International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

Smart Parking using IoT, Computer Vision and Android Application

Arem Kiran Kumar Reddy[1], Jithesh C J[2], Merline Biju[3], Sneha S Ganer[4], Dr. Josephine Prem Kumar[5]

[1, 2, 3, 4]*UG student, Department of Computer Science and Engineering, Cambridge Institute of Technology, Bangalore, Karnataka, India*

[5]*Professor, Department of Computer Science and Engineering, Cambridge Institute of Technology, Bangalore, Karnataka, India*

ABSTRACT

We all know the difficulty of finding a parking space in the city, and the trouble that incorrectly parked vehicles cause. The search for a parking space adds to the traffic congestion and excess consumption of fuel. In this project, we combine the use of IoT, Computer Vision and mobile application to save time and reduce the stress of finding a spot to park one's vehicle safely and without causing inconvenience to anyone. The system displays the real-time status of the parking lot to users, and allows them to book a spot in advance so that waiting to get a parking space can be avoided. We have used Raspberry Pi as a microcontroller. Pi camera is used to scan the license place of the vehicle in order to generate an e-ticket, which the driver can produce when they want to retrieve their vehicle. The number of slots available is indicated by keeping track of the arrival and departure of the vehicles. This system achieves security of vehicles, and ease of parking.

Key Words: Internet of Things (IoT), Computer Vision, Android Application, Smart Parking, Security.

1. INTRODUCTION

The Internet of Things (IoT) is a fast-developing technology in recent years. IoT is the concept of connecting devices such as cars, mobile phones and other household devices, and enabling communication between them. The IoT devices are equipped with network connectivity and electronic components like sensors and actuators. With these components, IoT devices can send data or even be controlled remotely by utilizing the internet network infrastructure. The goal of a Smart City is to create services to facilitate access to public facilities and improve their quality. This aim can be realized by spreading the use of IoT technology on the infrastructure of public facilities in urban areas.

One example of an important public facility in urban areas is public parking for vehicles. Finding a parking slot to park their vehicle is a troublesome issue to the drivers all the time. It has led to traffic congestion which is a huge problem on a global scale. Also, it has been found that it has led to the excessive burning of the world's oil reserves. According to a report, Smart Parking system could benefit in saving 2,20,000 gallons of fuel by 2030 and 3,00,000 gallons of fuel by 2050, if it is executed perfectly.

Here, we propose an idea to realize a Smart Parking structure using IoT, Computer Vision and Android application. The concept of a smart parking lot includes sensors, microcontrollers, and LCDs. The Android application allows the user to interact with the system through actions such as booking a slot and making payment at the exit. Computer Vision is used to verify the scanned license plate with the registered user. The sensor reads the number of vacant parking spots and sends the data to the Raspberry Pi. Then the Raspberry Pi directs the number of slots to be displayed on the LCD. This improves the accessibility of parking lots for the community. With this system being implemented, users need not search for the availability of parking space in person.

1.1 Related Works

Several solutions have been proposed in previous works given in [1], [2], [3], [4], and [5] for improving the parking process. The summary of each solution is discussed as follows.

In paper [1], the system allows the users to book the slot via SMS. The SMS sent is processed by micro-RTU (Remote Terminal Unit). The confirmation of booking is done by micro-RTU, which then sends the details of reservation like password and lot number to the user. The password is valid for a certain period of time and is used to enter the parking area. The cost of implementation is high and the microcontroller has to take a lot of load which can crash the system. We have overcome this issue and reduced the load on the microcontroller by avoiding the usage of micro-RTU and SMS system.

In paper [2], the design and implementation of the system is based on Wireless Sensor Networks (WSNs). It consists of wireless sensor networks, embedded web-server, central web-server and mobile phone application. In the system, low-cost WSN modules

are deployed into each parking slot equipped with one sensor node. The state of the parking slot is detected by the sensor node, which periodically reports to the embedded web-server through the deployed WSNs. Wi-Fi networks are used to send this information to the central web-server in real-time. The driver can find vacant parking lots using the android application on their mobile phone. This system does not provide a reservation feature and multilevel parking inside an infrastructure.

Paper [3] aims to present an intelligent system for parking based on image processing. The proposed system processes the image captured at the parking lot and provides the status of slots. Cameras are used as sensors to take photos to show the vacancy of slots. The availability of slots is used to guide a driver to the vacant slot rather than wasting time to find one. The weather conditions affect the system in terms of visibility. The camera should be in a position where it can see all the slots and not be obstructed by other objects. Navigation to the parking lot is not provided in this system.

Paper [4] integrates networked sensor/actuator and radio frequency identification (RFID) technologies through the Internet of things (IoT). Based on this integration, a scalable car parking framework (CPF) is proposed. The prototype explores clustering of sensors (sensing boards) into a single mote using the I2C protocol. This mote stores the events in the local database. Each car has to be equipped with an RFID tag which increases the implementation cost. No guidance is provided to the parking lot here.

Paper [5] uses a mobile application, computer vision and IoT technologies to implement the smart parking system. This system guarantees the security of the parked vehicle using automatic security bollards. It uses NodeMCU as a microcontroller and ultrasonic sensors as proximity sensors. CCTV cameras are also used for verifying readings from the IoT devices. The system displays the live status of the parking spaces in the parking lots to the users of the mobile application. Th cost of implementation is high due to the usage of both IoT devices and cameras to check the vacancy of slots.

1.2 Proposed System

The smart parking system that we propose is implemented using IoT devices, Computer Vision and a mobile application that is connected to the cloud. It consists of three sections: the first section is the booking of a slot on the app by a user. The app is used by the user to check the availability of slots in the desired parking area. The second section contains the cloud-based database which acts as a mediator between the app and parking area. It is updated in real-time according to the change in status of slots. The third section is the parking area which includes IoT devices like Raspberry Pi, IR sensors and Pi camera. Scanning and verification of the license plate at the entrance of the parking lot is done using Computer Vision. The user pays the parking fare through the app before exiting the parking area.

The use of efficient sensors and quick data processing units makes management of the parking area easy and saves the time of users by providing real-time status of slots. The cost of implementing such a system is significantly lower than that of the works mentioned in the previous section. This system is comparatively more efficient as almost all the processes are automated/online.

2. METHODOLOGY

2.1 Internet of Things (IoT)

The internet of things is a system of interrelated computing devices, mechanical and digital machines, and objects that have the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. An IoT system consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware which collect, send and act on data they acquire from their environments. The sensor data collected by IoT devices is either sent to the cloud to be analyzed or analyzed locally by connecting to an IoT gateway. These devices can communicate with other related devices and act on the information they get from one another. Most of the work can be done by these devices without human intervention, though human interaction may be required to set them up, give them instructions or access the data. Our system uses IoT devices like Raspberry Pi, IR sensors and Pi camera.

2.2 Computer Vision

Computer vision deals with extracting information from the input images or videos to have a proper understanding of them to predict the visual input like that of a human brain. Computer vision is focused on the core technology of automated image analysis which has applications in many fields. Machine vision is the process of combining automated image analysis with other methods and technologies to provide automated inspection in industrial applications. Techniques like character segmentation and character recognition are used in our system.

2.3 Android Application

Android is an operating system and programming platform for mobile devices, such as phones and tablets. Android includes a software development kit (SDK) that helps developers write original code to create apps for Android users. Android App is a software that runs on an Android device or emulator. Android apps are written in Java and use Java core libraries. Android Studio is an integrated

development environment (IDE) which allows developers to set up the environment, write code, build, run, debug, test and publish the app. Android Studio is used to develop our app "ParkSmart", which can be installed and used on any Android device.

2.4 Cloud database

A cloud database is a database that runs on a cloud computing platform. Access to the database is provided as-a-service. Users can either run databases on the cloud independently, using a virtual machine image, or purchase access to a database service which is maintained by a cloud database provider. Some of the databases available on the cloud are SQL-based and some use a NoSQL data model. Cloud Firestore is a cloud-hosted, NoSQL database that iOS, Android, and web apps can access directly through native SDKs. It stores data in a structure that looks like a tree, but the data is in fact stored as documents. Collections and documents are primary building blocks of the Cloud Firestore. subcollections can be created within documents and hierarchical data structures can be built that scale as the database grows.

3. MODELLING AND ANALYSIS

3.1 Hardware and Software Requirements

Table-1: Specification of the hardware components used.

Hardware	Specification
Raspberry Pi 3b+	64-bit quad-core processor running at 1.4 GHz, dual-band 2.4 GHz and 5 GHz wireless LAN, Bluetooth 4.2/BLE, 1GB memory.
Pi Camera	25mm x 20mm x 9mm, resolution: 5MP, 2592 x 1944 pixel static images, 1080p30 video.
IR proximity sensors	2cm x 3cm x 4cm, effective distance range: 2 ~ 10cm, working voltage: 3.3V-5V.
LCD display	16cm x 2cm x 2cm, 16x2 Alphanumeric Display (JHD162A), input: 5V.
2-channel Relay module	140 mm x 55 mm x 19.3 mm, input: 5V, output: AC250V 10A and DC30V 10A.
DC motor	Shaft diameter: 6mm with internal hole, RPM: 150, input: 12V DC.
Battery	9x HW Battery 6F22 9V, Carbon Zinc Battery.

Table-2: Specification of the software used.

Software	Version
Raspberry pi – Thonny	3.3.6
Android Studio	4.2.1
Firebase	9.10.2

3.2 System Design

The steps involved in using the smart parking system, with the help of the application, is shown in figure 1 and explained below:-

Step 1: Install the Android app on the mobile device and register using the user's email-id, driver's license number and vehicle registration number.

Step 2: Check for availability of parking slots in the desired parking lot and book a slot.

Step 3: Upon receiving the confirmation of booking email, navigate to the parking lot within the time specified.

Step 4: At the entrance of the parking lot, the license plate is scanned for verification of booking and the user is allowed to enter and park the vehicle.

Step 5: The user can then tend to his business without worrying about the safety of his vehicle.

Step 6: Once the user is ready to leave, the license plate is scanned again at the exit, and the user is prompted to pay the parking fare through the app.

Step 7: After successful payment of the parking charges, the user can leave the parking lot.

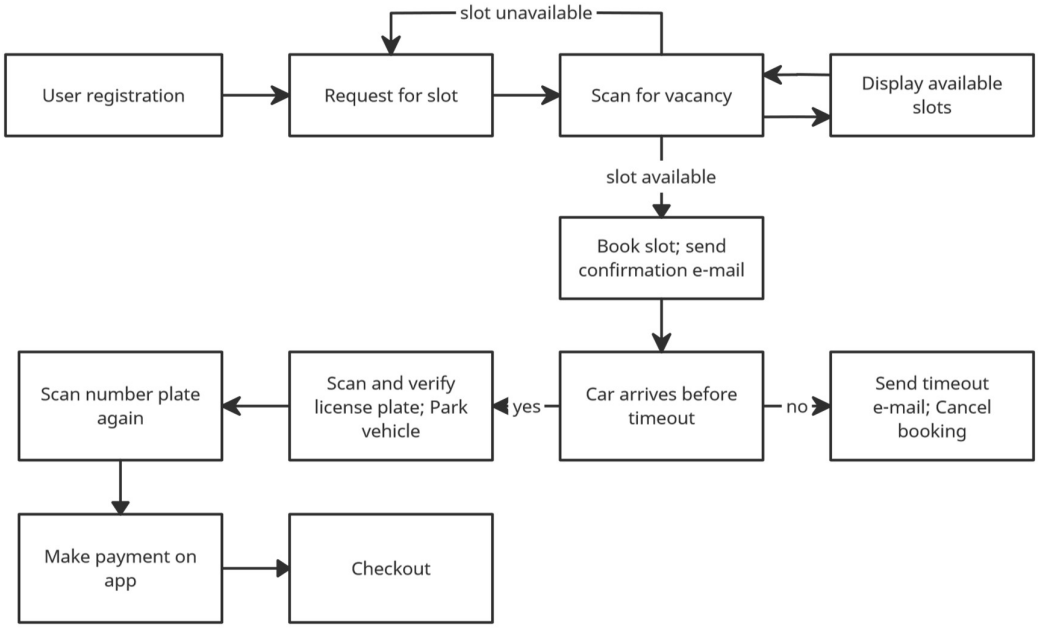


Figure-1: System Design

3.3 Implementation

The user registers on the app and searches for the desired parking lot. He checks for the availability of slots, which is obtained from the database, and then books a slot. The app is connected to the database and any action done in the app is updated in the database. After successful booking, a mail is sent to the user, along with a time limit within which he has to reach the parking lot.

On reaching the parking lot, the vehicle's number plate is scanned and recognized by the Pi camera. The camera captures the image which is then positioned based on the aspect ratio. The positioned license plate image is pre-processed by graying it because sometimes the color of the vehicle and the license plate might be the same. Processes like character segmentation and character recognition are followed to give the final detected license plate number. The license plate number detection process is done using the OpenCV library.

The detected number is sent to the Raspberry Pi to verify the booking. The license plate number of the user who booked a slot is fetched from the database and is matched with the detected number. If the match is successful, the Raspberry pi directs the DC motor, through the relays (as shown in Figure 2), to open the barricade. The user can then enter the parking lot and choose the nearest available slot by looking at the vacant slot numbers displayed on the LCD. Once the vehicle is successfully parked in the slot, the IR sensor in the slot senses the vehicle and informs the Raspberry pi about the occupied slot. This information is used to update the LCD to display the vacant slots. The database is also updated with the occupancy and the parking timer starts.

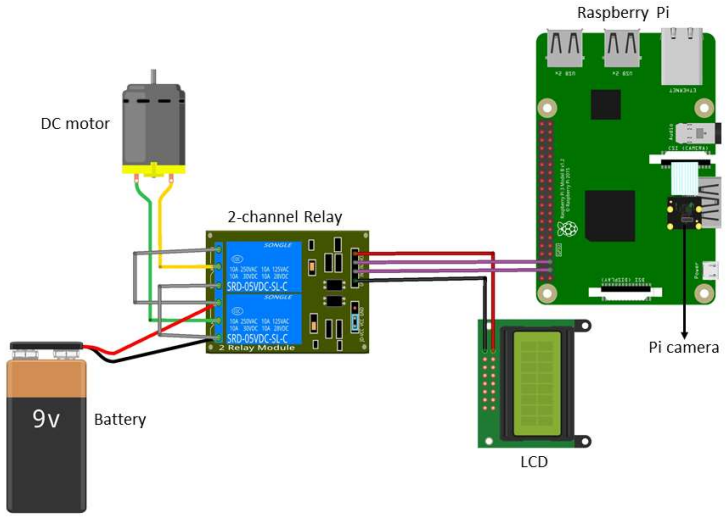


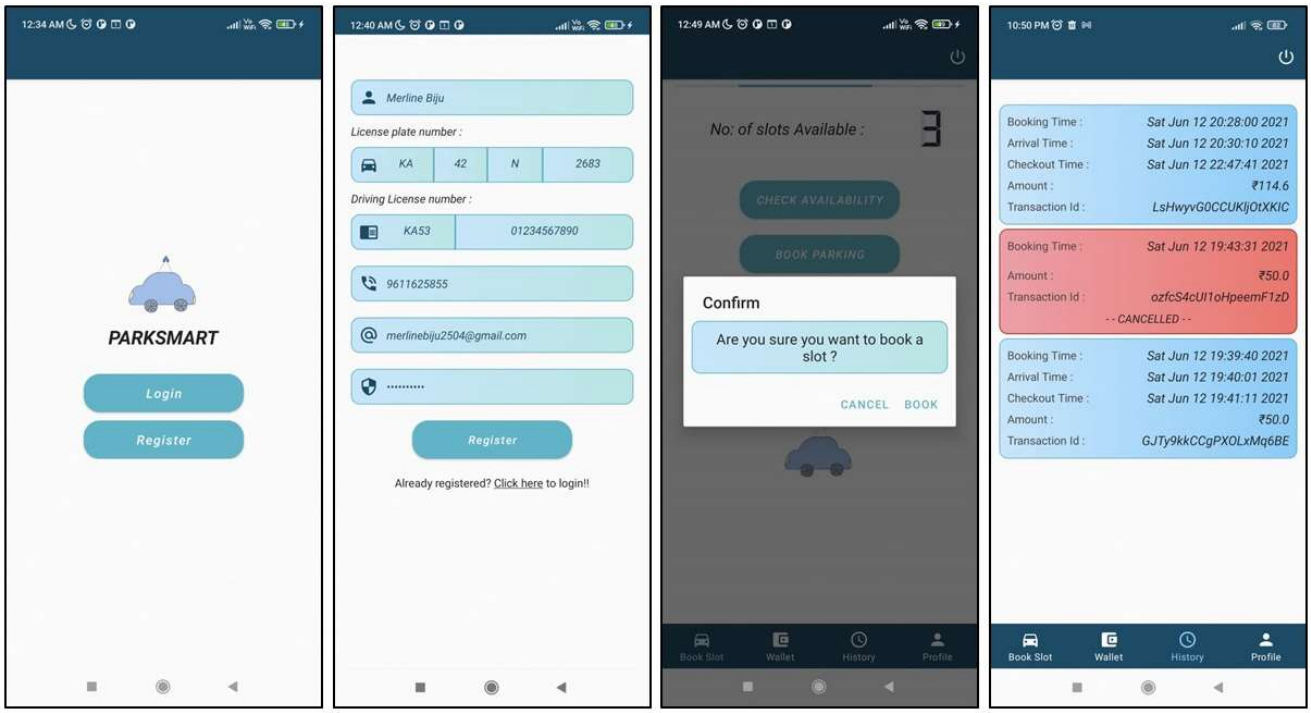
Figure-2: Circuit diagram

When the driver returns to the vehicle and takes it to the exit, the license plate is scanned again and the database is updated about the vacancy of the slot, and the parking timer stops. The parking fare is generated and the user is prompted to confirm payment of the parking fare. This is done in order to avoid vehicle theft and other security issues. After successful payment through the app, the user is then allowed to move out of the parking lot.

4. RESULTS AND DISCUSSION

4.1 Module 1: Android App

The user registers on the app with his details, as shown in figure 3 (b). The app then authenticates the user by using his email-id and sending the verification link. Once this verification is completed the user can book a slot based on the availability (figure 3(c)). The user can navigate to the parking lot using the maps embedded in the app. The booking is verified on reaching the parking lot. At the exit, the user is prompted on the app to make the payment of the parking fare. The user confirms payment and leaves the lot. The parking history is updated in the app upon the user's checkout from the lot, as shown in figure 3(d).



(a) (b) (c) (d)

Figure-3: App module

4.2 Module 2: Parking Lot

The parking lot is equipped with a Pi camera for scanning the license plate, Raspberry Pi as the microcontroller, DC motor and Relay to open the barricade, IR sensors to sense the occupancy of the slot and LCD to display the vacant slots, as shown in figure 4. On the vehicle's arrival, the camera scans and recognizes the number and the detected number is used for booking verification as shown in figure 5. After the verification, the user can park in the desired slot. The IR sensor then senses the vehicle and updates the Raspberry Pi. The LCD then displays the updated slots available as shown in figure 6(b). When the vehicle leaves the slot, the IR sensor again updates the Raspberry Pi, which prompts the LCD to display the updated vacant slots. At the exit of the parking lot, the license plate is scanned again and the parking fare is generated. The user confirms the payment and the barricade opens to allow the user to exit from the parking lot.

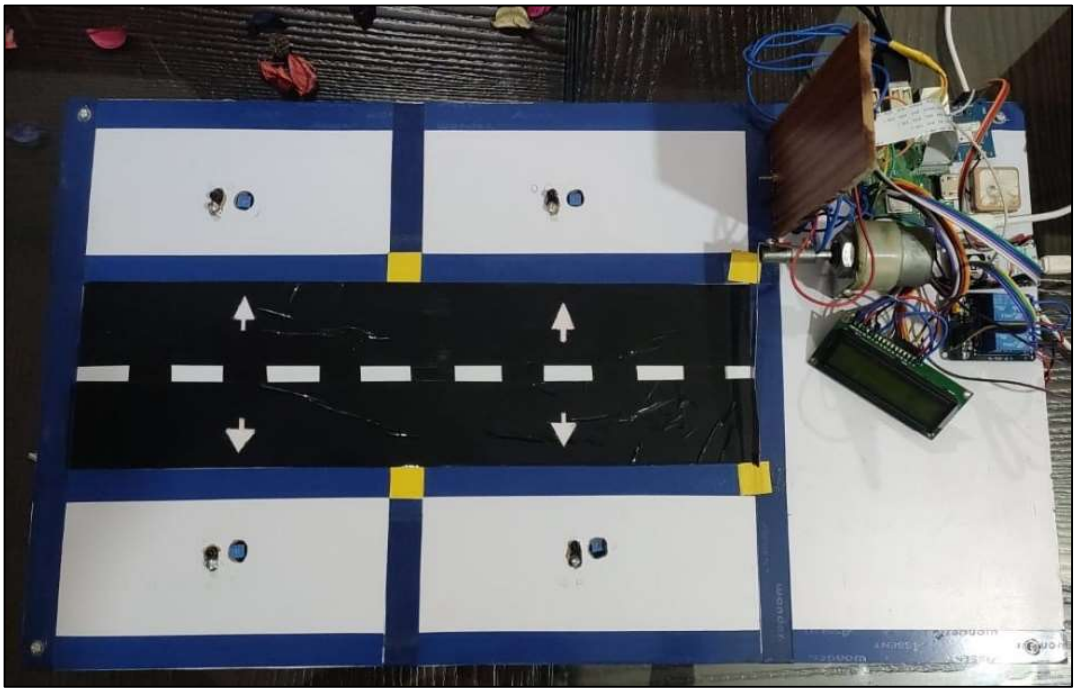
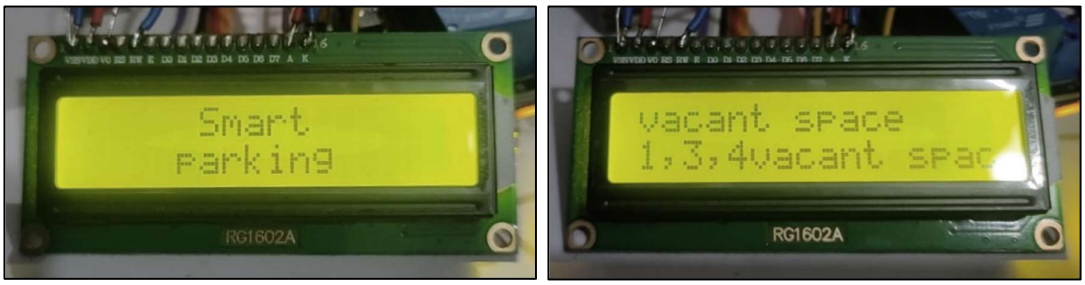


Figure-4: Parking lot module

Figure-5: Number plate recognition



(a) (b)

Figure-6: LCD showing vacant slots

4.3 Discussion

The proposed system is highly scalable in the sense that it can be deployed in various places like malls, multiplexes, party halls, bus stands, railway stations, airports, parks and other places of tourist interest. This system has the advantage of being implemented for multilevel parking as well. Since CCTVs are already installed in most parking lots, they can be incorporated into the system for monitoring purposes. For large-scale implementation, other reliable sensors such as ultrasonic sensors can also be used. This system is a remarkable step in the realization of the concept of Smart cities.

5. CONCLUSION

The parking problem is quite acute in developing countries like India. The smart parking system helps both the users and administrators. It helps the users in finding out the availability of a parking slot and ensuring that their vehicle is safely parked. It helps the administration to allocate the vacant slot in a methodical and organized manner as the system is automated. It not only saves users time but also reduces the stress of having to physically search for slots. Users find easier to park their vehicle with the booking system, paperless ticket and cashless payments. The booking system enables the users to find an available parking slot, even before arriving. The paperless ticket system eliminates the possibility of missing the ticket and it is also ecofriendly. Cashless payment will ease the payment process. The vacant slot number guides the user to a vacant parking slot without the need to go around. The proposed system has been tested under different scenarios and the tests have provided positive results, thus concluding that the proposed method is a more advanced alternative for the currently deployed system.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our friends Abhishek Saseendran and Kiran R. Naik for their valuable contributions to our project. We would also like to thank our family members for their support and encouragement throughout the completion of the project.

REFERENCES

[1] Noor Hazrin Hany Mohamad Hanif, Mohd Hafiz Badiozaman, Hanita Daud, “Smart parking reservation system using short message services (SMS)”, 2010, International Conference on Intelligent and Advanced Systems, pp. 1-5, doi: 10.1109/ICIAS.2010.5716179.

[2] Jihoon Yang, Jorge Portilla, Teresa Riesgo, “Smart Parking Service based on Wireless Sensor Networks”, IECON 2012, 38th Annual Conference on IEEE Industrial Electronics Society, pp. 6029-6034, doi: 10.1109/IECON.2012.6389096.

[3] Hilal Al-Kharusi, Ibrahim Al-Bahadly, “Intelligent Parking Management System Based on Image Processing”, 2014, World Journal of Engineering and Technology, Vol. 2, Issue 2, Pg. 55-67.

[4] ElMouatezbillah Karbab, Djamel Djenouri, Sahar Boulkaboul, Antoine Bagula, “Car Park Management with Networked Wireless Sensors and Active RFID”, 2015, IEEE International Conference on Electro/Information Technology (EIT), pp. 373-378, doi: 10.1109/EIT.2015.7293372.

[5] Madhur Dixit, Srimathi C, Robin Doss, Seng Loke, M. A. Saleemdurai, “Smart Parking with Computer Vision and IoT Technology”, 2020, 43rd International Conference on Telecommunications and Signal Processing (TSP), pp. 170-174, doi: 10.1109/TSP49548.2020.9163467.

[6] Faiz Shaikh1, Nikhilkumar B S, Omkar Kulkarni, Pratik Jadhan, Saideep Bandarkar, “A survey on Smart Parking System”, 2015, International Journal of Innovative Research in Science, Engineering and Technology, Vol. 4, Issue 10, pp.9933-9939, doi:10.15680/IJIRSET.2015.0410086.