RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt26212

Benchmarking YOLO Variants on an Unseen Video: A Comparison of Inference Speed, GFLOPs, and Recall

T.G. Vibha [a, b, *], S. Theodore Chandra [a], S. Sivaramakrishnan [c]

[a] Department of Electronics and Communication Engineering, Dayananda Sagar University, Bengaluru, Karnataka, India

[b] Department of Electronics and Communication Engineering, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India

[c] School of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India

* Corresponding Author Email: vibhatg@gmail.com

DOI: https://doi.org/10.54392/irjmt26212

Received: 18-09-2025; Revised: 06-03-2026; Accepted: 15-03-2026; Published: 28-03-2026

Abstract: Object detection serves as a fundamental task in the field of computer vision and recent developments in YOLO family aim to enhance the real time detection performance. However, the generalization performance of recent YOLO models on unseen video datasets remains underexplored. Analysing the model performance on unseen datasets is essential for assessing robustness in real world deployments. This work provides a systematic comparison of recent YOLO versions and their variants. The study evaluates YOLOv9, YOLOv10 and YOLOv11 models on an unseen MOT20 video dataset for multi pedestrian detection. Pedestrian detection is important because it forms the basis for many computer vision tasks involving human interaction, crowd monitoring, behaviour analysis, and traffic management the models and their variants are evaluated using the metrics: recall, inference speed and GFLOPs. The experimental results indicates that the variant YOLOv9-m achieves highest recall of 43.9% among all evaluated models, while YOLOv11-n showed marginally lower recall value of 40.9%. However, YOLOv11-n exhibits significantly faster inference speed (9.6ms per image) compared to YOLOv9-m (26.3ms per image) and fewer computational resources-(6.5 Vs 131.3). In contrast YOLOv10 exhibits significantly lower recall (28%) despite its increased efficiency. These findings highlight the inherent trade-offs between accuracy-efficiency in recent YOLO architectures. The study offers the understanding of strengths and limitations of modern YOLO models, aiding in model selection for real time computer vision applications.

Keywords: GFLOPs, Inference speed, Recall, Unseen video, Yolo

1. Introduction

Detecting, locating and tracking of the object in a video plays a vital role in numerous applications such as surveillance, traffic monitoring, sports analytics, wildlife monitoring and so on [1]. Detection is challenging as well due to complex visual data, low image resolution, rough image quality [2]. Deep learning techniques are used for the detection and tracking purpose because of its effectiveness in extracting the features automatically. Deep learning models are playing efficient role in tracking objects under various detection challenging scenarios like occlusion, motion blur, dense crowd and so on [3]. Pedestrian detection is an important application of object detection for surveillance, autonomous driving and traffic monitoring systems. Consequently, accurate detection of multiple pedestrians is essential to support these applications. However, detection of multiple pedestrians in a video comes up with various challenging aspects like varying poses, occlusion and dense scenes. These challenges increase missed detections and inaccurate localization under real-time constraints. This necessitates the efficient detection frameworks for detecting multiple pedestrians under crowded environments [4, 5].

The object detection methods are broadly classified under two categories-two-stage and one-stage approaches [6, 7]. Two stage detectors such as Faster R-CNN achieve higher accuracy by generating region proposals before classification but at the cost of slower inference. However, one stage detectors such as YOLO and SSD perform the detection under single pass making them faster and suitable for real -time applications.

The YOLO (You Only Look Once) architecture has attracted widespread interest under single-stage object detection framework because of its higher accuracy and low latency. The YOLO framework has undergone successive architectural modifications leading to improved accuracy and computational efficiency. The initial YOLO model focused on single

stage detection. Subsequent later versions from YOLOv3 through YOLOv8 included the architectural innovations such as multi-scale feature extraction, pyramid-based feature fusion and anchor-based learning mechanisms. These advancements have enabled YOLO models to deliver competitive detection performance [8, 10]

Recent advancements in YOLO versions - YOLOv9, YOLOv10, and YOLOv11 incorporate new architectural designs and training strategies for improving detection accuracy and computation efficiency.YOLOv9 improves feature learning through Programmable gradients and Generalized Efficient Layer Aggregation Network(GELAN).This enables better learning of complex patterns in dense environments.YOLO10 focuses on anchor free end-to-end detection framework eliminating the need for Non Maximum Suppression(NMS) reducing post-processing overhead and improving inference speed.YOLOv11 further refines its architecture and training process for achieving improved detection accuracy as well as efficiency [10, 11]. The YOLOv9, YOLO10, and YOLOv11 models are available in multiple sizes to suit different performance and efficiency needs. YOLOv9 comes in nano, small, medium, compact, and extended versions, each designed to balance speed, accuracy, and computational efficiency. YOLOv10 and YOLOv11 is offered in nano, small, medium, balanced, large, and extra-large variants, each tailored to different levels of accuracy, speed, and computational demands.

Although YOLOv9, YOLOv10, and YOLOv11 perform well on widely used COCO datasets, their evaluation under crowded pedestrian environments is limited. The MOT20 [12] dataset is featured with crowded pedestrian videos which includes heavy occlusions, scale variation and illumination changes. However, the comparative study of YOLO variants on MOT20 remain scarce. Hence a systematic approach to study the behaviour of these models on MOT20 is necessary. Addressing this research gap will provide valuable insights in understanding the generalization capabilities, computational efficiency and suitability of YOLOv9, YOLOv10, and YOLOv11 models for crowded real world applications. Multiple variants of each YOLO model are evaluated to analyse the impact of model scaling on accuracy and efficiency.

The study emphasizes on the comparative analysis of the performance differences among the models YOLOv9 to YOLOv11. The models, along with their variants, are evaluated on an unseen video to assess their ability to generalize over crowded pedestrian scenario. An unseen video from MOT20 dataset is chosen for experimentation to ensure the evaluation is done under challenging conditions.

The models are assessed using three key performance metrics: recall percentage, GFLOPs and inference speed. Recall is used as the primary accuracy metric which measures the pedestrian detection ability of the models in dense and occluded environments. GFLOPs and inference speed are used to analyse computational efficiency. GFLOPs are used to measure the computational complexity of each model and the inference speed indicates the real time processing capability. Visualization results are presented for best-performing models to assess their detection behaviour in crowded scenario. Additionally, a comparative table for all the model variants is provided in terms of recall, GFLOPs and inference speed

These metrics collectively provide insights into the trade-offs between detection accuracy and computational efficiency. The comparative analysis of the models YOLOv9 to YOLOv11 reflects their strengths and limitations and their suitability in real-world applications.

2. Literature Review

The object detectors are generally classified as one stage and two stage as discussed in by Karbouj *et al* [13]. The deep learning models such as RCNN, Fast RCNN and Faster RCNN fall under the category two stage detectors whereas YOLO series, RetinaNet and EfficientNet are examples of one stage detectors. The one stage detectors go with single stage classification and recognition against the two stage detectors which does the region proposal first followed by the classification and regression [14]. The paper [15] demonstrates that the YOLO versions from YOLOV8 through YOLOv11 shows better performance in terms of speed and accuracy against the Faster RCNN.

YOLOv9 is introduced by wang *et al* [16] to overcome the problem of information bottleneck faced in earlier YOLO versions and other deep neural networks. The two architectures such as Programmable Gradient information (PGI) and GeLan (Generalized Efficient Layer Aggregation Network) are used in YOLOv9.PGI overcomes the data loss problem during training via reversible architecture and also reducing error accumulation during training epochs leading to reliable gradients. Gelan performs better feature fusion strengthening the backbone and neck of YOLOv9 leading to improved efficiency.

YOLOv10 developed by wang *et al* [17] represents substantial enhancements in real time detection. This is achieved by employing few unique methods such as

a) Eliminating the need of NMS (NON-MAXIMUM SUPPRESSION) in training process by adopting dual label assignment (one to one head and one to many head) thus allowing the model to learn filtering the redundant detections naturally without NMS. This leads to reduced computational cost and inference time.

b) To improve the efficiency of the model, down sampling operation in YOLOv10 is performed separately for reducing spatial dimensions and increasing number of channels. Initially 1*1 convolution is applied to enhance the channel depth while preserving the spatial dimensions and is followed by 3*3 convolution to reduce spatial features keeping number of channels intact. This decoupled down sampling approach reduces the computational cost which would be higher if both the operations were performed simultaneously.

c) Rank-Prioritized stages are used in YOLOv10 as against other YOLO series where the layers in all stages are treated equally. The last layer in each stage is analysed and rank is assigned based on the extent of redundant information present. A higher rank is provided for stages with less redundancy and Lower rank is assigned for higher redundancy. Based on these ranks the stages are sorted in ascending order and the blocks with lower ranks are replaced with compact inverted block structure until degradation in the performance is noticed.

d) Lower overhead is achieved by selecting a lightweight architecture for classification.

YOLOv11 as explained in the paper [18] employs a C3k2 block which utilizes smaller kernel sizes to enhance computational efficiency while preserving model's effectiveness in capturing key features. This approach enhances processing speed and minimizes computational overhead. Also, the improved SPPF (Spatial Pyramid Pooling – Fast) module helps detect objects of different sizes by capturing multi-scale information without using much computing power. The C2PSA (Convolutional block with Parallel Spatial Attention) module uses parallel spatial attention to help the model focus on important spatial details, enhancing detection accuracy, particularly in challenging environments. The table below highlights the key architectural innovations of YOLOv9, YOLOv10, and YOLOv11. Table 1 provides a comparative overview of YOLO-based one-stage object detection models, highlighting the datasets used and their key contributions.

3. Methodology

Motivated by the enhanced architectures of YOLOv9, YOLOv10 and YOLOv11 a comparison analysis among the models and their variants is conducted. The analysis is done on an unseen MOT20 video test dataset. The effectiveness of the models YOLOv9, YOLOv10 and YOLOv11 is evaluated on this unseen dataset. The selected parameters for assessing the model performance are: Recall inference speed and GFLOPs. The workflow diagram highlighting the key steps in analysis is shown in figure 1. The unseen video from MOT20 dataset is provided as an input to Pretrained YOLOv9, YOLOv10 and YOLOv11 models and their respective variants. The models are evaluated using recall, inference speed and GFLOPs as performance metrics. A comparative analysis is conducted to assess the strengths and limitations of each model based on these metrics. Additionally, visualization outputs of best performing models are provided for clearer interpretations.

Table 1. Summary of YOLO variants, evaluation datasets, and main contributions

Authors	Year	Model	Dataset	Evaluation Metrics	Main Contribution
Redmon *et al.* [19]	2016	YOLOv1	COCO	mAP(mean Average Precision)	Unified architecture for object detection framework
Bochkovskiy *et al*. [20]	2020	YOLOv4	COCO	mAP,FPS	Enhanced detection performance with optimized training strategies
Wang *et al*. [21]	2022	YOLOv7	COCO	mAP, FPS	Achieved Benchmark results for real-time object detection
Afifah *et* al [22]	2023	YOLOv8	COCO	Precision, Recall	Anchor free detection approach
Wang *et al.* [16]	2024	YOLOv9	COCO	mAP, Recall	Introduced Programmable gradient information for effective learning
Khanam *&* Hussain [18]	2024	YOLOv11	COCO	Speed, GFLOPs	Developed Lightweight variants for efficient inference
Proposed Study		YOLOv9–YOLOv11 variants	Unseen crowded pedestrian video-MOT20	Recall, Inference Speed, GFLOPs	Analysis of accuracy–efficiency trade-offs in crowded pedestrian scenarios

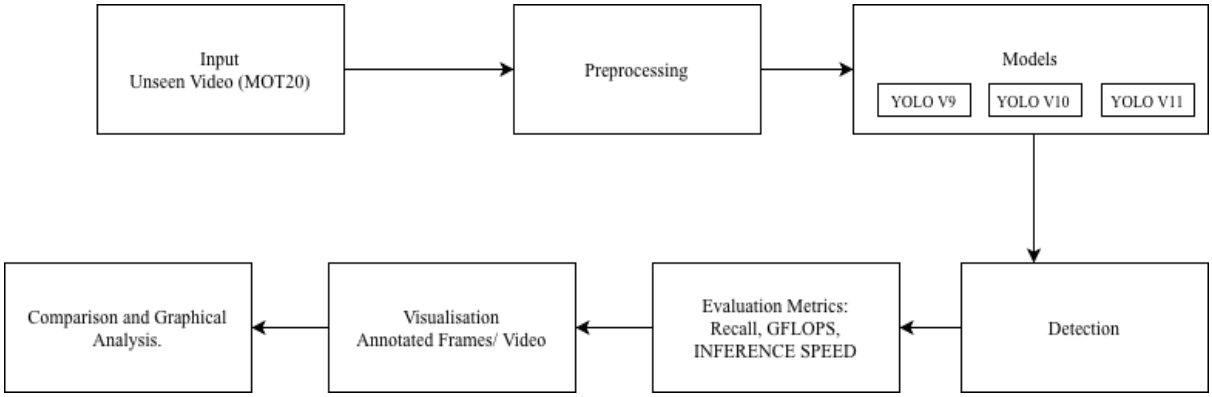

Figure 1. Block diagram representing the workflow

3.1 Models Used

The study compares the performance of the three states of art models: YOLOv9, YOLOv10 and YOLOv11. All three belong to the YOLO (You Only Look Once) family, which is widely recognized for achieving fast and accurate object detection.

YOLOv9 integrates transformer-based components in its architecture to enhance its ability to capture spatial and contextual relationships in images.

YOLOv10 is designed with a focus on efficiency, offering a streamlined structure suited for low-resource environments, such as mobile or embedded systems.

YOLOv11 further refines the model design by incorporating lightweight attention modules, enabling faster inference while still maintaining strong detection performance.

Each model is tested using the same dataset and input size to ensure a consistent basis for performance comparison across all evaluations.

3.2 Test Input

A single unfamiliar video from MOT20 dataset has been selected for comparing the performance of the models YOLOv9, YOLOv10 and YOLOv11. It is a crowded indoor train station video of length 23sec (585 frames) with a approximately 20330 pedestrian annotations (multiple pedestrians per frame). The video contains the visual scenarios like pedestrian movement across the frames, scale variation, variable lighting conditions and instances of occlusion thus making it suitable for assessing real-time object detection capabilities of the models considered for evaluation. All models were run on the same video using their pre-trained weights to ensure a fair, out-of-the-box comparison.

3.3 Evaluation Metrics

Metrics used for the comparative analysis of the models are: recall percentage, inference speed and GFLOPs.

Recall**:** is determined based on the number of detections made by the model as given by the equation 1.

$$\text{Recall} = \text{Total number of detections} / \text{Ground truth detections} \quad (1)$$

Ground truth detections for the test video is obtained from the public detections available in the MOT20 dataset which is 20330.

Inference speed: determines the how rapidly the model can process the input frame and generate detections. The inference speed is measured in milliseconds per frame. The inference speed of the model is dependent on the number of fused layers, features learnt by the model, preprocessing speed and also the type of hardware used GPU or CPU [23]. The inference speed per image is as defined by equation 2.

$$tinf = \frac{Ttotal}{N} \quad (2)$$

Where *tinf=inference time per image in ms*

$$Ttotal = total\ inference\ time\ measured\ by\ the\ model$$

N=Number of images processed

GFLOPs: represent the computational cost of a neural network, measuring how many billion floating-point calculations are needed to process an input [24]. The GFLOPs is as provided by equation 3.

$$GFLOPS = \frac{Total\ floating\ point\ operations\ in\ forward\ pass}{10^9} \quad (3)$$

Inference speed was measured during frame-wise processing and reported by the model as the average per-frame inference time. GFLOPs were obtained from model profiling utilities and are independent of dataset content.

The experimentation is conducted using different model sizes of YOLOv9, YOLOv10 and YOLOv11.Model sizes range from small, medium, Large to Extra-large. Different model sizes deliver variable performance. Hence, the comparison is carried out across various model sizes and between different models.

3.4 Experimental Setup

All experiments were conducted on Google Colab, utilizing a cloud-based NVIDIA Tesla T4 GPU with 16 GB VRAM and approximately 12 GB of available system RAM. The development environment was set up using Python, with key libraries including PyTorch, OpenCV and NumPy.

3.4.1 Algorithm 1: YOLO model benchmarking process

Input: Pedestrian Video sequence$V$, Pretrained YOLO model set M

Output: Performance metrics R (Recall, GFLOPs and Inference Speed)

1. Extract frames *F* from video set *V*
2. Preprocess *F* to match the input resolution required by the YOLO models.
3. For each pretrained-model $M_i \in$ {YOLOv8, YOLOv10, YOLOv11 and their variants} do:
   - Pedestrian detection on all frames$F$.
   - Compute evaluation metrics recall
   - Record model-reported inference speed tinf, and GFLOPs.
4. Visualize the detection results for qualitative analysis.
5. Perform comparative analysis across all models.

The algorithm describes the steps for pedestrian detection on an unseen MOT20 video. The performance of YOLO models and their variants are evaluated using recall, inference speed, and GFLOPs, followed by comparative analysis and visualizations of the results.

4. Results and Discussion

This section presents the comparative analysis of the three YOLO series YOLOv9 through YOLOv11. The models are evaluated on an unseen video to ensure an unbiased comparison and assess their capacity to generalize. For each model the analysis is conducted across different model sizes to understand their trade-offs in terms of recall, inference speed, G-flops. The obtained Inference speed and GFLOPs values are on a per image basis. Recall parameter is calculated based on the equation 1 provided in section 3.

4.1 Performance Comparison of YOLOv9, YOLOv10, YOLOv11, and Their Variants

The Table 2 below illustrates the performance differences among various sizes of YOLOv9 when evaluated on an unseen video.

From the Table2 it is observed that YOLOv9-medium size model exhibits higher recall (43.9%) with faster inference speed (24.7ms) than the larger variant YOLOv9-e (recall-41.3%, inference speed-42.3ms). Though YOLOv9-m's GFLOPs (131.2) are higher than YOLOv9-s (39.1) but are lower than YOLOv9-c (237.6) and YOLOv9-e(241.4) reflecting a favorable balance between computational cost and detection accuracy. The enhanced performance of YOLOv9-m is likely due to its deeper backbone and rich feature map capacity.YOLOv9-s provides faster inference(26.3ms) and lesser GFLOPS(39.1) due to its lightweight architecture but suffers from reduced recall(40.9%) as compared to YOLOv9-m. YOLOv9-c and YOLOv9-e with the increased network depth results in higher GFLOPs(~230) but with no improvement in recall over YOLOv9-m.Overall YOLOv9-m effectively balances the detection accuracy, computational efficiency and inference latency making it suitable for real-time applications.

The Table 3 reflects the performance of YOLOv10 models of different sizes when applied on an unseen video.

All YOLOv10 variants achieve faster inference (within ~25ms) and lesser computational cost (within ~100GFLOPs) but recall remains very low within (28%). The faster inference and low computational cost can be attributed to removal of NMS step. The results indicate that YOLOv10 prioritizes efficiency, and larger variants do not yield notable gains in recall despite higher GFLOPs. Although all YOLOv10 variants meets real-time speed requirements, their low recall restrict their practical applicability where detection accuracy is critical.

The Table 4 reflects the performance of YOLOv11 models of different sizes when applied on an unseen video.

Although all YOLOv11 variants achieve faster inference speed (within 30ms) and reasonable computational complexity below 100 GFLOPs (except for YOLOv11-x), YOLOv11-n attains highest recall (40.1%) on an unseen video. The recall value decreases for variants YOLOv11-s to YOLOv11-x approximately from 36% to 32%. This suggests that increased model complexity may not improve generalization capacity due to distributional shift. Therefore, the lightweight model YOLOv11-n generalizes better under domain shift, making itself more suitable for real-time deployments.

4.2 Comparative Analysis of YOLO variants: Accuracy, Efficiency, and Speed

Among the YOLO models, YOLOv9-m achieves the best recall at 43.9% but with moderate computational cost (GFLOPs-131.3) and faster inference speed (24.3ms) making it suitable for applications where detection accuracy is crucial.

Table 2. Performance analysis of YOLOv9 model variants

YOLOv9 variants	Total No of Detections	Inference speed	GFLOPs	Recall
YOLOv9-s	8333	26.3ms	39.1	40.9%
YOLOv9-c	8625	34.9ms	237.6	42.4%
YOLOv9-m	8939	24.7ms	131.3	43.9%
YOLOv9-e	8405	42.3ms	241.4	41.3%

Table 3. Performance analysis of YOLOv10 model variants

YOLOv10 variants	Total No of Detections	Inference speed	GFLOPs	Recall
YOLOv10-n	4939	10.8ms	8.6	24.2%
YOLOv10-s	5417	12.2ms	24.8	26.6%
YOLOv10-m	5685	13.7ms	63.9	27.9%
YOLOv10-b	5417	15.7ms	98.6	26.6%
YOLOv10-l	5032	18.6ms	127	24.7%
YOLOv10-x	5775	25.3ms	170.6	28.4%

Table 4. Performance analysis of YOLOv11 model variants

YOLOv11 variants	Total No of Detections	Inference speed	GFLOPs	Recall
YOLOv11-n	8164	9.6ms	6.5	40.1%
YOLOv11-s	7475	11.6ms	21.5	36.7%
YOLOv11-m	6621	13.7ms	68.0	32.5%
YOLOv11-l	6404	18.8ms	86.9	31.5%
YOLOv11-x	6587	29.8ms	194.9	32.4%

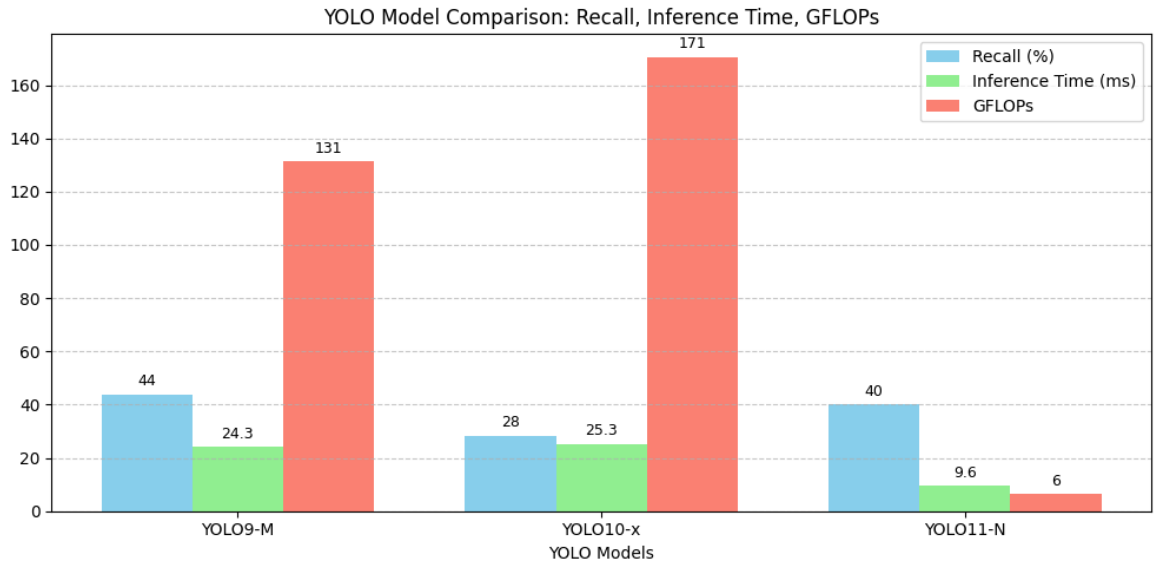

Figure 2. Model comparison using recall, inference speed, and GFLOPs.

YOLOv10-x despite having faster inference (25.3ms) records lowest recall at 28% making it least efficient overall. On the other hand, YOLOv11-n strikes a strong balance at slightly lower recall of 40.1% with high computational efficiency (and faster inference speed (9.6ms) making it highly efficient for real-world

applications. Figure 2 shows a comparative bar chart of the evaluated models with respect to recall, inference speed, and computational complexity (GFLOPs).

4.3 Comparison with Prior Benchmarking Studies

Experimental evaluation shows that YOLOv11-n offers the balanced performance between detection accuracy and computational efficiency, while YOLOv9-m achieves highest recall with moderate computational overhead. YOLOv10, however, exhibits lower recall despite faster inference. These findings align with the previous benchmarking works, which suggest that newer YOLO versions improve efficiency and larger models yield higher detection accuracy as discussed in papers [10, 11]. Performance difference observed for YOLOv10 may be due to domain shifts and architectural modification.

4.4 Visualization Detection Results for selected YOLO models.

The Figure 3a) and Figure 3b) below presents a comparative visualization of the pedestrians detected by the models YOLOv9-m and YOLOv11-n respectively on an unseen MOT20 video.

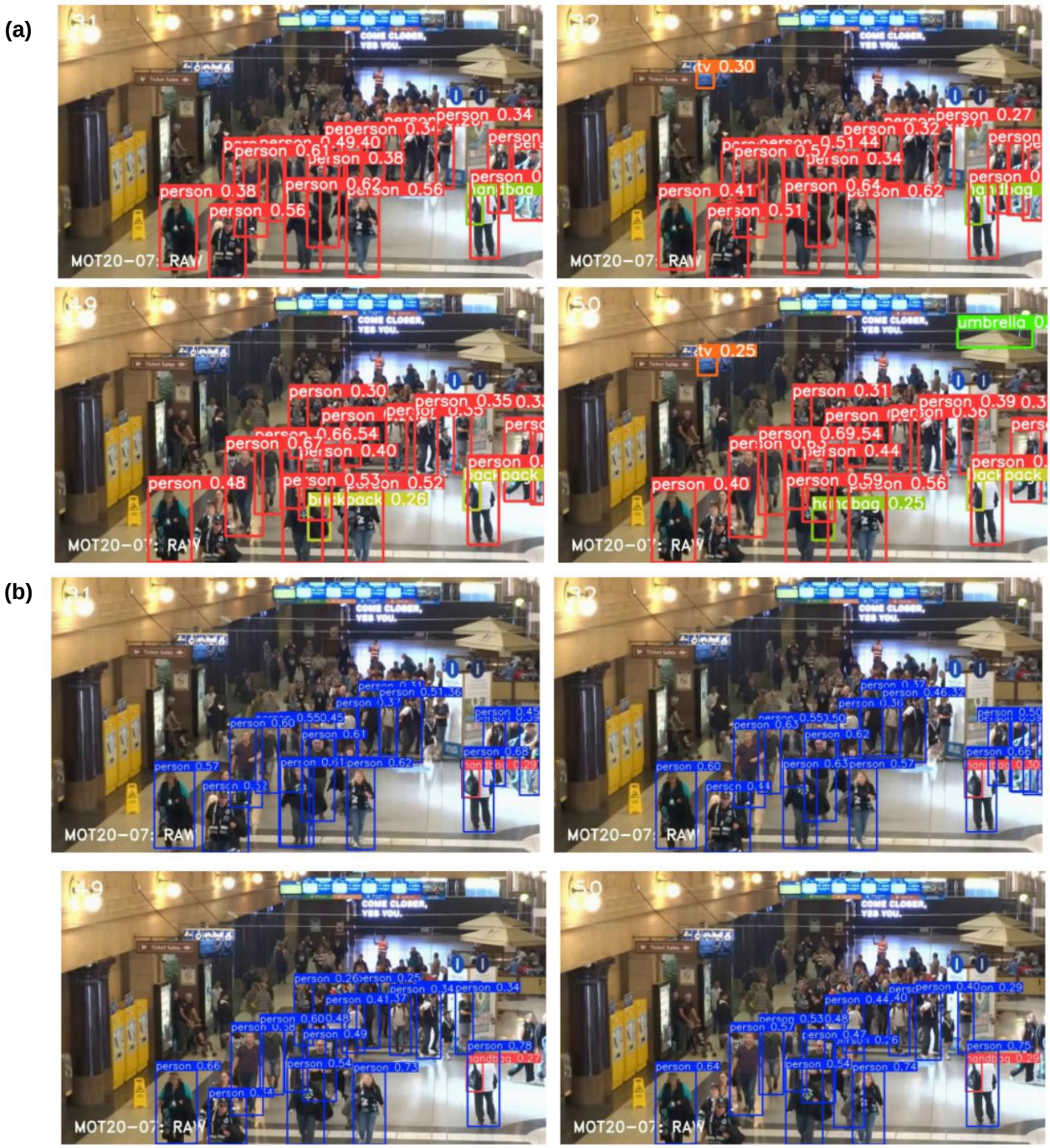

Figure 3. Detections result on an unseen video from MOT20 dataset **(a).** Detections from model-YOLOv9-m, **(b)** Detections from model-YOLOv11-n

The images present identical video frames overlaid with outputs from both models. Visual inspection also supports the fact that YOLOv9-m produces more detections compared to YOLOv11-n. However, the majority of detections from YOLOv11-n are associated with higher confidence.

4.5 Limitations and Future Work

This study evaluates pretrained YOLO models to analyse their generalization in terms of recall, inference speed and GFLOPs over an unseen video. The experiment was performed on a single MOT20 sequence with 585 frames and 20,330 annotated instances. The test limits to single run. Future work can include multi-run statistical validation and evaluation on multiple diverse unseen videos to further assess robustness and generalization.

5. Conclusion

The study compared YOLOv9, YOLOv10 and YOLOv11 models and their variants on an unseen MOT20 video dataset focusing on recall, inference speed and computational efficiency GFLOPs. The results suggest that generalization on unfamiliar data largely depends on maintaining balance between detection accuracy and computational efficiency. YOLOv9-m demonstrates stronger generalization, while YOLOv11-n achieves an effective balance between detection performance and computational efficiency. These characteristics make both the models suitable for real time applications depending on deployment constraints. The lower recall observed for YOLOv10 suggests that, optimization primarily focused on reducing computational cost may limit robustness on unfamiliar data.

From a deployment standpoint YOLOv9-m is more suitable for scenarios where accuracy is critical, where-as YOLOv11-n is more suitable for resource constrained environments. Overall, this work provides practical recommendations for selecting appropriate object detection models based on deployment needs and generalization priorities. To intensify real -world applicability, future work can consider minimal additional training for both the models incorporating lightweight fine tuning under demanding conditions such as occlusion, illumination changes, pose variation and background clutters. This can help in enhancing robustness under demanding operating conditions without significantly increasing computational needs. Using these approaches will allow object detection models to maintain high performance in real-world scenarios and adapt effectively to dynamic or unpredictable conditions.

References

[1] D. Nimma, O. Al-Omari, R. Pradhan, Z. Ulmas, R.V.V. Krishna, Ts. Yousef A. Baker El-Ebiary, V.S. Rao, Object detection in real-time video surveillance using attention based transformer-YOLOv8 model. Alexandria Engineering Journal, 118, (2024) 482-495. https://doi.org/10.1016/j.aej.2025.01.032

[2] C. Jiang, H. Ren, X. Ye, J. Zhu, H. Zeng, Y. Nan, M. Sun, X. Ren, H. Huo, Object detection from UAV thermal infrared images and videos using YOLO models. International Journal of Applied Earth Observation and Geoinformation, 112, (2021) 102912. https://doi.org/10.1016/j.jag.2022.102912

[3] B. Ganga, B.T. Lata, K.R. Venugopal, Object detection and crowd analysis using deep learning techniques: Comprehensive review and future directions. Neurocomputing, 597, (2024) 127932. https://doi.org/10.1016/j.neucom.2024.127932

[4] W. Chen, Y. Zhu, Z. Tian, F. Zhang, M. Yao, Occlusion and multi-scale pedestrian detection A review. Array, 19, (2023) 100318. https://doi.org/10.1016/j.array.2023.100318

[5] J. Tang, H. Lai, G. Gao, T. Wang, FEL-Net: A lightweight network to enhance feature for multi-scale pedestrian detection. Journal of King Saud University-Computer and Information Sciences, 36(8), (2024) 102198. https://doi.org/10.1016/j.jksuci.2024.102198

[6] Z.Q. Zhao, P. Zheng, S.T. Xu, X. Wu, Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 30(11), (2019) 3212 – 3232. https://doi.org/10.1109/TNNLS.2018.2876865

[7] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep learning for generic object detection: A survey. International journal of computer vision, 128(2), (2020) 261-318.

[8] M. Hussain, R. Khanam, In-Depth Review of YOLOv1 to YOLOv10 Variants for Enhanced Photovoltaic Defect Detection. Solar, 4(3), (2024) 351-386. https://doi.org/10.3390/solar4030016

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

[10] C.Y. Wang, H.Y. M. Liao, YOLOv1 to YOLOv10: The fastest and most accurate real-time object detection systems. APSIPA Transactions on Signal and Information Processing, 13(1), (2024) 1–38. https://doi.org/10.1561/116.20240058

[11] M. Hussain, (2024) Yolov5, yolov8 and yolov10: The go-to detectors for real-time vision. arXiv preprint arXiv:2407.02988.

https://doi.org/10.48550/arXiv.2407.02988

[12] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, L. Leal-Taixé, (2020) Mot20: A benchmark for multi object tracking in crowded scenes. arXiv preprint arXiv:2003.09003. https://doi.org/10.48550/arXiv.2003.09003

[13] B. Karbouja, A. Garabet, J. Topalian-Rivasa, J. Kruger, Comparative Performance Evaluation of One-Stage and Two-Stage Object Detectors for Screw Head Detection and Classification in Disassembly Processes. Procedia CIRP, 122, (2024) 527-532. https://doi.org/10.1016/j.procir.2024.01.077

[14] J. Anandakrishnan, A.K. Sangaiah, H. Darmawan, N.K. Son, Y.B. Lin, M. J.F. Alenazi, Precise Spatial Prediction of Rice Seedlings From Large Scale Airborne Remote Sensing Data Using Optimized Li-YOLOv9. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 18, (2024) 2226 – 2238. https://doi.org/10.1109/JSTARS.2024.3505964

[15] A. Sharma, V. Kumar, L. Longchamps, Comparative performance of YOLOv8, YOLOv9, YOLOv10, YOLOv11 and Faster R-CNN models for detection of multiple weed species. Smart Agricultural Technology, 9, (2024) 100648. https://doi.org/10.1016/j.atech.2024.100648

[16] C.Y. Wang, I.H. Yeh, H.Y.M. Liao, (2024) YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. Computer Vision – ECCV2024, Lecture Notes in Computer Science, Springer, Cham. https://doi.org/10.1007/978-3-031-72751-1_1

[17] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, YOLOv10: Real-time end-to-end object detection. Advances in Neural Information Processing Systems, 37, (2024) 107984-108011. https://doi.org/10.52202/079017-3429

[18] R. Khanam, M. Hussain, (2024) YOLOv11: An overview of the key architectural enhancements. arXiv preprint arXiv:2410.17725, https://doi.org/10.48550/arXiv.2410.17725

[19] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection. IEEE Conference on Computer Vision Pattern Recognition (CVPR), (2016) 779–788. https://doi.org/10.1109/CVPR.2016.91

[20] A. Bochkovskiy, C.Y. Wang, H.Y.M. Liao, (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934

[21] C.Y. Wang, A. Bochkovskiy, H.Y. M. Liao, YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. IEEE/CVF Conference Computer Vision Pattern Recognition (CVPR), IEEE, Canada. https://doi.org/10.1109/CVPR52729.2023.00721

[22] V. Afifah, S. Erniwati, YOLOv8 for object detection: A comprehensive review of advances, techniques, and applications. International Journal of Advanced Computing and Informatics, 2(1), (2026) 53–61. https://doi.org/10.71129/ijaci.v2i1.pp53-61

[23] R. Sapkota, Z. Meng, M. Churuvija, X. Du, Z. Ma, M. Karkee, (2024). Comprehensive performance evaluation of YOLO11, YOLOv10, YOLOv9, and YOLOv8 on detecting and counting fruitlet in complex orchard environments. Agriculture Communications. https://doi.org/10.32388/E9Y7XI

[24] Z. Qi, H. Kongfa, W. Tianshu, Y. Tao, Lightweight and polarized self-attention mechanism for abnormal morphology classification algorithm during traditional Chinese medicine inspection. Digital Chinese Medicine, 7(3), (2024) 256-263. https://doi.org/10.1016/j.dcmed.2024.12.005

Acknowledgement

I would like to express my very great appreciation to the co-authors of this manuscript for their valuable and constructive suggestions during the planning and development of this research work.

Authors Contribution Statement

T.G. Vibha. Conceptualization, Methodology, Software, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization. S Theodore Chandra: Resources, Supervision. S. Sivaramakrishnan: Supervision, Project administration. All the authors have read and agreed to the published version of the manuscript.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes