The International Society of Data Scientists
Journal of Data Science and Artificial Intelligence
Vol. 4, No. 1. Online ISSN: 2831-4794

# ASPS - AI-Driven Smart Parking Solutions with Integrated Web and Mobile Applications

Nhat Nguyen Huu Minh[a, b], Tin Pham Quoc*[a, b], Tan Duy Le[a, b]
[a]School of Computer Science and Engineering, International University, Ho Chi Minh City, Vietnam
[b]Vietnam National University, Ho Chi Minh City, Vietnam
*Corresponding author. E-mail: phamquoctin01@gmail.com; phone: +84-936768837

**ABSTRACT**

The research suggests a smart parking system using the combination of AIoT and Edge Computing, in there, Raspberry Pi collects the data from cameras and sensors and then transfers it to the local computer for AI to process and overcome the performance limitations of the Raspberry Pi and create the comprehensive parking system including the website for admin and mobile application for users. Previous research on smart parking systems has evolved both simple solutions using sensors to advanced systems that integrate AI and cameras (dual cameras system with 96.74% accuracy and fisheye camera system with YOLOv5 reached 94.3%), however, there are still some restrictions covering brightness, weather conditions, etc. The key features include license plate detection using YOLOv10, displaying superior results than others. This smart parking system solves the existing issues by leveraging edge computing to handle these AI tasks. Data collected from cameras and sensors will be processed locally, ensuring real-time performance. The web platform allows remote management, while the mobile app provides detailed information about the driver and also the online payment function. Future enhancements cover the slot reservation function and AI-powered available space detection besides other factors such as reducing traffic congestion, improving parking management, and decreasing pollution. The project will offer a scalable solution for urban areas with high demand for vehicles like Ho Chi Minh City combined with low-cost hardware/sensors and advanced AI models, from there can contribute to the development of a smart city with efficient and user-friendly smart parking system.

**Keywords:** YOLO, Raspberry Pi 4, External USB Camera, Web/Mobile Application, Internet of Things (IoT).

## 1. INTRODUCTION

The combination of Artificial Intelligence (AI) and the Internet of Things (IoT) [1], also known as AIoT, has become one of the most significant modern technological trends nowadays. AIoT systems enable enhanced machine interactions, efficient data management, and advanced analytics. These capabilities have been applied to many different fields, including smart homes, security systems, air quality management, energy grid optimization, healthcare, and weather forecasting. Despite these advantages, one painful challenge in densely populated urban areas, such as Ho Chi Minh City, is the missing of safe and efficient parking solutions. With over 10 million people and increasing traffic demands, the city faces issues such as illegal parking, traffic congestion, air pollution, and noise. The existing solutions provide only short-term, failing to address the root causes of parking inefficiencies, and sometimes exacerbating problems like theft and overcrowding.

Recent research on smart parking systems has commonly used the Raspberry Pi 4 for processing raw data. While the Raspberry Pi 4 is convenient and supports a wide range of functions including Wi-Fi, GPIO pins, and camera integration, however, this mini-computer still struggles to handle the heavy tasks that are required for advanced AI models. This is because its processing power isn't designed for deep learning and real-time analytics works.

As a result, when complex AI models are run directly on the Raspberry Pi 4, the system often faces slow response times, lower accuracy, or difficulty managing large numbers of data. Examples include tasks similar to detecting license plates or analyzing vehicle images in real-time can experience delays or errors, which can affect the effectiveness of the system—especially in busy environments where speed and accuracy are the most essential. Moreover, the limited computational capacity makes it tough to scale the system or integrate more advanced AI techniques.

To tackle these issues, edge computing technology [2] has become a practical solution. Instead of relying on all the Raspberry Pi, now, AI tasks are offloaded to a local machine with greater processing power. By leveraging edge devices, data is processed close to the source, allowing for faster response times, reduced latency, and enhanced system reliability.

In general, raw data from cameras and sensors are detected from USB cameras, which are connected to Raspberry Pi, and are sent to the local machine, where advanced AI models (YOLO) for object detection, are executed. The results are then sent directly to the React system in real-time to ensure smooth operations.

The study's objectives include creating a web-based platform for administrators and a mobile application for users. The website will enable remote management of parking spaces via a Wi-Fi connection and provide a dashboard with statistics, vehicle information, and license plate images. The mobile application will offer users detailed parking information, calculate parking fees, and facilitate payment through credit or debit cards. Future features include a vehicle search function, allowing users to locate their vehicles by entering license plate details. This data, analyzed by the AI model, will retrieve matching results and display relevant images.

The main contribution of this paper is as follows:

- Present the design and implementation of a real-time smart parking system, with a particular focus on edge computing implementation.
- Low-cost Sensors and Cameras Implementation.

## 2. BACKGROUND AND RELATED WORKS

Numerous research studies have explored different implementations of AIoT-based smart parking systems, analyzing their strengths and limitations. For instance, Telefonica I+D and the University of Cantabria [3] developed an IoT solution to alleviate traffic congestion by employing sensors distributed across the city. These sensors update parking availability on display boards at intersections, using data collected from parking bays, Pham Thanh Nam proposed a cost-efficient system integrating RFID and QR codes [4], complemented by a web portal and mobile app for checking parking slots. The system incorporates ultrasonic sensors to enhance automation levels. In addition, TIS Corporation in Vietnam deployed an automated parking management system combining license plate recognition and video surveillance with advanced video analytics.

Prabhu Ramaswamy's work further explored the integration of Raspberry Pi with Pi cameras and ultrasonic sensors for vehicle number plate recognition [5]. However, such systems face limitations due to the lack of advanced AI and deep learning applications, which restrict their ability to process inadequate input data effectively, resulting in suboptimal performance. OCR technology, although simple to implement, struggles with noisy or degraded plates, leading to low accuracy. The adoption of AI models like YOLO (You Only Look Once) represents a significant leap forward in overcoming these challenges. YOLO version 8, in particular, achieves an impressive 92% accuracy in real-time license plate detection. Its ability to detect multiple objects simultaneously with a single network transfer makes it ideal for smart parking applications. Despite its high accuracy, challenges remain, such as overlapping bounding boxes when objects are close together and difficulties in detecting plates under low-light conditions or from varied angles. To address these issues, techniques like image preprocessing for brightness enhancement, multi-angle camera placements, and diverse dataset training can improve detection performance [6].

Another notable system is the Social IoT-based platform that uses magnetometer sensors to detect vehicles at individual parking spots. The sensor data is collected via Wi-Fi and processed through cloud databases, with real-time updates provided through an Android app. This app not only displays parking availability but also supports online payments linked to credit or debit cards for seamless fee transactions. The system also utilizes Bluetooth Low Energy (BLE) for short-range communication, enabling localized processing between concentrators and electronic dashboards to analyze vehicle information [7].

The web application allows drivers to make online reservations for parking slots, individual users can see available parking spaces, choose their parking duration, input their vehicle details, and make their payment through the Internet. When the vehicle reaches the parking lot, an electrical barrier automatically checks the vehicle information using the license plate captured by cameras and compares it with the database. The mobile application platform also enhances this by allowing drivers to search for locations, navigate to their designated parking spaces, and extend their parking time if necessary.

After reviewing the web and mobile application for the users, the next part is to display the strengths and weaknesses, this smart parking system based on IoT has many advantages, including saving time and effort for drivers who no longer have to spend time looking for a parking spot, the result is that this also helps in reducing traffic congestion, enables efficient management by offering real-time information about the parking lot, which aids in effective monitoring and supervision. Beyond that, it also aids in reducing pollution by decreasing the amount of time spent searching for parking, which in turn contributes to a decrease in carbon emissions and the protection of the environment. In addition, the system includes electronic payment options, allowing for convenient and speedy parking fee transactions. Nevertheless, there are some drawbacks to the system, such as the absence of physical locking options for public parking and the lack of capabilities to analyze driver habits. Moreover, for accurate results, the license plate detection algorithm necessitates the vehicle to be fully immobilized. To overcome these limitations, experts suggest employing machine learning to examine data and anticipate driver actions, ultimately making the system more efficient and delivering improved services. They also recommend improving security measures to safeguard user data and prevent unauthorized entry [8].

Groups of authors from the topic Proposing Real-time Parking System for Smart Cities using two cameras [9] have brought to us a smart parking system working in real-time with the application of two cameras to detect the available slots in the parking space and identify the accurate information of the vehicle license plate. This system is designed to solve some particular issues compared to traditional parking systems, such as the lack of parking spaces, traffic congestion, and carbon emissions. The system includes two main modules: the Skyeye module and the LPR module, in detail, the Skyeye module uses the wide-angle camera to detect and monitor vehicles. Available spots are also the strength function of this module combined with sharing the status of parking with the user. When a new vehicle comes to the detection area, the Skyeye module will activate and send the data to the LPR module. LPR module - this module uses the high-definition camera to detect and identify the license plate, the mechanism action of this module is activated by the signal from the previous module Skyeye, and extracts all the alphabet numerals on the vehicle license plate. The system processing is described in detail by the start of the Skyeye module continuously working and progressing the images from the Skyeye's cameras, then, when the new vehicle arrives at the entry gate of the parking detected, the Skyeye module will activate the LPR module to detect and analyze the license plate. After analyzing all the characters on the license plate, the system will attach the ID for the new vehicle by the Centroid algorithm, and in the end, the system will update and display the information of parking slot status. Testing results of experiments of this system showed a high percentage of precision of detection with 96.74 in the reality environment in general. In detail, compared to YOLO version 2, the Skyeye module reached a precision of 99.48%, which is an outstanding result. Similar to the Skyeye module, the LPR module shows 94 percent accuracy along with the VN_LP module and an average percentage for the two modules is 96.74 indicating that this system can be highly recommended for application in reality. However, this system still has some weaknesses including the quality of input images may affect the accuracy of the detecting system, in addition, the system at the moment, cannot process all the reality situations for example weather conditions (sun, rain, day/night).

The established journal indicated an AIoT system (Artificial Intelligence of Things) for smart parking [10], using the fisheye camera - the surveillance camera that uses curved lenses, thanks to that users can observe wide-angle panoramic views of 180 degrees and 360 degrees, without being limited by blind spots and deep learning by YOLO version 5. This system also can detect the license plate and check the status of parking slots instead of using many expensive sensors such as ultrasonic sensors or magnetic field sensors. Before going into detail about how this system works, we need to mention the advantages and disadvantages of the fisheye camera, to begin, the advantages of this camera are expressed through the wide area of the monitor - this is one of the big advantages of this system because it helps this system reducing the cost by cutting off many cameras, however, in some cases, this advantage can cause opposite effect because the input images will be distorted so they need to have distortion technique and also the unwrapping technique to process images. Moreover, missing the open training datasets of detecting the vehicle from a fisheye surveillance camera may affect the precision when detecting different types of vehicles or license plates. Deep inside the system, the authors describe how the system works, firstly, applying the LaRecNet algorithm to handle the distorted images from the fisheye camera, then the YOLO version 5 model will be used to detect the area of the image that contains the license plate. Based on the position of the license plate relative to the area identified as a parking space, the system determines whether the parking space is empty or occupied, tracking the license plate and attaching the only typical ID for individual cars to manage the status of parking space by using the SORT technique. In the performance evaluation, the result shows that YOLO version 5 reaches a precision of approximately 94.3 percent with the time processing is 0.071 seconds, outstanding compared to other methods in the precision, speed, and size of datasets. To sum up, this AIoT system using the fisheye camera and YOLO version 5 is the effective solution and saves costs to detect the license plate along with checking the status of the parking space.

While these systems offer numerous—such as saving drivers time, reducing traffic congestion, and decreasing pollution by minimizing the time spent searching for parking—they still face limitations. These include the absence of physical locking mechanisms for public parking, the lack of driver behavior analysis, and the dependency on stationary vehicles for accurate license plate detection. Experts recommend employing machine learning to predict driver actions, enhancing user data security, and integrating automated features to address these shortcomings.

# 3. METHODOLOGY

## 3.1 License plate parameters selection

The proposed system for vehicle license plate detection has been designed to leverage AI-driven object detection and computer vision technologies. The following components and technologies have been identified for their effectiveness in meeting the needs of this real-time, automated detection system:

- **YOLO (You Only Look Once) Model**: This advanced object detection algorithm has been chosen for its high accuracy (96.95%) in detecting vehicle license plates in real time. YOLO is particularly useful for processing multiple objects in a single frame, which is crucial for handling busy parking lots or areas with multiple vehicles.
- **Raspberry Pi 4 with Camera**: A Raspberry Pi 4 paired with a high-resolution camera is used for capturing real-time footage. Its compact size and affordability make it an ideal option for deploying smart parking solutions in urban environments.
- **Computer Vision Techniques**: To enhance license plate detection, preprocessing methods such as image enhancement, brightness adjustments, and contrast modifications are employed. These help improve accuracy in challenging conditions like low light or glare from headlights.
- **Machine Learning for Data Optimization**: Machine learning models are applied to optimize the detection process, especially in scenarios where plates may be obscured, or the camera angle is not ideal. Training the system on diverse datasets ensures robust performance under various environmental conditions.
- **Cloud Integration for Data Storage**: Although the initial processing is handled locally via edge computing, cloud integration is used for storing detected license plates and related metadata for further analysis, reporting, or monitoring.

## 3.2 Data collection

A crucial aspect of this study was the collection and preparation of a comprehensive dataset to train our models effectively. For this purpose, we utilized a dataset of **20,000 images**, specifically focusing on license plates and vehicles from **Vietnam**. This localized approach ensures that the system is well-suited to the unique characteristics and conditions commonly found in Vietnam, such as specific license plate formats and environmental factors.

To build a reliable system, the dataset includes images representing various challenges commonly encountered in real-world applications:

- **Environmental Variations**: Images captured in different weather conditions (sunny, rainy, foggy) and lighting scenarios (daytime, night-time, low-light).
- **Vietnamese License Plate Format**: Often includes a single-row plate at the front of large vehicles (cars, trucks, etc.) and a two-row plate on motorbikes, as well as at the back of large vehicles.
- **Angle Variations**: License plates captured from multiple angles, including straight-on, tilted, and side perspectives.
- **Noise and Obstructions**: Data with blurred plates, dirt-covered plates, or partial obstructions to train the model for better generalization.

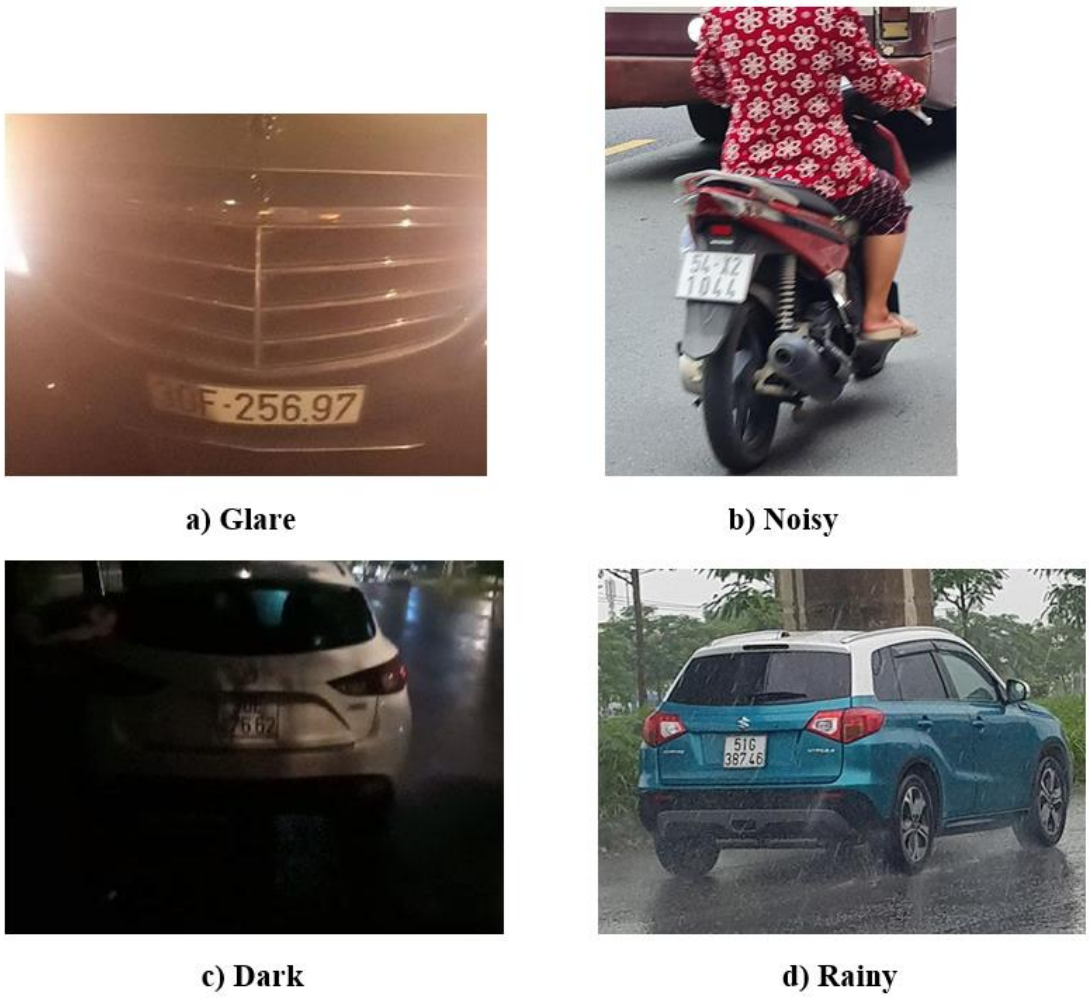



Figure 1: Diversity of environmental conditions

### 3.3 License plate detection

In this part of the study, we've focused on **License Plate Detection**. This stage plays a crucial role in the overall system but also introduces additional complexity, particularly when it comes to processing speed and system costs. As we move through these stages, the demand for computational resources increases, making it necessary to carefully optimize for efficiency and accuracy.

Fortunately, advancements in object detection models have allowed us to address these challenges effectively. The YOLO (You Only Look Once) model family, specifically versions 8, 9, 10, and 11, has shown impressive capabilities in recognizing even small objects with high accuracy. This is particularly important for license plate detection, where precision is critical for identifying plates in real time under varying conditions.

For this study, we've designed a series of detection models based on YOLOv8-m, YOLOv9-c, YOLOv10-b, and YOLOv11-l, each optimized for specific needs. The input image size for these models is set at 640, ensuring a balance

between high resolution and fast processing speeds. These models are tailored to work seamlessly, providing real-time detection of both vehicles and license plates with remarkable accuracy.

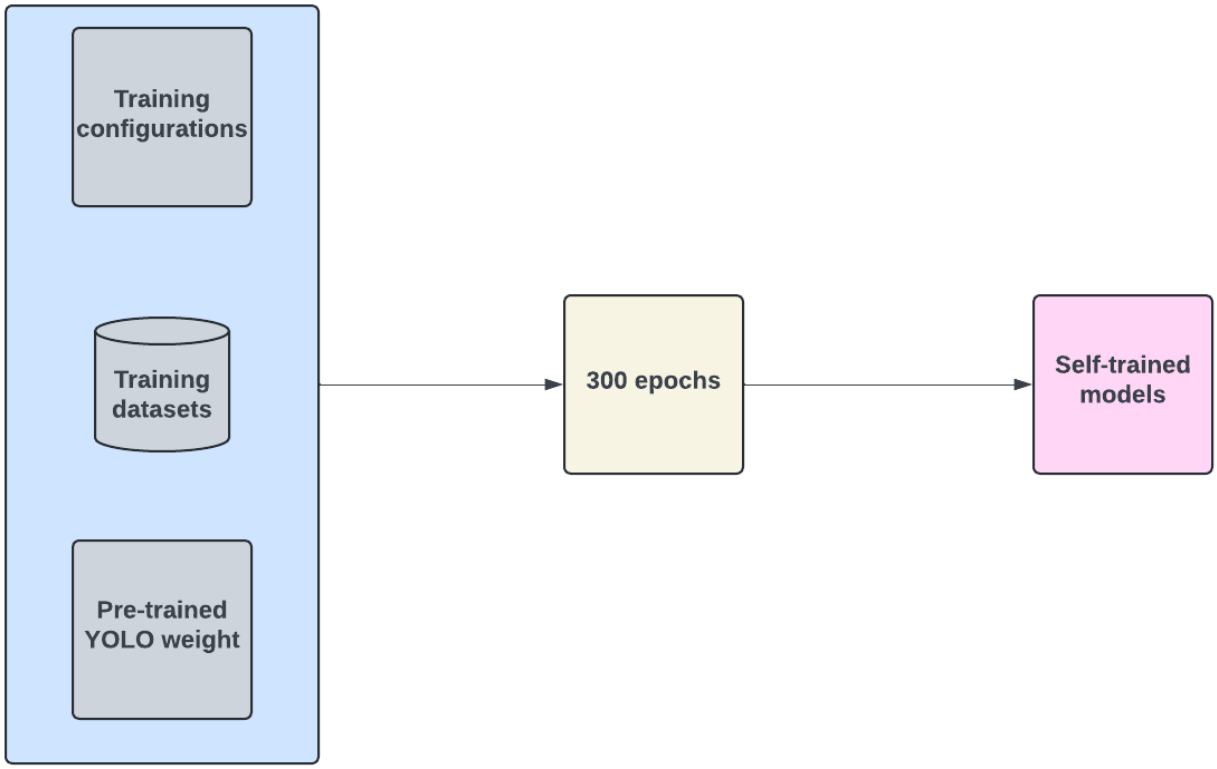



Figure 2: Training Pipelines

By leveraging the strengths of each YOLO version, we can process images efficiently, even in dynamic environments, ensuring that our smart parking system remains responsive and accurate. Despite the increasing complexity at each stage, the power of YOLO's advanced algorithms allows us to maintain optimal performance, enabling real-time detection without compromising the user experience.

*Contrast enhancement:* First, to improve image clarity, the original BGR image is first converted to the Lab color space, separating lightness (L) from color (a and b). We then apply CLAHE (Contrast Limited Adaptive Histogram Equalization) [18] to enhance the lightness channel, making features more visible, especially in low-light conditions.

$$H(r) = \sum_{s=0}^{r} for\ p(s)\ for\ r = 0, 1, 2, \dots, L-1 \quad (1)$$

where:

- p(s) is the probability of the intensity values.
- L is the number of intensity levels (256 for 8-bit images).

After that, we used with Hough Transform Algorithm to identify the edge and structure of the plate. This method works by mapping edge pixels detected by the Canny edge detector into a parameter space defined by $\rho$ and $\theta$, where a line is represented as:

$$\rho = x\,.cos\,(\theta) + y\,.sin(\theta) \quad (2)$$

Here, $\rho$ is the perpendicular distance from the origin to the line, and $\theta$ is the angle of the line's normal.

We applied the **Probabilistic Hough Transform**, which efficiently detects and outputs the endpoints of line segments. This approach allowed us to focus on significant structural lines while ignoring smaller, irrelevant details. The transform was configured with specific parameters: a resolution of $\Delta\rho = 1\ pixel$ and $\Delta\theta = \frac{\pi}{180}\ radians$, a threshold for minimum votes to confirm a line, a minimum line length of $\frac{\omega}{1,5}$, and a maximum line gap of $\frac{h}{3.0}$. These settings ensured the accurate detection of longer and continuous lines, which are crucial for identifying the plate's structure.

The resulting line segments were further analyzed to calculate the skew angle of the plate. This was done by averaging the orientations of prominent lines while excluding outliers, such as extreme angles caused by noise. The computed skew angle was then used to deskew the image, ensuring the plate was properly aligned for further processing steps.

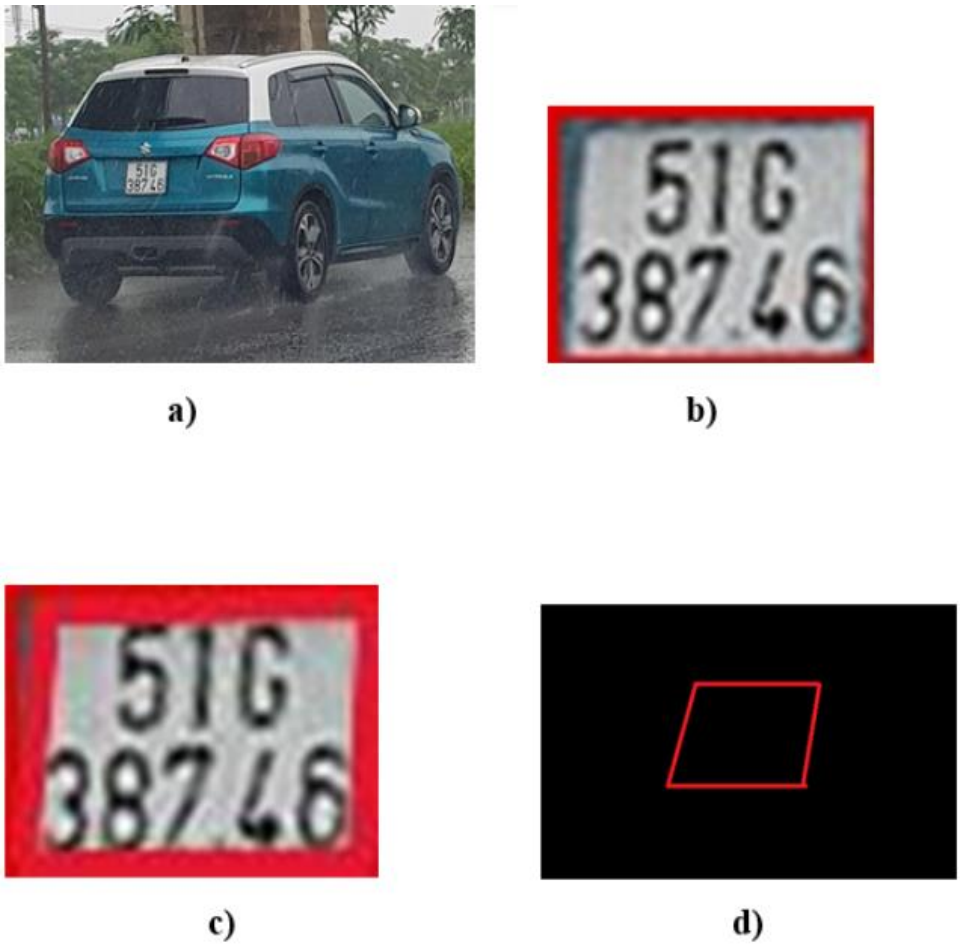



Figure 3: Frame of the license plate after being captured

To correct the skew of the license plate, we used a **rotation matrix** to adjust the image orientation based on the calculated skew angle. A rotation matrix is a mathematical transformation that rotates an image around a specified point, typically the image center, without altering its shape or size. For a given rotation angle $\theta$, the matrix is defined as:

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & tx \\ \sin(\theta) & \cos(\theta) & ty \end{bmatrix} \quad (3)$$

Here:

- $cos\ (\theta)$ and $sin\ (\theta)$ define a rotation.
- $tx$ and $ty$ are translation offsets to ensure the image rotates around its center.

In our implementation, the rotation matrix is computed using OpenCV where the center of the image serves as the pivot point. The translation offsets $tx$ and $ty$ is automatically adjusted to prevent cropping during rotation. The skew angle, derived from the Hough Transform, is passed $\theta$ to align the image properly.

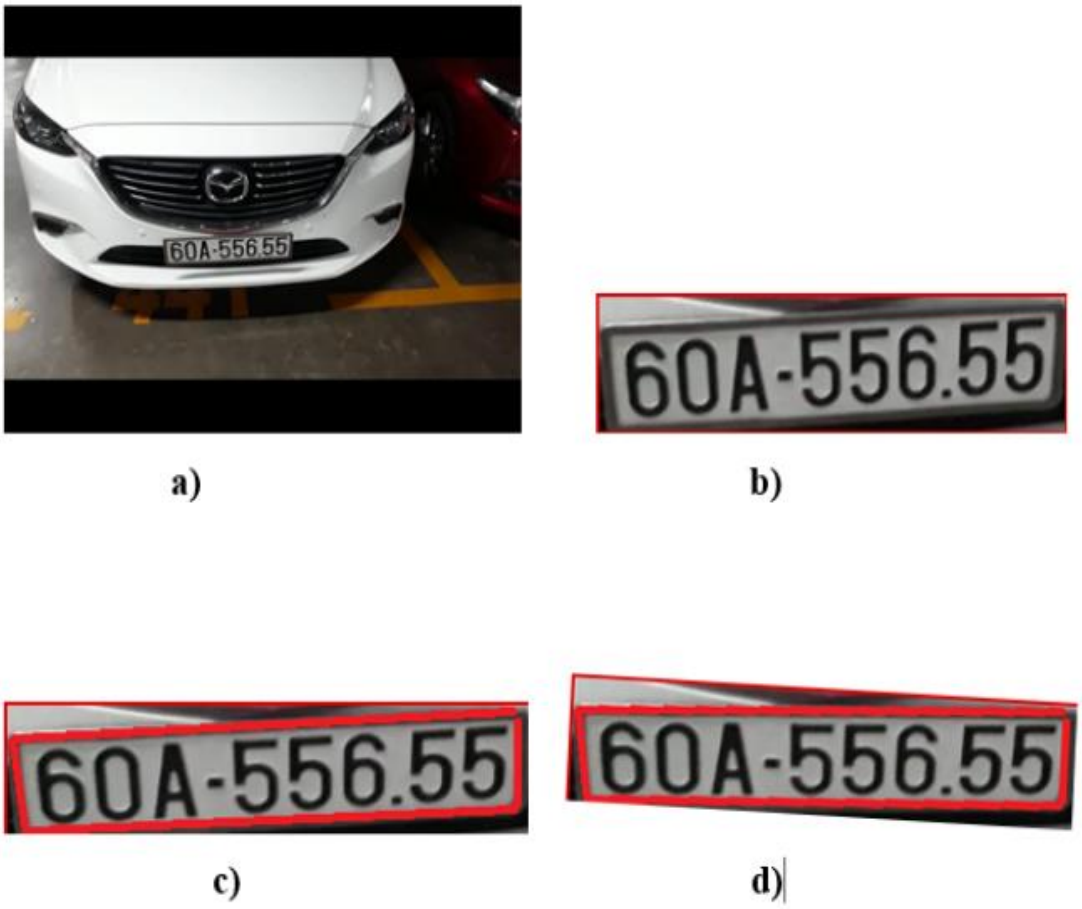



Figure 4: Result after skewing the image

### 3.4 Character segment recognition

In the next stage of the study, we implemented **Character Detection**, a critical component for accurately recognizing and processing the information displayed on detected license plates. This stage is essential for transforming image data into meaningful alphanumeric information that can be stored, analyzed, or used in decision-making processes. This stage almost uses the processed image in the previous state (License Plate Detection) to process, which is in Figure 7.

We also utilized YOLO models to train the **Character Detection model**, specifically versions 8, 9, 10, and 11 because each version brings unique enhancements that improve object recognition accuracy and speed. By integrating these

strengths, we have developed robust detection models based on YOLOv8-m, YOLOv9-c, YOLOv10-b, and YOLOv11-l. These models have been optimized to handle a wide range of license plate conditions, ensuring adaptability to real-world scenarios.



Figure 5: Result after the Character Recognition model is trained

### 3.5 Edge computing module

This study introduces our smart parking system, which utilizes an Edge Computing Module to offload computationally intensive tasks from the Raspberry Pi. Recognizing that the Raspberry Pi lacks the processing power required to handle advanced AI models like YOLO, we implemented a practical workaround to address this limitation. Specifically, our system employs two separate AI models, which would otherwise overwhelm the Raspberry Pi's capabilities.

To overcome this challenge, we decided to use a method that connect the Raspberry Pi to a local machine equipped with an external GPU. This approach allows the computationally demanding tasks to be executed efficiently on the external GPU while the Raspberry Pi focuses on less resource-intensive functions. By distributing the workload, we achieve a seamless and effective solution, ensuring that our smart parking system operates smoothly in real-time without compromising on performance or accuracy.

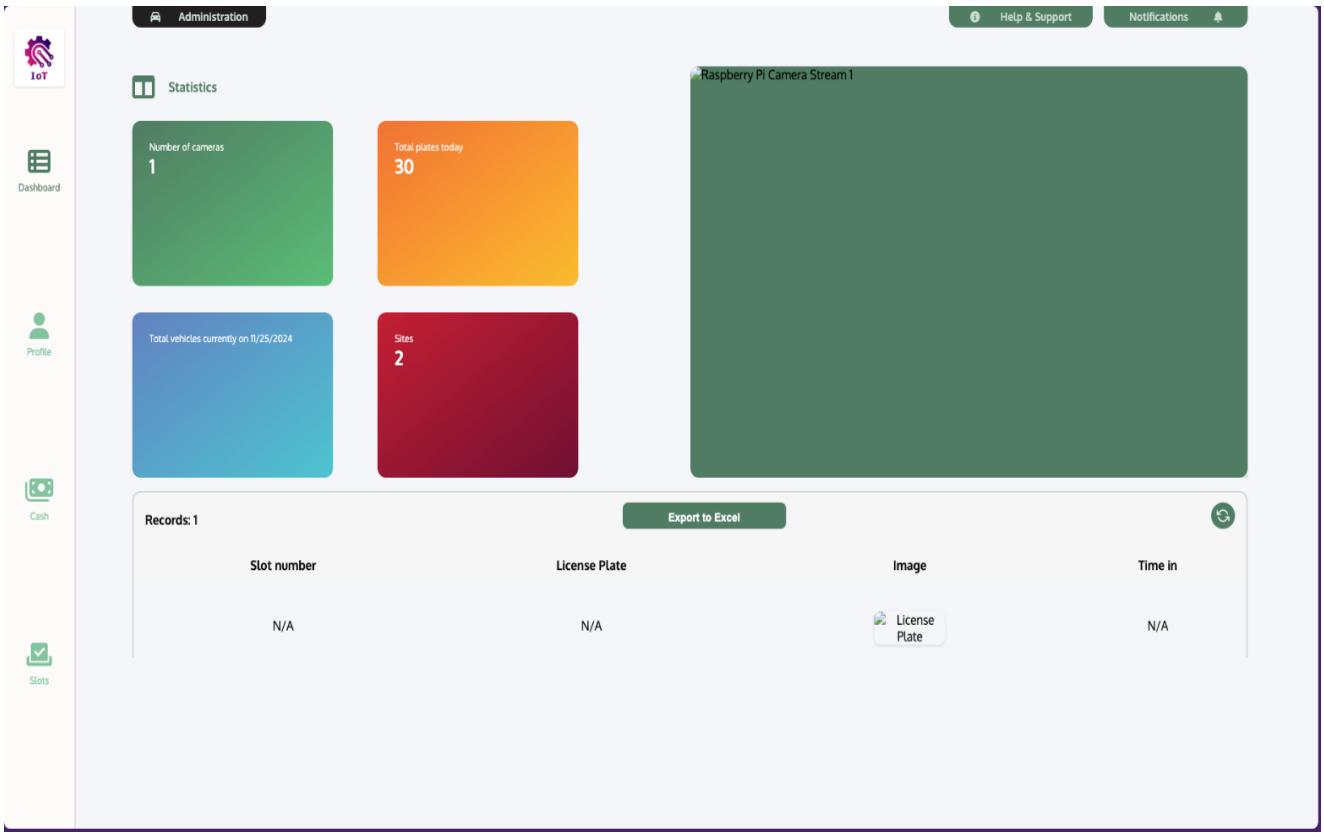



Figure 6: A web application dashboard overview

Moving to Figure 6 where the Dashboard of the website is located, this page has the mission to contain and update new raw data every day by dividing different colorful squares on the left-hand side, these squares including the statistics of the camera(s) exist in the parking slot, the sum of vehicle’s plates in a day,... Moreover, the right-hand side established the screen of the live stream camera that is connected directly to the Raspberry Pi 4 for surveilling and also increasing the security in the parking space, furthermore, after the raw data have been processed, these data will be presented into the divided columns table below covering with an individual title like Slot Number, License Plate, Image and Time In to the parking slot that all are fetching in real-time through the Firestore database.

# 4. IMPLEMENTATION AND EVALUATION

## 4.1 System Design

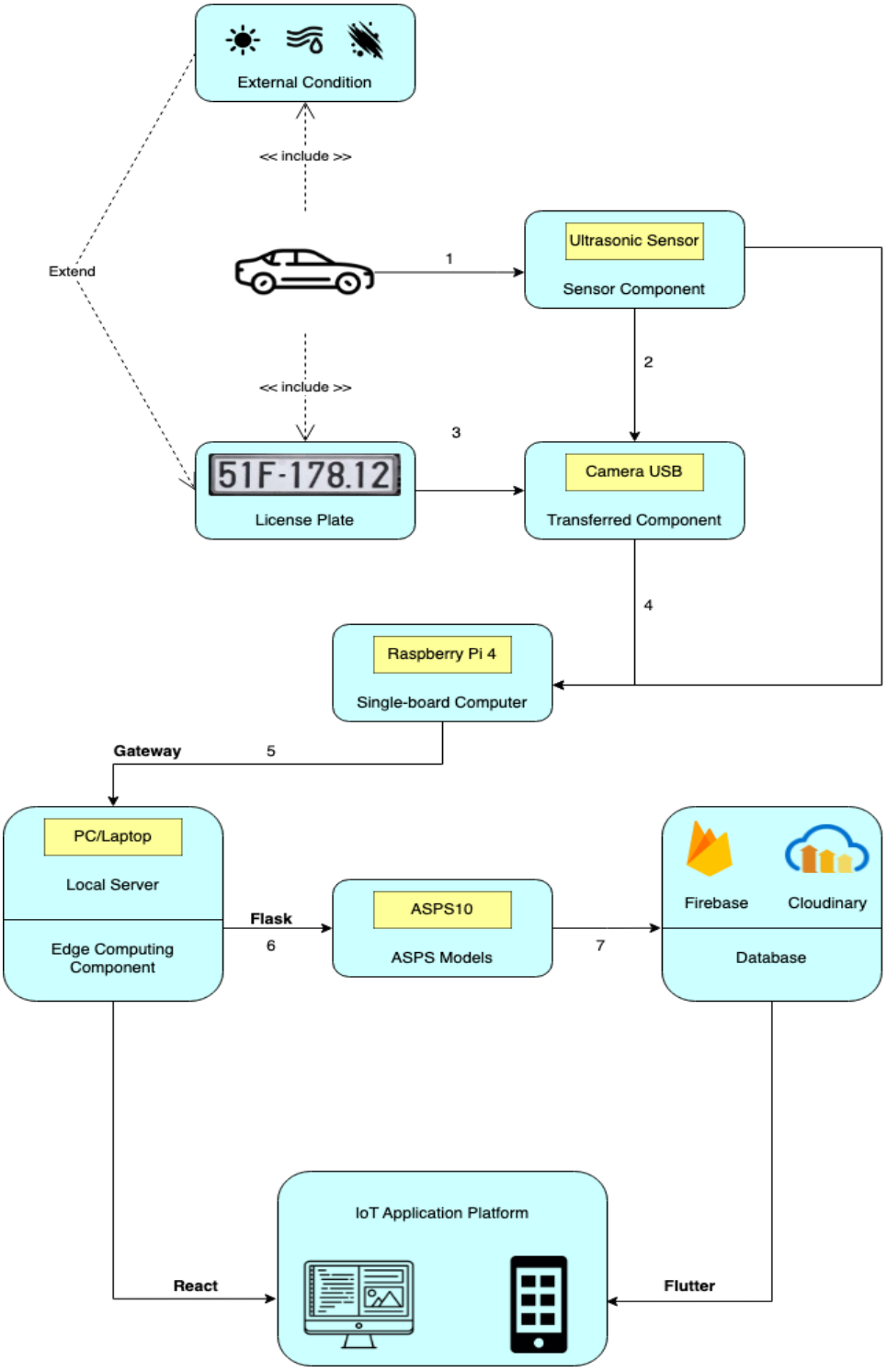



Figure 7: Real-time License Plate Processing System Diagram

The proposed system is illustrated in Figure 7, comprising several interconnected components: a sensor component for vehicle detection, a microcontroller component for data preprocessing, a transfer component for data delivery, and an edge computing component for processing and integrating data with IoT platforms. This architecture enables real-time license plate recognition and management.

The system workflow begins with an ultrasonic sensor detecting the presence of a vehicle in step 1. Upon detection, a USB camera captures the license plate image and transfers it to the Raspberry Pi 4 via a USB connection. As step 4, the Raspberry Pi preprocesses the image before forwarding it to a local server using a wireless gateway. On the local server, the ASPS10 model processes the data for license plate recognition. The processed information is then uploaded to a cloud database (Firebase and Cloudinary) for storage. APIs are used to visualize and interact with the data on web (React) and mobile (Flutter) applications, ensuring a seamless user experience.

### 4.2 Implementation

Our license plate detection project utilizes advanced technology to identify and recognize vehicle license plates efficiently. The system integrates real-time image processing with local computing power to ensure high accuracy and fast detection speeds. By leveraging the research and optimization of detection parameters, it can accurately identify license plates in various conditions, including low light and motion blur.

The project is designed to operate on scalable, cost-effective infrastructure, utilizing Raspberry Pi 4 Model B and open-source tools to streamline deployment. With the use of appropriate AI libraries and frameworks, the system provides users with an intuitive and seamless experience. Aimed at improving traffic management and enhancing security, this project contributes to a broader goal of supporting smart city initiatives. All components and tools are accessible, promoting openness and encouraging further innovation.




Figure 8: External hardware

By hosting the server locally, we ensure effective management of any issues that arise while maintaining full control over its performance. We also use Cloudinary as cloud storage to save all the captured images and display them in our web application. We recommend leveraging a server-side API integrated with an optimized web page and mobile application for seamless access to detection results and system status from any location.

The website's front end as shown in Figure 6, and with an overview of the result in Figure 9, built using the React framework, organizes the user interface into functional areas, including data visualization and system monitoring.

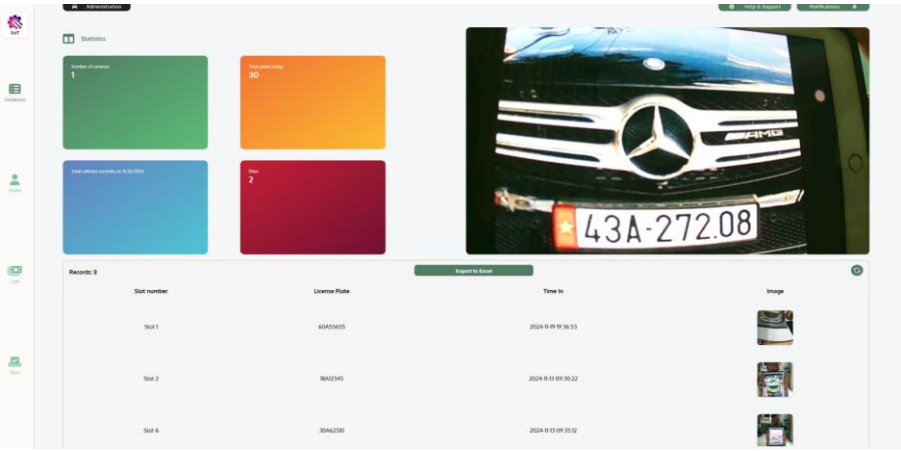



Figure 9: A dashboard with visualizing license plate data

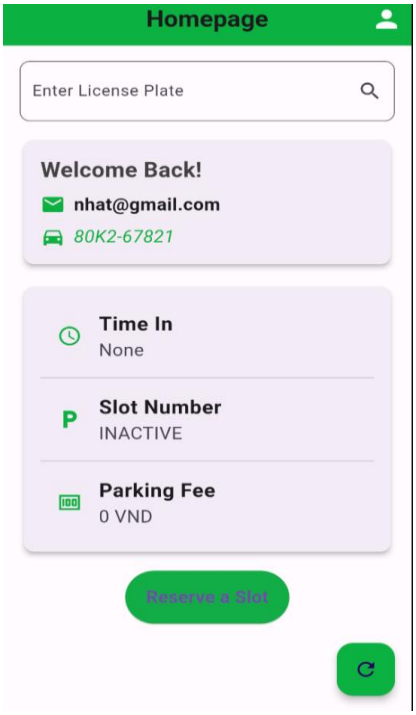



Figure 10: Homepage screen with basic information and reserve slot

Another site that this project has included besides the website for administrator above is the mobile application - where the screen will show the user some basic information that fetches at the same database in Firestore with the website like Time In, Slot Number, and Parking Fee. Additionally, we also designed two brand-new functions including a bar for searching for a vehicle in the parking slot more quickly and reserving a slot before coming to the parking slot.

### 4.3 Evaluation

Table 1: Average evaluation with some previous methods

| Ref. | **Image total** | **Correct rate** | **Error rate** | **Average (%)** | **Dataset Source** | **Challenges Addressed** | **Processing Speed** | **Hardware Used** | **Remarks** |
|---|---|---|---|---|---|---|---|---|---|
| Ref. [11] | No | No | No | 38.63 | - | Limited contextual testing | - | Not mentioned | Low accuracy due to limited scope |
| Ref. [12] | 700 | 682 | 18 | 97.43 | Private | Illumination and angle variations | - | Desktop GPU | High accuracy on small dataset |
| Ref. [13] | 1000 | 910 | 90 | 91.00 | - | Basic illumination challenges | - | Not mentioned | Generalizable but moderate accuracy |
| Ref. [14] | 1176 | 1125 | 51 | 95.7 | Private | Angle and background variations | ~1.12s/image | Desktop GPU | Robust for specific datasets |
| Ref. [15] | 11696 | 11111 | 585 | 95.00 | Private | Background and plate dimension variations | ~0.003s/image | GTX 1080 | Well-suited for high-end hardware |
| Ref. [16] | 805 | 747 | 58 | 92.85 | - | Background clutter and weather conditions | ~0.65s/image | Desktop GPU | Lightweight and efficient system |
| Ref. [17] | 900 | 870 | 30 | 96.7 | Public | Basic illumination challenges | ~0.06s/image | RTX 5000 | Optimized for edge devices |
| **Our method** | **20000** | **19390** | **610** | **96.95** | **Custom** | All (illumination, angles, weather, etc.) | ~0.04s/image | Rasberry Pi 4 + RTX 3080 | Most accurate and scalable solution |

In general, Table 1 highlights the effectiveness of the proposed method in solving challenges such as illumination, background variation, and license plate dimensions. The clear improvement in performance across a large dataset validates its practicality and applicability in real-world scenarios.

After completing the implementation of our proposed system, we conducted a detailed evaluation and comparison with related studies. While many existing systems address the fundamental needs of a smart parking solution, several limitations persist. These include insufficiently diverse datasets, a lack of seamless integration between hardware and software components, and minimal or superficial application of AI technologies, all of which hinder their overall performance. Our system has been specifically designed to overcome these shortcomings, focusing on robust AI integration, compatibility across hardware-software platforms, and leveraging a comprehensive, diverse dataset.

Moreover, since AI forms the backbone of our system, we prioritized benchmarking the proposed model against other established models. This comparison allows us to highlight our system's superior accuracy, adaptability, and ability to handle real-world challenges such as varying lighting conditions, different weather scenarios, and non-standard license plate orientations. By addressing these gaps and advancing the application of AI, our proposed solution not only bridges the deficiencies found in prior systems but also sets a higher standard for smart parking technology.

Table 2: Comparison between Proposal Smart Parking System and Related Systems

| Field | Detail | Our propose system | System [17] | System [12] | IseeCarRecognizer [16] |
|---|---|---|---|---|---|
| **Hardware Deployment** | Raspberry Pi | x | | | |
| **Publication** | Open source | x | | | |
| **Result step showing** | Correctness | x | x | x | x |
| | Completeness | x | x | x | x |

This table provides a comparative analysis between **our proposed system** and three related systems in terms of key technical and functional criteria. The purpose of this comparison is to highlight the strengths and unique features of the proposed system. Below are the clarified fields:

- **Raspberry Pi**: Indicates whether the system utilizes a Raspberry Pi, a cost-effective, versatile, and compact hardware platform suitable for edge computing and IoT applications.
- **Publication**: Determines if the system is open-source, allowing other developers and researchers to replicate or modify the project.
- **Result Step Showing**: Divided into two subcategories:
    - **Correctness**: Evaluate the accuracy and reliability of the results.
    - **Completeness**: If the system provides comprehensive output or meets the required functionalities in every scenario.

Benefits of our system:

- **Robust License Plate Recognition**: Capable of detecting license plates accurately under various conditions, including different angles and weather challenges.
- **Edge Computing Deployment**: Efficiently utilizes local resources for AI processing, reducing dependency on cloud services and ensuring real-time performance.
- **User-Friendly Design**: Includes a well-designed website interface that is cost-efficient, intuitive, and easy to monitor.

## 5. CONCLUSION, DISCUSSION AND FUTURE WORKS

The implementation of a smart parking system combined with Artificial Intelligence and the Internet of Things has brought Vietnam many great potential benefits and truly deserves deep attention. In the increasingly rapid urbanization revolution in Vietnam in general and Ho Chi Minh City in particular, the parking challenge is increasingly becoming a headache for experts in finding ways to solve this problem. The development of AIoT has become one of the global development trends as well as the most effective solution to solve the problem of lack of parking across Vietnam. IoT allows parking lots to operate more stably and securely, somehow reducing costs in paying human resources, while Artificial Intelligence plays a role in speeding up the process of detecting - processing the license plate information and also increasing efficiency in maximizing the number of empty spaces in the parking lot, from there increasing user attractiveness, aiming for users day by day become familiar with these smart parking lots. In addition to the benefits above, other projects similar to this project

are currently only being developed at a small and medium level and have not been fully established due to some cost issues related to technology and hardware as well as the installation process has had a great impact on the development of these projects across Vietnam as well as Ho Chi Minh City. To strongly push these similar projects to develop strongly in the coming time, the need to be sponsored from both sides in terms of technology and hardware so that the product can be optimized and definitely will bring more huge advanced impacts to businesses as well as the users so that users and businesses can experience the service in the best and most effective way but in reasonable price.

In conclusion, the AIoT-based smart parking system has proven more efficient than handwritten tickets. Integrating advanced technologies such as real-time license plate recognition, automated entry and exit logging, and cloud-based data management, this system minimizes human error and increases overall progress. This shift towards automation not only increases operational efficiency but also paves the way for smarter, more scalable urban parking solutions.

This smart parking system, with the weaknesses mentioned above, will need many major upgrades in the future, for example, the mobile application will be upgraded in terms of user interface, which can attract more customers, or the function of booking parking spaces before arriving will be upgraded soon helping to minimize the possibility of stopping in the parking lot leading to the congestion. In the end the improvement of detecting and recognizing license plates, based on the various datasets combined with more kinds of conditions will lead to the model with higher precision soon and using different types of models to find out which model serves the best performance among them, the result is that from there can minimize all the common errors that happen to previous systems had been installed.

## ACKNOWLEDGEMENTS

We gratefully acknowledge the AIoT Lab Vietnam for their invaluable support in this research.

## REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of things for smart cities,” IEEE Internet of Things journal, vol. 1, no. 1, pp. 22–32, 2014.

[2] Sheetal S Zalte-Gaikwad, “Edge Computing Technology: An Overview,” Special Issue on Current Research Trends in Management, Science and Technology, vol. 7, no. 1, pp. 96-99, March 2022.

[3] Telefornia I+D and the University of Cantabria, “Smartsantander: IoT experimentation over a smart city testbed,” In: Computer Network, 2013, pp. 9–10.

[4] Pham Thanh Nam, Pham Xuan Kien, “Building a model of smart parking system in vietnam based on internet of things and cloud server technolies,” In: TNU Journal of Science and Technology, 2021, pp. 201–206.

[5] Prabhu. Ramaswamy, “IoT smart parking system for reducing green house gas emission,” In: International Conference on Recent Trends in Information Technology (ICRTIT) (IEEE), 2016, pp. 1–6.

[6] V. P. V. K. Rishabh, Pushpendra Kumar, “A novel car license plate and parking slot detection approach based on yolo,”

[7] L. A. R. G. Alessandro Florisa, Simone Porcua, “A social iot-basedplatform for the deployment of a smart parking solution,” 2021.

[8] Awad Alharbi, George Halikias, Mohammad Yamin, Adnan Ahmed Abi Sen, “Web-based framework for smart parking system”, 2021.

[9] Phat Nguyen Huu, Loc Hoang Bao, “Proposing Real-time Parking System for Smart Cities using Two Cameras”, June 2021.

[10] Dia, I.; Ahvar, E.; Lee, G.M. “Performance Evaluation of Machine Learning and Neural Network-Based Algorithms for Predicting Segment Availability in AIoT-Based Smart Parking”. Network 2022, 2, 225–238. https://doi.org/10.3390/ network2020015.

[11] Mai Chi Bao, https://github.com/mrzaizai2k/VIETNAMESE LICENSE PLATE

[12] Vinh Du Mai, Duoqian Miao and Ruizhi Wang ,”Vietnam License Plate Recognition System based on Edge Detection and Neural Networks”, 2013.

[13] Siti Norul Huda Sheikh Abdullah, Marzuki Khalid and Rubiyah Yusof, ”License Plate Recognition using Multi-cluster and Multilayer Neural Networks”, pp.1818-1823, 2006.

[14] Zhenxue Chen, Guoyou Wang, Jianguo Liu and Chenyun Liu, ”Automatic License Plate Location and Recognition Based on Feature Salience”, International Journal of Cumputational Cognition 5, pp. 1- 9, 2007.

[15] Sergey Zherzdev, Alexey Gruzdev, ”LPRNet: License Plate Recognition via Deep Neural Networks”, pp. 1-6, 2018, https://doi.org/10.48550/arXiv.1806.10447.

[16] Tran Duc Duan, Tran Le Hong D, Tran Vinh Phuoc, Nguyen Viet Hoang ,”Building an Automatic Vehicle License-Plate Recognition System”, 2005.

[17] Quang Huy Che, Tung Do Thanh, Cuong Truong Van ,”Character Time-series Matching For Robust License Plate Recognition”, 2023, https://doi.org/10.1109/MAPR56351.2022.9924897.

[18] ALI M. REZA, ” Realization of the Contrast Limited Adaptive Histogram Equalization(CLAHE) for Real-Time Image Enhancement”, 2004, Journal of VLSI Signal Processing 38, 35–44.