Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1132

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

Hyperparameter Tuning Strategy for YOLOv8n in Real-Time Post-Accident Traffic Monitoring

I Nyoman Eddy Indrayana[1*]✉, Made Sudarma[2], I Ketut Gede Darma Putra[3], Anak Agung Kompiang Oka Sudana[3]

[1]Engineering Science Department, Faculty of Engineering, Universitas Udayana, Badung, Indonesia
[2]Electrical Engineering Department, Faculty of Engineering, Universitas Udayana, Badung, Indonesia
[3]Information Technology Department, Faculty of Engineering, Universitas Udayana, Badung, Indonesia
Corresponding Author: eddyindrayana@pnb.ac.id

Article Information

Article history:

No. 1132
Rec. April 02, 2026
Rev. May 12, 2026
Acc. May 19, 2026
Pub. May 20, 2026
Page. 1580 – 1599

Keywords:

- *Yolov8n*
- *Hyperparameter Optimization*
- *Post-Accident Detection*
- *Intelligent Traffic Monitoring*
- *Damaged Vehicle Detection*

ABSTRACT

Traffic accidents continue to provide a considerable difficulty in contemporary transportation systems, frequently leading to vehicle damage and heightened risks for pedestrians on streets. Precise and instantaneous identification of post-accident scenarios is thus crucial for facilitating swift response and sophisticated traffic management. This research introduces a streamlined object detection methodology utilizing YOLOv8n to recognize six essential traffic-related categories: bus, automobile, damaged vehicle, motorbike, pedestrian, and truck. The main aim is to examine the impact of hyperparameter modification on detection efficacy, specifically in recognizing damaged automobiles as signs of post-accident situations. Twelve model configurations were created by systematically altering three hyperparameters: learning rate (0.01, 0.001, and 0.0001), batch size (32 and 64), and optimizer type (Adam and MuSGD). All models underwent training for 200 epochs with a dataset derived from actual traffic situations, augmented by techniques such as grayscale transformation, blurring, and rotation. The performance evaluation utilized precision, recall, F1-score, mAP50, and mAP50:95. The findings indicate that hyperparameter selection substantially influences convergence stability and detection accuracy. The optimal model attained a mAP50 of 0.905 and a mAP50:95 of 0.751, utilizing a learning rate of 0.01, a batch size of 64, and the Adam optimizer. Moreover, substantial items like cars, buses, and trucks were identified with high precision, whereas damaged vehicles and pedestrians necessitated more meticulous calibration due to increased visual variability.The findings indicate that optimized lightweight models can attain competitive performance, rendering them appropriate for real-time intelligent traffic monitoring applications.

Jurnal Teknologi Informasi dan Pendidikan
Volume 19, No. 2, September 2026
https://doi.org/10.24036/jtip.v19i2.1132

How to Cite:

Indrayana, I. N. E., & et al. (2026). Hyperparameter Tuning Strategy for YOLOv8n in Real-Time Post-Accident Traffic Monitoring. Jurnal Teknologi Informasi Dan Pendidikan, 19(2), 1580-1599. https://doi.org/10.24036/jtip.v19i2.1132

1. INTRODUCTION

Road traffic collisions continue to be a primary source of injuries and deaths globally. The World Health Organization reports that millions of traffic events transpire each year, resulting in damaged vehicles and exposed vulnerable pedestrians on streets. The swift identification of post-accident scenarios is essential for emergency response systems and intelligent transportation infrastructures [1]. The proliferation of security cameras in urban settings has become computer vision technology an efficient instrument for automated traffic monitoring and incident detection [2].

Recent advancements in deep learning have markedly enhanced item detection efficacy in intricate situations. Detectors based on convolutional neural networks (CNN), such as the YOLO (You Only Look Once) series, have gained widespread acceptance owing to their equilibrium between detection precision and real-time efficacy [3]. The current YOLOv8 architecture exhibits enhanced detection accuracy [4], training stability, and streamlined deployment relative to earlier YOLO versions [5]. These attributes render YOLOv8 especially appropriate for intelligent transportation systems necessitating real-time analysis [6].

Numerous research has investigated the application of object detection models for traffic surveillance [7], encompassing vehicle detection, pedestrian detection, and accident identification [8], [9], [10]. Redmon et al. presented the YOLO framework as a cohesive detection architecture with real-time capabilities [11], [12]. Subsequent advancements, including YOLOv5 and YOLOv7 [4], [13], significantly improved detection precision and inference efficiency for traffic surveillance applications [6], [14]. Nevertheless, several studies concentrate predominantly on general vehicle identification instead of concurrently identifying post-accident indications, such as damaged automobiles and the presence of pedestrians [15]. Moreover, model performance is significantly affected by hyperparameter configurations, including learning rate, optimizer, and batch size [16]. Inadequate hyperparameter selection might result in unstable convergence or inadequate detection efficacy [17].

A further issue in practical traffic monitoring systems is the constraint of computer resources [18]. Extensive detection models like YOLOv8m, YOLOv8l, and YOLOv8x deliver superior accuracy but necessitate substantial GPU memory and computational resources [15], [19]. Conversely, lightweight architectures such as YOLOv8n provide expedited inference and reduced computing expenses, rendering them appropriate for edge devices or embedded traffic monitoring systems [6], [20]. Prior research indicates that lightweight detectors can attain competitive performance when effectively adjusted via hyperparameter tweaking and dataset adaptation [21], [22].

This paper presents a hyperparameter optimization method for the YOLOv8n model to enhance the detection of cars, damaged vehicle and people in roadway settings. The algorithm is taught to identify six categories: bus, car, damaged vehicle, motorbike, pedestrian, and truck. Three essential hyperparameters are rigorously assessed: learning rate, batch size, and optimizer type. Twelve model configurations are trained and evaluated based on precision, recall, F1-score, and mAP50 metrics, in addition to their convergence behavior across 200 epochs. This research primarily contributes to the establishment of a lightweight deep learning detection framework for post-accident road monitoring utilizing YOLOv8n, a thorough hyperparameter optimization analysis, and a detailed evaluation across six traffic-related categories, with a specific focus on damaged vehicles and pedestrians as essential post-accident indicators.

2. RESEARCH METHOD

2.1 Dataset and Preprocessing

The dataset utilized in this study was derived from authentic road traffic scenes gathered from two main sources: publicly accessible videos from the YouTube platform and surveillance footage acquired from Closed-Circuit Television (CCTV) cameras positioned along several major roads in Denpasar and Badung Regency, Bali. These sites exemplify the most busy urban traffic arteries in the region, where traffic density is significantly affected by everyday commuting and tourism-related transportation. The traffic dynamics in Denpasar and Badung are characterised by a predominance of motorbikes, congested urban conditions, tourism-induced traffic, and diverse vehicle interactions. These conditions generate extremely dynamic traffic scenarios characterised by frequent occlusion and erratic item movement, hence complicating post-accident detection considerably. The utilization of actual surveillance data guarantees that the dataset accurately represents authentic traffic conditions, including fluctuations in illumination, camera angles, traffic volume, and environmental influences. Video data acquired from these sources underwent a frame extraction process to transform continuous video streams into discrete image frames appropriate for object detection training. Frames were collected at designated intervals to prevent repetition while ensuring adequate variety in traffic conditions and item

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

appearances. This methodology enables the dataset to encompass a variety of road conditions, including disparate vehicle velocities, fluctuating traffic density, partial vehicle occlusion, and pedestrian crossings. Subsequent to the frame extraction process, all images were manually annotated with bounding boxes to delineate the objects of interest. The annotation procedure utilized conventional labeling tools that provide accurate localization of objects within each frame. Every object instance was categorized into one of the predetermined classes utilized in this investigation. The selection of object types was derived from an observational analysis of the most prevalent road users in urban traffic settings. Objects frequently encountered on public roadways were prioritized to guarantee the detection model's applicability for actual traffic monitoring purposes. Consequently, this study delineated six object classes: bus, car, damaged vehicle, motorbike, pedestrian, and truck. The automobile and motorcycle categories constitute the predominant vehicle types commonly seen in urban traffic in Bali, especially owing to the extensive utilization of motorcycles for everyday commuting. The bus and truck categories denote larger vehicles that are crucial for public transportation and products distribution, however they are less commonly observed than smaller vehicles. The pedestrian category was incorporated to signify vulnerable road users who frequently engage with motor traffic, particularly in proximity to crossroads, crossings, and roadside zones. This dataset uniquely includes the damaged vehicle class, specifically intended to facilitate the detection of post-accident situations. Impaired cars generally display structural distortion, fractured parts, or atypical alignment on the thoroughfare. These visual attributes distinguish them from complete automobiles and serve as crucial markers for identifying potential traffic incidents. The incorporation of this class is crucial for the development of sophisticated traffic monitoring systems that can detect early indicators of road accidents or perilous conditions. A data preprocessing phase utilizing data augmentation techniques was implemented to boost the dataset's diversity and robustness. Data augmentation is frequently employed in deep learning-based object detection to enhance dataset diversity and bolster model generalization capacity. This study included three augmentation techniques: grayscale transformation, image blurring, and rotation. The grayscale transformation was implemented to replicate fluctuations in lighting conditions and to prompt the model to concentrate on structural aspects of objects rather than color data. Blur augmentation was implemented to replicate motion blur and fluctuations in camera focus commonly observed in real-world traffic surveillance systems. Furthermore, rotation augmentation was executed by rotating images at +15 degrees and −15 degrees to replicate differences in camera orientation and object viewing angles. These augmentation procedures enable the model to acquire more resilient visual representations across varied environmental conditions, including alterations in lighting, motion artifacts, and perspective discrepancies. The augmentation technique resulted in a total of 1,121 photos utilized in this study, thereby enhancing the dataset for training the object detection model. The dataset seeks to accurately depict realistic traffic circumstances in metropolitan road settings by integrating several

item types and intricate visual scenarios. The trained detection model, utilizing data from actual road traffic in Denpasar and Badung, is anticipated to generalize effectively to real-world traffic monitoring systems, especially in smart city applications and post-accident detection frameworks.

2.2 YOLOv8 Architecture

This study utilizes an object detection model founded on the YOLOv8 architecture, the most recent iteration of the YOLO (You Only Look Once) series. YOLOv8 is classified as a single-stage object detector, doing object localization and classification concurrently within a singular neural network. This design facilitates expedited inference relative to two-stage detectors like region-based convolutional neural networks, rendering YOLO architectures exceptionally appropriate for real-time applications, including intelligent transportation systems and traffic surveillance.

The YOLOv8 architecture consists of three primary components: backbone, neck, and detecting head [23]. These components collaborate to extract visual features, consolidate multi-scale information, and generate final object detection predictions. The backbone extracts hierarchical visual features from the input image. In deep convolutional neural networks, initial layers generally extract low-level features like edges, textures, and color gradients, whereas subsequent layers acquire higher-level semantic representations, including object forms and structural patterns. YOLOv8 employs an advanced backbone architecture that integrates improved convolutional layers and residual connections to boost feature representation while preserving computational performance [24]. The backbone gradually diminishes the spatial resolution of the feature maps while augmenting the number of feature channels. This process enables the network to acquire progressively abstract visual information crucial for differentiating between various object categories. A significant attribute of the YOLOv8 backbone is its capacity to extract distinctive features in intricate situations where objects may manifest at varying scales, lighting conditions, or partial obstructions. This capacity is especially crucial in traffic monitoring situations where vehicles and pedestrians often intersect or appear at different distances from the camera.

The neck component is engineered to consolidate and enhance features acquired from various levels of the backbone network [24]. In object identification tasks, things can manifest at different sizes inside an image; consequently, the detection model must proficiently identify both small and large items. The YOLOv8 neck utilizes a multi-scale feature aggregation technique that integrates feature maps from various backbone stages [23]. This approach allows the model to maintain both intricate spatial details and overarching semantic information. Consequently, the network is capable of detecting objects across a broad spectrum of scales, ranging from diminutive pedestrians to substantial vehicles like buses and trucks. The feature aggregation procedure is executed using a series of upsampling and downsampling processes integrated with convolutional

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

layers. These processes facilitate the integration of information from deeper semantic layers with spatially detailed features from preceding layers, resulting in enhanced feature representations for object detection. Multi-scale feature fusion is especially advantageous for traffic scene analysis, as objects in roadway surroundings frequently manifest at varying distances from the camera. For instance, individuals or motorbikes positioned at a considerable distance from the camera may comprise only a minimal number of pixels, yet substantial vehicles like buses may encompass a significant amount of the image.

The detection head constitutes the ultimate phase of the YOLOv8 architecture, wherein object predictions are produced [25]. Utilizing the consolidated feature maps generated by the neck network, the detecting head forecasts three essential outputs for each identified object. Bounding box coordinates delineate the spatial position of the identified object inside the image. The object confidence score indicates the likelihood that a predicted bounding box encompasses a legitimate object. The class probability distribution denotes the probability that the identified object corresponds to one of the established categories. In contrast to previous YOLO iterations that utilized anchor-based detection methods, YOLOv8 employs an anchor-free detection approach [26]. This method streamlines the prediction process and diminishes the quantity of hyperparameters necessary for training. Eliminating specified anchor boxes enables the model to directly predict object locations from feature map responses, enhancing training stability and localization performance. The detection head functions across many feature map scales, enabling the model to concurrently identify objects of diverse sizes. This capacity is especially crucial for road traffic situations when entities like pedestrians, motorcycles, and vehicles display considerable size discrepancies.

2.2.1 Selection of YOLOv8n Variant

The YOLOv8 architecture has multiple model versions with varying complexities: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large). These versions essentially differ in network depth, parameter count, and computing demands. This study utilized the YOLOv8n version as the foundational detection model. YOLOv8n is the most lightweight variation of the YOLOv8 family, featuring substantially less characteristics than its larger counterparts [27]. This streamlined architecture enables the model to attain accelerated inference speeds while preserving competitive detection accuracy. Employing YOLOv8n is especially beneficial for intelligent traffic monitoring systems, where real-time processing is crucial. Traffic surveillance systems frequently analyze continuous video streams from numerous cameras concurrently, necessitating efficient models capable of performing inference with minimal computing delay. Despite its lightweight construction, YOLOv8n preserves robust detection efficacy owing to enhancements in feature extraction, anchor-free detection methodologies, and refined training procedures incorporated in the YOLOv8 framework. These attributes

render YOLOv8n particularly appropriate for implementation in resource-limited settings, such as edge computing devices or embedded traffic surveillance systems. This study seeks to get dependable detection performance for automobiles, damaged vehicles, and people in real-world road situations by integrating the efficiency of the YOLOv8n architecture with systematic hyperparameter optimization. In this study, the hyperparameters used can be seen in Table 1 and the combination of all hyperparameter values can be seen in Table 2.

Table 1. Hyperparameter Detail Yolov8n Approch

Hyperparameter	Value
learning rate	0.01, 0.001, 0.0001
batch size	32, 64
optimizer	MuSGD, Adam

Table 2. List of Hyperparameter Value Combinations

No	Code Model	Learning Rate	Batch Size	Optimizer
1	M01	0.01	32	Adam
2	M02	0.01	64	Adam
3	M03	0.001	32	Adam
4	M04	0.001	64	Adam
5	M05	0.0001	32	Adam
6	M06	0.0001	64	Adam
7	M07	0.01	32	MuSGD
8	M08	0.01	64	MuSGD
9	M09	0.001	32	MuSGD
10	M10	0.001	64	MuSGD
11	M11	0.0001	32	MuSGD
12	M12	0.0001	64	MuSGD

2.3 Evaluation Matrix

This study uses a confusion matrix to measure classification performance by comparing predicted labels with the actual labels (ground truth). This matrix summarizes prediction results into four categories: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). In object detection, TP indicates correctly detected objects, FP represents incorrect detections, TN refers to correctly rejected non-object cases, and FN indicates missed detections. Based on these four components, several evaluation metrics can be derived. Precision measures the proportion of correctly predicted positive detections among all predicted positives[28] and is formulated as (1):

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

meanwhile, Recall (or sensitivity) measures the proportion of correctly detected positive objects among all actual positive instances (2) [29]:

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

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

and F1-score represents the harmonic mean of precision and recall (3) [27]. These metrics provide a balanced and reliable assessment of the proposed model's detection capability, particularly in distinguishing correct detections from false alarms and missed objects in traffic monitoring scenarios.

$$F1 - score = 2X \frac{Precision\ X\ Recall}{Precision+Recall} \quad (3)$$

3. RESULTS AND DISCUSSION

This section provides a thorough assessment of the suggested hyperparameter optimization approach implemented on the YOLOv8n model for post-accident traffic item recognition. Twelve model configurations were created by methodically altering three primary hyperparameters: learning rate, batch size, and optimizer, and trained for 200 epochs. The efficacy of each model was quantitatively evaluated utilizing typical object detection metrics, encompassing precision, recall, F1-score, mean Average Precision at an IoU threshold of 0.5 (mAP50), and mean Average Precision across various IoU thresholds (mAP50:95). The incorporation of both mAP50 and mAP50:95 metrics offers a more thorough evaluation of model performance, with mAP50 assessing detection accuracy at a specific IoU threshold, and mAP50:95 gauging resilience over various levels of localization stringency. This dual assessment is especially crucial in post-accident situations, where precise localization of damaged automobiles and people is essential for dependable incident detection. The outcomes of the twelve model configurations are presented in Tables 3 to 14, illustrating the influence of hyperparameter selection on detection performance across six item categories: buses, automobiles, damaged vehicles, motorcyclists, pedestrians, and trucks. This comparative study enables the identification of the ideal hyperparameter combination that achieves the best balance of detection accuracy, localization precision, and generalization capability.

Tables 3 through 8 encapsulate the assessment outcomes of models trained with the Adam optimizer. Table 3 delineates the performance of model M01, succeeded by Table 4 for model M02, Table 5 for model M03, Table 6 for model M04, Table 7 for model M05, and Table 8 for model M06. Each table presents a thorough assessment of detection performance across all item categories, encompassing parameters such as precision, recall, F1-score, mAP50, and mAP50:95.

Table 3. Yolov8n model evaluation results (M01)

Class	Precision	Recall	F1-Score	mAP:50	mAP50:95
all class	0.830	0.894	0.861	0.904	0.739
bus	0.867	0.977	0.919	0.967	0.899
car	0.910	0.966	0.937	0.978	0.863
damageVehicle	0.893	0.912	0.902	0.942	0.837
motorcycle	0.697	0.779	0.736	0.746	0.447

pedestrian	0.776	0.762	0.769	0.832	0.572
truck	0.838	0.971	0.900	0.960	0.814

Table 4. Yolov8n model evaluation results (M02)

Class	Precision	Recall	F1-Score	mAP:50	mAP50:95
all class	0.879	0.850	0.864	0.905	0.751
bus	0.892	0.828	0.859	0.964	0.900
car	0.945	0.950	0.947	0.974	0.857
damageVehicle	0.931	0.889	0.910	0.947	0.844
motorcycle	0.778	0.775	0.776	0.754	0.479
pedestrian	0.827	0.720	0.770	0.801	0.567
truck	0.900	0.938	0.919	0.988	0.862

Table 5. Yolov8n model evaluation results (M03)

Class	Precision	Recall	F1-Score	mAP:50	mAP50:95
all class	0.862	0.861	0.861	0.898	0.742
bus	0.856	0.893	0.874	0.957	0.888
car	0.921	0.963	0.942	0.972	0.838
damageVehicle	0.920	0.901	0.910	0.949	0.848
motorcycle	0.706	0.798	0.749	0.704	0.429
pedestrian	0.773	0.738	0.755	0.820	0.555
truck	0.997	0.875	0.932	0.985	0.892

Table 6. Yolov8n model evaluation results (M04)

Class	Precision	Recall	F1-Score	mAP:50	mAP50:95
all class	0.926	0.818	0.869	0.894	0.735
bus	0.920	0.817	0.865	0.953	0.889
car	0.956	0.936	0.946	0.970	0.848
damageVehicle	0.952	0.866	0.907	0.937	0.855
motorcycle	0.783	0.798	0.790	0.717	0.440
pedestrian	0.966	0.680	0.798	0.821	0.561
truck	0.977	0.812	0.887	0.967	0.818

Table 7. Yolov8n model evaluation results (M05)

Class	Precision	Recall	F1-Score	mAP:50	mAP50:95
all class	0.683	0.615	0.647	0.692	0.381
bus	0.837	0.967	0.897	0.947	0.827
car	0.952	0.915	0.933	0.940	0.742
damageVehicle	0.926	0.912	0.919	0.936	0.783
motorcycle	0.792	0.732	0.761	0.711	0.410
pedestrian	0.683	0.615	0.647	0.692	0.381
truck	1.000	0.490	0.658	0.795	0.642

Table 8. Yolov8n model evaluation results (M06)

Class	Precision	Recall	F1-Score	mAP:50	mAP50:95
all class	0.772	0.712	0.741	0.733	0.517
bus	0.784	0.967	0.866	0.946	0.771
car	0.892	0.897	0.894	0.930	0.676
damageVehicle	0.798	0.826	0.812	0.898	0.715
motorcycle	0.696	0.616	0.654	0.627	0.349
pedestrian	0.551	0.532	0.541	0.496	0.226

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

truck	0.909	0.438	0.591	0.500	0.366

Conversely, Tables 9 through 14 exhibit the evaluation outcomes of models trained with the MuSGD optimizer. Table 9 pertains to model M07, Table 10 to model M08, Table 11 to model M09, Table 12 to model M10, Table 13 to model M11, and Table 14 to model M12. Like the Adam-based models, each table presents comprehensive performance measurements for all six object classes, facilitating a direct comparison between the two optimization methodologies.

Table 9. Yolov8n model evaluation results (M07)

Class	Presisi	Recall	F1-Score	mAP:50	mAP50:95
all class	0.886	0.838	0.861	0.891	0.748
bus	0.924	0.815	0.866	0.938	0.876
car	0.962	0.933	0.947	0.969	0.856
damageVehicle	0.928	0.901	0.914	0.952	0.860
motorcycle	0.691	0.731	0.710	0.675	0.430
pedestrian	0.872	0.683	0.766	0.827	0.581
truck	0.939	0.968	0.953	0.984	0.884

Table 10. Yolov8n model evaluation results (M08)

Class	Presisi	Recall	F1-Score	mAP:50	mAP50:95
all class	0.903	0.806	0.852	0.897	0.751
bus	0.909	0.830	0.868	0.955	0.898
car	0.956	0.941	0.948	0.978	0.866
damageVehicle	0.909	0.874	0.891	0.942	0.840
motorcycle	0.760	0.760	0.760	0.727	0.459
pedestrian	0.919	0.683	0.784	0.824	0.576
truck	0.966	0.750	0.844	0.957	0.867

Table 11. Yolov8n model evaluation results (M09)

Class	Presisi	Recall	F1-Score	mAP:50	mAP50:95
all class	0.780	0.861	0.819	0.868	0.662
bus	0.856	0.994	0.920	0.953	0.887
car	0.912	0.933	0.922	0.962	0.780
damageVehicle	0.890	0.934	0.911	0.935	0.781
motorcycle	0.600	0.798	0.685	0.729	0.457
pedestrian	0.568	0.754	0.648	0.713	0.442
truck	0.857	0.751	0.801	0.918	0.623

Table 12. Yolov8n model evaluation results (M10)

Class	Presisi	Recall	F1-Score	mAP:50	mAP50:95
all class	0.865	0.826	0.845	0.875	0.668
bus	0.848	0.983	0.911	0.875	0.669
car	0.932	0.941	0.936	0.959	0.775
damageVehicle	0.906	0.879	0.892	0.926	0.779
motorcycle	0.751	0.731	0.741	0.700	0.407
pedestrian	0.825	0.673	0.741	0.756	0.468
truck	0.928	0.750	0.830	0.946	0.705

Table 13. Yolov8n model evaluation results (M11)

Class	Presisi	Recall	F1-Score	mAP:50	mAP50:95
all class	0.735	0.437	0.548	0.472	0.293
bus	0.733	0.833	0.780	0.874	0.628
car	0.810	0.617	0.700	0.751	0.463
damageVehicle	0.729	0.443	0.551	0.558	0.362
motorcycle	0.667	0.596	0.630	0.512	0.250
pedestrian	0.472	0.135	0.210	0.138	0.054
truck	1.000	0	0	0	0

Table 14. Yolov8n model evaluation results (M12)

Class	Presisi	Recall	F1-Score	mAP:50	mAP50:95
all class	0.668	0.489	0.565	0.472	0.295
bus	0.685	0.983	0.807	0.876	0.664
car	0.724	0.661	0.691	0.736	0.457
damageVehicle	0.707	0.593	0.645	0.627	0.379
motorcycle	0.569	0.567	0.568	0.463	0.225
pedestrian	0.326	0.127	0.183	0.130	0.046
truck	1.000	0	0	0	0

This section includes graphical visuals alongside the tabular assessment data to facilitate a clearer comparative examination of model performance across all settings of YOLOv8n. These visualizations depict the distribution and variability of essential assessment measures, such as precision, recall, F1-score, mAP50, and mAP50:95, for each model and object class. Figure 1 illustrates the comparative performance across all classes, emphasizing the efficacy of each model based on the five evaluation parameters.

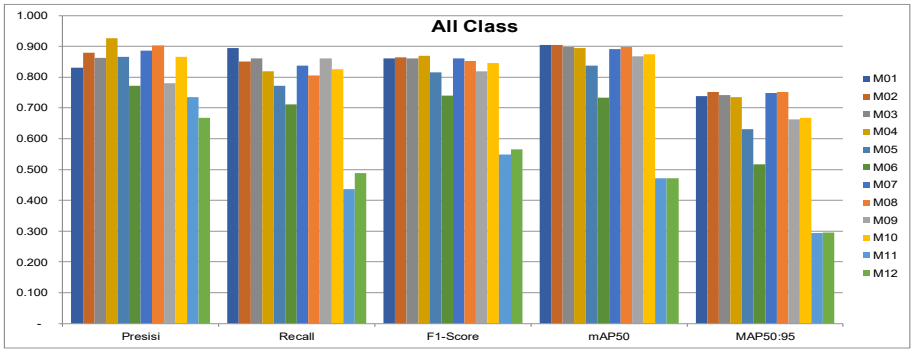
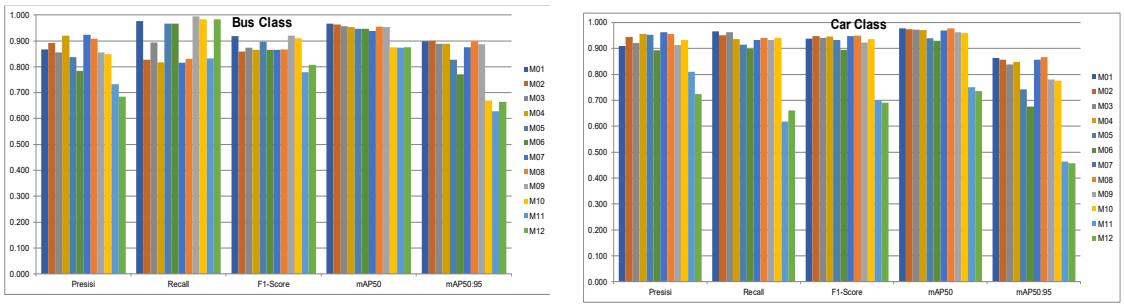


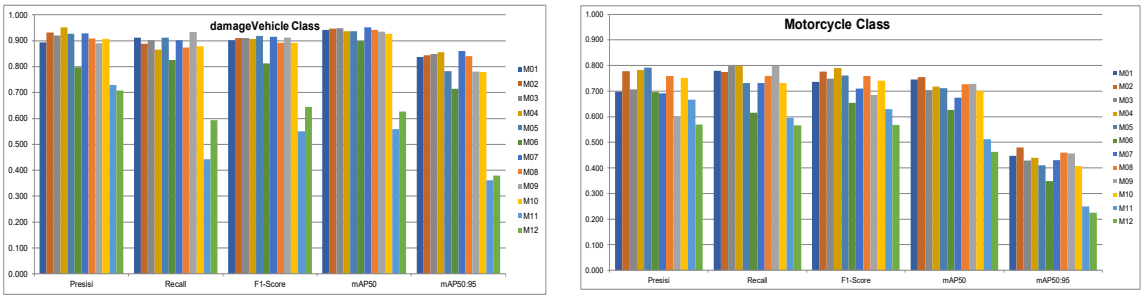
Figure 1. Evaluation chart for all class

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id



(a) (b)

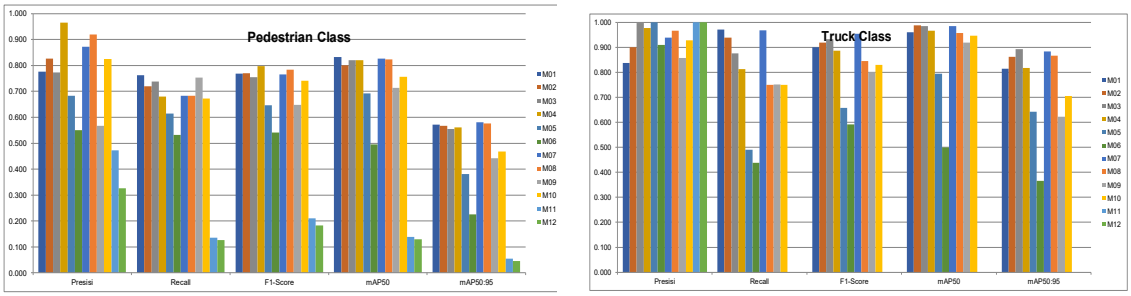
Figure 2. Evaluation chart for (a) bus class ,(b) car class



(a) (b)

Figure 3. Evaluation chart for (a) damageVehicle class, (b) motorcycle class

This image offers a comprehensive overview of model consistency and efficacy throughout the full dataset. Additionally, class-specific performance comparisons are illustrated in the following pictures. Figure 2(a) depicts the comparison of precision, recall, F1-score, mAP50, and mAP50:95 for the bus category, whereas Figure 2(b) displays the analogous comparison for the vehicle category.



(a) (b)

Figure 4. Evaluation chart for (a) pedestrian class, (b) truck class

Figure 3(a) illustrates the performance comparison for the damaged vehicle category, whereas Figure 3(b) pertains to the motorbike category. Furthermore, Figure 4(a)

displays the assessment outcomes for the pedestrian category, whereas Figure 4(b) illustrates those for the vehicle category.

The experimental findings illustrated in Tables 3 through 14 and Figures 1 through 4 indicate that hyperparameter adjustment is essential for influencing the detection efficacy of the YOLOv8n model across all object categories. Of the twelve assessed configurations, models trained with learning rates between 0.001 and 0.01 consistently demonstrated enhanced performance relative to those taught with a lower learning rate of 0.0001. Specifically, model M02 (learning rate = 0.01, batch size = 64, Adam optimizer) attained the highest mAP50 (0.905) and mAP50:95 (0.751), demonstrating robust generalization ability and accurate object localization. Concurrently, model M04 (learning rate = 0.001, batch size = 64, Adam) attained the highest F1-score (0.869), indicating an ideal equilibrium between precision and recall.

A discernible trend evident in all models is the substantial influence of batch size on performance. Models trained with a batch size of 64 consistently surpassed those trained with a batch size of 32, indicating that bigger batch sizes yield more stable gradient estimation and enhanced convergence behavior. The Adam optimizer exhibited superior resilience and consistency relative to MuSGD, especially in scenarios with low learning rates. This is apparent in models M11 and M12, where the integration of a minimal learning rate (0.0001) with MuSGD led to significant performance deterioration, with mAP50 decreasing to 0.472 and mAP50:95 falling below 0.30, signifying ineffective convergence.

An in-depth investigation of distinct item classes demonstrates significant discrepancies in detection performance. Categories like car, bus, and truck shown consistently superior performance across the majority of configurations, with mAP50 values surpassing 0.95 in multiple models. This can be ascribed to their very substantial object size and unique visual characteristics, which facilitate the model's learning process. Conversely, the motorbike and pedestrian categories had relatively inferior performance, with mAP50 values typically fluctuating between 0.70 and 0.83. These classes face increased problems owing to reduced object sizes, frequent occlusion, and heightened diversity in appearance and orientation. The damaged vehicle class, which constitutes the primary contribution of this work, exhibited robust yet more sensitive performance. Although some models attained F1-scores over 0.90 and mAP50 values surpassing 0.94, a decline in performance was noted with inferior hyperparameter settings. This trend can be attributed to the significant intra-class variety of damaged cars, which may display various structural deformations, partial occlusion, and unusual forms. Thus, precise identification of damaged automobiles necessitates meticulous adjustment of hyperparameters to guarantee resilient feature learning. Moreover, extreme hyperparameter settings, especially very low learning rates in conjunction with MuSGD, resulted in severe performance deterioration across several classes. In the truck category, models M11 and M12 exhibited zero recall and zero mean Average Precision (mAP), signifying total training failure. This underscores the essential

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

need of choosing suitable hyperparameter values to prevent underfitting and inefficient gradient updates.

The examination of the experimental data for the damageVehicle class indicates that the learning rate is the most influential hyperparameter impacting mAP50 performance, in comparison to the optimiser and batch size. Models trained with a learning rate of 0.01 consistently attained optimal performance, with mAP50 values between 0.942 and 0.952. As the learning rate diminished to 0.001, performance exhibited considerable stability, with mAP50 values ranging from 0.926 to 0.949. A subsequent reduction to 0.0001 resulted in a notable decrease, with mAP50 falling to 0.558 and 0.627 across several configurations. The damagedVehicle class exhibits a pronounced sensitivity to learning rate parameters owing to its intricate visual attributes, including structural deformation, uneven forms, and diverse object orientations. Minuscule learning rates impede the optimisation process and restrict the model's capacity to properly learn discriminative deformation characteristics. The optimiser also affected detection performance, albeit to a lesser extent than the learning rate. The Adam optimiser exhibited enhanced stability and robustness, especially at reduced learning rates. For example, with a learning rate of 0.0001, Adam attained a mAP50 of 0.898, whereas MuSGD fell below 0.63. Conversely, batch size exerted a negligible influence on performance, mostly enhancing training stability rather than significantly improving accuracy.

This section conducts a comparative examination of the lightweight YOLOv8n architecture against the more resource-demanding YOLOv8l and YOLOv8x variations to assess the efficacy of the proposed lightweight detection framework. The results for the Yolov8l model are presented in Table 15, while the results for the Yolov8x model are displayed in Table 16.

Table 15. Yolov8l model evaluation results using learning rate 0.01, batch size 64, Optimizer=Adam

Class	Presisi	Recall	F1-Score	mAP:50	mAP50:95
all class	0.868	0.862	0.865	0.898	0.738
bus	0.845	0.908	0.875	0.963	0.901
car	0.937	0.959	0.948	0.972	0.855
damageVehicle	0.88	0.879	0.879	0.912	0.786
motorcycle	0.775	0.694	0.732	0.738	0.436
pedestrian	0.836	0.768	0.801	0.818	0.596
truck	0.939	0.965	0.952	0.984	0.856

The two larger architectures were trained use the ideal hyperparameter configuration derived from YOLOv8n, specifically a learning rate of 0.01, a batch size of 64, and the Adam optimiser. This study seeks to evaluate the balance between detection accuracy and computing complexity in post-accident traffic monitoring contexts. This section evaluates precision, recall, F1-score, mAP50, and mAP50:95, while also comparing inference performance to determine the viability of real-time deployment. Table 17 illustrates the comparative inference performance of Yolov8n, Yolov8s, and Yolov8x. All

inference studies were performed on Google Colab with an NVIDIA T4 GPU featuring 16 GB of GDDR6 memory.

Table 16. Yolov8x model evaluation results using learning rate 0.01, batch size 64, Optimizer=Adam

Class	Presisi	Recall	F1-Score	mAP:50	mAP50:95
all class	0.838	0.888	0.862	0.888	0.756
bus	0.843	0.967	0.901	0.967	0.896
car	0.938	0.977	0.957	0.98	0.877
damageVehicle	0.855	0.879	0.867	0.932	0.829
motorcycle	0.744	0.697	0.720	0.67	0.441
pedestrian	0.756	0.81	0.782	0.805	0.607
truck	0.894	0.998	0.943	0.973	0.886

The comparative results aim to elucidate whether lightweight architectures like YOLOv8n can attain competitive detection performance compared to larger models, while simultaneously preserving markedly lower computational demands and accelerated inference speed, which are essential for intelligent transportation systems and edge-based traffic surveillance applications.

Table 17. Comparison of inferences between Yolov8n, Yolov8l, and Yolov8x

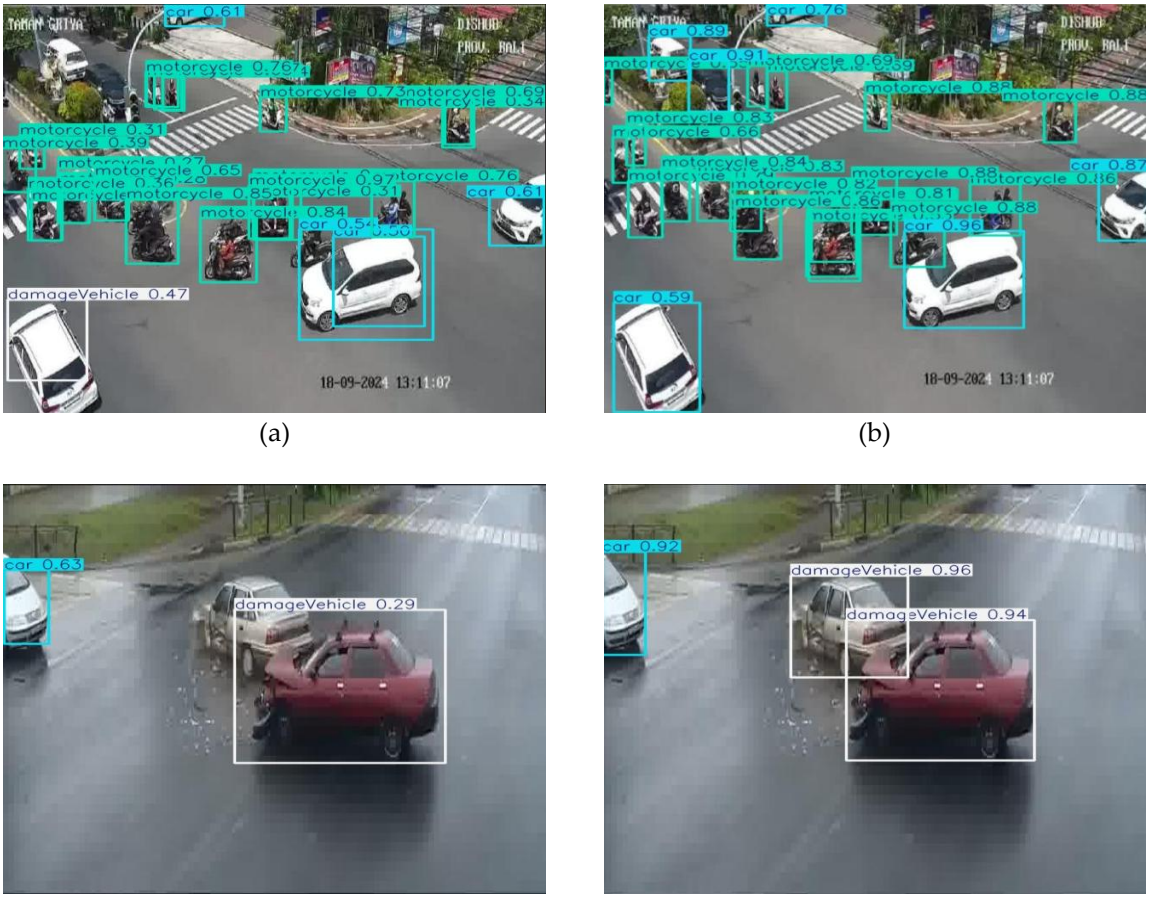
Yolov8 variant	Inference (ms)
Yolov8n	4.4
Yolov8l	15.2
Yolov8x	22.6

The experimental results in Tables 15, Tabel 16 and Table 17 indicate that the lightweight YOLOv8n design outperformed the bigger YOLOv8l and YOLOv8x models under identical hyperparameter settings. YOLOv8n achieved the highest all-class mAP50 of 0.905, surpassing YOLOv8l and YOLOv8x, which had mAP50 values of 0.898 and 0.888, respectively. The findings suggest that augmenting model complexity does not consistently enhance detection accuracy, especially for the post-accident traffic dataset utilised in this investigation. The inference analysis presented in Table 17 underscores the efficacy of the lightweight design. YOLOv8l necessitated roughly 3.4 times the inference duration of YOLOv8n, but YOLOv8x demanded approximately 5.1 times longer. Larger models typically offer more representational capacity. However, they also entail significant processing demands, diminishing their applicability for real-time intelligent transportation systems. The findings indicate that the optimised hyperparameter setup was more suitable with YOLOv8n, facilitating enhanced feature learning and generalisation, particularly for intricate classes like damagedVehicle. The results indicate that YOLOv8n achieves an optimal equilibrium between detection precision and computing economy, rendering it exceptionally appropriate for real-time traffic surveillance following accidents.

This study conducted qualitative testing to assess the behavior of two YOLOv8n configurations, M02 and M11, which are the best and worst-performing models,

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

respectively. Model M02 was chosen because to its superior mAP score compared to all other configurations, but model M11 demonstrated the lowest mAP, signifying inadequate convergence and detection performance. Figure 5 illustrates a comparative assessment of these two models in both crowded (many objects) and sparse (limited objects) traffic conditions.



(a) (b)

(c) (d)

Figure 5. Test result for (a) many objects using M11 model, (b) many objects using M02 model, (c) few objects using M11 model, (d) few objects using M02 model

Figure 5(a), depicting a congested traffic scenario using model M11, demonstrates a significant misclassification wherein a vehicle intended to be categorized as an automobile is erroneously identified as a damaged vehicle. This indicates that M11 exhibits inadequate feature discrimination, especially in intricate scenarios with overlapping items. Conversely, Figure 5(b) illustrates that model M02 accurately identifies and classifies all objects within a similarly packed environment, showcasing its robustness and capacity to manage occlusion and high object density. Under sparse situations, as illustrated in Figure 5(c), model M11 fails to identify one instance of a damaged car, leading to a false negative error that is

especially significant in post-accident detection contexts. This suggests that the model is insensitive to nuanced visual characteristics linked to vehicle damage. In contrast, Figure 5(d) illustrates that model M02 effectively identifies all pertinent objects, including damaged automobiles, with elevated confidence levels, hence affirming its reliability under diverse traffic situations. Model M02 exhibits superior generalization, precise classification, and resilient detection in both dense and sparse situations, whereas M11 displays misclassification and missed detection resulting from an inadequate training setup. These findings corroborate the quantitative evaluation results and underscore that inadequate hyperparameter selection can significantly impair detection performance, particularly in safety-critical contexts such as post-accident traffic monitoring.

The findings affirm that the YOLOv8n architecture, despite its lightweight configuration, can attain superior detection performance when accompanied by well optimized hyperparameters. The results further highlight that hyperparameter selection significantly influences performance more than model architecture scaling in some situations, especially when addressing intricate object categories like damaged automobiles and pedestrians.

4. CONCLUSION

This work conducted a thorough examination of hyperparameter tuning for a lightweight object identification model utilizing YOLOv8n to facilitate post-accident traffic monitoring. The model was developed to identify six essential item categories: bus, automobile, damaged vehicle, motorcycle, pedestrian, and truck, which signify both typical and atypical traffic scenarios on streets. Twelve experimental configurations were assessed by systematically altering the learning rate, batch size, and optimizer across 200 training epochs.

The experimental findings indicate that hyperparameter selection is crucial for influencing detection accuracy and training stability. Models trained with a moderate learning rate (0.001–0.01) and an increased batch size (64) consistently demonstrated enhanced performance across all assessment criteria, including precision, recall, F1-score, mAP50, and mAP50:95. Moreover, the Adam optimizer demonstrated superior robustness and stability relative to MuSGD, especially in scenarios involving low learning rates. The best performance in this study was achieved using a learning rate of 0.01, a batch size of 64, and the Adam optimizer, which produced an mAP50 of 0.905 and an mAP50:95 of 0.751. These findings demonstrate that effective hyperparameter adjustment can substantially improve the efficacy of lightweight detection algorithms.

An exhaustive per-class analysis demonstrated that object classes exhibiting larger and more uniform visual characteristics, such as car, bus, and truck, attained superior detection accuracy, whereas more complex classes like motorcycle and pedestrian necessitated meticulous tuning due to their reduced size and increased variability. The

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

study of the confusion matrix revealed distinct misclassification patterns, especially among visually analogous classes such as car and damaged vehicle, as well as motorbike and pedestrian, offering critical insights into the model's shortcomings and potential areas for enhancement.

These findings suggest numerous potential avenues for future research. Initially, incorporating temporal analysis techniques like multi-object tracking (e.g., DeepSORT) could improve the system's capacity to consistently detect dynamic post-accident occurrences. Secondly, the integration of sophisticated techniques like attention processes or transformer-based architectures could enhance detection efficacy for difficult item categories. Third, the improved YOLOv8n model is deployable on edge computing devices to facilitate real-time intelligent transportation systems and smart city applications, especially in resource-limited settings.

REFERENCES

[1] H. Kumar, C. Singh, A. K. Singh, N. Kumar, Tanu, and N. S. Talwandi, "Smart AI-Enabled Accident Detection and Emergency Response System for Abandoned Roads," in *Proceedings of Data Analytics and Management*, A. Swaroop, B. Virdee, S. D. Correia, and Z. Polkowski, Eds., Cham: Springer Nature Switzerland, 2026, pp. 181–192.

[2] V. A. Adewopo and N. Elsayed, "Smart City Transportation: Deep Learning Ensemble Approach for Traffic Accident Detection," *IEEE Access*, vol. 12, pp. 59134–59147, 2024, doi: 10.1109/ACCESS.2024.3387972.

[3] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," *Computers*, vol. 13, no. 12, 2024, doi: 10.3390/computers13120336.

[4] I. Purwita Sary, E. Ucok Armin, S. Andromeda, E. Engineering, and U. Singaperbangsa Karawang, "Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection Using Aerial Images," *Ultima Computing : Jurnal Sistem Komputer*, vol. 15, no. 1, 2023.

[5] R. Bai, F. Shen, M. Wang, J. Lu, and Z. Zhang, "Improving Detection Capabilities of YOLOv8-n for Small Objects in Remote Sensing Imagery: Towards Better Precision with Simpliied Model Complexity Improving Detection Capabilities of YOLOv8-n for Small Objects in Remote Sensing Imagery: Towards Better Precision with Simplified Model Complexity," 2023, doi: https://doi.org/10.21203/rs.3.rs-3085871/v1.

[6] M. Bakirci, "Utilizing YOLOv8 for enhanced traffic monitoring in intelligent transportation systems (ITS) applications," *Digit. Signal Process.*, vol. 152, p. 104594, Sep. 2024, doi: 10.1016/j.dsp.2024.104594.

[7] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 108–118, Jun. 2000, doi: 10.1109/6979.880968.

[8] I. N. E. Indrayana, M. Sudarma, I. K. G. D. Putra, and A. A. K. O. Sudana, "Improve Nighttime Highway Vehicles and Pedestrian Detection Using Yolov8+CLAHE," in *2024 Ninth International Conference on Informatics and Computing (ICIC)*, IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/ICIC64337.2024.10956682.

[9] E. Zadobrischi and M. Negru, "Pedestrian detection based on TensorFlow YOLOv3 embedded in a portable system adaptable to vehicles," *2020 15th International Conference on Development and Application Systems, DAS 2020 - Proceedings*, pp. 21–26, 2020, doi: 10.1109/DAS49615.2020.9108940.

[10] J. Sang *et al.*, "An Improved YOLOv2 for Vehicle Detection," *Sensors*, vol. 18, no. 12, p. 4272, Dec. 2018, doi: 10.3390/s18124272.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[12] L. Cheng, D. Zhang, and Y. Zheng, "Road Object Detection in Foggy Complex Scenes Based on Improved YOLOv8," *IEEE Access*, vol. 12, pp. 107420–107430, 2024, doi: 10.1109/ACCESS.2024.3438612.

[13] 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," Dec. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/make5040083.

[14] "Vehicle Classification and Counting for Traffic Analysis based on Single-stage YOLOv8 Model," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*, pp. 41–53, Jun. 2024, doi: 10.33103/uot.ijccce.24.2.4.

[15] N. U. A. Tahir, Z. Long, Z. Zhang, M. Asim, and M. ELAffendi, "PVswin-YOLOv8s: UAV-Based Pedestrian and Vehicle Detection for Traffic Management in Smart Cities Using Improved YOLOv8," *Drones*, vol. 8, no. 3, p. 84, Feb. 2024, doi: 10.3390/drones8030084.

[16] I. N. Eddy Indrayana, M. Sudarma, I. K. Gede Darma Putra, and A. A. Kompiang Oka Sudana, "Cross-Domain Fine-Tuning of YOLOv8 for Vehicle Detection in Daytime and Nighttime Traffic Surveillance," in *2025 IEEE International Conference on Data and Software Engineering (ICoDSE)*, 2025, pp. 136–142. doi: 10.1109/ICoDSE68111.2025.11351551.

[17] I. Damaj, S. K. Al Khatib, T. Naous, W. Lawand, Z. Z. Abdelrazzak, and H. T. Mouftah, "Intelligent transportation systems: A survey on modern hardware devices for the era of machine learning," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5921–5942, Sep. 2022, doi: 10.1016/j.jksuci.2021.07.020.

[18] J. Lee and K. Hwang, "YOLO with adaptive frame control for real-time object detection applications," *Multimed. Tools Appl.*, vol. 81, no. 25, pp. 36375–36396, Oct. 2022, doi: 10.1007/s11042-021-11480-0.

[19] L. Wang, H. Wang, S. Letchmunan, R. Xiao, O. H. Ahmed, and Z. Liu, "A systematic literature review of lightweight YOLO models for object detection," *PeerJ Comput. Sci.*, vol. 11, p. e3357, Nov. 2025, doi: 10.7717/peerj-cs.3357.

[20] S. Pudaruth, I. M. Boodhun, and C. W. Onn, "Reducing Traffic Congestion Using Real-Time Traffic Monitoring with YOLOv8," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 10, 2024, doi: 10.14569/IJACSA.2024.01510109.

[21] N. B. A. Karna, M. A. P. Putra, S. M. Rachmawati, M. Abisado, and G. A. Sampedro, "Toward Accurate Fused Deposition Modeling 3D Printer Fault Detection Using Improved YOLOv8 With Hyperparameter Optimization," *IEEE Access*, vol. 11, pp. 74251–74262, 2023, doi: 10.1109/ACCESS.2023.3293056.

P.ISSN: 2086 – 4981
E.ISSN: 2620 – 6390
tip.ppj.unp.ac.id

[22] M. Sohan, T. Sai Ram, and Ch. V. Rami Reddy, "A Review on YOLOv8 and Its Advancements," in *Data Intelligence and Cognitive Informatics*, S. and F.-G. P. Jacob I. Jeena and Piramuthu, Ed., Singapore: Springer Nature Singapore, 2024, pp. 529–545. doi: 10.1007/978-981-99-7962-2_39.

[23] H. Duan, Y. Zhang, S. Zhu, and Z. Liang, "Lightweight multi-scale vehicle object detection algorithm based on optimized YOLOv8," Pattern Analysis and Applications, vol. 28, no. 4, p. 183, Dec. 2025, doi: 10.1007/s10044-025-01562-2.

[24] A. Sundaresan Geetha, M. A. R. Alif, M. Hussain, and P. Allen, "Comparative Analysis of YOLOv8 and YOLOv10 in Vehicle Detection: Performance Metrics and Model Efficacy," *Vehicles*, vol. 6, no. 3, pp. 1364–1382, Aug. 2024, doi: 10.3390/vehicles6030065.

[25] M. Zhang and Z. Zhang, "Research on Vehicle Target Detection Method Based on Improved YOLOv8," *Applied Sciences*, vol. 15, no. 10, p. 5546, May 2025, doi: 10.3390/app15105546.

[26] X. Shi, X. Zhang, Y. Su, and X. Zhang, "Positive Anchor Area Merge Algorithm: A Knowledge Distillation Algorithm for Fruit Detection Tasks Based on Yolov8," *IEEE Access*, vol. 13, pp. 34954–34968, 2025, doi: 10.1109/ACCESS.2025.3544361.

[27] M. Bakirci, "Real-Time Vehicle Detection Using YOLOv8-Nano for Intelligent Transportation Systems," *Traitement du Signal*, vol. 41, no. 04, pp. 1727–1740, Aug. 2024, doi: 10.18280/ts.410407.

[28] A. D. Desta and C. Jian, "Enhancing YOLOv8 for Vehicle Detection in Intelligent Traffic Management," *Metallurgical and Materials Engineering*, vol. 31, no. 4, pp. 190–200, Apr. 2025, doi: 10.63278/1423.

[29] F. Qanouni, H. El Massari, N. Gherabi, and M. El Badaoui, "Road Accident Detection using SVM and Learning: A Comparative Study," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 5, 2024, doi: 10.14569/IJACSA.2024.0150565.