Performance-Efficiency Tradeoff Analysis of YOLOv8 Variants for Real-Time Multiclass Vehicle Detection in High-Density Traffic

Manuscript received November 25, 2025; revised January 6, 2026

Dede Kurniadi
Department of Computer Science
Institut Teknologi Garut
Garut, West Java, Indonesia
dede.kurniadi@itg.ac.id

Asri Mulyani
Department of Computer Science
Institut Teknologi Garut
Garut, West Java, Indonesia
asrimulyani@itg.ac.id

Nuraisah
Department of Informatics Engineering
Institut Teknologi Garut
Garut, West Java, Indonesia
2106080@itg.ac.id

Abstract*— The growing number of vehicles in Indonesia increases the need for an efficient and reliable traffic monitoring system. In Garut Regency, traffic monitoring is still carried out manually without the support of artificial intelligence, thus limiting the effectiveness of real-time traffic analysis. This study develops and evaluates a CCTV image-based vehicle classification model using YOLOv8 with a focus on application in real-world traffic conditions. The development process follows the Machine Learning Life Cycle (MLLC) stages, including data acquisition, preprocessing, training, and model evaluation. The dataset comprises 1,200 CCTV traffic images from 10 locations in Garut Regency, supplemented by 7,426 additional images from the Roboflow platform to enhance the diversity of viewpoints and visual conditions. To address class imbalance, an undersampling technique is applied so that each vehicle category, motorcycle, car, truck, bus, and public transportation, has a balanced number of instances. Three YOLOv8 variants, namely Nano, Small, and Medium, are trained and evaluated using two testing schemes: a 70:20:10 data split and a 5-fold cross-validation method. Performance evaluation was conducted using the mean Average Precision (mAP), precision, recall, and inference speed metrics. The experimental results show that YOLOv8m with the 5-Fold Cross Validation scheme produces the best performance with mAP@50 of 0.947, precision of 0.932, and recall of 0.883, while YOLOv8n excels in terms of inference speed with an average of ±8.77 ms/frame. These findings suggest that the selection of YOLOv8 variants should consider the balance between accuracy and computational efficiency and confirm the potential of YOLOv8 as an initial component of an automated CCTV-based traffic monitoring system in real-world environments with limited resources.

Keywords*—5-Fold Cross-Validation, Class Imbalance, Traffic Monitoring, Vehicle Detection, YOLOv8.

I. INTRODUCTION

The growth in the number of motorized vehicles in Indonesia is expected to continue until 2023. Based on data from the Central Statistics Agency sourced from the Indonesian National Police, the number of motorized vehicles, including passenger cars, buses, trucks, and motorcycles, was recorded at 157,080,504 units in January 2023 [1]. This figure represents an increase compared to the previous year, which recorded 148,261,817 units. This surge in vehicle numbers has the potential to cause various traffic problems, such as congestion, which can directly impact the smooth flow of vehicles on the roads.

Traffic congestion is a major issue in urban and district areas with high transportation activity. In accordance with Article 249, Paragraph 3 of Law Number 22 of 2009 concerning Traffic and Road Transportation, the Traffic and Road Transportation Information and Communication System Control Center is responsible for providing support in handling violations, congestion, accidents, and other incidents quickly and efficiently [2]. This highlights the need for technology that can support these operational tasks, particularly in terms of traffic monitoring and analysis.

However, based on interviews with the Garut Regency Police, specifically the Information Systems and Communication Technology (SITIK) sub-division, it was discovered that to date, there has been no use of artificial intelligence (AI) in traffic analysis. Traffic monitoring is still conducted manually, namely by directly viewing CCTV footage. No system can automatically classify vehicles, indicating limitations in the use of technology to support efficient traffic monitoring and analysis.

Therefore, this research focuses on developing a vehicle classification model as a first step towards a more comprehensive automated traffic monitoring system. The presence of such a system is expected to support more efficient traffic management, while also contributing to efforts to mitigate future traffic problems. Several previous studies have discussed vehicle classification using various approaches, but they still leave several limitations. Research by [3] The dataset used was limited (comprising only 20 images) and consisted of a narrow range of vehicles, specifically motorcycles and cars, making it non-representative. Another study by [4] achieved 90% accuracy; however, the dataset used was obtained from the internet and consisted of a single video with a limited viewpoint, which does not accurately reflect actual traffic conditions. Meanwhile, research [5] Using the CNN method, which focused only on priority vehicles, resulted in low accuracy (69.23%) and failed to cover other vehicle categories. The next study by [6] used a dataset from Google Images with 86% accuracy, but again faced obstacles because the data did not represent real traffic on the highway. Meanwhile, research based on YOLOv5 [7] recorded 79.8% accuracy; however, the dataset was limited to CCTV footage from a single area, with a small sample size and no comprehensive classification evaluation.

To address these limitations, this study developed a vehicle classification model using YOLOv8, the latest version of YOLOv5 and YOLOv7, offering improvements in network architecture, computational efficiency, and detection accuracy. This study also compared three YOLOv8 variants (Nano, Small, and Medium), which have never been comprehensively explored in the context of vehicle

 EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/IISTR.esl.001702

classification, to determine which variant provided the best performance. Furthermore, this study expanded the category scope by adding public transportation, which has previously been rarely the focus of studies. The dataset used is more representative because it comes from CCTV traffic recordings belonging to the Garut Regency Police at 10 different locations, supplemented with external datasets from the Roboflow platform, which include various camera angles that have undergone labeling and data cleaning processes. The model training process follows the stages of the Machine Learning Life Cycle (MLLC), utilizing two testing schemes: the 70:20:10 data split and the 5-Fold Cross Validation method. The evaluation was conducted using the Mean Average Precision (mAP) metric and a confusion matrix (precision and recall) for five vehicle categories: car, motorcycle, bus, truck, and public transportation. With this approach, the research is expected to yield a more optimal vehicle classification model, making a significant contribution to the development of computer vision-based automated traffic monitoring systems in the future.

II. METHODOLOGY

This study applies an experimental approach to implement a vehicle classification model using YOLOv8 on traffic CCTV footage in Garut Regency. The process is based on the stages in MLLC [8]. All stages are described in detail in Fig. 1.

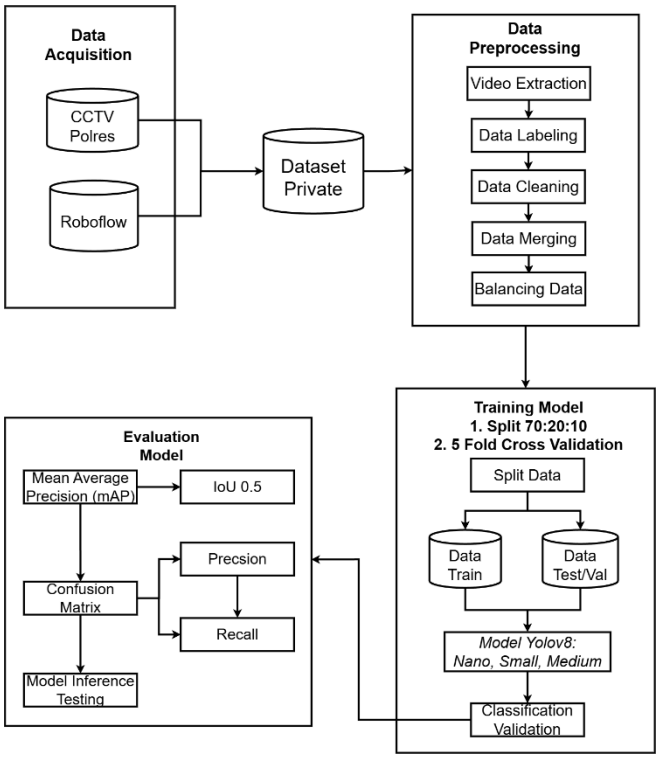

Fig. 1. Research Stages with the MLLC Method Approach

Based on Fig. 1, data were obtained from two sources: CCTV footage from the Garut Regency Police Department and an external Roboflow dataset. The preprocessing stage included video extraction, labeling, cleaning, merging, and class balancing. A random undersampling technique was used to equalize the distribution of data across vehicle classes. This approach was chosen because it is more appropriate for datasets with a large majority class, thus reducing model bias toward the dominant class. Alternatives, such as oversampling, risk data duplication, and increased overfitting, are therefore not used in this study.

Model training was conducted using three YOLOv8 variants (Nano, Small, and Medium) with two evaluation schemes: a 70:20:10 data split and a 5-fold cross-validation method. The selection of YOLOv8 was based on its advantages over previous versions (YOLOv5/YOLOv7), including an anchor-free architecture, more extensive augmentation support, and improved computational efficiency on modern GPU devices. In addition, a comparison of the three variants (n, s, m) was conducted to assess the trade-off between inference speed, computational requirements, and detection accuracy, which has not been widely explored in the context of vehicle classification in Indonesian traffic.

To reduce the risk of overfitting, training was supplemented with augmentation techniques, including flipping, mosaic, copy-paste, scaling, and mixup. The optimization process used AdamW, a variant of Adam that explicitly separates the weight decay process. This optimizer was chosen because it has proven to be more stable in maintaining model generalization on medium- to large-scale computer vision datasets [9], [10]. Training parameters were adjusted incrementally with a low learning rate (0.00015) to ensure stable convergence during the learning process. Evaluation was conducted using Mean Average Precision (mAP@0.5), precision, and recall metrics, which provide a comprehensive overview of the model's effectiveness in recognizing five vehicle categories (car, motorcycle, bus, truck, and public transportation). Additionally, a confusion matrix was used to analyze classification errors per class in more detail.

A. YOLOv8 Model

YOLOv8 is a CNN-based real-time object classification model developed by Joseph Redmon et al. This model works in a single stage to efficiently generate bounding boxes and object probabilities [11]. Its performance evaluation relies on the Average Precision (AP) and Mean Average Precision (mAP) metrics, which measure the agreement between the predicted results and the ground truth through the Intersection over Union (IoU) metric. Released by Ultralytics in January 2023, YOLOv8 supports various computer vision tasks such as object detection, segmentation, classification, tracking, and estimation [12]. Available in five variants: Nano (n), Small (s), Medium (m), Large (l), and Extra Large (x), each variant has different complexity and parameter capacity to balance the needs between speed and accuracy. This study uses three initial variants (n, s, m). The Medium variant was chosen as the upper limit because it provides increased accuracy with a relatively fast training time and still moderate memory consumption.

Each YOLOv8 variant has different architectural complexity and number of parameters, which have implications for detection performance. Conceptually, increasing the number of parameters in variants such as YOLOv8m increases the feature representation capacity, enabling more complex visual pattern recognition, but at the cost of increased inference time and higher memory requirements. Conversely, lightweight variants such as YOLOv8n have fewer parameters, enabling faster inference, but tend to lose detail on small objects. Thus, the selection of a model variant must consider the balance between accuracy and computational efficiency, taking into account the context of implementing a real-time traffic monitoring system. Key

 EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/IISTR.esl.001702

technical references regarding the structure and parameters of the YOLOv8 model can be found in the official Ultralytics documentation [13] as well as the YOLOv8 architecture paper by Terven et al. (2023) [12], which describes the development of an anchor-based to an anchor-free approach for improved classification efficiency.

During training, YOLOv8 implements a loss function such as Complete Intersection over Union (CIoU). The CIoU value can be calculated using Equation (1).

$$L_{cloU} = 1 - ioU + \frac{p^2(b^{gt})}{W_g^2 + H_g^2} + \alpha v \quad (1)$$

with IoU as the Intersection over Union between prediction and ground truth, $p^2(b^{gt})$ as the Euclidean distance between the centers of the bounding boxes, v as a measure of aspect ratio consistency, and α as a controlling factor for v.

Furthermore, to evaluate the detection performance, several evaluation metrics are used, including Mean Average Precision (mAP) and a Confusion Matrix, as well as Precision and Recall. The mAP value is formulated in Equation (2).

$$mAP = \frac{1}{n}\sum_{n}^{i=1} APi \quad (2)$$

where n is the number of classes in the dataset, and APi is the Average Precision value for the i-th class [14].

B. K-Fold Cross Validation

The K-fold Cross-Validation method is a model evaluation technique that divides the dataset into k equal parts (folds). Each iteration uses one fold as validation data, while the rest are used for training [15]. This process is repeated until all folds have been validated with data, and the results are averaged to obtain a stable and representative performance estimate [16]. This method is very useful when the dataset is limited and separate validation data is not available, because all data can be used optimally [17]. However, the evaluation results can be affected by the fold formation method, such as random sampling, which can produce performance variations depending on the distribution of data in each fold [18].

C. Data Augmentation

Data augmentation is a technique that modifies images to ensure they are recognized as distinct samples by a model, while maintaining visual similarity to human perception. The goal is to increase the variety of training data, improve the quality of the dataset, and reduce the risk of overfitting [19], [20]. his process typically includes rotation, random zooming, and flipping to enrich the visual patterns the model learns [21].

D. AdamW optimizer

Adam is an efficient adaptive optimization algorithm and is often used as an alternative to stochastic gradient descent [22]. AdamW is an improved version that explicitly separates the weight decay process to improve regularization [9], [10]. his approach has been proven effective in preventing overfitting and improving training performance, including on computer vision models [23].

III. Results and Discussion

The results of this study were compiled and analyzed based on the implemented Machine Learning Lifecycle (MLLC) stages, which spanned from data collection and processing, through model training, to evaluation and inference testing. Each stage made a significant contribution to the final performance of the YOLOv8-based vehicle classification model.

A. Data Acquisition

The research dataset was obtained from two primary sources: CCTV traffic recordings from the Garut Regency Police and a public dataset from the Roboflow platform. The CCTV data consists of 10 10-minute video recordings taken from 10 different locations in Garut Regency, including Tarogong Square, Bakorwil, Kerkof Roundabout, Suci Roundabout, Ciwalen, Leuwi Daun, Limbangan, Malangbong, Sukadana, and Wanaraja. All recordings were taken during the day with resolutions ranging from 720p to 1080p. An example of the CCTV dataset is shown in Fig. 2.



Fig. 2. Example of CCTV Video Recording Dataset

The second dataset originates from Roboflow, a collection of previously labeled data from other users that is available through the Roboflow platform. This additional dataset comes from 3 different users. The first source [24] contains vehicle images with classes truck, car, and bus, comprising 2,700 images. The second source contains vehicle images with the class bus only [25] includes 2,900 images. The third source [26] includes vehicle images of classes car, motorcycle, and license plates, totaling 1,600 images. The following is a collection of datasets shown in Figure 3.

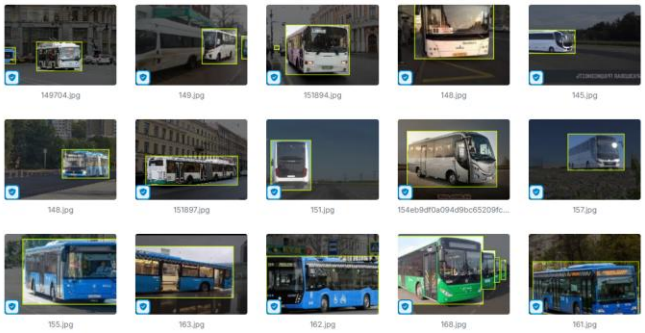

Fig. 3. Example Roboflow Dataset

B. Data Preprocessing

After all the data is collected, a preprocessing stage is carried out to ensure that the data is ready for use in model training.

1) Extraction Video: CCTV footage was extracted into static images (frames) at a 5-second capture interval. Each 10-minute video produced 120 frames, resulting in a total of 1,200 CCTV images. This approach was chosen to reduce temporal redundancy between adjacent frames while

maintaining a diverse representation of traffic conditions. An example of the extraction results is shown in Fig. 4.



Fig. 4. Extracted Raw CCTV Dataset

2) Data Labeling: All extracted images were then annotated using the Roboflow platform using the YOLO format. Five vehicle classes were used: car, motorcycle, truck, bus, and public transportation. The addition of the public transportation class was intended to accommodate local traffic characteristics that have not been widely studied in previous research. An example of the labeling results is shown in Fig. 5.

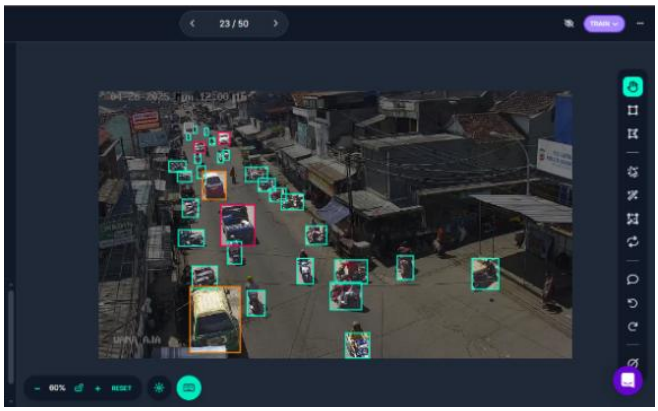

Fig. 5. Example of Roboflow Labeling

Initial analysis revealed a significant class imbalance, with vehicles such as motorcycles and cars being more prevalent than buses and trucks. This imbalance has the potential to impair the model's ability to recognize minority classes, necessitating a strategy for addressing them in the next stage.

3) Data Cleaning: The data cleaning process was performed on both dataset sources. In the CCTV dataset, cleaning involved standardizing file formats, removing invalid annotations, and verifying label consistency. In the Roboflow dataset, cleaning focused on removing irrelevant classes, such as license plates, and low-quality images (blurry or too dark). After this stage, the Roboflow dataset used consisted of 2,700 truck images, 2,069 bus images, 1,000 car images, and 400 motorcycle images.

4) Data Merging: After going through the cleaning stage, a data merging process was performed between the manually labeled dataset from the Police CCTV and the external dataset from Roboflow. These two datasets were then combined to form a more representative and balanced training dataset. The result of the data merging process is shown in Fig. 6.

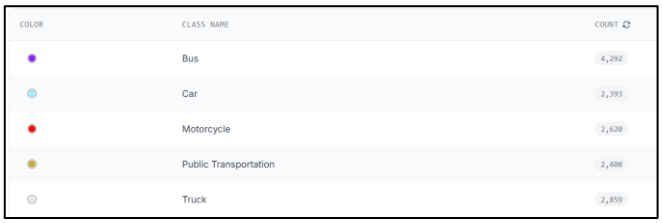

Fig. 6. Dataset Merging Results

Based on Fig. 6, the total number of bus classes reached 4,292 instances even though the additional data was only 2,500. This is because some of the Roboflow data is the result of augmentation of the same frame, such as rotation, flipping, or zooming. After merging, the dataset is compiled into a single version on the Roboflow platform. This step is important for experiment flexibility. During the versioning process, auto-orientation is performed to align the image orientation and resize it to a maximum size of 640×640 pixels using the 'fit within' method, which suits YOLOv8 input requirements and is efficient during training. The following Roboflow version data is shown in Fig. 7. Additionally, the augmentation process increases dataset diversity, helping the model recognize buses under different visual conditions, perspectives, and lighting variations. This contributes to improving model robustness, reducing overfitting risks, and enhancing detection performance when applied to real-world traffic monitoring scenarios with varying environmental conditions and image qualities.

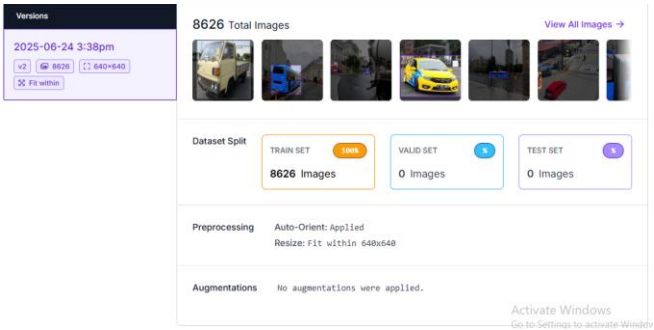

Fig. 7. Roboflow Dataset Version

5) Data Balancing: Analysis of the class distribution after data merging showed a significant imbalance, with the bus class as the dominant class. To reduce potential model bias towards the majority class, a random undersampling technique was applied by equalizing the number of instances for each class to 1,500. The class distributions before and after balancing are shown in Fig. 8 and Fig. 9.

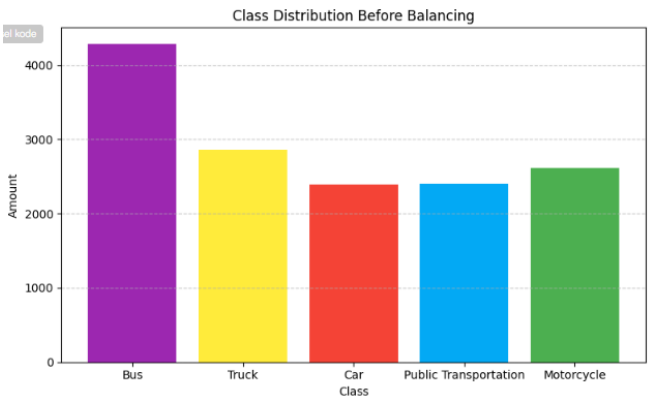

Fig. 8. Class Distribution Before Data Balancing

EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/IISTR.esl.001702

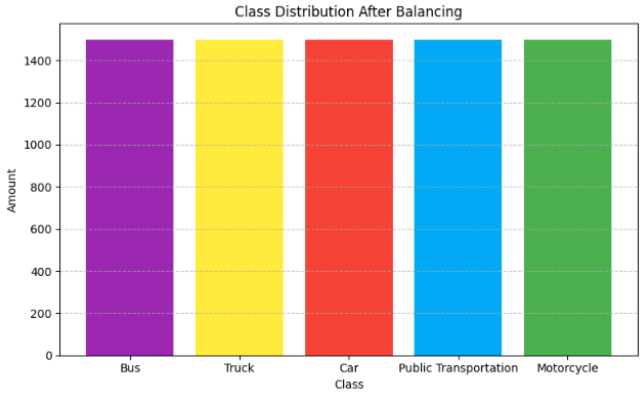

Fig. 9. Class Distribution After Data Balancing

Although undersampling techniques reduce the amount of data from the majority class, this approach was chosen to ensure that the model obtains a balanced representation of the classes during training. However, this strategy also has limitations, as it can potentially eliminate rich variations in the data; therefore, the results of this study need to be understood in this context.

C. *Training Model*

At this stage, model training is performed using data that has undergone preprocessing and class balancing. The training process is carried out using two approaches: proportional data division (70:20:10) and cross-validation.

1) Training Model Split 70:20:10: In the initial stage, training was conducted using the Garut Police CCTV dataset without any additional external data or class balancing. This approach aims to evaluate the basic ability of each YOLOv8 variant to recognize vehicles under limited data conditions. Three YOLOv8 variants (Nano, Small, and Medium) were trained for 50 epochs using the default YOLOv8 configuration with the AdamW optimizer.

TABLE I. YOLOV8 SPLIT 70:20:10 MODEL COMPARISON RESULTS

Model	Class	Image	Precision (%)	Recall (%)	mAP @50
YOLO v8n	Public Transportation	310	0.856	0.85	0.895
	Bus	52	0.4	0.582	0.448
	Car	383	0.723	0.828	0.84
	Motorcycle	395	0.739	0.798	0.806
	Truck	152	0.752	0.763	0.796
	All	400	0.748	0.728	0.784
YOLO v8s	Public Transportation	310	0.9	0.884	0.918
	Bus	52	0.739	0.614	0.634
	Car	383	0.858	0.837	0.894
	Motorcycle	395	0.817	0.82	0.85
	Truck	152	0.81	0.788	0.845
	All	400	0.825	0.788	0.828
YOLO v8m	Public Transportation	310	0.902	0.9	0.922
	Bus	52	0.78	0.65	0.717
	Car	383	0.868	0.85	0.9
	Motorcycle	395	0.827	0.808	0.86
	Truck	152	0.843	0.775	0.846
	All	400	0.844	0.797	0.849

The results in Table 1 show that YOLOv8m produces the best performance with an overall mAP@50 of 0.849, followed by YOLOv8s (0.828) and YOLOv8n (0.784). This difference in performance suggests that increasing architectural complexity offers advantages in extracting vehicle visual features, particularly in classes with high variations in shape and size. However, the performance in the bus and truck classes is still relatively lower than in other classes, indicating the model's limitations when faced with imbalanced data amounts and limited visual variations.

2) Training Model 5-Fold Cross Validation: To obtain more representative performance estimates, training was continued using the 5-Fold Cross Validation method with a combined CCTV and Roboflow dataset that had undergone a class cleaning and balancing process. In this scheme, each fold was trained for 50 epochs with adjusted parameter configurations, including a lower learning rate and the application of dropout to improve learning stability.

TABLE II. PARAMETER

Parameter	Value
Learning Rate	0.00015
Batch Size	16
Number of Epochs	50
Dropout	0.2
Optimizer	AdamW
Weight Decay	0.001

Based on Table 2, training used an optimizer with an initial learning rate of 0.00015, momentum of 0.937, weight decay of 0.001, a cosine scheduler, and dropout of 0.2 to prevent overfitting. The model was trained for 50 epochs with a batch size of 16 per fold. Augmentation was performed using HSV settings, flips, and geometric transformations, including mosaic, mixup, copy-paste, scale, and others. Training utilized an NVIDIA A100 GPU for improved efficiency on large models, such as YOLOv8m. After training with 5-Fold Cross Validation, varying performance results were obtained in each fold for the three YOLOv8 variants:

TABLE III. YOLOV8 BEST MODEL RESULTS

Model	Fold	Precision	Recall	mAP@50
YOLOv8n	4	0.910	0.870	0.939
YOLOv8s	4	0.931	0.870	0.946
YOLOv8m	2	0.896	0.895	0.947
YOLOv8m	5	0.932	0.883	0.947

The training results show that all three YOLOv8 variants have relatively stable performance across folds, with small variations in mAP@50. YOLOv8n achieves an average mAP@50 of around 0.93, YOLOv8s around 0.94–0.95, and YOLOv8m reaches a peak value of up to 0.947. This consistency suggests that the use of cross-validation is effective in reducing bias due to single data splits, providing a more realistic picture of performance for real-world applications.

YOLOv8n demonstrated superior computational efficiency with an average inference time of 8.77 ms/frame. Although its accuracy was lower than that of other variants, this model still maintained an mAP@50 above 0.93 in the cross-validation scheme, making it suitable for resource-constrained edge device deployment scenarios.

YOLOv8s strikes a balance between accuracy and speed. With an mAP@50 of 0.946 and an average inference time of 12.80 ms/frame, the model consistently classified five vehicle types without significant latency increases. These characteristics make YOLOv8s a relevant candidate for real-

time traffic monitoring systems that require an optimal compromise between precision and efficiency.

YOLOv8m demonstrated the highest accuracy performance with a maximum mAP@50 of 0.947 and good inter-fold consistency. This advantage is particularly evident in large vehicle classes such as trucks and buses, which are prone to partial occlusions and shape variations. However, this increase in accuracy is accompanied by a higher inference time, specifically an average of 17.63 ms/frame, making its use more suitable for server-based or GPU systems with adequate computing capacity.

Experimental results show a clear trade-off between accuracy and inference speed across the three YOLOv8 variants. Variants with more complex architectures can extract richer visual features, thereby improving the model's resilience to occlusion and lighting variations, albeit at the cost of increased latency. Therefore, the choice of YOLOv8 variant should be tailored to application needs and hardware limitations, rather than solely based on the highest accuracy value.

a) Comparison with Previous Research: A comparison of the results of this study with several previous studies is shown in Table 7. It is essential to note that this comparison is contextual, given the differences in datasets, class definitions, and evaluation metrics employed in each study.

TABLE IV. ACCURACY RESULTS WITH PREVIOUS RESEARCH

No	Research	Metode	Result Accuracy (%)
1.	This research	YOLOv8 (*Nano, Small, Medium*)	**Result:** YOLOv8n Accuracy 93% YOLOv8s Accuracy 94% YOLOv8m Accuracy 94%
2.	Vehicle Type Classification System Using Digital Image Processing Techniques [3]	Digital Image Processing Techniques	**Result**: 80% Accuracy
3.	Implementation of Real-Time Detection of Vehicle Type Classification in Indonesia Using the YOLOV5 Method [4]	YOLOv5	**Result**: 90% Accuracy
4.	Analysis and Application of Convolutional Neural Network Algorithm for Priority Vehicle Classification [5]	CNN	**Result**: 69% Accuracy
5.	Classification of Vehicle Types on Highways Using Yolov7 [6]	YOLOv7	**Result**: 86% Accuracy
6.	Implementation of the YOLOv5 Method for Vehicle Classification on CCTV in Tulungagung Regency [7]	YOLOv5	**Result**: 79% Accuracy

Overall, the YOLOv8-based approach in this study demonstrates competitive performance compared to previous methods, with higher accuracy and more systematic evaluation through cross-validation. Furthermore, the use of real-world CCTV data from multiple locations provides a more representative picture of actual traffic conditions compared to studies that rely solely on internet datasets or single camera viewpoints. However, these results should be interpreted within the limitations of the testing, particularly regarding the use of daytime data and the class balancing strategy through undersampling.

D. Model Evaluation

Model evaluation was conducted to analyze training stability, prediction quality, and model performance limitations at various evaluation thresholds. The analysis included learning curves, precision, recall, mAP@50, and mAP@50–95 metrics, as well as the confusion matrix for the best-performing model, YOLOv8m, at k-fold 5.

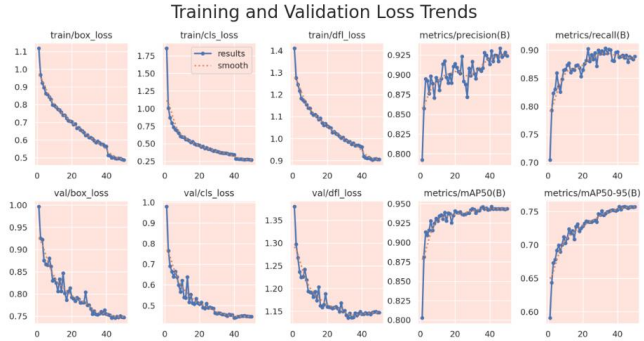

Fig. 10. Training and Validation Loss Trends graph

Fig. 10 shows a consistent downward trend in box loss, classification loss, and distribution focal loss values for both the training and validation data. This pattern indicates a stable learning process with no significant indication of overfitting. Precision and recall values gradually increased and reached a plateau at the end of training, with precision above 0.93 and recall around 0.88.

In addition to mAP@50 reaching values above 0.94, the evaluation of mAP@50–95 shows values approaching 0.75. The performance degradation at higher IoU thresholds reflects the model's challenges in precisely localizing bounding boxes, particularly for objects with partial occlusion, varying sizes, or large camera distances. This finding suggests that while the model is highly reliable in detecting vehicle presence, spatial accuracy can still degrade under complex visual conditions.

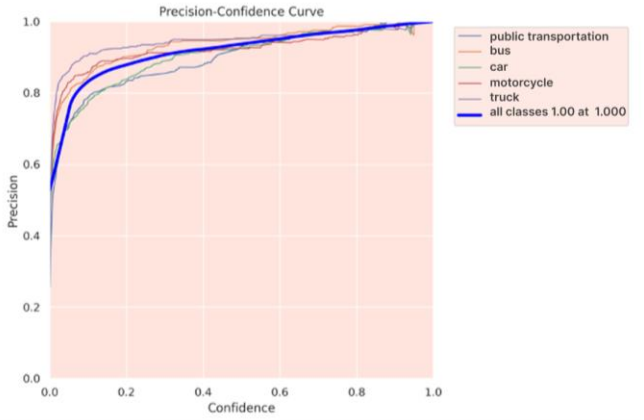

Fig. 11. Graphic of Precision

The precision-confidence curves in Fig. 11 show that increasing confidence values are directly proportional to increasing precision. Public transport, motorcycles, and trucks maintain high precision even at low confidence levels, while buses and cars require a higher confidence threshold to achieve stable predictions. This indicates visual ambiguity in large vehicles with similar shapes.

 EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/IISTR.esl.001702

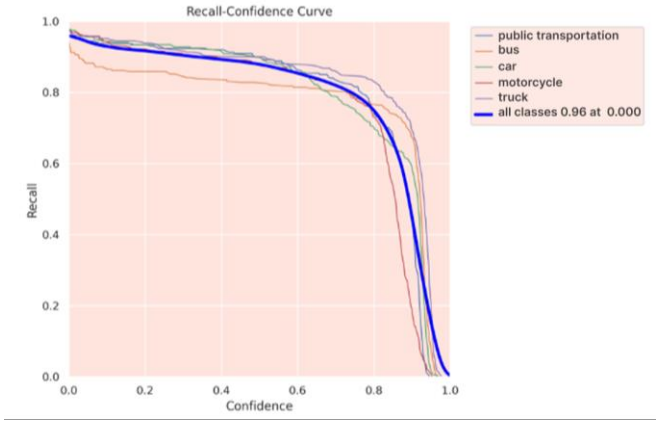

Fig. 12. Graphic Recall

In contrast, the confidence recall curves in Fig. 12 show that the highest recall is achieved at low to medium confidence thresholds, then decreases sharply at high thresholds. This phenomenon reflects a trade-off between model sensitivity and selectivity, where increasing the confidence threshold reduces false detections but increases the risk of false negatives, especially in the bus class.

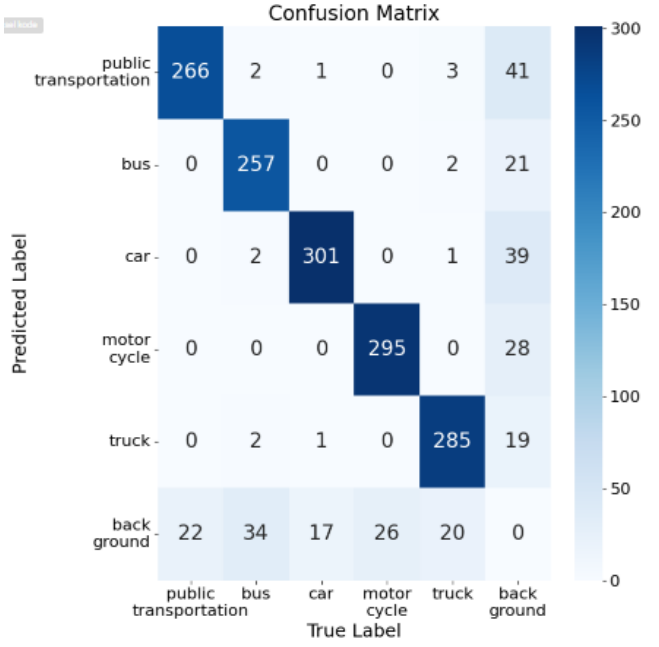

Fig. 13. Confusion Matrix

Fig. 13 shows the confusion matrix, which displays the number of model predictions in relation to the actual labels. The car class had the highest number of correct predictions (301), followed by motorcycles (295), trucks (285), public transportation (266), and buses (257). However, all classes experienced mispredictions in the background, particularly for public transportation (41), cars (39), and motorcycles (28). Background was also frequently misclassified as vehicles, such as public transportation (22) and buses (34). These findings indicate that the model still encounters difficulties distinguishing objects from complex backgrounds, especially in crowded traffic scenes. Similar object shapes, occlusions, and varying illumination conditions likely contributed to these classification inconsistencies.

Next, Figure 14 displays the normalized confusion matrix, which provides an overview of the prediction accuracy for each class as a proportion of the total actual data.

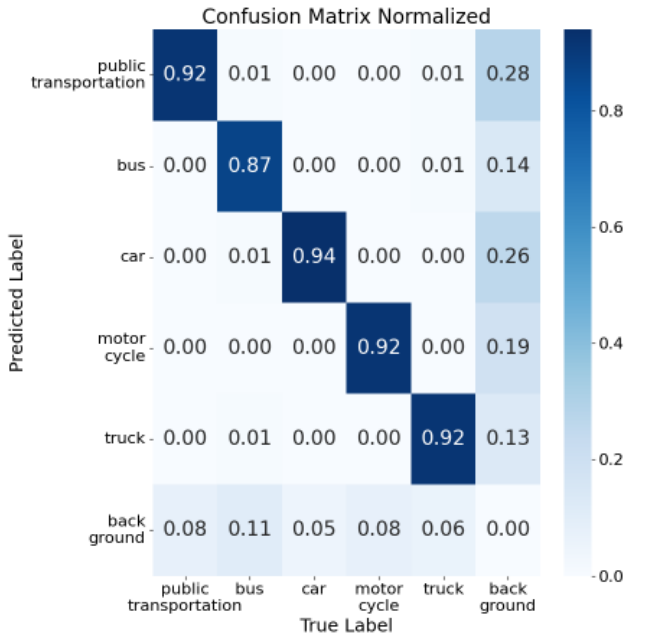

Fig. 14. Normalized Confusion Matrix

In Fig. 14, the normalized confusion matrix shows that the car class has the highest accuracy (0.94), followed by public transportation, motorcycle, and truck (0.92 each), while bus has the lowest accuracy (0.87). The prediction error for the background remains quite high, particularly for cars (0.26), public transportation (0.28), and motorcycles (0.19). This demonstrates the challenge in distinguishing vehicles from the background.

Based on the confusion matrix results, the lower recall values for the Bus and Car classes indicate challenges in consistently distinguishing between the two categories. This is likely due to the similar physical shape of buses and trucks, the large but visually varied object sizes, and occlusion caused by other vehicles partially obscuring the objects. Other factors, such as lighting conditions and camera angles, can also contribute to a reduction in the number of visual features that the model detects.

To improve performance in these classes, further research is recommended to incorporate training data with more diverse occlusion conditions and visual variations, including nighttime imagery and images captured under various weather conditions. Applying augmentation based on occlusion simulations or adding synthetic data could also be potential approaches to enhance the model's generalization to large variations in vehicle appearance. Moreover, experimenting with advanced detection architectures and attention-based mechanisms may further improve object discrimination accuracy in challenging and densely populated traffic environments.

1) Inference Testing: Inference testing was conducted to evaluate the model's generalization ability to new data not previously encountered during the training process. This testing focused on the best-performing YOLOv8m model, using CCTV traffic footage in Garut Regency.

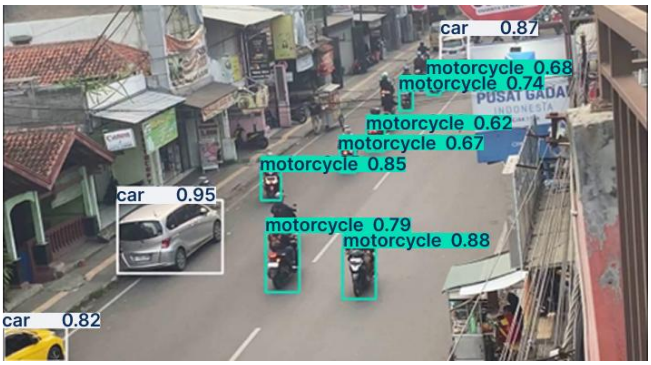

Fig. 15. Inference Testing Results

The inference results in Fig. 15 show that the model can detect and classify various vehicle types simultaneously in a single frame with a relatively high level of confidence, particularly for motorcycles and cars. This indicates that the model is quite reliable for dense traffic scenarios with many objects simultaneously. However, some detections show lower confidence values, particularly for vehicles that are far away, partially obscured, or at extreme viewing angles.

It should be noted that these inference tests were conducted in daylight conditions with a static camera, so the performance obtained may not necessarily reflect nighttime conditions, inclement weather, or low-light environments. Furthermore, the reported inference times do not include the model's prediction process but rather consider only the additional latency due to data acquisition, pre-processing, or post-processing. Therefore, these inference results should be understood as an initial indication of the model's capabilities, not as an evaluation of its end-to-end deployment in production systems.

Nevertheless, the test results show that YOLOv8m has strong potential as a core component in a CCTV-based automated traffic monitoring system. Further performance improvements can be achieved by incorporating training data with a broader range of conditions, including nighttime scenarios, diverse weather conditions, and occlusion simulation-based augmentation.

IV. Conclusion

This study successfully developed and evaluated a YOLOv8-based vehicle classification model using Garut Police CCTV footage data enriched with Roboflow datasets. The evaluation was conducted using two schemes, namely 70:20:10 data splitting and 5-Fold Cross Validation, on three YOLOv8 model variants (Nano, Small, and Medium). The test results showed that YOLOv8m with the 5-fold cross-validation scheme provided the best performance, with mAP@50 of 0.947, precision of 0.932, and recall of 0.883, accurately classifying five vehicle categories.

Further analysis revealed a trade-off between accuracy and inference speed. YOLOv8n had the fastest inference time, making it suitable for resource-constrained devices. YOLOv8s offered a balance between accuracy and computational efficiency, while YOLOv8m excelled in accuracy, albeit at the cost of a longer inference time. These findings highlight the flexibility of the YOLOv8 architecture in accommodating diverse implementation requirements, particularly in automated traffic monitoring systems. However, this study still has limitations, particularly the limited testing scope in daylight conditions with relatively ideal lighting. Furthermore, the lower recall values for the bus and car classes indicate challenges in distinguishing large vehicles that are visually similar and frequently experience occlusion. Therefore, future research is recommended to expand the training and testing data set to more diverse lighting conditions, weather conditions, and camera angles, as well as explore more complex YOLOv8 variants to improve the model's robustness and generalization in real-world traffic conditions.

References

[1] Badan Pusat Statistik, "Jumlah Kendaraan Bermotor Menurut Provinsi dan Jenis Kendaraan (unit), 2023," *.bps.go.id*, 2023.

[2] UU Lalu Lintas & Angkutan Jalan Nomor 22, *Undang-Undang Lalu Lintas Dan Angkutan Jalan Nomor 22 Tahun 2009*, vol. 19, no. 19. 2009.

[3] M. S. Maulana and B. I. Nugroho, "Sistem Klasifikasi Jenis Kendaraan Melalui Teknik Olah Citra Digital," vol. 11, no. 2, 2022.

[4] D. Iskandar Mulyana and M. A. Rofik, "Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5," *Jurnal Pendidikan Tambusai*, vol. 6, no. 3, pp. 13971–13982, 2022, doi: 10.31004/jptam.v6i3.4825.

[5] R. Abdulhakim, Carudin, and B. Arif Dermawan, "Analisis dan Penerapan Algoritma Convolutional Neural Network untuk Klasifikasi Kendaraan Prioritas," *Jurnal Sains dan Informatika*, vol. 7, no. 2, pp. 135–144, 2021, doi: 10.34128/jsi.v7i2.335.

[6] A. S. Pratama, Bayu Aditya, Sayuti Rahman, "Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Yolov7," *Detikproperti*, vol. 5, no. 4, pp. 119–121, 2023.

[7] R. Dwiyanto, D. W. Widodo, and P. Kasih, "Implementasi Metode You Only Look Once (YOLOv5) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung," *Seminar Nasional Inovasi Teknologi*, vol. 1, no. 1, pp. 102–104, 2022.

[8] Ibnu Daqiqil Id, *Machine Learning : Teori, Studi Kasus dan Implementasi Menggunakan Python*, 1st ed., no. July. Riau: UR PRESS, 2021. doi: 10.5281/zenodo.5113507.

[9] X. Chen *et al.*, "Evolved Optimizer for Vision," no. Figure 1, pp. 1–14, 2022.

[10] L. Wright and N. Demeure, "Ranger21: a synergistic deep learning optimizer," 2021.

[11] A. Amwin, "Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (YOLO)," *Universitas Islam Indonesia*, pp. 1–60, 2021.

[12] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[13] Ultralytics, "Ultralytics YOLO Docs," *Ultralytics.com*, 2025.

[14] I Gst Ngr Agung Hari Vijaya Kusuma, "Pengembangan Kursi Roda Otonom Berbasis Yolov8 Untuk Penghindaran Obstacle," vol. 1, 2024.

[15] V. Tiwari, R. C. Joshi, and M. K. Dutta, "Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images," *Ecological Informatics*, vol. 63, no. March, p. 101289, 2021, doi: 10.1016/j.ecoinf.2021.101289.

[16] I. K. Nti, O. Nyarko-Boateng, and J. Aning, "Performance of Machine Learning Algorithms with Different K Values in K-fold Cross-Validation," *International Journal of Information Technology and Computer Science*, vol. 13, no. 6, pp. 61–71, 2021, doi: 10.5815/ijitcs.2021.06.05.

[17] M. Nur, F. Amiruddin, D. A. Navastara, and A. Yuniarti, "Klasifikasi Jenis Kendaraan Darat di Indonesia Menggunakan Pendekatan Deep Learning," vol. 13, no. 3, 2024.

[18] P. Killeen, I. Kiringa, T. Yeap, and P. Branco, "Corn Grain Yield Prediction Using UAV-Based High Spatiotemporal Resolution Imagery, Machine Learning, and Spatial Cross-Validation," *Remote Sensing*, vol. 16, no. 4, pp. 1–39, 2024, doi: 10.3390/rs16040683.

[19] J. Sanjaya and M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.

[20] D. Kurniadi, A. A. Latif, A. Mulyani, and H. Aulawi, "Benchmarking YOLOv8 Variants with Transfer Learning for Real-Time Detection and Classification of Road Cracks and Potholes," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 9, no. 4, pp. 771–780, Aug. 2025, doi: 10.29207/resti.v9i4.6710.

[21] E. Tarisa and F. Ariany, "Perbandingan Kinerja Deep Learning Lenet Dan Alexnet Dengan Augmentasi Data Pada Identifikasi Anggrek," *JIKA (Jurnal Informatika)*, vol. 8, no. 1, p. 51, 2024, doi:

 EISSN: 2961-872X ● PISSN: 2961-8924 ● DOI: 10.56741/IISTR.esl.001702

10.31000/jika.v8i1.9923.
[22] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," *Neural Computing and Applications*, vol. 35, no. 23, pp. 17095–17112, 2023, doi: 10.1007/s00521-023-08568-z.
[23] L. Kah Meng, L. Jia Xin, H. Hooi Yi, Z. Arabee Abdul Salam, and N. Bo Wei, "A Machine Learning Approach for Face Mask Detection System with AdamW Optimizer," *Journal of Applied Technology and Innovation*, vol. 7, no. 3, pp. 2600–7304, 2023.
[24] Parking Area Multi-Object Detection, "Deteksi 3 Jenis Kendaraan Computer Vision Project," *Roboflow*, 2025.
[25] New Workspace, "All dataset bus Computer Vision Project," *Roboflow*, 2022.
[26] Roboflow, "Klasifikasi Kendaraan Computer Vision Project," *Roboflow*, 2024.