ASM Sc. J., 20(1), 2025
https://doi.org/10.32802/asmscj.2025.1977

YOLOv10 Algorithm for Real-Time Pedestrian Detection in Autonomous Vehicles

Y. Li[1,2] and W.Y. Leong[2*]

[1]*Heilongjiang Institute of Construction Technology, Harbin, Hei Longjiang Province, China*
[2]*Faculty of Engineering and Quantity Surveying, INTI International University, Nilai Malaysia*

Autonomous driving safety requires accurate pedestrian detection. This study introduces real-time pedestrian detection based on the YOLOv10 algorithm. Adding EfficientNet, C2F-DM, BiFormer, and a multi-scale feature fusion detection head to the backbone, neck, and multi-scale networks creates a real-time object detection model. Experiments demonstrate that YOLOv10 can detect multi-scale pedestrians in complicated settings. The implementation of YOLOv10 for pedestrian detection in Autonomous Vehicles advances the field of intelligent transportation systems and contributes to the broader goal of creating safer, more efficient autonomous driving technologies. Future work includes refining the algorithm for multi-object detection, reducing false positives, and enhancing robustness against environmental variability.

Keywords: YOLOv10; Autonomous Vehicles; Real-Time Pedestrian Detection; EfficientNet; BiFormer; process innovation

I. INTRODUCTION

Autonomous driving technology has advanced intelligent transportation systems, with pedestrian detection essential for driving safety. From Faster R-CNN to YOLOv10, target detection techniques have significantly advanced. Since 2015, Faster R-CNN has used region proposal networks and convolutional neural networks to recognise targets precisely (Leong, Leong & San Leong, 2024). Since 2016, the YOLO series algorithms have improved model structures and training methodologies, increasing detection speed and accuracy (Redmon & Farhadi, 2018). In addition, SSD and RetinaNet algorithms have solved problems in several dimensions (Tan, Pang & Le, 2020). The newest EfficientDet balances accuracy and speed for real-time pedestrian identification in complicated situations (Leong, 2022).

This paper presents real-time YOLOv10 pedestrian detection with multi-scale feature fusion and edge detection, enhancing accuracy and robustness with implications for autonomous driving, intelligent surveillance, and unmanned aircraft (Leong, 2019). (Mandic, Souretis, Leong, Looney, Van Hulle, & Tanaka, 2008).

II. LITERATURE REVIEW

From Faster R-CNN to YOLOv10, object detection techniques have advanced their ability to optimize structures and training approaches (Patil, Nawade, Nagarkar & Kadale, 2024).

A. Faster R-CNN

In 2015, Faster R-CNN was proposed for high-precision detection. It combines a Convolutional Neural Network (CNN) with a Region Proposal Network (RPN). CNN manages object categorisation and bounding box regression while RPN develops candidate regions, improving detection accuracy (Hussain & Khanam, 2024). Due to its complex computational method and high resource requirements, Faster R-CNN may not be suitable for real-time applications.

B. YOLO Series

From YOLOv1 to YOLOv10, the YOLO (You Only Look Once) series has improved detection speed and accuracy.

*Corresponding author's e-mail: waiyie@gmail.com

The YOLO series' end-to-end detection approach localises and classifies objects in one forward pass, lowering detection time (Alif & Hussain, 2024). Model structures and training methods are optimised in each YOLO version:

- YOLOv1: Single-pass detection, faster but struggles with small objects and complex scenes.
- YOLOv2 and YOLOv3: Improved accuracy and small object detection with multi-scale features.
- YOLOv4: Enhanced accuracy and speed with Bag of Freebies and Specials.
- YOLOv5 to YOLOv10: Optimised architecture, better feature extraction, and inference for complex environments.

C. SSD and RetinaNet

SSD (Single Shot MultiBox Detector) and RetinaNet are two crucial algorithms in object detection, each offering efficient solutions in various scalings (Mingxing, Ruoming & Le Quoc, 2020).

- SSD: SSD performs multi-scale detection for high-speed tasks, which is ideal for real-time applications but struggles with small object detection.
- RetinaNet: The Focal Loss function was introduced to mitigate the class imbalance issue, enhancing object detection accuracy and rendering it appropriate for intricate task settings.

D. EfficientDet

EfficientDet combines EfficientNet and BiFPN technologies to strike a harmonious equilibrium between precision and efficiency, making it well-suited for real-time applications in complex real-world settings. The evolution of algorithms demonstrates improvements in precision, speed, and computing efficiency (Tan *et al.*, 2024).

III. IMPROVED YOLOV10 MODEL ARCHITECTURE

To enhance the precision and robustness of pedestrian identification, YOLOv10 includes enhancements to the neck network, detecting head and backbone network (Wang *et al.*, 2024) (Li, Leong & Zhang, 2024). The network architecture of the EfficientNet-YOLOv10 model is seen in Figure 1.

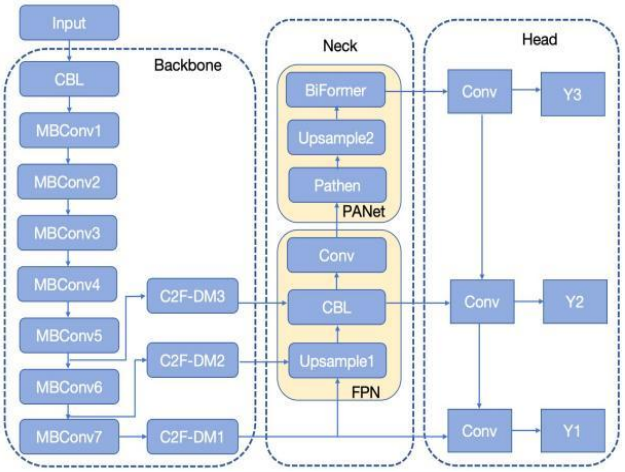

Figure 1. Network Structure of EfficientNet-YOLOv10 Model

A. Backbone Network Optimisation

Integration of C2F-DM modules and an enhanced EfficientNet topology enhances feature extraction in the backbone network of YOLOv10 (Wan *et al.*, 2018).

1. EfficientNet Structure

EfficientNet optimises convolutional layer depth, breadth, and resolution for feature extraction. The formula is as follows:

$$FLOPs \propto d \cdot w^2 \cdot r^2 \quad (1)$$

where d is the network depth, w is the network width, and r is the input image resolution.

This three-dimensional scaling is achieved utilising the compound scaling factor:

$$d = \alpha^{\phi},\ w = \beta^{\phi},\ r = \gamma^{\phi} \quad (2)$$

where α, β, γ are constants satisfying $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$.

2. C2F-DM Module

The C2F-DM module takes feature extraction to the next level by including dilated convolutions and cross-stage partial networks. The C2F-DM module is architecturally structured as follows:

$$\mathrm{C2F-DM}(x) = x + \mathrm{Conv}(x) + \mathrm{DilatedConv}(x) \quad (3)$$

where Conv(x) illustrates a standard convolution operation, a DilatedConv(x) depicts a dilated convolution procedure.

B. Feature Fusion Module

The enhanced YOLOv10 model incorporates the BiFPN (Bidirectional Feature Pyramid Network) to significantly extend the feature extraction capabilities. By employing weighted bidirectional feature fusion, BiFPN successfully combines features at various scales, thereby enhancing the resilience and precision of target identification. The computing procedure is outlined below:

$$P_i = \sum_j w_{ij} p_j \quad (4)$$

where P_i is the fused feature map, P_j is the input feature map, and w_ij is the weight coefficient, with all weights normalised such that $\sum_j w_{ij} p_j = 1$.

C. Detection Head Optimisation

The enhanced YOLOv10 model integrates optimisation techniques from EfficientDet into its detection head, resulting in precise identification by combining features at several scales (Janocha & Czarnecki, 2017). More precisely, the subsequent equations are employed for object categorisation and bounding box regression.

Classification Loss:

$$L_{cls} = -\sum_{i=1}^{N} (y_i \log(\hat{p_i}) + (1 - y_i) \log(1 - \hat{p_i})) \quad (5)$$

Bounding Box Regression Loss (Rezatofighi *et al.*, 2019):

$$L_{\mathrm{bbox}} = \sum_{i=1}^{N} \mathrm{Smooth}_{L1}(t_i - \hat{t_i}) \quad (6)$$

where y_i is the actual class, $\hat{p}_i$ is the predicted class probability, t_i is the proper bounding box parameter, $\hat{t}_i$ is the predicted bounding box parameter, and N is samples' number.

IV. EXPERIMENTS

A. Experimental Setup

To evaluate the practical application and performance of the improved YOLOv10 algorithm, we conducted experiments under controlled conditions using the same datasets and hardware configurations. The datasets used in the experiments include COCO, KITTI, and VOC, which are widely recognised benchmarks for object detection tasks. The hardware configuration for all experiments consisted of an NVIDIA Tesla V100 GPU, 32GB RAM, and an Intel Xeon E5-2698 v4 CPU, ensuring consistency in computational resources across all tested algorithms.

B. Experimental Data

The datasets utilised in the tests encompass a wide array of photographs with different degrees of intricacy, ranging from uncomplicated backgrounds to extensively patterned metropolitan landscapes. Specifically:

- COCO Dataset: Tests object identification capabilities using ordinary scenarios and items.
- KITTI Dataset: Images from autonomous cars in motion, including pedestrians, automobiles, and other essential items.
- VOC Dataset: Provides annotated item images and measures detection accuracy

C. Algorithms Compared

To assess the performance impact of YOLOv10, EfficientDet, Faster R-CNN, SSD, YOLOv3, YOLOv4, and YOLOv5 were evaluated (Henderson & Ferrari, 2017).

D. Performance Metrics

These indicators were used to assess how well each algorithm performed.

Mean Average Precision (mAP): Compares the identified items' accuracy to the markings on the ground to determine their correctness. It can be calculated using the following formula (Koslowsky *et al.*, 2006):

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i \quad (7)$$

where C is the total number of $i-th$ categories. AP_i is the average precision for the category (Ahmad & Rahimi, 2024).

Frames Per Second (FPS): Indicates algorithm speed for real-time applications. The calculation formula is as follows:

$$FPS = \frac{N}{T} \quad (8)$$

where N is the total number of frames processed, T is the total time to process these frames (in seconds).

Recall: Evaluates the algorithm's ability to identify all the photos' relevant things (Leong, Leong, & San Leong, 2024). The calculation formula is as follows:

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

where TP is the number of correctly detected objects. FN is the number of missed objects.

Inference Time: Refers to the time required for a trained deep-learning model to process new input data (such as images or text) and generate prediction results. The calculation formula is as follows:

$$T_{inference} = \frac{T_{total}}{N} \quad (10)$$

where $T_{\text{inference}}$ represents the inference time per sample. T_{total} represents the total inference time for processing N samples. N is the number of missed objects (Han *et al.*, 2015).

V. RESULT AND DISCUSSION

Within the experimental portion, we assessed the performance of different algorithms on these datasets by computing four quantitative metrics: mAP, FPS, Recall, *and Inference Time*.

Figure 2 illustrates the performance of each method.

A. Analysis of Results

- *mAP:* YOLOv10 achieved the highest *mAP*, indicating the best performance in detection accuracy. SSD has the lowest *mAP,* showing a particular shortfall in detection accuracy.
- *FPS:* YOLOv10 has the highest *FPS,* demonstrating a significant advantage in real-time processing, making it suitable for scenarios requiring high-speed processing.
- *Recall:* YOLOv10 also has the highest recall rate, meaning it detects more objects with fewer misses. SSD has a relatively low recall rate, indicating it may miss more objects during detection.
- I*nference Time:* Faster R-CNN exhibits a longer inference time, which renders it unsuitable for applications requiring rapid responses. Conversely, YOLOv10 demonstrates the shortest inference time, making it well-suited for real-time tasks such as autonomous driving.

B. Discussion

The experimental results show that the improved YOLOv10 outperforms other models in terms of mAP, FPS, and Inference Time, making it the top choice for real-time object detection. Higher mAP, FPS, and shorter Inference Time demonstrate robustness and versatility.

EfficientDet balances *mAP, FPS*, and *Inference Time*, making it suitable for high-performance applications, but it is more complex than YOLO models. SSD is fast with short Inference Time but has lower *mAP,* while Faster R-CNN is accurate but has a more extended Inference Time, making it unsuitable for real-time applications.

YOLOv10 is the best choice for high-accuracy, real-time applications, especially in scenarios with low tolerance for missed detections. For offline tasks requiring high accuracy without real-time demands, Faster R-CNN is still a solid option. For resource-limited applications like embedded devices, SSD provides a lightweight solution.

Optimising YOLOv10 and similar models can boost performance in complex scenarios. Future research may enhance these models by adding features or applying advanced training techniques (Leong, 2003; Leong, 2002).

Although YOLOv10 demonstrates exceptional performance across various datasets, its efficacy may diminish under certain conditions:

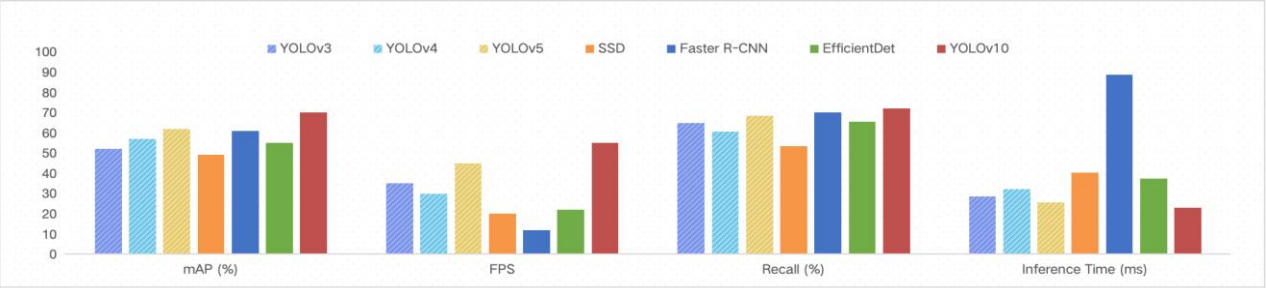

Figure 2. Performance Comparison of Object Detection Algorithms

- Complex Backgrounds: Under conditions of low contrast or inherent visual complexity, the model may struggle to distinguish the item from its immediate environment.
- Rapidly Changing Scenes: The real-time object identification capability of YOLOv10 may be subject to challenges in highly dynamic situations characterised by quick changes, which could result in missed detections or delayed responses.
- Resource-Constrained Edge Devices: Even with the enhancements implemented for edge computing in this work, YOLOv10 may still require assistance to achieve the necessary inference speed and accuracy in highly resource-constrained settings, restricting its suitability for real-time applications.

VI. FUTURE PROSPECTS AND DEVELOPMENT

Recent advances in object detection algorithms offer several promising research and improvement opportunities (Li, Zhang & Liu, 2024). For YOLOv10 and similar models, the following are several promising directions to explore:

- Time Series Analysis and Spatio-Temporal Feature Fusion: Integrating more contextual information into the model improves object detection, especially in autonomous driving and surveillance (Leong, 2024; 2025e).
- Reinforcement Learning and Adaptive Detection Mechanisms: Future studies can use reinforcement learning (RL) to construct adaptive detection systems that improve YOLOv10's performance in complex and unfamiliar contexts, increasing its practicality.
- Multi-Modal Sensor Data Fusion: Infrared imaging, audio sensors, and depth sensors (e.g., LiDAR) may be integrated with visual data in future research. These sensors can capture data from multiple dimensions, increasing detection in fog or poor light.
- Personalised Detection and Model Adaptive Evolution: Customised detection systems using online learning and model adaptive evolution may be researched in the future. This would allow YOLOv10 to dynamically alter model parameters based on user needs or scenarios, expanding its possibilities.

By embarking on these novel research avenues, the detection efficacy of YOLOv10 and associated models in intricate settings can be enhanced, broadening their range of applications and allowing them to showcase robust capabilities in more practical situations.

VII. REFERENCES

Ahmad, H. M & Rahimi, A 2024, 'SH17: A Dataset for Human Safety and Personal Protective Equipment Detection in Manufacturing Industry'. arXiv preprint arXiv:2407.04590.

Alif, MAR & Hussain, M 2024, 'YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain'. arXiv preprint arXiv:2406.10139.

Han, S, Pool, J, Tran, J & Dally, W 2015, 'Learning both weights and connections for efficient neural networks', Advances in Neural Information Processing Systems, vol. 28.

Henderson, P & Ferrari, V 2017, 'End-to-end training of object class detectors for mean average precision', in Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V, Springer International Publishing, vol. 13, pp. 198-213.

Hussain, M & Khanam, R 2024, 'In-depth review of yolov1 to yolov10 variants for enhanced photovoltaic defect detection', Solar, vol. 4, no. 3, pp. 351-386.

Janocha, K & Czarnecki, WM 2017, 'On loss functions for deep neural networks in classification', arXiv preprint arXiv:1702.05659.

Koslowsky, B, Jacob, H, Eliakim, R & Adler, SN 2006, 'PillCam ESO in esophageal studies: improved diagnostic yield of 14 frames per second compared with 4 fps', Endoscopy, vol. 38, no. 1, pp. 27-30.

Leong, WY, editor 2024, Industry 5.0: Design, standards, techniques and applications for manufacturing, Institution of Engineering and Technology.

Leong, WY 2025a, Generative AI-Powered Traffic and Mobility Solutions for Next-Generation Smart Cities, IEEE International Conference on Consumer Electronics – Taiwan (ICCE-TW), 2025, July 16-18, Kaohsiung, Taiwan.

Leong, WY 2025b, Soft Robotics: Engineering Flexible Automation for Complex Environments, Engineering Proceedings 2025, vol. 92, p. 65. doi: 10.3390/engproc2025092065

Leong, WY, Leong, WY, Kumar, R 2025c, 'Green Mobility Solutions through Intelligent Fleet Management and Smart Logistics', International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI2025), India, Jan 16-18, 2025.

Leong, WY 2025d, 'Digital Twin Models for Real-Time Failure Prediction in Industrial Machinery, ASM Science Journal, vol. 20, no. 1.

Leong, WY, Homer, J 2002, 'Hop selection in peer-to-peer WPAN networks', in ICCS 2002: 8th International Conference on Communications Systems, Vols. 1 and 2, Proceedings 2002 Jan 1, IEEE, vol. 2, pp. 870-872.

Leong, WY & Homer, J 2003, 'Enhancing interference mitigation in communication', Fourth International Conference on Information, Communications and Signal Processing 2003 and the Fourth Pacific Rim Conference on Multimedia, Proceedings of the 2003 Joint, Singapore, vol. 1, pp. 587-591.

Li, Y, Leong, W & Zhang, H 2024, 'YOLOv10-Based Real-Time Pedestrian Detection for Autonomous Vehicles', 2024 IEEE 8th International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, pp. 1-6. doi: 10.1109/ICSIPA62061.2024.10686546.

Li, Y, Zhang, H & Liu, C 2024, 'Teaching Reform and Practice of Mechanical Manufacturing and Automation Courses in the Context of Intelligent Manufacturing. In Proceedings of the 3rd International Conference on Educational Innovation and Multimedia Technology, EIMT 2024, March 29–31, 2024, Wuhan, China.

Leong, WY, Leong, WY, Kumar, R 2025e, 'IoTs applications in Supply Chain Management, International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI2025), Jan 16-18, 2025.

Mandic, D, Souretis, G, Leong, WY, Looney, D, Van Hulle, MM & Tanaka, T 2008, 'Complex empirical mode decomposition for multichannel information fusion', in Signal Processing Techniques for Knowledge Extraction and Information Fusion, pp. 243-260.

Mingxing, T, Ruoming, P & Le Quoc, VE 2020, 'Scalable and Efficient Object Detection', in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Patil, VK, Nawade, P, Nagarkar, R & Kadale, P 2024, 'Object Detection and Tracking Face Detection and Recognition', in Integrating Metaheuristics in Computer Vision for Real-World Optimization Problems, pp. 25-54.

Redmon, J & Farhadi, A 2018, 'YOLOv3: An Incremental Improvement', arXiv preprint arXiv:1804.02767.

Rezatofighi, H, Tsoi, N, Gwak, J, Sadeghian, A, Reid, I. & Savarese, S 2019, 'Generalized intersection over union: A metric and a loss for bounding box regression', in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658-666.

Tan, L, Liu, S, Gao, J, Liu, X, Chu, L & Jiang, H 2024, 'Enhanced Self-Checkout System for Retail Based on Improved YOLOv10', arXiv preprint arXiv:2407.21308.

Tan, M, Pang, R & Le, QV 2020, 'EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781-10790.

Wan, W, Zhong, Y, Li, T & Chen, J 2018, 'Rethinking feature distribution for loss functions in image classification', in

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9117-9126.

Wang, A, Chen, H, Liu, L, Chen, K, Lin, Z, Han, J & Ding, G 2024, 'YOLOv10: Real-time end-to-end object detection', arXiv preprint arXiv:2405.14458.