JEECS (Journal of Electrical Engineering and Computer Sciences)
Vol. 11, No. 2, December 2026, pp. 208-217
e-ISSN: 2579-5392 p-ISSN: 2528-0260

YOLO Nano-Based Vehicle Detection for Monitoring Traffic Density

Eko Prasetyo [1], Agus Subhan Akbar [2], Adiananda [3], Henry Nelon Still Heka [4]

[1,4]Department of Informatics, Faculty of Engineering, Universitas Bhayangkara Surabaya, Jl. Ahmad Yani 114, Surabaya, 60231, Indonesia

[2]Department of Information System, Faculty of Sciences and Technology, Universitas Islam Nahdlatul Ulama Jepara, Jl. Taman Siswa, Pekeng, Tahunan, Jepara, 59451, Indonesia

[3]Department of Electrical Engineering, Faculty of Engineering, Universitas Bhayangkara Surabaya, Jl. Ahmad Yani 114, Surabaya, 60231, Indonesia

Article Info

Article history:
Received: 18 August 2026
Revised: 26 October 2026
Accepted: 21 November 2026

Keyword:
Object detection
Police
Traffic monitoring
Vehicle
YOLO

Abstract

Traffic monitoring plays a crucial role in the smart city framework. This responsibility falls under the transportation and police departments, which are tasked with enforcing traffic regulations. To effectively monitor road density in real-time, a system featuring object detection technology is necessary. One such technology is You Only Look Once (YOLO), a lightweight and fast model with competitive performance. The evolution of YOLO, culminating in versions up to YOLO26, demonstrates its maturity as an object detection model. However, there has been a lack of empirical research evaluating YOLO's actual performance in detecting motorized vehicles. Therefore, this study conducted an empirical comparison experiment using the YOLO Nano model across several versions: YOLOv5, YOLOv8, YOLOv10, YOLOv11, YOLO12, and YOLO26. The experiment utilized the public Smart Traffic Light dataset, consists of 832 images and 6,208 instances of three vehicles types: heavy vehicles, cars, and motorcycles. Our experimental results indicate that YOLO12n achieved the most consistent and effective detection. It recorded mAP@50 of 0.956 and mAP@50-95 of 0.802, outperforming other variants. Furthermore, YOLO12n demonstrated high performance in precision, recall, and F1-score, achieving scores of 0.885, 0.932, and 0.908, respectively. These results highlight a balance between accurate predictions and the ability to detect the actual objects presented. Consequently, it is evident that YOLO, particularly its smaller variants, effectively support motor vehicle detection. This capability is invaluable for police and transportation agencies in monitoring vehicles on highways, enhancing overall traffic management.

Corresponding author:
Eko Prasetyo, eko@ubhara.ac.id

DOI: https://doi.org/10.54732/jeecs.v11i2.10

This is an open access article under the CC–BY license.

1. Introduction

Roads serve as public transportation routes that people use for their daily mobility with both private and public vehicles. A variety of vehicles are utilized by the public for road activities: personal mobility typically involves one or two people using motorcycles, while cars are often used by three to eight people. Heavy vehicles are employed for transporting large quantities of goods. Unfortunately, road traffic is not always smooth. Traffic desity tends to be high during peak hours, such as in the morning and evening, while it is usually lower during the daytime and nighttime. Density on the roads quickly lead to significant congestion, sometimes resulting in total traffic jam. Traffic management by the city government is a crucial task within the smart city framework. This requires a system that monitor traffic density quickly and in

real-time [1]. Additionally, vehicle monitoring is part of the police's responsibilities in enforcing traffic regulations [2].

Computer vision-based traffic monitoring utilizes images generated by CCTV cameras installed at various points throughout city roads. Several studies have focused on vehicle detection within this system. For instance, one study employingYou Only Look Once (YOLO) version 8 achieved a precision of 0.91, recall of 0.88, and mean average precision (mAP) of 0.93 [3]. Other research involving YOLO version 4 on CCTV datasets demonstrated an accuracy of up to 78% [4]. Additionally, using YOLOv8m, a study reported mAP scores of 94.8% at 50 and 77.3% at 50-95, based on a dataset of 4,723 images that included five object categories: Car, Motorcycle, Truck, Bus, and Background [5]. Another study utilizing YOLOv8 analyzed 11,157 images, achieving a precision of 96%, recall of 83%, and overall accuracy of 80% [6]. Further developments in YOLOv8 have been investigated in a study involving CCTV images from Madiun City, which integrated the Kalman Filter. This previous research achieved an mAP of 93% at 50, an accuracy of 93%, a precision of 96%, a recall of 85%, and an F1 score of 90% [7]. Another study focusing on the detection of cars, buses, and trucks noted an accuracy of 93% and an mAP of up to 79% at 50 [8]. These studies demonstrate that YOLO has significant potential for advancing smart city support systems, providing rapid and accurate motor vehicle detection on highways. Specifically, YOLO version 8 has been shown to perform well, with reported accuracy rates ranging from 78% to 93%.

The capability of YOLO to detect small objects can also be enhanced through various modifications, such as in the case of DSG-YOLO26. This model incorporates additional components for specific tasks, including Scale-Gated Cross-Level Fusion (SGCF) (which processes upsampled P3 semantics before they enter the P2 branch) and Scale-Sensitive Regression (SSR) for the reweighting of object locations [9]. Another development is YOLO26-EGH, a lightweight and accurate model designed to detect the ripeness stage of blueberries, thereby supporting precision harvesting. This model modifies the YOLO26n architecture by integrating the C2PSA-EMA, C3k2-GCB, and HSFPN modules and employing a knowledge distillation method [10]. Furthermore, the YOLO26 framework has been adapted into CSM-YOLO26m for sheep breed recognition; it integrates an inter-class similarity matrix to address phenotypic similarities in hybrid sheep, achieving an accuracy of 97.75% [11]. YOLO has also been combined with other models; for instance, DINOv3–YOLO26 aims to improve the detection of vegetable crops and weeds by incorporating a fine-tuned, self-supervised learning model (DINOv3 ViT) into the real-time YOLO26 detector architecture, resulting in up to a 5% performance improvement [12].

When an image or video is provided to the YOLO model, it is automatically resized to a standard resolution of 640x640 pixels. The resized image then enters the Backbone network, where it passes through a series of convolutional layers trained with MuSGD and the Small-Target-Aware Label Assignment (STAL) feature. The visual features extracted by the Backbone are passed to the Neck, which aggregates features across various scales. YOLO26 combines the sharp details of high-resolution (small) objects with a global semantic understanding of larger objects, resulting in a feature map rich in contextual information; this ensures that objects are recognized not only by their shape but also by their relationship to their surrounding environment. Subsequently, the features proceed to the final stage, the Head. YOLO26 employs a Dual-Head structure, which eliminates the conventional Distribution Focal Loss (DFL) to improve efficiency during deployment on hardware. Detection is split into two paths: the "One-to-Many Head" path, which generates a large number of traditional predictions and utilizes a post-processing algorithm, Non-Maximum Suppression (NMS), to discard overlapping detection boxes; and the One-to-One Head (NMS-Free) path, which uses a direct assignment method, allowing YOLO26 to predict up to 300 objects instantaneously without producing redundant (duplicate) bounding boxes. However, despite the ongoing advancements in YOLO, notably with the emergence of YOLO26 [13], there has yet to be empirical research validating its actual performance in motor vehicle detection. Furthermore, prior studies have primarily focused on versions of YOLO that are not optimized for size, despite the fact that real-time detection systems require fast and lightweight detection models.

This research presents an empirical study that involves a direct comparison of several versions of the YOLO model, specifically YOLOv5, YOLOv8, YOLOv10, YOLOv11, YOLOv12, and YOLOv26. All models utilize the smallest variant, known as Nano. The study employs the public Smart Traffic Light dataset, available for download from Roboflow, which contains 832 images and 6,208 instances. The dataset includes three types of vehicles: Heavy Vehicles (1,071 instances), Cars (4,135 instances), and Motorcycles (1,002 instances). Model performance is evaluated using metrics such as precision, recall, F1-score, mAP@50, mAP@75, and mAP@50-95 to assess both object localization and the accuracy of the identified vehicle types. The results of this research provide a direct comparison of the latest YOLO versions, specifically their effectiveness in detecting motor vehicles using the smallest variant.

2. Research Methodology

2.1. Research Flowchart

We conducted this research with several systematic steps starting from data collection to conclusions, as presented in Figure 1. The steps are as follows:

1. Collecting Data
   We explored the dataset repository portal to find a dataset for detecting motorized vehicles on the road. This dataset uses a vehicle object detection framework for motorcycles, cars, and heavy vehicles. We found a dataset titled "Smart Traffic Light" on the Roboflow portal. This dataset is a secondary dataset for our initial research in developing an intelligent motorized vehicle monitoring system for traffic enforcement.
2. Study Literature
   This research is supported by a literature review of various research journals published over the past five years to obtain a current overview of object detection research, including architecture, performance, datasets, and potential modifications for development. The literature review led us to experiment with YOLO with its lightest model to create a real-time detection model with acceptable and competitive performance.
3. Experimenting YOLO Nano
   The YOLO model variant chosen for this study is Nano, which features a compact and lightweight architecture. However, its performance is relatively lower compared to other larger architectural variants.
4. Performance Analysis
   We conducted experiments with YOLO, measuring performance through mAP, accuracy, precision, and recall. We evaluated performance on six YOLO Nano models, during both training and testing phases.
5. Performance Evaluation
   Evaluating models is essential for understanding their performance. We consider various performance criteria to ensure a thorough evaluation.
6. Conclusion
   The conclusion is reached by recommending the model with the best performance across multiple aspects. This model presents opportunities for improvements in performance, model size, and sensitivity for tasks related to road vehicles, such as traffic congestion, license plate detection, driver safety features, and driving behavior.

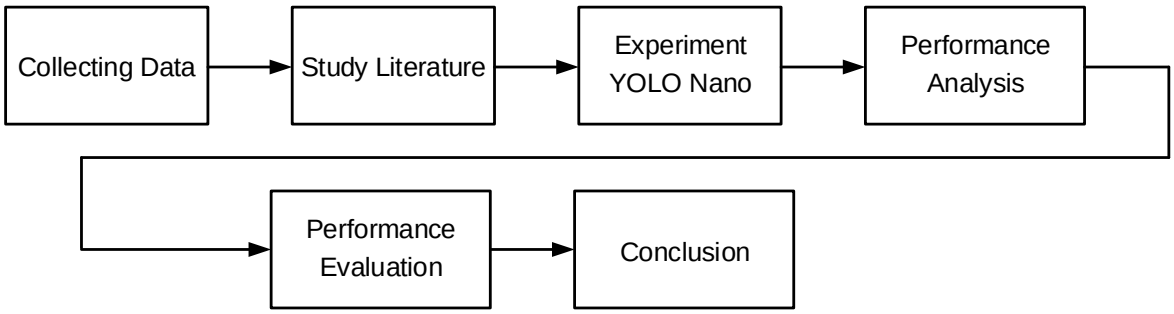

Figure 1. Research Flowchart

Table 1. Dataset description

Class	Train	Validation	Test	Total Instances
Motorcycle	749	68	185	1002
Car	3292	343	500	4135
Heavy Vehicle	943	55	73	1071
Total Instances	4984	466	758	6208
Total Images	672	80	80	

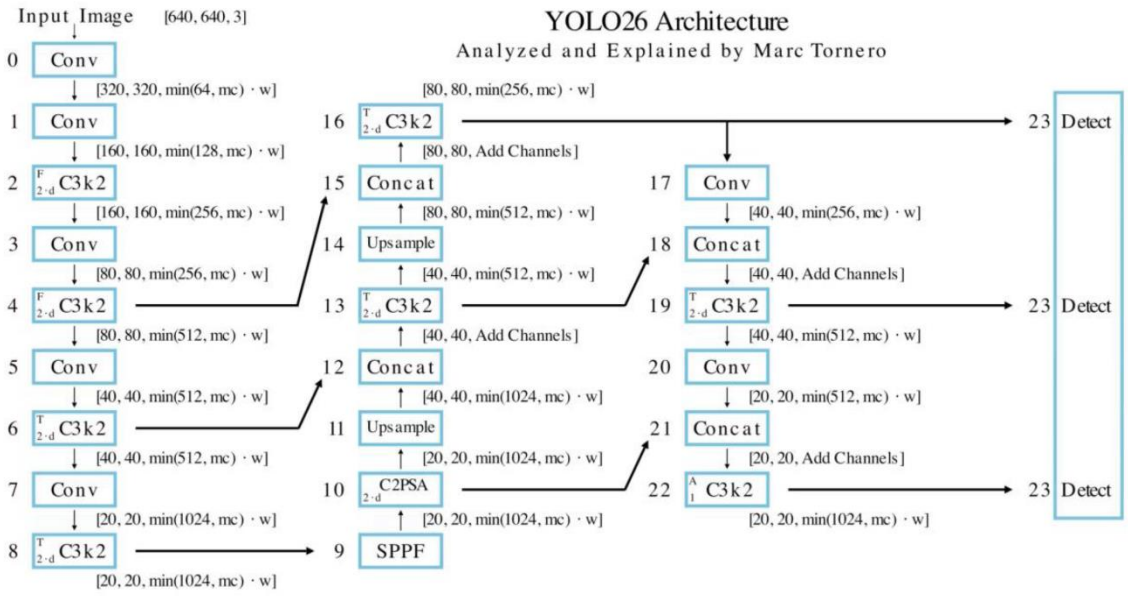

Figure 2. YOLO26 architecture ([14])

2.2. Dataset

We utilize an object detection dataset called Smart Traffic Light, which consists of 832 images extracted from CCTV footage on the highway. This dataset includes a total of 6,208 instances categorized into three types of vehicles: Motorcycle, Car, and Heavy Vehicle. Specifically, there are 1,002 instances of Motorcycles, 4,135 instances of Cars, and 1,071 instances of Heavy Vehicles. To optimize the dataset for training and evaluation, we reduce the instances to 4,986 for training, 466 for validation, and 758 for testing. The details of the dataset are presented in Table 1.

2.3. You Only Look Once (YOLO)

You Only Look Once (YOLO) is a variant of the one-stage detector family designed to achieve fast detection performance after the stages of feature generation, multi-scale feature fusion, and bounding box prediction in a single work sequence. This one-stage innovation offers hope in object detection with a trade-off between performance, computation, and portability with edge devices.

YOLOv5 is YOLO with a feature extraction architecture based on Cross Stage Partial Network (CSP) and Spatial Pyramid Pooling - Fast (SPPF) [15]. It is a pioneering version that provides a very lightweight Nano version (YOLOv5n) for real-time detection. The development of YOLOv8n is the use of CSP Bottleneck with 2 convolutions (C2f), anchor-free detection, and decoupled head, with more flexibility and object localization than the previous version [16]. YOLOv10n also improves efficiency with end-to-end NMS-free and consistent dual assignment. In addition, this version also tries to reduce the need for post-processing [17].

YOLO11n introduces Cross Stage Partial with kernel size 2 (C3k2) and Convolutional block with Parallel Spatial Attention (C2PSA) to improve feature quality and spatial attention to strengthen localization [18]. This model uses only about 2.62 million parameters and 6.5 GFLOPs. Furthermore, YOLOv12n develops an attention-centric approach through Area Attention (A^2) and R-ELAN, to improve context representation with computational efficiency [19].

Lastly, YOLO26n as the latest version is highly oriented towards edge deployment. This model uses around 2.41 million parameters and 5.4 GFLOPs [13]. It also no longer uses Distribution Focal Loss (DFL) and Non-Maximum Suppression (NMS), using Progressive Loss Balancing (ProgLoss), Scale-Targeted Attention Loss (STAL), and MuSGD to improve efficiency, training stability, and small object detection. Thus, YOLO with the Nano variant shows the direction of evolution of increasingly lightweight models, while maintaining detection performance in real-time applications, including vehicle monitoring on highways.

The YOLO26 architecture introduces a significant advancement in object detection efficiency by integrating streamlined core components that prioritize speed without compromising accuracy. The model employs a backbone featuring C3k2 blocks (a CSP Bottleneck structure utilizing optimized mini-kernels) to reduce CPU computational load, complemented by an attention mechanism (C2PSA) that selectively sharpens visual focus on salient features. Meanwhile, the Spatial Pyramid Pooling Fast (SPPF) component acts as a dimensional filter, rapidly aggregating multi-scale spatial context. Furthermore, features are channeled into a Dual-Head Architecture that separates the training path (One-to-Many) from the inference

path (One-to-One). This is bolstered by Small-Target-Aware Label Assignment (STAL), which enables detection of micro-objects without external NMS post-processing. The YOLO26 architecture is presented in Figure 2.

2.4. Metric Performance

The performance evaluation of the YOLO model in object detection uses several key metrics: precision, recall, F1-score, mAP@50, mAP@75, and mAP@50-95. These metrics measure the classification and localization accuracy of the bounding box. In the context of object detection, Intersection over Union (IoU) is the basis for measuring this performance. IoU measures the degree of overlap between the predicted bounding box and the ground truth. The formula for calculating IoU is presented in equation (1).

$$IoU = \frac{Area(B_{pred} \cap B_{gt})}{Area(B_{pred} \cup B_{gt})} \quad (1)$$

where B_{pred} is prediction bounding box, B_{gt} is ground truth bounding box. IoU ranges from 0 to 1, with higher values indicating greater accuracy in object location.

Precision measures the performance of the proportion of correct positive predictions from the predictions made, while recall shows the proportion of actual objects that were successfully found. The formula for calculating precision and recall is according to equations (2) and (3).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

with TP (True Positive), FP (False Positive), dan FN (False Negative). F1-score measures performance by calculating the balance between precision and recall. The formula is according to equation (4).

$$F_{1-score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

The higher the F1-score, the better the balance between false positives and missed objects.

Average Precision (AP) is the area under the Precision-Recall curve, while mAP is the average AP across all classes. mAP@50 calculates AP at IoU = 0.50, thus providing a relatively loose localization tolerance. mAP@75 uses IoU = 0.75 and provides a stricter localization evaluation. Meanwhile, mAP@50-95 calculates the average AP at IoU 0.50 to 0.95 with an interval of 0.05. The formula follows equation (5).

$$mAP_{50-95} = \frac{1}{10} \times \sum_{t=0.50}^{0.95} AP_t \quad (5)$$

This metric provides a more comprehensive evaluation of the detection quality and precision of the bounding box.

2.5. Software Specification

For this research experiment, we utilized Google Colaboratory with an available T4 GPU. The framework and libraries employed include PyTorch (for all YOLO models), Python 3.12, NumPy, Scikit-learn, Matplotlib, Pillow, and OpenCV. The dataset was downloaded from Roboflow and stored in Google Drive, along with pre-trained YOLO weights (.pt files) obtained from https://github.com/ultralytics/ultralytics.

3. Results and Discussions

3.1. Experimental Setup

In this section, we present the results of object detection experiments using the YOLO Nano variant across several models: YOLOv5n, YOLOv8n, YOLOv10n, YOLOv11n, YOLOv12n, and YOLOv12n. These experiments were conducted on Google Colaboratory with GPU resources during the training phase. All models were tested on the same dataset, maintaining consistent YOLO hyperparameter settings. The training, validation, and test datasets were divided as outlined in Table 1, while the hyperparameter settings are detailed in Table 2.

Table 2. YOLO experiment hyperparameter setup

Parameter	Setup
Epoch	50
Batch Size	16
Image Size	640 × 640
Initial learning rate	0.001429
Optimizer	AdamW
Momentum	0.9

Weight decay	0.0005
Augmentatin	• Blur: probability p=0.01, blur limit blur_limit=(3, 7) • MedianBlur: probability p=0.01, blur limit blur_limit=(3, 7) • ToGray: probability p=0.01, method method='weighted_average', number of channels num_output_channels=3 • CLAHE: probability p=0.01, clip limit clip_limit=(1.0, 4.0), grid size tile_grid_size=(8, 8)

3.2. Results

The results of our training and prediction experiments are summarized in Tables 3, 4, and 5. The training experiments utilized both training and validation data, allowing us to assess the performance of all models during training. As presented in Table 3, all YOLO models demonstrate relatively good detection performance. However, there are differences in consistency among the metrics: precision, recall, mAP@50, and especially mAP@50-95. We highlight mAP@50-95 as it serves as the most rigorous indicator; it calculates the average precision across Intersection over Union (IoU) values ranging from 0.50 to 0.95. This metric more accurately reflects the model's ability to produce precise bounding boxes under varying levels of localization difficulty.

According to Table 3, YOLO12n demonstrates the most consistent performance, achieving a precision of 0.8489, a recall of 0.9006, an mAP@50 of 0.9416, and the highest mAP@50-95 of 0.7882. The primary advantage of this model is its mAP@50-95, which indicates the best bounding box localization performance compared to other models. YOLO12n also strikes a good balance between precision and recall, allowing it to effectively locate objects while maintaining prediction accuracy. Another notable model is YOLO8n, which has an mAP@50-95 of 0.7863, a recall of 0.9195, and an mAP@50 of 0.9323. However, its precision is slightly lower at 0.8161. In contrast, YOLO5n boasts a high precision of 0.8581 but has a lower mAP@50-95 of only 0.7761. Lastly, YOLO26n produces the lowest mAP@50-95 at 0.7564. Based on these training results, further development of YOLO12n is highly recommended as a baseline model, especially if the goal is to enhance vehicle localization accuracy in complex road conditions.

We then tested the dataset, and the results are presented in Tables 4 and 5. The data in Table 4 indicate that the YOLO12n model had the most consistent overall performance, as shown in the "All" column. This model achieved a precision of 0.885, a recall of 0.932, and an F1-score of 0.908 across all classes. These results demonstrate a favorable balance between the model's ability to produce accurate predictions and its capacity to detect actual objects. In object detection, precision measures the ability to minimize false positives, while recall assesses the model's effectiveness in identifying all relevant objects. The F1-score combines these two metrics.

Analyzing the class-level performance, YOLO12n showed solid results for heavy vehicles, with a precision of 0.839, a recall of 0.945, and an F1-score of 0.889. For the car class, the model achieved a precision, recall, F1-score of 0.878, 0.936, and 0.906, respectively. The motorcycle class performed even better, with a precision, recall, F1-score of 0.937, 0.914, and 0.925, respectively. These results highlight YOLO12n's strong capability in handling vehicle data with varying characteristics. In comparison, the next best model, YOLO8n, showed an overall recall of 0.931 and an F1-score of 0.892, while YOLO11n produced an F1-score of 0.893. Although YOLO5n achieved a higher precision of 0.854, it had lower recall and F1-score values than YOLO12n. Taken together, these metrics indicate that YOLO12n is the best-performing model, providing a balance of performance across classes and minimizing the gap between precision and recall.

A key performance measure in object detection is Mean Average Precision (mAP), as shown in Table 5. The results indicate that YOLO12n exhibits the most consistent mAP performance overall. This model achieved an mAP@50 of 0.956, which is the highest among the YOLO Nano models. Additionally, at mAP@50-95, YOLO12n also delivered the best performance with a score of 0.802. This metric is particularly stringent because it calculates the average AP at Intersection over Union (IoU) values ranging from 0.50 to 0.95, providing a more accurate representation of object localization through bounding boxes at varying levels of overlap.

Table 3. YOLO performance during training

Model	P	R	mAP50	mAP@50-95
Yolov5n	0.8581	0.8830	0.9361	0.7761
Yolov8n	0.8161	**0.9195**	0.9323	0.7863
Yolo10n	**0.8616**	0.8450	0.9222	0.7716
Yolo11n	0.8181	0.8991	0.9334	0.7750
Yolo12n	0.8489	0.9006	**0.9416**	**0.7882**
Yolo26n	0.8447	0.8474	0.9107	0.7564

Table 4. YOLO performance during prediction with Precision, Recall and F1-score

Model	Precision				Recall				$F_{1\text{-score}}$			
	All	Heavy Vehicle	Car	Motorcycle	All	Heavy Vehicle	Car	Motorcycle	All	Heavy Vehicle	Car	Motorcycle
Yolov5n	0.854	0.822	0.850	0.890	0.910	0.885	0.948	0.897	0.881	0.852	0.896	0.893
Yolov8n	0.857	0.781	0.848	**0.943**	**0.931**	0.927	**0.970**	0.897	0.892	0.848	**0.905**	0.919
Yolo10n	0.880	0.863	0.856	0.922	0.870	0.864	0.914	0.832	0.875	0.863	0.884	0.875
Yolo11n	0.859	0.800	0.852	0.925	0.930	0.904	0.948	**0.937**	0.893	0.849	0.897	**0.931**
Yolo12n	**0.885**	0.839	**0.878**	0.937	**0.932**	**0.945**	0.936	0.914	**0.908**	**0.889**	**0.906**	0.925
Yolo26n	0.874	**0.902**	0.853	0.867	0.842	0.808	0.895	0.822	0.858	0.852	0.873	0.844

Table 5. YOLO performance during prediction with mAP@50, mAP@50-95 and mAP@75

Model	mAP@50				mAP@50-95				mAP@75
	All	Heavy Vehicle	Car	Motorcycle	All	Heavy Vehicle	Car	Motorcycle	All
Yolov5n	0.947	0.918	0.958	0.964	0.781	0.798	0.832	0.712	0.9302
Yolov8n	0.953	0.925	0.965	0.970	0.795	0.807	0.852	0.727	**0.9369**
Yolo10n	0.933	0.902	0.943	0.953	0.778	0.786	0.827	0.721	0.9149
Yolo11n	0.938	0.889	0.954	0.972	0.780	0.786	0.845	0.711	0.9087
Yolo12n	**0.956**	**0.930**	**0.962**	**0.974**	**0.802**	**0.814**	**0.849**	**0.742**	0.9347
Yolo26n	0.918	0.887	0.947	0.921	0.765	0.778	0.829	0.686	0.8937

Breaking down the performance by class, YOLO12n achieved an mAP@50-95 of 0.814 for heavy vehicles, 0.849 for cars, and 0.742 for motorcycles. These results are the highest for both heavy vehicles and cars and demonstrate competitive performance for motorcycles. From a different performance perspective, YOLO8n produced the highest mAP@75 at 0.9369, while YOLO12n achieved 0.9347. However, the advantage of YOLO8n was limited to this specific metric and did not extend to the overall mAP@50-95 performance.

3.3. Detection Results

We present several examples of detection results conducted in this study, as shown in Figure 2. Image 1 and Image 2 have a total of 17 ground truth objects. In the detection results, it can be seen that almost all models can detect all objects in Image 1, but YOLO11n, YOLO12n and YOLO26n detect another object which turns out to be a car. YOLO11n also adds 1 more car object to be detected. In Image 2, only YOLOv5n successfully detects 16 of the 17 objects, while the other models detected 11-15 objects. Image 3 shows that there are 6 objects in the ground truth, and only YOLO11n detects all 6 objects consisting of 1 heavy vehicle and 5 cars.

3.4. Discussions

Let us review the analyses presented earlier. Testing results from all YOLO Nano models on the Smart Traffic Light dataset, evaluated by three different groups, show that YOLO12n consistently delivers the best overall performance. In terms of precision, recall, and F1-score, YOLO12n achieved scores of 0.885, 0.932, and 0.908, respectively. These results indicate a strong balance between correctly predicting objects and accurately detecting actual vehicles. The performance of YOLO12n remains relatively stable across all vehicle classes, with an F1-score of 0.889 for heavy vehicles, 0.906 for cars, and 0.925 for motorcycles.

Furthermore, the reliability of YOLO12n is reinforced by its impressive mean Average Precision (mAP) evaluations. The model achieved an mAP@50 of 0.956 and an mAP@50-95 of 0.802, both of which are the highest among the six models tested. Notably, the mAP@50-95 score reflects enhanced object localization capabilities through bounding boxes at various Intersection over Union (IoU) levels. YOLO12n also stands out with the highest mAP@50-95 performance for heavy vehicles (0.814) and cars (0.849), while achieving competitive results for motorcycles (0.742). Although YOLO8n has a slightly higher mAP@75 score of 0.9369 compared to YOLO12n's 0.9347, we believe this slight advantage does not significantly alter the overall performance trend.

Therefore, our experimental testing results provide clear evidence that YOLO12n is the most consistent model with strong performance, making it an excellent choice for developing vehicle detection systems, especially when precise detection and accurate bounding box localization are essential, even with smaller models. Traffic monitoring systems require lightweight, fast detection models that can be integrated into edge devices. These devices are connected to other systems, such as Command Centers, Electronic Traffic Law Enforcement (ETLE) systems, cloud storage, and health and fire department services. By using lightweight, fast vehicle detection models, traffic data can be transmitted faster to the relevant systems.

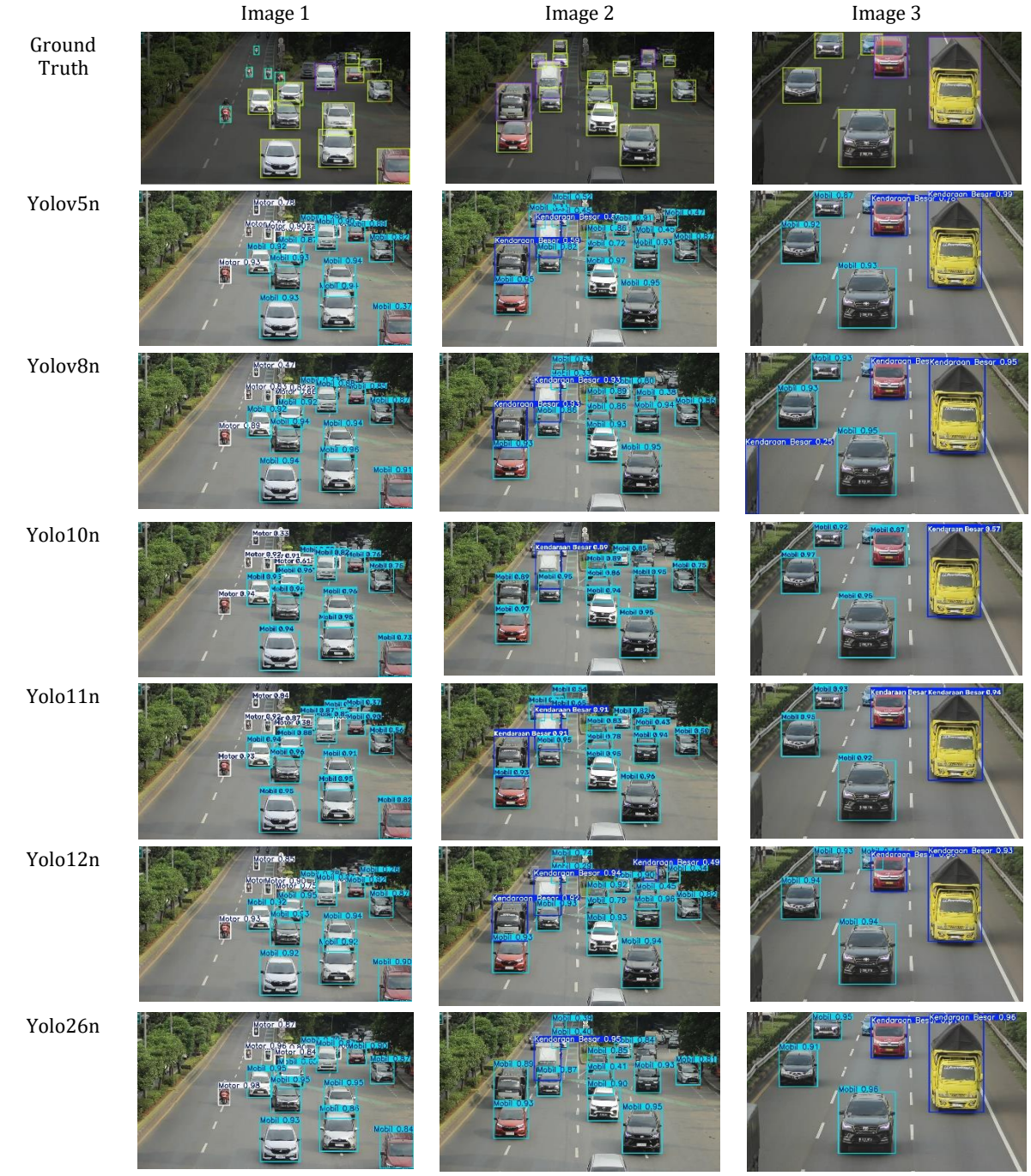

Figure 2. Samples of Detection Results

While YOLO12n shows superior performance, it does have limitations. Its Area Attention and R-ELAN-based architecture enhance feature representation capabilities but also require more memory access and complex computations compared to other object detection architectures. This results in increased memory usage, longer training times, and reduced inference speeds on CPUs or edge devices with limited resources. Additionally, previous research has indicated that YOLO12n may face issues with training stability and hardware sensitivity. Therefore, further optimization of the architecture and extensive testing on real-world edge devices are necessary before fully implementing YOLO12n in operational road vehicle monitoring. These potential challenges are significant concerns for relevant agencies, such as law enforcement and transportation departments.

4. Conclusion

The experimental results and discussions regarding the YOLO Nano variant demonstrate that vehicle detection using the Smart Traffic Light dataset can be effectively accomplished with consistent and reliable performance by the YOLO12n model. This model, a variant of the YOLO12 architecture, employs an attention-centric approach through Area Attention (A^2) and R-ELAN, aiming to enhance context representation while maintaining computational efficiency. In terms of performance metrics, YOLO12n shows impressive results in precision, recall, and mean Average Precision (mAP), outpacing other variants such as YOLOv5n, YOLOv8n, YOLO10n, YOLO11n, and the latest YOLO26n. Specifically, mAP at IoU thresholds of 0.50 and 0.50-0.95 achieves scores of 0.956 and 0.802, respectively. Although the mAP at 0.75 is not the highest, it does not diminish the model's strong performance in more stringent evaluations, such as mAP@50-95. The robustness and consistency of YOLO12n's performance are further evidenced by its scores of 0.885 in precision, 0.932 in recall, and 0.908 in F1-score. These achievements highlight the model's ability to balance accurate predictions with the detection of actual objects. Therefore, it is clear that YOLO with small variations can provide effective object detection performance, particularly for vehicles on highways. In summary, we have empirical evidence that YOLO12n is the most consistent model with solid performance, which can serve as a foundation for developing vehicle detection systems on highways. This advancement directly supports law enforcement and transportation agencies in monitoring vehicles effectively.

Despite the clear advantages of YOLO12n, there is still work to be done. YOLO12n is an object detection model that utilizes Area Attention and R-ELAN to enhance its feature representation capabilities. However, one challenge is that the attention mechanism requires more memory access and is more computationally complex than other object detection architectures. This addition may impact memory usage, training time, and reduce inference speed on CPUs or edge devices with limited resources. Therefore, further research will focus on improving the model's architecture to make it more efficient in terms of computational and memory resource usage while maintaining its detection performance. Additionally, real-world testing on edge devices is needed to address potential training instability and hardware sensitivity.

References

[1] N. Q. Iskandar and A. Ghofilin, "Narrative Review: Penerapan Kecerdasan Buatan untuk Sistem Deteksi Kendaraan di Smart City: Studi Mengenai YOLOv8 dan Optical Character Recognition," *Fuse-teknik Elektro*, vol. 4, no. 2, pp. 95–104, 2024, doi: 10.52434/JFT.V4I2.41890.

[2] R. Y. Arumdani and A. Irawan, "Analisa Peran Polisi Lalu Lintas Sebagai Upaya Penanganan Dan Penanggulangan Pelanggaran Lalu Lintas Tahun 2023-2024 (Studi Kasus di Satlantas Polres Sukoharjo)," *Indonesian Journal of Islamic Jurisprudence, Economic and Legal Theory*, vol. 3, no. 3, pp. 2470–2477, 2025, doi: 10.62976/IJIJEL.V3I3.1294.

[3] P. Kendaraan, J. Raya, M. Yolo, M. Arifudin, and S. Rohmah, "Pendeteksi Kendaraan di Jalan Raya Menggunakan YOLO," *Jurnal Teknik Indonesia*, vol. 4, no. 3, pp. 164–176, 2025, doi: 10.58860/JTI.V4I3.737.

[4] F. Raihan Wahidin, W. Witanti, E. Ramadhan, U. Jenderal Achmad Yani, K. Bandung, and P. Korespondensi, "Pengontrolan Lampu Lalu Lintas Menggunakan Teknologi Deteksi Kendaraan YOLOV4," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 12, no. 5, pp. 975–984, 2025, doi: 10.25126/JTIIK.2025125.

[5] M. V. P. Ardiansyah, J. Sahertian, and R. H. Irawan, "Penerapan Algoritma YOLO dalam Sistem Klasifikasi Kendaraan," *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, vol. 9, no. 2, pp. 1151–1157, 2025, doi: 10.29407/XMVT6268.

[6] P. M. Wening, J. Hendry, and A. W. Santoso, "Traffic Signs Detection System Based on You Only Look Once (Yolov8) using Raspberry Pi," *Jurnal Pekommas*, vol. 10, no. 1, pp. 61–69, 2025, doi: 10.56873/JPKM.V9I1.5832.

[7] L. P. Pratama and M. A. I. Pakereng, "Deteksi kendaraan di lalu lintas menggunakan Kalman Filter dan Yolo v8," *AITI*, vol. 22, no. 2, pp. 192–205, 2025, doi: 10.24246/aiti.v22i2.192-205.

[8] R. T. Elevan, D. Hermanto, and E. P. Widiyanto, "Deteksi Jumlah Kendaraan Roda Empat Menggunakan Yolo," *Integrative Perspectives of Social and Science Journal*, vol. 2, no. 2, pp. 1647–1666, 2025.

[9] S. Chen and S. Liu, "DSG-YOLO26: Detail-Preserving and Scale-Gated YOLO for UAV small object detection," *Signal Processing: Image Communication*, vol. 149, p. 117677, 2026, doi: 10.1016/J.IMAGE.2026.117677.

[10] H. Li *et al.*, "YOLO26-EGH:A lightweight knowledge-distilled detection model for blueberry ripeness

grading in greenhouse," *Scientia Horticulturae*, vol. 366, p. 115123, 2026, doi: 10.1016/J.SCIENTA.2026.115123.

[11] Y. Guo *et al.*, "Recognition of sheep breed containing hybrid samples based on CSM-YOLO26m model," *Computers and Electronics in Agriculture*, vol. 249, p. 111842, 2026, doi: 10.1016/J.COMPAG.2026.111842.

[12] B. Deng and Y. Lu, "DINOv3 meets YOLO26 for enhanced crop-weed detection toward precision vegetable weeding," *Computers and Electronics in Agriculture*, vol. 255, p. 112344, 2026, doi: 10.1016/J.COMPAG.2026.112344.

[13] R. Sapkota, R. H. Cheppally, A. Sharda, and M. Karkee, "YOLO26: Key Architectural Enhancements and Performance Benchmarking for Real-Time Object Detection," *arXiv*, 2025, doi: 10.48550/arXiv.2509.25164.

[14] M. Tornero-Soria, A. J. Sánchez-Salmerón, and E. Vendrell Vidal, "Toward a Deeper Understanding of YOLO26: Block-Level Architectural Analysis and Ablation Studies," *Applied Sciences*, vol. 16, no. 13, p. 6758, 2026, doi: 10.3390/APP16136758.

[15] R. Khanam and M. Hussain, "What is YOLOv5: A deep look into the internal features of the popular object detector," *arXiv*, pp. 1–8, 2024, [Online]. Available: http://arxiv.org/abs/2407.20892.

[16] M. Yaseen, "What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," *arXiv*, pp. 1–10, 2024, doi: 10.48550/arXiv.2408.15857.

[17] A. Wang *et al.*, "YOLOv10: Real-Time End-to-End Object Detection," *Advances in Neural Information Processing Systems*, vol. 37, 2024, doi: 10.52202/079017-3429.

[18] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," *arXiv*, pp. 1–9, 2024, doi: 10.48550/arXiv.2410.17725.

[19] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," *arXiv*, 2025, doi: 10.48550/arXiv.2502.12524.