International Journal of Transport Development and Integration
Vol. 9, No. 2, June, 2025, pp. 325-335
Journal homepage: http://iieta.org/journals/ijtdi

Mobile Phone Usage Warning System for Driver Focus Monitoring

Amanda Keshya Anggara, Prajna Deshanta Ibnugraha*, Simon Siregar, Anak Agung Gde Agung, Devie Ryana Suchendra

School of Applied Science, Telkom University, Bandung 40257, Indonesia

Corresponding Author Email: prajna@telkomuniversity.ac.id

https://doi.org/10.18280/ijtdi.090210

Received: 20 March 2025
Revised: 15 May 2025
Accepted: 21 May 2025
Available online: 30 June 2025

Keywords:
computer vision, driver behavior, handphone detection, real-time detection, YOLO (you only look once)

ABSTRACT

Driver distraction, particularly due to mobile phone usage, significantly contributes to road traffic accidents. This study proposes a real-time detection system using the YOLOv8 object detection algorithm to identify drivers using mobile phones. The system combines two datasets: one for phone usage behavior and another for phone object detection, aiming to improve recognition performance in various conditions. Data augmentation techniques such as zoom, blur, and noise were applied to simulate real-world scenarios including lighting variations and occlusion. The YOLOv8 model was trained and evaluated using this dataset combination, achieving a detection accuracy of 92.5% and a mean average precision (mAP@0.5) of 89.5%. These results demonstrate the model's ability to accurately detect mobile phone usage, even under challenging conditions. This approach presents a promising solution for early warning systems to monitor driver focus and reduce the risk of accidents caused by distraction, contributing to improved road safety through intelligent driver behavior detection.

1. INTRODUCTION

Today, technology is a key cornerstone in improving efficiency, safety, and driver readiness in various aspects of daily life. The rapid evolution of technology drives the need for continuous innovation to maintain excellence in various fields [1]. One solution that is becoming increasingly relevant is the development of intelligent technology-based systems. These systems use intelligent algorithms to automate the operation of tools or equipment, reducing the need for manual intervention by users. With this approach, smart technology not only offers efficiency, but also represents a significant step forward in the field of automation, allowing for more seamless integration into daily human routines and activities [2].

Smart systems have had a significant impact on various aspects of life, including smart homes, smart cities, and smart security systems. The application of these technologies improves efficiency, convenience, and safety in various human activities [3]. In traffic safety, intelligent systems play an important role in detecting and preventing risky behaviors, such as phone use while driving. Using intelligent algorithms, this technology can detect actions that may reduce a driver's concentration, such as holding a mobile phone or making a phone call. The implementation of these detection systems is not only relevant to improving driver safety but is also an important part of developing a smarter transportation infrastructure that can adapt to the increasingly complex challenges of the future.

Ensuring safety is a very important aspect of road safety. Every policy and action of the driver must follow safety protocols to minimize the risk of accidents [4]. According to the World Health Organization (WHO), drivers who use cell phones while driving have about four times higher risk of accidents compared to those who do not use cell phones [5]. Existing detection systems lack robustness in real-world scenario. These systems often perform inadequately under low-light conditions. They also struggle to accurately identify overlapping objects. These limitations emphasize the need for an intelligent, real-time system that accurately identifies mobile phone usage behavior while driving.

This study addresses those limitations by proposing a mobile phone usage detection system powered by YOLOv8, a deep learning-based object detection algorithm. Unlike its predecessors such as YOLOv5 or YOLOv7, YOLOv8 introduces advanced architectural components like the Efficient Layer Aggregation Network (ELAN), enabling better detection of small, overlapping objects with improved inference time and precision. These advantages make YOLOv8 well-suited for the challenges of real-time driver monitoring in dynamic environments.

There are several studies dedicated to the detection of driver behavior, particularly as it relates to distraction while driving. These studies attempt to provide detailed descriptions of the behavioral patterns of distracted drivers, including major categories such as visual, manual, and cognitive distraction [6]. Visual distraction occurs when the driver takes his or her eyes off the road, such as looking at the GPS or dashboard screen. Manual distraction involves activities that take the driver's hands off the wheel, such as holding food or adjusting the head unit. Meanwhile, distractions such as talking to passengers or thinking about personal matters [7].

This research aims to make a significant contribution to the technology of detecting the use of mobile phones while driving by overcoming the limitations of existing systems while improving the safety of drivers and other road users. By using advanced deep learning algorithms such as YOLOv8 and various data processing approaches, the results of this research are expected to provide new insights and become an important reference for the development of future driver behavior detection technologies.

The following section of this paper outlines the methodology used in this study, followed by the results and analysis, and a discussion of the findings.

2. RESEARCH AND METHODOLOGY

This section provides a detailed explanation of the research and methodology used. This methodology is a guide to the preparation of this research with several stages consisting of literature review, establishing a test strategy, conducting tests and performing analysis, and drawing conclusions based on the test results.

2.1 Literature review

When developing a model to detect distracted driving behaviors, such as phone use while driving, it is important to carefully consider and explore the different stages. According to Ma et al. [8], the development of detection models should consider many factors, such as driver behavioral characteristics, appropriate experimental conditions, and efficient integration of algorithms such as YOLOv8 with MHSA (Multi-Head Self-Attention) attention mechanisms. This optimization aims to improve detection accuracy, ensure computational efficiency, and enable real-time implementation in edge devices such as Jetson Nano or Raspberry Pi.

In such studies, it is important to have a thorough understanding of the specific requirements for detecting distracted driver behavior, such as phone use. This includes a thorough analysis of the relevant environmental and experimental conditions, as well as the characteristics of the driver's behavior to be detected. In addition, the algorithms used, such as YOLOv8 enhanced with attention mechanisms, must be designed to accurately and efficiently capture the dynamic nature of the detected objects in both test scenarios and real-time applications [9].

The development of the distracted driver behavior detection model consists of several main phases. The first phase involves identifying the characteristics of the driver's behavior, the nature of the distraction, and the relevant environmental conditions. Processing methods such as the integration of YOLOv8 with MHSA mechanism were chosen to improve the accuracy of behavior detection.

The next step is to implement the system in appropriate hardware, followed by validation through testing with real data. The test results compared different conditions such as phone use, fatigue, or other distractions. The integration of attention-based algorithms showed a significant improvement in the accuracy of distracted driving detection under varying driving conditions.

Identifying the behavioral characteristics of distracted drivers, such as phone use while driving, can be based on literature data and analysis of detection system specifications. Algorithms and processing methods, such as YOLOv8 with attention mechanism, should be carefully selected to ensure detection accuracy. The next process involves the design and implementation of a computer vision-based detection system that is integrated into the hardware to detect the driver's behavior in real time and provide warnings when necessary.

Following the development phase of the distracted driving detection system, the prototype system will enter the testing and characterization phase. In this phase, experiments will be conducted to determine the accuracy of detection of driver phone use and to validate the detection range. The detection model will be validated by evaluating its ability to accurately detect distracted behavior against other detection models that are accurate, using analytical calculations and theoretical modeling to ensure its effectiveness.

Some approaches to detecting driver distraction involve the use of smartphone sensors and AI (Artificial Intelligence) systems. Smartphone-based systems can use sensors such as accelerometers, gyroscopes, and magnetometers to detect changes in position or hand movements that characterize distracted behavior, including the use of a mobile phone [10]. Use of vehicle-mounted sensors or other devices, such as smartphones, to detect changes in driver behavior that may impair their ability to respond to dangerous situations [11].

Another approach is to develop driver behavior monitoring technologies to identify and assess significant distractions caused by driver behavior, such as the use of electronic devices while driving. Technologies that use a variety of sensors and wearable devices to detect deviations in the vehicle and in the driver's head movements to warn the driver of potential hazards [12]. Use motion sensors and the IMU (Inertial Measurement Unit) to track the driver's behavior in real time. This helps identify normal and abnormal behavior while driving [13]. Utilizing sensors to monitor the driver's physical and psychological state [14].

The implementation used in this study uses a system that detects when a phone is being held. This system works when a phone is visible in the frame and under conditions such as holding a phone or making a call. This condition can be influenced by the presence of sufficient light. That is, if there is enough light, the detection process will run smoothly and have high accuracy. This idea can be proven in several studies on this topic, such as research conducted by Peng et al. [15]. Their research introduces NLE-YOLO, a low-light targeting detection network developed based on YOLOv5 to overcome the problems of insufficient illumination and noise interference. In addition, YOLOv7 was trained and validated using the ExDark dataset to detect twelve classes in a dark environment [16]. The results showed a significant improvement in mAP (Mean Average Precision) compared to previous studies using the same dataset.

Efficiently Expanded Layer Aggregation (E-ELAN) technology developed in the YOLOv8-based detection system. In this study, the system is proposed to be optimized for low light environments. This detection model is designed to mimic real-world conditions where varying lighting levels affect the detection accuracy. To achieve the best performance, the incorporation of multi-scale features in E-ELAN is used to improve sensitivity to small objects. In contrast, the initial layers of the model, such as CSP, serve as an efficient base feature processing platform to support generalization capabilities under different lighting conditions.

The E-ELAN architecture can be seen in Figure 1, which is an explanation of the architecture.

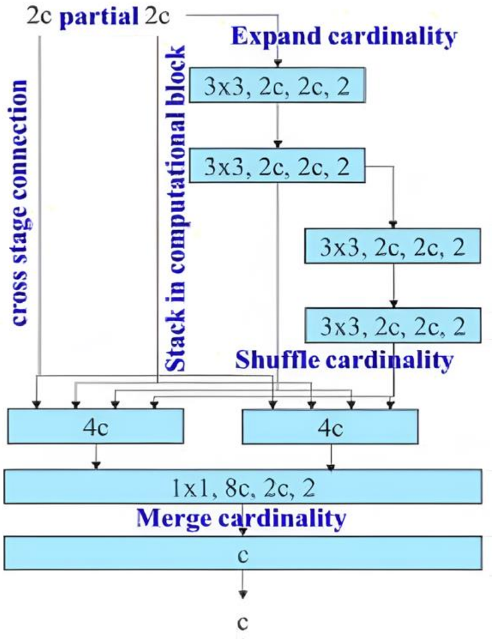


Figure 1. The architecture of E-ELAN [16]

In another research by Yu and Choi [17], the focus of this work is to provide information about the detection results. Figure 2 shows an approach that aims to improve risk assessment in autonomous driving by providing in-depth information for each detected object. Depiction of an object detector with depth estimation using images from a monocular camera.

Another research conducted is to create a framework for real-time object recognition on mobile devices through the co-design of compression and compilation by introducing YOLO bile [18], which is used to develop phone detection applications using mobile devices. This research discusses a real-time mobile phone usage monitoring system using the YOLOv5 algorithm [19]. In addition, this research was conducted to define various techniques to specifically detect objects running in real time with YOLO-LITE without using the GPU [20].

There are also other studies focusing on vehicle detection with YOLOv4 and vehicle tracking with DeepSORT, with models trained on specialized datasets including motorcycles, cars, buses, and trucks, using TensorFlow as the main platform and CUDA for computation [21]. Detection of vehicles against the direction in real time using computer vision based methods, with three main detections, namely, detection of vehicles with video recordings using the YOLOv3 algorithm to generate bounding boxes, tracking vehicles within a certain area using the centroid tracking algorithm, then identifying vehicles moving in the wrong direction by analyzing changes in centroid position in each frame, that when a vehicle is detected against the direction, the system will automatically capture its image for further documentation [22].

To detect mobile phone use in a video or image in real time, the YOLOv8 is used [19]. YOLO is a deep learning-based object recognition system that can quickly and accurately recognize and classify objects. The system works by training a model on a dataset containing images of people using a phone, allowing the model to recognize visual patterns such as hand position [23], device shape [24], and user interaction [25]. With this approach, mobile phone use can be efficiently detected under different lighting conditions and shooting angles.

The advantage of the YOLOv8 algorithm is its efficiency and accuracy in detecting objects in real time. This capability allows for fast response to changes in position and user interaction with devices such as phones. This makes YOLOv8 very suitable for applications that require fast and accurate monitoring of user activity. In addition, YOLOv8 is known for its easy integration into various computer vision-based systems and its support for various hardware, including those with limited resources [26]. Despite its significant advantages, this algorithm has limitations. One of these is the need for a large and diverse data set to achieve optimal results, as well as the challenge of detecting objects under poor lighting conditions or other environmental disturbances. Therefore, it is important to consider these elements when designing a YOLOv8-based system to ensure reliable results.

The YOLOv8 algorithm is available in several model and size configurations that can be customized to meet application-specific needs. This algorithm is suitable for use under predefined conditions. The research uses a YOLOv8 model with an architecture optimized for detecting small objects, such as mobile phones. With the ability to accurately detect objects within a wide field of view, YOLOv8 provides the flexibility to cover different user scenarios. The model is able to capture visual data from a relatively large area in a single observation frame, enabling detection of mobile devices and user interactions with better coverage than traditional object detection methods [27].

Another study [28] discussed the Automatic Passenger Counting (APC) system developed using computer vision with the Haar Cascade algorithm on the Raspberry Pi processing unit to detect passengers, including those wearing glasses and masks. The test results show an accuracy value of 60% with certain scaleFactor, minNeighbors, and minSize. The best results were obtained with the minimum values of the scaleFactor a minNeighbors variables. This system can be integrated with EDR (Event Data Recorder) to provide real-time vehicle information.

Based on Figure 2, the YOLOv8 architecture consists of three main components: Backbone, Neck-PAN, and Head. Backbone is responsible for extracting features from the input image, Neck-PAN integrates features at different scales (small, medium, and large), while the Head generates the final prediction in the form of bounding box, confidence score, class probability, and depth. In the context of mobile phone usage detection, this system acts as a receiver of visual signals. The Backbone acts as a dermis layer that captures the initial information, the Neck-PAN processes the information in a structured manner, and the Head acts as a decision layer that determines the presence, object type, and user interaction.

The system involves various key components, including performance evaluation through precision, recall, and accuracy metrics to ensure system reliability. In terms of hardware, cameras, processing units, and IoT devices are used to support the detection process. Frameworks such as TensorFlow and OpenCV are applied in data processing, while methods include YOLOv8 algorithm, data augmentation, preprocessing, transfer learning, and post-processing to improve system performance. Data management includes data collection, annotation and storage to ensure the completeness and quality of the data used. The focus of detection is to identify the presence of a phone and its use by the driver as a key element of the system. This system enables accurate, real-time detection of the user's interaction with the mobile phone.

Figure 3 shows the overall mind map of the mobile phone usage detection architecture.

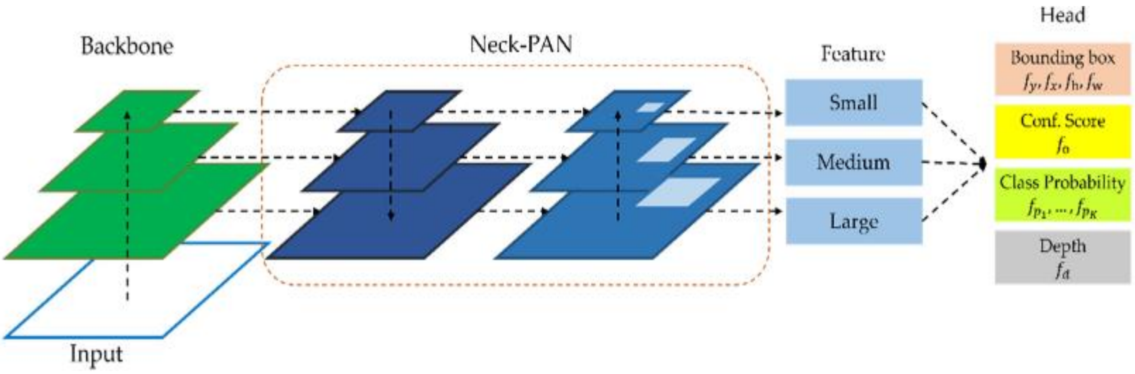


Figure 2. Network detection model [17]

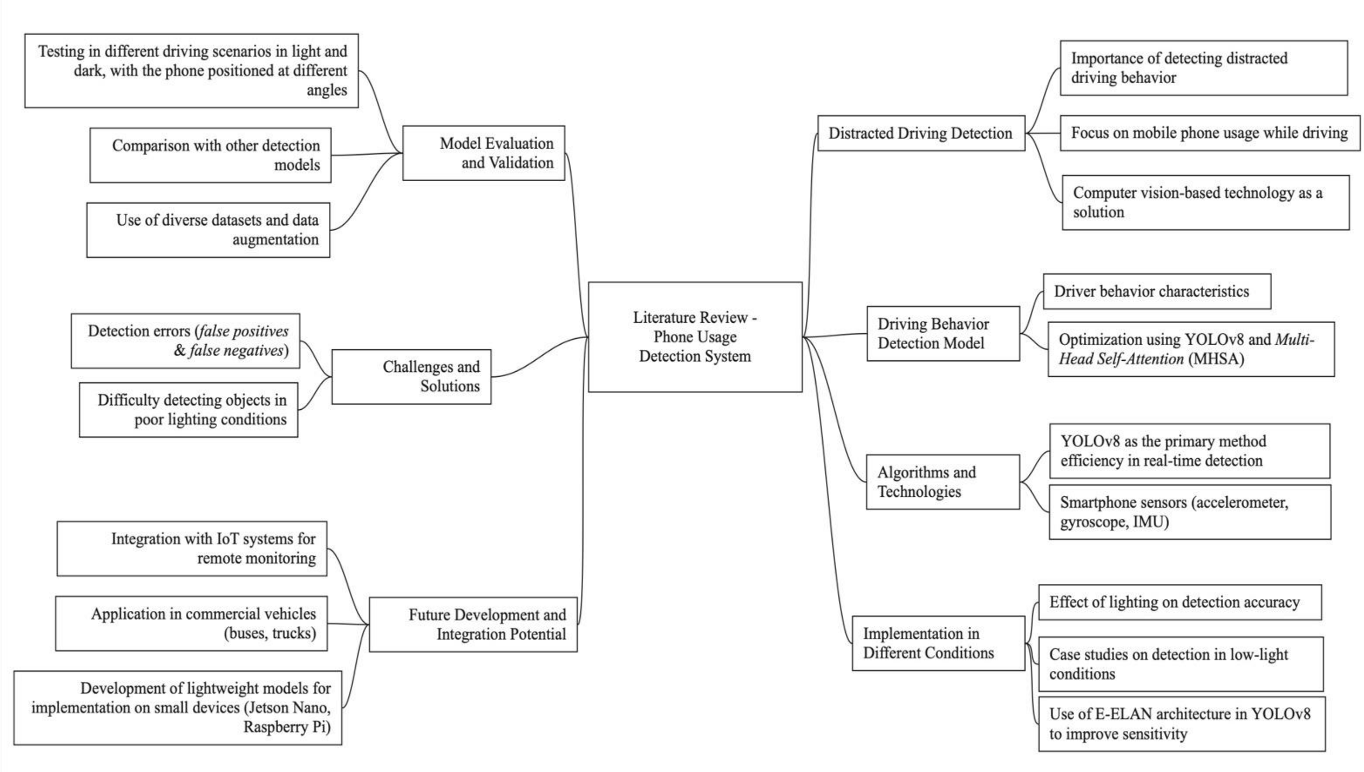

Testing in different driving scenarios in light and dark, with the phone positioned at different angles
Comparison with other detection models
Use of diverse datasets and data augmentation
Model Evaluation and Validation
Detection errors (*false positives* & *false negatives*)
Difficulty detecting objects in poor lighting conditions
Challenges and Solutions
Integration with IoT systems for remote monitoring
Application in commercial vehicles (buses, trucks)
Development of lightweight models for implementation on small devices (Jetson Nano, Raspberry Pi)
Future Development and Integration Potential
Literature Review - Phone Usage Detection System
Distracted Driving Detection
Importance of detecting distracted driving behavior
Focus on mobile phone usage while driving
Computer vision-based technology as a solution
Driving Behavior Detection Model
Driver behavior characteristics
Optimization using YOLOv8 and *Multi-Head Self-Attention* (MHSA)
Algorithms and Technologies
YOLOv8 as the primary method efficiency in real-time detection
Smartphone sensors (accelerometer, gyroscope, IMU)
Implementation in Different Conditions
Effect of lighting on detection accuracy
Case studies on detection in low-light conditions
Use of E-ELAN architecture in YOLOv8 to improve sensitivity

Figure 3. Previous studies of phone detection system

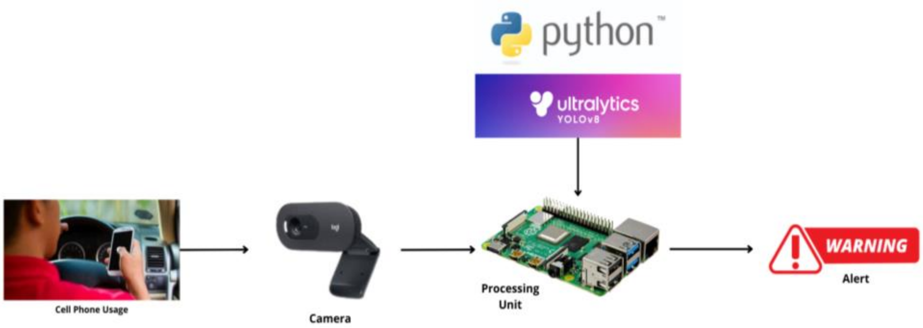


Figure 4. Mobile phone usage detection architecture system

Compared to earlier detection systems based on YOLOv5 and YOLOv7, YOLOv8 offers a more efficient architecture and improved accuracy for small-object detection. YOLOv8 integrates an ELAN, which enhances feature fusion and spatial representation, allowing it to detect phone usage even when there is partial occlusion or overlapping with the hand. These improvements are crucial for accurate real-time driver behavior monitoring.

2.2 System architecture

This study utilizes two datasets: A mobile phone usage dataset and a mobile phone object dataset. The phone usage dataset includes labeled instances of drivers using or not using phones in real vehicle settings. The phone object dataset contains diverse images of mobile phones under various angles and lighting conditions. Both datasets are representative of common in-cabin driver scenarios and were selected for their relevance to the real-world detection task.

Figure 4 is the system architecture that will be developed in this research. This system will work and be installed on the dashboard of the vehicle. The system uses the YOLOv8 algorithm with images or videos from the cabin camera processed through the ME-YOLOv8 pipeline to detect distraction behavior or signs of driver fatigue. The input data is optimized by pre-processing such as resizing to a resolution of 640×640 pixels and data enhancement [29].

This driver behavior detection system is designed to improve driving safety, with a focus on detecting phone use while driving. The system uses an in-cab camera aimed directly at the driver to monitor driver activity in real time. The camera captures visual data indicative of phone use, such as making a call or holding a phone. The data is then processed by the YOLOv8 algorithm running on the processing unit.

The system uses the YOLOv8 algorithm as the core of data processing because of its advantages in object detection speed and accuracy. YOLOv8, the latest development of the YOLO family of algorithms, is designed to detect and analyze objects in video in real time without requiring large computing resources. It provides reliable detection even under complex environmental conditions, such as lighting variations in a vehicle cabin. In addition, YOLOv8 supports multi-object identification, enabling the system to detect interactions between the driver's hands, face, and phone with high accuracy. Using YOLOv8, the system is not only fast, but also lightweight enough to run on even low-spec devices, making it a cost-effective yet effective solution for everyday in-vehicle use.

2.3 System workflow

The functionality of the mobile phone usage detection system can be illustrated in the workflow described in Figure 5.

The diagram above explains the working process of the driver's phone usage detection system, which aims to monitor and improve driving safety. The process starts with input from a camera that records the driver's condition in real-time. The visual data obtained is processed using the YOLOv8 algorithm to detect the presence of objects such as mobile phones in the driver's area. This algorithm was chosen due to its ability to accurately recognize various objects even in diverse lighting conditions.

Once the phone is detected, the next step is to evaluate the position of the object. The system checks whether the phone is in the driver's face or hand area, which indicates that the phone is in use. To ensure accuracy, the system also checks the trust score value generated by YOLOv8. This value indicates the level of confidence in the detection, and only objects with a confidence score greater than 40% are processed further. This 40% threshold was selected based on an internal ROC curve analysis, which showed an optimal trade-off between true positives and false positives at this point. Additionally, other works on YOLOv8 for real-time detection often use similar confidence levels. This step is designed to minimize detection errors, such as misidentifying other objects like phones.

If all conditions are met–the phone is detected, its position is in the relevant area, and the trust score is high enough–the system will act by alerting the driver. This warning can be an audible alarm, vibration, or visual notification, depending on the system configuration. The main objective of this process is to increase driver awareness of the dangers of using phones while driving, while helping to reduce the risk of traffic accidents due to visual or attention distractions. The system is designed as a preventive measure to improve road safety.

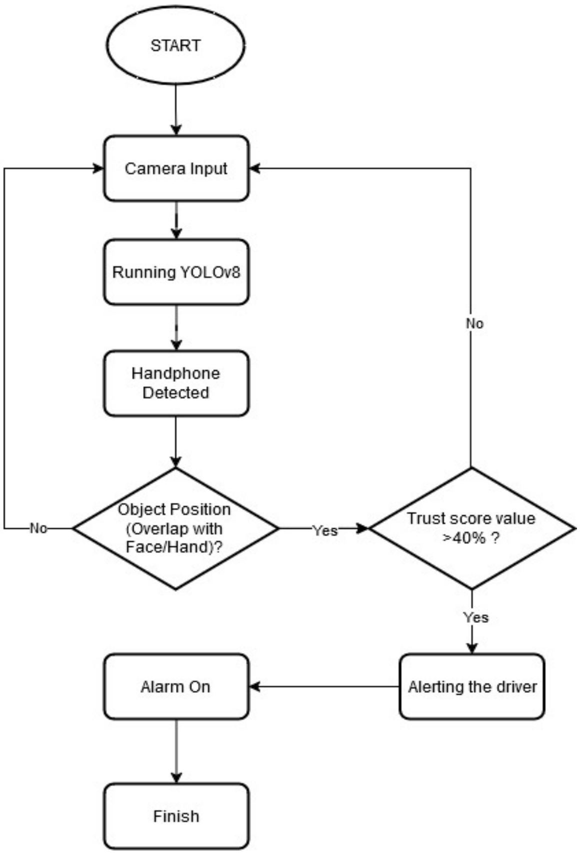


Figure 5. Flow chart of phone usage detection

2.4 Data processing and augmentation

Prior to training, all images were preprocessed and augmented. Preprocessing steps included resizing to 640x640 pixels and applying auto-orientation. Augmentation involved horizontal and vertical flips, random crops (0-12% zoom), blurring (up to 1.9px), and noise addition (up to 1.96%). These parameters were selected to simulate real-world distortions such as motion blur, occlusion, and variable lighting. The augmented dataset improved the model's robustness and generalization to diverse driving conditions.

Before conducting the test, two main datasets were prepared, namely the mobile phone usage dataset and the mobile phone dataset. The mobile phone usage dataset includes images of the driver holding a mobile phone, while the mobile phone dataset contains images of mobile phone objects in various angles and lighting conditions. The data was obtained from several sources, including real images taken directly and additional datasets uploaded through platforms such as Roboflow for augmentation. The annotation process was done manually by marking the relevant area of the object using a bounding box.

The details of the phone usage dataset and phone object dataset are presented in Table 1 and Table 2. These datasets were processed using various augmentation techniques, as listed in Table 3, to enhance robustness under different lighting and perspective conditions.

Table 1. Dataset of mobile phone usage

Class Name	Total Count	Training Count	Validation Count	Test Count
Calling	584	394	134	56
Not_calling	432	317	67	48
Notuse_phone	245	173	49	23

Table 2. Phone dataset

Class Name	Total Count	Training Count	Validation Count	Test Count
Handphone	1380	965	279	136

Table 3. Preprocessing and augmentation data

Stage	Technique	Description
Preprocessing	Resizing	Stretch to 640 × 640
	Auto-Orient	Applied
Augmentation	Flip	Horizontal, Vertical
	Crop	0% Minimum Zoom, 12% Maximum Zoom
	Blur	Up to 1.9px
	Noise	Up to 1.96% of pixels

The dataset has a total of 3264 images for the mobile phone dataset and 3707 images for the mobile phone usage dataset. Using the following conditions described in Table 3 regarding preprocessing and data augmentation.

The augmentation values were selected based on prior studies and experimental validation. For example, the 12% zoom level simulates camera drift, while up to 1.9px blur reflects motion-related distortions. Noise was injected to replicate low-quality sensor conditions in real vehicles. These techniques were carefully calibrated to preserve key visual features while introducing realistic variability to enhance the detection model's robustness.

Overall, image processing is divided into three subsets: training, validation, and testing. Most of the data was allocated for model training, while the rest was used for validation and testing to evaluate the performance of the algorithm. The total number of images for each class was adjusted to keep the dataset balanced.

2.5 Hardware implementation

The system is deployed on a laptop with octa-core processor specifications, 8GB RAM, and a 1080p USB webcam. This laptop was chosen for its balance of computing power and portability, making it suitable for in-vehicle applications. The lightweight configuration of the YOLOv8 model allows for smooth performance, achieving 10-15 frames per second with real-time object detection accuracy.

In future implementations, the system can be optimized for deployment on lightweight edge devices such as the NVIDIA Jetson Nano, Raspberry Pi 4, or other ARM-based platforms. Given YOLOv8's flexibility and efficient model architecture, the detection pipeline can be quantized and pruned to reduce computational load while maintaining real-time inference. This enables the system to be integrated into vehicles with constrained processing capabilities, further supporting large-scale, cost-effective deployment for real-world driver monitoring applications.

2.6 Algorithm benchmarking

To validate the effectiveness of YOLOv8 in this study, a comparative benchmark was conducted using two previous versions: YOLOv5 and YOLOv7. All models were trained and evaluated on the same dataset configuration, with consistent input size, training epochs, and batch sizes. The comparison focused on metrics including accuracy, mAP@0.5, precision, and recall, which are critical for object detection tasks involving overlapping and small-sized targets such as mobile phones.

Table 4. Comparing different algorithm

Model	mAP @0.5	Accuracy	Recall	Notes
YOLOv5	82.4%	86.0%	80.5%	Moderate Performance
YOLOv7	84.1%	80.5%	83.2%	Improved over YOLOv5
YOLOv8	89.5%	83.2%	85.6%	Best Performance (Proposed)

Table 4 summarizes the results of the benchmark comparison between YOLOv5, YOLOv7, and YOLOv8.

Based on the results in Table 4, YOLOv8 significantly outperformed both YOLOv5 and YOLOv7 in all key metrics. Its superior mean average precision (mAP@0.5) of 89.5% indicates a more reliable object detection capability, particularly under complex conditions like occlusion and overlapping objects. The improved performance is largely attributed to YOLOv8's advanced backbone and neck modules (such as ELAN and PAN-FPN), which enhance spatial feature extraction and multi-scale detection. These advantages justify the selection of YOLOv8 as the core detection engine for the proposed system.

3. RESULTS AND DISCUSSION

3.1 Result

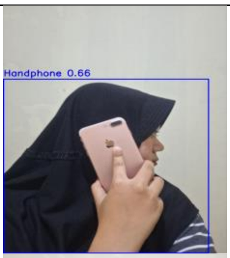
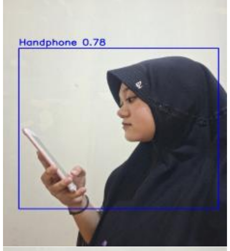
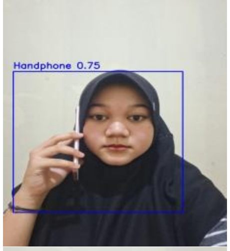
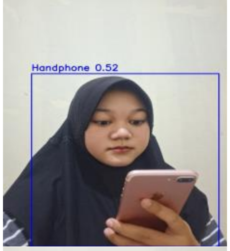
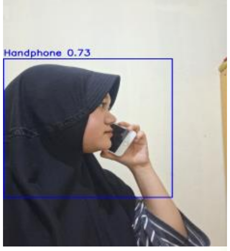
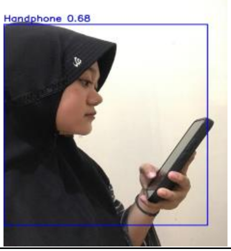
In this experimental setup, several considerations have been applied to enhance detection accuracy [30]. The detection system is designed to recognize driver mobile phone use by analyzing object interaction in the hand and face area, as shown in Table 5. The detection scenarios varied in lighting conditions, phone color, and driver positioning.

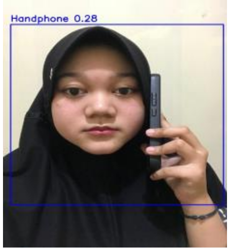
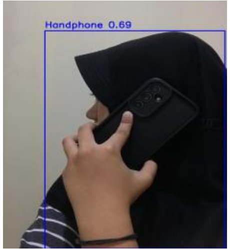
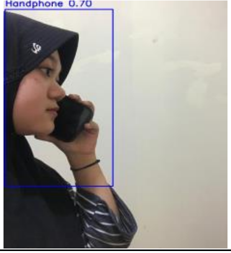
The resulting average is detected, which indicates good performance under conditions such as low light and face clarity. However, a black phone or dim lighting significantly reduces the detection confidence level, suggesting that improvements are needed for challenging conditions, such as incorporating infrared cameras or training with synthetic low-light data.

Nevertheless, in some cases, recognition was at a medium confidence level. Factors such as viewing angle, suboptimal lighting, or complex backgrounds may affect these results. In addition, there was one condition with a very low confidence level, most likely due to the object not being fully visible, obstructed, or in poor lighting conditions.

Overall, these results show that the detection system needs to be improved to achieve more stable performance over a wide range of conditions. Some steps that can be taken include improving the quality of the dataset with more image variations, applying data augmentation techniques to increase the model's robustness to changing conditions, and optimizing the model architecture to make it more robust in detecting objects in different situations. With these improvements, the system is expected to achieve a more consistent and accurate level of confidence in detecting objects in different scenarios.

Table 5. Detection results

Results	**Conditions**	**Status**
	Calling with the left side position and the phone in the right ear	Detected
	Holding the phone in a right-sided position	Detected
	Calling from a front-facing position	Detected
	Holding the phone facing forward	Detected
	Calling with the left side position and the phone in the left ear	Detected
	Holding a mobile phone in a left sideways position	Detected

	Calling with a front-facing position	Detected
	Calling with the phone facing right	Detected
	Calling with a left-facing position	Detected

A prototype phone usage detection system will be developed using a camera and the YOLOv8 algorithm to identify the position of the phone in the driver's face or hand area. The camera will be positioned to cover areas that are frequently in contact with the phone, such as the face and hands. The system is optimized to ensure detection accuracy despite obstructions or less-than-ideal lighting conditions.

Program the results of the created data sets with the corresponding data sets in Table 1 and Table 2, which are then combined in the program. Both data sets are combined in the program with the corresponding method and logic. The goal is to obtain a precise detection condition according to the detection of mobile phone usage, as shown in Table 4.

The calibration method is performed by using the output data from the YOLOv8 algorithm, which is then analyzed to optimize mobile phone detection. This procedure uses an object-specific detection technique [31] by improving the accuracy of the object detection to adjust the prediction value of the bounding box and the confidence score so that they are as close to the ideal condition as possible [32]. The object detection accuracy coefficient is determined based on the pre-tested training dataset, ensuring that the model can consistently detect the presence of phones with high accuracy [33]. The calibrated data sets in Table 1 and Table 2 were then programmed and integrated into the system to provide accurate and reliable phone detection.

Table 6. Data accuracy of dataset

Metrics	Value
Accuracy	92.5%
mAP@0.5	89.5%
Precision	83.2%
Recall	85.6%

From the detection results, the YOLOv8 algorithm achieved a detection accuracy of 92.5% on a combined dataset consisting of a phone holding dataset and a phone dataset. mAP@0.5 recorded 89.5%. The detection results are still good and indicate that the model has consistent performance in different lighting conditions and shooting angles., as shown in Table 6.

The high precision ensures that most detected phones are true positives, while the strong recall rate confirms the system captures most real instances of phone use. The mAP@0.5 value reflects a balance between precision and recall, which is essential in real-time detection systems that must minimize both false alarms and misses.

The results of the value of each metric shown in Table 5 show that the accuracy provided is very high with a value of 92.5%, which indicates that the detection rate is high. Thus, the detection can be appropriate and accurate under certain conditions.

The results of the object detection visualization that has been done in Table 4 show that YOLOv8 is able to detect mobile phones with accurate bounding boxes and recognize user interactions when holding a mobile phone based on overlapping with classes in the mobile phone holding dataset. Thus, the detection results can be more specific by combining two datasets at once with the overlap method.

The system uses detection logic based on the overlap between the bounding box of the mobile phone and the user's hand, which is analyzed using the combined data from two datasets. The combination of the phone holder dataset and the phone dataset enables the system to detect complex scenarios where the phone is in an overlapping position with objects such as hands. This is the basis for real-time phone usage detection. Here is the overlap logic program in Table 7.

Table 7. Overlap programs

```
Start Timer: time_start=current time

Detect objects in the frame using model_detect
Store detection results in results_detect

Detect phone objects in the frame using model_phone
Store detection results in results_phone
```

```
For each detected object in results_detect:
For each bounding box in detected object:
If confidence score of bounding box>0.60:
Get detected class label
Get confidence score

If detected class is "not_calling" or "calling":
Set label as "Handphone" with confidence score
Get bounding box coordinates (x1_detect, y1_detect, x2_detect, y2_detect)
Set handphone_found to False

For each detected phone in results_phone:
For each bounding box in detected phone:
If confidence score of bounding box>0.45:
Get bounding box coordinates (x1_phone, y1_phone, x2_phone, y2_phone)

If phone bounding box is fully inside detected object bounding box:
Set handphone_found to True
Break loop
```

The process starts by calculating the execution start time using time.time(). Next, two prediction models are run on the same frame: model_detect to detect the driver's pose, and model_phone to detect the presence of a phone. The system then iterates through the detection results from model_detect. For each bounding box detected with a confidence level greater than 0.60, the system identifies the object class ('not_calling' or 'calling') and stores the bounding box coordinate information. If the detected class is one of the two categories, the system creates a "Phone" label with confidence value. Next, for each valid driver detection, the system checks the result of model_phone. If a phone bounding box is found with a confidence greater than 0.45 that is completely inside the driver's bounding box (using coordinate comparison), the variable phone is set to True. This indicates that the system has detected the use of a mobile phone by the driver.

Error analysis reveals challenges with false positives and false negatives. Wallets are often misclassified as phones, especially under poor lighting. Some phones were not detected at all in dim environments. To address these, we propose introducing “wallet” as a negative class, applying adversarial training, and extending datasets with low-light or synthetic IR images.

The detection results show that the model performs better on the mobile phone dataset than on the holding phone dataset, with mAPs of 97.5% and 89.5%, respectively. This shows the greater challenge of user interaction detection, which involves multiple bounding boxes. Therefore, both datasets can help each other in detecting mobile phone use. The comparison bar chart of the mAP values of the two datasets is shown in Figure 6.

These results show that YOLOv8 can reliably detect phone use. However, the model needs to be enhanced under certain conditions such as low light or similar objects. This implementation has the potential to be used in a real-time monitoring system for mobile phone use while driving, using appropriate sensing tools.

Compared to previous works using YOLOv5 and YOLOv7 [15, 16, 19], YOLOv8 offers a more modular architecture and improved attention-based detection, particularly for small objects and overlapping classes. While YOLOv5-based systems reported mAP values around 80-85%, our approach achieves 89.5%, indicating a clear performance gain.

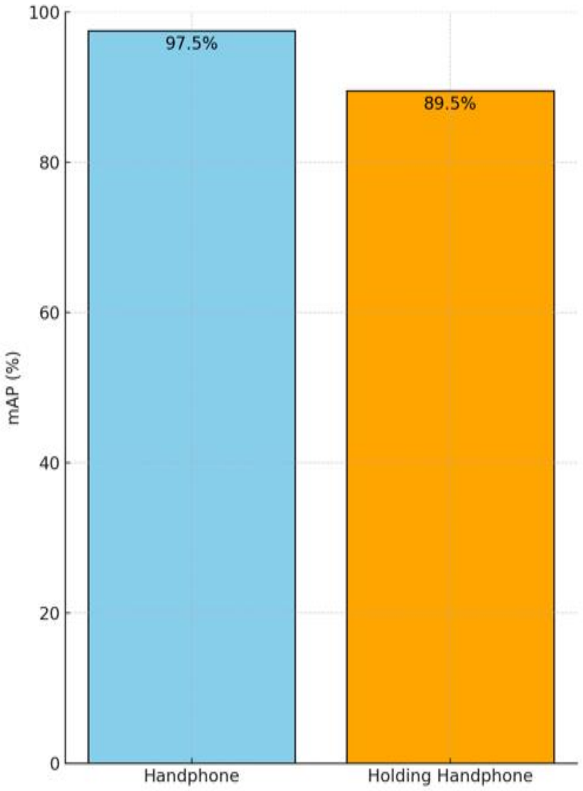


Figure 6. Bar chart comparison of both datasets

3.2 Discussion

This study confirms that combining interaction-based datasets with object datasets improves robust detection in real-world settings. YOLOv8's speed and compact architecture allow deployment on edge devices. However, challenges remain in detecting the occluded phones or handling lighting variations. Real-time application in public transport fleets is feasible, although privacy and hardware limitations must be addressed. Future directions include better datasets, integration with IoT systems, and GAN-enhanced low-light detection.

The method used ensures accurate detection capability under various lighting conditions and shooting angles. Incorporating the phone holding data set and the phone data set enables the system to detect complex scenarios where the phone is in an overlapping position with objects such as hands. This is the basis for real-time phone usage detection.

This research shows that the system can provide accurate data on mobile phone usage based on the position of the hand and device in the frame. To improve performance, the YOLOv8 algorithm was programmed with detection logic based on the overlap between the bounding box of the mobile phone and the user's hand, which was analyzed using combined data from two datasets. The implementation of this system has the potential to be applied to truck drivers, bus drivers, and conventional car drivers in general.

The developed system still has some weaknesses, such as difficulty detecting phones in low light conditions or when the device is partially covered by the hand. Further observations and development of additional datasets are needed to improve accuracy under these conditions. In addition, integration with IoT technology could be the next step to enable real-time remote monitoring of mobile phone usage detection results.

4. CONCLUSION

This research successfully developed a mobile phone usage detection system using the YOLOv8 algorithm, achieving a detection accuracy of 92.5% and mAP@0.5 of 89.5%. By combining two datasets—the mobile phone usage dataset and the mobile phone objects dataset—the system demonstrated improved reliability in detecting complex interactions between users and devices under diverse conditions. The YOLOv8 algorithm, a state-of-the-art deep learning approach, enabled real-time detection with high accuracy, focusing on identifying mobile phones and user interactions in the driver's face or hand area. The system was tested on lightweight edge devices, demonstrating potential for cost-effective and scalable deployment in real-world vehicle environments. Despite its promising performance, challenges such as a 42% false negative rate in low-light conditions and a 53% false positive rate for phone-like objects like wallets remain. Future work should focus on expanding the dataset with more diverse negative classes, applying adversarial training, and exploring infrared-based imaging to enhance robustness. Overall, this research contributes to improving road safety by providing an accurate, automated detection system that can be integrated into broader traffic safety and driver monitoring technologies.

ACKNOWLEDGMENT

We would like to thank the Embedded and Network System (ENS), Telkom University.

REFERENCES

[1] Englund, C., Aksoy, E.E., Alonso-Fernandez, F., Cooney, M.D., Pashami, S., Åstrand, B. (2021). AI perspectives in smart cities and communities to enable road vehicle automation and smart traffic control. Smart Cities, 4(2): 783-802. https://doi.org/10.3390/smartcities4020040

[2] Ozkan, M.F., Ma, Y. (2021). Modeling driver behavior in car-following interactions with automated and human-driven vehicles and energy efficiency evaluation. IEEE Access, 9: 64696-64707. https://doi.org/10.1109/ACCESS.2021.3075194

[3] Hossain, M.Y., George, F.P. (2018). IOT based real-time drowsy driving detection system for the prevention of road accidents. In 2018 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Bangkok, Thailand, pp. 190-195. https://doi.org/10.1109/ICIIBMS.2018.8550026

[4] Montella, A., Punzo, V., Chiaradonna, S., Mauriello, F., Montanino, M. (2015). Point-to-point speed enforcement systems: Speed limits design criteria and analysis of drivers' compliance. Transportation Research Part C: Emerging Technologies, 53: 1-18. https://doi.org/10.1016/j.trc.2015.01.025

[5] Jackisch, J., Sethi, D., Mitis, F., Szymañski, T., Arra, I. (2016). 76 European facts and the global status report on road safety 2015. Injury Prevention, 22(Suppl 2): A29. https://doi.org/10.1136/injuryprev-2016-042156.76

[6] Shajari, A., Asadi, H., Glaser, S., Arogbonlo, A., Mohamed, S., Kooijman, L., Alqumsan, A.A., Nahavandi, S. (2023). Detection of driving distractions and their impacts. Journal of Advanced Transportation, 2023(1): 2118553. https://doi.org/10.1155/2023/2118553

[7] Gjoreski, M., Gams, M.Ž., Luštrek, M., Genc, P., Garbas, J.U., Hassan, T. (2020). Machine learning and end-to-end deep learning for monitoring driver distractions from physiological and visual signals. IEEE Access, 8: 70590-70603. https://doi.org/10.1109/ACCESS.2020.2986810

[8] Ma, B., Fu, Z., Rakheja, S., Zhao, D., He, W., Ming, W., Zhang, Z. (2024). Distracted driving behavior and driver's emotion detection based on improved YOLOv8 with attention mechanism. IEEE Access, 12: 37983-37994. https://doi.org/10.1109/ACCESS.2024.3374726

[9] Alhawsawi, A.N., Khan, S.D., Rehman, F.U. (2024). Enhanced YOLOv8-based model with context enrichment module for crowd counting in complex drone imagery. Remote Sensing, 16(22): 4175. https://doi.org/10.3390/rs16224175

[10] Papatheocharous, E., Kaiser, C., Moser, J., Stocker, A. (2023). Monitoring distracted driving behaviours with smartphones: An extended systematic literature review. Sensors, 23(17): 7505. https://doi.org/10.3390/s23177505

[11] Kaiser, C., Stocker, A., Papatheocharous, E. (2021). Distracted driver monitoring with smartphones: A preliminary literature review. In 2021 29th Conference of Open Innovations Association (FRUCT), Tampere, Finland, pp. 169-176. https://doi.org/10.23919/FRUCT52173.2021.9435545

[12] Chen, L.W., Chen, H.M. (2020). Driver behavior monitoring and warning with dangerous driving detection based on the Internet of Vehicles. IEEE Transactions on Intelligent Transportation Systems, 22(11): 7232-7241. https://doi.org/10.1109/TITS.2020.3004655

[13] Liu, L., Wang, Z., Qiu, S. (2020). Driving behavior tracking and recognition based on multisensors data fusion. IEEE Sensors Journal, 20(18): 10811-10823. https://doi.org/10.1109/JSEN.2020.2995401

[14] Gupta, B.B., Gaurav, A., Chui, K.T., Arya, V. (2024). Deep learning model for driver behavior detection in cyber-physical system-based intelligent transport systems. IEEE Access, 12: 62268-62278. https://doi.org/10.1109/ACCESS.2024.3393909

[15] Peng, D., Ding, W., Zhen, T. (2024). A novel low light object detection method based on the YOLOv5 fusion feature enhancement. Scientific Reports, 14(1): 4486. https://doi.org/10.1038/s41598-024-54428-8

[16] Al-refai, G., Elmoaqet, H., Ryalat, M., Al-refai, M. (2023). Object detection in low-Light environment using YOLOv7. In Research Square. https://doi.org/10.21203/rs.3.rs-3365905/v1

[17] Yu, J., Choi, H. (2021). YOLO MDE: Object detection with monocular depth estimation. Electronics, 11(1): 76. https://doi.org/10.3390/electronics11010076

[18] Cai, Y.X., Li, H.J., Yuan, G., Niu, W., Li, Y.Y., Tang, X.L., Ren, B., Wang, Y.Z. (2020). YOLObile: Real-time object detection on mobile devices via compression-compilation co-design. arXiv preprint arXiv:2009.05697. https://doi.org/10.48550/arXiv.2009.05697

[19] Ejati, R.H.P., Mardhiyyah, R., Zulkhairi, Z., Istiqomah, N., Prasetya, R.I.B. (2023). Real-time smartphone usage surveillance system based on YOLOv5. IJID

(International Journal on Informatics for Development), 11(2): 242-251. https://doi.org/10.14421/ijid.2022.3766

[20] Pedoeem, J., Huang, R. (2018). YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers. arXiv preprint arXiv:1811.05588. https://doi.org/10.48550/arXiv.1811.05588

[21] Zuraimi, M.A.B., Zaman, F.H.K. (2021). Vehicle detection and tracking using YOLO and DeepSORT. In 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, pp. 23-29. https://doi.org/10.1109/ISCAIE51753.2021.9431784

[22] Rahman, Z., Ami, A.M., Ullah, M.A. (2020). A real-time wrong-way vehicle detection based on YOLO and centroid tracking. In 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, pp. 916-920. https://doi.org/10.1109/TENSYMP50017.2020.9230463

[23] Nguyen, H.C., Nguyen, T.H., Scherer, R., Le, V.H. (2023). YOLO series for human hand action detection and classification from egocentric videos. Sensors, 23(6): 3255. https://doi.org/10.3390/s23063255

[24] Glučina, M., Anđelić, N., Lorencin, I., Car, Z. (2023). Detection and classification of printed circuit boards using YOLO algorithm. Electronics, 12(3): 667. https://doi.org/10.3390/electronics12030667

[25] Hasan, M.A. (2023). Facial human emotion recognition by using YOLO faces detection algorithm. JOINCS (Journal of Informatics, Network, and Computer Science), 6(2): 32-38. https://doi.org/10.21070/joincs.v6i2.1629

[26] Hussain, M. (2024). Yolov5, YOLOv8 and YOLOv10: The go-to detectors for real-time vision. arXiv preprint arXiv:2407.02988. https://doi.org/10.48550/arXiv.2407.02988

[27] Sundaresan Geetha, A., Alif, M.A.R., Hussain, M., Allen, P. (2024). Comparative analysis of YOLOv8 and YOLOv10 in vehicle detection: Performance metrics and model efficacy. Vehicles, 6(3): 1364-1382. https://doi.org/10.3390/vehicles6030065

[28] Ibnugraha, P.D., Sani, M.I., Sari, M.I., Rizal, M.F., Hanifa, F.H., Kurniawan, A.P. (2023). Automatic Passenger Counting (APC) for online Event Data Recorder (EDR). In 2023 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD), Denpasar, Indonesia, pp. 89-93. https://doi.org/10.1109/ICoABCD59879.2023.10390960

[29] Debsi, A., Ling, G., Al-Mahbashi, M., Al-Soswa, M., Abdullah, A. (2024). Driver distraction and fatigue detection in images using ME-YOLOv8 algorithm. IET Intelligent Transport Systems, 18(10): 1910-1930. https://doi.org/10.1049/itr2.12560

[30] Jamtsho, Y., Riyamongkol, P., Waranusast, R. (2021). Real-time license plate detection for non-helmeted motorcyclist using YOLO. ICT Express, 7(1): 104-109. https://doi.org/10.1016/j.icte.2020.07.008

[31] Agrawal, P., Jain, G., Shukla, S., Gupta, S., Kothari, D., Jain, R., Malviya, N. (2022). Yolo algorithm implementation for real time object detection and tracking. In 2022 IEEE Students Conference on Engineering and Systems (SCES), Prayagraj, India, pp. 1-6. https://doi.org/10.1109/SCES55490.2022.9887678

[32] He, Y., Peng, Y., Wei, C., Zheng, Y., Yang, C., Zou, T. (2024). Automatic disease detection from strawberry leaf based on improved YOLOv8. Plants, 13(18): 2556. https://doi.org/10.3390/plants13182556

[33] Fang, W., Wang, L., Ren, P.M. (2019). Tinier-YOLO: A real-time object detection method for constrained environments. IEEE Access, 8: 1935-1944. https://doi.org/10.1109/ACCESS.2019.2961959