Indonesian Journal of Electrical Engineering and Computer Science
Vol. 39, No. 1, July 2025, pp. 656~668
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v39.i1.pp656-668

Boosting real-time vehicle detection in urban traffic using a novel multi-augmentation

Imam Ahmad Ashari[1,2], Wahyul Amien Syafei[3], Adi Wibowo[4]
[1]Doctoral Program of Information Systems, Universitas Diponegoro, Semarang, Indonesia
[2]Faculty of Science and Technology, Universitas Harapan Bangsa, Purwokerto, Indonesia
[3]Faculty of Engineering, Universitas Diponegoro, Semarang, Indonesia
[4]Faculty of Science and Mathematics, Universitas Diponegoro, Semarang, Indonesia

Article Info

Article history:

Received Oct 3, 2024
Revised Mar 12, 2025
Accepted Mar 26, 2025

Keywords:

Multi-augmentation
Multi-label
Urban traffic
Vehicle object detection
YOLO

ABSTRACT

Real-time vehicle object detection in urban traffic is crucial for modern traffic management systems. This study focuses on improving the accuracy of vehicle identification and classification in heavy traffic during peak hours, with particular emphasis on challenges such as small object sizes and interference from light reflections. The use of multi-label images enables the simultaneous detection of various vehicle types within a single frame, providing more detailed information about traffic conditions. You only look once (YOLO) was chosen for its capability to perform real-time object detection with high accuracy. Multi-augmentation techniques were applied to enrich the training data, making the model more robust to varying lighting conditions, viewpoints, object occlusions, and issues related to small objects. YOLOv8n and YOLOv9t were selected for their speed and efficiency. Models without augmentation, 10 single-augmentation techniques, and 5 multi-augmentation techniques were tested. The results show that YOLOv8n with multi-augmentation (scaling, zoom in, brightness adjustment, color jitter, and noise injection) achieved the highest mAP50-95 score of 0.536, surpassing YOLOv8n with single-augmentation Blur, which had an mAP50-95 of 0.465, as well as YOLOv8n without augmentation, which scored 0.390. Multi-augmentation proved to significantly enhance YOLO's performance.

Corresponding Author:

Imam Ahmad Ashari
Doctoral Program of Information Systems, Universitas Diponegoro
Semarang 50241, Indonesia
Email: imamahmadashari@students.undip.ac.id; imamahmadashari@uhb.ac.id

1. INTRODUCTION

As a developing country, Indonesia faces significant challenges related to high population growth, inadequate infrastructure, and increasing traffic congestion [1]. According to the Directorate General of Land Transportation of the Ministry of Transportation of the Republic of Indonesia, traffic accidents in 2019 were primarily caused by motorcycles, which accounted for more than 70% of total traffic accidents [2]. Research conducted in Indonesia, specifically on the Transyogi Cibubur road, shows that vehicle volume increases significantly during peak hours [3]. Estimating traffic density is crucial for improving transportation systems as it helps design more efficient traffic management strategies. Vehicle recognition and counting are the two primary steps in estimating traffic density [4]. The main challenge in this technology is achieving fast, accurate detection in complex environments. State-of-the-art (SoTA) models, trained on diverse datasets such as MS COCO, are robust but often struggle with monotonous backgrounds in surveillance or road videos, leading to reduced performance [5].

The inability to obtain images that meet desired criteria is problematic, resulting in dataset bias, overfitting, and inaccurate outcomes [6]. While deep learning algorithms have demonstrated outstanding performance in various computer vision tasks, limited labeled data can lead to overfitting, hindering the network's generalization to unseen data [7], [8]. Data augmentation is one of the most widely used tools in deep learning, underpinning many recent advances in fields such as classification, generative models, and representation learning [9]. Augmentation modules significantly reduce the need for manual labeling and facilitate the adoption of real-world applications [10]. Additionally, augmentation techniques are highly effective in addressing data scarcity challenges [11]. Data augmentation is applied to increase the number of images in the dataset, as the distribution of images across classes is often uneven [11]. Image augmentation techniques artificially expand the dataset, enabling systems to learn how images appear from different perspectives, such as when viewed from various angles or blurred due to adverse weather conditions [12].

One of the augmentation methods used in intelligent transportation case studies includes horizontal flipping, rotation, and Gaussian noise. In this research, data augmentation methods are applied to expand the dataset and address class imbalance issues [4]. Furthermore, in vehicle speed detection research, techniques such as Mosaic, random perspective, Mixup, HSV adjustments, vertical flip (Flipud), and horizontal flip (Fliplr) are employed for detecting three types of vehicles: cars, buses, and trucks [12]. Other studies employ brightness and contrast adjustment techniques for recognizing black smoke from vehicles in road traffic monitoring videos [13]. Research related to road areas involves dividing the road into sections and augmenting them using methods such as horizontal flipping, color jitter, AutoAug, and Mixup [14]. Additionally, in flood depth detection studies, augmentation techniques such as Mosaic, label smoothing, and cosine annealing are applied independently, with Mosaic proving most effective for this case [15]. In a study aimed at increasing training set diversity and preventing overfitting, multi-label data augmentation techniques such as random cropping, flipping, rotation, and color jittering were applied to improve the accuracy and efficiency of image annotation processes [16].

YOLO is a SoTA, real-time object detection algorithm that has gained significant attention in the computer vision community. YOLO is a single-stage detector, meaning it detects all objects in an image with a single forward pass through the convolutional neural network (CNN) [17]. Its real-time detection capability makes it ideal for applications requiring high-speed processing. In one study, YOLOv4 proved effective for real-time traffic sign detection in autonomous vehicles [18]. Additionally, YOLO has been successfully used to detect and classify human hand actions in egocentric videos [19]. Another study showed that the YOLO model significantly improved Ball Tracking, Goal Alignment, and Robot Avoidance tasks in humanoid robot soccer [20]. An enhanced version of YOLOv4 with SemiDSConv and the FIoU loss function demonstrated significant performance in underwater target detection [21]. In medical imaging, YOLO achieved faster computation times compared to SSD networks for ovarian tumor classification in ultrasound images [22]. A systematic literature review (SLR) of studies on traffic sign detection and recognition using YOLO, published between 2016 and 2022, highlighted advancements in detection, classification, and processing speed for traffic sign recognition systems across various YOLO versions. The results show that YOLO is capable of detecting and recognizing traffic signs with high accuracy and fast processing times [23]. Furthermore, research using the latest YOLO model, YOLOv9, has shown that it can handle multiple adverse conditions simultaneously in object detection tasks [24].

Many studies have demonstrated that data augmentation can significantly improve YOLO's performance in object detection and classification. For instance, research using YOLOv5 employed traditional data augmentation methods such as noise addition, cropping, flipping, rotation, brightness, and contrast adjustments to enrich the dataset [25]. These techniques have been applied to YOLOv5, as well as YOLOv6, YOLOv7, and YOLOv8, to enhance vehicle detection and classification in surveillance footage [26]. Specifically, in YOLOv7, researchers expanded the DAWN dataset by applying augmentation techniques such as blur, saturation, brightness, darkness, noise, exposure, hue, and grayscale [27]. Additionally, for pothole detection using YOLOv8, techniques such as rotation, scaling, and flipping were applied to create more image variations, thereby improving detection performance [28]. In vehicle detection case studies, augmentation methods such as rotation, width and height shifts, brightness adjustments, zooming, and horizontal flipping were used to enhance the accuracy of YOLOv2 and YOLOv4 [29]. During traffic vehicle dataset augmentation, multi-label challenges arise, and training CNN models on small datasets is also challenging. Therefore, model enhancement through multi-augmentation techniques is necessary to further improve performance [16].

The contribution of this research is the development of the YOLO model using multi-augmentation techniques to enhance real-time detection accuracy of vehicle types and counts from CCTV footage. The model is expected to improve accuracy under diverse and complex urban traffic conditions. The development primarily focuses on addressing challenges related to traffic density during peak hours, small object detection, and varying lighting conditions.

2. METHOD

2.1. Methodology

We present a workflow for real-time multi-label vehicle detection in traffic using the YOLO model enhanced with multi-augmentation techniques. The diagram below illustrates each stage, from raw data acquisition to model evaluation, with an emphasis on the use of multi-augmentation techniques to improve the YOLO model's performance. The workflow diagram is shown in Figure 1.

Figure 1 illustrates the workflow for vehicle detection in traffic using the YOLO model enhanced with multi-augmentation techniques. The process begins with video capture from CCTV cameras recording vehicle traffic for analysis. The video is then processed in the data preprocessing stage to prepare it for annotation. Images are acquired by splitting the video recordings into frames at 5-minute intervals during data preprocessing. Each vehicle in the images is then annotated by type, including detailed information such as location and bounding box coordinates. This ensures the data is ready for further analysis or model training. After annotation, the data is split into 80% for training and 20% for validation, ensuring the model is properly trained and evaluated on unseen data [30]-[32]. Next, various augmentation techniques are applied to the dataset to increase data variability and improve model performance. The applied augmentation techniques include blur, brightness adjustment, contrast adjustment, color jitter, cropping, flipping, noise injection, rotation, scaling, and zoom in. The implemented augmentation techniques are illustrated in Figure 2.

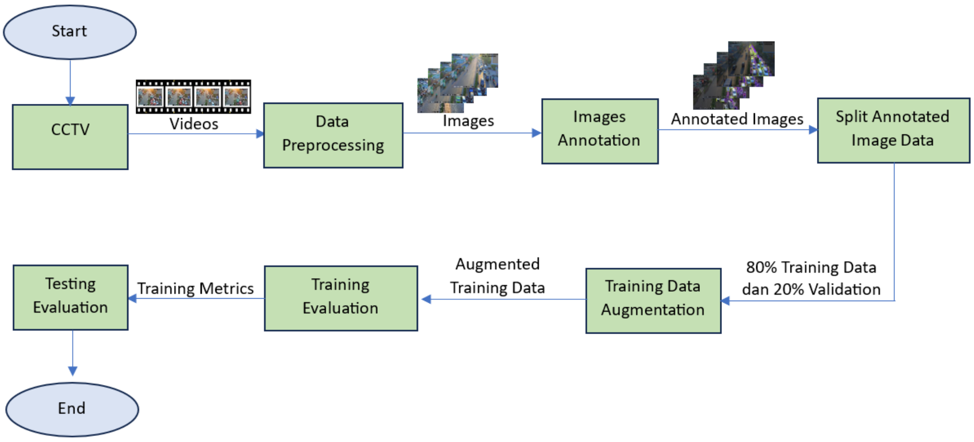

Figure 1. Model workflow

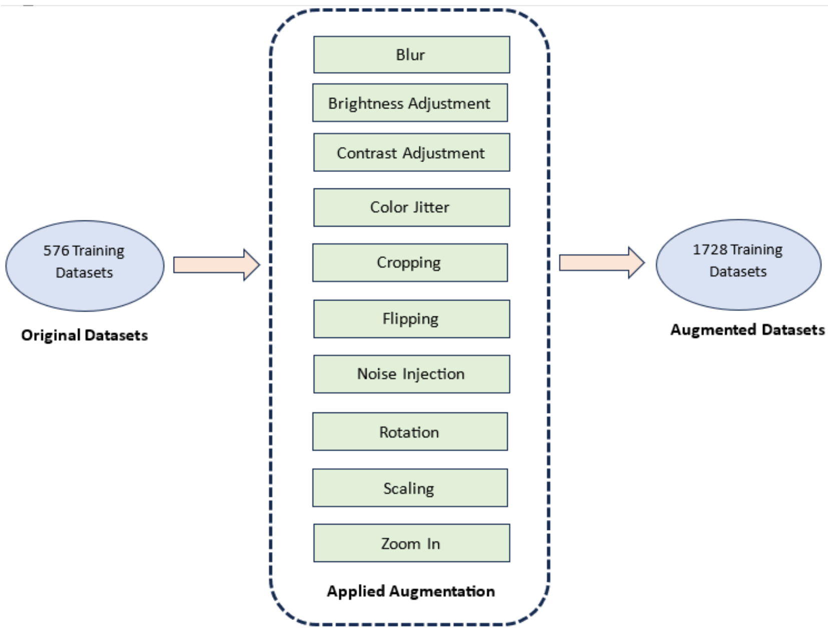

Figure 2. Implemented augmentation techniques

Figure 2 shows that each augmentation technique triples the dataset size, expanding 576 frames into 1,728 augmented frames to simulate real-world conditions like lighting changes, color variations, and visual disturbances, enhancing the model's accuracy and robustness. The next stage involves testing YOLOv8n and YOLOv9t with these augmentations to evaluate vehicle detection in traffic videos. The process includes training evaluation, testing on varied road conditions, and final assessment using testing metrics to measure accuracy and identify areas for improvement. This workflow, from data acquisition to model evaluation, emphasizes augmentation's role in enhancing YOLO's performance, as illustrated in Figure 3.

Figure 3 shows the design for testing the flipping augmentation technique with YOLO. The process begins with traffic image training samples undergoing horizontal flip ($x\_center = 1.0 - x\_center$) and vertical flip ($y\_center = 1.0 - y\_center$), producing transformed images. The augmented images are fed into the YOLO feature extractor to capture essential features. The YOLO output is processed by three main heads: the classification head for object classification, the detection head for object detection and marking, and the appearance embedding head for object embedding. This technique aims to enhance YOLO model performance by recognizing objects from various orientations.

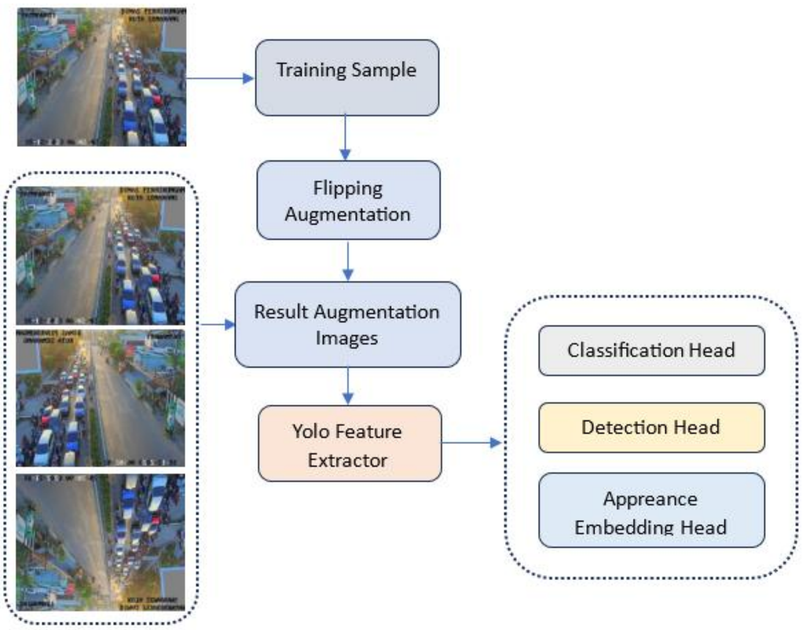

Figure 3. Flipping augmentation technique model design with YOLO

2.2. Research instrument

The dataset was collected from surveillance cameras at the Fatmawati traffic lights in Semarang City between December 19, 2023, and February 15, 2024, from 06:00 to 07:00 WIB. Videos were recorded in H.264 format at 1280×960 resolution and 25 FPS, focusing on vehicles in the right lane. Frames were extracted every 5 minutes using a calculated interval of 7500 frames, yielding 720 images categorized into motorcycles (31,481), cars (12,402), trucks (1,184), and buses (280). For training, 576 samples were used, while 144 samples were allocated for validation. Annotation was conducted using Roboflow [33]. Model analysis was performed on Google Colab using Python 3.10.12 and PyTorch 2.3.0 with CUDA 12.1 support.

2.3. Data analysis techniques

The data analysis began with collecting frames from CCTV traffic videos, which were uploaded to Roboflow for dataset management and annotation [34]. Vehicle classes motorcycles, cars, buses, and trucks were created to categorize and train the model effectively. Each image was annotated by marking vehicles according to their class, ensuring accurate labeling for training. The dataset was then split into 80% training (576 images) and 20% validation (144 images). Augmentation was performed using Python, tripling the dataset by applying two augmentation techniques per image while keeping one as the original. The augmentation technique values are detailed in Table 1.

Each augmentation technique from Table 1 uses specific values to generate image variations. The blur technique uses kernel sizes of 1 for image 2 and 2 for image 3. For brightness adjustment, the brightness factors are 0.8 for image 2 and 1.2 for image 3. Contrast adjustment uses alpha values of 1.5 for image 2 and 2.0 for image 3. In color jitter adjustment, a combination of brightness, contrast, and saturation factors is randomized within a range of 0.6 to 1.4, while the hue factor is randomized between -0.1 and 0.1. Image cropping is done with two types of crops: top crop from (0,0) to (width, height * 0.5) on image 2, and bottom crop from (0, height * 0.5) to (width, height) on image 3. Flipping includes both horizontal and vertical flips, with the horizontal flip center at x_center = 1.0 for image 2 and -x_center for vertical flip on image 3. Noise injection adds Gaussian noise with values randomized between 0 and 0.1 for both image 2 and image 3. Rotation is applied at 90 degrees for image 2 and 270 degrees for image 3. Scaling applies a factor between 0.8 and 1.2 for both image 2 and image 3, while zoom-in uses a factor of 1.2 for image 2 and 1.5 for image 3. Each value is carefully designed to create significant image variations, improving model performance under various vehicle recognition conditions. An example of the output from each data augmentation technique using the second value can be seen in Figure 4.

Table 1. Augmentation values

No	Aug	Value	Augmentation factor (image)			References
			1	2	3	
1	Blur	Kernel size	-	1	2	[35]
2	Brightness Adjustment	Brightness factor	-	0.8	1.2	[36]
3	Contrast Adjustment	Alpha	-	1.5	2.0	[37]
4	Color jitter	(Brightnes, contrast, saturation) and hue	-	Rand (0.6,1.4) and Rand (-0.1,0.1)	Rand (0.6,1.4) and Rand (-0.1,0.1)	[38]
5	Cropping	Crop height fraction	-	Top crop (0,0) to (width, height * 0.5)	Bottom Crop (0, height * 0.5) to (width, height)	[39]
6	Flipping	Horizontal and vertical flip	-	Horizontal flip x_center = 1.0 - x_center	Vertical flip y_center = 1.0 - y_center	[40]
7	Noise injection	Gaussian noise	-	Rand (0,0.1)	Rand (0,0.1)	[39]
8	Rotation	Rotation	-	90'	270'	[41]
9	Scaling	Scale image	-	Rand (0.8, 1.2)	Rand (0.8, 1.2)	[42]
10	Zoom in	Zoom in	-	1.2	1.5	[43]

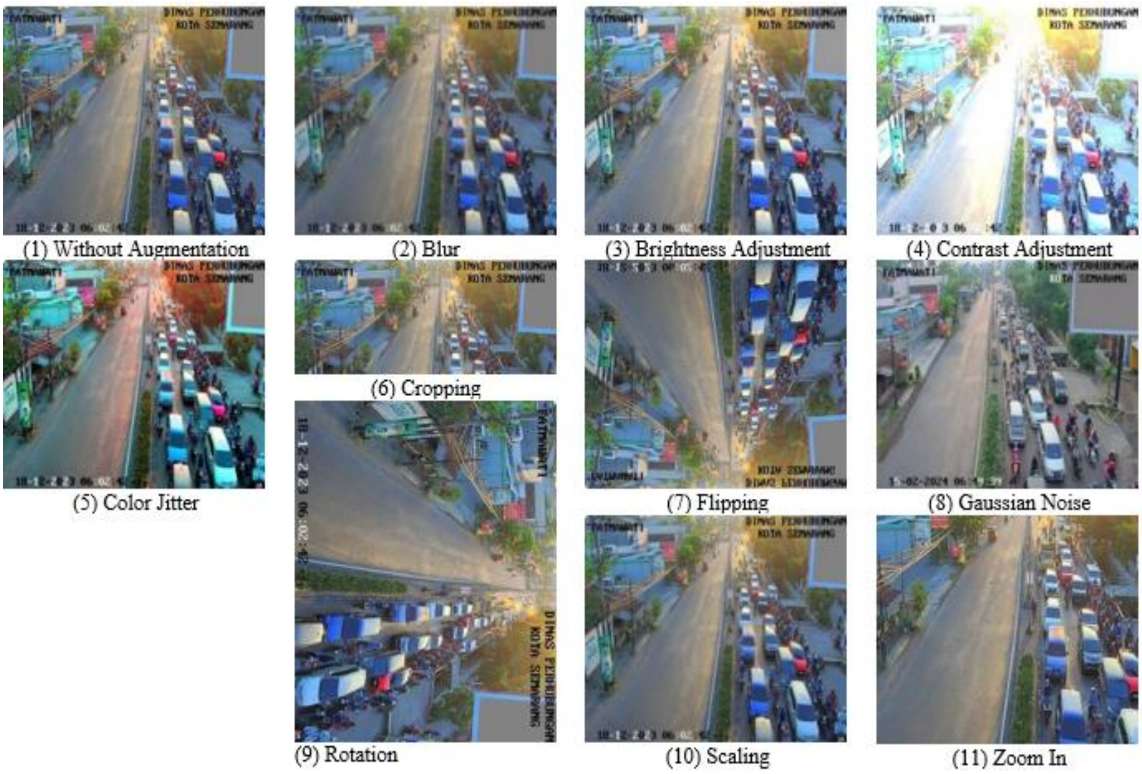

Figure 4. Example of data augmentation outputs

2.4. Multi-augmentation

Multi-augmentation in image dataset processing enhances model adaptability to real-world conditions by increasing dataset variability without additional data collection. Applying multiple techniques simultaneously improves object recognition under different lighting, sizes, orientations, and image qualities, making the model more robust. The augmentation techniques used in this study address key object recognition challenges and are listed in Table 2.

Table 2 shows that the first combination focuses on small objects and lighting, while the second deals with variations in angle and lighting. The third combination targets size and color variations, while the fourth focuses on detail enhancement. The final combination incorporates all augmentation techniques to ensure overall robustness. Mathematically, if X is the original dataset, then after applying the augmentation combinations, the new dataset Y can be represented as $\mathrm{Y} = \sum_{\mathrm{i}\text{-}1}^{\mathrm{n}} \mathrm{A_i(X)}$, where A_i is the i-th augmentation technique applied to dataset X. This results in a larger and more varied dataset, enhancing the model's ability to generalize.

Table 2. Multi augmentation technique

No	Augmentation technique combination	Focus	Reference
1	Scaling + Cropping + Brightness adjustment + Noise injection + Blur	Small objects and lighting	[44], [45], [46]
2	Rotation + Flipping + Brightness adjustment + Contrast adjustment + Color jitter	Angle and lighting variation	[45], [47]
3	Scaling + Zoom in + Brightness adjustment + Color jitter + Noise injection	Size and color variation	[45], [48]
4	Cropping + Zoom in + Contrast adjustment + Noise injection + Blur	Detail enhancement	[45], [48], [49], [40]
5	Scaling + Rotation + Brightness adjustment + Contrast adjustment + Noise injection	Complete combination for robustness	[40], [50], [51]

2.5. YOLO models

YOLO is widely used in transportation systems for its high detection accuracy and real-time performance [52]. YOLOv8 introduces five scaled versions-YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x-catering to different applications, with YOLOv8n optimized for speed, achieving 80.4 ms on a CPU using ONNX and 0.99 ms on an A100 GPU using TensorRT. Trained on the COCO dataset, which includes 80 object classes such as cars, motorcycles, buses, and trucks, YOLOv8n is ideal for real-time traffic monitoring on low-power devices [53], [54]. Similarly, YOLOv9 offers five variants-YOLOv9t, YOLOv9s, YOLOv9m, YOLOv9c, and YOLOv9e-designed for various needs, with YOLOv9t balancing speed and accuracy, featuring 2M parameters, 7.7B FLOPs, and a mAPval 50-95 score of 38.3% [55]. This study focuses on YOLOv8n and YOLOv9t, selecting lightweight models optimized for real-time use while maintaining sufficient accuracy, with adjusted default YOLO parameters detailed in Table 3.

The default batch size in YOLO was reduced from 64 to 8 to minimize overfitting, particularly on unclear or overly bright images, while the patience size was increased from 5 to 10 to prevent premature stopping. The dropout value was adjusted from 0.0 to 0.2 to enhance robustness, and the image size was set to 640×640 pixels to balance small object detection and inference speed [59]-[62]. The analysis process follows structured stages, including preprocessing, labeling, annotation, data augmentation, and dataset splitting for training and validation. Model training uses 80% of the dataset over 64 epochs with parameters listed in Table 2, while validation with the remaining 20% assesses precision, recall, and mAP. Finally, testing with 10 selected frames evaluates the best model's reliability, ensuring robust real-world performance.

Table 3. Adjusted parameters

Parameter	Value	References
Batch	8	[56]
Patience	10	[57]
Dropout	0.2	[58]
Image size	640	[59]

2.6. Performance evaluation measures

Evaluating a classification model requires a comprehensive analysis using several key metrics. Precision, which measures the accuracy of positive predictions among all predicted positives, and recall, which evaluates the proportion of correctly identified positives out of all actual positives, are critical metrics. Additionally, the mean average precision (mAP) metric is used to evaluate detected bounding boxes by comparing them to ground-truth boxes and assigning a corresponding score [59]. The equations for precision, recall, and mAP can be seen in (1)-(3).

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

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

$$\mathrm{mAP} = \frac{1}{n}\sum_{j=1}^{n} AP(j) \quad (3)$$

Where FP represents (false positive), TN indicates (true negative), TP means (true positive), and FN indicates (false negative). AP is (average precision), AP_j indicates the average precision for category i, and N is the number of classes. This study uses mean precision mAP50 and mAP50-95 as the primary metrics for measuring detection accuracy. Floating-point operations (GFLOPs), the number of parameters (Params), and detection frames per second (FPS) are used to assess efficiency and real-time performance. Additionally, the model's weight size (Size) is considered to evaluate its suitability for edge device implementation [63]. The equations for mAP50 and mAP50-95 are shown in (4) and (5).

$$\mathrm{AP50} = \frac{1}{n}\sum_{i=1}^{n} P_i^{IoU=0.5}\,(R_i^{IoU=0.5}) \quad (4)$$

$$\mathrm{AP50}-95 = \frac{1}{10}(AP50 + AP55 + \cdots + AP95) \quad (5)$$

Where P represents precision, the ratio of correctly predicted positive samples to all predicted positive samples. R represents recall, the ratio of correctly predicted positive samples to all actual positive samples. AP50 refers to the mean AP across categories when the intersection over union (IoU) threshold is set at 50%. AP50-95 reflects the average AP as the IoU threshold increases from 50% to 95% in 5% increments.

3. RESULTS AND DISCUSSION

3.1. Training validation

This study trained and validated YOLOv8n and YOLOv9t using 32 scenarios. The training data included 1 non-augmented dataset (576 samples), 10 single-augmentation datasets (1,728 samples each), and 5 multiple-augmentation datasets (6,336 samples each). Validation used 144 samples (20% of the initial dataset) and measured precision, recall, mAP50, and mAP50-95, with mAP50-95 as the primary metric for performance evaluation. Table 4 presents the performance of each model with its respective augmentations.

Table 4 shows that the YOLO model without augmentation achieved the highest mAP50-95 (0.390) in the YOLOv8n test, while the best single augmentation performance was with the blur technique (0.465), and multiple augmentations combining scaling, zoom in, brightness adjustment, color jitter, and noise injection reached the highest mAP50-95 (0.526). These results confirm that augmentation techniques enhance model performance, with multiple augmentations providing the best detection accuracy. In urban traffic during rush hours, where vehicles appear small and are affected by sunlight reflections, YOLOv8n outperformed YOLOv9t, demonstrating higher accuracy in complex conditions. The evaluation metrics for each class using multiple augmentations are detailed in Table 5.

Table 5 presents the evaluation metrics for different object classes, showing that the 'bus' class achieves the highest mAP50-95 (0.626) and precision (0.977), indicating strong detection performance with minimal false detections due to its large, easily recognizable features. The 'car' class has the highest recall (0.763), suggesting effective detection, likely because of its frequent appearance in the dataset, while also leading in mAP50 (0.85), reinforcing its reliability at an IoU threshold of 0.50. Conversely, the 'motorcycle' class has the lowest mAP50-95 (0.359) and relatively low precision (0.831), indicating higher detection errors, likely due to its smaller size and reflections. Figure 5 highlights common misclassifications, with motorcycles often mistaken for cars (1,303 times) and cars misidentified as background (394 times). Despite strong overall performance, the model needs improvements in handling small objects and challenging lighting conditions.

Table 4. Comparison of performance metrics of the tested models

No	Augmentation	Models	Best epoch	Precision	Recall	mAP50	mAP50-95
1	Without augmentation	YOLOv8n	64	0.819	0.550	0.673	0.390
2	Without augmentation	YOLOv9t	64	0.825	0.556	0.669	0.378
3	Blur	YOLOv8n	64	0.825	0.674	0.767	0.465
4	Blur	YOLOv9t	64	0.830	0.642	0.746	0.443
5	Brightness adjustment	YOLOv8n	64	0.774	0.701	0.757	0.462
6	Brightness adjustment	YOLOv9t	62	0.880	0.619	0.749	0.449
7	Contrast adjustment	YOLOv8n	49	0.594	0.071	0.325	0.203
8	Contrast adjustment	YOLOv9t	10	0.481	0.097	0.280	0.167
9	Color jitter	YOLOv8n	61	0.861	0.648	0.755	0.461
10	Color jitter	YOLOv9t	46	0.878	0.565	0.713	0.411
11	Cropping	YOLOv8n	60	0.784	0.540	0.646	0.356
12	Cropping	YOLOv9t	64	0.740	0.559	0.635	0.349
13	Flipping	YOLOv8n	64	0.853	0.603	0.732	0.436
14	Flipping	YOLOv9t	62	0.785	0.633	0.719	0.411
15	Noise injection	YOLOv8n	51	0.870	0.619	0.739	0.433
16	Noise injection	YOLOv9t	64	0.864	0.625	0.715	0.422
17	Rotation	YOLOv8n	62	0.779	0.559	0.663	0.371
18	Rotation	YOLOv9t	63	0.766	0.545	0.653	0.366
19	Scaling	YOLOv8n	61	0.836	0.669	0.753	0.464
20	Scaling	YOLOv9t	64	0.812	0.663	0.741	0.459
21	Zoom-In	YOLOv8n	64	0.786	0.640	0.722	0.426
22	Zoom-In	YOLOv9t	64	0.792	0.625	0.716	0.421
23	Scaling + Cropping + Brightness adjustment + Noise injection + Blur	YOLOv8n	63	0.893	0.677	0.777	0.511
24	Scaling + Cropping + Brightness adjustment + Noise injection + Blur	YOLOv9t	64	0.888	0.681	0.784	0.503
25	Rotation + Flipping + Brightness adjustment + Contrast adjustment + Color jitter	YOLOv8n	64	0.867	0.652	0.762	0.491
26	Rotation + Flipping + Brightness adjustment + Contrast adjustment + Color jitter	YOLOv9t	63	0.890	0.651	0.768	0.480
27	Scaling + Zoom in + Brightness adjustment + Color jitter + Noise injection	YOLOv8n	61	0.872	0.715	0.792	0.526
28	Scaling + Zoom in + Brightness adjustment + Color jitter + Noise injection	YOLOv9t	64	0.838	0.695	0.776	0.506
29	Cropping + Zoom in + Contrast adjustment + Noise injection + Blur	YOLOv8n	63	0.872	0.668	0.774	0.498
30	Cropping + Zoom in + Contrast adjustment + Noise injection + Blur	YOLOv9t	62	0.835	0.645	0.747	0.469
31	Scaling + Rotation + Brightness adjustment + Contrast adjustment + Noise injection	YOLOv8n	64	0.882	0.670	0.784	0.511
32	Scaling + Rotation + Brightness adjustment + Contrast adjustment + Noise injection	YOLOv9t	63	0.897	0.643	0.777	0.493

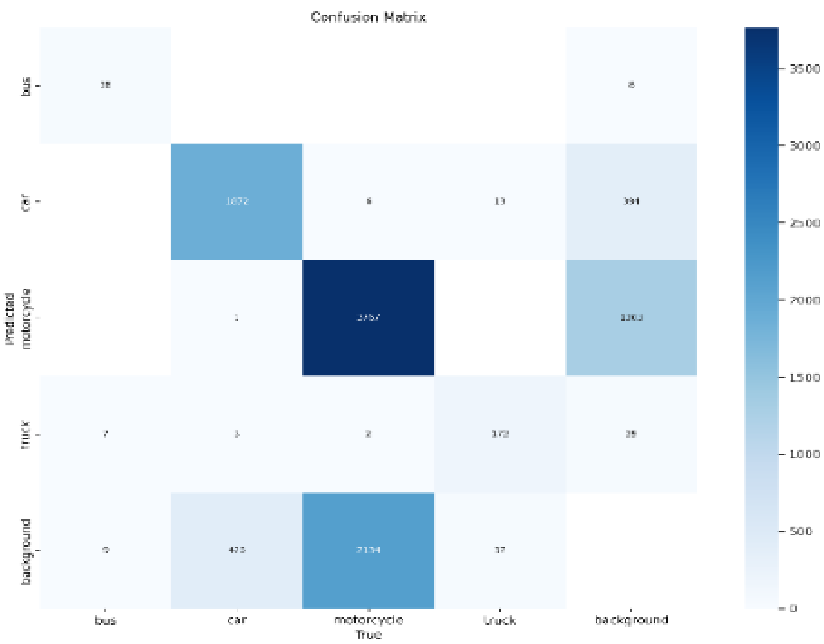

Figure 5. Confusion matrix

3.2. Model testing

The aim of this testing is to evaluate the developed model to determine whether it works efficiently and accurately in counting and classifying types of vehicles in traffic with multi-labels. In this experiment, we use the YOLO framework and multiple augmentations, which achieved the highest values in the training data. The 10 testing scenarios are listed in Table 6.

Table 6 evaluates the object detection model in various traffic scenarios, including congested traffic with light reflection, optimal lighting, smooth traffic with light reflection, and rain, to assess its adaptability to real-world challenges like detection errors from light reflections or reduced visibility due to rain. By simulating these conditions, researchers can evaluate the model’s resilience and functionality under different weather and lighting conditions. Testing across all vehicle classes ensures accurate detection, even for small objects, helping to identify weaknesses and improve the model’s reliability and accuracy in real-world applications. The test results are summarized in Table 7.

The mAP calculations for each testing scenario are illustrated in the graph. The graph clearly demonstrates how the model’s performance varies across different traffic and lighting conditions. The mAP calculations for each testing scenario are presented in Figure 6.

Table 5. Evaluation metrics class from Yolov8n using multi-augmentation

Class	Precision	Recall	mAP50	mAP50-59
All	0.872	0.715	0.792	0.526
Bus	0.977	0.788	0.850	0.626
Car	0.866	0.763	0.834	0.538
Motorcycle	0.831	0.559	0.672	0.359
Truck	0.814	0.749	0.810	0.582

Table 6. Model testing scenarios

No	Condition
1	Congested traffic with light reflection
2	Congested traffic with properly exposed
3	Smooth traffic with light reflection
4	Smooth traffic with properly exposed
5	Congested traffic in rain
6	Smooth traffic in rain
7	Traffic with all vehicle classes, Congested with light reflection
8	Traffic with all vehicle classes, Congested with properly exposed
9	Traffic with all vehicle classes, Smooth with light reflection
10	Traffic with all vehicle classes, Smooth with properly exposed

Table 7. Model testing results

No	Condition	Test results
1	Congested traffic with light reflection	41 cars, 39 motorcycles, 4 trucks, 0 buses
2	Congested traffic with properly exposed	36 cars, 107 motorcycles, 1 truck, 0 buses
3	Smooth traffic with light reflection	6 cars, 26 motorcycles, 0 trucks, 0 buses
4	Smooth traffic with properly exposed	9 cars, 20 motorcycles, 0 trucks, 0 buses
5	Congested traffic in rain	43 cars, 86 motorcycles, 7 trucks, 0 buses
6	Smooth traffic in rain	5 cars, 9 motorcycles, 0 trucks, 0 buses
7	Traffic with all vehicle classes, Congested with light reflection	15 cars, 37 motorcycles, 9 trucks, 0 buses
8	Traffic with all vehicle classes, Congested with properly exposed	15 cars, 57 motorcycles, 3 trucks, 0 buses
9	Traffic with all vehicle classes, Smooth with light reflection	10 cars, 16 motorcycles, 2 trucks, 1 bus
10	Traffic with all vehicle classes, Smooth with properly exposed	6 cars, 21 motorcycles, 4 trucks, 1 bus

The test results indicate that the model’s detection performance varies depending on traffic conditions. In dense traffic with light reflections, the mAP was 0.464, indicating that light reflections make object detection more challenging. Under optimal lighting conditions, the mAP increased to 0.623, suggesting that better lighting enhances detection accuracy. In smooth traffic with light reflections, the mAP increased to 0.825, and under smooth traffic with optimal lighting, the mAP reached 0.932, demonstrating the best performance.

During rain and dense traffic, the mAP dropped to 0.724, indicating that rain diminishes detection accuracy. In smooth traffic during rain, the mAP rose to 0.817. In dense conditions with light reflections and various types of vehicles, the mAP was 0.887, suggesting that the model still performed well in these conditions. Under optimal lighting, the mAP slightly decreased to 0.809, indicating that vehicle complexity affects performance. In smooth traffic with various vehicle types and light reflections, the mAP reached

0.818, and under optimal lighting, it increased to 0.89. Overall, the best performance was achieved under optimal lighting and smooth traffic, while rain or light reflections reduced accuracy.

Figure 6. mAP calculation for various traffic and lighting conditions

3.3. Challenges

The primary challenges in object detection within this dataset are the variations in lighting conditions and traffic density. The performance of object detection varies significantly depending on whether the images are affected by light reflections or are properly exposed. Object detection becomes more challenging under suboptimal lighting or disruptive light reflections, as well as in congested traffic conditions compared to smoother traffic. This suggests that poor lighting and high traffic density hinder the model's ability to accurately detect objects.

Moreover, the presence of all object classes in a single image further complicates the detection challenge. Images containing all types of objects (cars, motorcycles, trucks, and buses) show that the model struggles to identify and differentiate objects when there are multiple overlapping classes. Specifically, classes such as buses and trucks exhibit lower detection performance compared to cars and motorcycles, suggesting that the model may be less effective in detecting less common objects or those with less distinctive visual features. Improvement efforts should concentrate on adjusting for lighting variations, accommodating different road conditions, and enhancing the model's ability to detect more challenging classes. Despite these challenges, the proposed model exhibits reasonably good performance.

4. CONCLUSION

This study successfully demonstrates that using YOLO and multi-augmentation techniques for real-time vehicle detection in urban traffic can enhance accuracy and efficiency in identifying and classifying various types of vehicles. The utilization of multi-label images enables the detection of multiple vehicle types within a single frame, offering a more comprehensive view of traffic conditions. Implementing multi-augmentation increases the diversity of the training data, making the model more robust to varying lighting conditions, angles, and object obstructions.

The test results indicate that YOLOv8n with multi-augmentation (scaling, zoom in, brightness adjustment, color jitter, and noise injection) delivers the best performance, achieving a precision of 0.872, recall of 0.715, mAP50 of 0.792, and mAP50-95 of 0.526, surpassing other augmentation techniques. Traffic analysis shows that optimal lighting and smooth traffic yield the best detection performance, with a maximum mAP of 0.932, whereas poor lighting (glare) and high traffic density reduce detection accuracy, resulting in a minimum mAP of 0.464. It was also shown that multi-augmentation significantly outperforms single augmentation and no augmentation. In this study, YOLOv8n exhibited superior performance over YOLOv9t, particularly under complex traffic conditions and challenging lighting environments.

ACKNOWLEDGEMENTS

We would like to express our gratitude to the Semarang City Transportation Department for their assistance and support in providing the valuable dataset used in this research.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Imam Ahmad Ashari	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Wahyul Amien Syafei		✓				✓		✓	✓	✓	✓	✓		
Adi Wibowo	✓		✓	✓		✓			✓		✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] R. R. Almassar and A. S. Girsang, "Detection of traffic congestion based on Twitter using convolutional neural network model," *International Journal of Artificial Intelligence (IAES)*, vol. 11, no. 4, pp. 1448–1459, 2022, doi: 10.11591/ijai.v11.i4.pp1448-1459.

[2] Y. Umniyatun, M. I. Nurmansyah, Y. Farradika, T. B. Purnama, and D. N. Hidayat, "Motorcycle risky behaviours and road accidents among adolescents in Jakarta metropolitan area, Indonesia," *International Journal of Injury Control and Safety Promotion*, vol. 28, no. 3, pp. 339–346, 2021, doi: 10.1080/17457300.2021.1928229.

[3] J. Akbardin, D. Mohamad, and A. Z. Rahmadani, "Modeling volume-speed-density relationship on the Cibubur transyogi road based on traffic behavior during peak hours activity," *International Journal of Science and Society*, vol. 5, no. 4, pp. 842–850, 2023, doi: 10.54783/ijsoc.v5i4.855.

[4] U. Mittal, P. Chawla, and R. Tiwari, "EnsembleNet: a hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models," *Neural Computing and Applications*, vol. 35, no. 6, pp. 4755–4774, 2023, doi: 10.1007/s00521-022-07940-9.

[5] P. Y. Chen, J. W. Hsieh, M. Gochoo, and Y. S. Chen, "Mixed stage partial network and background data augmentation for surveillance object detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23533–23547, 2022, doi: 10.1109/TITS.2022.3206709.

[6] E. Goceri, "Medical image data augmentation: techniques, comparisons and interpretations," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12561–12605, 2023, doi: 10.1007/s10462-023-10453-z.

[7] R. Brennan, "Image data augmentation approaches: a comprehensive survey and future directions," *IEEE Access*, vol. 12, pp. 187536–187571, 2024, doi: 10.1109/ACCESS.2024.3470122.

[8] A. Ayub and H. W. Kim, "GAN-based data augmentation with vehicle color changes to train a vehicle detection CNN," *Electronics*, vol. 13, no. 7, pp. 1–14, 2024, doi: 10.3390/electronics13071231.

[9] B. Trabucco, K. Doherty, and M. Gurinas, "Effective data augmentation with diffusion models," *arXiv preprint arXiv:2309.01234*, pp. 1–22, 2023.

[10] I. S. Mohamed and L. K. Chuan, "PAE: portable appearance extension for multiple object detection and tracking in traffic scenes," *IEEE Access*, vol. 10, pp. 37257–37268, 2022, doi: 10.1109/ACCESS.2022.3160424.

[11] R. Mohammad, F. Saeed, A. A. Almazroi, F. S. Alsubaei, and A. A. Almazroi, "Enhancing intrusion detection systems using a deep learning and data augmentation approach," *Systems*, vol. 12, no. 3, pp. 1–18, 2024, doi: 10.3390/systems12030079.

[12] Z. Luo *et al*., "Enhanced YOLOv5s + DeepSORT method for highway vehicle speed detection and multi-sensor verification," *Frontiers in Physics*, vol. 12, no. February, pp. 1–16, 2024, doi: 10.3389/fphy.2024.1371320.

[13] K. Chen, H. Wang, and Y. Zhai, "A lightweight model for real-time detection of vehicle black smoke," *Sensors*, vol. 23, no. 23, p. 9492, 2023, doi: 10.3390/s23239492.

[14] R. Zhang, Y. Tian, Z. Xu, and D. Liu, "Design of anchor boxes and data augmentation for transformer-based vehicle localization," *Journal of Visual Communication and Image Representation*, vol. 90, no. August 2022, p. 103711, 2023, doi: 10.1016/j.jvcir.2022.103711.

[15] P. Zhong, Y. Liu, H. Zheng, and J. Zhao, "Detection of urban flood inundation from traffic images using deep learning methods," *Water Resources Management*, vol. 38, no. 1, pp. 287–301, 2024, doi: 10.1007/s11269-023-03669-9.

[16] X. Ke, J. Zou, and Y. Niu, "End-to-end automatic image annotation based on deep CNN and multi-label data augmentation," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2093–2106, 2019, doi: 10.1109/TMM.2019.2895511.

[17] M. G. Ragab *et al*., "A comprehensive systematic review of YOLO for medical object detection (2018 to 2023)," *IEEE Access*, vol. 12, pp. 57815–57836, 2024, doi: 10.1109/ACCESS.2024.3386826.

[18] F. N. Ortataş and E. Çetin, "Solution of real-time traffic signs detection problem for autonomous vehicles by using YOLOv4 and Haar cascade algorithms," *International Journal of Automotive Science and Technology*, vol. 7, no. 2, pp. 125–140, 2023, doi: 10.30939/ijastech..1231646.

[19] H. C. Nguyen, T. H. Nguyen, R. Scherer, and V. H. Le, "YOLO series for human hand action detection and classification from egocentric videos," *Sensors*, vol. 23, no. 6, pp. 1–24, 2023, doi: 10.3390/s23063255.
[20] H. Jati, N. A. Ilyasa, and D. D. Dominic, "Enhancing humanoid robot soccer ball tracking, goal alignment, and robot avoidance using YOLO-NAS," *Journal of Robotics and Control (JRC)*, vol. 5, no. 3, pp. 829–838, 2024, doi: 10.18196/jrc.v5i3.21839.
[21] C. Zhang, G. Zhang, H. Li, H. Liu, J. Tan, and X. Xue, "Underwater target detection algorithm based on improved YOLOv4 with SemiDSConv and FIoU loss function," *Frontiers in Marine Science*, vol. 10, no. March, pp. 1–13, 2023, doi: 10.3389/fmars.2023.1153416.
[22] V. Le, T. Pham, and C. Author, "Ovarian tumors detection and classification on ultrasound images using one-stage convolutional neural networks," *Journal of Robotics and Control (JRC)*, vol. 5, no. 2, pp. 561–581, 2024, doi: 10.18196/jrc.v5i2.20589.
[23] M. Flores-Calero *et al*., "Traffic sign detection and recognition using YOLO object detection algorithm: a systematic review," *Mathematics*, vol. 12, no. 2, pp. 1–31, 2024, doi: 10.3390/math12020297.
[24] J. Li, Y. Feng, Y. Shao, and F. Liu, "IDP-YOLOv9: improvement of object detection model in severe weather scenarios from drone perspective," *Applied Sciences*, vol. 14, no. 12, pp. 1–21, 2024, doi: 10.3390/app14125277.
[25] T. Delleji *et al*., "An upgraded-YOLO with object augmentation: Mini-UAV detection under low-visibility conditions by improving deep neural networks," *Operations Research Forum*, vol. 3, no. 4, pp. 1–27, 2022, doi: 10.1007/s43069-022-00163-7.
[26] S. X. Tan, J. Y. Ong, K. O. M. Goh, and C. Tee, "Boosting vehicle classification with augmentation techniques across multiple YOLO versions," *International Journal of Informatics and Visualization*, vol. 8, no. 1, pp. 45–54, 2024, doi: 10.62527/joiv.8.1.2313.
[27] N. Aloufi, A. Alnori, V. Thayananthan, and A. Basuhail, "Object detection performance evaluation for autonomous vehicles in sandy weather environments," *Applied Sciences*, vol. 13, no. 18, pp. 1–22, 2023, doi: 10.3390/app131810249.
[28] M. Khan, M. A. Raza, G. Abbas, S. Othmen, A. Yousef, and T. A. Jumani, "Pothole detection for autonomous vehicles using deep learning: a robust and efficient solution," *Frontiers in Built Environment*, vol. 9, no. January, pp. 1–17, 2023, doi: 10.3389/fbuil.2023.1323792.
[29] R. Al-Batat, A. Angelopoulou, S. Premkumar, J. Hemanth, and E. Kapetanios, "An end-to-end automated license plate recognition system using YOLO-based vehicle and license plate detection with vehicle classification," *Sensors*, vol. 22, no. 23, pp. 1–17, 2022, doi: 10.3390/s22239477.
[30] G. Ayana *et al*., "Vision-transformer-based transfer learning for mammogram classification," *Diagnostics*, vol. 13, no. 2, 2023, doi: 10.3390/diagnostics13020178.
[31] I. Yousif, L. Burns, F. El Kalach, and R. Harik, "Leveraging computer vision towards high-efficiency autonomous industrial facilities," *Journal of Intelligent Manufacturing*, vol. 35, 2024, doi: 10.1007/s10845-024-02396-1.
[32] T. Wu and Y. Dong, "YOLO-SE: improved YOLOv8 for remote sensing object detection and recognition," *Applied Sciences*, vol. 13, no. 24, 2023, doi: 10.3390/app132412977.
[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," *Journal of Robotics and Control (JRC)*, vol. 3, no. 6, pp. 790–799, 2022, doi: 10.18196/jrc.v3i6.15948.
[34] J. Miguel, P. Mendonça, A. Quelhas, J. M. L. P. Caldeira, and V. N. G. J. Soares, "Using computer vision to collect information on cycling and hiking trails users," *Future Internet*, vol. 16, no. 3, pp. 1–32, 2024, doi: 10.3390/fi16030104.
[35] A. H. Khan, R. M. Umer, M. Dunnhofer, C. Micheloni, and N. Martinel, "LBKENet: lightweight blur kernel estimation network for blind image super-resolution," in *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*, Lecture Notes in Computer Science, vol. 14234, pp. 209–222, 2023, doi: 10.1007/978-3-031-43153-1_18.
[36] D. Raimondo *et al*., "Detection and classification of hysteroscopic images using deep learning," *Cancers*, vol. 16, no. 7, pp. 1–10, 2024, doi: 10.3390/cancers16071315.
[37] A. Tupper and C. Gagné, "Analyzing data augmentation for medical images: a case study in ultrasound images," *arXiv preprint arXiv:2403.09828*, pp. 1–12, 2024.
[38] M. Pei, N. Liu, B. Zhao, and H. Sun, "Self-supervised learning for industrial image anomaly detection by simulating anomalous samples," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, 2023, doi: 10.1007/s44196-023-00328-0.
[39] N. Eldeen, K. Mohamed, and L. Seyedali, "A comprehensive survey of recent trends in deep learning for digital images augmentation," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2351–2377, 2022, doi: 10.1007/s10462-021-10066-4.
[40] K. Alomar, H. I. Aysel, and X. Cai, "Data augmentation in classification and segmentation: a survey and new strategies," *Journal of Imaging*, vol. 9, no. 2, pp. 1–26, 2023, doi: 10.3390/jimaging9020046.
[41] P. Oza, P. Sharma, S. Patel, F. Adedoyin, and A. Bruno, "Image augmentation techniques for mammogram analysis," *Journal of Imaging*, vol. 8, no. 5, pp. 1–22, 2022, doi: 10.3390/jimaging8050141.
[42] B. Awaluddin and C. Chao, "Investigating effective geometric transformation for image augmentation to improve static hand gestures with a pre-trained convolutional neural network," *Mathematics*, vol. 11, no. 23, pp. 1–23, 2023, doi: 10.3390/math11234783.
[43] 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," *Journal of Scientific Research, Education, and Technology (JSRET)*, vol. 2, no. 2, pp. 453–465, 2023.
[44] M. Nagaraju, P. Chawla, and N. Kumar, "Performance improvement of deep learning models using image augmentation techniques," *Multimedia Tools and Applications*, vol. 81, no. 7, pp. 9177–9200, 2022, doi: 10.1007/s11042-021-11869-x.
[45] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.
[46] Z. Wang, S. Yang, M. Shi and K. Qin, "An image augmentation method based on limited samples for object tracking based on mobile platform," *Sensors*, vol. 22, no. 5, p. 1967, 2022, doi: 10.3390/s22051967.
[47] J. Ma, C. Hu, P. Zhou, F. Jin, X. Wang and H. Huang, "Review of image augmentation used in deep learning-based material microscopic image segmentation," *Applied Sciences*, vol. 13, no. 11, pp. 1–18, 2023, doi: 10.3390/app13116478.
[48] M. E. Akbiyik, "Data augmentation in training CNNs: injecting noise to images," *arXiv preprint*, 2023. [Online]. Available: http://arxiv.org/abs/2307.06855.
[49] R. Takahashi, T. Matsubara and K. Uehara, "Data augmentation using random image cropping and patching for deep CNNs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 2917–2931, 2020, doi: 10.1109/TCSVT.2019.2935128.
[50] M. Cossio, "Augmenting medical imaging: A comprehensive catalogue of 65 techniques for enhanced data analysis," *arXiv preprint*, 2023. [Online]. Available: http://arxiv.org/abs/2303.01178.
[51] A. Mumuni and F. Mumuni, "Data augmentation with automated machine learning: approaches and performance comparison with classical data augmentation methods," *arXiv preprint*, 2024. [Online]. Available: https://arxiv.org/abs/2403.08352v1.

[52] B. Wang, Y. Y. Li, W. Xu, H. Wang and L. Hu, "Vehicle–pedestrian detection method based on improved YOLOv8," *Electronics*, vol. 13, no. 11, 2024, doi: 10.3390/electronics13112149.
[53] Ultralytics, "Ultralytics YOLOv8," *GitHub*, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics. [Accessed: Aug. 15, 2024].
[54] M. Hussain, "YOLOv1 to v8: unveiling each variant—a comprehensive review of YOLO," *IEEE Access*, vol. 12, no. February, pp. 42816–42833, 2024, doi: 10.1109/ACCESS.2024.3378568.
[55] Ultralytics, "YOLOv9: a leap forward in object detection technology," *Ultralytics*, 2024. [Online]. Available: https://docs.ultralytics.com/models/yolov9/#performance-on-ms-coco-dataset.
[56] D. Li, F. Ahmed, N. Wu and A. I. Sethi, "Yolo-JD: a deep learning network for jute diseases and pests detection from images," *Plants*, vol. 11, no. 7, p. 937, 2022, doi: 10.3390/plants11070937.
[57] K. Kim, K. Kim and S. Jeong, "Application of YOLO v5 and v8 for recognition of safety risk factors at construction sites," *Sustainability*, vol. 15, no. 20, p. 15179, 2023, doi: 10.3390/su152015179.
[58] 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.
[59] H. T. Vo, K. C. Mui, N. N. Thien and P. P. Tien, "Automating tomato ripeness classification and counting with YOLOv9," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 4, pp. 1120–1128, 2024, doi: 10.14569/IJACSA.2024.01504113.
[60] F. Song and P. Li, "YOLOv5-MS: real-time multi-surveillance pedestrian target detection model for smart cities," *Biomimetics*, vol. 8, no. 6, 2023, doi: 10.3390/biomimetics8060480.
[61] A. Li, S. Sun, Z. Zhang, M. Feng, C. Wu and W. Li, "A multi-scale traffic object detection algorithm for road scenes based on improved YOLOv5," *Electronics*, vol. 12, no. 4, pp. 1–16, 2023, doi: 10.3390/electronics12040878.
[62] Y. Wang *et al.*, "VV-YOLO: a vehicle view object detection model based on improved YOLOv4," *Sensors*, vol. 23, no. 7, 2023, doi: 10.3390/s23073385.
[63] R. Liu, M. Huang, L. Wang, C. Bi and Y. Tao, "PDT-YOLO: a roadside object-detection algorithm for multiscale and occluded targets," *Sensors*, vol. 24, no. 7, p. 2302, 2024, doi: 10.3390/s24072302.

BIOGRAPHIES OF AUTHORS

Imam Ahmad Ashari received his bachelor's degree in Informatics from Universitas Negeri Semarang in 2016 and his master's degree in Information Systems from Universitas Diponegoro in 2019. He is currently a lecturer at the Informatics Study Program, Universitas Harapan Bangsa. His research interests include the internet of things (IoT), computer vision, and artificial intelligence. He can be contacted at email: imamahmadashari@uhb.ac.id.

Wahyul Amien Syafei received a bachelor's degree from Diponegoro University, Indonesia in 1995, a master's degree from Sepuluh Nopember Institute of Technology, Indonesia in 2002, and a doctoral degree from Kyushu Institute of Technology, Japan in 2010. Currently, he is an associate professor at the Master Program of Electrical Engineering, Faculty of Engineering, Diponegoro University. He is also a lecturer at the Master Program of Electrical Engineering, Postgraduate School, Diponegoro University. His research interests include transmission channels, digital communication systems, multimedia, multimedia networks, computer graphics, data communication, wireless and mobile networks, multimedia telecommunication, information systems, and signal transformation processing. He can be contacted at email: wasyafe@gmail.com.

Adi Wibowo He received the B.Sc. degree in mathematics from Universitas Diponegoro, Indonesia, in 2005, the M.Sc. degree in computer science from Universitas Indonesia in 2011, and the Ph.D. degree in engineering from Nagoya University, Japan, in 2016. He has been an Assistant Professor with the Department of Informatics, Universitas Diponegoro, Indonesia, since 2006. His research interests primarily focus on artificial intelligence, deep learning, computer vision, bioinformatics, and data mining. He has authored or coauthored several publications in these areas. He can be contacted at email: adiwibowo@lecturer.undip.ac.id.