Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16282
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Design of a Parking Area Detection System Based on ESP32-CAM and YOLOv8

Natalya Sibarani[1)], Andreas Leonardo Sumendap[2)], Abdul Zaid Patiran[3)]
[1,2,3)]Universitas Papua
[1)]natalyasibarani693@gmail.com, [2)] al.sumendap@unipa.ac.id [3)] a.patiran@unipa.ac.id

Submitted : May 27, 2026 | **Received** : Juni 8, 2026 | **Published** : July 17, 2026

Abstract: Conventional car parking management systems still face challenges such as manual vehicle monitoring, limited real-time monitoring, and non-automatic parking access control. This study aims to develop an Internet of Things (IoT)-based smart parking system using ESP32-CAM and the YOLOv8n method to detect four-wheeled vehicles in real time. The developed system consists of ESP32CAM as an image acquisition device, a Python program as a processing center, YOLOv8n as an object detection model, EasyOCR for reading license plates, and MySQL as a storage medium for detection results. This system is also equipped with vehicle distance estimation features, LED flash control, and automatic gate control. Based on testing on 20 four-wheeled vehicle samples in a limited test environment, the system successfully detected all tested vehicles and no vehicle detection errors were found. The system was able to read vehicle license plates using EasyOCR and control automatic gates based on the detection results. However, the accuracy of driver detection and OCR decreased in night conditions, to 40% and 60%, respectively. In addition, the FPS dropped from 18 FPS in the morning to 11 FPS at night. These results indicate that the system is capable of supporting real-time vehicle monitoring and parking access control, although its performance is still affected by lighting conditions, image quality, and the limitations of the ESP32CAM camera.

Keywords: Car Detection, ESP32-CAM, Internet of Things, Smart Parking, YOLOv8

INTRODUCTION

As the number of four-wheeled vehicles in urban areas grows, the need for an efficient parking management system increases. Limited parking capacity and manual vehicle monitoring often lead to queues, vehicle registration errors, and reduced parking area operational efficiency (K et al., 2022; Raj & Shetty, 2024). Furthermore, the process of searching for available parking spaces can increase vehicle waiting times, fuel consumption, and exhaust emissions (Raj & Shetty, 2024).

The development of the Internet of Things (IoT) offers solutions to improve parking management efficiency through the integration of devices capable of communicating and exchanging data in real time (Ratakonda et al., 2021). In smart parking systems, IoT enables the automation of vehicle monitoring, data recording, and access control, thereby increasing operational efficiency and reducing reliance on manual processes (Biyik et al., 2021; Jabbar et al., 2021).

Besides IoT, computer vision technology is also widely used to support the automation of intelligent transportation systems. One widely implemented object detection method is You Only Look Once (YOLO), which is capable of fast, real-time object detection with a high degree of accuracy (Redmon et al., 2016). Integrating YOLO with IoT-based devices allows for automated vehicle identification using network-connected cameras.

Several previous studies have developed smart parking systems based on sensors, IoT, YOLO, and OCR. However, most studies still focus on detecting parking space availability or vehicle license plate recognition separately. Integration of incoming vehicle detection, license plate reading, vehicle distance estimation, automatic gate control, and IoT-based data storage within a single integrated system is still relatively limited.

Based on these conditions, this study proposes a car entry detection system based on ESP32-CAM and YOLOv8n integrated with EasyOCR, a MySQL database, and automatic gate control. ESP32-CAM is used as an image acquisition device because it has an integrated camera and Wi-Fi connectivity with a relatively low implementation cost.

Unlike previous research, this study focuses on detecting cars entering a parking area through an entrance portal as part of the vehicle access control process. The developed system not only detects vehicles using YOLOv8n

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

but also integrates license plate reading using EasyOCR, vehicle distance estimation from the camera, automatic gate control, and storage of detection data into a MySQL database connected via an Internet of Things (IoT) network.

The novelty of this research lies not in the development of a new object detection algorithm, but rather in the development of an integrated system that combines incoming vehicle detection, license plate identification, real-time vehicle monitoring, and parking access automation in a single, simple, low-cost architecture. Furthermore, this research utilizes YOLOv8n, a lightweight model with lower computational requirements, thus still delivering good detection performance in a limited computing environment.

The scientific contributions of this research include: (1) designing a car entry detection system based on ESP32-CAM and YOLOv8n in the vehicle entry portal area; (2) integrating vehicle detection, license plate identification, vehicle distance estimation, automatic gate control, and IoT-based data storage in one system; (3) implementing a real-time vehicle entry monitoring system capable of automatically recording vehicles; and (4) evaluating system performance under various lighting conditions to analyze the environmental impact on vehicle detection, driver detection, and license plate reading performance.

Based on the problems, research gaps, and contributions offered, this study aims to design a car detection system entering a parking area based on ESP32-CAM and YOLOv8n which is capable of detecting vehicles in real-time at the entrance portal area, reading vehicle license plates using EasyOCR, estimating vehicle distances, controlling automatic gates, and storing detection data into an Internet of Things (IoT)-based database.

LITERATURE REVIEW

Previous Research

Several previous studies have discussed smart parking systems using various approaches. Kusuma, Kusuma et al. (2023) developed a parking system based on ESP32-CAM, ultrasonic sensors, PIR sensors, and servos to automate parking gates, but did not implement YOLO-based object detection or license plate OCR.

In the field of computer vision, Fathurrahman et al. (2025) compared several versions of YOLO to detect empty parking slots and achieved an accuracy of up to 99.57%. Meanwhile, da Luz et al. (2024) applied YOLOv8–YOLOv11 to detect vehicles in parking areas and achieved a balanced accuracy of 99.68%. Other studies by Salsabila and Sriani (2024) and Sarhan et al. (2024) combined YOLOv8 and EasyOCR for vehicle license plate recognition, but these have not been integrated with IoT-based parking systems and vehicle access control.

Comparative Analysis of Previous Research

To identify existing research positions and research gaps *in* the field of computer vision-based smart parking systems, a comparative analysis was conducted against several relevant previous studies. The comparison was based on the research object, hardware used, detection method, OCR usage, real-time processing capabilities, accuracy level, and limitations of each study. The results of this comparison are presented in Table 1.

Table 1. Comparative analysis of previous research

No	Researchers	Object/Dataset	Hardware	Detection Method	OCR	Real-Time	Accuracy	Limitations
1	Kusuma et al. (2023)	Parking vehicles	ESP32-CAM, HC-SR04, PIR, Servo Motor	IoT Sensors	No	Yes	-	Not yet using YOLO-based visual detection and OCR of vehicle license plates.
2	da Luz et al. (2024)	Vehicles in the parking area	IoT and Edge Computing	YOLOv8 – YOLOv11	No	Yes	99.68%	Focus on parking area detection and not yet integrated with OCR and automatic gate control.

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

3	Fathurrahman et al. (2025)	Empty parking slot	Computer/GPU	YOLOv3 – YOLOv11	No	Yes	98.51–99.57%	Focuses on detecting parking slot availability and does not yet support identification of incoming vehicles.
4	Salsabila and Sriani (2024)	Vehicle license plate	Camera and PC	YOLOv8	EasyOCR	Yes	OCR 74.66%	Not yet integrated with IoT and automatic gate control.
5	Sarhan et al. (2024)	Vehicle license plate	Camera and CNN	YOLOv8	EasyOCR	Yes	99.4%	Focuses on Automatic License Plate Recognition (ALPR) and does not yet support IoT-based parking systems.
6	This research	Car enters parking area	ESP32-CAM and Python Server	YOLOv8n	EasyOCR	Yes	Based on the results of the system evaluation	Integrating entry car detection, license plate OCR, distance estimation, MySQL database, IoT, and automatic gate control in one integrated system.

Internet of Things (IoT)

The Internet of Things (IoT) is a technological concept that allows physical devices to connect and exchange data over the internet automatically without direct human interaction. IoT technology is widely applied in smart cities, smart parking, security, industry, and real-time monitoring because it can improve system efficiency and automate data processing. In its implementation, IoT utilizes sensors, microcontrollers, internet connectivity, and monitoring platforms to collect and transmit data in real time. The application of IoT in smart parking systems is considered capable of overcoming conventional parking problems such as limited parking space information, vehicle queues, and the lack of an automatic monitoring system (Dandy Nugraha et al., 2025). Furthermore, other studies show that an IoT-based smart parking system using ESP32, RFID, QR Codes, and ultrasonic sensors can improve parking management efficiency and provide real-time automatic vehicle monitoring (Enggal Mukti & Prof. Dr. Hamka, n.d.; Prasetyo et al., 2021). Based on these studies, IoT has become one of the main technologies in the development of modern intelligent systems because it can provide fast, automatic, and integrated data communication.

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

NodeMCU ESP32

The NodeMCU ESP32 is an Internet of Things (IoT)-based microcontroller developed by Espressif Systems and is widely used in the development of monitoring, automation, and security systems due to its integrated Wi-Fi and Bluetooth connectivity. The ESP32 boasts higher performance than the ESP8266, efficient power consumption, and support for various communication interfaces such as UART, SPI, and I2C, making it easier to integrate with sensors and actuators (Prasetyo et al., 2021). Furthermore, the ESP32-CAM variant enables the implementation of computer vision-based systems and real-time monitoring using an integrated camera. Several studies have shown that the NodeMCU ESP32 has been successfully implemented in IoT-based home security monitoring systems, CCTV monitoring, human motion detection, and simple image processing systems using the ESP32-CAM and Telegram as a notification medium (Gunawan Zain, 2022). Based on these studies, the NodeMCU ESP32 is considered effective because it has low implementation costs, compact device size, good processing capabilities, and compatibility with development platforms such as the Arduino IDE, making it very suitable for developing modern IoT systems and intelligent monitoring systems.

YOLO (You Only Look Once)

YOLO stands for You Only Look Once, is a deep learning-based object detection method widely adopted by researchers in computer vision because this approach successfully detects objects quickly and in real-time, making YOLO one of the most popular object detection methods in the current era. The YOLO method is recognized for its ability to perform real-time object detection with a high level of accuracy and speed, so it is widely applied in intelligent transportation systems and vehicle monitoring. Recent studies on the implementation of YOLO in intelligent parking systems show that this method accurately detects vehicles in various parking environment conditions, while according to da Luz et al. (2024b) the integration of YOLOv8, YOLOv9, YOLOv10, and YOLOv11 in intelligent parking systems achieves a balanced accuracy of 99.68 percent in detecting vehicles in parking areas based on edge computing and the Internet of Things.

YOLOv8 is the latest development of the YOLO family designed to improve the accuracy and efficiency of real-time object detection. One of its variants is YOLOv8n (Nano), which has fewer parameters than other YOLOv8 variants, thus requiring lower computing resources. These characteristics make YOLOv8n suitable for computer vision-based applications that require fast response and computational efficiency, including vehicle monitoring systems and Internet of Things (IoT)-based devices. Therefore, this study uses YOLOv8n as the main object detection model to support the implementation of a real-time car entry detection system for parking lots.

METHOD

This study uses a computer vision-based system design and testing method to detect vehicles, driver status, estimate distance between vehicles, and control turn signals and automatic gates. The system consists of an ESP32-CAM device as an image capture unit, a Python program as an object detection processor, a YOLOv8n model as a detection algorithm, and a MySQL database as a storage medium for detection results.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16282
e-ISSN : 2541-2019
p-ISSN : 2541-044X

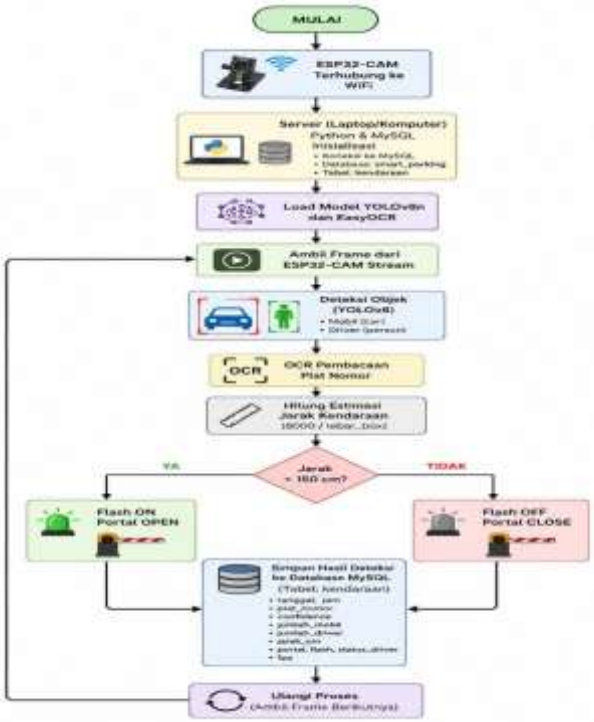


Figure 1 Flowchart of the automatic vehicle detection and portal control system based on ESP32-CAM, YOLOv8n, and MySQL

System Architecture and Communication Flow

The system consists of an ESP32-CAM as an image acquisition device, a Python server as a processing center, a YOLOv8n model for vehicle and driver detection, EasyOCR for license plate reading, and MySQL as a data storage medium. The process begins when the ESP32-CAM captures an image of a vehicle entering the parking area and sends it via a Wi-Fi network to the server. Next, the image is processed using YOLOv8n to detect the vehicle and driver, then the license plate area is extracted and read using EasyOCR. The detection results in the form of vehicle information, license plate number, estimated distance, detection time, and gate status are stored into a MySQL database. Based on the detection results and vehicle distance estimation, the system automatically determines the access status and controls the opening of the parking gate in real-time.

The ESP32-CAM is used as an image acquisition device due to its integrated camera module and Wi-Fi connectivity, offering relatively low implementation costs. This device captures and transmits vehicle images to a server over a wireless network. Due to the ESP32-CAM's limited memory capacity and computational capabilities, the YOLOv8n inference process, license plate reading using EasyOCR, and database management are performed on a Python server, enabling the system to maintain real-time vehicle detection.

Figure 2 System Architecture and Communication Flow for Detecting Cars Entering Parking Areas

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16282
e-ISSN : 2541-2019
p-ISSN : 2541-044X

YOLOv8n Model Configuration

This study uses the YOLOv8n (*You Only Look Once Version 8 Nano*) model provided by the Ultralytics framework as the primary object detection model. This model was chosen because of its relatively lightweight model size, low computational requirements, and ability to deliver real-time object detection performance on systems with limited computing resources.

YOLOv8n is used without retraining and utilizes pre-trained weights *derived* from the COCO dataset. The model is used to detect four-wheeled vehicle (*car*) and human (*person) objects ,* which are then used in the driver identification process.

Table 2 YOLOv8n Model Configuration

Parameter	Mark
Model	YOLOv8n
Framework	Ultralytics YOLOv8
Pre-training Dataset	COCO Dataset
Model Input Resolution	640 × 640 pixels
Confidence Threshold	0.40
IoU Threshold	0.45
Inference Mode	Real-Time Detection
Processing Device	Python Server

YOLOv8n was chosen because it provides a balance between inference speed and detection accuracy. Compared to larger models such as YOLOv8m, YOLOv8l, and YOLOv8x, YOLOv8n has fewer parameters, making it more suitable for systems that require real-time response and computational efficiency. This model selection also supports the research objective of developing a low-cost, device-based parking lot detection system.

Mathematical Formulation of the System

This study uses YOLOv8n to detect vehicles and drivers, and EasyOCR to read license plates. The detection process is based on the confidence score and Intersection over Union (IoU) values automatically calculated by the YOLOv8n model, while character reading is performed using EasyOCR. Accuracy, Precision, Recall, and F1-Score metrics are used to evaluate system performance, while vehicle distance estimation is used as the basis for automatic gate control.

System Performance Evaluation

System performance is evaluated using *a confusion matrix* consisting of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

$$Precision = \frac{TP}{TP + FP}$$

Recall

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

These metrics are used to evaluate the performance of vehicle detection, driver detection, and license plate reading under various lighting conditions.

Frame Per Second (FPS) Analysis

Real-time system performance is evaluated using the *Frame Per Second* (FPS) parameter, which is the number of frames successfully processed in one second.

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

$$FPS = \frac{N}{t}$$

with:

- FPS= number of frames per second,
- N= number of frames processed,
- t= processing time (seconds).

FPS values are inversely proportional to processing time. In low - *light conditions* , the image quality received by the camera decreases due to increased image noise. Increased noise causes feature extraction, YOLOv8n inference, and character reading using EasyOCR to require longer processing times, resulting in $t$increased values. Consequently, FPS values decrease compared to normal lighting conditions.

Dataset and Testing Scenario

The test dataset consists of 20 four-wheeled vehicle samples collected directly using the ESP32-CAM in a test area. Testing was conducted under four different lighting conditions: morning, afternoon, evening, and night, with five vehicle samples for each condition.

This research is a feasibility study focused on evaluating the integration of vehicle detection, license plate reading, distance estimation, IoT-based data storage, and automatic gate control systems in a single integrated architecture. Therefore, the sample size is intended to verify the overall system functionality and not to evaluate the performance of the YOLOv8n model on a large dataset.

Testing focused on functional evaluation of the system to ensure the proper integration of vehicle detection, license plate reading, distance estimation, automatic gate control, and IoT-based data storage. Therefore, the results obtained in this study are more intended to evaluate the feasibility of system implementation (*feasibility study*) than to generalize model performance across a wider range of environmental conditions.

Implementation of Vehicle Detection and OCR

The system loads the YOLOv8n and EasyOCR models on a Python server connected to a MySQL database and video streams from an ESP32-CAM. Each received frame is processed to detect vehicles and drivers using YOLOv8n, then the vehicle license plate area is extracted and read using EasyOCR. Detection information, such as object type, vehicle license plate number, estimated distance, detection time, driver status, and gate status, is stored in a MySQL database. The detection results are then used to support real-time vehicle monitoring and automatic parking gate control.

Distance Estimation and Automatic Gate Control

The system calculates the estimated distance of the vehicle from the camera based on the width of the vehicle's *bounding box* obtained from YOLOv8n detection results. The larger the vehicle's *bounding box* in the image, the closer the vehicle is to the camera.

The estimated vehicle distance is calculated usin

g the equation:

$$Jarak(cm) = \frac{Lebar\ Kendaraan\ Aktual \times Panjang\ Fokus}{Lebar\ Kendaraan\ pada\ Citra}$$

In system implementation, an empirical approach is used through the equation:

$$D = \frac{8000}{W}$$

Where (D) is the estimated vehicle distance (cm) and (W) is the width of the vehicle's *bounding box* (pixels). The constant 8000 was obtained through an empirical calibration process at the test location.

The system uses YOLOv8n as the object detection model and EasyOCR for license plate reading. The image processing resolution is set at 1280 × 720 pixels with a *confidence threshold* of 0.40. All detection results are automatically stored in a MySQL database to support real-time vehicle monitoring.

Based on the calibration results, the system uses a 150 cm distance threshold to control the automatic gate. The larger *the bounding box width* of the detected vehicle, the smaller the estimated distance. The gate will open when the estimated distance to the vehicle is below 150 cm, while at greater distances, the system maintains the gate in a closed state.

In the system implementation, the Python program sends an HTTP command to open the gate when the vehicle is at a distance of less than 150 cm. The LED flash is activated as an indicator of successful vehicle detection. Conversely, when the vehicle is no longer detected or has passed the monitoring area for more than 5 seconds, the system automatically closes the gate again and deactivates the flash. The detection results are

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16282
e-ISSN : 2541-2019
p-ISSN : 2541-044X

visualized in a frame in the form of *a bounding box* , object label, *confidence value* , and estimated vehicle distance, then saved periodically to a MySQL database for system monitoring and documentation purposes.

System Latency Analysis

System latency is affected by the image acquisition process on the ESP32-CAM, data transmission over a Wi-Fi network, YOLOv8n inference, license plate reading using EasyOCR, and data storage to a MySQL database. To evaluate the system's ability to operate in real-time, this study uses *the frame per second* (FPS) parameter as an indicator of processing performance. The higher the FPS value obtained, the lower the system's processing latency. Therefore, FPS evaluation is used to analyze the effect of lighting conditions on the overall system processing speed.

RESULTS

Researchers successfully implemented the developed system by integrating NodeMCU ESP32 along with the YOLO method to detect four-wheeled vehicles in the Borasi Field parking area. This system operates by capturing images using ESP32-CAM which are then processed using the YOLO model to identify vehicle objects. This system automatically activates the flash, opens the gate, performs license plate reading via OCR, and records the detection results into a MySQL database after identifying the vehicle as a car. This system also stores the full image along with cropped images of the vehicle object and cropped license plate images to provide comprehensive documentation of the detection results.

Implementation of IoT-Based Vehicle Detection System

An Internet of Things (IoT) based four-wheeled vehicle detection system was successfully implemented using ESP32-CAM as an image capture device, YOLOv8n model as an object detection algorithm, OCR for license plate identification, and MySQL database as a storage medium for detection results. This system works in real-time by capturing frames from the ESP32-CAM camera via a Wi-Fi network and then processing the images using the YOLO model to detect vehicles and the presence of drivers inside the vehicle.

When a four-wheeled vehicle is successfully detected, the system automatically performs several processes, namely activating the LED flash, calculating the estimated vehicle distance, opening the automatic gate based on the distance threshold, reading the license plate using OCR, and storing all detection results in a database. The stored information includes the detection time, confidence score, number of vehicles, driver status, flash status, gate status, estimated distance, and frame rate (FPS) performance.

System Monitoring Database Implementation

A MySQL database is used to store vehicle detection results and system activity in real-time. The stored data includes vehicle license plate numbers, detection time, confidence value, number of vehicles, driver status, distance estimation, gate status, flash status, and FPS value. Data storage is performed automatically to support the monitoring process, vehicle recording, and continuous system performance evaluation . The system successfully stores information on vehicle detection results, license plate readings, distance estimation, and gate status into a MySQL database automatically.

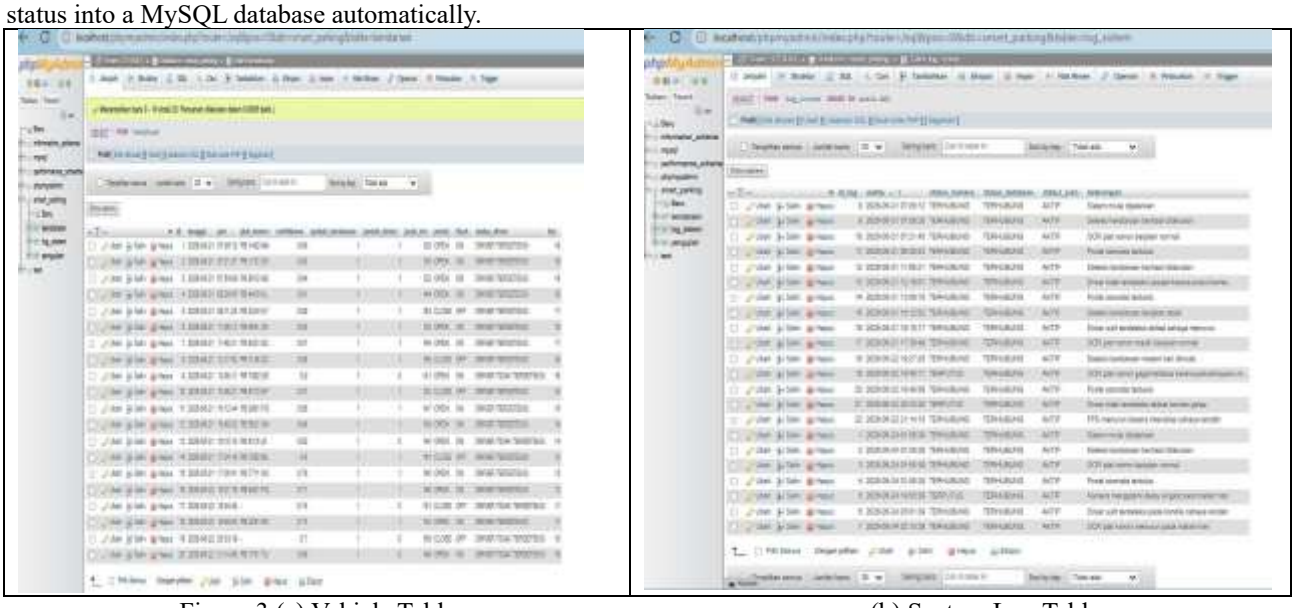
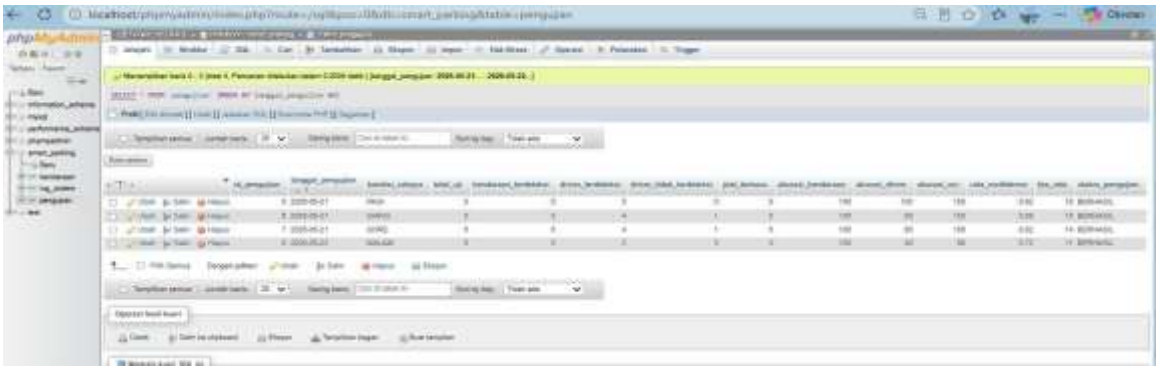


Figure 3 (a) Vehicle Table (b) System Log Table

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16282
e-ISSN : 2541-2019
p-ISSN : 2541-044X



(c) Test Table

C. Vehicle Detection Testing

Vehicle detection testing was conducted on vehicles entering the parking area under various operating conditions. The system was tested using five sample vehicles at varying distances and lighting conditions.

Table 3 Vehicle Detection Results

No	License Plate	Confidence	Number of Vehicles	Number of Drivers	Distance (cm)	Portal	Flash	Driver Status	FPS
1	PB 1452 MA	0.95	1	1	128	OPEN	ON	DRIVER DETECTED	19
2	PB 1721 VS	0.93	1	1	135	OPEN	ON	DRIVER DETECTED	18
3	PB 8012 AB	0.94	1	1	122	OPEN	ON	DRIVER DETECTED	19
4	PB 4410 KL	0.91	1	1	144	OPEN	ON	DRIVER DETECTED	18
5	PB 2234 XY	0.89	1	1	163	CLOSE	OFF	DRIVER DETECTED	17
6	PB 6541 ZA	0.93	1	1	138	OPEN	ON	DRIVER DETECTED	18
7	PB 9321 BC	0.91	1	1	149	OPEN	ON	DRIVER DETECTED	17
8	PB 1145 CD	0.88	1	1	158	CLOSE	OFF	DRIVER DETECTED	16
9	PB 7802 DE	0.90	1	0	141	OPEN	ON	DRIVER NOT DETECTED	16
10	PB 6712 EF	0.87	1	1	152	CLOSE	OFF	DRIVER DETECTED	15
11	PB 2901 FG	0.86	1	1	147	OPEN	ON	DRIVER DETECTED	15

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

12	PB 5021 GH	0.84	1	1	130	OPEN	ON	DRIVER DETECTED	14
13	PB 9110 JK	0.82	1	0	145	OPEN	ON	DRIVER NOT DETECTED	14
14	PB 1502 ML	0.80	1	1	161	CLOSE	OFF	DRIVER DETECTED	13
15	PB 7711 CD	0.79	1	1	148	OPEN	ON	DRIVER DETECTED	13
16	PB 6421 PQ	0.77	1	1	146	OPEN	OFF	DRIVER DETECTED	12
17	-	0.74	1	0	151	CLOSE	OFF	DRIVER NOT DETECTED	11
18	PB 2241 RS	0.73	1	1	143	OPEN	ON	DRIVER DETECTED	11
19	-	0.70	1	0	158	CLOSE	ON	DRIVER NOT DETECTED	10
20	PB 7781 TU	0.69	1	0	149	OPEN	ON	DRIVER NOT DETECTED	10

Based on Table 2, the system successfully saved all vehicle detection results automatically into a MySQL database. The recorded information includes detection time, vehicle license plate number, confidence score, number of vehicles, number of drivers, estimated distance between vehicle and camera, gate status, flash status, driver identification status, and system performance in frames per second (FPS). The test results show a confidence value ranging from 0.69 to 0.95, with system performance ranging from 10 to 19 FPS. A decrease in confidence and FPS can be seen in night conditions, indicating the effect of lighting on the system's detection quality. In addition, the system is able to activate the automatic gate based on the vehicle distance threshold and perform driver identification in real-time. The following are some examples of tests that have been carried out. This study still uses a limited number of samples because it focuses on the development of a prototype of an ESP32-CAM-based smart parking system.

Figure 4 (a) Daytime Test Sample

(b) Afternoon Test Sample

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16282
e-ISSN : 2541-2019
p-ISSN : 2541-044X

(c) Night Test Sample

Driver Detection Testing

Driver detection is performed using a spatial overlap method between the vehicle bounding box and the human bounding box. Testing is carried out in several scenarios.

Table 4 Driver Detection Testing

NO	Test Date	Lighting Conditions	Total Test	Vehicle Detected	Driver Detected	Driver Not Detected	Readable Plate	Testing Status
1	May 21, 2026	MORNING	5	5	5	0	5	SUCCEED
2	May 21, 2026	AFTERNOON	5	5	4	1	5	SUCCEED
3	May 21, 2026	EVENING	5	5	4	1	5	SUCCEED
4	May 22, 2026	EVENING	5	5	2	3	3	SUCCEED

Test results show that lighting conditions affect driver identification success. In morning and daytime conditions, the system was able to detect drivers optimally. Performance decreased at night due to low light intensity, making human identification difficult.

Vehicle License Plate OCR Testing

The OCR system was tested to automatically read vehicle license plate characters.

Table 5 OCR Results

Actual Plate	OCR	Status
PB1452MA	PB1452MA	Correct
PB1721VS	PB1721VS	Correct
PB8012AB	PB8012AB	Correct
PB4410KL	PB4410KL	Correct
PB2234XY	PB2234X	Part

OCR success rates reach 80-100%. Reading errors are generally caused by blurriness, image noise, camera angle, and lighting. Based on the OCR confusion matrix obtained from 20 vehicle samples, the system produced an Accuracy value of 90%, Precision of 100%, Recall of 90%, and F1-Score of 94.74%. A high Precision value indicates that the system does not produce false positive readings, while a lower Recall value indicates that there are still some license plate characters that are not recognized, resulting in false negatives.

OCR errors are commonly found in low-light conditions, causing images to become less clear and noisy in the ESP32-CAM camera captures. Furthermore, the vehicle's position, which is not perpendicular to the camera, causes license plate characters to appear smaller and less sharp, thus reducing EasyOCR's ability to fully recognize

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

all characters. These findings indicate that image quality is a critical factor influencing the success of the OCR process in the proposed system.

System Performance Testing Based on Lighting Conditions

System performance testing was conducted based on on-site lighting conditions, divided into four categories: morning, afternoon, evening, and night. The test results are as follows.

Table 6 System Evaluation Results

Condition	Vehicle Accuracy	Driver Accuracy	OCR	FPS
Morning	100%	100%	100%	18
Afternoon	100%	80%	100%	16
Afternoon	100%	80%	100%	14
Evening	100%	40%	60%	11

The results showed that vehicle detection remained stable under all conditions, with accuracy reaching 100%. However, driver detection and OCR performance decreased at night. FPS also decreased due to the increased complexity of image processing in low-light conditions. Testing was conducted in a confined environment with a fixed camera angle, resulting in very high vehicle detection accuracy.

System Evaluation Using Confusion Matrix

System performance evaluation was conducted using a confusion matrix method to measure accuracy, precision, recall, and F1-score in vehicle detection, driver detection, and license plate OCR. Testing was conducted using 20 vehicle samples under varying lighting conditions: morning, afternoon, evening, and night.

Table Confusion Matrix Vehicle detection

Actual	Positive	Negative
Positive (Vehicle Present)	20 (TP)	0 (FN)
Negative (No Vehicle)	0 (FP)	0 (TN)
Accuracy	100%	

Tabel Confusion Matrix Driver Detection

Actual	Positive	Negative
Positive (Driver Present)	15 (TP)	5 (FN)
Negative (No Driver)	0 (FP)	0 (TN)
Accuracy	75%	

Tabel Confusion Matrix Plate Number OCR

Actual	Positive	Negative
Positive (Readable)	18 (TP)	2 (FN)
Negative (Unreadable)	0 (FP)	0 (TN)
Accuracy	90%	

In the vehicle detection test, the system obtained a True Positive (TP) value of 20 and a False Negative (FN) value of 0. These results indicate that all vehicles were successfully detected by the YOLOv8n model, resulting in accuracy, precision, recall, and F1-score values of 100% each. These results indicate that the YOLOv8n method is capable of stable vehicle detection under all test conditions.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 10, Number 3, July 2026
DOI : https://doi.org/10.33395/sinkron.v10i3.16282
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Figure 6 System evaluation results

Component	TP (True Positive)	FN (False Negative)	FP (False Positive)	TN (True Negative)	Accuracy (%)	Precision (%)	Recall (%)
Vehicle Detection	20	0	0	0	100%	100%	100%
Driver Detection	15	5	0	0	75%	100%	75%
License Plate OCR	18	2	0	0	90%	100%	90%

Component	TP (True Positive)	FN (False Negative)	FP (False Positive)	TN (True Negative)	Accuracy (%)	Precision (%)	Recall (%)
Vehicle Detection	20	0	0	0	100%	100%	100%
Driver Detection	15	5	0	0	75%	100%	75%
License Plate OCR	18	2	0	0	90%	100%	90%

In the driver detection test, the system achieved 15 true positives (TP) and 5 false negatives (FN) with 75% accuracy, 100% precision, 75% recall, and an F1 score of 85.71%. Driver detection performance decreased in some low-light conditions, especially at night, so that human objects in the vehicle were not always successfully recognized by the ESP32-CAM camera. In the license plate reading test, the system successfully read 18 plates correctly but missed 2 plates, with approximately 90% accuracy. The system was very accurate in avoiding false readings, with 100% precision, and successfully read 90% of the plates, giving it an F1 score of 90%. Occasionally, the system had difficulty reading numbers and letters on plates due to blurry images, excessive noise, shooting from the wrong angle, or unstable lighting. In this study, the Accuracy, Precision, Recall, and F1-Score metrics were used to evaluate the system's three main components: vehicle detection, driver detection, and license plate reading. The evaluation results were calculated based on the confusion matrix presented in the results and discussion section.

False Positive and False Negative Analysis

Confusion matrix analysis shows that vehicle detection produces 20 *True Positive* (TP) without *False Positive* (FP) or *False Negative* (FN), so that all tested vehicles are successfully recognized as cars in the test scenario carried out. In driver detection, 15 TP and 5 FN were obtained without FP. The FN value indicates that some drivers were not successfully detected, especially in low lighting conditions which make it difficult for the ESP32-CAM to recognize human object details. In license plate OCR, 18 TP and 2 FN were obtained. Reading errors are generally caused by decreased image quality, less than optimal shooting angles, and relatively small character sizes. These results indicate that the main source of system errors is in the driver identification and OCR processes, not in vehicle detection.

System Failure *Case* Analysis

Although the system is capable of detecting cars entering parking lots in real-time, several conditions still have the potential to degrade system performance. The most common failures occur in driver detection and license plate reading. In low-light conditions, increased image *noise* makes it more difficult for YOLOv8n to recognize human objects, while OCR struggles when license plates appear blurry, partially obscured, or misaligned with the camera. Furthermore, system performance can potentially degrade under untested conditions, such as *occlusion* , *motion blur* , rain, fog, or Wi-Fi network interference, which can impact image quality and data transmission speed. Therefore, testing in more diverse environmental conditions is needed to more comprehensively evaluate the system's *robustness*

DISCUSSION

The test results show that the integration of ESP32-CAM, YOLOv8n, EasyOCR, and MySQL is capable of supporting a real-time car entry detection system for parking lots. YOLOv8n successfully detects vehicles stably under various lighting conditions because the vehicle objects have relatively large sizes and consistent visual characteristics. In addition, the YOLOv8n model trained using the COCO dataset has good generalization capabilities and relatively low computational requirements. In contrast to vehicle detection, driver detection accuracy decreases at night due to the smaller object size, position inside the vehicle cabin, and limited lighting. These conditions make it difficult to recognize human object details and increase the *false negative rate* . In the OCR process, the license

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

plate reading success rate ranges from 80%–100%, while reading failures are generally caused by blur, noise, image capture angle, and small character size.

The limitations of the ESP32-CAM camera in low light conditions cause image contrast to decrease and noise to increase. This not only affects detection and OCR accuracy but also reduces system performance from approximately 18 FPS in the morning to 11 FPS at night due to the longer processing time for each frame. In terms of scalability, the system has the potential to be applied to larger parking areas by adding multiple ESP32-CAM units connected to the same server and database. However, increasing the number of cameras will increase the requirements for network bandwidth, storage capacity, and computing resources. Practically, the proposed system can be used as a low-cost smart parking solution to support vehicle access automation and real-time data recording.

Research Limitations

This study still has several limitations. Testing was only conducted on 20 four-wheeled vehicles in one location with a fixed camera position, so the results of the study are more focused on evaluating the system's functionality rather than generalizing performance across various operational conditions. The testing also did not include diverse weather conditions, variations in image capture angles, vehicles blocking each other (*occlusion*), or heavy vehicle traffic. In addition, the study has not compared YOLOv8n with other models such as YOLOv5, YOLOv7, MobileNet-SSD, and Haar Cascade. The OCR evaluation is also still limited because it has not tested the condition of license plates that are tilted, dirty, blurred, or experiencing light reflection, and has not used special metrics such as Character Error Rate (CER) and Word Recognition Rate (WRR). Therefore, this study is positioned as a *prototype feasibility study* that aims to evaluate the feasibility of implementing the proposed system.

The main objective of this study is to evaluate the successful integration of a smart parking system based on ESP32-CAM, YOLOv8n, EasyOCR, and automatic gate control in a single IoT architecture. Therefore, testing was conducted as a *feasibility study* to verify the overall functionality of the system. Evaluation of model performance using a large-scale dataset is beyond the scope of this study and will be a direction for further development. In addition, system performance is still affected by lighting conditions and image quality obtained from the ESP32-CAM camera.

Further Research Development

Further research can be conducted by increasing the amount of test data, expanding the research location, and testing the system under various weather conditions, lighting conditions, and vehicle density. Comparative evaluation with other object detection models is also needed to obtain a more comprehensive performance picture. Furthermore, future research can focus on improving OCR accuracy, optimizing driver detection in low-light conditions, implementing CER and WRR metrics, and integrating the system with web-based platforms or mobile devices to support more flexible parking access monitoring and management.

CONCLUSION

This research successfully designed and developed a prototype of a car detection system entering a parking area based on ESP32-CAM and YOLOv8n integrated with EasyOCR, MySQL database, and an Internet of Things (IoT) based automatic gate control mechanism. The developed system is capable of detecting vehicles in real-time via an ESP32-CAM camera, reading vehicle license plates using EasyOCR, activating automatic gates based on the detection results, and storing vehicle data and documentation into a database for monitoring purposes.

Based on testing on 20 four-wheeled vehicle samples in a limited testing environment, the system successfully detected all tested vehicles and demonstrated that the integration of ESP32-CAM, YOLOv8n, EasyOCR, MySQL database, and automatic gate control can be implemented in real-time to support the detection process of cars entering the parking area. The test results showed that vehicle detection did not produce classification errors in the test scenarios carried out, while system errors were still found in the driver identification process and vehicle license plate reading as indicated by the emergence of false negative values in several test samples. Vehicle detection performance was relatively stable under various lighting conditions, although driver detection and license plate reading performance decreased in low light conditions due to the limited image quality produced by the ESP32-CAM camera.

The results of this study indicate that the proposed approach is suitable for use as an initial validation stage (prototype feasibility study) in the development of a computer vision-based and Internet of Things-based vehicle access automation system. However, this study still has limitations in the number of test samples, the variety of test environments, and the lack of comparison with other object detection methods.

Further research requires testing with a larger data set, a wider variety of environmental conditions, and comparative evaluation against other object detection models to achieve more comprehensive validation. Furthermore, improving OCR accuracy, optimizing detection in low-light conditions, and developing a web-based or mobile monitoring system could be potential future directions.

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

REFERENCES

Biyik, C., Allam, Z., Pieri, G., Moroni, D., O'fraifer, M., O'Connell, E., Olariu, S., & Khalid, M. (2021). Smart parking systems: Reviewing the literature, architecture, and the way forward. *Smart Cities* , *4* (2), 623–642. https://doi.org/10.3390/smartcities4020032

da Luz, GPCP, Sato, GM, Gonzalez, LFG, & Borin, JF (2024a). *Smart Parking with Pixel ROI Selection for Vehicle Detection Using YOLOv8, YOLOv9, YOLOv10, and YOLOv11* . http://arxiv.org/abs/2412.01983

da Luz, GPCP, Sato, GM, Gonzalez, LFG, & Borin, JF (2024b). *Smart Parking with Per-Pixel ROI Selection for Vehicle Detection Using YOLOv8, YOLOv9, YOLOv10, and YOLOv11* . http://arxiv.org/abs/2412.01983

Dandy Nugraha, Tohir, T., & Muhammad Ilman, S. (2025). Design and implementation of a miniature smart parking system with an Internet of Things-based parking reservation system. *JITEL (Journal of Telecommunications, Electronics, and Electric Power)* , *5* (3), 167–178. https://doi.org/10.35313/jitel.v5.i3.2025.167-178

Enggal Mukti Muhammadiyah University Prof. Dr. Hamka, D. (nd). Ministry of Communication and Digital Smart Parking System Based on Internet of Thing. In *JAMASTIKA* (Vol. 4).

Fathurrahman, M., Nugroho, A., & Al Wafi, AZ (2025). Comparative Study of YOLO Versions For detecting vacant car parking spaces. *JITK (Journal of Computer Science and Technology)* , *10* (4), 833–848. https://doi.org/10.33480/jitk.v10i4.6236

Gokulkrishna, S., Harsheetha, J., Akshaya, S., & Jeyabharathi, D. (2021). An IoT based Smart Outdoor Parking Systems. *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021* , 1502–1506. https://doi.org/10.1109/ICACCS51430.2021.9441766

Gunawan Zain, S. (2022). *Development of a Home Security Monitoring System Using Nodemcu-Based CCTV* . *5* (3).

Impana Manik, K., Kiswanto, D., Defiyanti, A., Shaleh Lbn Gaol, A., Computer Science Study, P., Medan State, U., Willem Iskandar, J., Estate, M., Percut Sei Tuan, K., & Deli Serdang, K. (2026). Integration of Ultrasonic Sensors and Computer Vision (YOLO) Based on ESP32-CAM for Object Classification in Parking Systems. *National Journal of Computing and Information Technology (JNKTI)* , *9* (1).

Jabbar, WA, Wei, CW, Azmi, NAAM, & Haironnazli, NA (2021). Raspberry Pi IoT-based parking management system for smart campus [Formula presented]. *Internet of Things (Netherlands)* , *14*. https://doi.org/10.1016/j.iot.2021.100387

K, VD, Bhatta, A., & Ghimire, T. (2022). IoT-Based Smart Parking System. *International Journal Engineering and Advanced Technology* , *11* (5), 105–108. https://doi.org/10.35940/ijeat.E3583.0611522

Kusuma, VA, Arof, H., Suprapto, SS, Suharto, B., Sinulingga, RA, & Ama, F. (2023). An internet of things-based touchless parking system using ESP32-CAM. *International Journal of Reconfigurable and Embedded Systems* , *12* (3), 329–335. https://doi.org/10.11591/ijres.v12.i3.pp329-335

Prasetyo, D., Pranata, A., & Ramadhan, PS (2021). "Implementation of the Internet of Things (IoT) in Smart Microcontroller-Based Parking System for Apartment Owners * Computer Systems, STMIK Triguna Dharma ** Computer Systems, STMIK Triguna Dharma ** Computer Systems, STMIK Triguna Dharma. *CyberTech Journal* , *4* (6). https://ojs.trigunadharma.ac.id/

Raj, A., & Shetty, S.D. (2024). Technologies, tools, and challenges of smart parking systems for implementation in smart city environments: a survey based on IoT & ML perspectives. *International Journal of Machine Learning Learning and Cybernetics* , *15* (7), 2673–2694. https://doi.org/10.1007/s13042-023-02056-5

Ratakonda, B., Avvari, P., Rajarao, B., Sreevani, V., & Ganapathi Raju, N.V. (2021). Smart Parking for Smart Cities using IoT. *E3S Web Conference* , *309*. https://doi.org/10.1051/e3sconf/202130901128

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Integrated, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Acknowledgements* , *December 2016* , 779–788. https://doi.org/10.1109/CVPR.2016.91

Salsabila, N., & Sriani, S. (2024). Improving Automatic Vehicle License Plate Recognition with YOLOv8 and EasyOCR. *Journal of Information Systems and Informatics* , *6* (3), 1577–1597. https://doi.org/10.51519/jurnalisi.v6i3.848

Sarhan, A., Abdel-Rahem, R., Darwish, B., Abou-Attia, A., Sneed, A., Hatem, S., Badran, A., & Ramadan, M. (2024). Egyptian vehicle license plate recognition based on YOLOv8, Easy-OCR, and CNN. *Journal of Electrical Engineering Information Systems and Technology* , *11* (1). https://doi.org/10.1186/s43067-024-00156-y

*name of corresponding author