ISSN 2723-6471

EagleEyes: An Artificial Intelligence-Based Approach for Automatic Traffic Violation Detection Using Deep Learning

Windu Gata[1,*], Muhammad Haris[2], Maria Irmina Prasetiyowati[3], Sony Harianto[4]

[1,2,4]Computer Science, Faculty of Information Technology, Universitas Nusa Mandiri, Jakarta, Indonesia

[3]Department of Informatics, Faculty of Engineering and Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia

(Received: June 20, 2025; Revised: August 20, 2025; Accepted: November 25, 2025; Available online: December 9, 2025)

Abstract

Rapid urbanization and the advancement of smart city programs in Indonesia necessitate intelligent, automated solutions for traffic monitoring and law enforcement. This study introduces EagleEyes, an artificial intelligence–based framework designed for automatic detection of multiple traffic violations by integrating the YOLOv8 deep learning architecture with Optical Character Recognition (OCR) for vehicle license plate identification. YOLOv8 was selected due to its anchor-free design, decoupled detection head, and enhanced feature fusion modules, which collectively improve detection accuracy, convergence speed, and small-object recognition compared to YOLOv5 and YOLOv7, while maintaining lightweight computational efficiency suitable for real-time applications. The proposed system was trained on a multi-class dataset representing common Indonesian violations, including seat belt non-compliance, helmet absence, motorcycle overcapacity, and unreadable license plates. Experimental results demonstrate robust performance, achieving a precision of 0.91, recall of 0.92, and mean average precision (mAP@0.5) of 0.96 at the optimal epoch, with an average inference speed of 25 frames per second and total training time of approximately 15 minutes on an NVIDIA RTX GPU. The OCR module attained an average recognition accuracy of 98.7%, although its performance decreased for vehicles captured beyond a five-meter distance due to reduced clarity and illumination inconsistencies. Implemented as a web-based application using the Flask framework, EagleEyes enables flexible browser-based visualization, and can be seamlessly integrated into Indonesia's Electronic Traffic Law Enforcement (ETLE) infrastructure. Overall, the system demonstrates high potential to enhance smart city traffic management through scalable, AI-driven, and ethically responsible automation.

Keywords: Traffic Violation Detection, Artificial Intelligence, Deep learning, Video Processing, Object Detection

1. Introduction

Deep learning-based computer vision has enabled automated image and video interpretation for intelligent transportation systems [1], [2], [3], [4], [5]. In Indonesia, common violations such as missing helmets, unbuckled seat belts, overcapacity on motorcycles, and unreadable license plates remain difficult to monitor through manual enforcement [6], [7], [8], [9]. Hence, an AI-powered automated detection system is essential to support traffic regulation and smart city initiatives [10], [11]. Among various detection models, the You Only Look Once (YOLO) architecture is widely recognized for its high accuracy and real time performance in object detection tasks [12], [13], [14].

YOLOv8 was selected because it provides balanced advancements over both earlier and newer YOLO architectures. Compared to YOLOv5 and YOLOv7, it employs an anchor-free decoupled head and enhanced C2f feature modules that significantly improve precision, recall, and convergence speed, particularly for small and occluded traffic objects [15], [16]. While the newer YOLOv9 introduces additional layers and GELAN blocks, YOLOv8 remains more efficient for real-time deployment on edge devices due to its lighter structure and faster inference time. Therefore, YOLOv8 offers the best trade-off between accuracy, robustness, and computational efficiency for intelligent traffic violation detection. Several studies have been conducted but mostly focused on single violations, such as helmet detection [17], or seat belt detection using Convolutional Neural Networks–Support Vector Machine (CNN-SVM) [18] and

*Corresponding author: Windu Gata (windu@nusamandiri.ac.id)
DOI: https://doi.org/10.47738/jads.v7i1.1097

Convolutional Neural Networks–Long Short-Term Memory (CNN-LSTM) approaches [19], [20]. Before the development of the proposed traffic violation detection system, the authors previously designed an automated Indonesian license plate detection and recognition framework employing OCR and Convolutional Neural Network (CNN) based models. Under daylight conditions, the OCR module achieved an average character recognition accuracy of approximately 99% at a distance of less than 4 meters, demonstrating high reliability for vehicle identification tasks. This prior system established a robust unique plate recognition mechanism capable of tracing and verifying vehicles following a detected violation. Consequently, the earlier work provides a fundamental basis and operational backbone for the integrated violation recognition architecture presented in this study.

License plate detection using OCR methods has also been explored in previous studies [21], [22] however, these approaches have not yet been integrated into broader traffic violation detection frameworks. Even recent studies utilizing YOLOv8 have mainly focused on general vehicle monitoring rather than specifically addressing traffic violations [23], [24]. Furthermore, most existing studies utilize foreign datasets that do not represent Indonesia's traffic characteristics, including vehicle types, plate styles, and driver behavior [25], [26], [27]. Several deep learning approaches, such as video super-resolution models [28], [29], and feature selection optimization for high-dimensional data [30], [31], highlight the importance of algorithmic adaptation to local environmental contexts. Additionally, research on improving classification speed and accuracy through thresholding and feature selection [32], [33] provides valuable insights for further enhancing system performance.

The main purpose of this research is to develop and assess an automated system for detecting traffic violations using the YOLOv8 deep learning model. The proposed system is capable of recognizing several types of violations, such as the absence of seat belt use, non-compliance with helmet regulations, overloading on motorcycles, and vehicle license plate identification. To accomplish this, an experimental methodology is applied, which involves collecting and annotating datasets from Indonesian CCTV footage to accurately reflect local traffic conditions. The annotated images are then used to train the YOLOv8 model, while an OCR component is incorporated to extract license plate details from the detected regions. The system's effectiveness is measured through key performance indicators, including precision, recall, and mean Average Precision (mAP). The final output presents annotated visual results for each detected infraction, confirming the system's potential for monitoring application in smart city traffic management environments.

2. Proposed Method

In previous studies, as listed in table 1, topics related to traffic violations based on computer vision and deep learning have been discussed [10], [11], [15]. Helmet violations have been studied using YOLOv5, which proved effective in detecting helmet usage among motorcycle riders [1], [17]. Meanwhile, YOLOv8 has attracted growing interest due to its superior precision and inference speed for vehicle tracking tasks [8], [21], [22]. Several studies have also integrated OCR modules for automatic license plate reading, achieving fairly accurate results [6], [17]. To the best of current knowledge, no prior study has simultaneously addressed four primary traffic violations seat belt non-compliance, helmet absence, triple motorcycle riding, and license plate recognition within a unified YOLOv8-based automated system. Existing literature predominantly focuses on isolated violation categories or employs foreign and synthetic datasets, which limits applicability to the nuanced and heterogeneous traffic conditions characteristic of Indonesian road environments.

Table 1. State of the Art and Novelty

Application Area	Challenges	Methods	Source	Difference with Our Research
Seat Belt Detection	Sensors easily manipulated, traditional methods inefficient, poor lighting and misuse issues.	CNN-SVM, YOLO, YOLOv7, Deep Learning	[12], [16], [23]	Previous studies focused only on seat belt detection, while our research integrates seat belt, helmet, license plate, and overcapacity detection into a single YOLOv8 system.
Traffic Violations & Driver	Occlusion, inconsistent lig hting, high traffic density, environmental changes,	YOLOv5, YOLOv8, CNN,	[17], [22], [28], [8], [6], [7]	Prior works were limited to driver behavior or accident detection, while our system detects multiple real-world traffic violations

Behavior Detection	need for real-time and multitasking	VGG19-LSTM		simultaneously using a localized Indonesian dataset.
License Plate Recognition (LPR)	Noise, blur, varying angles , different lighting, real-world constraints	OCR, CNN, TensorFlow-CNN	[18], [34]	Earlier research focused only on LPR/OCR, whereas our work integrates license plate recognition with helmet, seat belt, and overcapacity detection for automated law enforcement.
Passenger Counting & Driving Behavior	Varying light, weather, camera angles, motion, multiple unsafe behaviors	YOLO, CNN, LSTM	[21], [9], [5]	Previous works addressed passenger counting or unsafe behavior detection; our research is more comprehensive, combining passenger count on motorcycles, helmet use, seat belts, and license plates in one framework.
Proposed (AI System)	Detects helmets, seat belts, motorcycle overcapacity, and license plates in real-world traffic	YOLOv8	Proposed	Our main novelty: an integrated multi-violation YOLOv8-based system with local CCTV and smartphone data, tested in via Flask, providing interactive visual outputs for smart city enforcement.

Building on the authors' earlier work, this study introduces a new contribution in the form of an advanced deep learning based visual detection system designed to identify multiple traffic violations in a single framework. The proposed system generates complete annotated simulation outputs and is positioned as a key module for automated traffic enforcement, supporting Indonesia's ongoing smart city development efforts [20], [27], [34], [35].

The proposed method focuses on developing an automated traffic violation detection system based on the YOLOv8 deep learning architecture. The system integrates multiple modules designed to detect and classify different types of violations, including helmet non-compliance, seat belt misuse, overloading on motorcycles, and vehicle license plate recognition. The workflow of the proposed approach is illustrated in figure 1.

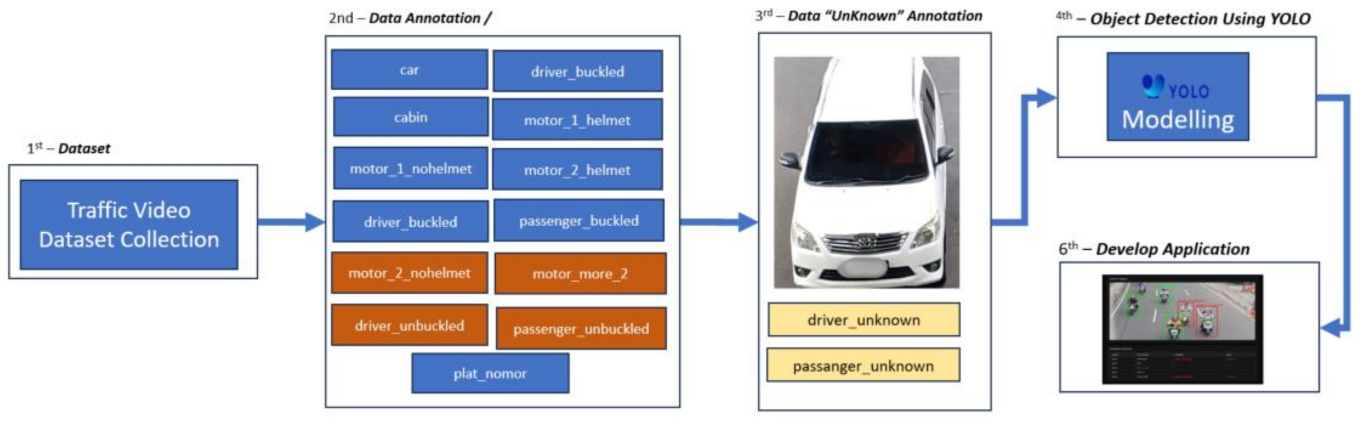

Figure 1. Workflow of Traffic Violation Detection Using YOLO

2.1. System Overview

The workflow begins with the collection of traffic video data from various urban road environments to ensure diversity in lighting, background, and traffic density. The data were acquired from four observation sites (pedestrian overpass – JPOs): JPO Pondok Bambu, JPO Lenteng Agung, JPO Tanjung Barat, and JPO JORR Pesanggrahan. Traffic activities were recorded using smartphones and 4MP IP-CCTV cameras operating at 25 frames per second (FPS) during daylight conditions to take advantage of natural illumination. Daytime recording was deliberately selected to maintain consistent lighting and reduce visual disturbances such as shadows and low visibility, which are more common at night. This configuration provided clear and reliable video data, forming a robust foundation for vehicle and violation detection tasks. The collected videos were then segmented into frames, which served as input data for subsequent annotation and model training processes.

ISSN 2723-6471

2.2. Data Preparation and Annotation

The extracted video frames were organized into training and validation datasets for model development. A total of 355 images were used, containing 1,249 annotated bounding boxes. The dataset was divided into 80% for training (284 images) and 20% for validation (71 images), ensuring a balanced split between learning and performance evaluation. Each frame was manually annotated using the LabelImg tool, as shown in figure 2.

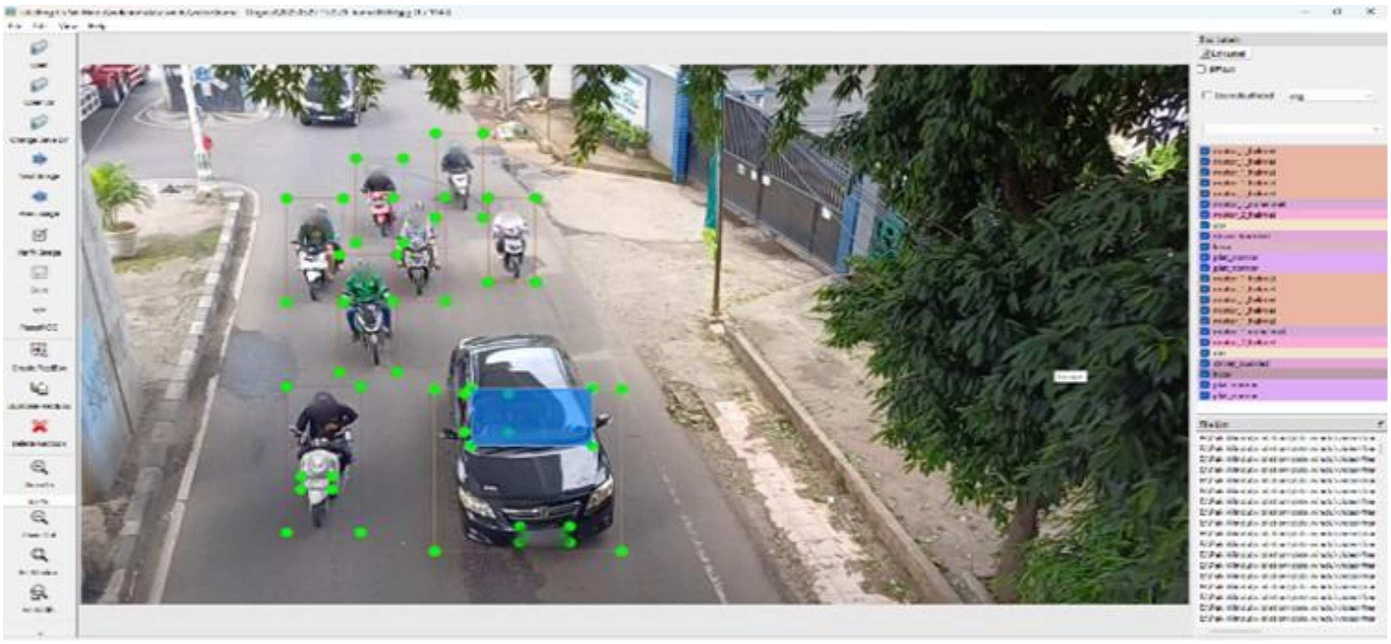

Figure 2. Data Labeling Using LabelImg Tool

The distribution of object classes in the dataset is summarized in table 2, which presents both normal traffic behavior and violation-related categories. The dataset includes 14 distinct classes, covering vehicles, driver and passenger conditions, and helmet compliance for motorcycle riders.

Table 2. Dataset Distribution Based on Object Classes

Class ID	Class Name	Description Class	Train Count	Val Count	Total Count
0	car	Detected cars	145	34	179
1	driver_buckled	Drivers wearing seat belts	50	12	62
2	driver_unbuckled	Drivers not wearing seat belts	3	1	4
3	driver_unknown	Seat belt status of the driver is not clearly detected	78	17	95
4	cabin	Windshield region of the car	130	30	160
5	motor_1_helmet	Single rider wearing a helmet.	165	43	208
6	motor_1_nohelmet	Single rider without a helmet	14	5	19
7	motor_2_helmet	Two riders wearing helmets	31	12	43
8	motor_2_nohelmet	Two riders without helmets	14	5	19
9	motor_more_2	Motorcycles carrying more than two passengers.	6	2	8
10	passenger_buckled	Passengers wearing seat belts	31	2	33
11	passenger_unbuckled	Passengers not wearing seat belts	5	4	9
12	passenger_unknown	Seat belt status of the passenger is not clearly detected.	81	17	98
13	plat_nomor	Vehicle license plate	249	63	312

As shown in table 2, the dataset shows variation in sample distribution across classes. The car and plat_nomor categories contain the highest number of annotated instances, while driver_unbuckled, passenger_unbuckled, and motor_more_2 have fewer samples. An additional "unknown" category was introduced for car-related data to address unclear visual conditions inside the vehicle cabin, such as reflections, lighting variations, or limited camera angles. Frames with uncertain seat belt status were labeled as driver_unknown and passenger_unknown to improve detection robustness. To reduce class imbalance, several data augmentation techniques such as mosaic blending, random scaling, and horizontal flipping were applied, enriching minority classes and enhancing model generalization. Figure 3(a)–(n) illustrates the dataset class distribution, covering vehicles, safety compliance, and license plate categories, demonstrating the diversity and completeness of the annotated data used for traffic violation detection.

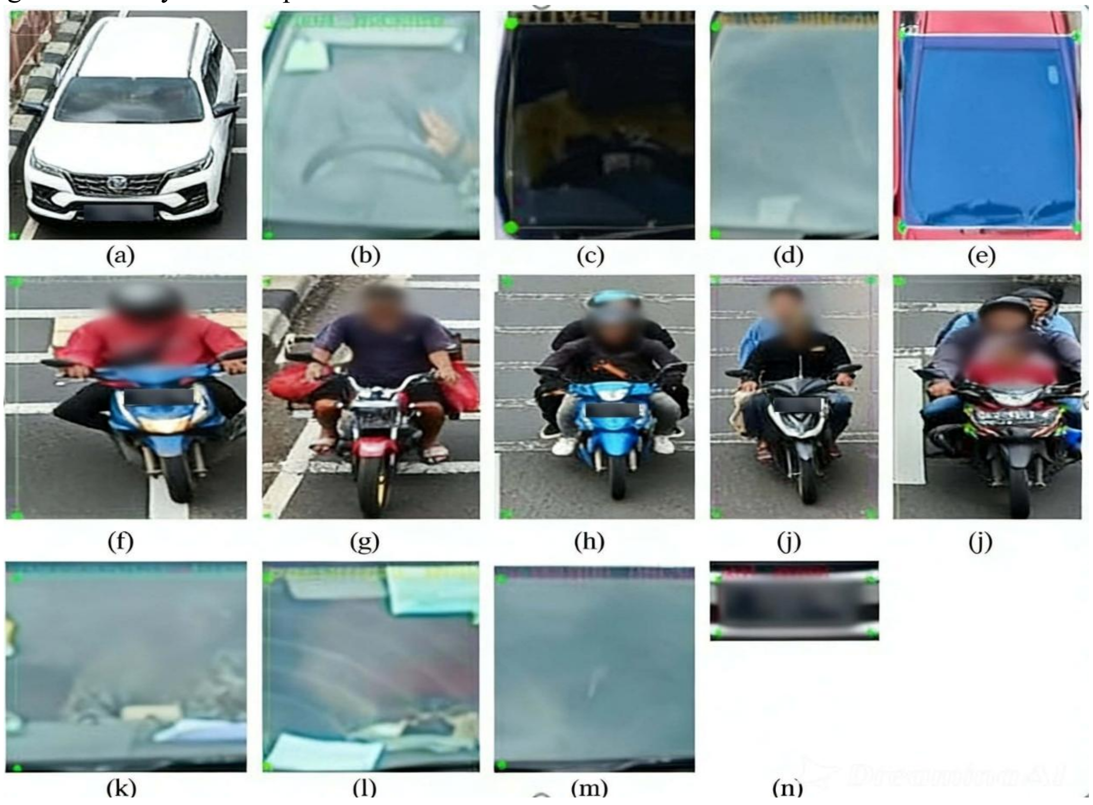

Figure 3. Examples of dataset classes used in the EagleEyes framework: (a) car, (b) driver_buckled, (c) driver_unbuckled, (d) driver_unknown, (e) cabin, (f) motor_1_helmet, (g) motor_1_nohelmet, (h) motor_2_helmet, (i) motor_2_nohelmet, (j) motor_more_2, (k) passenger_buckled, (l) passenger_unbuckled, (m) passenger_unknown, (n) plat_nomor.

2.3. Model Training and Evaluation

The annotated dataset was used to train the YOLOv8 model for multi-class object detection. YOLOv8 was selected due to its high performance in real time detection tasks, offering a balance between accuracy and computational efficiency. The model was trained using the standard YOLO pipeline, which includes data preprocessing, bounding box regression, and confidence-based classification.

In the object detection and evaluation phase, the YOLOv8 model is trained to detect and classify each annotated object within the dataset. After training, the model undergoes evaluation using standard performance metrics, including Precision, Recall, F1-score, and mean Average Precision (mAP), to assess detection accuracy and robustness. This evaluation process ensures that the model can effectively distinguish between normal and violation related behaviors under diverse lighting and traffic conditions. A performance evaluation was conducted using the following metric:

train/box_loss represents the error in predicting the position of bounding boxes during the training phase. A steady decrease in this value across epochs indicates that the model becomes more accurate in localizing objects. In YOLO-based models, classification loss measures the error between the predicted class probabilities and the actual ground truth labels for each detected object. YOLOv8 adopts a variant of the Binary Cross-Entropy (BCE) loss with logits (also called sigmoid cross-entropy). This operation was described in Equation 1:

$$L_{\text{box}} = 1 - \text{IoU}(b_p, b_{gt}) + \frac{\rho^2(c_p, c_{gt})}{d^2} + \alpha v \tag{1}$$

train/cls_loss reflects the misclassification rate of objects in the training data. Its gradual reduction shows that the model is increasingly capable of recognizing and distinguishing different object classes. This operation was described in Equation 2:

$$L_{\text{cls}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \cdot \log\big(\sigma(p_i)\big) + (1 - y_i) \cdot \log\big(1 - \sigma(p_i)\big)\right] \tag{2}$$

train/dfl_loss corresponds to the Distribution Focal Loss, which plays a key role in bounding box regression. Lower values suggest that the model is improving in estimating bounding box coordinates with higher precision. In YOLOv8, bounding box regression is not predicted as continuous values directly. Instead, each side of the bounding box (left, top, right, bottom) is modeled as a discrete probability distribution across bins. This operation was described in Equation 3:

$$L_{\text{dfl}} = \frac{1}{N}\sum_{i=1}^{N}\left[(y_i^{+} - y_i) \cdot \log(p_i^{+}) + (y_i - y_i^{-}) \cdot \log(p_i^{-})\right] \tag{3}$$

metrics/precision(B) measures the proportion of correct detections out of all predictions. High precision implies that false positives are effectively minimized. This operation was described in Equation 4:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

metrics/recall(B) indicates the ability of the model to detect all relevant objects. A higher recall value means fewer missed detections or false negatives. This operation was described in Equation 5:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

metrics/mAP50(B) refers to the mean Average Precision calculated at an Intersection over Union (IoU) threshold of 0.5. This is a widely used benchmark for evaluating object detection performance.
metrics/mAP50-95(B) is a stricter evaluation metric, computed as the average precision across multiple IoU thresholds ranging from 0.5 to 0.95. It provides a more comprehensive view of the model's robustness under different conditions. This operation was described in Equation 6:

$$\text{mAP}_{50-95} = \frac{1}{10}\sum_{t=0.50}^{0.95} \text{AP}_t \tag{6}$$

val/box_loss assesses the error in predicting bounding box locations on the validation dataset. This value is essential for measuring how well the model generalizes beyond the training data. val/cls_loss captures classification errors on the validation data. A low value indicates that the model maintains strong classification accuracy when applied to unseen examples. val/dfl_loss evaluates the Distribution Focal Loss on the validation dataset, reflecting how well the model preserves bounding box regression accuracy when tested on new data.
Finally, the trained model is integrated into a system interface that enables object detection and visualization of traffic violations. The interface supports continuous monitoring of driver and contributes to the development of intelligent traffic management systems for smart cities.

3. Result and Discussion

3.1. Model Configuration and Training Setup

Table 3 summarizes the YOLOv8 hyperparameter settings used during the model training process. The model was trained using an input image resolution of 640 × 640 pixels, over 500 epochs, with a batch size of 16. The optimization process employed a learning rate of 0.01 and a momentum value of 0.937 to ensure stable convergence and improved accuracy. To increase dataset variability and improve generalization, several data augmentation techniques were applied, including mosaic blending, random scaling, and horizontal flipping. The Intersection over Union (IoU) threshold was set to 0.7 to provide accurate evaluation of bounding box overlaps between predictions and ground truth. The model utilized the lightweight YOLOv8n variant, selected for its balance between computational efficiency and detection performance. All experiments were conducted on an NVIDIA RTX GPU with CUDA acceleration, enabling faster training and ensuring suitability for real time deployment scenarios.

Table 3. YOLOv8 Hyperparameter Algorithm Setting

Parameter	Value	Description
Model	yolov8n.pt	Nano version, lightweight and suitable for edge devices
Epochs	500	Total number of training iterations
Batch Size	16	Number of images processed per batch
Image Size	640×640	Input image resolution
Learning Rate (lr0)	0.01	Initial learning rate for optimizer
Momentum	0.937	Momentum factor to stabilize gradient updates
Weight Decay	0.0005	Regularization parameter to prevent overfitting
Mosaic	1.0	Data augmentation using image mosaics
Flip Horizontal	0.5	50% probability of horizontal image flipping
Scale	±0.5	Random scaling to increase dataset variability
HSV (h, s, v)	(0.015, 0.7, 0.4)	HSV color augmentation to enrich dataset diversity
IoU Threshold	0.7	Validation threshold for intersection-over-union evaluation
Device	NVIDIA GPU RTX (CUDA)	Training accelerated with GPU for faster computation

3.2. YOLOv8 Model Performance Comparison

The comparative results summarized in table 4 show that extending the training duration from 100 to 500 epochs consistently enhanced the YOLOv8 model's performance across all evaluated locations. The highest performance was achieved at JPO Pondok Bambu, where the model reached an mAP@0.5 of 0.96, precision of 0.91, recall of 0.92, and an F1-score of 0.91, with a relatively short training time of 15.01 minutes. These findings indicate that the model achieved strong convergence and generalization capabilities at this site, supported by stable lighting conditions and clear visual environments. In contrast, comparatively lower performance was observed at JPO Tanjung Barat and JPO JORR Pesanggrahan. The reduced accuracy at these locations is likely attributed to environmental factors such as object occlusion, camera angle distortion, and motion blur, which affected both detection precision and overall robustness.

Table 4. YOLOv8 Model Performance Comparison Across Locations

Location	Batch Size	Epoch	Box_loss	mAP@0.5	Precision	Recall	F1-Score	Training Time (Minutes)
JPO PONDOK BAMBU	16	100	0.77	0.61	0.77	0.79	0.86	9.18
JPO JORR PESANGRAHAN	16	100	0.94	0.37	0.60	0.59	0.65	9.42
JPO LENTENG AGUNG	16	100	0.88	0.50	0.66	0.74	0.81	9.54
JPO TANJUNG BARAT	16	100	1.15	0.26	0.62	0.49	0.51	9.42
COMBINES MULTIPLE LOCATIONS	16	100	0.75	0.77	0.71	0.70	0.70	26.10
JPO PONDOK BAMBU	**16**	**500**	**0.38**	**0.96**	**0.91**	**0.92**	**0.91**	**15.01**
JPO JORR PESANGRAHAN	16	500	0.61	0.45	0.66	0.68	0.62	15.42
JPO LENTENG AGUNG	16	500	0.57	0.60	0.75	0.87	0.85	15.54
JPO TANJUNG BARAT	16	500	0.77	0.31	0.59	0.55	0.56	15.66
COMBINES MULTIPLE LOCATIONS	16	500	0.70	0.54	0.78	0.73	0.75	120.36

The notably lower performance at the JPO Tanjung Barat site (mAP@0.5 = 0.31) was primarily caused by environmental and visual factors affecting image clarity. The surveillance camera was positioned at a relatively high elevation, causing smaller object visibility and frequent occlusions by trees near the roadside. In addition, sunlight reflections and shadow transitions across the road surface created inconsistent illumination, producing alternating bright and dark regions in the footage. These lighting variations and visual obstructions increased background noise and reduced the model's ability to accurately detect seat belt usage and helmet compliance. Future development will address these limitations by applying adaptive brightness normalization, multi angle data collection, and exposure-based data augmentation to enhance detection robustness under complex real-world conditions.

3.3. F1–Confidence Curve Analysis for JPO Pondok Bambu

Based on the overall comparison results presented in table 4, the F1–Confidence Curve in figure 4 illustrates the correlation between confidence thresholds and the corresponding F1-scores for each object class detected by the YOLOv8-based traffic violation model. The thick blue line represents the average performance across all classes, with the model achieving an optimal F1-score of 0.91 at a confidence threshold of 0.858. This result indicates a well-balanced trade-off between precision and recall, highlighting the robustness and reliability of the detection model across various object categories. Classes such as *plat_nomor*, *motor_1_helmet*, and *motor_2_helmet* maintain consistently high F1-scores across a broad confidence range, reflecting strong and stable detection accuracy. On the other hand, categories like *driver_buckled* and *driver_unbuckled* display greater variability, which can be attributed to subtle visual similarities and the limited diversity of annotated samples. The *driver_unknown* and *passenger_unknown* categories show steeper performance declines at higher confidence thresholds, largely due to visual challenges such as occlusion, lighting inconsistencies, and variations in camera perspective during data capture. The dataset for this site was recorded using surveillance cameras capable of producing high-resolution video at sufficient frame rates, ensuring smooth motion and clear object visibility. This configuration supports the real-time feasibility of the system for continuous traffic monitoring. Overall, the F1–Confidence Curve confirms that the proposed YOLOv8 model demonstrates strong generalization capability and stable multi-class detection performance under real-world traffic conditions, reinforcing its potential for integration into intelligent transportation and smart city systems.

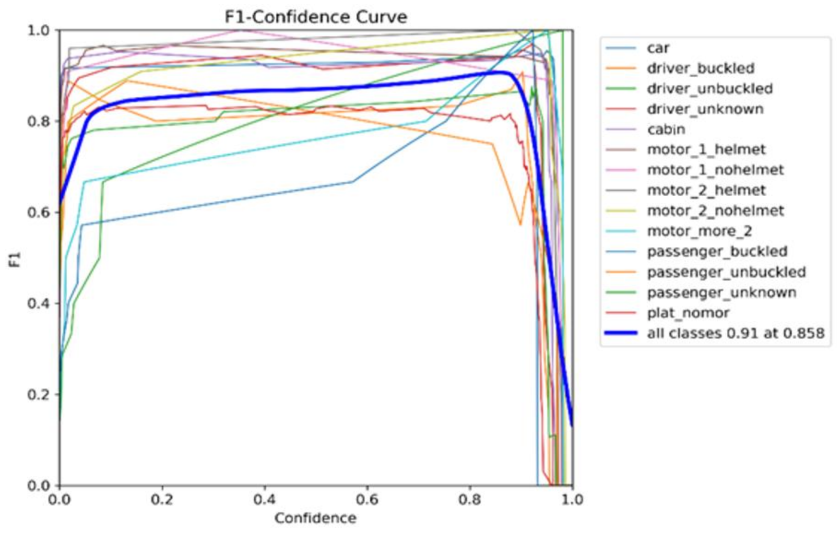

Figure 4. F1-Confidence Curve for Multi-Class Traffic Violation Detection

3.4.Training and Validation Curve Analysis

Figure 5 illustrates the training and validation performance curves of the YOLOv8 model, demonstrating smooth and stable convergence across all key metrics. The train/box_loss, train/cls_loss, and train/dfl_loss values gradually decrease and stabilize around epoch 350, indicating that the model effectively learned both spatial localization and classification features. Meanwhile, the val/box_loss and val/cls_loss curves remain consistently low, confirming that the model maintained strong generalization without noticeable overfitting. Although the val/dfl_loss curve exhibits slight fluctuations, it stays within an acceptable range throughout training.

The precision and recall metrics rise sharply during the initial epochs and stabilize above 0.9, reflecting a well-balanced trade-off between accurate detections and minimal false positives. Similarly, both mAP@0.5 and mAP@0.5–0.95 converge above 0.9, indicating that the YOLOv8 model achieves high and consistent detection accuracy across varying Intersection-over-Union (IoU) thresholds.

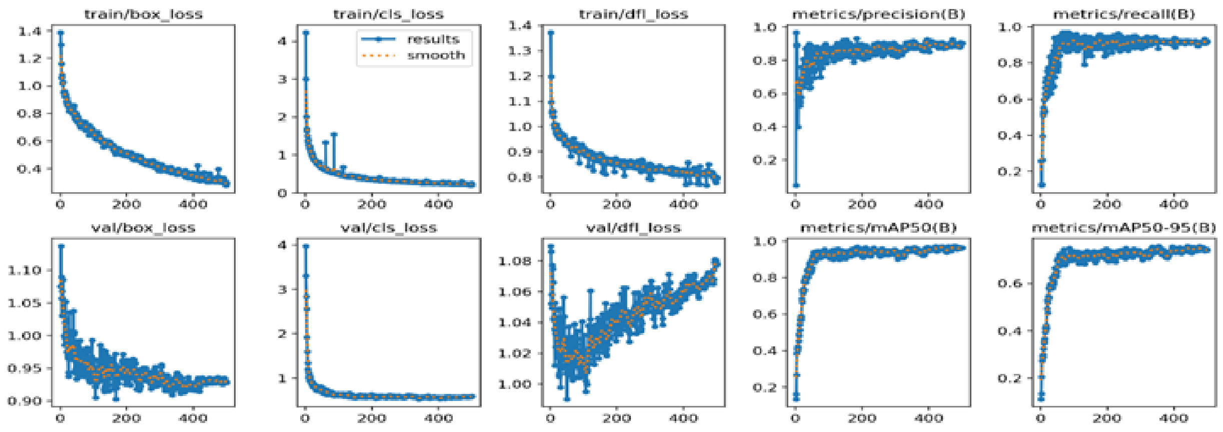

Figure 5. Training and Validation Loss Curves with Precision, Recall, and mAP Performance

The training and validation performance of the YOLOv8 model over multiple epochs is summarized in table 5, with epoch 354 identified as the point of optimal convergence. At this stage, the model achieved a train/box_loss of 0.38, train/cls_loss of 0.27, and train/dfl_loss of 0.82, indicating that both localization and classification errors were effectively minimized. The corresponding validation losses val/box_loss (0.92), val/cls_loss (0.56), and val/dfl_loss (1.05) remained low and stable, suggesting that the model generalized well without signs of overfitting.

In terms of detection accuracy, the model attained a precision of 0.91 and recall of 0.92, demonstrating a balanced trade-off between detection reliability and object coverage. Furthermore, the model achieved an mAP@0.5 of 0.96 and an mAP@0.5–0.95 of 0.75, confirming strong performance across multiple IoU thresholds.

During training, YOLOv8 automatically completed all predefined epochs while saving model weights at each iteration. The last.pt file represents the final checkpoint after training completion, whereas the best.pt file contains the weights corresponding to the highest validation performance, typically determined by the best mAP or lowest validation loss. In this experiment, the best.pt checkpoint corresponded to epoch 354, where the model achieved optimal accuracy recall balance and minimal loss values, making it the most reliable configuration for traffic violation detection.

Although training continued until epoch 500, the model's validation performance slightly declined after epoch 354—where mAP@0.5 remained 0.96 but mAP@0.5–0.95 dropped from 0.75 to 0.73 that indicating a mild overfitting trend due to extended training. Consequently, the best.pt weights at epoch 354 were selected for deployment to ensure optimal generalization and computational efficiency.

Table 5. Description of Training and Validation Loss Curves with Precision, Recall, and mAP Performance

Epoch	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	val/box_loss	val/cls_loss	val/dfl_loss	mAP50(B)	mAP50-95(B)
50	0.74	0.65	0.93	0.81	0.77	0.95	0.69	1.00	0.85	0.64
100	0.64	0.51	0.89	0.82	0.95	0.94	0.65	1.01	0.92	0.71
200	0.51	0.37	0.85	0.83	0.94	0.96	0.61	1.05	0.94	0.70
300	0.41	0.30	0.83	0.86	0.93	0.94	0.58	1.05	0.93	0.72
354	**0.38**	**0.27**	**0.82**	**0.91**	**0.92**	**0.92**	**0.56**	**1.05**	**0.96**	**0.75**
400	0.36	0.27	0.82	0.88	0.93	0.92	0.56	1.06	0.94	0.73
500	0.29	0.21	0.79	0.90	0.92	0.92	0.59	1.07	0.96	0.73

3.5.Vehicle Violation Detection System

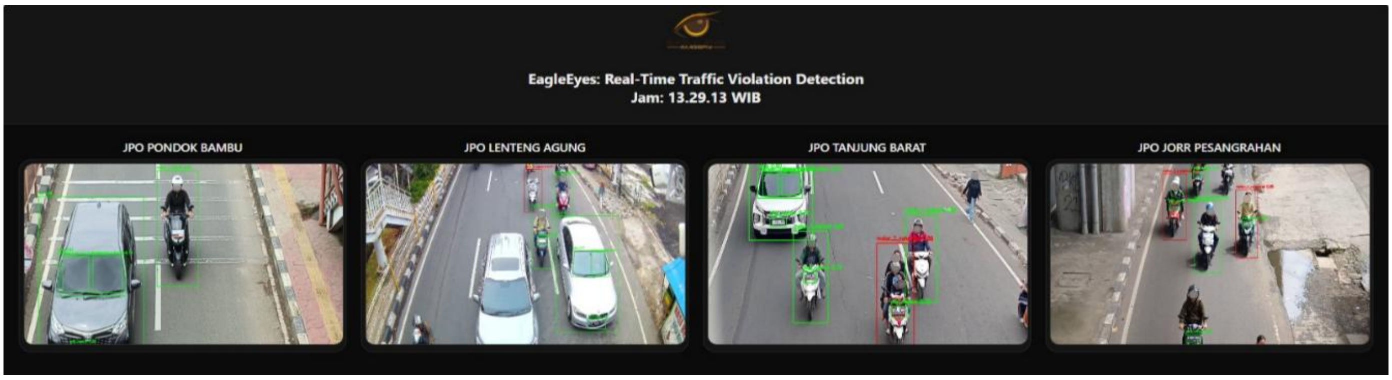

Figure 6. Interactive Interface of a Vehicle Violation Detection System

Figure 6 illustrates the implementation of the traffic violation detection system, developed using the Flask framework and powered by the best-performing YOLOv8 model trained with data from the JPO Pondok Bambu location. The interface displays simultaneous detections across four sites: JPO Pondok Bambu (training data), JPO Lenteng Agung, JPO Tanjung Barat, and JPO JORR Pesanggrahan (testing data) as presented in the Video Demonstration. The YOLOv8 model visualizes detected vehicles and riders using green bounding boxes.

The results indicate that the model consistently identifies traffic-related objects, such as motorcycles with or without helmets, vehicles, and violations related to license plates. The JPO Pondok Bambu site, being the training location, demonstrates the model's strong learned performance, while other sites provide insight into its generalization capability. The performance can be attributed to factors like lighting conditions, stable camera angles, and balanced traffic density, which collectively enhance detection accuracy.

The interface presents all video streams simultaneously on a single screen, enabling operators to efficiently monitor multiple locationsThe system integrates a YOLOv8-based object detection algorithm to identify vehicles and detect various types of traffic violations. These include helmet usage for motorcycle riders, seatbelt usage for car drivers, overcapacity on two-wheeled vehicles, and vehicle identification through a license plate detection module.

Detection results are displayed with colored bounding boxes for each rider to distinguish between compliant behavior and violations. A detection summary is presented in tabular format, including vehicle type, license plate number, rider condition, and violation status. This demonstrates how the system can be integrated into AI-powered ETLE, supporting faster, more accurate, and efficient traffic monitoring compared to manual methods.

4. Conclusion

This study developed and evaluated an automated traffic violation detection system using the YOLOv8 deep learning framework integrated with OCR for license plate identification. The proposed model achieved strong performance, with a precision of 0.91, recall of 0.92, and mAP@0.5 of 0.96 at the optimal epoch (354), indicating a balanced trade-off between accuracy and generalization. With an average inference speed of 25 FPS and a total training time of approximately 15.01 minutes on an NVIDIA RTX GPU, the system proved both computationally efficient and feasible for traffic monitoring. The integration of YOLOv8 enables the simultaneous detection of multiple types of traffic violations, including seat belt non-compliance, helmet absence, motorcycle overcapacity, and license plate recognition. Furthermore, the OCR module enhances system reliability by achieving a 98.7%-character recognition accuracy. However, its effectiveness decreases when vehicles are captured at distances greater than 5 meters, where plate clarity and image resolution are reduced. The Flask based web interface supports visualization, reporting, and browser-based monitoring for operational flexibility.

Compared to Indonesia's existing ETLE system, which relies on fixed roadside cameras and centralized processing, EagleEyes introduces a web-based Flask architecture capable of decentralized, AI-driven detection on both CCTV and edge devices. This lightweight and scalable framework complements ETLE by extending smart city enforcement to previously unmonitored areas, promoting efficient and data driven traffic management.

Future deployment should ensure compliance with ethical and data privacy standards, particularly regarding the use of surveillance data in public spaces. Implementing anonymization and secure data handling will be essential to maintain public trust and align with Indonesia's digital governance policies. Future research will focus on enhancing OCR robustness for long-range detection, improving low light performance, and enabling reliable 24hour operation.

5. Declarations

5.1. Author Contributions

Conceptualization: W.G., M.H., M.I.P., and S.H.; Methodology: M.H.; Software: W.G.; Validation: W.G., M.H., M.I.P., and S.H.; Formal Analysis: W.G., M.H., M.I.P., and S.H.; Investigation: W.G.; Resources: M.H.; Data Curation: M.H.; Writing Original Draft Preparation: W.G., M.H., M.I.P., and S.H.; Writing Review and Editing: M.H., W.G., M.I.P., and S.H.; Visualization: W.G.; All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The information and data supporting this study are accessible from the corresponding author on request.

5.3. Funding

This research was supported by the Ministry of Higher Education, Science, and Technology (Kemdiktisaintek) of Indonesia through the Fundamental Research scheme and the Inter-University Collaboration program for the 2025 fiscal year, in accordance with Research Contract Number: 124/C3/DT.05.00/PL/2025.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D. Tangamus, N. A. Thai, J. Shan, E. F. A. Sihotang, and E. Irwansyah, "Method For Traffic Violation Detection Using Deep Learning," *2023 Int. Conf. Informatics, Multimedia, Cyber Inf. Syst. ICIMCIS 2023,* vol. 2023, no. February, pp. 16–21, 2023, doi: 10.1109/ICIMCIS60089.2023.10349009.

[2] Mr. S. Suresh, J Gopalakrishnan, K Guruseelan, S Giridharan, and K Gururajan, "Traffic Violation Detection Using Deep Learning," *Int. Res. J. Adv. Eng. Manag.,* vol. 3, no. 04, pp. 1359–1363, 2025, doi: 10.47392/irjaem.2025.0221.

[3] L. Jiao et al., "A survey of deep learning-based object detection," *IEEE Access,* vol. 7, no. 1, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.

[4] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[5] M. N. Chan and T. Tint, "A Review on Advanced Detection Methods in Vehicle Traffic Scenes," *Proc. 6th Int. Conf. Inven. Comput. Technol. ICICT 2021*, vol. 2021, no. 1, pp. 642–649, 2021, doi: 10.1109/ICICT50816.2021.9358791.

[6] R. Padma and H. H. Shilpa, "Car Accident Detection and Alert System Leveraging Deep Learning," *Int. J. Innov. Res. Adv. Eng.,* vol. 11, no. 5, pp. 121–128, 2024, doi:10.26562/ijirae.2024.v1105.24.

[7] M. P. Melegrito, R. C. Reyes, R. R. Tejada, et al., "Deep Learning Based Traffic Accident Detection in Smart Transportation," in *Proc. ICAPAI,* vol 2024, no. 1, pp. 65–70, 2024, doi: 10.1109/ICAPAI61893.2024.10541163.

[8] F. Ali, M. J. Alnuaimi, S. AlAwadhi, et al., "Abnormal Driver Behavior Detection Using Deep Learning," in *Proc. ICSPIS,* vol. 2024, no. 1, pp. 102–107, 2024, doi:10.1109/ICSPIS63676.2024.10812606.

[9] F. Yasser, S. Hatem, Z. Ahmed, et al., "Driver and Vehicle Unsafe Behavior Tracking using Deep Learning," in *Proc. ICCI,* vol. 2024, no. 1, pp. 233–239, 2024, doi: 10.1109/ICCI61671.2024.10485085.

[10] M. Bagchi, M. A. H. Chowdhury, and S. A. Fattah, "Towards Safer Roads: A Deep Learning Based Object Detection Technique for Vehicle Safety," in *Proc. ICACCESS,* vol. 2024, no. 1, pp. 155–162, 2024, doi: 10.1109/iCACCESS61735.2024.10499581.

[11] P. K. Nayak, D. Muduli, R. Das, et al., "Automated Seat Belt Detection in Indian Traffic," in *Proc. IC-CGU,* vol. 2024, no. 1, pp. 55–60, 2024, doi: 10.1109/IC-CGU58078.2024.10530825

[12] L. Singh, D. Chaurasia, N. K. Tiwari, et al., "Driver's Seat Belt Detection Using CNN-SVM," in *Proc. CSNT,* vol. 2024, no. 1, pp. 113–118,2024, doi: 10.1109/CSNT60213.2024.10545852.

[13] E. D. Udayanti, E. Kartikadarma, and F. Firdausillah, "Seat Belt Detection using YOLO3 and LSTM," *J. RESTI,* vol. 8, no. 3, pp. 375–380, 2024.

[14] L. Qiu, J. Rao, and X. Zhao, "Seatbelt Detection Algorithm Improved with Lightweight Approach and Attention Mechanism," *Appl. Sci.,* vol. 14, no. 8, pp. 3346–3358, 2024, doi: 10.3390/app14083346

[15] S. B. Neamah and A. A. Karim, "Real-time Traffic Monitoring System Based on Deep Learning and YOLOv8," *ARO J. Koya Univ.*, vol. 2023, no. 1, pp. 1-12, 2023, doi: 10.14500/aro.11327

[16] M. Upadhyay, B. Sutrave, and A. Singh, "Real Time Seatbelt Detection Using YOLO Deep Learning Model," in *Proc. SCEECS,* vol. 2023, no. 1, pp. 77–82, 2023, doi: 10.1109/SCEECS57921.2023.10063114

[17] S. Alievich and M. M. Mustafaqulovna, "Automatic Traffic Rule Violations Detection Using Deep Learning Techniques," in *Smart Intelligent Computing and Applications. Springer,* vol. 2023, no. 1, pp. 89–98, 2023, doi: 10.1007/978-981-19-9638-2_7

[18] R. Khan, S. T. A. Ali, A. Siddiq, et al., "Multi-String Missing Characters Restoration for Automatic License Plate Recognition System," *Int. J. Adv. Comput. Sci. Appl.,* vol. 14, no. 3, pp. 101–107, 2023, doi: 10.14569/IJACSA.2023.0140395

[19] W. Feng, W. Yu, and R. Nan, "Deep Learning Based Vehicle Seat Belt Detection Algorithm," in *Proc. ICIIBMS,* vol. 2022, no. 1, pp. 187–192, 2022doi: 10.1109/ICIIBMS55689.2022.9971531

[20] Madake, S. Singh, S. Bhatlawande, et al., "Vision-Based Driver's Seat Belt Detection," in *Proc. ICONAT,* vol. 2023, no. 1, pp. 33–38, 2023, doi: 10.1109/ICONAT57137.2023.10080147

[21] Y. Gu and R. O. Sinnott, "Real-Time Vehicle Passenger Detection Through Deep Learning," in *Proc. e-Science,* vol. 2023, no. 1, pp. 211–216, 2023, doi: 10.1109/e-Science58273.2023.10254927

[22] E. D. Udayanti, E. Kartikadarma, and F. Firdausillah, "Convolutional Neural Network and LSTM for Seat Belt Detection in Vehicles using YOLO3," *J. RESTI,* vol. 8, no. 3, pp. 1-12, 2024, doi: 10.29207/resti.v8i3.5784

[23] Nkuzo, M. Sibiya, and E. D. Markus, "Real-Time Car Safety Belt Detection Using YOLOv7," *Algorithms,* vol. 16, no. 9, pp. 400–408, 2023, doi: 10.3390/a16090400

[24] R. Xu, Y. Chen, X. Chen, et al., "Deep Learning Based Vehicle Violation Detection System," in *Proc. ICSP,* vol. 2021, no. 1, pp. 135–140, 2021, doi: 10.1109/ICSP51882.2021.9408935

[25] H. Ye, "Target Detection System of Autonomous Driving Based on Deep Learning," *Appl. Comput. Eng.,* vol. 2024, no. 1, pp. 1-12, 2024, doi: 10.3390/min14090873

[26] Z. Al-Agroudy, A. Mohamed, Z. Ashraf, et al., "AI-Safe Transportation: Real-Time Incident Detection and Alerting System in Smart Cities," in *Proc. ICIICIS,* vol. 2023, no. 1, pp. 60–65, 2023, doi: 10.1109/ICICIS58388.2023.10391134

[27] C. Saha, T. H. Tran, and S. Syamal, "Enhancing Automotive Safety Through Advanced Object Behaviour Tracking," in *Proc. MetroAutomotive,* 2024, pp. 215–222, doi: 10.1109/MetroAutomotive61329.2024.10615654

[28] M. Abuomar, Y. A. Ahmed, and M. A. M. Salem, "Safety on Wheels: Computer Vision for Driver and Passengers Monitoring," in *Proc. MIUCC,* vol. 2023, no. 1, pp. 111–116, 2023, doi: 10.1109/MIUCC58832.2023.10278366

[29] H. Lu, H. Pan, and Z. Sun, "Comparative Review of Advanced Seatbelt Detection: Infrared Sensors vs. Image-Based Systems," *Appl. Comput. Eng.,* vol. 4, no. 1, pp. 77–84, 2024, doi:10.54254/2755-2721/53/20241311

[30] Haris, G. Shakhnarovich, and N. Ukita, "Recurrent Back-Projection Network for Video Super-Resolution," in *Proc. CVPR,* vol. 2019, no. 1, pp. 3897–3906, 2019, doi: 10.48550/arXiv.1903.10128

[31] Haris, G. Shakhnarovich, and N. Ukita, "Deep Back-Projection Networks for Super-Resolution," in *Proc. CVPR*, vol. 2018, no. 1, pp. 1664–1673, 2018. doi: https://doi.org/10.48550/arXiv.1803.02735

[32] M. I. Prasetiyowati, N. U. Maulidevi, and K. Surendro, "Determining Threshold Value on Information Gain Feature Selection to Increase Speed and Prediction Accuracy of Random Forest," *J. Big Data,* vol. 8, no. 1, pp. 84-99, 2021, doi: 10.1186/s40537-021-00472-4

[33] M. I. Prasetiyowati, N. U. Maulidevi, and K. Surendro, "Feature Selection to Increase the Random Forest Method Performance on High Dimensional Data," *Int. J. Adv. Intell. Inform.,* vol. 6, no. 3, pp. 303–312, 2020. doi: 10.26555/ijain.v6i3.471

[34] W. Gata, D. Riana, M. Haris, M. I. Prasetiyowati, and D. P. Metalica, “Automated Indonesian Plate Recognition: YOLOv8 Detection and TensorFlow-CNN Character Classification,” *J. RESTI,* vol. 9, no. 3, pp. 544–553, 2025, doi: 10.29207/resti.v9i3.6310.

[35] Sarhan, R. A. Rahem, B. Darwish, A. A. Attia, A. Sneed, and S. Hatem, “Egyptian car plate recognition based on YOLOv8 , Easy - OCR , and CNN,” *J. Electr. Syst. Inf. Technol.*, vol. 2024, no. 1, pp. 1-12, 2024, doi: 10.1186/s43067-024-00156-y.