Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi
Vol.10 No.2 July 2025, P-ISSN : 2502-3470, E-ISSN : 2581-0367

Implementation of Car and Motorcycle Detection in "No Parking" Sign Areas Using Web-Based YOLOv8

Somantri[1*], Miya Kurnia[2], Sakti Aji Wijaya[3], Rizqy Mawardi[4], Gina Purnama Insany[5]

[12345]*Informatics Department, Universitas Nusa Putra, Sukabumi, Indonesia*

[1] somantri@nusaputra.ac.id (*)

[2,3,4,5][miya.kurnia_ti21, sakti.aji_ti21, rizqy.mawardi_ti21, gina.purnama]@nusaputra.ac.id

Received: 2025-05-30; Accepted: 2025-08-04; Published: 2025-08-16

Abstract— Illegal parking in no-parking zones often causes traffic and road transport disruptions, thus requiring a technology-based solution to detect such violations automatically. This study aims to develop a parking violation detection system for two-wheeled and four-wheeled vehicles using the YOLOv8 object detection model. The system development employs the prototype method, which allows for iterative creation and testing to achieve optimal results. The system is implemented on IoT devices (Raspberry Pi 4, webcam, buzzer) and integrated with a Laravel-based web dashboard for monitoring violations. The YOLOv8 model is trained on a dataset and evaluated using precision, recall, and mean Average Precision (mAP) metrics at Intersection over Union (IoU) thresholds of 50% (mAP50) and 50–95% (mAP50-95), as well as inference speed to assess real-time capability. Evaluation results show the model achieves an mAP50 of 96.3% with high precision, although recall for the motorcycle class is lower compared to the car class. The system is capable of providing real-time alerts via buzzer when a parking violation is detected and displaying violation data on the web dashboard. The YOLOv8-based parking violation detection system has been successfully implemented through prototype development, and the system operates as expected according to predefined specifications.

Keywords— Illegal Parking; Detection, Internet of Things (IoT); You Only Look Once; YOLOv8; CNN.

I. INTRODUCTION

Illegal parking practices are commonly carried out by individuals or groups who park their vehicles in un-authorized or unofficial areas, such as sidewalks, roads, parks, or green spaces [1]. Illegal on-street parking is a significant issue in Indonesia, largely due to the increasing number of vehicles, limited parking space, and the inefficiency of public transportation. Parking violations have a direct impact on traffic performance and often cause congestion, especially in areas with high activity levels. Although there is a prohibition in Law No. 22 of 2009 and PP No. 34 of 2006 on using roads and sidewalks as parking areas, violations are still rampant [2].

In Palabuhan Ratu, Sukabumi, Indonesia, the regulation regarding no-parking zones is outlined in Sukabumi City Regulation No. 5 of 2018, which references Law No. 22 of 2009. One of the critical points is Palabuhan Ratu Square, which often experiences traffic disruptions due to haphazard parking. Sukabumi Regency Regulation No. 2 of 2018 also regulates parking fees on public roads, but does not specifically limit the duration of parking.

YOLO excels in object detection because it can simultaneously predict bounding boxes and class probabilities in a single inference pass. This advantage enables fast, real-time object detection, making YOLO suitable for hardware-constrained environments [3]. YOLOv8 is the latest version of YOLO, which improves detection accuracy by optimizing its architecture and feature extraction. As a one-stage detector, it performs object localization and classification in a single pass, demonstrating strong generalization across diverse real-world scenarios [4].

This study aims to design an effective YOLOv8 model for detecting two-wheeled and four-wheeled vehicles in no-parking areas, integrate it with IoT devices to develop a tool for detecting illegal parking violations with a web-based output, and perform selection and adjustment of the YOLOv8 model to support the effectiveness of the IoT-based detection system.

The results of a literature review of several previous studies that were used as references by the author. Research [5] implemented a license plate detection system using the YOLOv8 architecture. Their approach focused on accurately detecting vehicle license plates with the support of deep learning, and the results showed high precision and recall. However, the research was limited to license plate recognition and did not address illegal parking behaviour or utilize IoT integration for real-time responses [5]. Researchers [6] investigated the performance of the YOLOv8 model for detecting driver drowsiness conditions. Their study used vehicle-based video analysis to recognize signs of driver fatigue. While their method effectively identified drowsy drivers using image features, it was not intended for vehicle or parking violation detection, nor did it offer real-time alert mechanisms or web-based monitoring[6]. Research [7] developed a system for vehicle detection and classification using the YOLOv7 algorithm. Their model was able to categorize different vehicle types on the road with good accuracy, demonstrating YOLO's capabilities for traffic scene analysis. However, the system was not designed to detect parking violations or integrate alert/IoT features for enforcement [7]. Research [8] presented a real-time traffic sign detection system in Indonesia using YOLOv11. This research successfully identified various traffic signs from live video feeds, aiming to enhance road safety. Despite its real-time

DOI : https://doi.org/10.25139/inform.v10i2.10408

performance, the study did not consider vehicle detection or illegal parking, nor did it utilize IoT or web-based solutions [8]. YOLOv8 for vehicle object tracking and counting [9]. Their system achieved high accuracy in tracking and enumerating vehicles in traffic images. The main limitation was that it only focused on counting vehicles and did not consider violation types, parking duration, or provide any IoT-driven warning or web monitoring functionality [9].

This research utilizes YOLOv8 for real-time detection of both cars and motorcycles, particularly in areas with "No Parking" signs, introducing a violation duration threshold that must be exceeded before an incident is registered. The system uniquely integrates IoT hardware. Raspberry Pi and a buzzer to provide immediate on-site warnings, while a web-based dashboard enables remote monitoring and management. This combination of high-accuracy detection, duration-based violation logic, and IoT integration with web services creates a practical and comprehensive solution for automated parking enforcement, bridging the gap between object detection research and actionable smart city traffic management.

II. Research Methodology

The theoretical basis serves as a conceptual basis in the development of the proposed system, while also explaining the theories and technologies that support this research. The following discussion encompasses the concepts of illegal parking, deep learning technology, the Internet of Things (IoT), the Convolutional Neural Network (CNN) method, the YOLOv8n algorithm, and the model performance evaluation technique employed in the automatic parking vehicle detection system.

1) Illegal Parking: Illegal parking occurs when vehicles stop in prohibited areas, such as sidewalks or within a 100-meter radius of an official no-parking zone. This problem arises due to limited space in urban areas, where the provision of parking spaces must compete with other land uses that have been determined [1].

2) Internet of Things (IoT): The Internet of Things (IoT) is a term that refers to a network of physical devices connected to the internet, enabling them to interact with each other and exchange data and information. More broadly, IoT is part of the digitalization process that connects the physical world with the digital world via the internet and other supporting technologies [10].

3) You Only Look Once (YOLOv8): Object detection is a machine learning capability that utilizes camera sensors to capture digital images [11]. YOLO (You Only Look Once) was first introduced by Joseph Redmon and his team in 2015, which is a popular deep learning architecture for real-time object detection [12]. YOLOv8 was launched on January 10, 2023. This version is the latest in the YOLO family, designed to detect objects more accurately and faster, making it ideal for real-time applications such as video surveillance systems and autonomous vehicles [13]. The efficiency of this model also allows implementation on low-spec hardware without sacrificing performance. YOLOv8 is an advanced object detection model designed to detect objects in images or videos with high speed and accuracy[14]. YOLOv8 aims to enhance performance in detecting various types of objects and situations [15].

4) Convolutional Neural Networks (CNNs): Convolutional Neural Networks (CNNs) are a type of deep learning algorithm specifically designed to process and analyze visual data, such as images and videos. CNNs effectively extract and interpret features from visual data through automated learning processes [16][17].

5) Laravel: Laravel is an open-source PHP framework designed to simplify web application development. It provides a wide range of built-in features, such as routing, middleware, Eloquent ORM, Blade, validation, authentication, and REST API support, that accelerate development without compromising quality or security. This study examines the linkage between Laravel and the Internet of Things (IoT) to develop a system that accurately and efficiently delivers car and motorcycle detection results via an API [18]. The successful implementation is expected to make a meaningful contribution to the development of robust, scalable, and integrated IoT applications.

Based on the theoretical basis described, it can be concluded that the development of an illegal parking detection system utilizing deep learning and IoT is supported by mature concepts and technologies. Starting from an understanding of illegal parking as an urban problem, a deep learning approach utilizing CNN and YOLOv8 architectures for real-time object detection is combined with IoT integration for data collection and exchange. Performance evaluation using a confusion matrix and mAP ensures system accuracy. Thus, this study has a strong theoretical basis for developing an effective automatic detection system to address illegal parking problems.

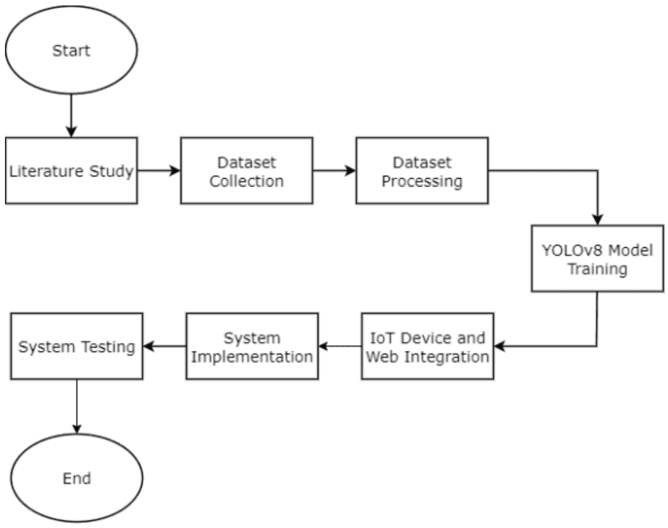

Fig.1. Methodology Stages

The research on the manufacture of car and motorcycle detectors uses a qualitative method approach. This research is

DOI : https://doi.org/10.25139/inform.v10i2.10408

designed through several important and structured stages to ensure that the research results can be scientifically accounted for and follow the objectives that have been set. The flow of research stages is illustrated in Fig. 1. The research method's stages begin with a literature study, followed by the collection of datasets consisting of car and motorcycle images. These datasets then go through a processing phase that includes augmentation and labelling. After processing, the YOLOv8 model is trained using the prepared datasets to enable accurate vehicle detection. Once training is completed, the process continues with the integration of IoT devices and a web interface, ensuring that hardware components such as the Raspberry Pi, webcam, and buzzer can communicate effectively with the web system. After successful integration, the system is implemented and deployed. The final stage involves comprehensive testing to assess the system's performance and reliability in real-world conditions.

The approach adopted in this study utilizes the prototyping method, which involves creating an initial software version to demonstrate concepts, test designs, and identify potential issues and solutions. This method enables users to gain a practical understanding of system functionality. The prototype developed in this research aims to provide an early visualization of the application to be built, enabling users to evaluate the design and refine it accordingly. Evaluation outcomes from users serve as the primary reference for developing the final application, constituting the research's ultimate product and output. The prototyping approach is highly relevant, as it supports the iterative development of the YOLOv8n detection model and facilitates direct integration with IoT devices, such as Raspberry Pi, and web interfaces, emphasizing practical system development and continuous testing for valid, real-world applicability. The prototyping stages in developing the YOLOv8-based parking violation detection system with a web dashboard include identifying user requirements, developing the prototype, evaluating the prototype, and implementing it into a fully functional system.

A. Literature Study

The literature study is an approach used to collect, analyze, and interpret information from various written sources such as books, scientific articles, official documents, and previous research reports. In this study, the author conducted a thorough search and analysis of references, including journals, articles, and documentation related to the development of Artificial Intelligence (AI), particularly the YOLOv8n algorithm and its application in real-time vehicle detection. The reviewed literature primarily focuses on the application of the Internet of Things (IoT) in automation systems, particularly in the transportation sector. This study aims to assess the performance of YOLOv8 in detecting two-wheeled and four-wheeled vehicles, understand IoT system integration in real-world settings, and examine previous research to identify the uniqueness and contributions of this study. The insights gained serve as a foundation for designing a YOLOv8n-based vehicle detection system connected to a camera and tested directly in areas prone to parking violations.

1) Observation: In this study, the author conducted direct observations along Palabuhan Ratu Square, which is marked with "No Parking" signs in Palabuhan Ratu, Sukabumi, Indonesia. The purpose of this observation was to identify the conditions and issues related to illegal parking violations occurring throughout the area and to examine the monitoring processes implemented by the Department of Transportation of Sukabumi Regency. As illustrated in Fig. 2, observations indicate that numerous two-wheeled and four-wheeled vehicles are parked illegally, both on road shoulders and sidewalks, particularly during peak hours. This situation results in road narrowing, traffic congestion, and poses a risk to pedestrian safety, as sidewalks can no longer serve their intended function.

Fig.2. Observation Palabuhan Ratu Square

2) Interview: The interview method involves direct interaction between the researcher and relevant stakeholders to collect in-depth information regarding the needs, expectations, and challenges associated with the system being developed. This interview aimed to gather more detailed and practical field data, which is summarized. The findings highlighted a clear distinction between on-street and off-street parking, with the Department of Transportation only overseeing government-regulated parking areas, excluding those managed privately. Palabuhan Ratu, Sukabumi, Indonesia, was identified as a hotspot for frequent parking violations. Additionally, the department expressed support for research initiatives that utilize technology to monitor and manage illegal parking activities.

B. Dataset Collection

The first approach involves training the model using the original dataset without augmentation, while the second uses an augmented dataset to address the issue of class imbalance. The author collected an image dataset consisting of 4,500 cars (four-wheeled vehicles), 4,500 motorcycles (two-wheeled vehicles), and 1,000 negative class images. This dataset is used to train the YOLOv8n model [19].

C. Dataset Processing

DOI : https://doi.org/10.25139/inform.v10i2.10408

In data asset processing, there are two key stages: the augmentation process and the labelling dataset, which are crucial steps in preparing data for training the violation detection model.

1) Dataset augmentation: Image data augmentation is a technique to artificially increase the size and variety of a training dataset by applying small transformations to existing images without changing their original identity. The model is evaluated on a held-out test set comprising original and augmented images that were unseen during training, yielding an unbiased estimate of generalization [20]. In Figs. 3 and 4, data augmentation techniques are applied to enrich the diversity of images used in training the object detection model, thereby enhancing its ability to detect objects under various visual conditions.

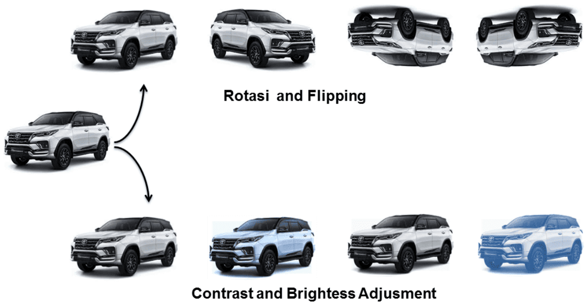

Fig.3. Augmentation Car

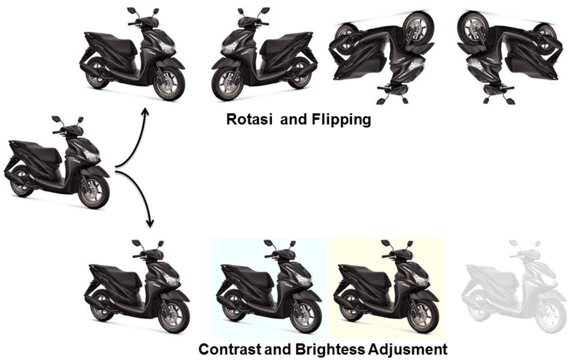

Fig.4. Augmentation Motorcycle

2) Dataset Labelling: Data labelling is a crucial first step in machine learning, where the dataset is classified into specific categories. This process is a crucial foundation for proceeding to the data processing stage.

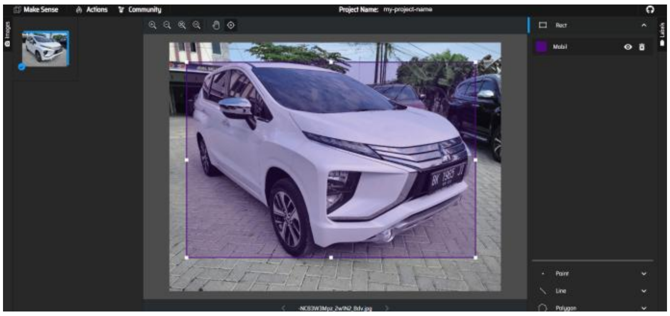

Fig.5. Labelling Dataset

As illustrated in Fig.5, vehicle object labelling in this study was performed using makesense.ai, an online annotation platform that enables the creation of bounding box labels in multiple formats. This tool was selected due to its accessibility and support for export formats compatible with YOLO-based training. Accurate annotation ensures that the model can effectively learn to recognize and differentiate between vehicles and non-vehicle objects, thereby enhancing detection accuracy. The labelled dataset produced through this process was subsequently used for training, validation, and testing phases of the model development pipeline. While pre-processing is a crucial step before training the object detection model, it provides coordinate points with the bounding box [21].

D. YOLOv8 Train

The YOLOv8n model training process was conducted to achieve optimal performance in detecting cars and motorcycles, utilizing 9,000 vehicle images (comprising cars and motorcycles) and 1,000 non-vehicle images, which were divided into 80% training, 10% validation, and 10% testing sets. Training includes parameter settings such as a learning rate of 0.001, a burn-in period of 1,000, and data augmentation (angle 0, saturation 1.5, exposure 1.5, and hue 0.1) to prevent overfitting and increase data variation. Evaluation is carried out based on the mAP value over 50 training epochs to measure the detection accuracy and labelling capabilities of cars and motorcycles. The results of this training serve as the foundation for an IoT-based vehicle detection system that is accurate and reliable under various environmental conditions. The dataset composition is presented in Table I.

TABEL I
TRAINING, TESTING AND VALIDATION

Sample	***Training* (80%)**	***Testing* (10%)**	**Validasi (10%)**	**Total**
Car	3.600	450	450	4.500
Motorcycle	3.600	450	450	4.500
Negative	800	100	100	1.000
			Total Dataset	10.000

This division was chosen so that most of the data can be used to train the model effectively, while the validation data is used

DOI : https://doi.org/10.25139/inform.v10i2.10408

to check the model's performance during training and ensure that it is not overfitting.

E. IoT Device and Web Integration

The development of an IoT-based parking violation detection system requires careful planning and a thorough needs analysis to ensure the system functions effectively in real-world conditions. Based on the results of observations and interviews, the system must meet several interconnected requirements. It should be capable of automatically and in real-time detecting vehicles parked in no-parking zones, which is supported by the use of a webcam to capture images in the field. These images are then processed locally using a Raspberry Pi integrated with an IoT-based system[22]. The system should also be equipped with an IoT-based warning device, such as a buzzer, that responds to detected violations. Finally, all detection data should be displayed in real-time through a web-based interface, enabling efficient monitoring and response.

1) IoT device schematic: During the development of this device, the installation and design stages involved assembling and testing core components, such as the webcam, Raspberry Pi 4, and buzzer, to ensure they function as expected. This process included hardware installation, network setup, and software configuration to support the vehicle detection system. This stage is crucial to ensure the system operates effectively and is ready for deployment in the field. The construction of the detection device, which utilizes artificial intelligence and IoT, also involved planning the component layout and connecting hardware elements. For a clearer understanding, the diagram below illustrates the arrangement of components and how the camera, microcontroller, and buzzer are interconnected within the system, as shown in Fig.6.

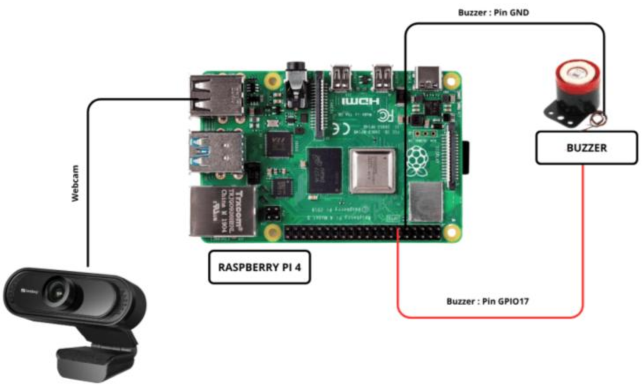

Fig.6. Tool Design Scheme

As illustrated in Fig. 6, the device comprises a Raspberry Pi 4 as the control unit, a webcam for image acquisition, and a buzzer for auditory warnings. The webcam connects directly to the Raspberry Pi, enabling real-time image capture and YOLOv8-based object detection. The buzzer’s signal line is wired to GPIO17, and its ground is connected to GND. When a parking violation is detected, the Raspberry Pi drives GPIO17 to activate the buzzer and emit an audible alert.

2) Architecture, IoT, and Web Integration: The car and motorcycle detection system is designed to be implemented directly in the field, with a flow and devices arranged optimally according to real conditions and needs, to effectively achieve research objectives.

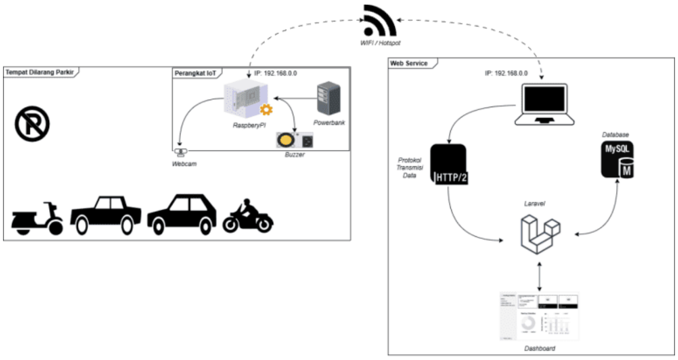

Fig.7. IoT device architecture with web

As illustrated in Fig. 7, the proposed system architecture integrates IoT devices with web services to detect and monitor parking violations in real-time. A Raspberry Pi equipped with a high-definition webcam captures images of vehicles in restricted parking zones and processes them locally using the YOLOv8 object detection model. When a vehicle remains stationary for more than 30 seconds, the system records the image, the vehicle's classification, and the timestamp. This information is transmitted via HTTP to a Laravel-based server, stored in a MySQL database, and displayed on a web dashboard using WebSocket communication for real-time updates, eliminating the need for manual refreshes. Additionally, a buzzer connected to the Raspberry Pi provides immediate on-site alerts for violations.

III. Result and Discussion

The training and evaluation process of the YOLOv8n model is a critical step in developing an automatic vehicle detection system. The YOLOv8n (You Only Look Once version 8, nano variant) model was selected for its computational efficiency and inference speed, making it highly suitable for real-time systems, particularly in environments with limited hardware resources. Training was conducted on the Google Colab platform, utilizing GPU acceleration to enhance computational performance and optimize model parameters. The dataset consisted of two main object classes: four-wheeled vehicles (cars) and two-wheeled vehicles (motorcycles), with annotations formatted according to YOLO standards. During training, performance metrics such as box loss, classification loss, and distribution focal loss (DFL) were monitored, along with evaluation metrics including precision, recall, and mean Average Precision at IoU thresholds of 0.5 (mAP50) and across the 0.5–0.95 range (mAP50–95).

A. YOLOv8n Model Training Results

In this study, the YOLOv8n model (You Only Look Once version 8, nano variant) was used due to its advantages in computational efficiency and high inference speed, making it

DOI : https://doi.org/10.25139/inform.v10i2.10408

well-suited for implementation in IoT and web-based systems. The dataset consisted of labelled vehicle images divided into two classes: cars (four-wheeled vehicles) and motorcycles (two-wheeled vehicles). The labelling process followed the YOLO format, which stores bounding box coordinates and object class information in separate text files.

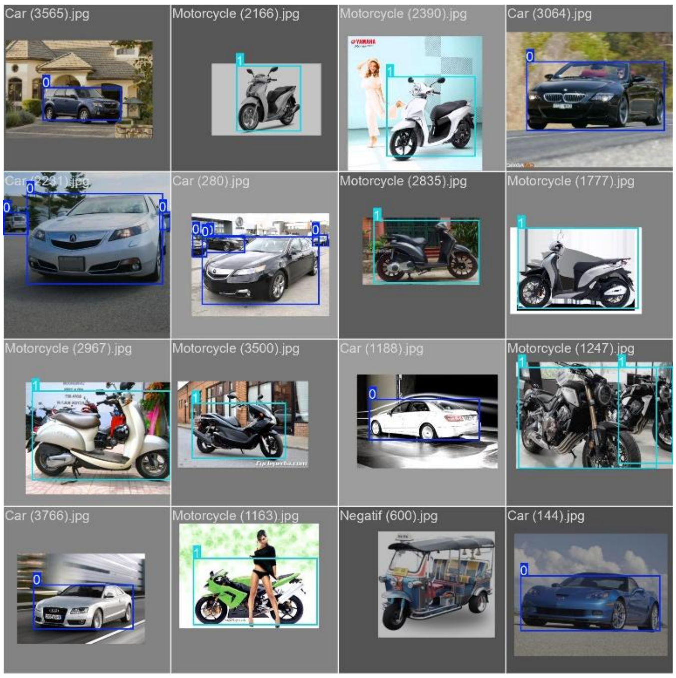

Fig.8. Training Result

As illustrated in Fig. 8, the trained YOLOv8n model was evaluated through visualization results, where each image displays the model's predictions of vehicle objects, including bounding boxes and class labels. Label "0" represents cars (four-wheeled vehicles), while label "1" represents motorcycles (two-wheeled vehicles). The test images used for visualization vary in several conditions, including bright and dim lighting, simple and complex backgrounds, and different vehicle angles. The purpose of this visualization is to evaluate the model's generalization ability on previously unseen data and to identify potential classification or detection errors. The results demonstrate that the model provides fairly accurate predictions, with the majority of vehicle objects successfully detected and classified into the correct categories.

A confusion matrix in Fig.9 illustrates the model's prediction performance against the actual data, where the rows represent the predicted classes and the columns represent the true classes. In this matrix, the main diagonal shows the number of True Positives (TP) for each class—440 for cars and 475 for motorcycles—indicating correct predictions. The background class has 0 TP, indicating that no correct predictions were made for the background. False Positives (FPs) occur when the model incorrectly predicts a class, such as 155 motorcycles misclassified as cars and 39 instances (11 cars and 28 backgrounds) misclassified as motorcycles. False Negatives (FN) reflect actual instances that were not correctly identified: 11 cars predicted as motorcycles, and 207 motorcycles misclassified as either cars (155) or background (52). Understanding these TP, FP, and FN values provides a detailed view of the model's accuracy in detecting and classifying each vehicle type.

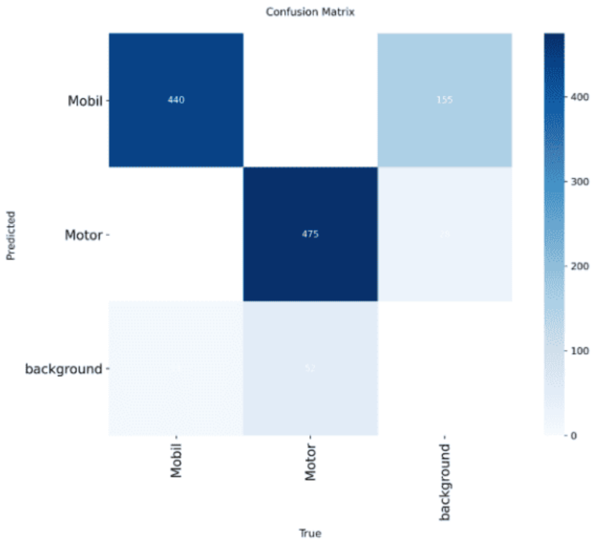

Fig.9. Confusion Matrix

In Fig. 10, it can be seen that during the 50 epochs of training, the YOLOv8n model exhibits a consistent decrease in training and validation losses (box_loss, cls_loss, and dfl_loss), indicating improved localization, classification, and bounding box regression performance over time. At the same time, evaluation metrics such as precision, recall, mAP50, and mAP50-95 demonstrate a steady improvement, reflecting increased detection accuracy and generalization capability on validation data. These results indicate that the trained YOLOv8n model achieves high efficiency and reliability in automatically detecting vehicles.

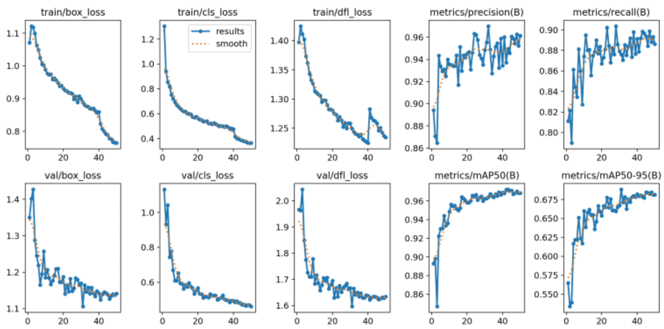

Fig.10. Confusion Matrix Losses & Evaluation Metrics for 50 Epochs

Fig.11 does not explicitly display the confidence threshold but illustrates how precision changes as recall increases, which typically occurs by lowering the confidence threshold. An ideal curve would approach the top-right corner, indicating both high precision and high recall. Based on the graph, the Car class shows an almost ideal curve, with a very high Average

DOI : https://doi.org/10.25139/inform.v10i2.10408

Precision (AP) of 98.4%, indicating that the model can accurately detect nearly all cars. In contrast, the Motorcycle class has a slightly lower curve with an AP of 94.1%, suggesting that the model struggles slightly to maintain high precision when aiming for high recall in motorcycle detection. Overall, the graph demonstrates excellent model performance, with a mean Average Precision (mAP) of 96.3%, while also highlighting areas for improvement, particularly in stabilizing precision for motorcycles at higher recall levels.

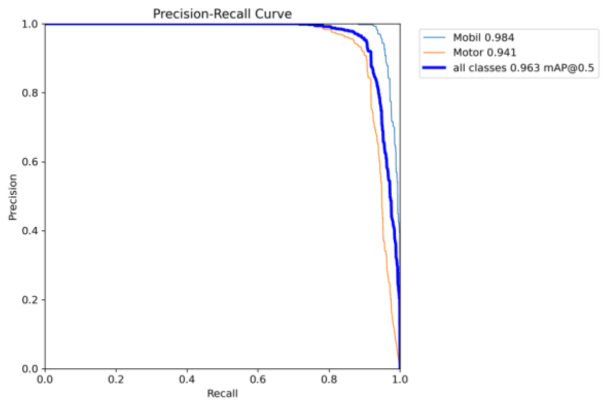

Fig.11. Precision-Recall Curve

Fig.12 shows that the model achieves 100% precision at a confidence threshold of 0.894 across all combined classes. It also illustrates that the Motorcycle class reaches high precision more quickly than the Car class, indicating that motorcycle predictions tend to be more accurate at low to moderate threshold levels. However, the Car class also eventually achieves maximum precision as the confidence threshold approaches its optimal range (~0.89–0.90). This Precision-Confidence Curve illustrates that increasing the confidence threshold can significantly reduce false detections to zero, albeit at the cost of potentially missing some objects, resulting in lower recall.

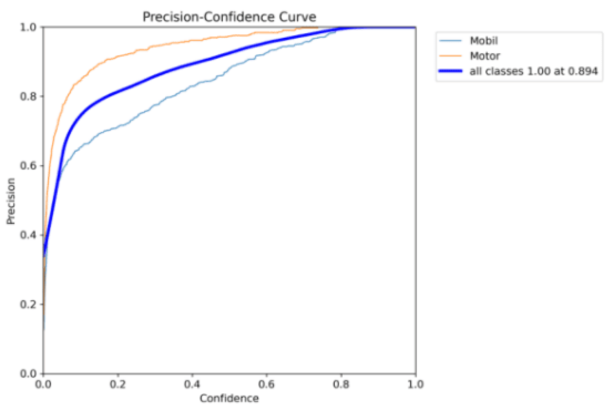

Fig.12. Precision-Confidence Curve

Fig.13 shows a downward trend, indicating that as the confidence threshold increases, the number of accepted predictions decreases, resulting in more missed detections (lower recall). The maximum recall (~98%) is achieved at the lowest threshold (0), indicating that approximately 2% of objects remain undetected, even at the lowest confidence level. The recall for the Motorcycle class drops more rapidly than for the Car class, suggesting that motorcycle predictions tend to have lower confidence scores and are thus filtered out earlier as the threshold rises. This indicates that maintaining high recall for motorcycles is more challenging compared to cars when the confidence threshold is increased. Overall, the graph suggests that the model is more confident in detecting cars than motorcycles.

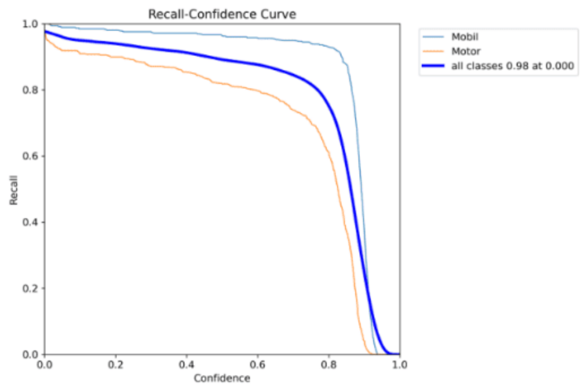

Fig.13. Recall-Confidence Curve

Fig.14 displays a curved pattern, indicating that the F1-score initially increases with the confidence threshold until it reaches an optimal peak, then declines as the threshold becomes too high and recall drops significantly. In the given graph, the highest overall F1-score across all classes is 0.91, achieved at a confidence threshold of approximately 0.603. The Car class reaches a higher maximum F1-score of 0.93 compared to the Motorcycle class, which peaks at 0.87, suggesting that the model is more accurate and stable in detecting cars than motorcycles. This graph clearly shows that using a confidence threshold of around 0.60 provides an optimal balance between precision and recall across all classes.

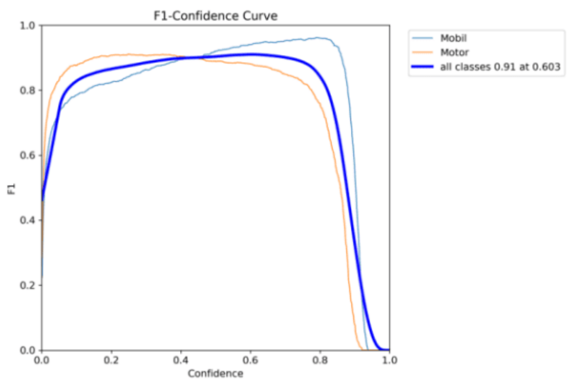

Fig.14. F1-Confidence Curve

Based on the evaluation results presented through several performance graphs, the YOLOv8n model demonstrates excellent capability in detecting vehicle objects, particularly for

DOI : https://doi.org/10.25139/inform.v10i2.10408

the Car class. The Precision-Recall curve shows that the Car class achieves an Average Precision (AP) of 98.4% with an almost ideal curve. In comparison, the Motorcycle class obtains an AP of 94.1%, indicating that the model encounters some difficulty in maintaining high precision when attempting to achieve high recall for motorcycles. The Precision-Confidence curve reveals that 100% precision is achieved at a confidence threshold of 0.894 across all classes, with the Motorcycle class reaching high precision earlier than the Car class.

Meanwhile, the Recall-Confidence curve exhibits a declining trend in recall as the confidence threshold increases, with a steeper drop observed in the Motorcycle class. This suggests that the model is more confident in detecting cars than motorcycles. The F1-Score curve indicates that the highest overall F1-score of 0.91 is achieved at a threshold of approximately 0.603, with the Car class achieving a peak F1-score of 0.93 and the Motorcycle class 0.87. Therefore, it can be concluded that setting the confidence threshold around 0.60 provides an optimal balance between precision and recall. While the model performs reliably in vehicle detection, further improvement is needed to enhance accuracy in motorcycle detection.

B. *Vehicle Detection Model Validation*

The model validation process was conducted using a dataset comprising 1,000 images, with a total of 978 object instances distributed across two main classes. The Car class comprised 451 instances, while the Motorcycle class had a higher distribution with 527 instances. Performance metrics were measured using a standard confidence threshold, resulting in several key findings. First, the model exhibited varying detection capabilities between the two object classes. Second, there was a significant difference in precision and recall values between the Car and Motorcycle classes. Third, the mAP results across different thresholds indicated the model's consistent performance in object detection.

TABEL II
VALIDATION TEST RESULT

Class	Precision	Recall	mAP50	mAP50-95
All	0,955	0,876	0,963	0,688
Car	0,927	0,956	0,984	0,729
MotorCycle	0,984	0,796	0,941	0,647

Based on Table II, the model demonstrates varying performance across the two object classes tested. For the Motorcycle class, the model achieves a very high precision of 0.984, but the recall is relatively lower at 0.796, compared to the Car class, which records a higher recall of 0.956. This indicates the model's tendency to produce false negatives when detecting motorcycle objects. Overall, the model achieves a mean Average Precision at an IoU threshold of 0.5 (mAP50) of 0.963 (96.3%), reflecting excellent detection accuracy, with the best performance observed in the Car class (mAP50 of 0.984). However, the drop in mAP50–95 to 0.688 suggests a decline in accuracy as the IoU threshold becomes more stringent, indicating that the model lacks consistency in detecting objects with a high overlap between predicted and ground truth bounding boxes.

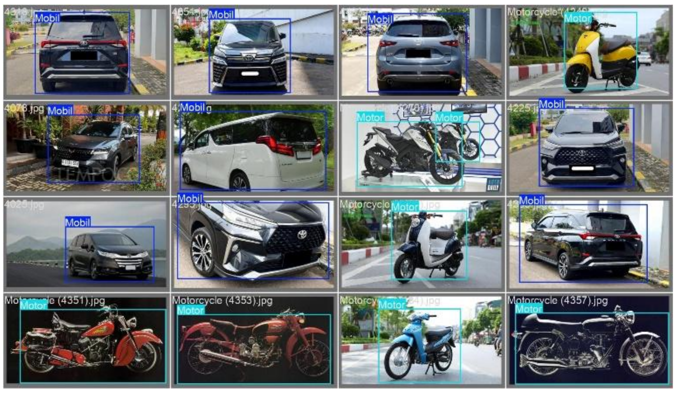

Fig.15. YOLOv8 Model Validation Result

Fig.15 presents the validation results of the vehicle object detection model using YOLOv8. The model was tested to recognize two main object classes: "Car" and "Motorcycle." Each detected object is marked with a blue bounding box, accompanied by its class label and confidence score, which indicates the model's certainty in its prediction. The confidence values range from 0.3 to 0.976, with most predictions exhibiting high confidence (⩾ 0.9), indicating a strong identification capability and reliability in object recognition. However, some predictions fall within the lower confidence range (0.3–0.7), which may indicate potential false positives or uncertainty in object identification, particularly in images with uncommon angles or complex lighting conditions. These validation results demonstrate that the YOLOv8 model can detect and classify vehicles with good and stable accuracy across diverse image conditions. The visual outcomes support the conclusion that the model performs reliably and is suitable for real-world implementation in an automated vehicle detection system.

C. *IoT Device Implementation With YOLOv8n Model*

To deploy the vehicle detection system directly in real-world environments, the trained YOLOv8n model was integrated into an Internet of Things (IoT) device based on the Raspberry Pi 4 platform. This implementation enables the system to operate independently, without relying on external server infrastructure. By utilizing a combination of hardware components, such as a webcam for image capture and a 12V buzzer as a warning actuator, the system is capable of performing real-time vehicle detection and providing immediate physical responses at the location when an object is detected.

Fig.16. IoT device Design Results

DOI : https://doi.org/10.25139/inform.v10i2.10408

Fig.16 shows the prototype of the IoT device, which consists of a power bank, a 12V buzzer, a Raspberry Pi 4, and a webcam. The components are arranged compactly for easy field deployment. When the YOLOv8 model detects a vehicle with a confidence score above a set threshold (e.g., 0.8), the Raspberry Pi triggers the buzzer via GPIO pins as a warning signal. This design enables real-time detection and immediate physical alerts, demonstrating the model's practical application in IoT-based monitoring systems.

D. *Results of Observations of Parking Violators*

Field observations were conducted in the area surrounding Palabuhan Ratu, Sukabumi, Indonesia, to capture actual conditions related to illegal parking behaviour in zones marked with "No Parking" signs. The image below serves as visual documentation of the observed violations.

Fig.17. Parking Violators on Palabuhan Ratu Square

As shown in Fig.17, numerous vehicles can be observed violating parking regulations in areas marked with "No Parking" signs around Palabuhan Ratu, Sukabumi, Indonesia. These violations, which include both two-wheeled and four-wheeled vehicles, involve improper parking on sidewalks, road shoulders, and directly beneath prohibition signs. The high frequency and widespread distribution of these violations indicate a significant lack of enforcement and public compliance with parking regulations. This highlights the urgent need for a real-time, automated monitoring system to address and reduce such behaviour in busy public spaces.

E. *Implementation of Web-Based Car and Motorcycle Detection System*

As a result of training the YOLOv8n model, the system was implemented as a web-based application to facilitate practical user interaction with the vehicle detection system. This approach allows users to upload images directly, receive detection results automatically, and access data visualizations through an interactive web interface.

1) Dashboard Display: As the primary information hub, the dashboard of the parking violation detection system is designed to present detection results in a concise, real-time, and structured manner, making it easier for users to comprehensively monitor vehicle violation activities. Fig.18 displays the dashboard page, which serves as the main interface of the system after the user has successfully logged in. The dashboard presents two primary visual data elements: a pie chart showing the total number of parking violations and a bar chart illustrating violation activity over time. The data is categorized by vehicle type, cars and motorcycles. Additionally, the dashboard features a preview of the latest violation images, complete with object labels, timestamps, and test location information. All elements on this page are arranged dynamically and informatively to support fast and efficient analysis and monitoring of violation activities by users.

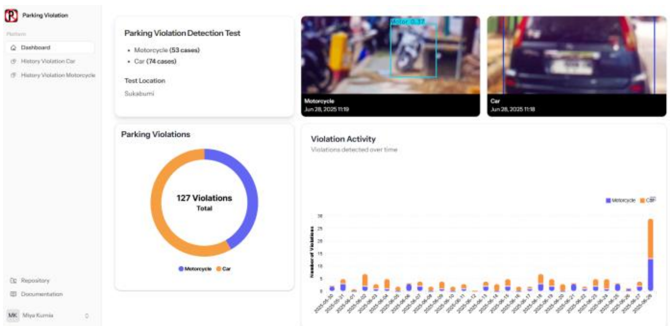

Fig.18. Dashboard Display

2) Violation History Display: To support the tracking and documentation of parking violations detected by the system, the web interface is equipped with a history violation feature that serves as a repository for detection result data. Fig.19 and Fig.20 display the interface of the history violation page within the parking violation detection system, presenting detection data in a visual format. The page is divided into two main sections based on vehicle type: "Violation Motorcycle" and "Violation Car." Each violation is shown as a card containing a snapshot of the incident, the object class label (e.g., "Motorcycle"), and the timestamp of the violation. This interface design aims to simplify the user's navigation through violation records, organized by time and vehicle type. Additionally, this feature plays a vital role in supporting the digital and structured verification and documentation process of parking violations.

DOI : https://doi.org/10.25139/inform.v10i2.10408

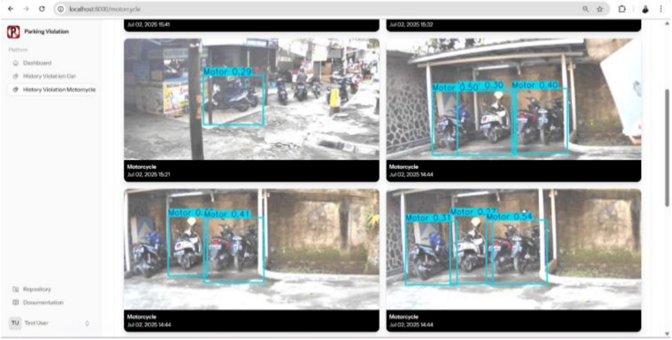

Fig.19 View History of Motorcycle Parking Violators

Fig.20. View History of Car Parking Violators

F. *IoT device and Web Integration test results at the testbed*

This testing method aims to ensure that each component of the device operates according to predetermined specifications, regardless of its internal structure. Using Black Box Testing, the focus is on the given input and the resulting output, thereby comprehensively validating the device's performance and reliability across various usage scenarios.

TABEL III
RESULT BLACKBOX TESTING IOT DEVICE & WEB

Testing Parameters	Time	IoT Devices	Web	status
	>30s	Object detected, activate buzzer	Send database to web	Pass
	>30s	Object detected, activate buzzer	Send database to web	Pass
	>30s	Object detected, activate buzzer	Send database to web	Pass
	>30s	Object detected, activate buzzer	Send database to web	Pass
	>30s	Object detected, activate buzzer	Send the database to the web	Pass

The results of BlackBox testing on the parking violation detection system prototype, which utilizes the YOLOv8n model integrated with a webcam and Raspberry Pi, and is equipped with a 12V buzzer as a warning device, as well as a web-based output system, demonstrate consistent and accurate performance. During testing, the system successfully detected car and motorcycle objects parked in no-parking areas for more than thirty seconds. The buzzer responded appropriately, activating for thirty seconds whenever a violation was detected. Furthermore, the system reliably transmitted violation data—including captured images, object class labels, timestamps, and location information to the web dashboard in real-time. This output was presented on the user interface, enabling users to review and monitor violations in real-time. These results confirm that the prototype meets the specified requirements and shows strong potential for practical implementation as an automated, real-time parking violation monitoring system.

IV. CONCLUSION

This research successfully developed and tested an automated detection system for monitoring parking violations involving cars and motorcycles in no-parking zones, utilising the YOLOv8n model integrated with IoT hardware and a web dashboard. The system achieved high detection accuracy (mAP50 96.3%), with better performance for cars than for motorcycles. Real-time tests showed that the prototype could consistently detect violations and trigger on-site buzzer alerts, while reliably sending violation data to a live web dashboard for monitoring. The system met all design requirements and demonstrated practical feasibility, particularly in areas with frequent instances of illegal parking. Overall, the study demonstrates that combining deep learning, IoT, and web technologies offers an effective solution for automated parking enforcement and has significant potential for broader smart city applications.

REFERENCES

[1] R. Ramadhan, “Evektivitas Penertiban Parkir Liar Kota Surabaya Berdasarkan Perda Nomor 3 Tahun 2018,” 2024.

[2] P. R. Kristiawan, D. A. S. Dewi, and S. Suharso, “Implementasi Peraturan Pemerintah Nomor 34 Tahun 2006 Tentang Jalan Berkaitan Dengan Pemeliharaan Jalan (Studi Kasus Jalan yang Menjadi Kewenangan Kabupaten Magelang),” *Borobudur Law Review*, vol. 2, no. 1, pp. 30–39, Feb. 2020, doi: 10.31603/burrev.3919.

DOI : https://doi.org/10.25139/inform.v10i2.10408

[3] A. Bintang Zaidan, M. Azka Firdaus, A. Ilham Irianto, and P. Seren Wesal, “Deteksi Objek Kendaraan Tank Dengan Model YOLO Dalam Pengawasan Wilayah Darat,” *Agustus*, vol. 2, no. 8, pp. 807–815, 2024.

[4] M. H. K. Rizal, J. Sahertian, and R. H. Irawan, “Perbandingan akurasi klasifikasi kendaraan menggunakan YOLOv8 dan Faster R-CNN,” in Proc. SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi), vol. 9, Kediri, Indonesia, 2025, pp. 388–395.

[5] L. Satya, M. R. D. Septian, M. W. Sarjono, M. Cahyanti, and E. R. Swedia, “Sistem Pendeteksi Plat Nomor Polisi Kendaraan Dengan Arsitektur YOLOV8,” *Sebatik*, vol. 27, no. 2, pp. 753–761, Dec. 2023, doi: 10.46984/sebatik.v27i2.2374.

[6] E. U. Armin, A. Purnama Edra, F. I. Alifin, I. Sadidan, I. P. Sary, and U. Latifa, “Performa Model YOLOv8 untuk Deteksi Kondisi Mengantuk pada pengendara mobil,” *BRAHMANA: Jurnal Penerapan Kecerdasan Buatan*, vol. 5, no. 1, pp. 67–76, 2023.

[7] Y. K. Rohiman, B. Kanata, L. Ahmad, and I. Akbar, “Bulletin Of Computer Science Research, Deteksi dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (Yolov7),” *Media Online)*, vol. 5, no. 3, pp. 268–276, 2025, doi: 10.47065/bulletincsr.v5i3.509.

[8] A. I. Pradana, H. Harsanto, and W. Wijiyanto, “Deteksi Rambu Lalu Lintas Real-Time di Indonesia dengan Penerapan YOLOv11: Solusi Untuk Keamanan Berkendara,” *Jurnal Algoritma*, vol. 21, no. 2, pp. 145–155, Nov. 2024, doi: 10.33364/algoritma/v.21-2.2106.

[9] N. J. Hayati, D. Singasatia, M. R. Muttaqin, T. Informatika, S. Tinggi, and T. Wastukancana, “Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)V8 Untuk Menghitung Kendaraan,” *KOMPUTA : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 2, 2023.

[10] I. , K. I. L. , & A. N. Somantri, “S. Somantri, I. L. Kharisma, and N. Angelina, “Rancang Bangun Aplikasi,” *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 7 No. 2, pp. 721–730, Sep. 2023.

[11] A. Setiyadi, E. Utami, and D. Ariatmanto, “Analisa kemampuan algoritma YOLOv8 dalam deteksi objek manusia dengan metode modifikasi arsitektur,” *J. Sains Komput. Inform. (J-SAKTI)*, vol. 7, no. 2, pp. 891–901, Sep. 2023.

[12] S. Tamang, B. Sen, A. Pradhan, K. Sharma, and V. K. Singh, "Enhancing COVID-19 safety: Exploring YOLOv8 object detection for accurate face mask classification," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 2, pp. 892–897, 2023.

[13] R. Vaghela *et al.*, "Land Cover Classification for Identifying the Agriculture Fields Using Versions of YOLO V8," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 18, pp. 8672–8684, 2025, doi: 10.1109/JSTARS.2025.3547058.

[14] G. Fairuz Mumtaz, J. Zeniarja, A. Luthfiarta, and A. N. Imam Muttaqin, "Optimizing Face Recognition and Emotion Detection in Student Identification Using FaceNet and YOLOv8 Models," *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 10, no. 1, pp. 34–44, Jan. 2025, doi: 10.25139/inform.v10i1.9304.

[15] R. S. I. , H. W. A. , & R. W. K. Sihombing, “Implementasi Yolo V8 Untuk Mendeteksi Mata Uang Rupiah Emisi Tahun 2022 Ber-Output Audio,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8 No.4, pp. 5900–5905, 2024.

[16] F. Badri, S. U. Ruhmana Sari, and S. A. Bin Hamzah, "Analysis of Driver Drowsiness Detection System Based on Landmarks and MediaPipe," *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 10, no. 1, pp. 21–28, Jan. 2025, doi: 10.25139/inform.v10i1.9325.

[17] “S. R. Suartika E. P, I Wayan, Wijaya Arya Yudhi, ‘Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101,’ J. Tek. ITS, vol. 5, no. 1, p. 76, 2021.

[18] V. Y. P. Ardhana, M. T. Hidayat, M. Jannah, Sumiati, P. Rini, and N. Sari, "Implementasi RESTful API Pada Laravel dan Simulator IoT Wokwi Untuk Pengukuran Suhu dan Kelembaban Menggunakan Metode Waterfall," Arcitech: Journal of Computer Science and Artificial Intelligence, vol. 3, no. 2, pp. 93–109, Dec. 2023, doi: 10.29240/arcitech.v3i2.9334.

[19] I. C. Wicaksana, R. Anggi Pramunendar, G. W. Saraswati, and G. Alfa Trisnapradika, "Skin Lesion Classification Using YOLOv11 on the HAM10000 Dataset," *Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 10, no. 1, pp. 45–52, Jan. 2025, doi: 10.25139/inform.v10i1.9310.

[20] U. A. Hasib, R. M. Abu, J. Yang, U. A. Bhatti, C. S. Ku, and L. Y. Por, “YOLOv8 framework for COVID-19 and pneumonia detection using synthetic image augmentation,” Digital Health, vol. 11, pp. 1–18, 2025, doi: 10.1177/20552076251341092.

[21] "I. Maulana, N. Rahaningsih, and T. Suprapti, ‘Analisis Penggunaan Model Yolov8 (You Only Look Once) Terhadap Deteksi Citra Senjata Berbahaya,’ JATI (Jurnal Mhs. Tek. Inform., vol. 7, no. 6, pp. 3621–3627, 2024, doi: 10.36040/jati.v7i6.8271.”.

[22] A. A. B. Ismail, A. A. K. Abdul Ghani, M. H. M. Nor, and S. Subramaniam, "Low-cost smart parking system using Raspberry Pi and IoT," *Telecom*, vol. 5, no. 1, pp. 51–65, Jan. 2024, doi: 10.3390/telecom5010005.

DOI : https://doi.org/10.25139/inform.v10i2.10408