PARK-A-LOT

Manoj B Chandak,

Abstract: ***With the increasing difficulty in finding parking spaces and the increasing need for security, an automated system to guide vehicles to parking will come as rescue. With this project of ours we aim to develop an approach that will allow automated entry to the employee vehicles that are registered with the office. This system will automatically detect if a vehicle has been registered and accordingly open the gates of the institute for the vehicle along with which it will also display and guide vehicles to vacant parking spaces and alternate parking areas in case of occupied parking lots.***

Keywords : Monitored Parking, Professional Parking, license Plate, Number Plate, CCA.

I. INTRODUCTION

With this project we are aiming to develop an approach and design a framework to tackle the problem of manual monitoring of the institute entrances and parking lots by automatically tracking parking occupancies and providing alternate parking spaces along with providing automated security for vehicle entrance in institution or company campuses.

We have divided our project into two major modules

Professional Places

For this module, we will be detecting the license plates of the vehicles and extracting their vehicle numbers using machine learning techniques which then will be verified if they exist in the database of the registered vehicle numbers of the employees. Once the number has been identified, the vehicle will be allowed to enter the campus. If an employee comes, with a new vehicle that has not been registered previously then the employee will have to scan their QR code on their ID card which will enter the vehicle number in front of their name in the database, keeping a log for future references [1,10]

Monitored parking spaces

The entering and exiting vehicles will be monitored using sensors, according to which the number of available parking spaces will be determined. Once the parking lot is filled the rest of the vehicles will be diverted to an alternate parking areas. The directions to which will either be displayed on screen or could be accessed using mobile applications [7].

II. PARKING FOR PROFESSIONAL PLACES

This module with respect to the professional organizations where only employees working in the organisation are allowed to enter into the premises and park their vehicle in the vicinity. To implement the above module every employee has to register his vehicle with the organisation as well as he will be provided the provision of bringing alternate vehicle if he as Identity card. So for every employee two columns will be maintained one is primary vehicle and other for alternate vehicle [3].

III. METHODOLOGY

The following diagram explains how the system will work for parking available at professional places.

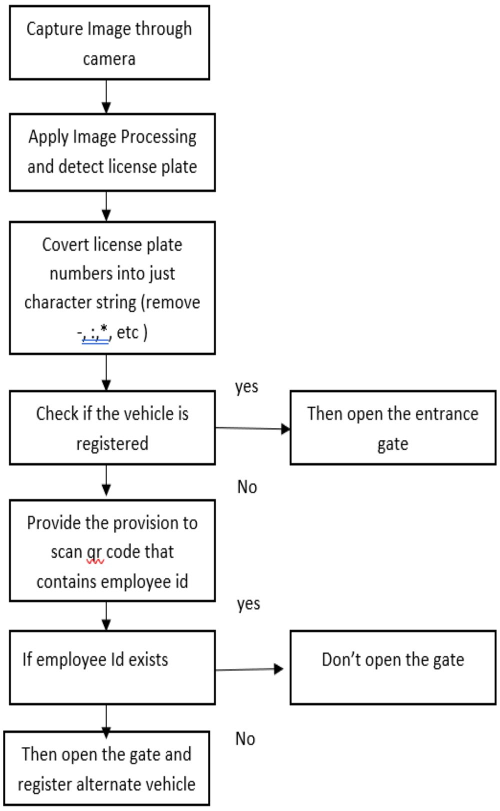

Capture Image through camera
Apply Image Processing and detect license plate
Covert license plate numbers into just character string (remove -,:,*, etc)
Check if the vehicle is registered
yes
Then open the entrance gate
No
Provide the provision to scan qr code that contains employee id
yes
If employee Id exists
Don't open the gate
No
Then open the gate and register alternate vehicle

Fig.1 Flow diagram for professional parking

The above diagram explains the working of the module for professional places. When the employee comes to the parking. The image of license plate will be captured through live camera. Then image processing will be done and the characters of the number plate will be extracted. Then the string will be matched in database. If such vehicle exists then gate will open and the employee will be allowed to park. If the vehicle entry is not there then employee will be asked to scan his Id which is nothing but a QR code which contains id number. If the scanned id exists then alternate vehicle of employee will be registered and he will be allowed to park else employee won't be allowed to enter into premises [2].

Manuscript published on 30 August 2019.
***Correspondence Author(s)**
Dr. Manoj B. Chandak, Computer Science and Engineering Department, Shri Ramdeobaba College of Engineering and Management, Nagpur, India,

IV. STEPS FOR LICENSE PLATE DETECTION

Step1: Take the original image and convert into gray scale using open cv library.

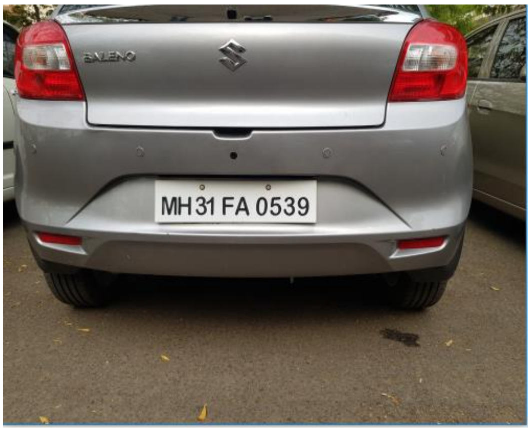


Fig.2 Original Image

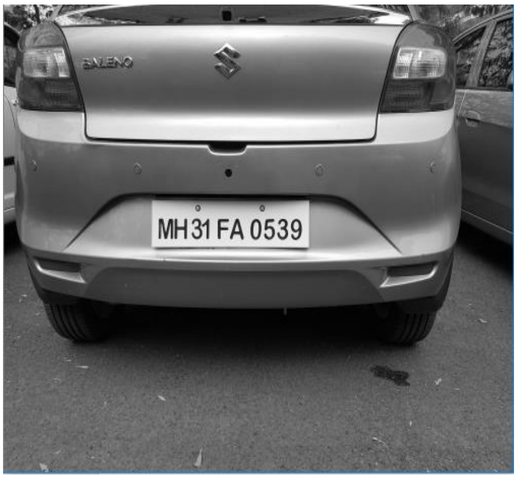


Fig.3 Gray Scale Image

Step2: Apply bilateral filter to the gray scale image. Bilateral filter is used to preserve images when various operations are done on image

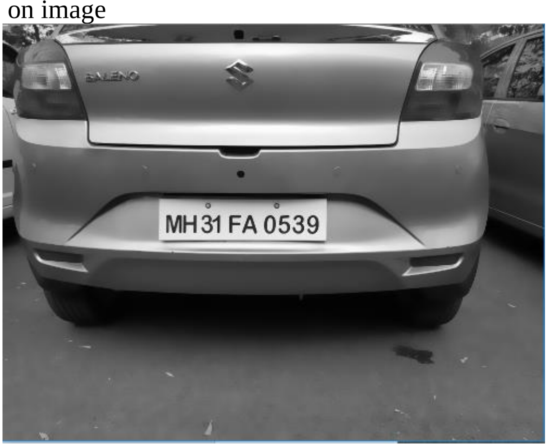


Fig.4 Image after applying Bilateral filter

Step3: Apply Canny Edge detection method to detect edges in images

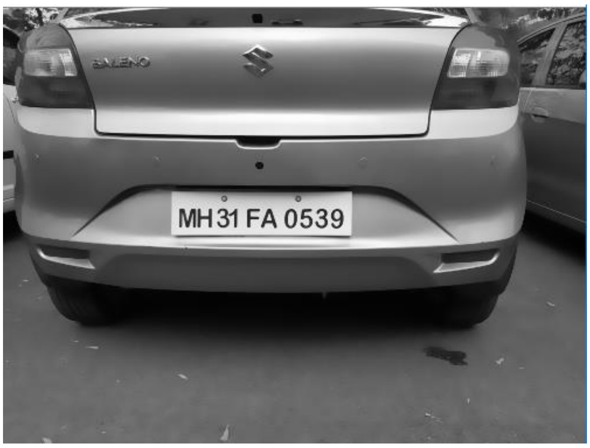


Fig.5 Image after applying Canny Edge Detection Method

Step4: Then use contours method to find all continuous points having same intensity. The contours are useful for object detection, shape analysis and recognition [6].

Step5: Mask the other part other than number plate and show only the license plate part. Convert all other pixels to black.

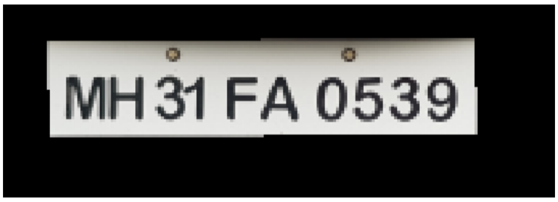
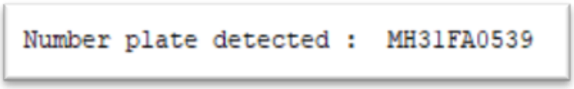


Fig.6 Masked Image

Step6: Then use teserract engine for detecting number plate characters by specifying the languages [6]



```
Number plate detected :  MH31FA0539
```

Fig.7 Number Plate detected

An Alternate method that was attempted :

License Plate Detection (Plate localization): This is the first stage and at the end of this stage, we will be able to identify the license plate’s position on the car. In order to do this, we need to read the image and convert it to grayscale. In a grayscale image, each pixel is between 0 & 255. We now convert it to a binary image in which a pixel is either complete black or white.

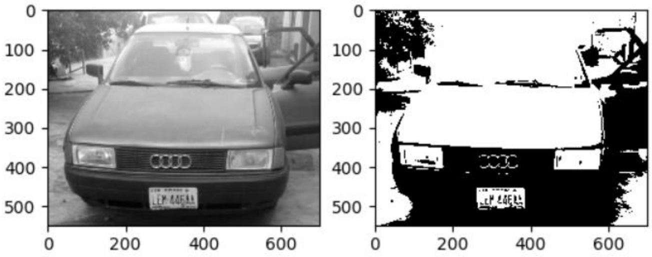


Fig.8 Conversion into gray scale

We need to identify all the connected regions in the image, using the concept of connected component analysis (CCA). Other approaches like edge detection and morphological processing can also be used. CCA basically helps us group and label connected regions on the foreground. A pixel is said to be connected to another if they both have the same value and are adjacent to each other [8]

The previous file was imported so that we can access the values we have there. The measure.label method was used to map all the connected regions in the binary image and label them. Calling the regionprops method on the labelled image will return a list of all the regions as well as their properties like area, bounding box, label etc. We used the patches.Rectangle method to draw a rectangle over all the mapped regions [4,5]

From the resulting image, we can see that other regions that do not contain the license plate are also mapped. In order to eliminate these, we will use some characteristics of a typical license plate to remove them.

1. They are rectangular in shape.
2. The width is more than the height.
3. The proportion of the width of the license plate region to the full image ranges between 15% and 40%.
4. The proportion of the height of the license plate region to the full image is between 8% & 20%.

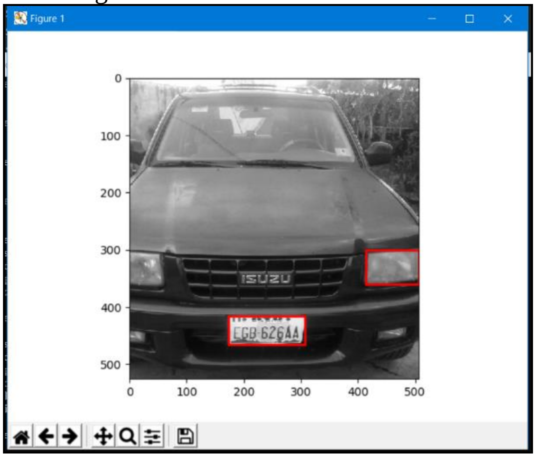


Fig.9 Plate Like Object Detection

Can tweak these characteristics if it doesn't work for the shape of the license plate (the probability of that is low though). Other regions that are likely not license plates are eliminated. However, it's still possible that certain areas (headlamps, stickers etc) that look exactly like a license plate are also being marked. To eliminate those other regions, we'll need to do a vertical projection. The concept is implemented by adding all the pixels in each column. It is assumed that the license plate area will have lots of pixel values due to the fact that characters are written on it [8]

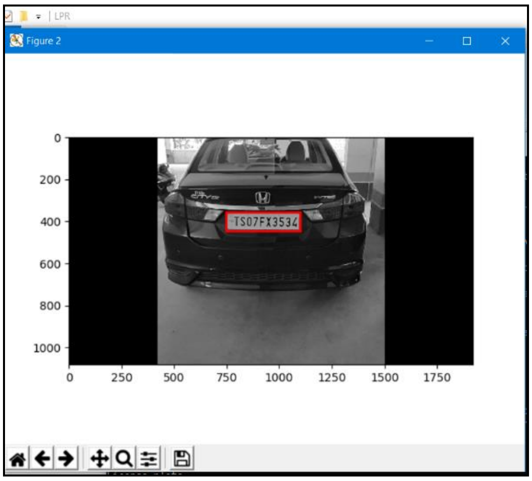


Fig.10 License Plate Detection

A. Character Segmentation

This is the stage where we map out all the characters on the license plate. We'll be using the concept of CCA here also.

We introduced plate_like_objects that is a list of all the regions on the car that look like a license plate. Then a CCA was done on the license plate and each character was resized to 20px by 20px. This was done because of the next stage that has to do with recognition of the character.

In order to keep track of the order of the characters, the column_list variable was introduced to take note of the starting x-axis of each region. This can then be sorted to know the correct order of the characters [8]

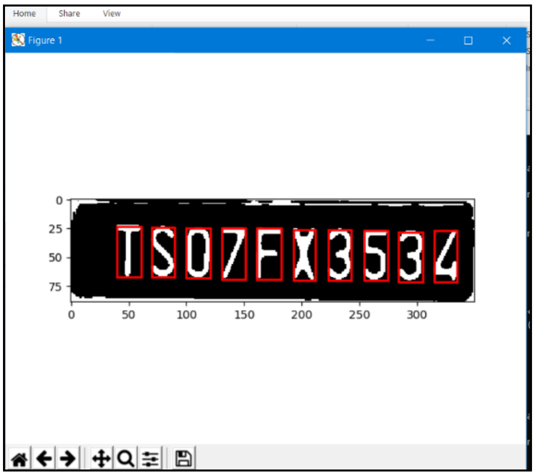


Fig. 11 Character Segmentation

B. Character Recognition

This is going to be the last stage, it's at this stage we introduce the concept of machine learning. Machine learning can simply be defined as the branch of AI that deals with data and processes it to discover pattern that can be used for future predictions. The major categories of machine learning are supervised learning, unsupervised learning and reinforcement learning. Supervised learning makes use of a known dataset (called the training dataset) to make predictions.

We'll be taking the path of supervised learning because we already have an idea of how As, Bs and all the letters look like. Supervised learning can be divided into two categories; classification and regression. Character recognition belongs to the classification category.

All we need now is to get a training data set, choose a supervised learning classifier, train a model, test the model and see how accurate it is, then use the model for prediction..

There are several classifiers we can use with each of them having its advantages and disadvantages. We'll use SVC (support vector classifiers) for this task because it gave the best performance. However, this does not necessarily mean that SVC is the best classifier.

We'll have to install the scikit-learn package for this stage. Each character in the training dataset was used to train the svc model. A 4-fold cross validation was also done to determine how accurate the model is and then the model was persisted to a file so that predictions can be done without training a model anymore.

Now that we have a trained model, we can attempt to predict the characters that we earlier segmented [8]

V. EMPLOYEE VERIFICATION

An employee might not always bring the registered vehicle to the institution. This calls for a facility to let the employee get alternate vehicles and keeping log of these vehicles for future uses. We make use of the unique IDs (QR Codes) that are provided to the employees by the organization. We have used the pyzbar library of python for detection and decoding of the QR codes. The decode function identifies the points that form a quad and mark that region as the QR code. It returns the following details present in the QR Code: type, data and location. Type indicates the kind of code that has been scanned i.e. whether it is a barcode or a QR code, Data indicates the ID in our case and location provides the coordinates of the quad that is detected around the QR Code (Fig 1). Let's say an employee gets an unregistered vehicle to the institute. The gate won't open as the vehicle is not registered. Instead, a scanner will pop up on the screen asking the employee to scan his ID. Once the QR Code is scanned, the ID is extracted from it in the form of a string and it is then verified for its presence in the database. If the ID provided by the employee is a valid ID, the vehicle number will be stored against this ID as an alternate vehicle and the gate will be opened for the vehicle to enter. This will also ensure security as a person, who isn't a member of the institution, will not be allowed entry as their ID will not be available in the database [9,10].

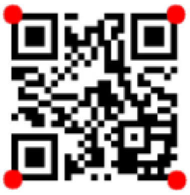


Fig. 12 Red Points indicate the Quad of a QR Code

VI. MONITORED PARKING SPACES

The entering and exiting vehicles will be monitored according to which the number of available parking spaces will be determined. Once the parking lot is filled the rest of the vehicles will be diverted to an alternate parking areas. The directions to which will either be displayed on screen or could be accessed using mobile applications Sensors installed at the parking system will determine whether the parking space is occupied or not and will display the vacant parking spaces on screens [7]

The actual working of the hardware is as follows:

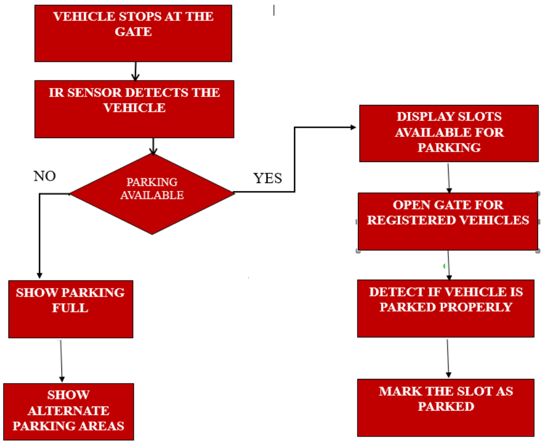

VEHICLE STOPS AT THE GATE
IR SENSOR DETECTS THE VEHICLE
PARKING AVAILABLE
NO
YES
DISPLAY SLOTS AVAILABLE FOR PARKING
OPEN GATE FOR REGISTERED VEHICLES
DETECT IF VEHICLE IS PARKED PROPERLY
MARK THE SLOT AS PARKED
SHOW PARKING FULL
SHOW ALTERNATE PARKING AREAS

Fig. 13 Methodology

The vehicle will stop at the entrance gate. The IR Sensors are deployed on the gate . It will verify is there any space remaining that is not occupied. If yes it will display the slots available for parking. Mark the slot as unavailable as soon as the vehicle is occupied on that place. On other hand if parking is not available then it will display that the parking is full and it will show nearby alternate parking areas to that place on the digital display [7]

The hardware components used for the model are :

A. IR Sensor :

An infrared sensor is an electronic instrument that is used to sense certain characteristics of its surroundings. It senses certain item when placed close to it and glows green light. It can detect upto a distance of 2-10 cm. Here IR Sensors are placed at the gate and they are useful to detect the vehicle [7]

B. Arduino Uno:

Arduino Uno is an open source microcontroller. The board consists of 14 digital and 6 analog pins. The code for the working of gate and sensors was written in Arduino [7]

C. Servo Motor:

A servo motor is a device which can rotate or push an object with a force. In the module it is used to open or close the gate as soon as the condition is fulfilled [7]

D. LEDs:

They are the light emitting device when the current flows through them . Here LEDs were used to display the available slots and the filled slots. Available slot was given by green light and the filled one was given by red light [7]

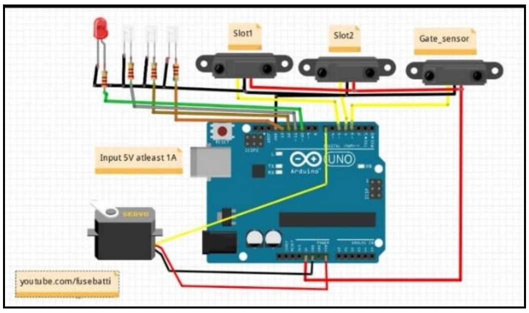


Fig.14 Circuit Diagram

The figure represents the circuit diagram of the module designed. It contains three IR Sensors, four LEDs, servo motor, Arduino board, jumpers. All the inputs of sensors are connected series to each other as well as the grounds are connected in series. The LEDs are connected to specific pin no in Arduino. Servo motor is connected to the analog power supply of the Arduino [7]

VII. ALTERNATE PARKING SPACES

We made an android application which can be used to find alternate parking spaces in case of Full parking. The Vehicles will be guided to alternate public or private parking spaces in the vicinity. The radius can be set by the user to find the parking spaces in the set radius. Along with this the alternate parking space available around will also be displayed on screens.

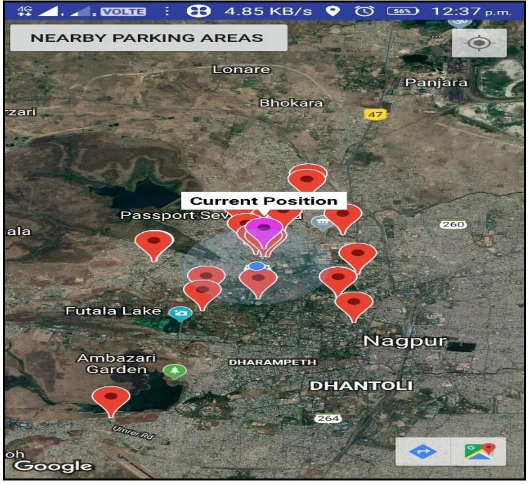


Fig.15 Detectting Current Location

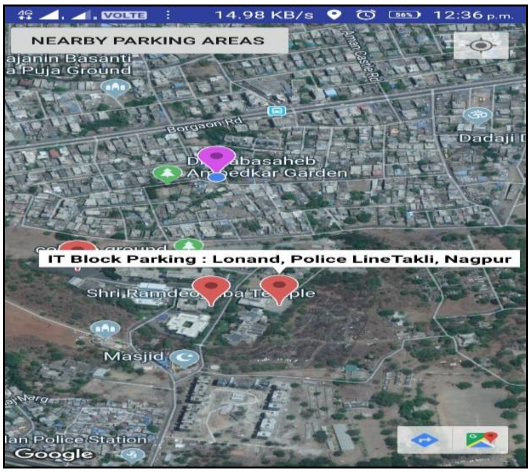


Fig.16 Nearby Parking Spaces

VIII. CONCLUSION

There has been a growing interest in automatic, faster and more intelligent systems than the current approaches due to the security controls and ease of application. With this project of ours we have developed an approach and designed a framework to tackle the problem of manual monitoring of the institute entrances and parking lots by automatically tracking parking occupancies and providing alternate parking spaces along with providing automated security for vehicle entrance in institution or company campuses. It provides the benefits of systematic parking, security control and time management.

REFERENCES

1. Mello C, Costa D, " A Complete System For Vehicle Licence Plate Recognition ", in 16th *International Conference on Signals and Image Processing* , IEEE Xplore , 2009
2. S.Davidson, "The Use Of Automatic Number Plate Recognition In Managing To The Boots Company Headquarter Site", in *IEEE Int. Conf. on Security Technology*, London, 2001.
3. H.Lee, "Real Time Automatic Vehicle Management System Using Vehicle Tracking and Car Plate Number Identification", IEEE ICME, Baltimore, USA, 2003.
4. T.Sirithinaphong, "The Recognition of Car License Plate for Automatic Parking System", *Int. Symp. on Signal Processing and its Applications*, Australia, 1999.
5. M.Sezgin et al. "Survey over image thresholding techniques and quantitative performance evaluation", Journal of Electronic Imaging, 2004, vol. 13, no. 1, pp. 317-327.
6. Dave A.(2018) , License Plate Detector. Retrieved from https://github.com/apoorva-dave/LicensePlateDetector
7. Minhaj A. (2018), Arduino Automated Parking, Retrieved from https://ashrafminhajfb.blogspot.com/2018/08/arduino-automated-parking-garage.html
8. Oladegi F.(2017) , Developing a License Plate Recognition System with Machine Learning in Python. Retrieved from https://blog.devcenter.co/developing-a-license-plate-recognition-system-with-machine-learning-in-python-787833569ccd
9. Rosebrock A. (2018) An OpenCV barcode and QR code scanner.Retrieved from https://www.pyimagesearch.com/2018/05/21/an-opencv-barcode-and-qr-code-scanner-with-zbar/
10. .Database Connection, Retrieved from https://www.w3schools.com/python/python_mysql_getstarted.asp

AUTHORS PROFILE

Dr.M.B.Chandak, is Professor and Head, Department of Computer Science and Engineering at Shri Ramdeobaba College of Engineering and Management, Nagpur. He has total 25+ years of Academic Experience. His area of interest Algorithm Design, Big Data Analytics and IoT.