Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

Enhancement of YOLOv9 Model for Traffic Vehicle Detection using Augmentation Techniques

Imam Ahmad Ashari*[1,4], Wahyul Amien Syafei[2], Adi Wibowo[3]

[1]Doctoral Program in Information Systems, Universitas Diponegoro, Indonesia
[2]Electrical Engineering, Universitas Diponegoro, Indonesia
[3]Informatics, Universitas Diponegoro, Indonesia
[4]Information Technology, Universitas Harapan Bangsa, Indonesia

Email: [1]imamahmadashari@uhb.ac.id

Received : Jul 30, 2025; Revised : Feb 5, 2026; Accepted : Feb 6, 2026; Published : Apr 15, 2026

Abstract

Traffic vehicle detection is a crucial component in developing intelligent transportation systems, with object detection models like YOLO (You Only Look Once) often preferred for their speed and accuracy. However, challenges remain in detecting vehicles under diverse lighting conditions and small object scales, even with advanced models such as YOLOv9. To address these limitations, image augmentation techniques are employed to enhance model robustness by providing broader data variation. This study investigates the impact of multiple image augmentation methods on the YOLOv9t model for traffic vehicle detection. The techniques evaluated include Blur, Brightness Adjustment, Contrast Adjustment, Color Jitter, Cropping, Flipping, Noise Injection, Rotation, Scaling, and Zoom-In. Results reveal that Scaling and Brightness Adjustment significantly improve detection accuracy, achieving mAP50-95 values of 0.450 and 0.449, respectively. Conversely, methods such as Contrast Adjustment, Rotation, and Cropping produced unsatisfactory outcomes, with Contrast Adjustment performing the worst at only 0.167. Without augmentation, the baseline mAP50-95 was 0.378, emphasizing the vital role of augmentation in improving detection performance, especially under challenging conditions. These findings highlight the importance of selecting appropriate augmentation techniques to optimize YOLOv9t performance, with further improvements possible through combining multiple methods. Compared to approaches that solely focus on enhancing model architecture, the proposed augmentation-based strategy proves more effective in addressing real-world challenges, strengthening resilience against lighting variations and small object detection. This contribution supports the development of more accurate and reliable multilabel vehicle detection systems, advancing safer and more efficient intelligent transportation solutions.

Keywords : *Augmentation Techniques, Traffic Vehicle Detection, YOLO, YOLO9, YOLOv9t model*

1. INTRODUCTION

Traffic vehicle detection has become a critical field in the development of intelligent transportation technology. Models such as YOLO (You Only Look Once) are widely applied because of their speed and accuracy [1][2]. Nevertheless, challenges persist in detecting vehicles under varying lighting and weather conditions, even with the latest YOLOv9 [3][4]. To overcome these limitations, image augmentation techniques provide a practical solution by enriching data diversity and enhancing model performance [5][6].

Image augmentation methods such as contrast enhancement, rotation, and flipping have been proven to improve detection accuracy by equipping models with adaptability to diverse scenarios [7][8]. This capability is crucial in traffic vehicle detection, where conditions are highly dynamic [9][10]. Implementing augmentation within YOLOv9 can therefore strengthen accuracy and reliability in complex environments [11][12].

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

Previous studies indicate that combining robust detection models with appropriate augmentation yields significant performance improvements [13][14]. However, limited research has directly examined the impact of augmentation on YOLOv9 in traffic detection tasks [15][16]. This study aims to fill that gap by exploring and optimizing augmentation techniques to improve YOLOv9's effectiveness in varied conditions [17][18].

By employing YOLOv9t, a fast and precise variant within the YOLOv9 family [19]., this research highlights augmentation as a practical strategy to enhance multi-label vehicle detection. The outcomes are expected to support safer and more efficient intelligent transportation systems [20][21], while offering recommendations on selecting augmentation techniques to address small objects and lighting challenges.

2. METHOD

This research aims to improve the performance of the YOLOv9 model in detecting traffic vehicles through the application of image augmentation techniques. The research methodology consists of several key stages, including data collection, data pre-processing, image augmentation, YOLOv9 model training, model performance evaluation, and result analysis. The workflow of the model is depicted in Figure 1.

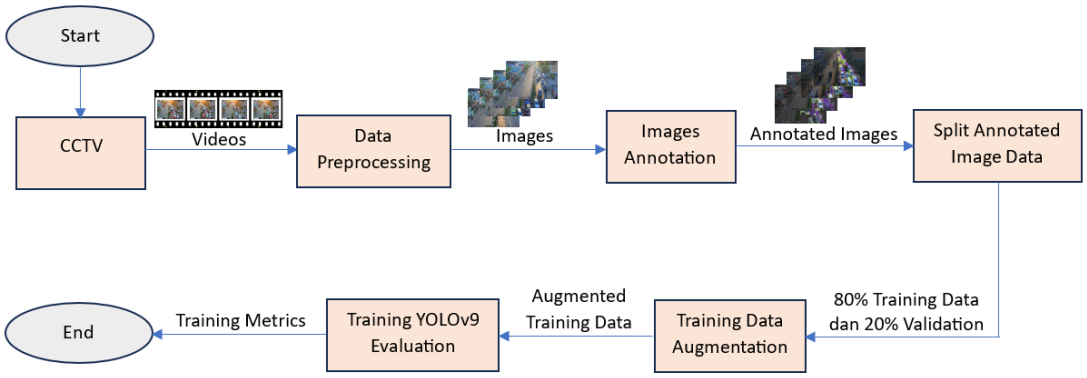

Figure 1. Model Workflows

The vehicle detection process using YOLOv9 begins with collecting CCTV videos, which are converted into image sequences, followed by annotation to label vehicles and other relevant objects. The annotated images are then split into training and validation datasets. A crucial step is applying image augmentation to the training data, aimed at improving the model's robustness by introducing greater variability. The choice of augmentation techniques in this study is driven by the main challenges of traffic vehicle detection, namely the presence of small objects and varying lighting conditions. Techniques such as rotation and flipping are applied to provide different viewing perspectives of vehicles, while brightness and contrast adjustments are used to address changes in illumination during daytime and nighttime. Other methods, such as cropping and scaling, are selected to help the model recognize vehicles of different sizes, and noise injection is introduced to make the model more resilient to visual disturbances. By evaluating these augmentation methods, the primary goal is to identify the most effective strategy to enhance YOLOv9's accuracy in handling dynamic and complex traffic environments.

2.1. Data Collection

The dataset in this research was obtained from surveillance cameras installed at the Fatmawati traffic light in Semarang City. The video was recorded using the H.264/MPEG-4 AVC codec with a

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

resolution of 1280x960 pixels and a frame rate of 25 FPS, providing sharp and smooth image quality. Data collection was conducted from December 19, 2023, to February 15, 2024, daily between 06:00 and 07:00 WIB. The collected data focused on vehicles on the right side of the road. An example of a captured image from the CCTV camera is shown in Figure 2. The data collection process involved extracting frames from the recorded video every 5 minutes. A total of 720 images were generated from the entire extraction process. These datasets were then classified into four main classes: motorcycles, cars, trucks, and buses. Of the dataset, 80% or 576 images were used for training purposes, while the remaining 20% or 144 images were used for validation. This division was designed to ensure that the model received representative data for both training and validation, enabling it to achieve optimal performance in detecting various types of vehicles.

Figure 2. CCTV Camera Captured Image

2.2. Data Pre-Processing

In the data pre-processing stage, CCTV videos were converted into images to enable further analysis. The video was divided into 5-minute frame segments using Python, where libraries such as cv2 handled video processing, NumPy supported array manipulation, os managed file operations, and PIL saved frames as images. The frame interval was determined by multiplying the FPS with the duration in seconds, resulting in one extracted frame every 7,500 frames for a 25 FPS video. This method ensures consistency, as frames are sampled systematically based on frame counts rather than exact timestamps, minimizing errors from variations in video playback or system adjustments.

The script iterated through the video, capturing one representative frame at each 5-minute interval and saving it in the designated directory with a timestamp-based filename. This structured extraction process guarantees that images are consistently captured at uniform time intervals, creating a reliable dataset for subsequent annotation, augmentation, and model training. The approach was selected because it balances efficiency with accuracy, ensuring sufficient temporal coverage of traffic dynamics while avoiding redundant frame data.

2.3. Application of Image Augmentation Techniques

The augmentation techniques aim to alter or modify the original images to produce new variations useful for training the model. The augmentation techniques used include Blur, Brightness Adjustment, Contrast Adjustment, Color Jitter, Cropping, Flipping, Noise Injection, Rotation, Scaling, and Zoom-In. Each technique had different augmentation factors applied to the images [22][23].

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

The blur technique used kernel size to blur the image with varying levels of augmentation. Brightness Adjustment and Contrast Adjustment modified the brightness factor and alpha of the image to produce different variations in brightness and contrast. Color jitter involved changing the brightness, contrast, saturation, and hue with random factors to create more color variations. The cropping technique involved cutting the top or bottom of the image at a specified proportion to produce perspective variations in the image.

The flipping technique involved horizontal and vertical flipping, which changed the orientation of the image. Noise injection added Gaussian noise to the image to increase texture variation. Rotation changed the orientation of the image at specific angles, such as 90 and 270 degrees. Scaling changed the size of the image within a certain range, while zooming in enlarged the image with a specific augmentation factor. All these techniques were designed to increase the diversity of image data and help train a more robust and reliable model. The values of each applied augmentation technique can be seen in Table 1.

Table 1. Image Augmentation Values

No	Aug	Value	Augmentation Factor (Image)			Ref
			1	2	3	
1	Blur	Kernel Size	-	1	2	[24]
2	Brightness Adjustment	Brightness Factor	-	0.8	1.2	[25]
3	Contrast Adjustment	Alpha	-	1.5	2.0	[26]
4	Color Jitter	Brightness	-	Rand(0.6, 1.4)	Rand(0.6,1.4)	[27]
		Contrast	-	Rand(0.6, 1.4)	Rand(0.6, 1.4)	
		Saturation	-	Rand(0.6, 1.4)	Rand(0.6, 1.4)	
		Hue	-	Rand(-25.5, 25.5)	Rand(-25.5, 25.5)	
5	Cropping	Crop Height	-	Top Crop (0,0) to (width, height * 0.5)	Bottom Crop (0, height * 0.5) to (width, height)	[28]
6	Flipping	Horizontal and Vertical Flip	-	Horizontal Flip x_center = 1.0 - x_center	Vertical Flip y_center = 1.0 - y_center	[29]
7	Noise Injection	Gaussian Noise	-	Rand(0,0.1)	Rand(0,0.1)	[28]
8	Rotation	Rotation	-	90’	270’	[23]
9	Scaling	Scale Image	-	Rand(0.8, 1.2)	Rand(0.8, 1.2)	[30]
10	Zoom In	Zoom In	-	1.2	1.5	[31]

2.4. YOLOv9 Model Training

YOLOv9 is the latest generation in the YOLO family, known for its ability to detect objects in real-time with significant performance improvements. In this version, YOLOv9 introduces five model variants with different scales to meet various application needs: YOLOv9t, YOLOv9s, YOLOv9m, YOLOv9c, and YOLOv9e. Among these variants, the YOLOv9t model is the smallest, offering very high processing speed with only 2 million parameters and 7.7 billion floating-point operations per second (FLOPs). This model is highly efficient in detecting objects in the COCO dataset, which includes 80 object classes, such as cars, motorcycles, buses, and trucks. With an mAPval 50-95 value of 38.3%, YOLOv9t is highly suitable for applications requiring real-time object detection with limited computational resources [32]. Although its accuracy is slightly lower than larger models, the speed and efficiency of this model make it an ideal choice for applications that require fast and efficient processing.

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

In this training, the YOLOv9t model was used and trained for 64 epochs to detect various objects. The training involved a total of 1,728 training data and 144 validation data, processed through several augmentation techniques to increase the dataset's diversity and richness. Each augmentation technique was analyzed separately to assess its effectiveness on the model's performance. The YOLOv9t training process in traffic vehicle detection is shown in Figure 3.

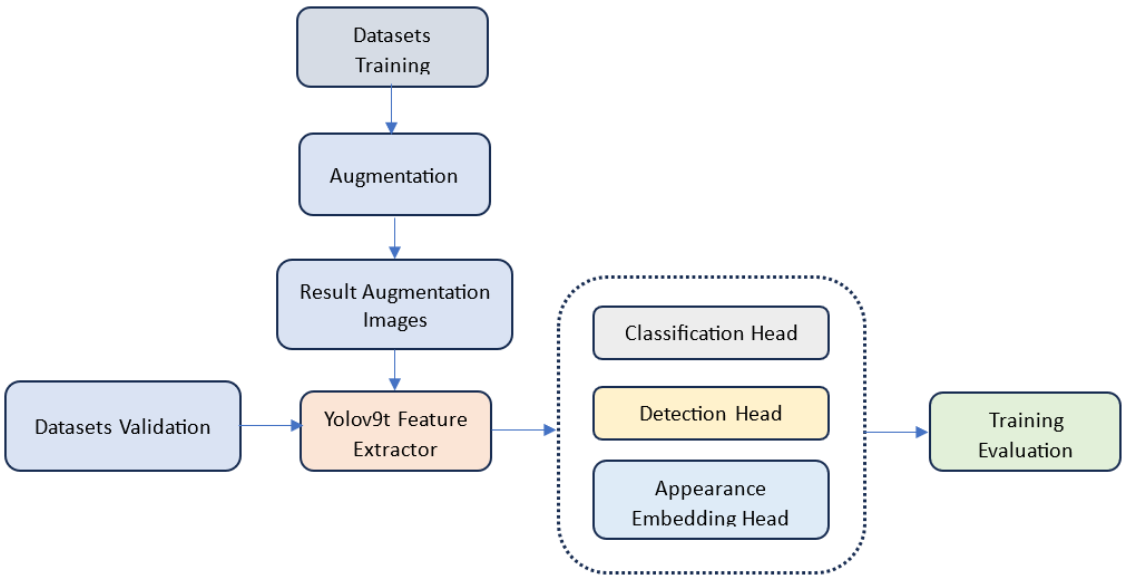

Figure 3. CCTV Camera Captured Image

Figure 3 shows the workflow of the YOLOv9t model training process, starting with the use of the training dataset, which then undergoes augmentation to generate additional image variations. The results of this augmentation are fed into the YOLOv9t Feature Extractor, which extracts essential features from the images. The validation dataset is also processed through this feature extractor to test the model's performance. The YOLOv9t model then uses three main "heads"—Classification Head, Detection Head, and Appearance Embedding Head—to perform classification, object detection, and appearance embedding. Finally, the results of this process are evaluated to measure the model's performance in object detection.

2.5. Tools and Environment Used

Labeling and annotation were conducted using Roboflow, a free online platform that facilitates labeling without requiring additional software installation [33]. The model analysis for this research was conducted in Google Colab, with hardware specifications including 51 GB of RAM, 15 GB of GPU, and 201.2 GB of disk space. Programming was done using Python 3.10.12 and PyTorch 2.3.0, supported by CUDA version 12.1.

2.6. Model Performance Evaluation

The model was trained and validated until the loss function reached a stable state, where the average loss did not undergo significant changes. Object detection quality, which requires drawing bounding boxes around each detected object in the image, was confirmed by evaluating the object detector's performance using three metrics (i.e., precision, recall, and mAP) as shown in Equations 1, 2, and 3 [34].

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \quad (3)$$

Precision is the ratio of the number of True Positives (TP) to the total number of positive predictions (True Positives + False Positives). It measures how accurately the model's positive predictions match actual positives. True Positives refer to correct predictions for existing objects, while False Positives represent incorrect predictions for non-existent objects. Recall, on the other hand, is the ratio of True Positives (TP) to the total number of actual events (True Positives + False Negatives). It evaluates the model's ability to detect all real occurrences. False Negatives are actual events that the model failed to detect. The mean Average Precision (mAP) is the average of the Average Precision (AP) across multiple thresholds. It is a commonly used metric to assess the overall performance of an object detection model. Here, N represents the number of thresholds used, and AP_i is the Average Precision at the i-th threshold. Precision and recall provide insights into the model's accuracy and completeness, while mAP combines these aspects to offer a comprehensive metric for the model's performance across various thresholds.

3. RESULT

3.1. Testing Results

Testing was conducted on several image augmentation techniques using the YOLOv9t model with an image size of 640x640 pixels. This testing aimed to evaluate the impact of various augmentations on the model's performance in detecting objects in images. Table 2 displays the model evaluation results based on several metrics: Precision, Recall, mAP50, and mAP50-95, as well as the number of best epochs obtained during training. Each augmentation technique was tested to observe how each method affects the model's detection performance.

Table 2. Image Augmentation Values

No	Aug	Precision	Recall	mAP50	mAP50-95
1	No Augmentation	0.825	0.556	0.669	0.378
2	Blur	0.830	0.642	0.746	0.443
3	Brightness Adjustment	0.880	0.619	0.749	0.449
4	Contrast Adjustment	0.481	0.097	0.280	0.167
5	Color Jitter	0.878	0.565	0.713	0.411
6	Cropping	0.740	0.559	0.635	0.349
7	Flipping	0.785	0.633	0.719	0.411
8	Noise Injection	0.832	0.625	0.738	0.440
9	Rotation	0.766	0.545	0.653	0.366
10	Scaling	0.812	0.663	0.741	0.459
11	Zoom-In	0.792	0.625	0.716	0.421

Based on the analysis results of the YOLOv9t model with various augmentation techniques, the Scaling technique showed the best performance with an mAP50 score of 0.741 and an mAP50-95 score of 0.459, indicating that this technique helps the model recognize objects of various sizes, thus enhancing detection accuracy in traffic environments. This may be because Scaling makes the model more robust to changes in object size, which frequently occur in traffic situations where vehicles can appear in different sizes depending on their distance from the camera.

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

Other augmentation techniques such as Brightness Adjustment and Blur also yielded good results with mAP50-95 scores of 0.449 and 0.443, respectively, as these techniques increase data diversity in terms of brightness and sharpness, allowing the model to better adapt to variations in lighting and image sharpness in dynamic traffic conditions. In particular, the improvement seen with Brightness Adjustment highlights its effectiveness in handling low-light or overly bright environments, which are common in real-world traffic scenarios such as early mornings or evenings. The ability of the model to maintain relatively high accuracy under these conditions demonstrates that augmentation can effectively simulate lighting variations during training, thereby reducing performance degradation when facing such challenges in real applications.

On the other hand, Contrast Adjustment showed the lowest performance with an mAP50 of 0.280 and mAP50-95 of 0.167, possibly because extreme contrast changes make it difficult for the model to identify object features. Compared to data without augmentation, most augmentation techniques improved accuracy, especially in terms of recall and mAP50-95. This shows that augmentation effectively enriches the training data and strengthens the model's ability to recognize objects in diverse traffic environments.

Overall, using augmentation techniques makes the model more robust and ready to handle condition variations in test data, with Scaling, Brightness Adjustment, and Blur proving to be the most effective techniques, while Contrast Adjustment proved less beneficial. A comparison chart of mAP50-95 accuracy for each augmentation technique can be seen in Figure 4, which further illustrates how certain techniques significantly outperform others, particularly in challenging conditions such as low-light environments.

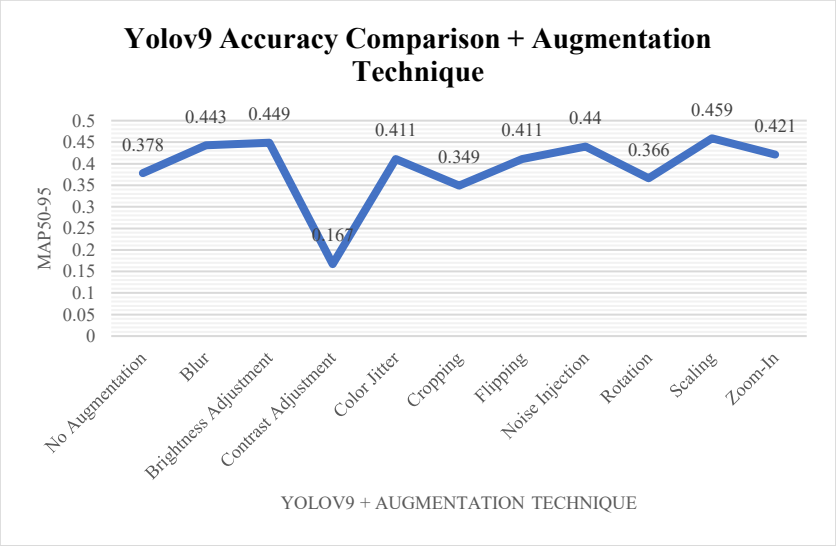

Figure 4. YOLOv9 mAP50-95 Accuracy Comparison with Various Augmentation Technique

3.2. Testing Results of YOLOv9t with Scaling Augmentation Technique

Based on the testing results, YOLOv9t trained with the Scaling augmentation technique shows superior performance compared to other augmentation techniques. The Scaling technique successfully enhances the model's accuracy in detecting vehicles of various sizes and scales, effectively addressing the problem of object size differences in traffic images. This is evident from the higher evaluation metrics, particularly in mAP50-95, precision, and recall, compared to other augmentation techniques. The per-class object evaluation shows significant performance improvement in consistently detecting both small and large objects. Detailed evaluation metrics per object class are presented in Table 3.

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

Table 3. The Per-Class Object Evaluation

Class	Img	Ins	Pre	Rec	mAP50	mAP50-95
Bus	44	54	0.875	0.646	0.783	0.533
Car	141	2301	0.873	0.723	0.803	0.469
Motorcycle	143	5909	0.822	0.505	0.628	0.293
Truck	99	223	0.855	0.646	0.77	0.506

Based on the detection results of YOLOv9t with the Scaling augmentation technique, the model's performance varies for each object class analyzed: Bus, Car, Motorcycle, and Truck. The "Bus" class appears in 44 images with 54 vehicle instances, where precision reaches 0.875, meaning 87.5% of the model's predictions for this class are correct. The recall value of 0.646 indicates that the model successfully detects approximately 64.6% of all "Bus" objects. With an mAP50 value of 0.783 and an mAP50-95 value of 0.533, the "Bus" class demonstrates quite good detection performance. For the "Car" class, which appears in 141 images with a total of 2301 vehicles, precision is 0.873, slightly below the "Bus" class. The recall for this class is 0.723, the highest among all classes, indicating the model's ability to detect most "Car" vehicles. The mAP50 value for the "Car" class is 0.803, while the mAP50-95 is 0.469.

In the "Motorcycle" class, the model detects a total of 5909 vehicles from 143 images with a precision of 0.822 and recall of 0.505. Although the precision is quite high, the low recall indicates that the model struggles to detect all "Motorcycle" objects. This is also reflected in the mAP50 and mAP50-95 values, which are 0.628 and 0.293, respectively, the lowest among all classes. The "Truck" class, with 99 images and 223 vehicle instances, has a precision of 0.855 and recall of 0.646. The model achieves an mAP50 of 0.770 and an mAP50-95 of 0.506. Overall, the class with the best detection performance is "Bus", based on the combination of the highest precision and mAP50-95. Conversely, the "Motorcycle" class shows the worst performance, mainly due to low recall and significantly lower mAP50-95 compared to other classes. This performance disparity can be attributed to the variation in vehicle sizes and shapes, where the model seems to have more difficulty detecting smaller objects or those with more varied features, like "Motorcycle". The confusion matrix values of this model can be seen in Figure 5.

Figure 5 shows the Confusion Matrix of the detection results for four object classes (Bus, Car, Motorcycle, and Truck) and the background using YOLOv9t. The "Motorcycle" class has the highest number of correct predictions with 3500 instances, while the "Bus" class has the lowest number of correct predictions with 35 instances. Some misclassification errors occurred between the "Car" and "Background" classes, where many "Car" instances were classified as "Background."

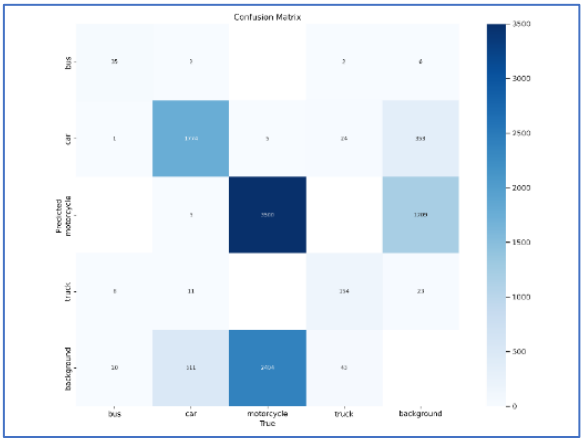

Figure 5. Confusion Matrix

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

Based on the evaluation results, the YOLOv9t model using the Scaling augmentation technique exhibits varying performance across object classes. The "Car" class consistently has the best performance in terms of mAP50, F1, and recall, although there are significant misclassification errors with the "Background" class. The "Motorcycle" class has the worst performance in almost all metrics, including the lowest mAP and steeper declines in the F1 and precision curves, despite having the highest number of correct predictions in the confusion matrix. Overall, the model's performance is strong for some classes but requires improvements in the classification of the "Motorcycle" class and in reducing the misclassification between the "Car" and "Background" classes.

4. DISCUSSIONS

The results of this study demonstrate that image augmentation techniques play a crucial role in improving the performance of the YOLOv9t object detection model, particularly in traffic environments with challenging conditions such as fluctuating lighting and varying object scales. The Scaling technique produced the highest mAP50-95 score, showing its strength in helping the model recognize objects at different distances. Similarly, Brightness Adjustment proved effective in enhancing adaptability to outdoor lighting variations. On the other hand, Contrast Adjustment decreased performance, emphasizing that augmentation methods must be carefully selected to avoid obscuring critical object features.

Per-class analysis revealed that the model was more accurate in detecting larger and consistent objects, such as buses and trucks, while smaller objects like motorcycles remained more difficult to detect due to variability in shape and size. Misclassifications between "Car" and "Background" also indicate that crowded or visually complex traffic scenes continue to pose challenges. Evaluation using recall and mAP50-95 metrics highlighted their importance as more sensitive indicators of model generalization in real-world, multi-label contexts compared to precision.

In conclusion, this study confirms the effectiveness of well-designed augmentation strategies in increasing the robustness of YOLOv9t for vehicle detection. Future research should investigate combining multiple augmentation techniques to leverage their complementary strengths, as well as applying these strategies to alternative detection models such as Faster R-CNN, YOLOv11, or transformer-based architectures. This direction will not only test the generalizability of augmentation but also open opportunities to improve detection of smaller or ambiguous objects, reduce misclassification across classes, and enhance overall model performance in more complex traffic scenarios.

5. CONCLUSION

Based on the study results, it can be concluded that not all image augmentation techniques positively impact the performance of YOLOv9t object detection. The most effective augmentation techniques for improving traffic vehicle detection accuracy are "Scaling" and "Brightness Adjustment," which effectively address issues related to object scale and lighting, achieving mAP50-95 values of 0.450 and 0.449, respectively. These techniques significantly enhanced the model's accuracy in complex vehicle detection scenarios. Conversely, augmentation techniques such as "Contrast Adjustment," "Rotation," and "Cropping" proved less effective, even reducing model performance, with the lowest mAP50-95 value of 0.167 for "Contrast Adjustment." This indicates that the effectiveness of augmentation heavily depends on the specific techniques applied and the characteristics of the dataset used. However, the highest mAP50-95 value achieved in this study is still not optimal. Future research could explore the combination of multiple augmentation techniques to provide more significant accuracy improvements, offer richer data variations, and support model performance in more complex environmental conditions. Therefore, selecting and combining the right augmentation techniques is

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

crucial for enhancing model performance, especially in multi-label traffic environments with challenges such as small objects and poor lighting conditions.

The findings of this study also have direct implications for the development of intelligent transportation technologies, as more accurate and robust vehicle detection models can support real-time traffic monitoring, congestion management, and road safety systems. By improving detection performance under diverse conditions, this research contributes to creating smarter and more reliable transportation infrastructures, ultimately facilitating safer and more efficient urban mobility.

6. IMPLICATIONS

The findings of this study carry several important practical implications for the development and application of intelligent transportation systems. First, the demonstrated effectiveness of Scaling and Brightness Adjustment in improving detection accuracy highlights their potential for direct implementation in real-time traffic monitoring systems. By enhancing the model's ability to detect vehicles of varying sizes and under different lighting conditions, traffic authorities can achieve more reliable data collection from CCTV networks, which is critical for congestion analysis and traffic flow optimization.

Second, the improved robustness of YOLOv9t through carefully selected augmentation techniques can support the deployment of intelligent traffic control systems, such as adaptive traffic lights, which rely on accurate and timely vehicle detection to adjust signal timings dynamically. This not only reduces congestion but also improves road safety by ensuring smoother traffic flow.

Third, the research outcomes are highly relevant for autonomous vehicle development, as robust detection models trained with effective augmentation techniques can enhance the ability of autonomous systems to recognize and respond to vehicles in diverse and challenging environments. This contributes to safer navigation and better decision-making in real-world traffic scenarios.

Lastly, the integration of these findings into smart city initiatives can enable more efficient traffic management, improved public transportation monitoring, and better resource allocation in urban mobility planning. Overall, this study provides practical insights that bridge the gap between theoretical model optimization and real-world applications in intelligent transportation technologies.

CONFLICT OF INTEREST

The authors hereby state that there are no conflicts of interest either between the authors or with the object of study presented in this manuscript.

ACKNOWLEDGEMENT

We extend our gratitude to the Semarang City Transportation Department for their assistance and support in providing a valuable dataset for this research.

REFERENCES

[1] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.

[3] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," pp. 1–7, 2021, [Online]. Available: http://arxiv.org/abs/2107.08430.

[4] Ultralytics, "YOLOv9: A Leap Forward in Object Detection Technology," 2024. https://docs.ultralytics.com/models/yolov9/#what-tasks-and-modes-does-yolov9-support.

[5] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[6] D. L.-P. Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, "Mixup," *Iclr*, no. March, pp. 1–8, 2018.
[7] L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," 2017, [Online]. Available: http://arxiv.org/abs/1712.04621.
[8] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart Augmentation Learning an Optimal Data Augmentation Strategy," *IEEE Access*, vol. 5, pp. 5858–5869, 2017, doi: 10.1109/ACCESS.2017.2696121.
[9] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, no. Section 3, pp. 113–123, 2019, doi: 10.1109/CVPR.2019.00020.
[10] S. Yun, "CutMix," *Iccv*, pp. 6023–6032, 2019.
[11] M. Sandler, A. Howard, M. Zhu, and A. Zhmoginov, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv*, pp. 4510–4520, 2018.
[12] G. Ghiasi *et al.*, "Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2917–2927, 2021, doi: 10.1109/CVPR46437.2021.00294.
[13] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Moco V1," *arXiv*, pp. 9729–9738, 2019.
[14] A. Kolesnikov *et al.*, "Big Transfer (BiT): General Visual Representation Learning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12350 LNCS, pp. 491–507, 2020, doi: 10.1007/978-3-030-58558-7_29.
[15] Q. V. Le Mingxing Tan, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks Mingxing," *Can. J. Emerg. Med.*, vol. 15, no. 3, p. 190, 2013.
[16] Z. Liu *et al.*, "Swin Transformer," *2021 IEEE/CVF Int. Conf. Comput. Vis.*, pp. 9992–10002, 2021, [Online]. Available: https://ieeexplore.ieee.org/document/9710580/.
[17] A. Dosovitskiy *et al.*, "an Image Is Worth 16X16 Words: Transformers for Image Recognition At Scale," *ICLR 2021 - 9th Int. Conf. Learn. Represent.*, 2021.
[18] S. Qiao, L. C. Chen, and A. Yuille, "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 10208–10219, 2021, doi: 10.1109/CVPR46437.2021.01008.
[19] S. Singh, S. Kilaru, S. T. Gupta, and P. Ravisankar, "Traffic Management in India Using YOLOv9 for Emergency and Regular Vehicle Detection," vol. 12, no. 5, pp. 1–8, 2024.
[20] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7," *Cvpr*, pp. 7464–7475, 2023.
[21] X. Li *et al.*, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, no. NeurIPS, pp. 1–11, 2020.
[22] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image Data Augmentation for Deep Learning: A Survey," 2022, [Online]. Available: http://arxiv.org/abs/2204.08610.
[23] P. Oza, P. Sharma, S. Patel, F. Adedoyin, and A. Bruno, "Image Augmentation Techniques for Mammogram Analysis," pp. 1–22, 2022.
[24] A. H. Khan, R. M. Umer, M. Dunnhofer, C. Micheloni, and N. Martinel, "LBKENet:Lightweight Blur Kernel Estimation Network for Blind Image Super-Resolution," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 14234 LNCS, pp. 209–222, 2023, doi: 10.1007/978-3-031-43153-1_18.
[25] M. Pei, N. Liu, B. Zhao, and H. Sun, "Self-Supervised Learning for Industrial Image Anomaly Detection by Simulating Anomalous Samples," *Int. J. Comput. Intell. Syst*., vol. 16, no. 1, 2023, doi: 10.1007/s44196-023-00328-0.
[26] A. Tupper and C. Gagné, "Analyzing Data Augmentation for Medical Images : A Case Study in Ultrasound Images."
[27] M. Pei, N. Liu, B. Zhao, and H. Sun, "Self-Supervised Learning for Industrial Image Anomaly Detection by Simulating Anomalous Samples," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, 2023, doi: 10.1007/s44196-023-00328-0.
[28] N. Eldeen, K. Mohamed, and L. Seyedali, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artif. Intell. Rev.*, vol. 55, no. 3, pp. 2351–2377, 2022, doi: 10.1007/s10462-021-10066-4.

Vol. 7, No. 2, April 2026, Page. 811-822
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.2.5196

[29] K. Alomar, H. I. Aysel, and X. Cai, "Data Augmentation in Classification and Segmentation: A Survey and New Strategies," *J. Imaging*, vol. 9, no. 2, 2023, doi: 10.3390/jimaging9020046.
[30] B. Awaluddin and C. Chao, "Investigating Effective Geometric Transformation for Image Augmentation to Improve Static Hand Gestures with a Pre-Trained Convolutional Neural Network," 2023.
[31] M. Firdaus, M. R. Arief, and U. A. Yogyakarta, "Impact of Data Augmentation Techniques on the Implementation of a Combination Model of Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP) for the Detection of Diseases in Rice Plants.," vol. 2, no. 2, pp. 453–465.
[32] Ultralytics, "YOLOv9: A Leap Forward in Object Detection Technology," *Ultralytics*, 2024. https://docs.ultralytics.com/models/yolov9/#performance-on-ms-coco-dataset.
[33] A. H. N. Hidayah, A. R. Syafeeza, N. A. Razak, W. H. M. Saad, Y. C. Wong, and A. A. Naja, "Disease Detection of Solanaceous Crops Using Deep Learning for Robot Vision," *J. Robot. Control*, vol. 3, no. 6, pp. 790–799, 2022, doi: 10.18196/jrc.v3i6.15948.
[34] S. S. Park, V. T. Tran, and D. E. Lee, "Application of various yolo models for computer vision-based real-time pothole detection," *Appl. Sci.*, vol. 11, no. 23, 2021, doi: 10.3390/app112311229.