International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Parkade: Smart Parking System by using Image Processing

Pranshu Srivastava, Ms. Charu Singh, Lokesh Babu Chaurasia, Lalit Pratap Singh, Rahul Singh

Department Of Computer Science and Engineering

Babu Banarasi Das Institute of Technology and Management, Lucknow

ABSTRACT:

As we know, car parking is a major problem in urban areas. when we go to the huge parking lot, we face a lot of problems like where to park their vehicle, security issues, damage(dent) by another vehicle and many more. Our aim to develop an application to provide the solution of parking like ease of finding parking space, security (surveillance system are necessary to ensure the security of car parked from car theft). We will be using OpenCV library for implementing our project. OpenCV support a wide variety of programming language like tesseract (open-source text recognition engine to extract printed text from images). It can process images and videos to identify object, faces. In this project we will be using the live feed coming from the webcam to create a digital parking system with functionalities. We will detect number plate of a vehicle after detecting the number plate we will send a message to the owner mobile about the vacant parking spot where you have to park the vehicle and also, we will keep track of the vehicle like at which time vehicle has entered and leave the parking lot.

Key Words: OpenCV, Tesseract, Image Processing, Optical Character Recognition,

firebase

1.INTRODUCTION:

Finding a vacant parking space is a common problem in most urban cities which especially occurs in popular and well-travelled places like shopping complexes, stadiums and other well-travelled areas or tourist attraction spots. This situation has become more serious especially during their peak time, be it holiday seasons, sales carnivals or any other festivals. This problem arises as most of the time, as patrons come by their own transports, resulting in abundance or high number or transports competing for a few vacant parking spaces.

How many times has it happened to you that you are searching for a parking spot by driving around and around the parking lot? How convenient would it be if your phone could tell you exactly where the closest parking spot is! With the high percentage of vehicle ownership in the India, parking has become a conflicting and confusing situation for a number of people. Whether at an airport, bus stations and shopping centres, problems with parking are an everyday occurrence. Due to the importance of parking, cities study and analyse parking programs and performance on an ongoing basis. The following list identifies the kinds of problems that typically occur in a community. Inadequate information for motorists on parking availability and price. Motorists are likely to be frustrated if they expected abundant and free parking but find limited or expensive parking, or if they must spend excessive time searching for a parking space Inefficient use of existing parking capacity. Local zoning ordinances, building codes, and other development practices can result in an oversupply of parking spaces and an inefficient use of existing parking.

1.1 OpenCV:

OpenCV (Open-Source Computer vision) is permitted for both scholastic and commercial use. It is a library of programming functions mainly aimed at real-time computer vision. OpenCV's application has wide areas which includes 2D and 3D feature toolkits, Ego motion estimation, Facial recognition system, Gesture recognition, Motion understanding, Object identification Segmentation and recognition and Motion tracking. OpenCV contains libraries of pre-defined functions supportive in image processing. Since it is open source, it was chosen as the platform to test the project. Using OpenCV libraries we have implemented image processing mechanisms like RGB to grayscale conversion.

1.2 python:

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy highlights code readability and its syntax allows programmers to express concepts in less lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

1.3 Tesseract:

Tesseract package contains an OCR engine - libtesseract and a command line program - tesseract. The lead developer is Ray Smith. Tesseract has Unicode (UTF-8) support and can recognize more than 100 languages "out of the box". It can be trained to recognize other languages. Tesseract supports various output formats: plain-text, hocr(html), pdf. The Tesseract engine was originally developed as proprietary software at Hewlett Packard labs in Bristol, England and Greeley, Colorado between 1985 and 1994, with some more changes made in 1996 to port to Windows, and was some migration from C to C++ in 1998. A lot of the code was written in C, and then some more written in C++. Since then, all the code has been converted to at least compile with a C++ compiler. Tesseract is available for Linux, Windows and Mac OS X, however, due to limited resources only Windows and Ubuntu are rigorously tested by developers.

1.4 Firebase:

The Firebase Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, real-time events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

2. SYSTEM OVERVIEW:

A car enters the parking lot and the parking lot is checked for empty slots. If there exists any then the number plate of the car is scanned using the camera and stored in database and let

into the parking lot by displaying the slots available. If there are no empty slots available, then the same will be displayed. When the car exits then the number plate is scanned again using the camera and compared with the time of its arrival in Fig-2.1.

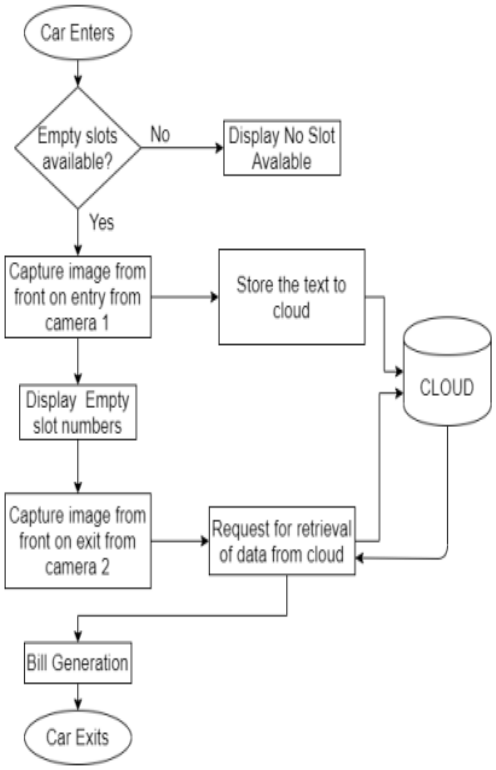

Fig -2.1: Flowchart of Smart Parking System

3. IMPLEMENTATION:

3.1 Tesseract:

When the car enters parking lot, we capture the image of the plate. We perform image processing methods on the captured image. We then find the contours of the plate and crop the image along those contours. The processed image is then fed to Tesseract algorithm for character recognition. The extracted image is then checked whether it exists in cloud or not. If the plate already exists in the cloud, we extract the information from it and produce bill based on time of stay. If the plate does not exist in cloud, we upload it with time of entry to the firebase.

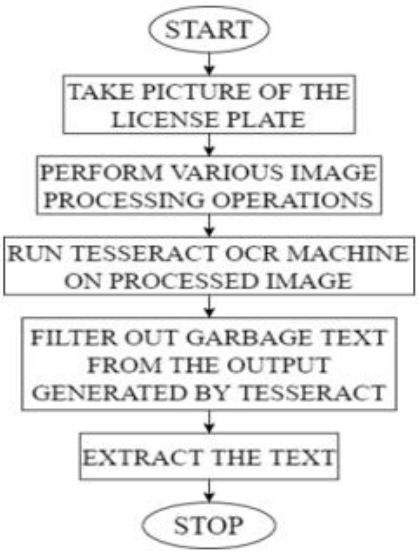

Fig -3.1: Flowchart of Plate Recognition

3.2 OpenCV:

The still image file is a frame of the video file and after the code is executed, the still image the quadrilaterals created to zone each parking slot are outlined either red or green. Red means that the computer recognizes the parking space within the quadrilateral as occupied by a car and green means it is available for parking.

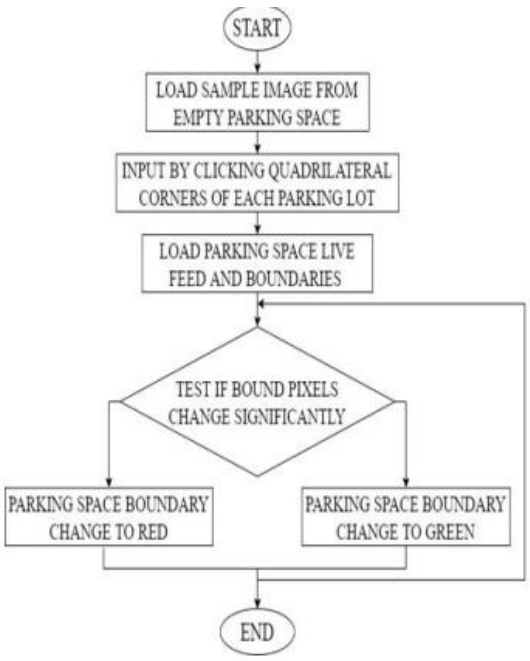

Fig -3.2: Flowchart of Empty slot Detection

Running the program, the Empty slot detection system has been exacted in demonstrating whether a parking spot is occupied or not. Figures 3.2(a,b,c) and 3.2(d) show the condition of the parking structure and the building, masked with coloured rectangles. Rectangle shape zones were set as input before the program was run. In the event that a vehicle is inside the given Rectangle shape zone, the pixel colour inside will change, and the average colour would likewise vary. This would incite the program to change the colour from green to red.

Fig -3.2(a): Parking garage where occupied space was marked red and unoccupied space was marked green.

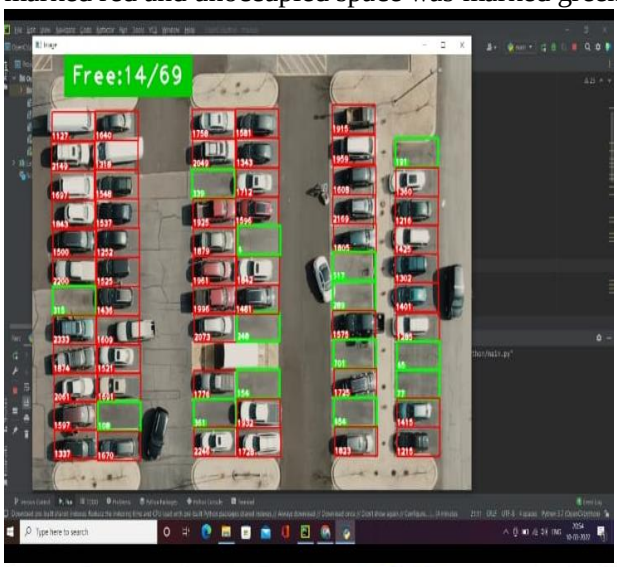

Fig -3.2(b): unoccupied space was marked green

Fig -3.2(c): occupied space marked red

At the point when the vehicles occupy a specific parking slot, the layout diverts quickly from green to red since the average colour has changed. For the number plate detection, the image of the number plate is captured by pressing the button and OCR is processed to get the characters. The recognized number plate character is stored in the form of text in the Firebase along with the time. This will be used again when the car leaves the parking. Fig.6(a) shows the image of the car which was taken at entrance

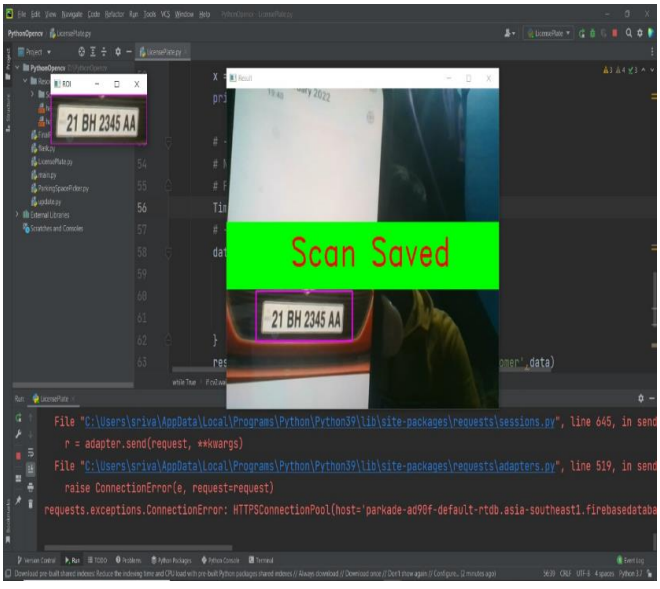

Fig -3.2(d) Number plate image captured at entrance.

3.3 Admin website:

Website that connects with parking rules and regulations in that page we provide so many information like how many cars is there in parking lot, how many cars exit from the parking lot along with time and date and also provide the previous reports of parking status.

Fig.3.3(a), Fig.3.3(b) shows the output of the characters detected from image of car taken at entrance and exit. It also shows unoccupied space again along with colour green.

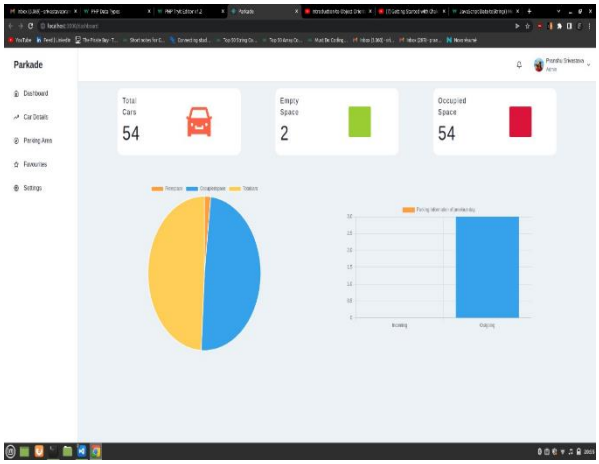

Fig-3.3(a): Number plate image captured at entrance.

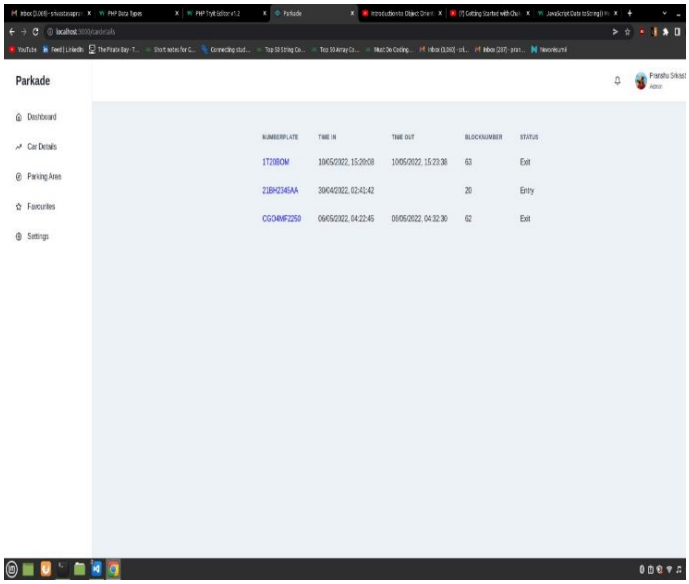

Fig -3.3(b): number plate detection at entrance along with exit.

4. CONCLUSIONS:

Image Processing is very crucial for extracting any information from an image. In this study, a proposed plan for a smart parking system based on image processing has been effectively tried and run with a few videos taken from indoor parking garages. The system works precisely in deciding regardless of whether the parking slots are occupied or not by showing a red outline if a vehicle is inside or consuming a parking spot and afterward turns green when it is unoccupied. In the number plate detection, we first applied image processing algorithms to images and afterward those images were utilized in Tesseract software to acquire the text from the images. Different images have distinctive text styles, length, width and font, so different images require different levels of digital image processing techniques. Out of these image results for a single image the most appropriate image is then applied to Tesseract for obtaining the text from an image. Therefore, after digitally processing the image, we have accomplished better and near to perfection outputs.

5.Reference:

[1]. Yong sheng, C.; Demin, L.; Guanylin, Z. The Guidance System of UAV Parking Space Based on INTERNET of Things. Available online: https://www.zhangqiaokeyan.com/patent-detail/06120109146805.html (accessed on 12 December 2020).

[2]. Xin Peng, C.; Xingang, H.; Wei, X. Intelligent Parking Lot Parking Space Guidance Management System. CN. Available online: https://www.zhangqiaokeyan.com/patent-detail/06120365383.html (accessed on 12 November 2020).

[3] K. Cheng, “Indoor Parking DAHUA,” 2018. [Online]. Available:

https://www.youtube.com [Accessed 2018].

[4] KETOSI, “[KETOSI] Audi R8 V10 Parking Lot Accident CCTV,” 2013. [Online]. Available:

https://www.youtube.com [Accessed 2018].