Motorcycle Parking Violation Detection System Using YOLOv7 with Region of Interest Mapping and Object Area Calculation

Haerunnisya Makmur, Wulandari, Muhammad Fajar B*, Andi Baso Kaswar, Dyah Darma Andayani, Fhatiah Adiba, Abdul Wahid, Satria Gunawan Zain

Department of Computer Engineering, State University of Makassar, Makassar, Indonesia

Received 31 July 2024; received in revised form 18 October 2024; accepted 28 October 2024

DOI: https://doi.org/10.46604/aiti.2024.14075

Abstract

The large number of motorcycle users has created challenges, particularly related to parking violations, which can lead to traffic congestion, hinder emergency access, disrupt pedestrian pathways, and inconvenience other users. Therefore, this study aims to detect motorcycle parking violations in unsupervised restricted areas using YOLOv7 to classify non-parking, parking, and personal objects. The best model is achieved at the 28th epoch with an mAP value of 0.953 at the 0.5 threshold. Parking restriction areas are defined using a Region of Interest (ROI), where violations depend on the parking object's detected coverage within the ROI exceeding 50%. By employing an area calculation method, the results show better performance compared to methods without area calculation, achieving a recall of 89.7%, precision of 82.6%, and F1-score of 86.2% with a confidence threshold of 0.5.

Keywords: computer vision, parking violation, region of interest, motorcycle, YOLOv7

1. Introduction

Mobility is one of the important factors in strengthening the economy, which evolves along with the population's activities to meet their needs, especially in the context of transportation accessibility and efficiency [1-3]. In Indonesia, personal vehicles such as cars and motorcycles have become the most common and convenient means to commute and conduct any activity [4-5], where motorcycle is the preferred choice for the public due to the efficiency of driving time and affordability, especially in urban areas with heavy traffic [6]. Based on data from Statistics Indonesia, motorcycle dominates the number of vehicles with 132,433,679 units or around 84.3% of the total 157,080,504 vehicle units in 2023 [7].

Despite providing ease of mobility and accessibility, the increasing number of motorcycle vehicles also poses various challenges, especially related to traffic violations [8-9]. One of the common violations that often occurs in society is parking violation [10], which can be ascribed to limited parking spaces, lack of clear signs, and lax enforcement of parking rules [11-12]. Motorcycle parking lots in small and restricted areas require more attention in enforcing parking rules, as they can incur various problems such as traffic congestion, hindering emergency access, disrupting pedestrian pathways, and creating inconvenience for other users [9]. In addition, the officer's manual enforcement of parking rules is often inefficient, ineffective, and time-consuming due to limited human resources. Therefore, the application of computer vision technology is indispensable to ensure the orderly management of parking areas. This technology can identify vehicles and monitor parking areas without requiring the presence of field officers.

* Corresponding author. E-mail address: fajarb@unm.ac.id

English language proofreader: Chih-Wei Chang

Several studies have conducted parking violation detection using multifarious computer vision and machine learning methods. For instance, one study detected car parking violations on the side of the highway using the Faster Region-based Convolutional Neural Network (Faster R-CNN) with an accuracy of 77.9% [13]. Another study detected taxi parking violations by applying semantic segmentation of PSPNet and YOLOv3, resulting in an accuracy of 96.1% [14]. In addition, some studies detect parking violations by using Region of Interest (ROI) to define parking restriction zones. Akhawaji et al. [15] detected and tracked vehicles using a Gaussian Mixture Model and Kalman Filter, marking vehicles as violators if stayed within the ROI for more than sixty seconds without moving, reaching the F1-measure of 88%.

Similarly, a different study applied MobileNet to detect violations when vehicles remained in the ROI for one minute, eliciting a precision of 98.7% [16]. Further research also detected double parking by employing background subtraction to identify vehicles in the ROI, declaring a violation if the vehicle was stationary for more than six counts, achieving 91% accuracy [17]. Another study defined parking areas using ROI, detecting violations when vehicles were parked outside the designated area, resulting in a precision and recall of 97% and 95%, respectively [18]. These studies demonstrate that the use of ROI is an effective approach for detecting vehicle parking violations.

However, the focus of these studies has primarily been on cars. One notable study related to motorcycle parking violations, conducted by Hernández-Díaz et al. [19] detected motorcycle violations in pedestrian zones by classifying data into four categories: motorcycles with motorcyclists in crosswalks, motorcycles with motorcyclists outside crosswalks, pedestrians in crosswalks, and only motorcycles outside crosswalks. This study employed YOLOv8, Single Shot MultiBox Detector (SSD), and MobileNet, with YOLOv8 achieving the highest mean average precision (mAP) of 84.6%.

These findings, along with the research presented by Yang and Yu [14], highlight that YOLO proves to be an effective method for object detection, demonstrating its capability to accurately identify parking violations across different contexts. Meanwhile, Wang et al. [20] used block matching and motion detection techniques to identify violations involving two-wheeled vehicles such as bicycles, classifying violations when these vehicles remained outside the parking area for more than five minutes, resulting in an average F1-score of 79%. Despite this advancement, motorcycle parking violation detection still presents challenges. Motorcycles are volumetrically smaller, more maneuverable, and often park in irregular positions, hindering the reliability of both time-based and motion-based detection methods, which function well for cars. Moreover, the rapid movement and frequent stops of motorcycles complicate the distinction between legal and illegal parking.

To address these challenges, this study focuses on detecting motorcycle parking violations in restricted areas that are unsupervised by officers. The proposed method entails creating a classification model using YOLOv7 to identify objects in the parking area, including parked motorcycles, non-parked motorcycles, and persons. This model is specifically designed for rapid detection of potential parking violations, such as driverless parked motorcycles, without requiring time-based vehicle monitoring. Furthermore, ROI is established for the restricted parking area, and the area of the parking object within this ROI is calculated. A violation is flagged if the area occupied by the parking object within the ROI reaches 50% or more. This area-based approach aims to ensure accurate detection of violations, even for objects that are not entirely within the ROI but still in violation. The study aims to enhance the efficiency of motorcycle parking violation detection and improve parking management, ultimately contributing to safer and more organized urban environments.

2. Research Methods

The research methods consist of data acquisition, preprocessing and data augmentation, annotation, split data, classification, violation detection, implementation, and evaluation. The methods of the research are graphically depicted in Fig. 1, in which the area with the red line indicates the main focus of the proposed method for detecting motorcycle parking violations.

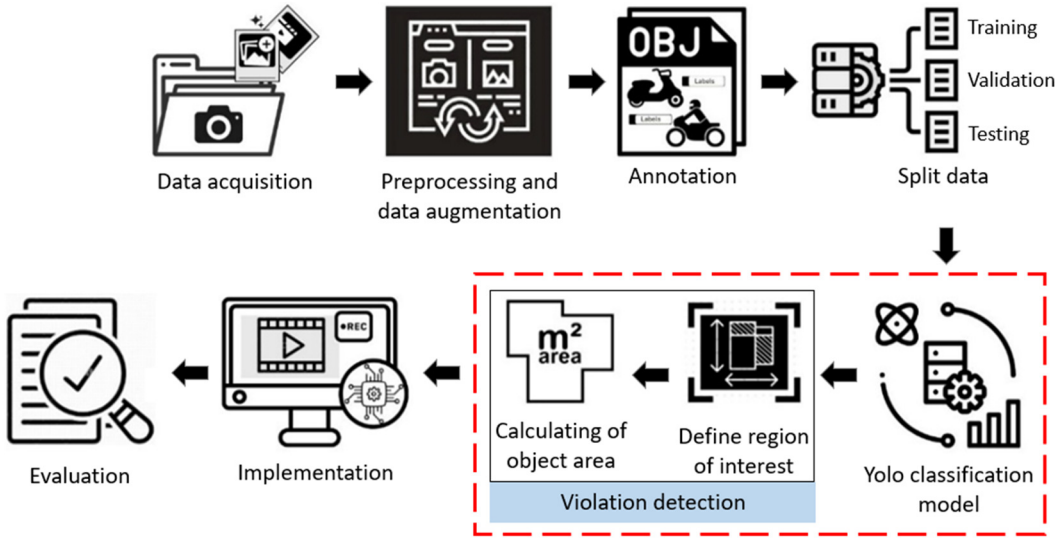


Fig. 1 The research methods

2.1. Data acquisition

The data consists of photos and videos showing activities in the parking lot area in front of the Teknol building of the Department of Informatics and Computer Engineering, State University of Makassar. The photo was taken using a smartphone camera with a total of 600 images, while the video was taken using a webcam with a resolution size of 1080 × 1920 pixels and a speed of 30 frames per second (fps), consisting of 32 videos with a total duration of 254 minutes. Data was collected from the 3rd floor of the Teknol building.

2.2. Preprocessing and data augmentation

Preprocessing is the stage carried out to process raw data before further processing by an algorithm or model [21]. From the 600 images, 180 images were selected by only taking images that have clear objects. In addition, a cropping process was carried out to focus the objects in the image and resize them to reduce the image size. Concerning the video data, preprocessing involved converting the video into a series of frames, where 2,462 frames were selected. Furthermore, data augmentation was carried out through the flip process, which changed the horizontal orientation of the image to obtain more diverse motorcycle position data. The final amount of data used in the parking, non-parking, and person classification object process is presented in Table 1.

Table 1 Total data used

Data	Baseline data	Augmentation data	Total
Image	180	41	221
Video frame	2,462	617	3,079
		Total	3,300

2.3. Annotation

(a) Parking (b) Person (c) Non-parking

Fig. 2 Example of object classes for annotation

The YOLO annotation is a labeling process to mark objects in the image with the appropriate label to enable the object to be recognized and understood during model training [22]. The tool used for YOLO annotations is LabelImg, where the object in the image is annotated using a bounding box surrounding the object and then labeled appropriately. The annotation result is saved in YOLO annotation format (.txt), which contains the normalized bounding box coordinates (relative to the image size) and the object label. The labels for object annotation consist of three scenarios: parking, person, and non-parking, as shown in Fig. 2.

The annotated object for the parking class shown in Fig. 2(a) is a parked motorcycle, which means the absence of a rider on the motorcycle. In the person class shown in Fig. 2(b), the annotated object is a human. Meanwhile, the annotated object in the non-parking class is a motorcycle that is being driven, as shown in Fig. 2(c). In certain images, multiple object classes were annotated, signifying the image contains annotations for various types of objects. The cumulative results of the annotations are presented in Table 2.

Table 2 Total object annotations

Class	Number of annotations
Non-parking	2,689
Parking	2,520
Person	2,533
Total	7,742

2.4. Classification model

The classification model was formed using YOLOv7, which is an object detection algorithm that can recognize and identify objects in an image [23]. This model is used to classify parking, non-parking, and person objects. In this process, the 3,300 data was divided into 80% train data (2,640 images), 10% validation data (330 images), and 10% test data (330 images). The following details of the number of object annotations for each class in training, testing, and validation data are presented in Table 3. The hardware used during training is a laptop device equipped with Windows 11 64-bit, 11th Gen Intel(R) Core(TM) i5-1135G7 2.40 GHz (8 CPUs), 8 GB RAM, and NVIDIA GeForce MX350 5.8 GPU (2 GB Dedicated, 3.8 GB Shared). The hyperparameters used during model training are presented in Table 4.

Table 3 Distribution of the number of object annotations based on training, testing, and validation data

Class	Training	Testing	Validation
Non-parking	2,129	281	279
Parking	2,023	248	249
Person	2,021	265	247
Total	6,173	794	775

Table 4 Hyperparameters used for training model

Hyperparameters	Value
Image size	416
Batch size	4
Epoch	28
Optimizer	SGD (Stochastic Gradient Descent)
Learning rate	Adaptive Learning Rate
Loss function	BCE (Binary Cross-Entropy)

2.5. Parking violation detection

The proposed method for detecting parking violations commences by developing a classification model to identify parking, non-parking, and person objects, as previously explained in the annotation and classification subchapters. The next steps involve forming an ROI and calculating the bounding box area of the parking object within it, enabling the system to detect parking violations. Specifically, ROI refers to a specific area or region in an image that is selected for further analysis [24]. In this study, ROI is used to mark the parking restriction area. The ROI formation process is customized to the street area captured in the video. This area remains consistent and unchanged owing to stable video footage. An illustration of the ROI can be seen in Fig. 3, where ROI formation is performed using the pixel coordinates of the rectangular street area in the frame.

The coordinate points x1 and x2 indicate the x coordinates of the upper left and lower right corner points, while y1 and y2 indicate the y coordinates of the same points. Thus, x1 and y1 represent the upper left point of the area (start point), while x2 and y2 represent the lower right point of the area (endpoint).

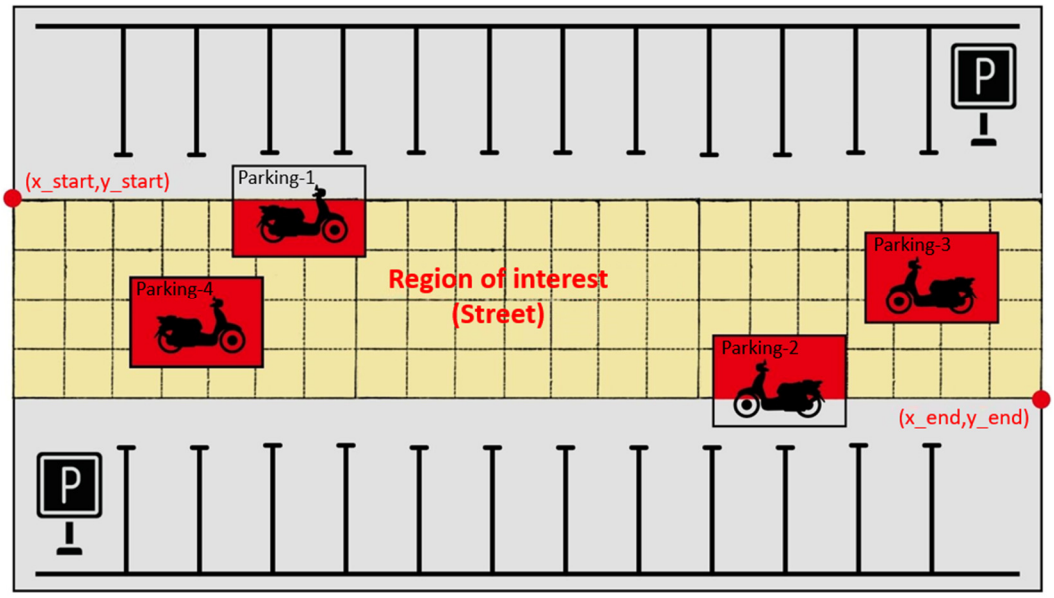


Fig. 3 Illustration of the region of interest

By forming ROI, the focus of object detection is on the parking class within the ROI. A parking object is considered to be in violation if all four coordinate points on the object's bounding box are between the ROI coordinates, as shown in Fig. 3 for parking-3 and parking-4. However, this method is sometimes disadvantageous, especially when a parking object cannot be detected as a violation if only two coordinate points on the object's bounding box are inside the ROI, as seen in Fig. 3 for parking-1 and parking-2 objects. Therefore, the determination of whether a detected object violates the rule or not is based on the percentage of the parking object's bounding box area that falls into the ROI (the intersection area between the bounding box of the parking object and the ROI). To calculate the percentage of object area, the following formula is established:

$$\text{Object area } (\%) = \frac{\text{Intersection area of the bounding box object}}{\text{Area of the bounding box object}} \times 100 \tag{1}$$

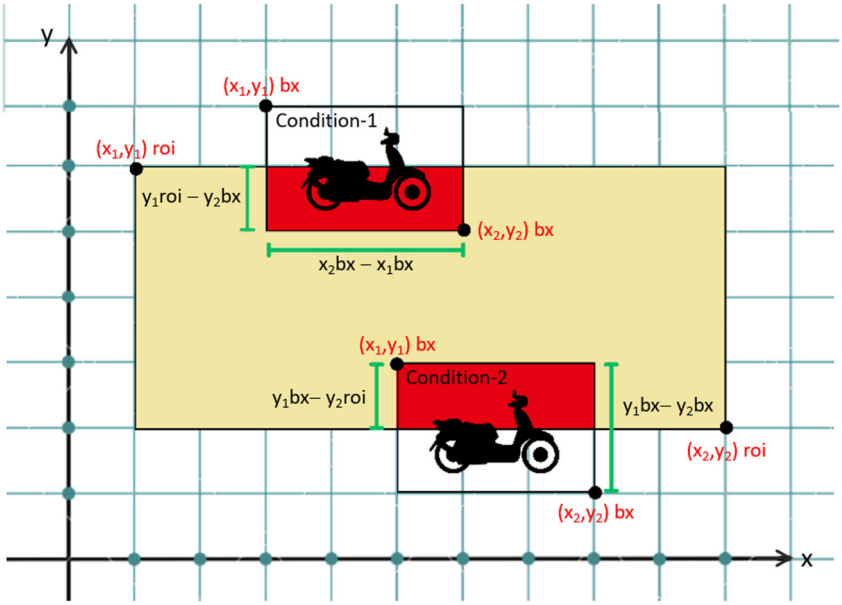


Fig. 4 Illustration of the object area calculation in cartesian diagram

To calculate the area of intersection of the bounding box object and the area of the bounding box object itself, the concept of the rectangular area formula is used, by multiplying the length by the width. Fig. 4 shows the calculation of the bounding box's length and width in a cartesian diagram, illustrating the interrelationship between the coordinate points in the ROI and the bounding box in determining the object area. Based on Fig. 4, the intersection area of the bounding box object and the area of the bounding box are calculated as follows:

$$\text{Intersection area of the bounding box object} = \begin{cases} (y_1roi - y_2bx) \times (x_2bx - x_1bx), \\ \qquad if\ \ y_1bx > y_1roi\ \ \&\ \ y_2bx > y_2roi \\ (y_1bx - y_2roi) \times (x_2bx - x_1bx), \\ \qquad if\ \ y_1bx < y_1roi\ \ \&\ \ y_2bx < y_2roi \end{cases} \tag{2}$$

$$\text{Area of the bounding box object} = (x_2bx - x_1bx) \times (y_2bx - y_1bx) \tag{3}$$

From Eq. (2), the intersection area calculation is based on two conditions: if the detected object's start point is larger than the ROI start point, the area is calculated; if the object's endpoint is smaller than the ROI endpoint, the object is considered outside the ROI endpoint. If both the start and end points are within the ROI, the area is deemed 100% inside the ROI. In Eq. (3), the bounding box area is calculated using the start and end coordinate points.

The overall general architecture of the proposed parking violation detection system is illustrated in Fig. 5. It consists of two main modules: the object detection module and the violation detection module. The object detection module processes each video frame and detects vehicles using the YOLOv7 algorithm, identifying objects within the ROI. The violation detection module calculates the area of any detected parking object within the ROI thereafter. A violation is flagged when 50% or more of the object's area falls within the ROI.

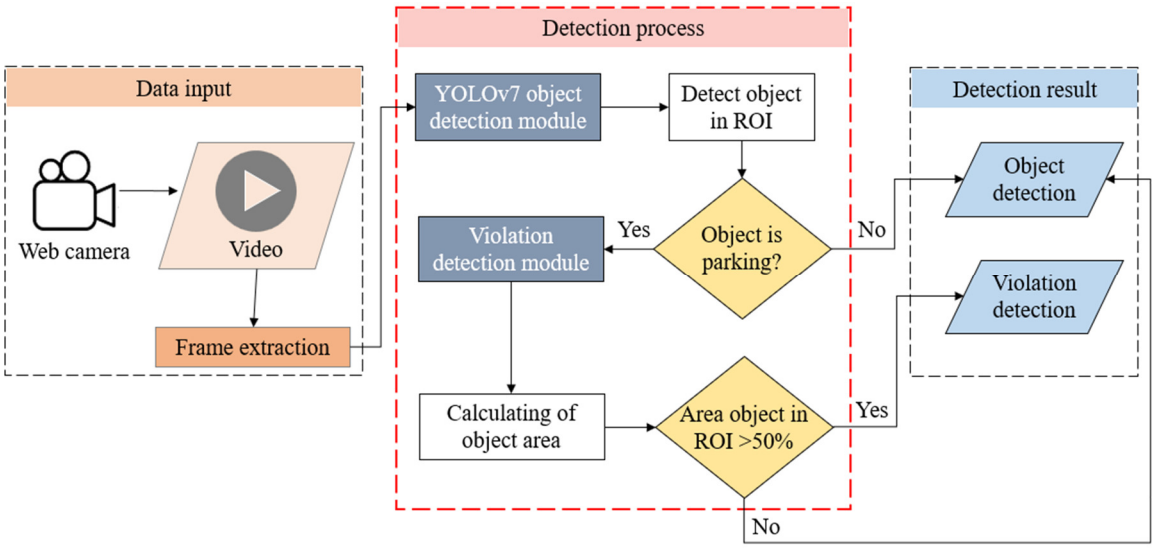


Fig. 5 General architecture of parking violation detection system

2.6. *Implementation*

(a) Place A (b) Place B (c) Place C

Fig. 6 Three different locations for system implementation

The implementation stage was carried out by applying the system that has been developed to detect parking violations to assess the effectiveness and reliability of the model in detecting parking violations. This implementation process involves applying the model to the acquired video, which includes real-life parking violations in three different locations, as shown in Fig. 6. Detailed information on the duration and description of the videos during the system implementation at each location is presented in Table 5.

Table 5 Duration and description of videos for system implementation

Place	Duration	Description
A	0:22:44	Features the parking lot in front of the Teknol Building of the Department of Informatics and Computer Engineering at the State University of Makassar, recorded from the 3rd floor.
B	0:10:48	Features the street area in front of a housing estate that is often bustling with daily activities, recorded from the 3rd floor of a resident's house.
C	0:18:51	Features the street area in front of At-Taubah Mosque, recorded from the 2nd floor of the mosque.

2.7. Evaluation

Evaluation, the process of measuring the performance and accuracy of a system [25], is performed using a confusion matrix to assess the effectiveness of the model in classifying parking, non-parking, and person objects. Additionally, the confusion matrix is used to measure the results of the system implementation. The components of the confusion matrix are presented in Table 6, which is then used to calculate the recall, precision, and F1-score values [26].

Table 6 Confusion matrix

		Predicted	
		Positive	Negative
Actual	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

True positive (TP) is the number of correct predictions for the positive class, while true negative (TN) is the number of correct predictions for the negative class. False positive (FP) is the number of false predictions for the positive class, and false negative (FN) is the number of false predictions for the negative class. Recall shows how many positive cases are found by the model, and it is calculated as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

Precision indicates how many of the model's positive predictions are correct, and it is calculated using:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

Meanwhile, the F1-score balances the two and provides a more comprehensive score of the model's performance, and it is calculated as:

$$\text{F1-score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{6}$$

3. Results and Discussion

The parking violation detection system is developed by building a model that can detect objects in the parking area. The detected objects include a person, a non-parked motorcycle, and a parked motorcycle. The sample dataset used in this study is shown in Fig. 7, which comprises two frames and their labels from videos taken with different brightness conditions.

Fig. 7 Example dataset of video frames for building a detection model

The results of the object annotations are shown in Fig. 7, where the non-parking object class is labeled as 0, the parking object as 1, and the person object as 2. To avoid class imbalance, certain objects were intentionally left unannotated. This approach was adopted to maintain a balanced dataset and prevent any class from dominating the annotations, which could negatively impact the model's performance during training. By selectively annotating the objects, a fairer distribution among all classes can be achieved. Additionally, since parking objects appear consistently in most frames, they were annotated differently across frames for variety. The labeling of parking and non-parking classes includes two different object orientations, vertical and horizontal, enabling the system to recognize all variations of object forms.

(a) Upward non-parking (b) Upward parking (c) Downward non-parking (d) Downward parking

Fig. 8 Vertical variations of non-parking and parking objects

(a) Right-facing non-parking (b) Right-facing parking

(c) Left-facing non-parking (d) Left-facing parking

Fig. 9 Horizontal variations of non-parking and parking objects

Vertical objects are categorized into two types: one when the object is facing up and the other when it is facing down, as shown in Figs. 8 and 9. Similarly, horizontal objects are also divided into two categories: one when the object is facing right and the other when it is facing left. After labeling or annotating all objects according to their respective classes, the model was trained using the YOLOv7 architecture. The results obtained at different epochs are presented in Table 7.

Table 7 Results of the model training experiments

Test		Epoch				
		13	28	35	40	45
non-parking (%)	TP	95	99	98	98	98
	Precision	92	95	95	95	95
	Recall	95	97	93	93	93
	map 50	96	97	97	97	97
	mAP 50:95	69	71	72	72	72
parking (%)	TP	94	98	98	98	98
	Precision	79	82	80	80	80
	Recall	81	83	88	88	88
	mAP 50	91	94	94	94	94
	mAP 50:95	64	69	70	70	70
person (%)	TP	91	95	92	92	92
	Precision	85	89	89	89	89
	Recall	84	84	83	83	83
	mAP 50	91	94	94	94	94
	mAP 50:95	46	51	50	50	50

The model has achieved decent performance in detecting objects for each class at the 13th epoch. At this point, the detection accuracy for each class exhibited satisfactory and stable results. However, when the number of epochs increased to 28, a significant improvement emerged in detection accuracy across all classes. This improvement indicated that the model continued to learn and enhance its ability to detect objects as the epochs increased. The accuracy of each class improved, demonstrating that the model became more effective and precise in recognizing patterns in the training data.

Furthermore, when the number of epochs increased to 35, the accuracy improvement was no longer maximized. Some indications are interpreted that the model's performance declined, particularly in the person class, with no improvement in accuracy observed at the 40th and 45th epochs. This decline was attributed to overfitting, where the model became too fitted to the training data, incurring decreased performance when handling new data [27]. As a result, the chosen model was trained up to the 28th epoch with the corresponding confusion matrix, as shown in Fig. 10.

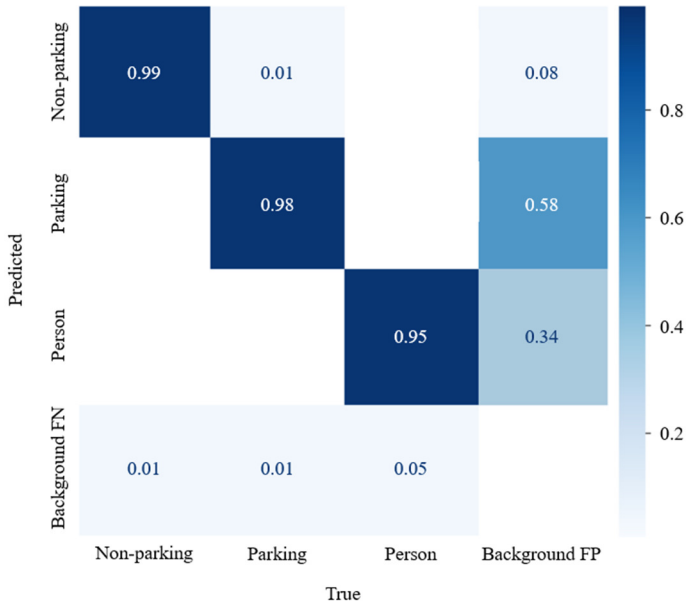


Fig. 10 Confusion matrix of the selected model at the 28th epoch

The non-parking class has the highest TP rate, correctly detecting objects with 99%, followed by the parking class at 98% and the person class at 95%. However, a weakness is observed in the prediction of background elements being incorrectly detected as objects, resulting in FP, especially in the parking and person classes, where FP values are relatively high. This high FP rate is due to the large number of parking and person objects in the dataset that actually exist but were not labeled to avoid an imbalance in the number of data annotations. Despite the high FP rate, the model performs well overall, as the TP values for each class are quite high, indicating robust detection capabilities. Background errors FP are primarily due to unlabeled objects in the dataset rather than inherent weaknesses in the model. Additionally, the background FN value, which represents errors when objects that should have been detected are incorrectly identified, as the background is relatively low. Such a result indicates that, despite some background errors, the model remains effective in detecting most relevant objects.

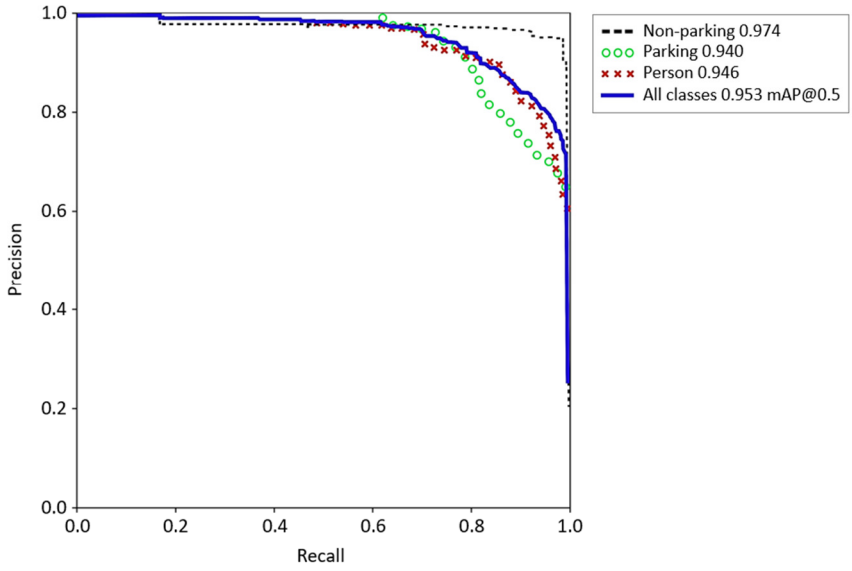


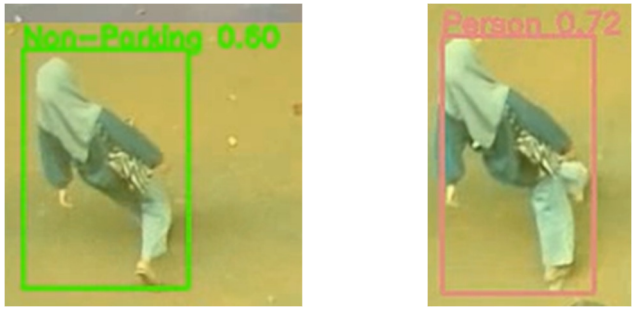
Fig. 11 Precision-recall curve result

Based on the precision-recall curve graph in Fig. 11, which illustrates the relationship between precision and recall, the mAP value at a threshold of 0.5 for all classes reaches 0.953. Such a finding reveals that the model is capable of identifying and classifying objects. The non-parking class demonstrates the highest performance, achieving an almost perfect precision-recall value of 0.974, as indicated by the curve nearly reaching the upper right corner. The parking class has a precision-recall value of 0.940, denoting that although its performance is slightly lower than non-parking, the model remains reliable in detecting parking objects. Meanwhile, the person class achieves a precision-recall value of 0.946, slightly above parking, indicating that the model is effective in detecting most person objects as well. Subsequently, the selected model was implemented on video with the results, as shown in Fig. 12.

Fig. 12 Object detection results using YOLOv7

At this stage, the model was utilized to detect and classify objects in video footage to evaluate the detection performance in more dynamic and realistic situations. The video implementation aims to assess how effectively the model can identify moving objects and tackle challenges such as lighting changes, varying viewing angles, and potential occlusion. Fig. 12 unveils the detection results identified by the model in a certain frame of the video. From these results, it can be seen that the model has been able to predict parking, non-parking, and person objects in the parking area. Initially, the model was designed to detect only two classes of objects: parking and non-parking.

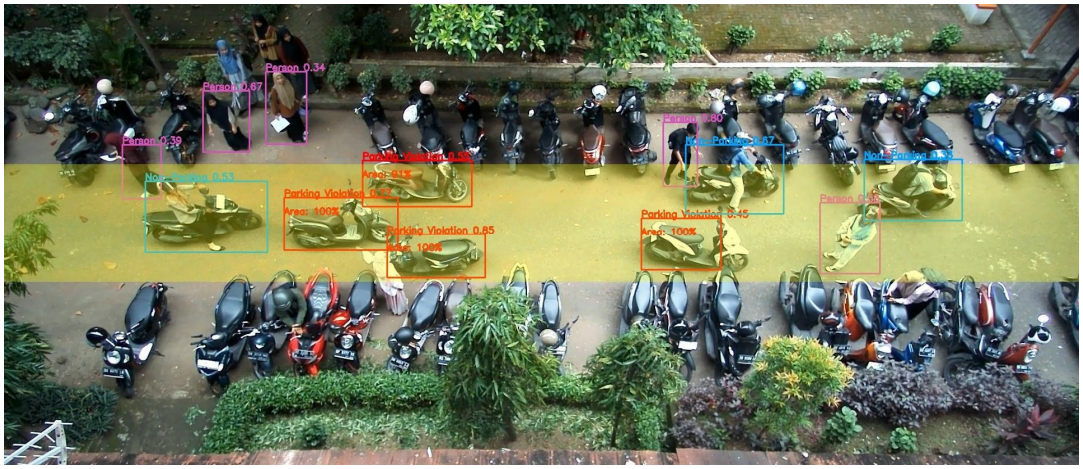
However, a significant issue was found during initial testing, where the model frequently misclassified passers-by in the parking area as non-parking objects, as shown in Fig. 13(a). This phenomenon yielded many inaccurate detections, particularly in situations with high human activity around the parking area. This error indicates that the model requires improvement to effectively differentiate between parking, non-parking, and people moving around. Therefore, a new class was added to the model, i.e., the person class. After the addition of the person class, the model was re-implemented and tested on the same data, resulting in a significant improvement in detection accuracy. Fig. 13(b) shows the updated detection result, where a human walking in the parking lot is now successfully detected and correctly identified as a person.



(a) Wrong detection (b) Correct detection

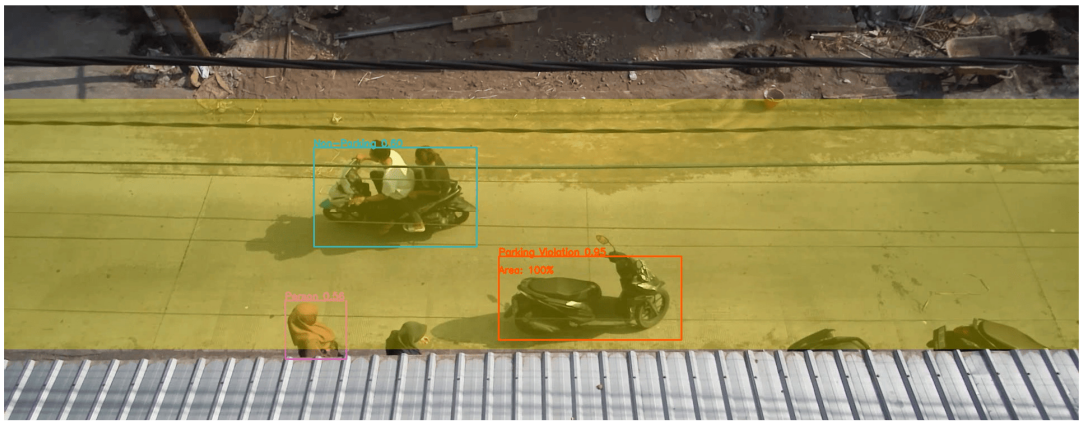
Fig. 13 Model detection results before and after adding person class

This breakthrough demonstrates that adding the person class enhanced the ability of the model to distinguish between different objects in the parking lot environment. Furthermore, the implementation of the system to detect motorcycle parking violations using the proposed method, which includes the formation of ROI and the calculation of the parking object area, was also successfully carried out, as shown in Fig. 14. The motorcycle parking violation detection system demonstrates effective performance across the three different locations. Specifically, parking objects were successfully detected as parking violations when the area reached 50% or more while within the ROI. Additionally, non-parking objects were correctly identified as not committing parking violations when inside the ROI. This significantly accurate classification of non-parking objects reflects the ability of the system to differentiate between designated parking areas and areas where parking is prohibited.

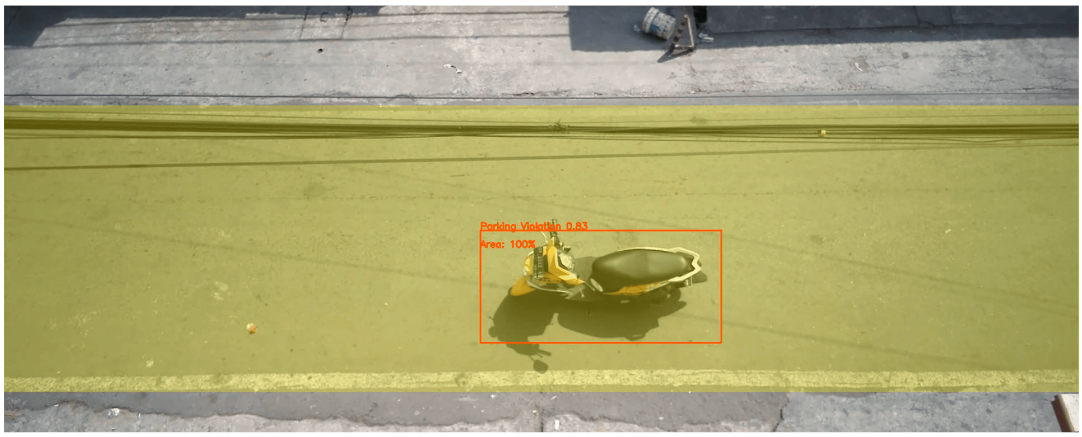


(a) Place A

Fig. 14 System implementation results for detecting motorcycle parking violations in three different locations



(b) Place B



(c) Place C

Fig. 14 System implementation results for detecting motorcycle parking violations in three different locations (continued)

In the following phase, a system evaluation was conducted to validate the performance of the system in detecting motorcycle parking violations in three different locations. The system evaluation compared the proposed method, which utilizes object area calculation for detected parking objects within the ROI, with a scenario that does not use area calculation. Using object area calculation achieves an average recall of 91.5%, significantly higher than the 78.6% recall without object area calculation, as shown in Table 8.

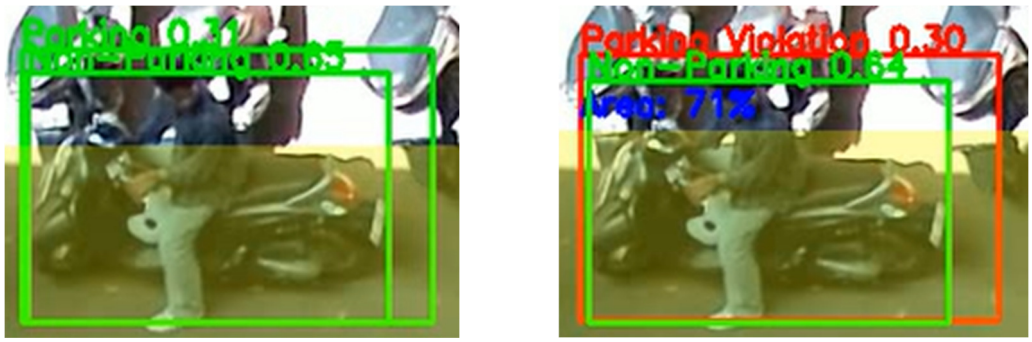
Table 8 Comparison of the system evaluation with and without object area calculation

Place	Real	With object area calculation					Without object area calculation				
		System (TP)	Error detection		Recall (%)	Precision (%)	System (TP)	Error detection		Recall (%)	Precision (%)
			FN	FP				FN	FP		
A	37	35	2	14	94.6	71.4	28	9	4	75.7	87.5
B	5	5	0	2	100	71.4	4	1	1	80	80
C	5	4	1	1	80	80	4	1	1	80	80
Average					91.5	74	Average			78.6	82.5
F1-score (%)					82.7		F1-score (%)			80.5	

This difference is due to certain parking objects whose bounding box points were not entirely within the ROI, incurring their failure to be detected as violations as shown in Fig. 15(a). However, by applying the area calculation for parking objects, as illustrated in Fig. 15(b), an object with more than 50% of its area within the ROI was successfully detected as a parking violation. However, the average precision values in Table 8 yield the opposite result. Using object area calculation results in a lower precision of 74%, compared to 82.5% when object area calculation is not applied. As illustrated in Fig. 16(a), a non-parking object detected when parking was not classified as a violation due to two of its bounding boxes that were outside the ROI. Conversely, Fig. 16(b) shows a misclassified object that was still recognized as a parking violation because more than 50% of its area was within the ROI.

(a) Wrong detection (b) Correct detection

Fig. 15 Detection results without and with the application of the object area calculation method



(a) Detected as not committing a violation (b) Detected committing a violation

Fig. 16 Double detection results without and with the application of the object area calculation method

Based on the results obtained, the use of object area calculation yields an overall higher F1-score of 82.7%, compared to 80.5% without using object area calculation. Therefore, it can be concluded that the proposed method of calculating the parking object area within the ROI enhances the effectiveness of the system in detecting parking violations, despite the increase in the number of non-violating objects that were incorrectly identified as violations. The system implementation used a confidence threshold set at 0.3, enabling the mis-detected object in Fig. 16 to still be considered valid due to its confidence score of 0.3. Therefore, a different scenario was tested by setting the confidence threshold to 0.5.

Table 9 System evaluation with confidence threshold set at 0.5

Place	Real	System (TP)	Error detection		Recall (%)	Precision (%)
			FN	FP		
A	37	33	4	6	89.2	84.6
B	5	5	0	1	100	83.3
C	5	4	1	1	80	80
Average					89.7	82.6
F1-score (%)					86.2	

Based on the results presented in Table 9, the average recall value achieved is 89.7%, reflecting a decrease compared to the results in Table 8 with object area calculation. This reduction occurred because two objects that committed parking violations were not detected due to having a confidence score of less than 0.5. However, the recall still surpassed the average recall from Table 8 without object area calculation. The average precision obtained is 82.6%, which is higher than the average precision in Table 8. Therefore, setting the confidence threshold at 0.5 is deemed effective, as evidenced by the F1-score which reached a higher value of 86.2%.

4. Conclusions

From the research conducted on the motorcycle parking violation detection system, the use of the YOLOv7 for classifying non-parking, parking, and person objects achieved its optimal performance at the 28th epoch with an mAP score of 0.953 at a threshold of 0.5. The classification model is designed to expeditiously detect parking violations when a parking object is

located within a parking restriction area or ROI. A detected parking object is considered a violation if its area within the ROI reaches 50% or more. By deploying this object area calculation, the F1-score increased to 82.7%, higher than the 80.5% F1-score obtained without the object area calculation. Additionally, the model demonstrated more effective performance in detecting violations with a confidence threshold of 0.5, resulting in a recall of 89.7%, precision of 82.6%, and an F1-score of 86.2%. The fast object detection capability of YOLOv7, along with its ability to detect multiple objects simultaneously in a single image enables the parking violation system to work efficiently, even in dense parking environments across three different locations. Future developments could focus on nighttime detection by training the model on a larger dataset and incorporating the utilization of infrared cameras.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] D. Lin and J. Cui, "Transport and Mobility Needs for an Ageing Society From a Policy Perspective: Review and Implications," International Journal of Environmental Research and Public Health, vol. 18, no. 22, article no. 11802, 2021.

[2] P. Ribeiro, G. Dias, and P. Pereira, "Transport Systems and Mobility for Smart Cities," Applied System Innovation, vol. 4, no. 3, article no. 61, 2021.

[3] K. Mouratidis, S. Peters, and B. van Wee, "Transportation Technologies, Sharing Economy, and Teleactivities: Implications for Built Environment and Travel," Transportation Research Part D: Transport and Environment, vol. 92, article no. 102716, 2021.

[4] M. Z. Irawan, P. F. Belgiawan, A. K. M. Tarigan, and F. Wijanarko, "To Compete or Not Compete: Exploring the Relationships Between Motorcycle-Based Ride-Sourcing, Motorcycle Taxis, and Public Transport in the Jakarta Metropolitan Area," Transportation, vol. 47, no. 5, pp. 2367-2389, 2020.

[5] I. Sefriyadi, I. G. A. Andani, A. Raditya, P. F. Belgiawan, and N. A. Windasari, "Private Car Ownership in Indonesia: Affecting Factors and Policy Strategies," Transportation Research Interdisciplinary Perspectives, vol. 19, article no. 100796, 2023.

[6] H. Udjari, M. Warka, S. Suhartono, and O. Yudianto, "Legal Protection Against Children Who Are Passed On in the Transportation of Two - Wheeled Vehicles on the Highway," Research, Society and Development, vol. 9, no. 11, article no. e63191110374, 2020.

[7] BPS Indonesia, Statistical Yearbook of Indonesia 2024, Jakarta: BPS-Statistics Indonesia, 2024.

[8] K. Sumit, V. Ross, K. Brijs, G. Wets, and R. A. C. Ruiter, "Risky Motorcycle Riding Behaviour Among Young Riders in Manipal, India," BMC Public Health, vol. 21, article no. 1954, 2021.

[9] S. Yu and W. D. Tsai, "The Effects of Road Safety Education on the Occurrence of Motorcycle Violations and Accidents for Novice Riders: An Analysis of Population-Based Data," Accident Analysis & Prevention, vol. 163, article no. 106457, 2021.

[10] F. Fatmasari and N. W. R. Mumpuni, "Effectiveness of the Application of Administrative Criminal Sanctions Against Illegal Parking in the Malioboro Area," Literasi Hukum, vol. 7, no. 2, pp. 107-120, 2023.

[11] N. Y. Loong, A. Jamaludin, and R. A. Hashim, "Factors Affecting the Behavioral Intention to Park Legally Among Urban Malaysian in Kuala Lumpur, Malaysia," Asian Journal of Social Sciences and Management Studies, vol. 10, no. 1, pp. 43-51, 2023.

[12] G. Hendrawan, Y. P. Siregar, and G. Widiartana, "The Impact of Illegal Parking on Traffic Connection along Pasar Kembang Yogyakarta Road: Problems and Solutions," International Journal of Social Science and Human Research, vol. 6, no. 12, pp. 7951-7957, 2023.

[13] R. A. Hamzah, C. Setianingsih, R. A. Nugrahaeni, S. R. Hanafia, and F. Fuadi, "Parking Violation Detection on the Roadside of Toll Roads With Intelligent Transportation System Using Faster R-CNN Algorithm," 6th International Conference on Informatics and Computational Sciences, pp. 169-174, 2022.

[14] Q. Yang and L. Yu, "Recognition of Taxi Violations Based on Semantic Segmentation of PSPNet and Improved YOLOv3," Scientific Programming, vol. 2021, article no. 4520190, 2021.

[15] R. Akhawaji, M. Sedky, and A. H. Soliman, "Illegal Parking Detection Using Gaussian Mixture Model and Kalman Filter," IEEE/ACS 14th International Conference on Computer Systems and Applications, pp. 840-847, 2017.

[16] C. K. Ng, S. N. Cheong, and Y. L. Foo, "Lightweight Deep Neural Network Approach for Parking Violation Detection," Proceedings of the 2018 VII International Conference on Network, Communication and Computing, pp. 332 - 337, 2018.

[17] N. Fadilah, S. Y. Soon, and H. Radi, "Embedded Automated Vision for Double Parking Identification System," Indonesian Journal of Electrical Engineering and Computer Science, vol. 10, no. 3, pp. 1221-1226, 2018.

[18] T. Ludwisiak and M. Mazur-Milecka, "Automated Parking Management for Urban Efficiency: A Comprehensive Approach," 16th International Conference on Human System Interaction, pp. 1-4, 2024.

[19] N. Hernández-Díaz, Y. C. Peñaloza, Y. Y. Rios, J. C. Martinez-Santos, and E. Puertas, "A Computer Vision System for Detecting Motorcycle Violations in Pedestrian Zones," Multimedia Tools and Applications, in press. https://doi.org/10.1007/s11042-024-19356-9

[20] J. Wang, Z. Chen, P. Li, B. Sheng, and R. Chen, "Real-Time Non-Motor Vehicle Violation Detection in Traffic Scenes," IEEE International Conference on Industrial Cyber Physical Systems, pp. 724-728, 2019.

[21] K. Maharana, S. Mondal, and B. Nemade, "A Review: Data Pre-Processing and Data Augmentation Techniques," Global Transitions Proceedings, vol. 3, no. 1, pp. 91-99, 2022.

[22] C. Dewi, R. C. Chen, Y. C. Zhuang, X. Jiang, and H. Yu, "Recognizing Road Surface Traffic Signs Based on Yolo Models Considering Image Flips," Big Data and Cognitive Computing, vol. 7, no. 1, article no. 54, 2023.

[23] N. Chitraningrum, L. Banowati, D. Herdiana, B. Mulyati, I. Sakti, A. Fudholi, et al., "Comparison Study of Corn Leaf Disease Detection based on Deep Learning YOLO-v5 and YOLO-v8," Journal of Engineering and Technological Sciences, vol. 56, no. 1, pp. 61-70, 2024.

[24] M. Paul, P. K. Podder, and M. R. Hassan, "Eye Tracking, Saliency Modeling and Human Feedback Descriptor Driven Robust Region-of-Interest Determination Technique," IEEE Access, vol. 10, pp. 98612-98624, 2022.

[25] N. Chandrasekhar and S. Peddakrishna, "Enhancing Heart Disease Prediction Accuracy Through Machine Learning Techniques and Optimization," Processes, vol. 11, no. 4, article no. 1210, 2023.

[26] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," Augmented Human Research, vol. 5, no. 1, article no. 12, 2020.

[27] Y. Peng and M. H. Nagata, "An Empirical Overview of Nonlinearity and Overfitting in Machine Learning Using COVID-19 Data," Chaos, Solitons & Fractals, vol. 139, article no. 110055, 2020.