iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71284

International Journal For Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Comparative Analysis of YOLOv8, YOLOv9, and YOLOv10 for Object Detection: Performance Metrics and Real-World Applications

Tumanan Silvanus[1], Palikhe Bishwas Shankar[2], Abdurrahman Dhiya[2]
Computer Science, Nanjing University of Information Science and Technology, China

Abstract: In computer vision, object detection is still an essential job with applications in robotics, autonomous cars, and surveillance. The YOLO (You Only Look Once) model family is a well-liked option for real-time object recognition because of its reputation for striking a balance between speed and accuracy. Using the Pascal VOC 2012 dataset, this study presents a comparative examination of three YOLO variants: YOLOv8, YOLOv9, and YOLOv10. Key metrics included in the research are F1-score, accuracy, recall, and mean Average accuracy (mAP). Furthermore, a loss curve analysis is performed to evaluate each model's training effectiveness. The findings show that YOLOv9, which excels in identifying smaller and more complex items, has the best recall to accuracy ratio. Although YOLOv10 exhibits a modest underperformance in peak accuracy, its improved computing efficiency renders it the best option for real-time workloads. Even though YOLOv8 is the quickest, it has trouble with little objects and complex sceneries. This study helps each model's use in a variety of real-world circumstances by offering insightful information about its advantages and disadvantages.
Keywords: Object Detection, YOLOv8, YOLOv9, YOLOv10, Pascal VOC 2012, Real-Time Applications, Mean Average Precision (mAP).

I. INTRODUCTION

Object detection is an essential component of computer vision, with applications ranging from security and medical to driverless automobiles and robotics. By offering an efficient trade-off between detection accuracy and inference speed, the YOLO (You Only
New architectural elements designed to improve performance in real-world scenarios are included with every iteration of YOLO. Because YOLOv8 enhanced detection speed while maintaining a respectable level of accuracy, it is ideal for real-time applications where decreased latency is crucial [4]. In particular, for smaller and more complex objects, YOLOv9 increased detection accuracy by using advanced feature pyramids and post-processing techniques [16]. YOLOv10, the most recent version, is suitable for usage in resource-constrained scenarios since it strikes a compromise between speed and precision to increase computational efficiency [6].
It is essential to compare YOLOv8, YOLOv9, and YOLOv10 since each model tackles a distinct set of real-world object identification issues. Knowing how these models function in a variety of scenarios is what makes this comparison relevant. For example, low-latency models are required for real-time object identification systems such as autonomous driving, yet high detection accuracy may be prioritized in medical imaging applications [15]. Our goal is to offer valuable insights into the advantages and disadvantages of these models through a thorough comparative study, hence facilitating their implementation across a range of businesses.
In this work, the standard object detection benchmark PASCAL VOC 2012 dataset is used to assess the performance of YOLOv8, YOLOv9, and YOLOv10 [5]. Our analysis centers on crucial performance indicators, including F1-score, accuracy, recall, and mean Average accuracy (mAP), offering a thorough assessment of every model's capacity for detection [14]. Furthermore, we examine each model's corresponding loss curve to assess the models' training efficiency, emphasizing the models' convergence and stability.

II. LITERATURE REVIEW

A. Evolution of YOLO Models

Since its introduction by Redmon et al. in 2016, the YOLO (You Only Look Once) model has become a standard in the object recognition area. YOLO's unique architecture allows it to evaluate an entire image in a single neural network run, giving it unparalleled speed in object recognition—a feature that makes it perfect for real-time applications [1]. Significant increases in accuracy and performance have been achieved by making several adjustments to the original YOLO model [5].

International Journal For Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available At www.ijraset.com

By using methods including multi-scale training, anchor boxes, and batch normalization, YOLOv2 and YOLOv3 brought gains in accuracy [12]. Bochkovskiy et al.'s 2020 development of YOLOv4 added additional functionality, such as Spatial Pyramid Pooling (SPP) and Cross Stage Partial networks (CSPNet), to further progress the system [2]. Thanks to these improvements, YOLOv4 was able to attain cutting-edge performance in terms of accuracy and speed.

After YOLOv4, YOLOv5 was released, pushing the limits of what was feasible for real-time object identification even though it wasn't from the original inventors. Because of its versatility, modular design, and compatibility with several deployment platforms, YOLOv5 has become widely used [8]. Even so, YOLOv6, YOLOv7, and later YOLOv8, YOLOv9, and YOLOv10 were officially released after its widespread success. The goal of these latter iterations was to improve training techniques, optimize the architecture even more, and improve loss functions in order to increase performance metrics across different benchmarks.

B. YOLOv8, YOLOv9, and YOLOv10: Advancements and Comparisons

When YOLOv8 was first presented to the larger research community, it focused on efficiency and flexibility, which made it appropriate for use on edge devices with constrained computational resources [4]. To balance speed and accuracy, it used a unique loss function and innovative methods such as EfficientNet backbones.

Further advancements in network design were brought forth by YOLOv9, which primarily concentrated on improving feature representation by utilizing sophisticated attention mechanisms [18]. These modifications were intended to boost overall mAP and enhance the model's capacity to identify tiny objects, especially in difficult datasets.

The most recent version, YOLOv10, improved upon the strengths of its predecessors with a more robust feature pyramid network and stronger self-supervised learning approach integration. YOLOv10 was able to maintain competitive inference speeds and achieve better precision and recall because to these enhancements [17].

C. Comparative Studies and Performance Benchmarks

The effectiveness of YOLO models on various datasets and under various settings has been compared in a number of studies. Research, for example, comparing YOLOv3 with YOLOv4 demonstrated the notable gains in speed and accuracy that YOLOv4 made, especially when it came to object detection in complicated situations [7]. Research also showed that YOLOv5 outperformed its predecessors in terms of accuracy and processing efficiency.

There is, however, a glaring lack of studies directly comparing YOLOv8, YOLOv9, and YOLOv10, especially when utilizing standardized datasets as Pascal VOC 2012. This is in contrast to the substantial literature on previous YOLO models. Instead of a thorough, head-to-head comparison under the same conditions, existing studies usually concentrate on the performance of particular models.

D. Research Gaps

Although YOLO models' individual performance is extensively documented, there is a dearth of comprehensive research that directly compares YOLOv8, YOLOv9, and YOLOv10. This disparity is especially noticeable in research using standardized datasets such as Pascal VOC 2012, which offers a uniform baseline by which to compare object detection models. Furthermore, little study has been done on the computational trade-offs that these models involve, such inference time and resource usage, which are important considerations in real-world applications.

III. METHODOLOGY

This chapter compares the performance of the YOLOv8, YOLOv9, and YOLOv10 models by describing the experimental setup and assessment measures. The F1-confidence curve, precision-recall curve, and loss curves—which offer information on each model's accuracy, precision, recall, and training efficiency—are among the primary metrics that were assessed.

A. Experimental Setup

1) Hardware and Software

Google Colab was used in conjunction with a T4 GPU on a MacBook M1 Pro for doing the research. The Pascal VOC 2012 dataset was processed using the Roboflow platform, and the YOLO models were created using the Ultralytics YOLO package [4].

- Library: Ultralytics YOLO
- Hardware: MacBook M1 Pro, Google Colab with T4 GPU
- Dataset Preparation: Pascal VOC 2012 (17,112 images) via Roboflow

International Journal For Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

- Models: YOLOv8n, YOLOv9t, and YOLOv10n
- Training Epochs: 10 epochs

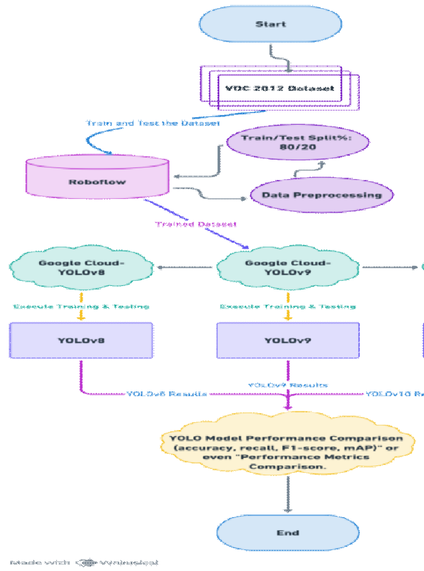


Figure 3.1: Workflow of YOLO Model Training and Evaluation.

This flowchart illustrates the step-by-step process from data acquisition and preprocessing to the comparison of results across the YOLOv8, YOLOv9, and YOLOv10 models. The Pascal VOC 2012 dataset is first processed and split into training and testing sets using Roboflow. The trained dataset is then used to train each YOLO model on separate Google Cloud instances. After training, the models' results are compared using evaluation metrics such as accuracy, recall, F1-score, and mean Average Precision (mAP).

2) *Dataset*

The Pascal VOC 2012 dataset, containing 17,112 images across 20 object categories, was used for both training and evaluation [10]. The dataset was preprocessed using Roboflow to ensure uniformity in data handling across all models.

3) *Training Procedure*

Each model was trained using the Pascal VOC 2012 dataset across ten epochs. Typical data augmentation methods (such clipping and flipping) were used to improve the generalization of the models. For improved convergence, the Adam optimizer was applied with a progressively lower learning rate. To establish a solid baseline for comparison, pre-trained COCO weights were used to start all models [11].

B. *Evaluation Metrics*

The models were evaluated using the following metrics:

- Precision: The ratio of true positive detections to the total predicted positives, reflecting the model's accuracy in its predictions.
- Recall: The ratio of true positive detections to all actual positives, indicating the model’s ability to identify relevant objects.
- F1-Score: The harmonic mean of precision and recall, providing a balance between the two metrics.
- Mean Average Precision (mAP): Calculated over different Intersection over Union (IoU) thresholds, mAP provides a holistic measure of detection accuracy.
- Loss Curves: These curves track the models’ training progress and convergence, providing insights into training efficiency and stability.

C. *YOLO Model Performance Comparison*

The performance of YOLOv8n, YOLOv9t, and YOLOv10n was evaluated using several key metrics: the F1-confidence curve, precision-recall curve, and loss curves. These metrics provide insight into the models' accuracy, precision, recall, and training efficiency.

International Journal For Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available At www.ijraset.com

To illustrate the complete workflow of data processing, training, and evaluation for each YOLO model, the flowchart in Figure 3.1 presents the experimental setup. It outlines the steps from dataset preparation to the comparison of the models' results using metrics like accuracy, recall, F1-score, and mAP.

IV. RESULTS AND ANALYSIS

This chapter presents the performance analysis of YOLOv8n, YOLOv9t, and YOLOv10n models using key evaluation metrics. We focus on precision, recall, F1-score, mAP, and loss curves to highlight the strengths and weaknesses of each model.

A. Precision, Recall, and F1-Score

To assess the balance between accuracy and recall across various confidence levels, the precision-recall and F1-confidence curves were created [9]. These curves are essential for comprehending the models' object detection performance in different settings.

1) Precision-Recall Curves

The Precision-Recall Curves for YOLOv8, YOLOv9, and YOLOv10 are shown in Figure 4.1. These curves illustrate how each model performs in terms of true positive and false positivepredictions.

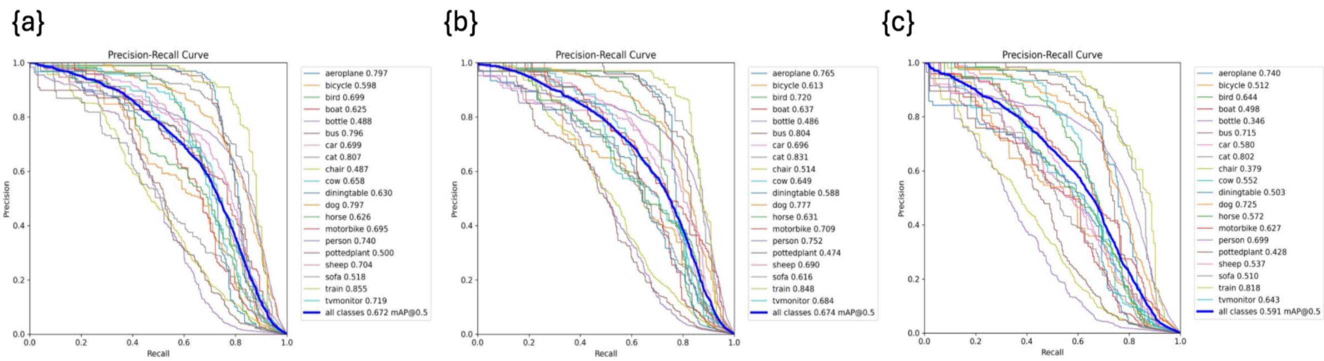


Figure 4.1: Precision-Recall curves for {a}YOLOv8, {b}YOLOv9, and {c}YOLOv10, showing the trade-off between precision and recall.

Performance Overview

- YOLOv8: Peak F1-score of 0.65 at a confidence threshold of 0.372. Performs well on larger, more distinct objects but struggles with smaller objects like bottles and chairs.
- YOLOv9: Peak F1-score of 0.66 at a confidence threshold of 0.342. Improved handling of smaller objects and complex scenes, particularly for aeroplanes and persons.
- YOLOv10: Peak F1-score of 0.59 at a confidence threshold of 0.225. Stable across categories but slightly underperforms compared to YOLOv9 for smaller objects.

2) F1-Confidence Curves

The F1-Confidence Curves highlight the balance between precision and recall as confidence thresholds change. Figure 4.2 presents the combined F1-confidence curves for all three models.

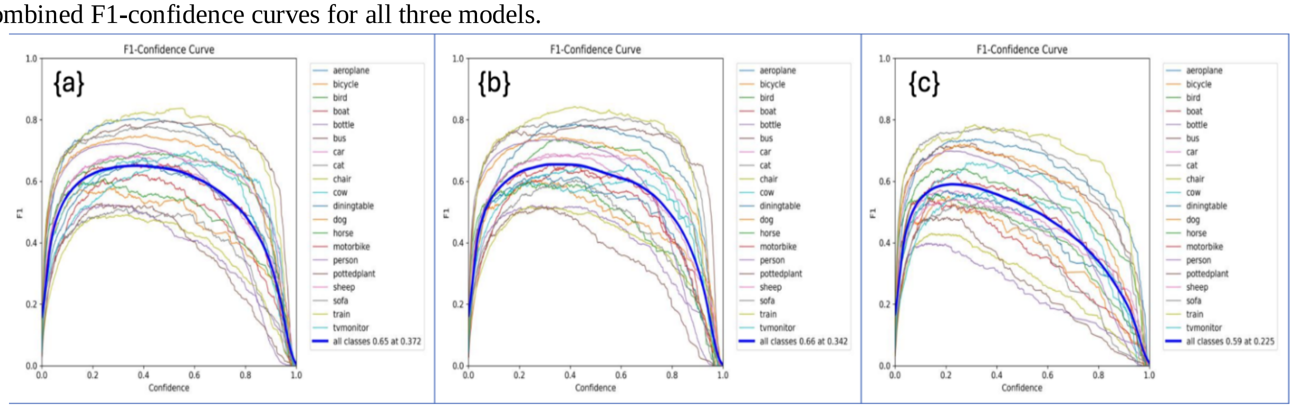


Figure 4.2: F1-score curves for {a}YOLOv8, {b}YOLOv9, and {c}YOLOv10, illustrating the balance between precision and recall.

B. Loss Curves

Loss Curves, which display the training process across ten epochs, are used to assess the models' training efficacy. The speed at which each model converges during training is shown by these graphs.

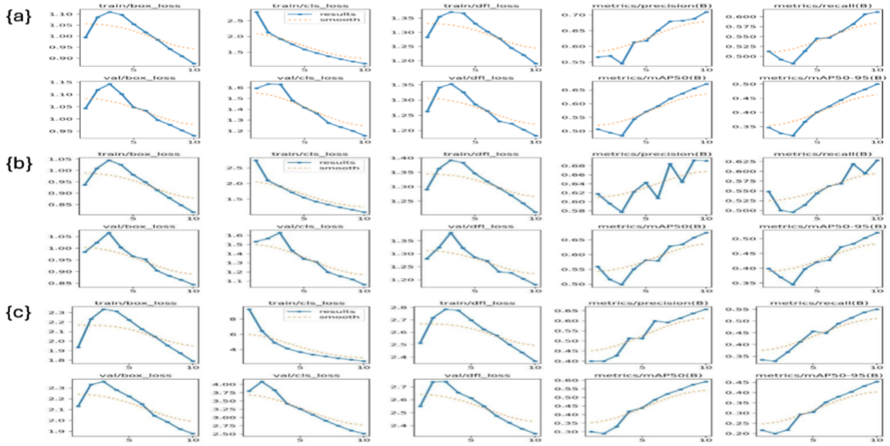


Figure 4.3: Combined Loss curves (box loss, classification loss, and distribution focal loss) for {a}YOLOv8, {b}YOLOv9, and {c}YOLOv10.

- YOLOv8: Exhibits moderate convergence but reaches a higher final loss value compared to the other models.
- YOLOv9: Shows faster convergence and lower final loss, indicating efficient training and better optimization.
- YOLOv10: Achieves the lowest final loss, with rapid convergence, making it well-suited for real-time applications.

C. Mean Average Precision (mAP)

The mAP is a critical metric for evaluating detection accuracy across different IoU thresholds. The results for mAP50 and mAP50-95 are shown in Table 4.1.

TABLE 4.1: MAP RESULTS FOR YOLOV8, YOLOV9, AND YOLOV10.

model	mAP50	mAP50-95
YOLOv8	0.506	0.346
YOLOv9	0.559	0.398
YOLOv10	0.591	0.453

- YOLOv10 shows the best overall mAP performance, particularly for mAP50-95, which considers a broader range of IoU thresholds.
- YOLOv9 offers a good balance between accuracy and computational efficiency.
- YOLOv8 lags behind in mAP but remains the fastest model.

D. Confusion Matrix

For simplicity of comparison, the confusion matrices for YOLOv8, YOLOv9, and YOLOv10 have been consolidated into a single figure. These matrices provide information on how accurately the models classify different types of objects.

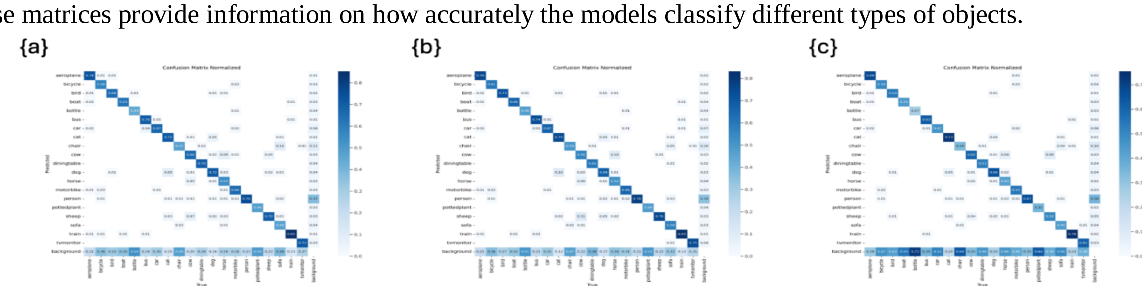


Figure 4.4: Normalized confusion matrix comparing {a}YOLOv8, {b}YOLOv9, and {c}YOLOv10, showing classification accuracy across all object categories.

- YOLOv8: Exhibits strong performance in identifying large, distinct objects but struggles with smaller categories.
- YOLOv9: Demonstrates improvements in identifying challenging categories such as bottles and chairs.
- YOLOv10: Shows balanced performance, though it slightly underperforms compared to YOLOv9 for smaller objects.

E. Summary of Findings

- YOLOv9 strikes the best balance between precision, recall, and efficiency, making it the top performer across various object categories.
- YOLOv10 offers enhanced computational efficiency and stability, making it ideal for real-time applications.
- YOLOv8 remains the fastest but is less suitable for tasks requiring high accuracy with smaller objects.

Each model presents distinct advantages depending on the application, with YOLOv9 best suited for high-accuracy detection, YOLOv10 for balanced real-time applications, and YOLOv8 for speed-critical tasks.

V. DISCUSSIONAND CONCLUSION

In this chapter, we discuss the implications of the results presented in Chapter 4 and provide a conclusion based on the comparative analysis of YOLOv8n, YOLOv9t, and YOLOv10n models. We also outline potential directions for future research.

A. Discussion

The performance evaluation of YOLOv8n, YOLOv9t, and YOLOv10n models highlights several key insights that are critical for choosing the appropriate model for specific applications.

1) Precision, Recall, and F1-Score

YOLOv9t consistently outperforms the other models in terms of F1-score and precision-recall balance, particularly for smaller and more complex objects [16]. This makes YOLOv9t ideal for applications requiring high accuracy in challenging environments.

YOLOv8n, while fast, shows limitations in detecting smaller objects, which could be a drawback in applications where fine-grained detection is necessary. However, its speed advantage may be useful in scenarios where real-time processing is prioritized.

YOLOv10n strikes a balance between efficiency and accuracy, showing robust performance across different object categories. This model is well-suited for real-time applications where computational efficiency is crucial.

2) Loss Curves and Training Efficiency

YOLOv10n demonstrates the most efficient training process, with rapid convergence and the lowest final loss values, making it an excellent choice for deployment in resource-constrained environments.

YOLOv9t also converges efficiently but achieves slightly higher final loss values compared to YOLOv10n. Its performance in terms of accuracy compensates for this, particularly in detecting small objects.

YOLOv8n shows slower convergence and higher final loss values, indicating potential overfitting or challenges in optimizing the model for the Pascal VOC 2012 dataset.

3) Mean Average Precision (mAP)

YOLOv10n achieves the highest mAP50-95, indicating strong performance across a range of IoU thresholds. This suggests that YOLOv10n is capable of maintaining accuracy even in more demanding detection tasks.

YOLOv9t follows closely, with its superior handling of smaller objects reflected in a slightly lower but still impressive mAP score.

YOLOv8n lags behind in mAP, which aligns with its struggles in detecting smaller objects. Despite this, it remains a viable option for applications where speed is more critical than high accuracy.

4) Confusion Matrix Analysis

The combined confusion matrix shows that YOLOv9t and YOLOv10n excel in categories with smaller or more complex objects, such as bottles and chairs, compared to YOLOv8n.

YOLOv8n shows strong performance in categories with larger, more distinct objects like aeroplanes and persons but falters with smaller categories, indicating a need for further optimization in these areas.

B. Conclusion

This comparative analysis of YOLOv8n, YOLOv9t, and YOLOv10n on the Pascal VOC 2012 dataset provides valuable insights into the strengths and limitations of each model:

YOLOv9t emerges as the best overall performer, particularly in scenarios requiring high accuracy and precision for smaller objects.

YOLOv10n offers the best balance between computational efficiency and detection accuracy, making it suitable for real-time applications where resources are limited.

YOLOv8n, while the fastest, is best suited for applications where rapid processing is critical, but high accuracy with smaller objects is less of a concern.

C. Future Work

Improving Small Object Detection: Future research should focus on enhancing the detection capabilities of YOLO models for small objects, potentially through model architecture improvements or advanced data augmentation techniques.

Hybrid Models: Exploring hybrid approaches that combine the strengths of YOLOv9t and YOLOv10n could lead to models that offer both high accuracy and computational efficiency.

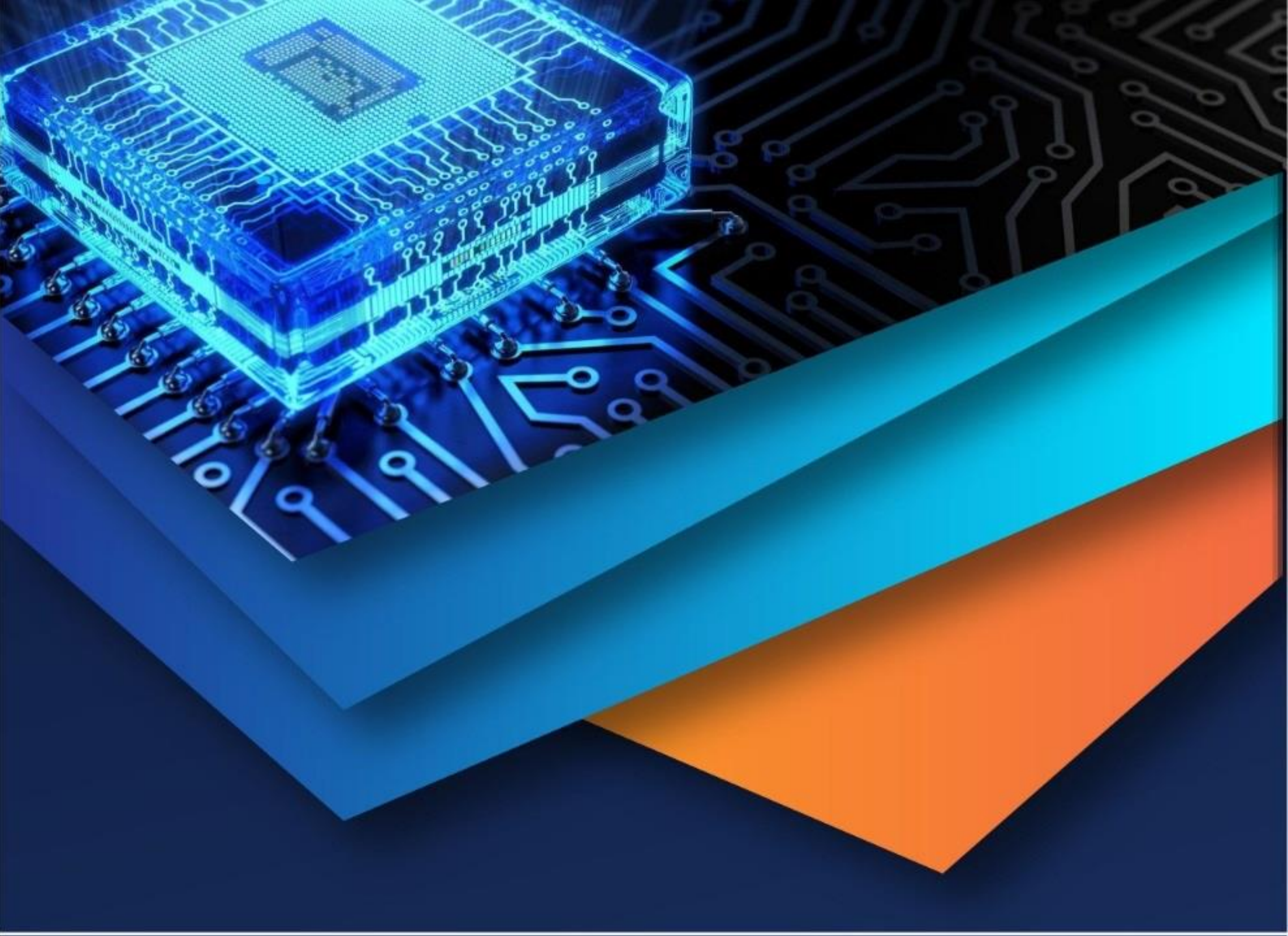
Deployment in Real-World Scenarios: Further evaluation of these models in diverse, real-world environments can provide additional insights into their practical applications, leading to more tailored model selection based on specific use cases.

VI. ACKNOWLEDGMENT

The authors wish to thank their colleagues at Nanjing University of Information Science and Technology for their support and feedback during the research process. This work was not supported by any external funding.

REFERENCES

[1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779-788.
[2] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
[3] Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. arXiv preprint arXiv:2207.02696.
[4] Jocher, G., Chaurasia, A., Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. Retrieved from https://github.com/ultralytics/ultralytics.
[5] Terven, J., & Cordova-Esparza, D. (2023). A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond.
[6] Li, C., Li, L., Jiang, H., et al. (2022). YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications.
[7] Safie, S. I., Kamal, N. S. A., Yusof, E. M. M., et al. (2023). Comparison of SqueezeNet and DarkNet-53 based YOLO-V3 Performance for Beehive Intelligent Monitoring System. 2023 IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE).
[8] Gillani, I. S., Munawar, M. R., Talha, M., et al. (2023). YOLOv5, YOLO-X, YOLO-R, YOLOv7 Performance Comparison: A Survey.
[9] Padilla, R., Netto, S. L., & Da Silva, E. A. B. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. 2020 International Conference on Systems Signals and Image Processing (IWSSIP).
[10] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft COCO: Common Objects in Context.
[11] Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137-1149.
[12] Farhadi, A., & Redmon, J. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
[13] Feng, C., Zhong, Y., Gao, Y., et al. (2021). TOOD: Task-aligned One-Stage Object Detection. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 626-643.
[14] Shuo, W. (2021). Research Towards YOLO-Series Algorithms: Comparison and Analysis of Object Detection Models for Real-Time UAV Applications. Journal of Physics: Conference Series, 1948(1), 012021.
[15] Sanchez-Matilla, R., Poiesi, F., & Cavallaro, A. (2016). Online Multi-Target Tracking with Strong and Weak Detections. Proceedings of the European Conference on Computer Vision (ECCV), 9914, 84-99.
[16] Diwan, T., Anirudh, G., & Tembhurne, J.V. (2022). Object Detection using YOLO: Challenges, Architectural Successors, Datasets and Applications. Multimed. Tools Appl., 82, 9243–9275.
[17] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., & Ding, G. (2024). YOLOv10: Real-Time End-to-End Object Detection. arXiv:2405.14458. https://doi.org/10.48550/arXiv.2405.14458.
[18] Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2024). YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. arXiv:2402.13616. https://doi.org/10.48550/arXiv.2402.13616.



10.22214/IJRASET

INDEX COPERNICUS 45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY