the images with the help of automated feature extraction. Its rapid development is led by AI, in turn, it is responsible for various crucial activities like self-driving cars, environment monitoring, or real-time decision making in robotics, and the field of geospatial analysis, among others. Deep learning has brought about a renaissance in the field of object detection whereby convolutional neural networks (CNNs) have taken over from the traditional feature engineering method [14]. Leading structures like Fast R-CNN [1], Single Shot MultiBox Detector (SSD) [13], and the YOLO series [15] amalgamate feature extraction, localization, and classification to end-to-end learning frameworks. YOLO is particularly prominent with its one-stage structure, which allows for instant inferencing through a single pass. YOLOv8 [25] is an evolution of the previous versions of YOLO and has been improved through a more sophisticated reorganization of the architecture, which makes it more precise in detecting various objects. However, some issues are difficult to solve, such as the elimination process of instances because of occlusions, difficulties in identifying very small targets, and resource-constrained environments used for inference. The performance of YOLOv8 is still unsurpassed with respect to benchmarks like Pascal VOC for object classification and MS COCO for detection in complex scenes. Besides CSPDarknet in YOLOv4 and YOLOv5, the underlying structure has gone through a change to accommodate of zero-step attention generation. The optimizations have cut the weight of parameters and floating-point operations (FLOPs) without sacrificing fast inference time and enabled the model to be more accurate. These advancements are what propel YOLOv8 to an efficient, promising real-time object detection system.

Our research introduces an advanced object identification model utilizing the YOLOv8 framework, emphasizing the enhancement of performance in critical metrics including accuracy, recall, and mean average precision (mAP) at an IoU threshold of 0.5. The alterations to the YOLOv8 design aim to enhance the identification of small objects, a continual problem in real-world applications. Significant improvements comprise the introduction of a new high-resolution detection scale (P2) and enhanced feature integration in the neck and head components of the network. The P2 scale facilitates the acquisition of intricate features essential for small object detection, while the reconfigured neck utilizes bidirectional connections and variable weighting to improve multi-scale feature integration. Collectively, these enhancements result in heightened accuracy, resilience, and versatility across many visual scenarios.

RELATED WORK

The object detection task has been boosted considerably with the help of the progress made in deep neural networks, which resulted in better precision in recognizing and localizing objects in numerous images. The approaches to the detection of the object are mainly of two kinds. The former one contains two-stage detectors that have Region Proposal Networks (RPNs) to output candidate regions that, in turn, are use for classification and bounding box regression. The most famous models following this pattern are R-CNN [2], Faster R-CNN [16], Mask R-CNN [3], and SPP-Net [4]. These models are usually very good when it comes to accuracy, especially in situations that are quite hard. At the same time, their multi-stage processing and largely increased computing demand limit the potential use of these models in real-time applications and in resource-scarce devices, especially in those cases in which the detection targets are small or occluded. The second group includes one-stage detectors like those in the YOLO sfamily [21], which do not need proposals of regions as they predict both bounding boxes and class scores in one go. The advantage of this method is the quick performance of the algorithm while maintaining the accuracy and the efficiency. YOLO models can be a perfect fit for real-time applications and are often widely observed in the fields of autonomous driving [17], robotics [27], and defense systems [19].

YOLOv8 is an upgrading of the YOLO architecture that has outperformed the previous state of the art across multiple object detection benchmarks. It includes models in five sizes YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x tailored to different performance levels and computational resources. The YOLOv8 model is visualized in the manner explained in [30]. New studies have shone a light on the YOLOv8 framework by carrying out affecting research. Li and Jia [9] improved multi-class military target detection by integrating PP-LCNet and the ParNet attention module, boosting precision and real-time performance. Wu et al. [23] presented Adaptive Kernel Convolution (AKConv), Multi-Scale Dilated Attention (MSDA), and a Wise-IoU loss function, enhancing feature representation and precision. Khalili et al. [7] improved feature fusion, introduced a fourth detection scale, and substituted CIoU with PIoU loss, therefore enhancing small-object recognition with negligible computational overhead. Zhou et al. [29] enhanced YOLOv8 by adding SimAM attention to the neck, employing the C2fDCN module for adaptive feature fusion, and adopting Dynamic Head (DyHead) for scale-aware detection. Wu et al. [24] restructured the model by replacing large-scale layers with smaller ones and integrating recursive gated convolutions alongside multiple CBAM modules to improve aerial image analysis. Zeng et al. [26] introduced a C4 feature extraction block and a DyHead-based detection head tailored for small-object detection. Liang et al. [11] incorporated Efficient Channel Attention, substituted PANet with BiFPN, and applied EIoU loss to boost accuracy in dense crowd scenes. Seth and Sivagami. [18] utilized image pre-processing techniques, including histogram equalization, gamma correction, and contrast stretching, to increase data diversity and enhance model generalization.

YOLOv8 has become more advanced than its predecessors primarily with the up-gradation of the backbone and neural network head while decreasing the model's complexity. It uses the CSPDarknet backbone from YOLOv5 [22], but it also has increased efficiency with the help of such optimizations as depth-wise separable convolutions and optimized CSP Modules [5]. The model replaces legacy IoU losses like GIoU [10] and DIoU [28] by the better and more robust EIoU and SIoU losses for improved convergence and bounding box accuracy. Besides, the old spatial pyramid-pooling module (spp) was instead changed to the faster SPP-Fast [6] for the purpose of better multi-scale context capturing. Therefore, the feature aggregation is also optimized through bidirectional connections [12] and weighted fusion [20], which in turn leads to spatial detail and semantic information balance. Thus, YOLOv8 is a system that is accurate to a higher degree, much faster when making an inference and that can be deployed on any hardware platform, with no limits of the research aimed at furthering the improvement in the loss functions, attention mechanisms, and multi-scale fusion.

PROPOSED APPROACH

The original YOLOv8 model encountered difficulties in accuracy, particularly regarding small object recognition, and required speed enhancements for real-time application. The constraints affected the YOLOv8n version variant in industrial and professionals applications. In reply, certain architectural changes were implemented in YOLOv8n, enhancing both precision and efficiency.

Key modifications implemented in YOLOv8n include the introduction of a brand new detection scale (P2) along with a carefully designed feature concatenation and processing strategy. These changes significantly impact the neck and head of the model and thus deliver the message that the model can now handle a variety of features that exist at different scales in a better and more efficient manner while the information is being moved between the network layers in a very clear and effective way. Exploiting the P2 scale by the fine-tuned model of the network, the model acquires the detection capability of smaller objects with substantially higher accuracy. Consequently, the more robust local and contextual understanding is ensured because of the improved feature-processing mechanism. These breakthroughs are in concord with the current trend in multi-scale feature processing paradigms, in other words, that higher rates of the fusion of more powerful multi-scale features lead to the most accurate signals and better overall detection performance. On the other hand, as a whole, these enhancements foster more powerful multi-scale feature fusion, yielding superior detection accuracy and overall model robustness.

P2 Detection Scale

Incorporating the P2 detection scale is a changing improvement for the personalized YOLOv8 architecture. While the original YOLOv8 model has P3, P4, and P5 scales which are good enough for medium to large objects. Unfortunately, it is an impossible task for these scales to detect the small objects as their feature map quality is inferior and they are incapable of capturing the finer details of the object at the same time. This issue is solved by the P2 scale, which makes resolution and accuracy better for small objects. The P2 scale, which has higher spatial precision (P2/4) for learning finer-grained features, was added to the model to make it better at identifying small objects. enabling them to detect small objects that the original YOLOv8 scales often missed. The P2 scale was introduced to preserve important features such as textures, edges, and fine structures, which are often lost at higher levels like P4 or P5.

The P2 scale enhances small object detection and overall identification accuracy by preserving greater spatial detail. Moreover, the P2 scale not only maintains the features but also accelerates the gradient propagation during training. Reducing the distance between supervised layers and early backbone layers in the FPN enhances supervision signals, allowing lower layers to acquire more discriminative characteristics, which is particularly advantageous for small object identification. To enable optimal detection across multiple scales and object sizes, our custom YOLOv8 architecture adopts a comprehensive feature fusion strategy. This involves adding multiple Upsample, Concat, and C2f layers within the model head, enhancing the capture and utilization of multi-level features. These components are defined as follows:

- **Upsample Layers:** Increase the spatial resolution of feature maps, enabling the network to preserve and exploit fine-grained details at different scales.
- **Concat Layers:** Concatenate features from multiple detection scales, facilitating effective multi-scale object detection by combining contextual information from varying resolutions.
- **C2f Layers:** Advanced feature filtering modules that refine concatenated features through repeated convolutional operations, improving feature representation quality.

The integration of the P2 scale and improved feature fusion enhances object detection, especially for small targets, increasing accuracy and adaptability across varied scenarios.

Multi-Scale Feature Fusion

The utilization of the augmented YOLOv8's multi-scale feature fusion greatly aids the feature integration across the detection scales, specifically the quality improvement of small object detection. The contribution is hierarchical feature fusion manner, which firstly enriches features of different scales and channels through the use of such operations as upsampling, concatenation, and C2f processing.

- **P4 to P5 Fusion:** P4 features are upscaled and fusioned with the P5 layer features producing a more informative map, which is further enhanced by the C2f layers through an iterative application of the convolution operation for another round of feature extraction.

- **P3 to P4 Fusion:** The P3 features are firstly upsampled and then they are merged with the processed P4 features, and then the C2f layers are activated for retrieving some of the information by increasing the feature space and filling in the missing features in the mid-scale object detectors.
- **P2 to P3 Fusion:** The P2 features are reshaped to a bigger size to create P3 features, and then combined with the fused P3 features and processed with C2f layers to get the best results from small to mid scale object detection.

The new P2 detection scale adds to the small object detection ability by blending upsampled P2 features with P3 features, enhanced through C2f layers. This mix benefits length and cross-resolution multi-scale feature improvement, as in the new texture details are captured vastly. The utilization of this feature combination by the model broadens spatial as well as contextual awareness, providing the possibility for more precise detection of either small and complex objects or those located amidst clusters of small targets in various scenarios. Joining the P2 configuration and advanced fusion layers to YOLOv8 not only enhances the model's capability to find small objects but also permits it to get back fine details that were lost at lower resolutions.

Architectural Refinements and Parameter Optimization

For YOLOv8, its depth, width, and channel capacity were optimized to come up with a network that has a good trade-off between the accuracy and the computational cost. The new YOLOv8n network can in one side perform the task intended for its complexity, on the other hand, allow for application in various fields with the varied computing capability, and object scales. The proposed method enhances YOLOv8 by adding a P2 detection scale, improving multi-scale fusion, and optimizing parameters, resulting in better small object detection and robustness without losing real-time performance.

EXPERIMENTS AND OBTAINED RESULT

This section presents the experimental setup, including dataset preprocessing, training configurations, and environment. It uses quantitative metrics like mAP, recall, and precision, along with qualitative assessments, to evaluate model performance and reproducibility.

Datasets description

The model was trained and tested on two datasets, with the first being a set of 4,000 person-class images precisely annotated from Freepik and Pexels. Stratified sampling was used to ensure the dataset was well balanced for training, validation, and testing. A standardized preprocessing pipeline was used for orientation and resizing to 640×640 pixels as well as stochastic augmentations, which added to the diversity and to the robustness of the model. Using the high quality, augmented images that YOLOv8's detection accuracy could be improved, overfitting reduced, and the performance in real-world applications improved.

In the second dataset from DreamsTime, there were 1,882 images grouped into the seven classes that were divided into the training, validation, and testing sets. We employed Roboflow to multiply the training set images through augmentations such as pivoting, color modifications, and mirroring that indeed tripled their number to 4,512, which benefited dataset diversity as well as generalization and detection robustness of the model.

Evaluation Metrics

The model's performance was evaluated using established metrics that quantify predictive accuracy and effectiveness across different scenarios.

$$P_{precision} = \frac{TP}{TP+FP} \tag{1}$$

$$R_{recall} = \frac{TP}{TP+FN} \tag{2}$$

$$AP = \int_0^1 P(R)dR \tag{3}$$

$$mAP = \frac{1}{C} \sum_{c\in C} AP(c) \qquad (4)$$

Where:

- **TP (True Positives):** Correctly identified objects belonging to the target class.
- **FP (False Positives):** Incorrectly predicted objects that do not belong to the target class.
- **FN (False Negatives):** Objects from the target class that were not detected.
- **Precision:** The fraction of relevant instances among the retrieved instances.
- **Recall**: The fraction of relevant instances that were successfully retrieved.
- **AP (Average Precision):** The area under the precision-recall curve for one class.
- **mAP (mean Average Precision):** The mean of AP across all classes.

Precision, recall, AP, and mAP are key metrics for evaluating detection accuracy, with mAP providing an overall performance measure. Computational efficiency is assessed using FPS, model parameters, and inference cost, all critical for real-time and resource-constrained deployment. Recall represents the proportion of correctly identified positive samples [8].

Results and Analysis

Experiments were conducted using Kaggle Notebooks, leveraging GPU acceleration for training and evaluation. The model backbone is YOLOv8, implemented in PyTorch for flexibility and performance. Two variants were compared: YOLOv8n (baseline) and an improved version with architectural enhancements. Both were trained under identical settings, detailed in Table 1, to ensure fair evaluation.

Table 1. Training Configuration Parameters

Parameter	Value
Depth-multiple	0.33
Width-multiple	0.25
Input Size	640x640
Epochs	100
Batch Size	16-32-64

Results on Dataset 1

Pretrained YOLO Models

Table 2 presents the performance of the YOLOv8n, Improved YOLOv8n, YOLOv10n, and YOLOv11n models in terms of computational complexity, accuracy, and execution time, evaluated with a differents batch size on Dataset 1.

Table 2. Performance Comparison of Pretrained YOLO Models on Dataset 1

	Yolov8n			**Improved Yolov8n**			**Yolov10n**			**Yolov11n**		
Batch	16	32	64	16	32	64	16	32	64	16	32	64
GFLOPs		8.2			12.6			8.4			6.4	
Precision (%)	97.2	96.8	96.4	96.8	**97.0**	96.8	94.6	94.8	93.6	96.8	96.0	96.9
Recall (%)	94.2	95.2	95.5	**96.2**	**95.3**	**95.7**	91.0	91.0	93.6	94.9	94.8	95.3
mAP 50 (%)	98.2	98.3	98.5	**98.9**	**98.8**	**98.8**	97.1	97.1	97.4	98.2	98.1	98.1
Map 50-95 (%)	86.2	86.4	86.3	**87.5**	86.3	**86.5**	83.9	83.9	83.9	86.0	86.1	86.3
Execution Time (H)	5.307	5.655	5.390	5.688	**4.350**	**4.480**	5.790	5.811	5.833	5.535	5.805	5.722

Across all batch sizes (16, 32, and 64), the Improved YOLOv8n model consistently delivers the best overall performance, achieving the highest mAP at 50 (98.9%–98.8%) and mAP at 50–95 (87.5%–86.5%), as well as strong recall values (up to 95.7%). Notably, at batch size 32, it combines top-tier accuracy with the fastest training time (4.350 hours), indicating excellent architectural optimization for efficiency and learning. The baseline YOLOv8n also performs robustly, especially at batch size 16 with the highest precision (97.2%) and strong mAP scores, making it well suited for real-time systems prioritizing precision and speed. In contrast, YOLOv10n, despite moderate GFLOPs, consistently underperforms with the lowest recall and mAP scores across all batches, suggesting it is less suitable for precision-critical tasks. Meanwhile, YOLOv11n, the model with the lowest computational complexity (6.4 GFLOPs), achieves good precision (up to 96.9%) but lags behind in mAP and training efficiency, indicating limited gains from its increased depth. Overall, Improved YOLOv8n emerges as the most effective model, balancing accuracy, computational demand, and training speed across batch sizes.

Training from Scratch

To evaluate the models' ability to learn from the ground up, we retrained each one from scratch (random initialization of weights), analyzing convergence, accuracy, and efficiency without the benefit of pretraining.

In table 3. Across all batch sizes, Improved YOLOv8n consistently outperforms other models, delivering the best balance of precision, recall, and mAP scores, while maintaining competitive or even shorter training times. YOLOv8n performs well as a baseline with strong accuracy and low computational cost. In contrast, YOLOv10n shows the weakest results in recall and mAP across all settings, indicating poor generalization. YOLOv11n achieves high precision but underperforms in recall and mAP 50–95, limiting its overall effectiveness.

Table 3. Performance Comparison of YOLO Models from Scratch on Dataset 1

	Yolov8n			**Improved Yolov8n**			**Yolov10n**			**Yolov11n**		
Batch	16	32	64	16	32	64	16	32	64	16	32	64
GFLOPs		8.2			12.6			8.4			6.4	
Precision (%)	95.6	95.8	96.9	**96.9**	96.8	95.8	93.6	95.0	93.4	96.7	97.2	96.1
Recall (%)	93.5	92.8	92.2	**93.7**	**94.5**	**94.2**	90.6	90.2	90.8	91.6	92.8	91.1
mAP 50 (%)	97.4	97.3	97.1	**98.5**	**98.1**	**98.2**	96.0	96.1	97.8	97.3	97.2	97.0
mAP50-95 (%)	83.1	83.0	82.6	**83.1**	**83.5**	**83.4**	78.7	78.5	79.6	81.0	81.9	80.3
Execution Time (H)	5.180	5.370	5.685	5.520	6.102	**5.000**	5.684	5.898	5.757	5.590	6.080	5.820

Therefore, the Improved YOLOv8n consistently demonstrates the most favorable balance between detection accuracy, convergence speed, and generalization ability, making it the most robust and practical choice among the models evaluated.

Results on Dataset 2

Pretrained YOLO Models

As shown in Table 4, the Improved YOLOv8n consistently demonstrates the best overall detection performance, achieving the highest recall and mAP at 50, though with increased computational complexity and training time.

YOLOv8n stands out for its efficient balance between precision, accuracy, and speed, making it a strong baseline model. YOLOv10n is the most lightweight and efficient in terms of GFLOPs and runtime, but it suffers from lower accuracy across metrics. YOLOv11n, despite having the highest precision in some configurations, fails to deliver corresponding gains in recall and mAP, suggesting that its added complexity does not translate into better generalization.

Table 4. Performance Comparison of YOLO Models from Scratch on Dataset 2

	Yolov8n			**Improved Yolov8n**			**Yolov10n**			**Yolov11n**		
Batch	16	32	64	16	32	64	16	32	64	16	32	64
GFLOPs		8.2			12.6			8.4			6.4	
Precision (%)	92.9	91.8	92.4	91.7	**93.4**	**94.5**	89.5	93.3	92.8	95.9	88.4	89.4
Recall (%)	87.8	85.8	87.6	**88.9**	**88.3**	**89.6**	80.4	87.7	86.1	82.3	84.5	88.9
mAP 50 (%)	92.5	92.9	90.8	**93.2**	**93.5**	**91.9**	87.5	91.3	91.6	90.9	90.8	91.4
mAP50-95 (%)	74.6	75.6	74.3	72.2	**76.2**	**78.1**	63.0	74.1	75.3	74.9	75.6	73.3
Execution Time (H)	1.491	1.459	**1.423**	2.334	2.426	2.335	1.511	1.493	1.523	1.954	1.949	1.858

YOLO Models Trained from Scratch

Table 5. Confirm the dominance of the Improved YOLOv8n model, which consistently achieves the highest performance across all key metrics, maintaining strong accuracy even as batch sizes increase. YOLOv8n continues to strike an effective balance between detection performance and computational efficiency, often offering the shortest training times. YOLOv10n shows competitive precision but struggles with recall and mAP, limiting its overall effectiveness. YOLOv11n, while lightweight and sometimes fastest in execution, consistently underperforms in both accuracy and generalization, making it the least suitable choice across scenarios.

Table 5. Performance of YOLO Models Trained from Scratch on Dataset 2

	Yolov8n			**Improved Yolov8n**			**Yolov10n**			**Yolov11n**		
Batch	16	32	64	16	32	64	16	32	64	16	32	64
GFLOPs		8.2			12.6			8.4			6.4	
Precision (%)	80.2	89.1	81.5	**88.1**	93.4	**92.5**	79.7	94.2	80.0	89.5	86.5	81.3
Recall (%)	82.3	82.0	81.6	**85.7**	**87.1**	**83.5**	78.8	78.0	80.5	80.4	74.8	76.1
mAP 50 (%)	86.0	88.7	85.7	**90.1**	**91.6**	**91.3**	83.2	85.0	85.2	87.5	84.2	83.8
mAP50-95 (%)	64.7	66.2	65.9	**67.6**	**68.6**	**68.6**	62.3	64.1	63.9	63.1	63.3	62.9
Execution Time (H)	1.579	1.408	1.250	2.240	2.512	2.536	1.889	1.540	1.509	1.544	2.130	1.899

Training Dynamics and Generalization

The visual comparisons in the figure 1. illustrate the superior detection capabilities of the Improved YOLOv8n model across a range of challenging scenarios. Case (a), highlights a difficult detection case involving small and obscured objects under low-contrast conditions. YOLOv8n and YOLOv10n fail to detect any object. YOLOv11n misidentifies a tree as a person (confidence 0.70), likely due to similar shape and low contrast. Only the Improved YOLOv8n successfully detects a distant pedestrian with a confidence of 0.44, showing superior sensitivity and robustness in challenging conditions likely due to advanced fine-tuning. In (b), the results highlight differences in detection accuracy and confidence, with the custom model demonstrating more consistent and reliable performance.

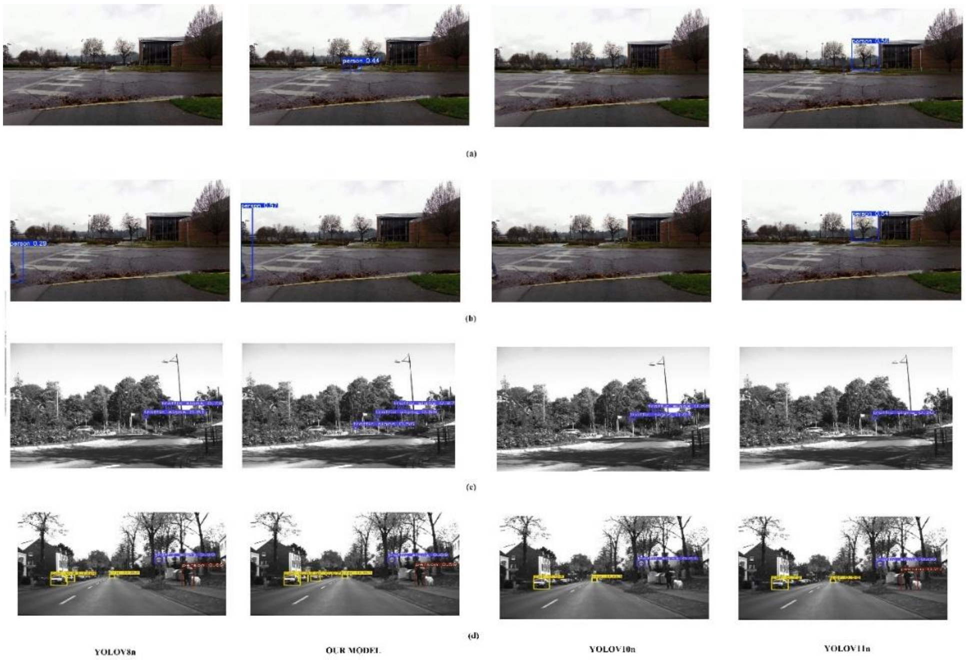

Figure 1. Comparative Object Detection Performance of YOLO Variants in Challenging Visual Conditions. In (c), our model shows better robustness in low-visibility conditions, while in (d), it delivers more accurate and complete detections in a complex urban scene compared to other models.

CONCLUSION AND PERSPECTIVES

This paper proposed an improved version of the YOLOv8n model, which was customized to enhance small object detection on a custom datasets. The recommended modifications, including the addition of a new detection scale (P2) and an improved feature fusion method, have significantly improved the model's ability to detect small-scale objects that were difficult to detect previously. By refining the multi-scale feature aggregation process and integrating additional Upsample, Concat, and C2f layers, the enhanced YOLOv8n model demonstrated superior performance in terms of precision, recall, and mAP. This research compared the performance of various object detection models such as YOLOv8n, YOLOv10n, YOLOv11n, and the optimized custom model on two distinct datasets. The experiments conducted under different training conditions (e.g., training from scratch versus pretrained models, and varying batch sizes) reveal that the customized YOLOv8n model consistently outperforms baseline models in precision, recall, and mAP.

Surprisingly, the custom model performed optimally in most environments, particularly with pretrained weights, for both mAP at threshold = 0.5 and mAP at threshold between 0.5 and 0.95. These findings underscore the importance of architectural optimization and transfer learning in enhancing object detection accuracy, even for lightweight models suitable for real-time applications. Subsequent research will focus on incorporating adaptive attention mechanisms, optimized network architectures, and semi-supervised learning techniques to further improve the operational efficiency and generalizability of the model. Additionally, extending the model's application to multi-object tracking and real-time video processing will broaden its utility in practical scenarios. By further optimizing its structure and training process, the high-performance YOLOv8n model can serve as a more robust and efficient tool for identifying small objects across diverse real-time environments.

REFRENCES

[1] Girshick, R. (2015). *Fast R-CNN* (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1504.08083.

[2] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation (Version 5). arXiv. https://doi.org/10.48550/ARXIV.1311.2524.

[3] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN (Version 3). arXiv. https://doi.org/10.48550/ARXIV.1703.06870.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), Computer Vision –ECCV 2014 (Vol. 8691, pp. 346–361). Springer International Publishing. https://doi.org/10.1007/978-3-319-10578-9_23.

[5] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1704.04861.

[6] Hussain, M. (2024). YOLOv5, YOLOv8 and YOLOv10: The Go-To Detectors for Real-time Vision (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2407.02988.

[7] Khalili, B., & Smyth, A. W. (2024). SOD-YOLOv8—Enhancing YOLOv8 for Small Object Detection in Traffic Scenes (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2408.04786.

[8] Krolkral, N. W., Mohamed Faraoun, K., Bousahba, N., Rezzouk, B., & Hamouda, I. A. (2023). Improved YOLOv5s for Object Detection. 2023 International Conference on Electrical Engineering and Advanced Technology (ICEEAT), 1–6. https://doi.org/10.1109/ICEEAT60471.2023.10425837.

[9] Li, F., & Jia, J. (2024). Multi-Class Military Target Detection Algorithm Based on Improved YOLOv8. 2024 5th International Conference on Machine Learning and Computer Application (ICMLCA), 431–435. https://doi.org/10.1109/ICMLCA63499.2024.10753821.

[10] Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., & Yang, J. (2020). Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2006.04388.

[11] Liang, R., & Wu, T. (2025). Enhancement of YOLOv8 model for dense crowd scenes: Incorporating an improved feature pyramid with attention mechanisms. In H. Yuan & L. Leng (Eds.), Fourth International Conference on Computer Vision, Application, and Algorithm (CVAA 2024) (p. 21). SPIE. https://doi.org/10.1117/12.3055731.

[12] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path Aggregation Network for Instance Segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8759–8768. https://doi.org/10.1109/CVPR.2018.00913;

[13] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), Computer Vision –ECCV 2016 (Vol. 9905, pp. 21–37). Springer International Publishing. https://doi.org/10.1007 /978-3-319-46448-0_2.

[14] Patel, S., & Patel, A. (2021). Object Detection with Convolutional Neural Networks. In A. Joshi, M. Khosravy, & N. Gupta (Eds.), Machine Learning for Predictive Analysis (Vol. 141, pp. 529–539). Springer Singapore. https://doi.org/10.1007/978-981-15-7106-0_52.

[15] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection (Version 5). arXiv. https://doi.org/10.48550/ARXIV.1506.02640.

[16] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (Version 3). arXiv. https://doi.org/10.48550/ARXIV.1506.01497.

[17] Sarda, A., Dixit, S., & Bhan, A. (2021). Object Detection for Autonomous Driving using YOLO algorithm. 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), 447–451. https://doi.org/10.1109/ICIEM51511.2021.9445365.

[18] Seth, Y., & Sivagami, M. (2025). Enhanced YOLOv8 Object Detection Model for Construction Worker Safety Using Image Transformations. IEEE Access, 13, 10582–10594. https://doi.org/10.1109/ACCESS.2025.3527511.

[19] Singh, S., & G N, R. (2024). Military Based Object Detection in Satellite Imagery by Optimising YOLOv8. 2024 IEEE Space, Aerospace and Defence Conference (SPACE), 165–168. https://doi.org/10.1109/SPACE63117.2024.10667819.

[20] Tan, M., Pang, R., & Le, Q. V. (2019). EfficientDet: Scalable and Efficient Object Detection. https://doi.org/10.48550/ARXIV.1911.09070.

[21] Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction, 5(4), 1680–1716. https://doi.org/10.3390/make5040083.

[22] Wang, C.-Y., Liao, H.-Y. M., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., & Hsieh, J.-W. (2019). CSPNet: A New Backbone that can Enhance Learning Capability of CNN (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1911.11929.

Journal of Information Systems Engineering and Management
2025, 10(55s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[23] Wu, D., Fang, C., Zheng, X., Liu, J., Wang, S., & Huang, X. (2024). AMW-YOLOv8n: Road Scene Object Detection Based on an Improved YOLOv8. Electronics, 13(20), 4121. https://doi.org/10.3390/electronics13204121.

[24] Wu, Q., Li, X., Xu, C., & Zhu, J. (2024). An Improved YOLOv8n Algorithm for Small Object Detection in Aerial Images. 2024 9th International Conference on Signal and Image Processing (ICSIP), 607–611. https://doi.org/10.1109/ICSIP61881.2024.10671469.

[25] Yaseen, M. (2024). What is YOLOv8: An In-Depth Exploration of the Internal Features of the NextGeneration Object Detector (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2408.15857.

[26] Zeng, W., Wu, P., Wang, J., Hu, G., & Zhao, J. (2024). C4D-YOLOv8: Improved YOLOv8 for Object Detection on Drone-captured Images. In Review. https://doi.org/10.21203/rs.3.rs-4658932/v1.

[27] Zhao, H., Tang, Z., Li, Z., Dong, Y., Si, Y., Lu, M., & Panoutsos, G. (2024). Real-time object detection and robotic manipulation for agriculture using a YOLO-based learning approach (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2401.15785.

[28] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2019). Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression (Version 1). arXiv. https://doi.org/10.48550/ARXIV.1911.08287.

[29] Zhou, W., Zhu, C., & Miao, D. (2024). Object Detection Model of YOLOv8-CSD for UAV Images. 2024 7th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), 77–83. https://doi.org/10.1109/PRAI62207.2024.10826612.

[30] Ultralytics, ``Issue \#189 on Ultralytics GitHub repository,'' GitHub, 2023. [Online]. Available: \url{https://github.com/ultralytics/ultralytics/issues/189}. [Accessed: May 30, 2025].