ENHANCED YOLOV8 WITH BIFPN AND MHSA FOR TRAFFIC VEHICLE DETECTION

Ngo Truong An[1,2]*, Huynh Huu Hung[3], Tran Thi Hoang Oanh[2]

[1]*Master's Student, Computer Science, The University of Danang - University of Science and Technology, Vietnam*
[2]*Dong A University, Vietnam*
[3]*The University of Danang - University of Science and Technology, Vietnam*

*Corresponding author: anngo270398@gmail.com

(Received: April 15, 2025; Revised: June 17, 2025; Accepted: June 19, 2025)

DOI: 10.31130/ud-jst.2025.23(9D).558E

Abstract - Accurately detecting vehicles in urban traffic scenarios is a complex task, especially when dealing with cluttered backgrounds, diverse object scales, and high vehicle density. In this study, we propose an improved YOLOv8-based model tailored for vehicle detection in such challenging environments. The enhancement lies in the integration of a Bidirectional Feature Pyramid Network (BiFPN), which boosts multi-scale feature fusion, and a Multi-Head Self-Attention (MHSA) module, designed to strengthen the model's capacity to understand broader spatial context. Together, these components help the model better distinguish between densely arranged vehicles. We conducted in-depth experiments on the Vehicles-COCO dataset, and the results demonstrate that our YOLOv8-BiFPN-MHSA variant outperforms the original YOLOv8 not only in Precision but also in mAP. Our model achieves significantly higher mAP@0.5 and mAP@0.5:0.95, along with an overall improvement in detection performance. These enhancements highlight the stability, efficiency, and strong potential for real-world traffic monitoring systems.

Key words - YOLOv8 architecture; object detection; bidirectional feature pyramid network (BiFPN); multi-head self-attention (MHSA).

1. Introduction

In recent years, intelligent transportation systems (ITS) have increasingly leveraged advanced computer vision techniques to support urban planning, traffic regulation, and public safety. A core task in this domain is real-time, accurate vehicle detection in complex environments. However, traditional methods based on handcrafted features or shallow models often fail in scenes with heavy occlusion, cluttered backgrounds, or small-scale objects [1].

The rise of deep learning, especially convolutional neural networks (CNNs), has significantly boosted object detection by improving feature extraction and classification [2]. Among them, the YOLO (You Only Look Once) series stands out for its balance between speed and accuracy. The latest version, YOLOv8, features an optimized backbone and detection head, making it suitable for real-time vehicle detection [3], [4]. Still, YOLOv8 faces challenges in detecting small or partially occluded vehicles in dense traffic, mainly due to limited multiscale feature fusion and lack of effective attention mechanisms [5].

To overcome these issues, we propose an improved YOLOv8 model that incorporates two key components: Bidirectional Feature Pyramid Network (BiFPN) and Multi-Head Self-Attention (MHSA). BiFPN enables better multiscale feature fusion through learnable weights [6], while MHSA - originating from Transformer-based models - captures long-range dependencies and spatial context, aiding in the detection of crowded or occluded vehicles [7].

Our enhanced model, YOLOv8-BiFPN-MHSA, shows significant performance gains. On the Vehicles-COCO dataset, it achieves a mAP@0.5 of 73.1%, outperforming baseline YOLOv8 (71.2%) by 1.9 percentage points. Improvements are also noted in mAP@0.5:0.95 and precision for medium and small objects, indicating stronger generalization under various traffic conditions [8].

2. Related works

The growing complexity of urban transportation and the surge in vehicle volume pose major challenges for real-time traffic monitoring, particularly in detecting small, occluded, or overlapping vehicles. Traditional methods like background subtraction, motion tracking, and handcrafted-feature classifiers have limited robustness in such scenarios [9] - [11]. As a result, deep learning-based computer vision approaches, especially those using CNNs, are increasingly favored [12].

Recent developments aim to optimize detection models for better speed and accuracy in real-time applications. Techniques such as depthwise separable convolutions and lightweight attention modules like More Efficient Channel Attention (MECA) help reduce parameters and computation while preserving strong feature extraction [13], [14]. To retain fine details in small vehicles lost during downsampling, modules like Atrous Spatial Pyramid Fast (ASPF) use dilated convolutions with varied receptive fields to enhance context and spatial sensitivity [13], [14].

Alongside network improvements, new feature fusion strategies like Bidirectional Feature Pyramid Network (BiFPN) enhance multiscale object handling. BiFPN introduces weighted fusion and better cross-scale connectivity, combining shallow and deep features for more robust detection across varying object sizes and densities.

Large-scale, real-world traffic datasets also play a critical role by enabling comprehensive model evaluation under diverse conditions, such as lighting, weather, and traffic density [15] - [17].

In conclusion, advancements in deep learning - through architectural optimization, feature fusion (e.g., BiFPN),

attention mechanisms (e.g., MHSA), and realistic datasets - are propelling the development of efficient, scalable real-time vehicle detection systems for complex urban environments.

3. Proposed Approach

In this section, we describe in detail the proposed improvements to the YOLOv8 architecture for traffic vehicle detection in complex urban environments. The enhanced model architecture referred to as YOLOv8-BiFPN-MHSA is built upon the YOLOv8n baseline and integrates two key modules: the Bidirectional Feature Pyramid Network (BiFPN) and the Multi-Head Self-Attention (MHSA) mechanism. The following subsections provide a comprehensive explanation of the improved YOLOv8n model structure, followed by a detailed discussion of the MHSA module and the BiFPN-based feature fusion strategy.

3.1. YOLOv8 Improved Model

In the fast-evolving field of object detection, the YOLO (You Only Look Once) series has become a benchmark for balancing speed and accuracy [19]. The version YOLOv8, has gained attention for its architectural upgrades and improved performance [19]. Available in variants - YOLOv8n,s, m, l, and x - it addresses a range of computational needs [19]. This study uses YOLOv8n, the lightweight version, as the baseline due to its efficiency for real-time applications. Its architecture is shown in Figure 1.

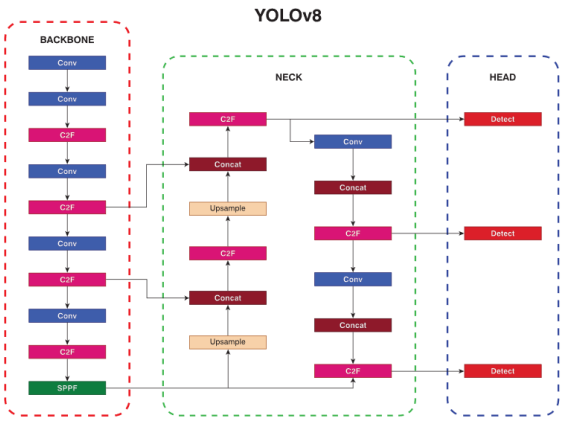

Figure 1. *YOLOv8n network structure*

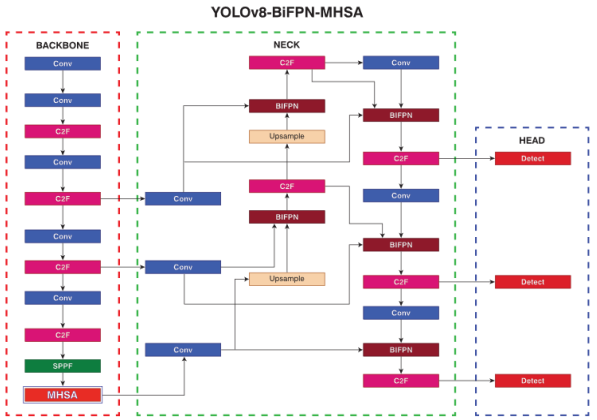

Figure 2. *Improved YOLOv8n network structure*

Although YOLOv8n delivers strong overall performance, it struggles with detecting small or partially obscured vehicles in dense traffic. To address this, we propose two key enhancements, illustrated in Figure 2. First, the Multi-Head Self-Attention (MHSA) module is integrated to help the model focus on small-scale objects and capture spatial dependencies [19]. Second, the Bidirectional Feature Pyramid Network (BiFPN) is added to improve multiscale feature fusion, enhancing both semantic understanding and fine detail extraction [6]. Combined, these improvements significantly enhance the model's ability to detect vehicles of various sizes in complex scenes.

3.2. Multi-Head Self-Attention Mechanism

Drawing inspiration from the principles of multi-head self-attention in Transformer architectures, the MHSA mechanism enables the model to capture long-range dependencies between different spatial positions within an image. Specifically, MHSA computes interactions across spatial locations by projecting the input feature map into three distinct spaces: Query (Q), Key (K), and Value (V), using 1x1 convolutional layers for linear transformations [20].

These projections are defined as:

$$Q = Conv2d_Q(X),\ K = Conv2d_K(X),\ V = Conv2d_V(X) \quad (1)$$

where $X \in R^{RxCxWxH}$ denotes the input feature map with batch size B, channel dimension C, and spatial dimensions W and H.

Following this, each of the projected matrices Q, K, and V is reshaped to separate the feature dimension into multiple heads, with each head capturing different subspaces of representation [21]:

$$Q, K, V \in R^{B \times h \times \frac{C}{h} \times (W \times H)} \quad (2)$$

where B is the batch size, h is the number of attention heads, C is the total number of feature channels, W is the width, and H is the height of the input feature map. The term $W \times H$ corresponds to the spatial resolution after flattening the 2D feature map into a sequence. This reshaping aligns with the multi-head self-attention mechanism implemented in our architecture.

The attention energy E between queries and keys is computed via matrix multiplication:

$$E = Q^T \times K \quad (3)$$

Optionally, relative positional embeddings can be incorporated into the attention scores to enhance the model's ability to capture positional information. These embeddings are learned parameters [20]:

$$E' = E + RelPos(Q) \quad (4)$$

where RelPos(Q) is computed by combining learned horizontal and vertical positional encodings.

The attention weights A are obtained by applying a softmax function over the last dimension of the energy tensor to ensure proper normalization:

$$A = Softmax(E') \quad (5)$$

Subsequently, the attention output is produced by

performing another matrix multiplication between the value matrix V and the normalized attention weights:

$$O = V \times A^T \quad (6)$$

Finally, the output is reshaped back to the original spatial dimensions:

$$X\sim \in R^{B \times C \times W \times H} \quad (7)$$

3.3. Bidirectional Feature Pyramid Network

The original YOLOv8 architecture employs a neck design that combines the strengths of the Feature Pyramid Network (FPN) and Path Aggregation Network (PAN), as illustrated in Figure 3a and Figure 3b. FPN enhances semantic representation by transmitting deep, high-level features to shallow layers, while PAN complements this by channeling low-level spatial information from shallow to deeper layers. Together, they form the PANet structure, which enables effective fusion of features across multiple scales and supports improved object detection [22].

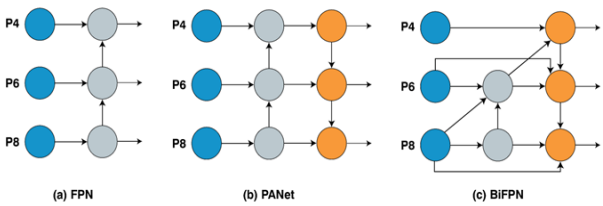

Figure 3. *Neck feature network design: (a) FPN, (b) PANet, and (c) BiFPN*

Despite its strengths, our investigation revealed a key limitation in the PANet-based design: critical features from the backbone may be diluted or lost during the FPN-to-PAN transition. To address this, we replaced the neck with a Bidirectional Feature Pyramid Network (BiFPN), as shown in Figure 3c. Unlike the conventional FPN+PAN, BiFPN introduces bidirectional connections and learnable fusion weights, enabling the network to prioritize relevant features across scales [23]. For effective multiscale fusion, BiFPN employs a learnable weighted summation, combined with Swish activation and normalization to maintain stability and support gradient flow.

Let $\{x_i\}_{i=1}^n$ denote the set of input feature maps to be fused, and $\{w_i\}_{i=1}^n$ denote the corresponding learnable weights. The output of the fusion $\hat{x}$ is calculated as follows:

Swish Activation: Each weight w_i is activated using the Swish function:

$$Swish(w_i) = w_i \,.\, \sigma(w_i) = \frac{w_i}{1+e^{-w_i}} \quad (8)$$

Weight Normalization: The weights w_i are normalized by dividing by the sum of all Swish values of the weights, with a small constant ϵ (typically 10^{-4}) added to avoid division by zero:

$$\widetilde{w}_i = \frac{w_i}{\sum_{j=1}^n Swish(w_j) + \epsilon} \quad (9)$$

Fused Feature Map Calculation: Finally, the fused feature map $\hat{x}$ is computed as the weighted sum of the input feature maps x_i, each multiplied by its corresponding normalized weight $\widetilde{w}_i$:

$$\hat{x} = \sum_{i=1}^n \widetilde{w}_i \,.\, x_i \quad (10)$$

These structural refinements collectively strengthen the model's ability to detect vehicles across varying sizes and densities, thus offering a more robust solution for real-time traffic monitoring and intelligent transportation systems.

4. Experiments and Analysis of Results

4.1. Experimental Setup and Dataset

4.1.1. Dataset

In this study, we utilize the Vehicles-COCO dataset, constructed and preprocessed via the Roboflow platform, for vehicle detection. It consists of 18,998 high-resolution images featuring various vehicle types-cars, motorcycles, buses, and trucks-captured under diverse environmental conditions, making it ideal for real-world traffic surveillance.

Each image is carefully annotated with bounding boxes to ensure high precision in localization and classification. The COCO-format annotations allow easy integration with modern deep learning models. The dataset is divided into three subsets: 70% for training, 20% for validation, and 10% for testing. This split supports effective model learning while ensuring objective evaluation on unseen data.In this study, we utilize the Vehicles-COCO dataset, which was constructed and preprocessed using the Roboflow platform, to address the task of vehicle.

4.1.2. Experimental Setup

This experiment was conducted on the Kaggle platform using an NVIDIA Tesla P100 GPU with 16GB of RAM. The implementation was performed using Python 3.10.12 programming language, PyTorch 2.4 framework, and CUDA 12.1. The image input size was set to 640 × 640 pixels. The training process lasted for 200 epochs with a batch size of 32. The optimizer used was Stochastic Gradient Descent (SGD), and mixed precision training (AMP) was disabled. The early stopping strategy was configured with a patience of 10 epochs.

4.1.3. Evaluation Metrics

In evaluating the performance of the vehicle detection model, several key metrics were used to measure the accuracy and detection capability of the model:

Precision: Precision is defined as the ratio of correctly detected vehicles (True Positives - TP) to the total number of predicted positive instances, including false positives (FP), where vehicles are mistakenly identified as vehicles. This metric assesses the model's ability to accurately identify vehicles without confusion.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (11)$$

Recall: Recall measures the model's ability to detect all actual vehicles. It is calculated as the ratio of correctly detected vehicles (TP) to the total number of actual vehicles, including those that the model failed to detect (False Negatives - FN).

$$\text{Recall} = \frac{TP}{TP+FN} \quad (12)$$

Average Precision (AP) measures the model's accuracy across different recall levels by averaging the precision

values at each detection rank. Here, P_{ri} represents the precision at the *i*-th recall level, and *r* is the total number of recall levels considered.

$$AP = \frac{\sum P_{ri}}{\sum r} \quad (13)$$

Mean Average Precision (mAP): Mean Average Precision is computed by averaging the AP values across all classes. This provides an overall performance metric that reflects the model's accuracy in detecting all types of vehicles in the vehicle detection task.

$$mAP = \frac{AP}{num\_class} \quad (14)$$

These metrics are essential for evaluating the quality of the model in vehicle detection, providing an overall view of the classification accuracy and precision of the YOLOv8-BiFPN-MHSA model in this task.

4.2. Comparative Experiment of Attention Module

To assess the impact of different attention mechanisms on enhancing YOLOv8 for our dataset, we conducted a series of controlled comparative experiments. Each attention module was independently integrated into the YOLOv8 framework to evaluate its effect on detection performance in practical scenarios. Table 1 presents the results for three popular attention mechanisms: SimAM, CBAM, and MHSA. The evaluation uses standard metrics: number of parameters (M), precision (P), recall (R), and mean average precision at IoU 0.5 (mAP@0.5)

Table 1. *Comparison of results with different attention modules*

Model	Attention Mechanisms	Parameters	Precision (%)	Recall	mAP@ 0.5 (%)
YOLOv8n	MHSA	3,2	71,2	56,6	64,5
	CBAM	3,07	71	56,1	64,3
	SimAM	3,01	71,3	56,8	64,6

4.3. Ablation Experiment

Table 2 presents the ablation study evaluating the impact of various modules integrated into the YOLOv8n architecture, including SimAM, CBAM, MHSA, and BiFPN. The experimental results offer clear evidence of each component's contribution to the model's performance, particularly in precision and mAP. These findings highlight how each enhancement - individually and in combination - helps improve detection accuracy, especially for small and complex objects.

Table 2. *Comparison of results from ablation experiments*

Model	Parameters/M	Precision (%)	Recall (%)	mAP@ 0.5 (%)	mAP@ 0.5-0.95 (%)
YOLOv8n	3,01	71,2	56,5	64,4	44,5
YOLOv8n + SimAM	3,01	71,3	56,8	64,6	44,6
YOLOv8n + CBAM	3,07	71	56,1	64,3	44,5
YOLOv8n + MHSA	3,20	71,2	56,6	64,5	44,6
YOLOv8n + BiFPN	3,14	72,4	56,5	65,5	45,5
YOLOv8n + BiFPN + MHSA (Ours)	**3,34**	**73,1**	**56,6**	**65,7**	**45,6**

Integrating various modules into the YOLOv8n architecture has significantly enhanced object detection performance. Adding the SimAM attention mechanism led to a modest 0.2% increase in mAP@0.5–0.95 and improved precision from 71.2% to 71.3%, with a slight increase in recall from 56.5% to 56.8%, indicating better spatial feature focus. CBAM, however, showed negligible impact, with a slight drop in precision to 71.0% and recall to 56.1%, suggesting limited effectiveness. MHSA (Multi-Head Self-Attention) offered a marginal 0.1% mAP gain and a small recall increase to 56.6%, while maintaining precision at 71.2%, reflecting its ability to capture long-range dependencies. In contrast, BiFPN (Bidirectional Feature Pyramid Network) yielded a more notable boost, raising mAP@0.5–0.95 by 1.0%, precision to 72.4%, and maintaining recall at 56.5%, demonstrating improved multi-scale feature aggregation. When combined with MHSA, the model achieved its best performance, reaching 45.6% in mAP@0.5–0.95, 73.1% in precision, and 56.6% in recall - improvements of 1.1%, 1.9%, and 0.1% respectively over the original YOLOv8n. These findings underscore the synergistic effect of combining BiFPN with attention mechanisms, particularly for detecting small and complex objects across scales.

4.4. Comparison with Other YOLO Variants

Table 3 presents a comparative analysis of four lightweight object detection models: YOLOv8n, YOLOv10, YOLOv11, and the proposed model (Ours). YOLOv11 has the fewest parameters (2.59M), followed by YOLOv10 (2.70M) and YOLOv8n (3.01M), highlighting YOLOv11's compactness. Our model has the highest parameter count (3.34M) but achieves the best detection performance.

Table 3. *Comparison of results from other YOLO Variants*

Model	Parameters/M	Precision (%)	Recall	mAP@ 0.5 (%)	mAP@ 0.5-0.95 (%)
YOLOv8n	3,01	71,2	56,5	64,4	44,5
YOLOv10n	2,70	69,8	54,5	62,9	43,7
YOLO11n	2,59	71,8	55,9	64,2	44,5
YOLOv8n BiFPN MHSA(Ours)	**3,34**	**73,1**	**56,6**	**65,7**	**45,6**

For mAP@0.5, YOLOv8n leads existing models (64.4%), slightly ahead of YOLOv11 (64.2%) and YOLOv10 (62.9%), while our model reaches 65.7%. In mAP@0.5:0.95, YOLOv8n and YOLOv11 tie at 44.5%, YOLOv10 scores 43.7%, and our model achieves 45.6%. YOLOv11 shows the highest precision (71.8%) among baselines, followed by YOLOv8n (71.2%) and YOLOv10 (69.8%), while our model surpasses all with 73.1%.

In terms of recall, YOLOv8n performs best among baseline models with 56.5%, followed closely by YOLOv11 (55.9%) and YOLOv10 (54.5%). Our model achieves the highest recall at 56.6%, indicating improved detection coverage without sacrificing precision.

Overall, while YOLOv10 and YOLOv11 improve efficiency, they do not significantly outperform YOLOv8n in accuracy. Our model shows consistent gains across all metrics, balancing accuracy and efficiency for real-time use.

4.5. Visualization of the Results

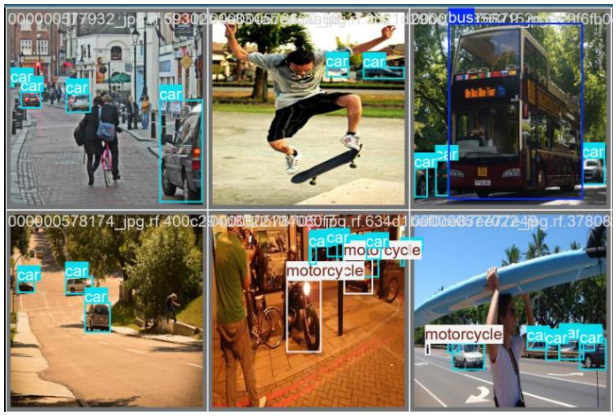

Figure 4. *The illustrative results of object detection and classification predictions*

To provide a visual representation of how effectively the proposed model detects vehicles, Figure 4 presents several output samples generated by the YOLOv8-BiFPN-MHSA model on the test set of the Vehicles-COCO dataset. The selected images depict various real-world traffic scenarios, including dense traffic conditions, partially occluded vehicles, and small-sized objects. It is evident that the model can accurately localize vehicles even in complex environments, with predicted bounding boxes closely aligning with ground truth annotations. Moreover, the integration of BiFPN and MHSA significantly enhances the model's ability to detect closely spaced vehicles and those embedded in cluttered backgrounds.

5. Conclusion

This study presents an enhanced YOLOv8-based architecture tailored for urban traffic vehicle detection, addressing key challenges such as small object detection, occlusion, and complex backgrounds. By integrating the Bidirectional Feature Pyramid Network (BiFPN) and Multi-Head Self-Attention (MHSA), the proposed YOLOv8-BiFPN-MHSA model achieves more effective multi-scale feature fusion and captures long-range spatial dependencies. Experimental results on the Vehicles-COCO dataset confirm the advantages of this approach, with the model achieving the highest performance across all evaluation metrics, including a precision of 73.1%, recall of 56.6%, mAP@0.5 of 65.7%, and mAP@0.5:0.95 of 45.6%. Compared to other attention mechanisms and lightweight YOLO variants, the proposed model demonstrates superior accuracy and robustness, particularly in detecting small, overlapping, or densely packed vehicles. These findings validate the effectiveness of combining BiFPN and MHSA and highlight the model's potential for real-world deployment in intelligent traffic surveillance systems.

REFERENCES

[1] C. Chen *et al.*, "Enhanced YOLOv5: An Efficient Road Object Detection Method," *Sensors*, vol. 23, no. 20, art. 8355, 2023.

[2] J. Redmon *et al.*, "You Only Look Once: Unified, Real-Time Object Detection", in *Proc. IEEE CVPR*, 2016, pp. 779–788.

[3] N. U. A. Tahir *et al.*, "PVswin-YOLOv8s: UAV-Based Pedestrian and Vehicle Detection for Traffic Management", *Drones*, vol. 8, no. 3, art. 84, 2024.

[4] H. Guo *et al.*, "Research on Vehicle Detection Based on Improved YOLOv8 Network", arXiv:2501.00300, 2025.

[5] X. Liu *et al.*, "YOLOv8-FDD: A Real-Time Vehicle Detection Method Based on Improved YOLOv8", *IEEE Access*, vol. 12, pp. 136280–136296, 2024.

[6] M. Tan, R. Pang, Q. V. Le, "Eff1icientDet: Scalable and Efficient Object Detection", in *Proc. IEEE CVPR*, 2020, pp. 10781–10790.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017..

[8] C. Lv *et al.*, "Vehicle detection and classification using an ensemble of EfficientDet and YOLOv8", *PeerJ Comput. Sci.*, vol. 10, art. e2233, 2024.

[9] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Convolutional channel features", in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 82–90.

[10] G. Mandellos, I. Keramitsoglou, and C. Kiranoudis, "A background subtraction algorithm for detecting and tracking vehicles", *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1619–1631, 2011.

[11] J. Cao, W. Zhang, Z. He, and H. Sun, "Robust vehicle detection in aerial images based on convolutional neural networks", *IEEE Access*, vol. 9, pp. 45620–45629, 2021.

[12] M. T. Islam, R. S. Amin, and M. Murshed, "A comprehensive review of deep learning-based approaches for traffic object detection", *Sensors*, vol. 24, no. 1, p. 222, 2024.

[13] N. Li, S. Tang, Y. Fan, and Y. Zhang, "YOLOv8-MECA-ASPF: A lightweight vehicle detection algorithm based on multi-attention and context perception", in *Proc. 2024 IEEE Int. Conf. Intell. Transp. Syst. (ITSC)*, 2024, pp. 1–6.

[14] H. Li, Y. Song, and X. Sun, "Lightweight network design for efficient object detection using MECA and ASPF modules", *Neurocomputing*, vol. 540, pp. 25–34, 2023.

[15] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking", *Comput. Vis. Image Underst*, vol. 193, pp. 102907, 2020.

[16] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite", in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 3354–3361.

[17] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu, "Vision meets drones: A challenge", in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 643–659.

[18] X. Wang *et al.,* "Non-Local Neural Networks", in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[19] J. Terven and D. M. Cordova-Esparza, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS", *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023.

[20] I. Bello *et al.*, "Attention Augmented Convolutional Networks", in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 3286–3295.

[21] A. Dosovitskiy et al., "An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale", in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection", *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2117–2125.

[23] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation", *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8759–8768.