BIMA
BULLETIN OF INTELLIGENT MACHINES AND ALGORITHMS
Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

A Progressive Training Framework for Robust YOLOv11-Based Vehicle Detection Across Domain Shifts in Real-World CCTV Environments

Aditya Gunaldi[1*], Budiman[2], Chairul Habibi[3], Nur Alamsyah[4]

Informatics[1,2]
Information System[3,4]
Universitas Informatika dan Bisnis Indonesia[1,2,3,4]
https://unibi.ac.id/[1,2,3,4]
aditya.g22@student.unibi.ac.id[1*]

Abstract. The performance of deep learning-based vehicle detection models often deteriorates when applied to real-world CCTV environments due to domain shift caused by variations in lighting, occlusion, glare, and changes in camera viewpoint. This study aims to develop a YOLOv11-based progressive training framework to improve the model's generalization ability under heterogeneous operational conditions. The proposed method consists of four stages: base training, fine-tuning, and a supervised progressive domain adaptation strategy implemented through sequential fine-tuning on labeled target-domain CCTV images before OpenVINO-based inference optimization. The model is trained using a source dataset and adapted to a target dataset representing real-world CCTV conditions. Evaluation was conducted using Precision, Recall, mAP@50, mAP@50–95, loss curve analysis, per-vehicle-class evaluation, visual testing on CCTV video, as well as latency and throughput measurements. The results show that base training achieved an mAP@50 of 0.919 and built a robust feature representation, while fine-tuning maintained performance stability with an mAP@50 of 0.915. Although domain adaptation reduced mAP@50 to 0.811 due to domain shift, the model demonstrated improved generalization capabilities and maintained more consistent detection under low-light conditions, glare, occlusion, and heavy traffic. OpenVINO INT8 optimization increased inference speed from 4.86 FPS to 9.08 FPS with minimal accuracy loss. These findings demonstrate that the progressive training framework effectively bridges differences in data distribution while producing a vehicle detection model that is more robust, efficient, and suitable for deployment in real-time CCTV-based traffic monitoring systems.

Keywords: YOLOv11; Progressive Training; Domain Adaptation; Vehicle Detection; Intelligent Transportation Systems

1. INTRODUCTION

Image- and video-based vehicle detection is a critical component of intelligent transportation systems because it supports real-time traffic flow monitoring, congestion analysis, and operational decision-making [1], [2]. In practice, the need for accurate systems is increasing as vehicle volumes and the complexity of urban environments grow, particularly in areas with heavy traffic such as Indonesia [3]. Therefore, the development of vehicle detection models that are not only precise but also stable under real-world conditions has become an increasingly relevant issue in current computer vision research [4], [5].

Developments in the YOLO family demonstrate that single-stage approaches remain the preferred choice for object detection due to their balance between inference speed and accuracy [1], [2]. However, the performance of detection models in real-world environments often degrades when faced with variations in lighting, occlusion, glare, traffic density, and small objects—particularly in CCTV images, which have a different data distribution from the training data [2], [6]. These findings confirm that high accuracy in the source domain does not guarantee performance robustness in the operational domain.

Several recent studies report that low-light conditions and dense environments cause a significant decline in detection capabilities, particularly for small and overlapping objects [7], [8]. In traffic scenarios, two-wheeled vehicles are generally more difficult to recognize than larger vehicles due to their small bounding box dimensions and high sensitivity to changes in viewing angle and image quality [4], [5]. Thus, the main challenge is not only recognizing vehicle objects but also maintaining detection sensitivity under heterogeneous visual conditions.

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

Various approaches have been proposed to address these issues, ranging from feature resolution enhancement, multi-scale feature extraction, super-resolution, attention mechanisms, to transformer-based architectures [2], [9]. In the automotive domain, YOLO-based methods are also being continuously developed to support real-time detection in CCTV and adaptive traffic control systems [10], [11]. However, most studies still focus on improving accuracy within the training domain, while cross-domain generalization capabilities in real-world CCTV environments have not yet been a primary concern.

Recent studies indicate that domain shift remains a fundamental obstacle in the application of object detection to real-world conditions [12], [13]. Domain adaptation and transfer learning strategies have proven effective in aligning feature distributions between source and target domains, but their effectiveness heavily depends on the visual disparity between domains and the characteristics of the detected objects [14], [15]. In the context of vehicles, these approaches are crucial because homogeneous source images often do not represent the complexity of operational CCTV footage, particularly at night or during traffic congestion.

Accordingly, a progressive approach involving base training, fine-tuning, and domain adaptation is increasingly used to build feature representations incrementally before the model is tested on the more challenging target domain [5], [16]. This strategy is considered more stable than training directly on target data because it allows the model to learn general patterns first and then adjust parameters in a more targeted manner [12], [15]. However, recent empirical reports still indicate that a performance drop occurs when the model is confronted with extreme distribution shifts, particularly for small objects and under low-light conditions [2], [8].

Based on this literature review, there remains a research gap regarding the integration of progressive training strategies with domain adaptation explicitly designed for CCTV-based vehicle detection under complex operational conditions [13], [16]. Some studies have not evaluated gradual performance changes across each training phase, have not distinguished the contributions of base training and fine-tuning to generalization, and have not linked quantitative results to comprehensive real-world testing [4], [10]. These gaps underscore the need for a more systematic and adaptive approach.

This study aims to develop a progressive training framework for YOLOv11-based vehicle detection that encompasses base training, fine-tuning, domain adaptation, and evaluation in real CCTV environments. The contribution of this study lies in the design of a phased learning process to bridge the source and target domains, ensuring that the model is not only optimized for training data but also more robust against variations in lighting, occlusion, glare, and traffic density. The scope of this research focuses on multi-class vehicle detection, evaluation of detection metrics, analysis of performance changes at each stage, and implementation testing in operational CCTV scenarios.

2. RELATED WORK

Advances in the You Only Look Once (YOLO) model have made real-time object detection increasingly accurate and efficient for various applications, including smart transportation systems and CCTV-based traffic monitoring. The evolution of the architecture up to YOLOv11 has brought improvements in feature extraction mechanisms, computational efficiency, and the ability to detect small objects without significantly increasing the model's complexity. These advantages make YOLOv11 one of the most promising approaches for implementation in real-time vehicle detection systems. However, most evaluations have been conducted on datasets with relatively homogeneous data distributions, so its generalization ability to real-world operational conditions has not yet been fully tested [17], [18].

Several studies have adapted YOLO for vehicle detection in CCTV environments by applying data augmentation, transfer learning, and fine-tuning. These approaches have been shown to improve the mean Average Precision (mAP), accelerate training convergence, and reduce the risk of overfitting when training data is limited. Additionally, lightweight models such as YOLOv11n demonstrate a good balance between accuracy and computational efficiency, making them suitable for implementation on edge devices. However, these performance improvements are generally achieved under lighting conditions and viewpoints that closely resemble the training data, so accuracy often drops when the model is exposed to changes in data distribution [5], [19].

To address these limitations, several studies have begun integrating attention mechanisms, feature resolution enhancement, and specialized architectures to detect small objects as well as occluded objects. These approaches can improve detection sensitivity in complex traffic environments, particularly when vehicles are closely packed together or under adverse weather conditions. Nevertheless, most studies focus on modifying model architectures, while the stepwise learning process—involving base training, fine-tuning, and domain adaptation—is still rarely evaluated systematically. Furthermore, the success of these methods is generally measured only using the mAP metric without considering the model's ability to maintain detection stability under real-world CCTV conditions [16], [20].

Other studies have shown that inference optimization using OpenVINO can improve computational efficiency through model conversion and quantization without causing a significant drop in accuracy. This approach is highly relevant for implementing vehicle detection systems on CPU-based devices because it can increase throughput and reduce latency. However, inference optimization is generally treated as a separate stage from

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

the learning process, so the relationship between training strategies, generalization ability, and implementation efficiency has not been extensively discussed in an integrated manner.

Based on this review, there are three main research gaps. First, previous research has focused more on improving accuracy in the source domain than on model robustness to domain shifts. Second, few studies have evaluated the contribution of each training stage (base training, fine-tuning, and domain adaptation) to gradual changes in model performance. Third, the integration of progressive training, evaluation in real CCTV environments, and OpenVINO-based inference optimization remains very limited. Therefore, this study proposes a YOLOv11-based progressive training framework that integrates these three training stages before performing inference optimization using OpenVINO. This approach is expected to produce a model that not only achieves high accuracy in the training domain but is also more adaptive to variations in lighting, occlusion, glare, and changes in data distribution in real-world CCTV implementations.

3. METHODS

Figure 1 illustrates the research stages applied in the development of a YOLOv11-based vehicle detection model using a progressive training strategy, which consists of base training, fine-tuning, domain adaptation, and concludes with the model evaluation stage.

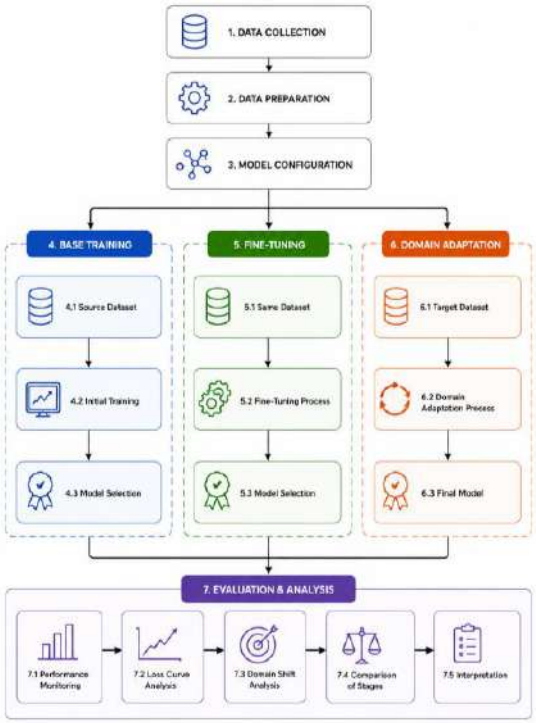


Fig. 1. Research workflow of the proposed progressive training framework for YOLOv11-based vehicle detection

3.1. Data Collection

This study employed two datasets representing different visual domains to evaluate the robustness of the proposed progressive training framework. The first dataset served as the source domain, consisting of publicly available vehicle images collected from Kaggle and Roboflow repositories. These images were characterized by relatively homogeneous acquisition conditions and were used to establish the initial feature representation during the base training and fine-tuning stages. In contrast, the target domain comprised vehicle images extracted from real-world CCTV video streams, representing operational traffic monitoring environments with greater visual complexity, including variations in illumination, traffic density, object scale, occlusion, and camera viewpoints.

The datasets contain four vehicle categories, namely car, bus, truck, and motorbike. Before model training, all images were manually annotated, validated, and cleaned to ensure consistency between image files and object labels. Invalid annotations and unmatched image-label pairs were automatically removed to produce structurally consistent datasets suitable for object detection training.

Table 1 summarizes the characteristics of the source and target datasets used in this study. After the validation process, the source-domain dataset consisted of 2,724 training images, 259 validation images, and 129 testing images, whereas the target-domain dataset comprised 1,286 training images, 415 validation images, and 237 testing images. All subsets contained complete image-label pairs without missing annotations, ensuring the integrity of the training and evaluation processes.

Because the original datasets exhibited class imbalance, particularly for the bus and truck categories, oversampling was applied exclusively to the training subset to improve class representation while preserving the validation and testing distributions. This strategy was intended to mitigate model bias toward majority classes and improve the learning of minority-class feature representations without introducing data leakage.

TABLE 1. Summary of Source and Target Dataset Characteristics

Characteristics	Source Domain	Target Domain
Data source	Kaggle and Roboflow datasets	Real-world CCTV video frames
Purpose	Base training and fine-tuning	Domain adaptation
Vehicle classes	Car, Bus, Truck, Motorbike	Car, Bus, Truck, Motorbike
Training images	2,724	1,286
Validation images	259	415
Testing images	129	237
Annotation	Manual annotation (YOLO format)	Manual annotation (YOLO format)
Data validation	Validated and cleaned	Validated and cleaned

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

3.2. Data Preparation

All data undergoes a preprocessing stage that includes object annotation, image format normalization, splitting the dataset into training, validation, and test sets, and applying data augmentation to increase sample diversity. This process aims to produce a more representative data distribution so that the model can learn object characteristics more robustly and reduce the risk of overfitting during training.

3.3. Model Configuration

The object detection model is configured using the YOLOv11 architecture with the following training parameters: image size (input size), number of epochs, batch size, learning rate, optimizer, and YOLO's built-in loss function. During the optimization process, the model parameters are updated using the gradient descent algorithm based on the minimization of the loss function as shown in Equation (1).

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta) \quad (1)$$

where θ represents the model parameters, η is the learning rate, and $L(\theta)$ is the total loss function.

3.4. Base Training

The base training phase aims to build a basic feature representation using the source dataset. The model is trained from initial weights (pretrained weights) until it achieves the best weights based on validation performance. During this process, the YOLO loss function is minimized through a combination of localization loss (bounding box regression), classification loss, and bounding box distribution loss, which can generally be expressed in Equation (2).

$$L_{total}=L_{box}+L_{cls}+L_{DFL} \quad (2)$$

where L_{box} is the bounding box loss, L_{cls} is the classification loss, and L_{DFL} represents the Distribution Focal Loss.

3.5. Fine Tuning

The fine-tuning stage uses the best weights from the initial training as the starting point for optimization. This process aims to refine the model parameters through more targeted adjustments without altering the previously learned feature representations. Parameter updates are performed incrementally using a smaller learning rate so that the model becomes more stable and has better generalization capabilities regarding data variations in the source domain.

3.6. Domain Adaption

The final stage of the proposed framework employs a supervised progressive domain adaptation strategy to improve the model's robustness when deployed in real-world CCTV environments. Unlike adversarial domain adaptation methods that explicitly align feature distributions through auxiliary networks, the proposed approach performs domain adaptation by sequentially fine-tuning the detector using labeled images collected from the target domain. This strategy enables the model to progressively adapt its learned representations while preserving the discriminative features acquired during the previous training stages.

The adaptation process begins by loading the best-performing weights obtained from the base training and fine-tuning stages. These weights serve as the initialization for the subsequent learning process on the target-domain dataset. Instead of training the detector from scratch, the previously learned parameters provide a strong representation of general vehicle characteristics, allowing the optimization process to focus primarily on adapting the model to the visual characteristics of CCTV imagery, including illumination changes, nighttime scenes, glare from vehicle headlights, dense traffic, partial occlusion, and viewpoint variations.

Unlike conventional transfer learning approaches that freeze part of the network during adaptation, all parameters of the YOLOv11 detector remain trainable throughout this stage. Updating the entire network enables both the backbone and detection head to gradually adjust to the new feature distribution while maintaining the semantic representations learned from the source domain. This full-network optimization is particularly important because the target dataset contains substantial differences in object appearance and environmental conditions that cannot be effectively addressed by updating only the detection head.

During the adaptation process, the training configuration is intentionally kept consistent with the previous learning stages to preserve optimization stability. The same input image resolution, optimizer, loss function, and object detection architecture are maintained, while the model parameters are refined using labeled target-domain data. Consequently, the adaptation process focuses on learning domain-specific visual characteristics rather than modifying the underlying detection architecture.

Conceptually, the proposed strategy seeks to reduce the discrepancy between the feature distributions of the source domain and the target domain through progressive supervised learning. Rather than explicitly minimizing a domain discrepancy loss, feature alignment is achieved implicitly by continuously updating the detector using annotated target-domain samples. The optimization objective can therefore be expressed as minimizing the difference between the learned feature representations of the two domains, as formulated in Equation (3).

$$\min \; D(F_s, F_t) \quad (3)$$

where F_s and F_t denote the feature representations extracted from the source and target domains,

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

respectively, and D(·) represents the distribution discrepancy between both domains. As the optimization proceeds, the detector gradually learns feature representations that are more invariant to changes in illumination, traffic density, camera viewpoint, and object scale, thereby improving its generalization capability in heterogeneous CCTV environments.

The overall adaptation procedure can be summarized as follows. First, the detector is trained on the source dataset to obtain the optimal model parameters. Second, the best-performing weights are transferred to initialize the adaptation stage. Third, supervised fine-tuning is performed using labeled target-domain images while keeping the complete YOLOv11 network trainable. Finally, the adapted model is selected based on validation performance and subsequently evaluated on real-world CCTV scenarios. This progressive adaptation mechanism forms the final learning stage before inference optimization using OpenVINO.

Figure 2 illustrates the proposed supervised progressive domain adaptation strategy adopted in this study. The training process begins with base training using the source dataset to learn general vehicle representations. The best-performing model is subsequently refined through fine-tuning, after which the updated weights are transferred to the target dataset for supervised domain adaptation. Finally, the adapted YOLOv11 model is optimized using OpenVINO prior to deployment in a real-world CCTV traffic monitoring environment. This sequential learning process enables the detector to gradually reduce the effect of domain shift while preserving the discriminative features acquired during the previous training stages.

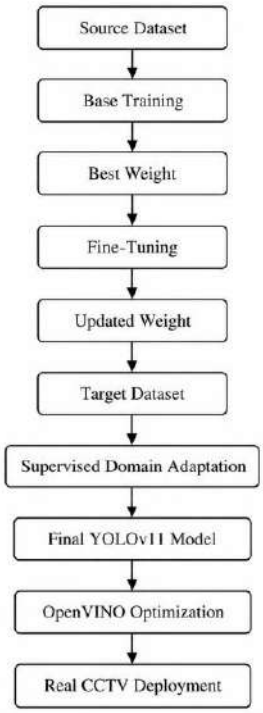


Fig. 2. Proposed supervised progressive domain adaptation strategy for bridging the source and target domains using sequential fine-tuning prior to OpenVINO-based deployment.

As shown in Figure 2, knowledge acquired from the source domain is progressively transferred to the target domain through sequential parameter optimization rather than training a new model from scratch. This strategy allows the detector to retain robust feature representations learned during the initial training while adapting to the visual characteristics of CCTV imagery, including illumination changes, glare, occlusion, traffic density, and viewpoint variations. Consequently, the proposed framework emphasizes progressive knowledge transfer as the primary mechanism for improving generalization under heterogeneous operational conditions.

3.7. Evaluation and Analysis

The evaluation phase was conducted to measure the model's performance at each stage of progressive training through the analysis of detection metrics, including Precision, Recall, mAP@50, and mAP@50–95, along with an evaluation of the training curve and an analysis of performance changes resulting from domain adaptation. Precision and Recall were calculated using Equations (4) and (5).

$$\text{Precision} = \frac{TP}{TP+FP} \tag{4}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{5}$$

where TP denotes True Positive, FP denotes False Positive, and FN denotes False Negative. Furthermore, the Average Precision (AP) value is calculated as the area under the Precision–Recall curve in Equation (6).

$$AP = \int_0^1 P\,(R)\,dR \tag{6}$$

while Mean Average Precision (mAP) is obtained by averaging the AP values across all classes, as shown in Equation (7).

$$mAP = \frac{1}{N}\sum_{i=1}^{N} A\,P_i \tag{7}$$

where N denotes the number of object classes. The analysis at this stage serves as the basis for evaluating the effectiveness of the progressive training strategy in improving the model's generalization ability in a real-world CCTV environment.

4. RESULTS AND DISCUSSIONS

4.1. Progressive Training Performance

The progressive training strategy is implemented through three sequential learning stages: base training, fine-tuning, and domain adaptation. This approach is designed to build feature representations incrementally, starting with

BIMA
BULLETIN OF INTELLIGENT MACHINES AND ALGORITHMS
Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

learning general vehicle characteristics on a base dataset, followed by refining model parameters using more specific data, and culminating in adaptation to the characteristics of the target domain, which represents real-world CCTV operational conditions. With this approach, the model is expected not only to achieve high detection accuracy in the source domain but also to maintain its generalization ability when faced with different data distributions.

A summary of the model's performance at each training stage is presented in Table 2. The initial training (base training) yielded an mAP@50 of 0.919, an mAP@50–95 of 0.735, with a precision of 0.892 and a recall of 0.864. This performance indicates that the model has adequately learned the visual characteristics of vehicles in the source dataset, resulting in a good balance between detection accuracy and object retrieval capability.

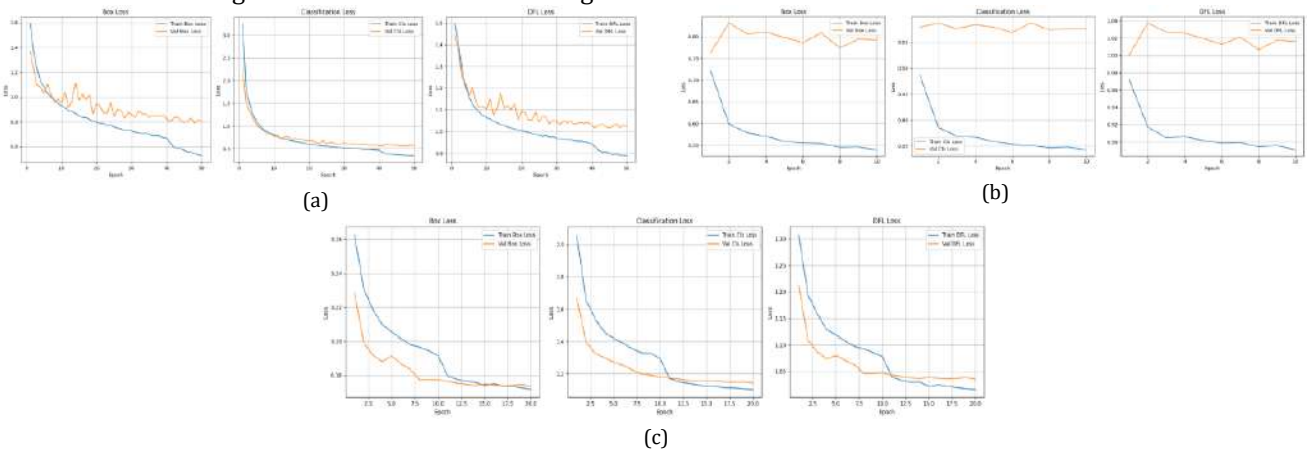
The fine-tuning stage was performed using the best weights obtained from the initial training with the aim of refining the feature representation without making aggressive parameter changes. Although the increase in mAP values was relatively small compared to the base model, the results in Table 2 show that detection performance remained stable with an mAP@50 of 0.915 and an mAP@50–95 of 0.736. This stability indicates that the fine-tuning process plays a greater role in adjusting parameters so that the model becomes more consistent with the characteristics of the training data without experiencing performance degradation due to overfitting.

TABLE 2. Summary of YOLOv11 Model Training Results

Model	Epoch	mAP@50	mAP@50-95	Precision	Recall	Val Box Loss	Val Cls Loss	Val DFL Los
Train YOLOv11n	50	0.91901	0.73520	0.89208	0.86359	0.80392	0.56723	1.02223
Fine Tuning	10	0.91458	0.73617	0.88335	0.84527	0.79230	0.57522	1.01626
Domain Adaptation	20	0.81052	0.59859	0.82632	0.72429	0.17343	1.14253	1.03528

The next stage is domain adaptation, which is the process of adapting the model to datasets with different visual characteristics, such as nighttime lighting conditions, vehicle headlight glare, high traffic density, and a predominance of small objects. As shown in Table 2, the adaptation process results in a decrease in the mAP value compared to the previous two stages. This is a common consequence when a model is faced with a domain shift—that is, a change in the data distribution between the source domain and the target domain. Nevertheless, the model is still able to maintain a relatively high level of precision, indicating that its ability to discriminate between vehicle classes remains intact despite the increased complexity of the observation environment.

The dynamics of the learning process at each training stage can be observed through the loss curves shown in Figure 3(a), Figure 3(b), and Figure 3(c). During the base training stage, all loss components experienced a rapid decline in the early epochs before gradually reaching a state of convergence. This pattern indicates that the model successfully built a basic feature representation. Upon entering the fine-tuning stage, the loss decrease proceeds more steadily with smaller fluctuations compared to the previous stage. This indicates that the parameter refinement process occurs in a controlled manner and is able to enhance learning stability without altering the feature representations acquired during the initial training.



(a) (b) (c)

Fig. 3. Comparison of loss curves at each stage of progressive training: (a) Base Training, (b) Fine-Tuning, and (c) Domain Adaptation.

In contrast, the loss curve during the domain adaptation phase shows a slower convergence process accompanied by greater fluctuations. This phenomenon indicates that the model requires a more complex adaptation process to learn the new feature distribution in the target domain. Differences in lighting characteristics, camera angles, and traffic density levels mean that the visual representations learned in the source domain cannot be fully applied

BIMA
BULLETIN OF INTELLIGENT MACHINES AND ALGORITHMS
Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

directly to the target domain, so the optimization process requires a greater number of iterations to reach a stable state.

A comparison of performance trends across the three training stages is summarized in Figure 3, which shows the overall changes in precision, recall, and mAP values. These results demonstrate that the progressive training strategy is not solely aimed at achieving the highest accuracy on the training dataset, but rather at building a model capable of maintaining detection performance when faced with more complex variations in environmental conditions. Thus, each training stage serves a complementary function: base training establishes a foundational feature representation, fine-tuning refines the model parameters, and domain adaptation enhances generalization capabilities across heterogeneous CCTV operational conditions. These findings form the basis for a more in-depth evaluation in the following section regarding the impact of domain adaptation on vehicle detection performance in real-world environments.

4.2. Domain Adaptation Analysis

The application of domain adaptation is a crucial step in this research to evaluate the model's ability to maintain detection performance when faced with data distributions that differ from those used in the initial training process. Unlike the base training, which uses a dataset with relatively homogeneous visual characteristics, this stage utilizes a domain dataset that more realistically represents actual CCTV operational conditions, including nighttime lighting, glare from vehicle headlights, high traffic density, occlusion, and a predominance of small objects. These differing characteristics require the model not only to recognize objects based on previously learned visual features but also to adapt its feature representations to more complex environmental variations.

Changes in model performance following the domain adaptation process are summarized in Table 2, while the evolution of evaluation metrics during training is visualized in Figure 4. Compared to the fine-tuned model, the mAP@50 value decreased from 0.915 to 0.811, while mAP@50–95 decreased from 0.736 to 0.599. This decline indicates that the model faces greater challenges in localizing and classifying objects in the target domain, which has a different visual distribution. This phenomenon is a common characteristic of the domain adaptation process, where the feature representations obtained from the source domain do not yet fully align with the characteristics of the new domain.

The reduction in detection performance observed after supervised domain adaptation is primarily attributed to the substantial visual discrepancy between the source and target domains rather than to a deficiency of the proposed training strategy. The target-domain CCTV images contain considerably more challenging conditions, including nighttime illumination, headlight glare, partial occlusion, increased traffic density, and a higher proportion of small objects than the source-domain dataset. These factors modify the underlying feature distribution learned during the initial training stages, making object localization and classification substantially more difficult. Consequently, the model requires additional optimization to adapt previously learned feature representations to the new visual characteristics while preserving its discrimination capability across vehicle categories.

This interpretation is supported by both the quantitative and qualitative evaluations. The class-wise analysis indicates that the motorbike category experiences the largest reduction in recall and mAP because motorcycles generally occupy fewer pixels, frequently overlap with surrounding vehicles, and are more susceptible to illumination changes than larger vehicle classes. Conversely, car, bus, and truck maintain relatively stable precision owing to their larger geometric structures and more distinguishable visual features. Furthermore, the visual evaluation under daytime and nighttime CCTV conditions demonstrates that, despite the decrease in aggregate mAP, the adapted model preserves more consistent vehicle detection under complex operational scenarios. These observations suggest that the adaptation process mainly influences detection sensitivity rather than the model's capability to discriminate among vehicle categories.

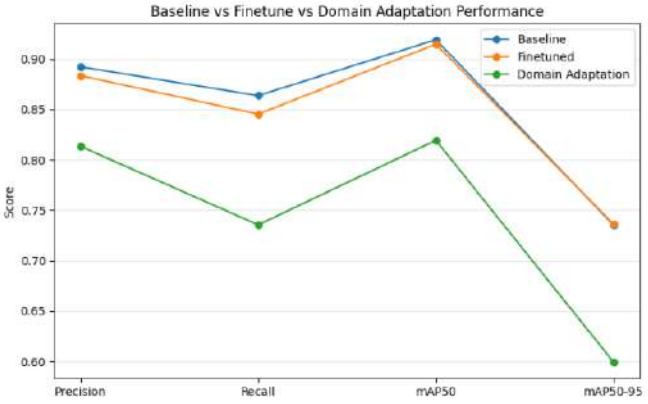


Fig. 4. Training Comparison Chart

Although there was a decrease in the mAP value, the change in performance did not indicate a comprehensive degradation. The precision value remained at a relatively high level, while recall experienced a greater decline compared to other metrics. This pattern indicates that the model is still capable of generating accurate predictions for objects that are successfully recognized, but has become more selective, resulting in some objects no longer being detected. In other words, the adaptation process affects detection sensitivity more than the model's ability to distinguish between vehicle classes.

This trend is more clearly visible in Figure 5, which shows a comparison of precision and recall during each training stage. The fine-tuned model maintains a balance between

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

these two metrics, whereas the domain-adapted model exhibits a more pronounced decrease in recall. This indicates that the visual complexity of the target domain increases the likelihood of false negatives, particularly when objects are occluded, relatively small in size, or located in areas with low lighting contrast. Conversely, the stable precision values suggest that the model's predictions remain reliable even as the number of successfully detected objects decreases.

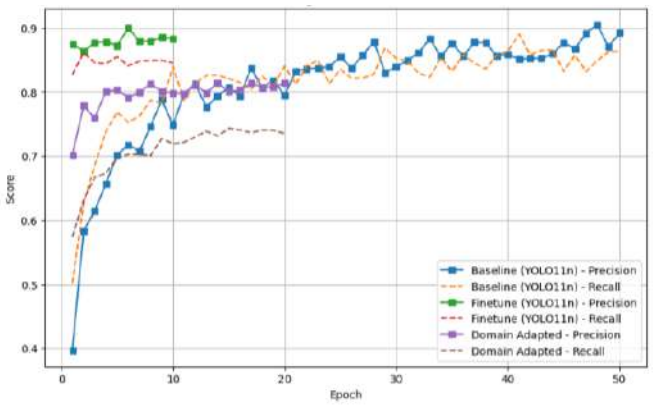


Fig. 5. Comparison Chart of Precision and Recall

Changes in detection performance are also reflected in the evolution of mAP@50 shown in Figure 6. Models resulting from base training and fine-tuning exhibit a faster convergence process with relatively stable mAP values, whereas the domain adaptation model requires a longer training process before reaching a converged state. This pattern indicates that parameter adjustments in the target domain occur gradually in response to changes in the data distribution. Thus, the initial performance drop during the adaptation process should not be interpreted as a model failure, but rather as a consequence of the learning process adapting to visual characteristics not previously encountered during training.

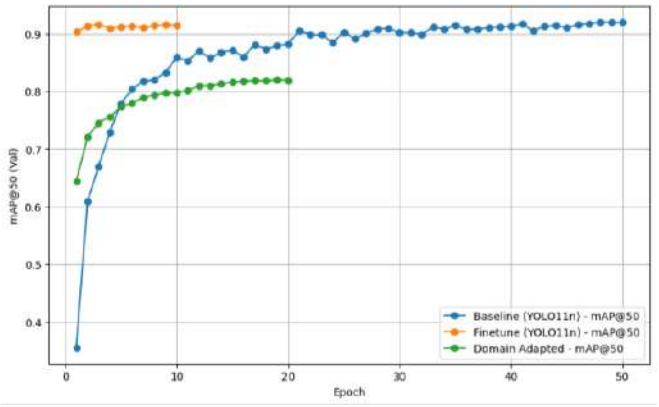


Fig. 6. Comparison Chart of mAP@50

Further analysis of the performance of each vehicle class is presented in Table 3. The evaluation results show that the "car" and "bus" classes still maintain relatively high precision levels after the domain adaptation process, whereas the "motorbike" class experiences the most significant performance decline, particularly in the recall and mAP@50–95 metrics. The characteristics of motorcycles—which have smaller object sizes, often overlap with other vehicles, and are more sensitive to changes in lighting—make the feature extraction process more difficult compared to larger vehicles. In contrast, vehicles such as cars and buses have more consistent geometric shapes, so their feature representations are easier to preserve even when domain changes occur.

TABLE 3. Metrics Evaluation Results by Class

Model	Class	Precision	Recall	mAP50	mAP50-95
Base	Bus	0.876	0.92	0.961	0.805
Base	Car	0.907	0.905	0.961	0.8
Base	Motorbike	0.836	0.847	0.907	0.675
Base	Truck	0.838	0.854	0.908	0.757
Fine Tune	Bus	0.879	0.849	0.914	0.739
Fine Tune	Car	0.926	0.874	0.952	0.799
Fine Tune	Motorbike	0.841	0.807	0.893	0.656
Fine Tune	Truck	0.876	0.854	0.869	0.721
Domain	Bus	0.813	0.826	0.865	0.741
Domain	Car	0.89	0.815	0.887	0.667
Domain	Motorbike	0.768	0.58	0.693	0.365
Domain	Truck	0.831	0.676	0.798	0.623

These findings are supported by the results of the visual evaluation shown in Figure 7(a) and Figure 7(b). Under daytime conditions, the domain-adapted model was able to maintain consistent detection across various traffic density levels with relatively stable bounding boxes. Meanwhile, under nighttime conditions, the model was still able to recognize most vehicles, although the confidence scores tended to be lower compared to those of the source-domain model. This decrease in confidence is primarily influenced by glare, uneven lighting, and reduced object texture information due to low-light conditions. Nevertheless, the number of successfully detected objects still shows an increase compared to the model that has not undergone the adaptation process, particularly for the "motorbike" class, which previously was more frequently misclassified under low-light conditions.

BIMA
BULLETIN OF INTELLIGENT MACHINES AND ALGORITHMS
Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

(a) (b)

Fig. 7. Vehicle detection results using the Domain Adaptation model under (a) daytime and (b) nighttime conditions.

Overall, the analysis results show that domain adaptation shifts the model's characteristics from one originally oriented toward optimal performance in the source domain to a model that is more adaptive to real-world operational conditions. Although there was a decline in some dataset-based evaluation metrics, the model's ability to maintain detection in heterogeneous CCTV environments improved. These findings indicate that the success of domain adaptation is not only measured by the magnitude of the mAP value but also by the model's ability to maintain detection stability when faced with variations in lighting, camera angles, traffic density, and visual complexity—all of which are commonly encountered in CCTV-based traffic monitoring system implementations.

4.3. Per-Class Detection Performance

A class-level detection performance evaluation was conducted to analyze the model's ability to recognize the visual characteristics of each vehicle type after undergoing base training, fine-tuning, and domain adaptation. This analysis provides a more comprehensive picture than aggregate metrics, as each class has different data distributions, object sizes, and levels of visual complexity. Therefore, interpreting performance on a per-class basis is crucial for identifying the model's strengths and limitations across each vehicle category.

A summary of the evaluation results is presented in Table 3, which shows the precision, recall, mAP@50, and mAP@50–95 values for each vehicle class across the three training stages. In general, the model's performance varies across classes, indicating that the visual characteristics of objects have a direct impact on detection capabilities. The "car" class consistently achieves the best performance on nearly all evaluation metrics, while the "motorbike" class shows relatively lower values compared to the other classes, particularly after the domain adaptation process.

In the base training model, all classes achieved high accuracy levels, with mAP@50 values above 0.90. The "car" and "bus" classes demonstrated a good balance between precision and recall, indicating that the model is capable of consistently recognizing and localizing both types of vehicles in the source domain. This is due to the relatively stable visual characteristics of four-wheeled vehicles, their larger object size, and the sufficient number of training samples, which allowed the feature representations to be learned effectively during the initial training process.

The fine-tuning stage maintains this performance with relatively minor metric changes, as shown in Table 3. Although the increase in mAP values is not particularly significant, the fine-tuning process yields a more stable performance distribution across all vehicle classes. This indicates that adjusting the model parameters reinforces the previously learned feature representations without drastically altering the model's fundamental characteristics. This stability also indicates that the fine-tuning process serves as a model refinement stage before adaptation to more complex domains.

More noticeable changes in performance occur after domain adaptation is applied. All classes experience a decrease in mAP@50 and mAP@50–95 values, though the magnitude of the decrease varies across vehicle categories. The "car" class still maintained relatively high performance with a good balance of precision and recall, while the "bus" and "truck" classes experienced a decline in detection capability, although the precision values for both remained at an adequate level. In contrast, the "motorbike" class showed the most significant performance decline, particularly in the recall and mAP@50–95 metrics, indicating an increase in the number of objects that were not successfully detected in the target domain.

These differences are not solely influenced by the amount of training data but also by the visual characteristics of each class. Two-wheeled vehicles generally have smaller bounding boxes, more varied shapes, and are more frequently occluded in heavy traffic conditions. Additionally, low lighting and light reflections from vehicles at night make it more difficult to distinguish the visual features of motorcycles from the background and other surrounding objects. In contrast, vehicles such as cars, buses, and trucks have larger object dimensions and relatively consistent contours, so their feature representations can be maintained even when the data distribution changes.

The analysis of the bounding box distribution shown in Figure 8, Figure 9, and Figure 10 provides further insight

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

into this phenomenon. In the target domain dataset, the object distribution shows a predominance of small vehicles with more varied positions compared to the source dataset. The variation in the size and location of these objects increases the complexity of the feature extraction process, particularly for the "motorbike" class, resulting in a greater decrease in recall and mAP compared to other classes. In contrast, the distribution of large objects in the "car," "bus," and "truck" classes still provides a sufficiently consistent visual representation for the model to learn from.

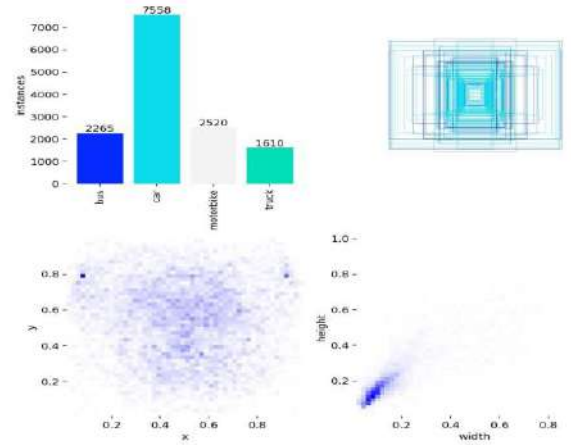


Fig. 8. Label Distribution Plot for Base Training

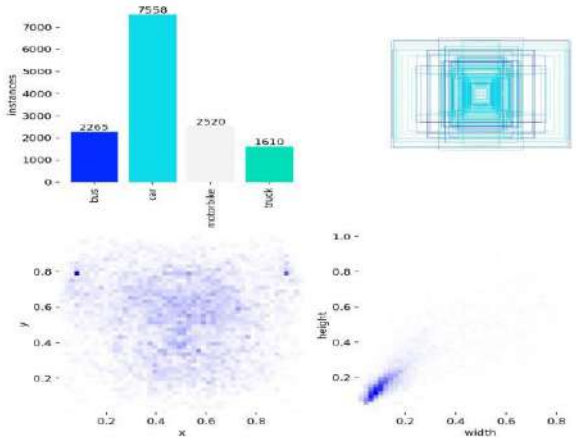


Fig. 9. Label Distribution Plot for Fine-Tuning

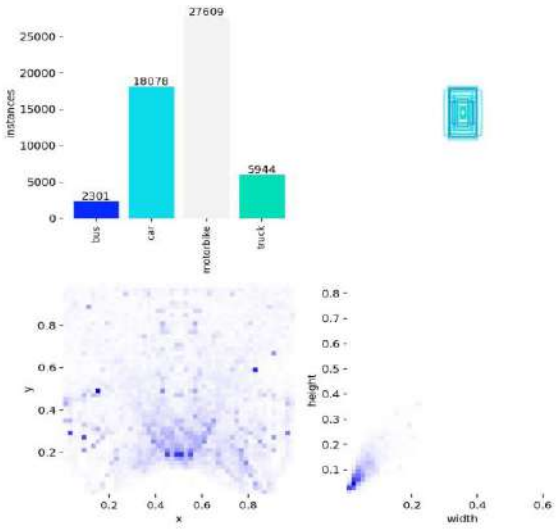


Fig. 10. Label Distribution Graph for Domain Adaptation

Overall, the evaluation at the class level shows that the effectiveness of progressive training is reflected not only in improved aggregate metrics but also in the model's ability to maintain detection performance across each vehicle category. Although the domain adaptation process caused a decrease in accuracy for some classes—particularly motorcycles—the model's ability to distinguish between vehicle types remained well-preserved. These findings indicate that the main challenge in implementing vehicle detection in CCTV environments lies not in the classification process, but in improving detection sensitivity for small objects under low-light conditions, occlusion, and high traffic density. These conditions are among the key aspects that need to be considered in the development of domain adaptation methods in future research.

4.4. Real-world CCTV Evaluation

After undergoing base training, fine-tuning, and domain adaptation, the model was further evaluated in operational scenarios using real-time CCTV video streams. Unlike dataset-based evaluations that rely on annotations as ground truth, testing at this stage aimed to assess the model's ability to maintain detection consistency under dynamic traffic conditions. The evaluation focused on detection stability, the ability to recognize various types of vehicles under diverse lighting conditions, and the model's resilience to visual challenges commonly encountered in urban environments, such as occlusion, vehicle density, changes in camera viewpoint, and lighting disturbances.

Examples of detection results under daytime conditions are shown in Figure 11(a), Figure 11(b), and Figure 11(c), while test results under nighttime conditions are presented in Figure 12(a), Figure 12(b), and Figure 12(c). This visual presentation of the results provides an overview of how the model's prediction characteristics change after each stage of progressive training, ensuring that the evaluation relies not only on quantitative metrics

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

but also takes into account the quality of detection under actual operational conditions.

Fig. 11. Comparison of vehicle detection results under daytime conditions using the (a) Base Training, (b) Fine-Tuning, and (c) Domain Adaptation models.

Under daytime conditions, the base training model was able to detect most vehicles with relatively accurate bounding boxes, as shown in Figure 11(a). Detection of the "car" and "truck" classes was consistent, with relatively high confidence scores. However, for some objects that were overlapping or smaller in size, variations in confidence scores were still observed, indicating uncertainty in the model's classification. This indicates that the feature representations obtained from the initial training are sufficient for environments with optimal lighting but still have limitations when facing higher traffic complexity.

After the fine-tuning process, the detection results in Figure 11(b) show improved prediction consistency compared to the base model. The bounding boxes appear more stable in tracking vehicle positions, while confidence values across classes have become more uniform. This improvement indicates that the fine-tuning process successfully optimized the model parameters so that the feature representations learned in the previous stage can be utilized more effectively. Although the improvement in quantitative metrics at this stage is relatively small, the visual results show that the model produces more stable predictions under heavy traffic conditions.

More interesting changes are observed in the domain adaptation model, as shown in Figure 11(c). Although dataset-based evaluation shows a decrease in mAP compared to the fine-tuned model, testing on CCTV video reveals that the model is able to maintain detection consistency under more diverse environmental conditions. Vehicles located in areas with complex backgrounds or experiencing mild occlusion can still be recognized well. Furthermore, the model demonstrates better ability to maintain detection of objects appearing in various positions within the camera's field of view, indicating that the adaptation process has enhanced the model's generalization ability regarding the characteristics of the target domain.

The differences in model characteristics become even more apparent when testing is conducted under nighttime conditions. In Figure 12(a), the base-training model is still capable of detecting the main vehicle, but confidence scores decrease for most objects. Low-light conditions, glare from vehicle headlights, and reduced object contrast make some vehicles—especially smaller ones—more difficult to recognize. This impact is evident in the reduced detection consistency for the "motorbike" class, which is the class with the highest level of visual complexity.

BIMA
BULLETIN OF INTELLIGENT MACHINES AND ALGORITHMS
Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

(a) (b) (c)

Fig. 12. Comparison of vehicle detection results under nighttime conditions using the (a) Base Training, (b) Fine-Tuning, and (c) Domain Adaptation models.

The results shown in Figure 12(b) indicate that the fine-tuning process improves prediction stability compared to the base model. Detection of four-wheeled vehicles remains effective despite a decrease in lighting intensity. However, variations in confidence scores for objects experiencing occlusion or located in areas with uneven lighting intensity are still evident, indicating that parameter adjustments in the source domain have not yet fully addressed changes in visual characteristics under nighttime conditions.

The model resulting from domain adaptation, shown in Figure 12(c), demonstrates more adaptive performance. The model is able to maintain the number of detected objects even though the observation environment is affected by low lighting and significant glare. For some vehicles, the confidence values are indeed lower compared to daytime testing; however, the objects are still successfully recognized, allowing the tracking process to proceed continuously. These results indicate that domain adaptation does not always lead to higher confidence scores but can enhance the model's sensitivity to the presence of objects in environments that were previously difficult for the initially trained model to recognize.

These visual evaluation results align with the quantitative analysis in the previous section, where the domain adaptation process primarily influenced the balance between precision and recall rather than the classification accuracy across vehicle classes. In real-world implementations, improving detection sensitivity has more significant implications than simply maintaining a high mAP value, as traffic monitoring systems require the ability to recognize as many vehicles as possible so that tracking, volume counting, speed estimation, and density analysis can proceed consistently.

4.5. Inference Optimization Using OpenVINO

After the best model was obtained through the progressive training and domain adaptation stages, the next step focused on optimizing the inference process so that the model could be efficiently implemented on a CPU-based traffic monitoring system. In this study, optimization was performed by converting the YOLOv11 model to the OpenVINO Intermediate Representation (IR) format, followed by the application of INT8-based Post-Training Quantization (PTQ). This approach aims to improve computational efficiency without retraining, ensuring that the model retains the feature representation characteristics acquired during the learning process.

A comparison of inference performance across each model configuration is presented in Table 4, which includes implementations using PyTorch FP32 as the baseline, OpenVINO FP32, OpenVINO FP16, and OpenVINO INT8. The evaluation was conducted using the same hardware, ensuring that the observed performance differences fully reflect the impact of optimization on inference runtime. The parameters analyzed include latency, throughput in frames per second (FPS), and detection metrics such as precision, recall, mAP@50, and mAP@50–95.

TABLE 4. Inference Comparison

Model	Latency (s)	FPS	mAP50	mAP50-95	Precision	Recall
PyTorch	0.2057	4.8609	0.8095	0.5980	0.8249	0.7246
FP32	0.1503	6.6532	0.8095	0.5980	0.8249	0.7246

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

FP16	0.1484	6.7359	0.8095	0.5979	0.8245	0.7248
INT8	0.1101	9.0771	0.8064	0.5822	0.8375	0.7167

The results in Table 4 show that converting the model from PyTorch to OpenVINO FP32 has improved computational efficiency without affecting detection performance. Latency decreased compared to the PyTorch implementation, while processing speed increased from 4.86 FPS to 6.65 FPS, with mAP@50 and mAP@50–95 remaining at the same levels. These findings indicate that the optimizations performed by the OpenVINO runtime are capable of improving execution efficiency through more optimal management of the computation graph without altering the model's behavior during the inference process.

The use of FP16 precision provided a relatively limited performance improvement compared to OpenVINO FP32. The reduction in latency was only minor, while the FPS value saw a not-very-significant increase. Furthermore, changes in the evaluation metrics were barely noticeable, indicating that the use of FP16 numerical representation has not yet provided substantial benefits on the CPU devices used in this study. Thus, the primary benefit of the OpenVINO implementation in this configuration stems more from runtime optimization than from changes in the model's numerical precision.

The best performance was achieved after applying INT8-based Post-Training Quantization. As shown in Table 4, this configuration resulted in the greatest latency reduction as well as an increase in throughput of up to 9.08 FPS. Compared to the initial implementation using PyTorch, this optimization yielded a significant increase in inference speed, making the model better suited for real-time video processing needs. This efficiency gain was achieved by simplifying the numerical representation of the model's parameters from FP32 to INT8, which directly reduces computational requirements during the inference process.

However, the improvement in computational efficiency was accompanied by a relatively small decline in detection performance. The mAP@50–95 values decreased compared to the FP32 configuration, while recall decreased slightly after quantization. Conversely, the precision value actually showed an increase. This pattern indicates that the INT8 model becomes more selective in generating predictions, thereby reducing some false positives, albeit with the consequence of an increased likelihood that some objects will go undetected. Thus, the quantization process results in a trade-off between computational efficiency and detection sensitivity; however, these changes remain within an acceptable range for traffic monitoring applications.

The practical implications of this optimization are more evident in real-time system implementations. A comparison of inference results using PyTorch and OpenVINO INT8 on a CCTV video stream is shown in Figure 13. Visually, the OpenVINO configuration produces more responsive bounding box updates with more consistent inter-frame intervals compared to the PyTorch implementation. This improved stability has a direct impact on the tracking process, as vehicle identities can be better maintained when objects move continuously within the camera's field of view. Additionally, more uniform processing intervals enhance the reliability of vehicle speed estimates, which are calculated based on the object's positional displacement between frames.

Fig. 13. Comparison of Inference with Real-time Video

The benefits of inference optimization are reflected not only in increased FPS but also in the overall stability of the system during the traffic analysis process. With shorter inference times, the system is able to generate information updates more quickly, allowing for more continuous vehicle volume counting, speed estimation, density analysis, and Level of Service (LOS) determination. The consistency of the system's output can be observed through the example analytical results presented in Table 4 and the real-time detection visualization in Figure 14, which show that the inference process remains stable even though the video is processed in real time.

TABLE 4. Example CSV Output

Start	End	Volume	PCU/h	Density PCU/km/lane	...	Average Speed Total	LOS (pendekatan HCM-interpretatif)
2026-01-29T21:48:03	2026-01-29T21:49:03	31	726	27.02	...	26.86	B
2026-01-29T21:50:00	2026-01-29T21:51:00	57	1809	55.17	...	32.79	C
2026-01-29T21:51:00	2026-01-29T21:52:00	44	948	60.64	...	15.63	D
2026-01-29T21:52:00	2026-01-29T21:53:00	70	1668	56.84	...	29.35	C
2026-01-29T21:53:00	2026-01-29T21:54:00	44	810	20.28	...	39.95	B

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

Start	End	Volume	PCU/h	Density PCU/km/lane	...	Average Speed Total	LOS (pendekatan HCM-interpretatif)
2026-01-29T21:54:00	2026-01-29T21:55:01	52	1533	54.29	...	28.24	C
...	...	...	...	...	...	...	...
2026-01-29T22:17:18	2026-01-29T22:18:18	62	1767	43.95	...	40.2	C
2026-01-29T22:18:18	2026-01-29T22:19:18	49	1407	33.97	...	41.42	B
2026-01-29T22:19:18	2026-01-29T22:20:18	35	1344	29.3	...	45.88	B
2026-01-29T22:20:18	2026-01-29T22:21:18	47	1140	28.61	...	39.85	B
2026-01-29T22:21:18	2026-01-29T22:22:18	34	1005	33.39	...	30.1	B

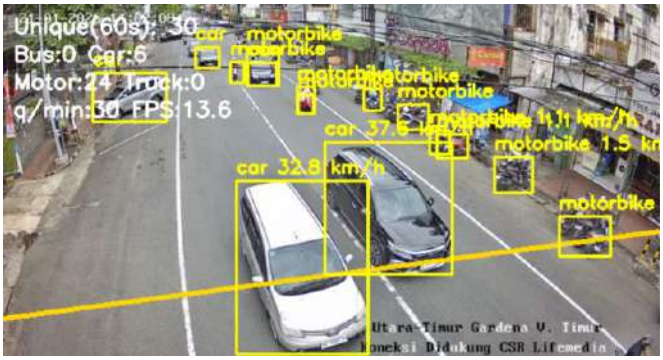


Fig. 14. Visualization of Object Detection Results

Overall, the optimization results show that OpenVINO provides a significant improvement in computational efficiency without causing any meaningful degradation in detection performance. The combination of progressive training, domain adaptation, and OpenVINO INT8-based inference optimization produces a model that not only has adequate detection capabilities but also meets the implementation requirements for CPU-based traffic monitoring systems. These findings indicate that the successful deployment of object detection models in operational environments is determined not only by accuracy but also by the model's ability to maintain a balance between detection quality, inference speed, and processing stability in real-time scenarios.

4.6. Discussion

The results of the study show that a progressive training strategy—comprising base training, fine-tuning, and domain adaptation—is capable of gradually building feature representations, thereby improving the model's adaptability to changes in data distribution. These findings are consistent with the concepts of transfer learning and domain adaptation, which emphasize the transfer and adjustment of feature representations from the source domain to the target domain to improve generalization [12], [15]. Furthermore, the stepwise approach has been shown to yield more stable convergence compared to direct training on the target domain [16]. However, the success of adaptation is not always reflected in an increase in mean Average Precision (mAP), but rather in the ability to maintain detection quality under complex CCTV operational conditions.

Compared to previous studies, the developed model exhibits distinct performance characteristics. Li et al. (2024) reported a 3–5% increase in mAP following domain adaptation, but this was accompanied by a decrease in precision under low-light conditions [8]. Nikouei et al. (2025) found an increase in recall at night, accompanied by a rise in false positives [2]. In contrast, this study shows that although mAP decreases after domain adaptation, precision remains high and stable. This indicates the model's ability to maintain discrimination between vehicle classes, resulting in more selective and consistent detection compared to approaches focused on increasing recall. The following Table 5 summarizes a comparison with previous studies.

TABLE 5. Comparison of the Current Study with Previous Studies

Study	Main Method	Evaluation Focus	Key Results	Strengths	Limitations
[8]	Domain Adaptation	mAP, Precision	mAP increased by 3–5%, while precision decreased under low-light conditions.	Improved overall detection accuracy across domains.	Performance remained sensitive to low-light environments.
[2]	Domain Adaptation	Recall, False Positives	Recall improved during nighttime detection, but false positives also increased.	High detection sensitivity under challenging conditions.	Increased false-positive rate reduced detection reliability.
[12]	OpenVINO-Based Inference Optimization	Throughput, Latency	Throughput improved by approximately 6–7 FPS with lower inference latency.	Significantly enhanced computational efficiency for real-time deployment.	Focused on inference efficiency without evaluating detection

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

Study	Main Method	Evaluation Focus	Key Results	Strengths	Limitations
					robustness or domain adaptation.
This Study	Progressive Training + Domain Adaptation + OpenVINO	mAP, Precision, Detection Stability, Throughput	Base training achieved mAP@50 = 0.919; fine-tuning maintained stable performance (mAP@50 = 0.915); domain adaptation reduced mAP@50 to 0.811 while preserving precision and improving robustness under real-world CCTV conditions; OpenVINO INT8 increased throughput to 9.08 FPS.	Integrates progressive training, domain adaptation, and inference optimization into a unified framework; improves detection robustness, generalization, and real-time deployment efficiency.	Recall improvement remained limited under challenging scenarios, particularly for small and occluded vehicles (e.g., motorcycles).

The decrease in mAP following domain adaptation, accompanied by stable precision, indicates that domain shift affects detection sensitivity (recall) more than discriminative ability. These findings are consistent with previous reports on the impact of low-light conditions, glare, and occlusion on object detection [2], [8]. Although dataset-based metrics declined, visual evaluation demonstrated consistent detection in real-world environments. Therefore, evaluations of domain adaptation should consider detection stability in addition to mAP.

The main contribution of this study is the integration of progressive training, domain adaptation, and OpenVINO inference optimization into a single, comprehensively evaluated framework. Unlike previous studies that examined components separately, this research demonstrates the interdependence of learning stages on generalization ability. Specifically, base training builds foundational representations, fine-tuning stabilizes parameters, and domain adaptation enhances the model's resilience to variations in CCTV conditions. These findings enrich the literature on staged learning strategies for YOLO-based vehicle detection.

In terms of implementation, the OpenVINO optimization increased throughput to 9.08 FPS with a significant reduction in latency, while the decrease in detection performance remained within reasonable limits. Compared to Wang et al. (2025), who reported an increase of 6–7 FPS, these results demonstrate higher efficiency without sacrificing detection stability [12]. This confirms the model's suitability for CPU-based real-time implementation in traffic monitoring systems. Practical implications include improved accuracy in analyzing vehicle volume, speed, and density, as well as determining the Level of Service (LOS).

Overall, this study supports the hypothesis that progressive training enhances the model's generalization ability in CCTV environments with domain shifts. Although absolute accuracy improvements were not observed across all metrics, the model demonstrated resilience to variations in lighting, occlusion, and changes in data distribution. Thus, the success of a vehicle detection system in real-world implementation is determined by a balance between accuracy, generalization, detection stability, and inference efficiency—not solely by the mAP value.

CONCLUSIONS

This study develops a YOLOv11-based progressive training framework that integrates base training, fine-tuning, domain adaptation, and inference optimization using OpenVINO to improve vehicle detection capabilities in real-world CCTV environments. The results show that base training is capable of building robust feature representations, while fine-tuning enhances learning stability without causing performance degradation. Although domain adaptation results in a decrease in mAP due to domain shift, the model demonstrates improved generalization ability and maintains detection quality under complex operational conditions, including low lighting, glare, occlusion, and traffic density. Furthermore, OpenVINO INT8-based inference optimization significantly improves computational efficiency while maintaining detection performance at an acceptable level for real-time implementation. These findings indicate that the success of a vehicle detection system is determined not only by accuracy on the test dataset but also by a balance between generalization ability, detection stability, and inference efficiency. Thus, the proposed framework contributes to the development of smarter, more robust, and adaptive transportation systems ready for deployment in CCTV-based operational environments. Despite the encouraging results, this study has several limitations that should be considered when interpreting the findings. The evaluation was conducted using a single experimental configuration for each stage of the progressive training framework, and therefore did not include repeated training runs or formal statistical significance testing. As a result, the reported performance differences are interpreted based on the observed evaluation metrics and qualitative analyses rather than statistical inference. Future research should incorporate repeated experiments with different random seeds and appropriate statistical analyses, such as paired t-tests, Wilcoxon signed-rank tests, or bootstrap confidence intervals, to further assess the robustness and reproducibility of the proposed framework under diverse operational conditions.

REFERENCES

[1] C. Gheorghe, M. Duguleana, R. G. Boboc, and C. C. Postelnicu, "Analyzing Real-Time Object Detection with YOLO Algorithm in

Vol. 1 No. 5 July 2026
E-ISSN: 3132-5115
DOI: 10.65780/bima.v1i5.27

Automotive Applications: A Review," *CMES*, vol. 141, no. 3, pp. 1939–1981, 2024, doi: 10.32604/cmes.2024.054735.

[2] M. Nikouei *et al.*, "Small object detection: A comprehensive survey on challenges, techniques and real-world applications," *Intelligent Systems with Applications*, vol. 27, p. 200561, Sep. 2025, doi: 10.1016/j.iswa.2025.200561.

[3] B.-S. Indonesia, "Number of Motor Vehicle by Type - Statistical Data." Accessed: Jul. 11, 2026. [Online]. Available: https://www.bps.go.id/en/statistics-table/2/NTcjMg==/number-of-motor-vehicle-by-type--unit-.html

[4] D. J. Marcelleno and M. P. K. Putra, "PERFORMANCE EVALUATION OF YOLOV8 IN REAL-TIME VEHICLE DETECTION IN VARIOUS ENVIRONMENTAL CONDITIONS," *J. Tek. Inform. (JUTIF)*, vol. 6, no. 1, pp. 269–279, Feb. 2025, doi: 10.52436/1.jutif.2025.6.1.3916.

[5] I. Ashari, I. S. M. Negara, and A. S. S. A, "Lightweight YOLO Models for Real-Time Multi-Vehicle Detection," *SinkrOn*, vol. 9, no. 3, pp. 1795–1810, Aug. 2025, doi: 10.33395/sinkron.v9i3.15071.

[6] R. J. Iskandar, C. Fatichah, and A. Yuniarti, "Object Detection in Low-Light Conditions: A Comparison using YOLOv5 and YOLOv8," in *2024 4th International Conference of Science and Information Technology in Smart Administration (ICSINTESA)*, Balikpapan, Indonesia: IEEE, Jul. 2024, pp. 558–563. doi: 10.1109/ICSINTESA62455.2024.10748090.

[7] S. Singh, R. Kumari, P. Pallavi, and P. Saurabh, "A systematic review of deep learning methods for low-light image enhancement and object detection," *Discov Appl Sci*, vol. 8, no. 3, p. 241, Jan. 2026, doi: 10.1007/s42452-025-08051-5.

[8] J. Li, X. Wang, Q. Chang, Y. Wang, and H. Chen, "Research on Low-Light Environment Object Detection Algorithm Based on YOLO_GD," *Electronics*, vol. 13, no. 17, p. 3527, Sep. 2024, doi: 10.3390/electronics13173527.

[9] A. Aldubaikhi and S. Patel, "Advancements in Small-Object Detection (2023–2025): Approaches, Datasets, Benchmarks, Applications, and Practical Guidance," *Applied Sciences*, vol. 15, no. 22, p. 11882, Jan. 2025, doi: 10.3390/app152211882.

[10] R. Saragih, I. Gultom, F. Ikorasaki, and T. M. Nainggolan, "Implementation of YOLOv8 for Object Detection in Urban Traffic Surveillance A Case Study on Vehicles and Pedestrians from CCTV Imagery," *Journal of ICT Aplications and System*, vol. 4, no. 1, pp. 16–25, Jun. 2025, doi: 10.56313/jictas.v4i1.430.

[11] R. Juliansyah *et al.*, "Development of a Real-Time Traffic Density Detection Website Using YOLOv8-Based Digital Image Processing with OpenCV," *Journal of Information Systems and Informatics*, vol. 6, no. 4, pp. 2649–2678, Dec. 2024, doi: 10.51519/journalisi.v6i4.912.

[12] H. Wang and H. Qian, "SDG-YOLOv8: Single-domain generalized object detection based on domain diversity in traffic road scenes," *Displays*, vol. 87, p. 102948, Apr. 2025, doi: 10.1016/j.displa.2024.102948.

[13] L. Folkman, K. A. Pitt, and B. Stantic, "A data-centric framework for combating domain shift in underwater object detection with image enhancement," *Appl Intell*, vol. 55, no. 4, p. 272, Jan. 2025, doi: 10.1007/s10489-024-06224-0.

[14] S. Zhang, H. Tuo, J. Hu, and Z. Jing, "Domain Adaptive YOLO for One-Stage Cross-Domain Detection".

[15] A. Gomaa and A. Abdalrazik, "Novel Deep Learning Domain Adaptation Approach for Object Detection Using Semi-Self Building Dataset and Modified YOLOv4," *World Electric Vehicle Journal*, vol. 15, no. 6, p. 255, Jun. 2024, doi: 10.3390/wevj15060255.

[16] Y. Guo, Y. Yamamoto, H. Yaginuma, and Y. Taniguchi, "Vehicle detection in CCTV with global-guided self-attention and convolution," *Complex Intell. Syst.*, vol. 11, no. 10, p. 458, Sep. 2025, doi: 10.1007/s40747-025-02073-7.

[17] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," Oct. 23, 2024, *arXiv*: arXiv:2410.17725. doi: 10.48550/arXiv.2410.17725.

[18] Y. Wan and W. Zhu, "YOLO-PBE: An Improved YOLOv11 Vehicle Detection Algorithm for Complex Traffic Scenes," *CMC*, vol. 0, no. 0, pp. 1–10, 2026, doi: 10.32604/cmc.2026.084474.

[19] A. K. Sarvaiya, M. K. Vala, B. R. Solanki, A. C. Rathod, and H. R. Khunt, "Size Synergistic Optimization of Preprocessing and Data Augmentation for Accelerated Convergence in YOLO11-based Vehicle Detection," *SSRG-IJECE*, vol. 13, no. 4, pp. 212–221, Apr. 2026, doi: 10.14445/23488549/IJECE-V13I4P117.

[20] Z. Liu, J. Zhang, X. Zhang, and H. Song, "Robust Object Detection in Adverse Weather Conditions: ECL-YOLOv11 for Automotive Vision Systems," *Sensors*, vol. 26, no. 1, p. 304, Jan. 2026, doi: 10.3390/s26010304.