IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022

DOI: 10.17148/IJARCCE.2022.117112

Detail Oriented Parking using Python and Web Development

N. Swapna[1], Tandu Adarsh[2], Thadur Bharath[3], Vemula Ganesh[4]

[1-4]Department of Electronics and Computers Engineering, Sreenidhi Institute of Science and Technology,

Hyderabad, India

Abstract: The Automatic Parking System (APS) is a globally acknowledged methodology for identifying car licence plates. The number plate of a vehicle can be recognised in real time using APS technologies. Vehicle parking is a crucial part of any transportation system because vehicles are frequently parked at destinations. With an increase in the number of motor vehicles on the road, particularly in developing nations, a vehicle identification system that is effective, economical, and efficient is required. Security guards are utilised for security, parking, and details entrance in most gated communities, restaurants, malls, and other establishments, which is a time-consuming operation that requires keeping track of in the books. This solution was presented to automate the procedure and improve security by ensuring proper detail entry of vehicles. It also shows the parking details of the vehicle with vehicle number search and OTP verification, it shows the applicant parked details as date, in time, out time, and the total charges to be paid by the applicant and we provide a scanner so that they can pay online. The software program was created using an object-oriented analysis and design process, and it uses Optical Character Recognition (OCR) to recognize and capture the car number plate using the camera. The proposed method reduces the time it takes to register from 30 seconds to 6 seconds, while also improving security and ensuring correct data. We may also find the parking and pay option costs using the time calculation.

LITERATURE REVIEW:

In recent years, significant research and development of algorithms in intelligent transportation has gotten increased attention. For traffic control and law enforcement of traffic laws, an automatic, rapid, accurate, and robust vehicle plate recognition system has become necessary, and ANPR is the solution. This research focuses on an enhanced OCR-based licence plate identification system that employs a neural network-trained collection of object characteristics.
To enhance accuracy, a blended algorithm for licence plate recognition is developed and compared to existing approaches. License Plate Localization, Plate Character Segmentation, and Plate Character Recognition are the three key components that make up the system. The technique is tested on 300 national and international LP photos of motor vehicles, and the results are convincing the main requirement.

APPLICATIONS:

Since the licence number is the primary, most widely acknowledged, human readable, and mandated identification of motor vehicles, Automatic Number Plate Recognition (ANPR) has a wide range of uses. Some of the topics are:
1. Parking: ANPR can help with ticketless parking fee management, parking access automation, vehicle location guidance, parking fee charging, car theft prevention, "lost ticket" fraud, and fraud by changing tickets.
2. Access Control: License plate recognition automates vehicle access control management, increasing security, logistics carpool management, security guide assistance, event logging, event management, access diary keeping, and analysis and data mining possibilities. It is also incredibly successful in border and law enforcement controls.
3. Tolls on highways, motorways, roads, and bridges: Tolls are a typical manner of funding highway, motorway, road, and bridge improvements. Efficient road tolling decreases non-payment fraud, increases charging effectiveness, and reduces the amount of staff necessary to process exception occurrences, all of which can be achieved with ANPR.

REQUIREMENTS:

Dataset: Images of Indian vehicles where the number plate is clearly visible.

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022

DOI: 10.17148/IJARCCE.2022.117112

Fig 1 data base

Programming language: Python3
Libraries used: sys, glob, os, glob, numpy, cv2, PIL, pytesseract, re, MySQLdb mysql.connector
Methodology:
Pre-processing, detection, recognition, and searching are the four primary aspects of the suggested methodology, as indicated in the diagram below.

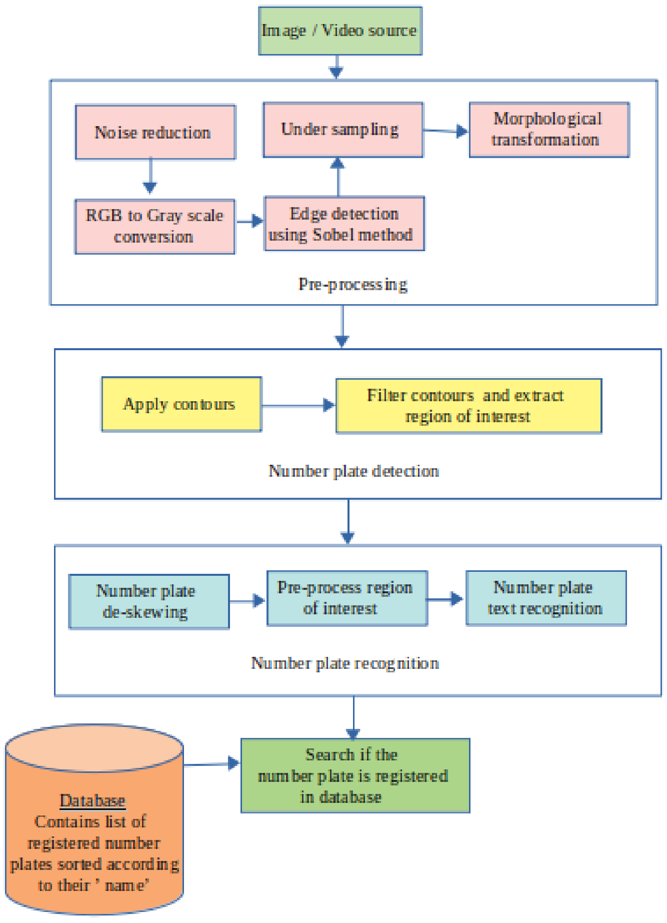


Image / Video source
Noise reduction
Under sampling
Morphological transformation
RGB to Gray scale conversion
Edge detection using Sobel method
Pre-processing
Apply contours
Filter contours and extract region of interest
Number plate detection
Number plate de-skewing
Pre-process region of interest
Number plate text recognition
Number plate recognition
Database Contains list of registered number plates sorted according to their ' name'
Search if the number plate is registered in database
Fig 2 Block diagram

BRIEF DESCRIPTION OF STEPS:

Step 1: Image pre-processing

Step 1.1: Noise reduction:
The goal of Gaussian smoothing and filtering is to reduce noise and detail. This will be useful for subsequent picture processing. Gaussian filter can be represented mathematically for an image as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

To obtain a smoothed image, the input image is made to convolve with this 2-D 'G' matrix. The following function in

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022

DOI: 10.17148/IJARCCE.2022.117112

OpenCV can be used to apply Gaussian smoothing:,

cv2.GaussianBlur(image,(5,5),0)

where (5,5) is the filter size and '0' is the model to obtain the value of standard deviation (sigma).

Step 1.2: RGB to Grayscale conversion:

Converting an RGB image to grayscale saves a lot of time because we only have to conduct convolution with the Sobel filter over one 2D matrix instead of three channels in an RGB image.

Another reason is because in picture edge detection, we are primarily interested in seeing intensity changes, which is easier to do in a gray-scaled image.

Run filter over image

$$\frac{\partial \boldsymbol{f}}{\partial x} = \boldsymbol{S}_x \otimes \boldsymbol{f} \qquad \frac{\partial \boldsymbol{f}}{\partial y} = \boldsymbol{S}_y \otimes \boldsymbol{f}$$

Image gradient

$$\nabla \boldsymbol{f} = \left[\frac{\partial \boldsymbol{f}}{\partial x}, \frac{\partial \boldsymbol{f}}{\partial y}\right]$$

Step 1.3: Edge detection using sobel method :

The gradient of picture intensity at each pixel within the image is calculated via Sobel edge detection. It determines the direction and pace of change in the most significant increase from light to dark. When utilising the Sobel method to find edges, the following phrases are used:

cv2.Sobel(img2,cv2.CV 8U,1,0,ksize=3)

in OpenCV is used to detect edges with a kernel size of 3.

Step 1.4: Under-sampling: The algorithm for number plate identification and recognition is designed to run at a constant frame rate. Image processing techniques for high-resolution photographs are, unsurprisingly, sluggish. In fact, considering photographs with such a high quality is superfluous. If the resolution exceeds a predetermined threshold, this stage decreases it.

Step 1.5: Morphological transformation :

Morphological modifications include top-hat and black-hat filters. The top-hat operation is used to highlight bright things of interest against a dark background, whereas the black-hat operation (also known as bottom-hat) is used to highlight dark objects against a bright background. Top-hat results are added to the original image in this work, whereas black-hat results are deleted.

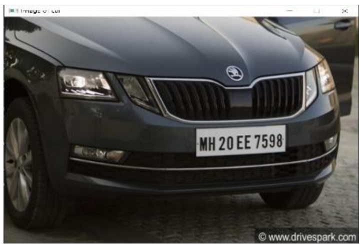


Fig 3 Input image of car

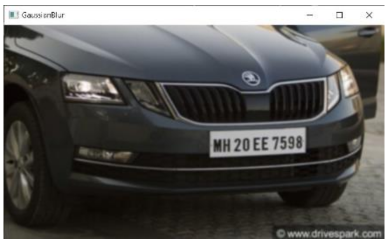


Fig 4 Gaussian blur effect

DOI: 10.17148/IJARCCE.2022.117112

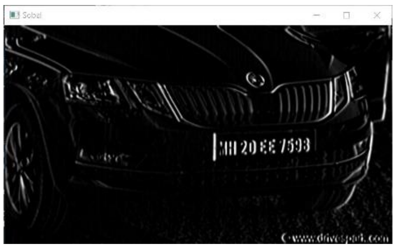


Fig 5 Sobel Effect

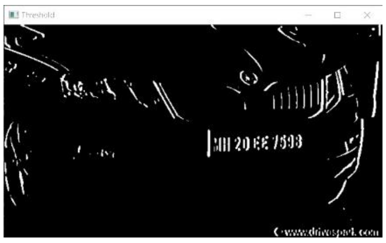


Fig 6 Threshold effect

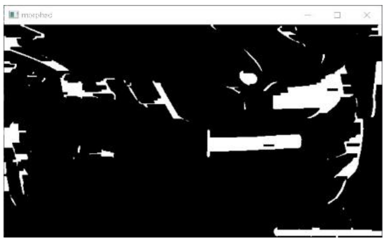


Fig 7 Morphed effect

Step 2: Number plate detection

Step 2.1: Apply Counters: The algorithm for creating contours is Contour Tracing, also known as Border Following. A contour is a line that connects spots of equal intensity along a border. Locating contours in OpenCV is similar to finding a white object against a black backdrop, hence the Inversion operation must be used during the Adaptive Gaussian Thresholding stage.

Step 2.2: Filter Contours and extract region of interest: Contours are used for small areas, particularly sharp edges and noise outliers. Although a human eye may immediately 609ecognize that such outlines are superfluous, this must be factored into the software.
Bounding boxes were initially put to each contour. The minimum contour area, minimum contour width and height, and minimum and maximum permissible aspect ratios were then examined for each contour. As a result, most of the unneeded contours were filtered out, bringing us closer to our goal of detecting number plates.

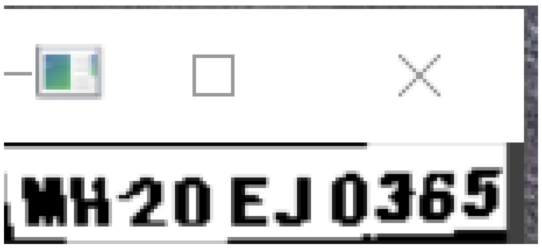


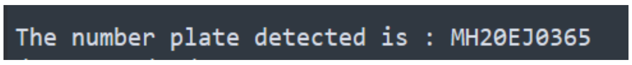
Fig 8 Captured number plate

Step 3: Number plate recognition

Step 3.1: Number plate de-skewing: The amount of rotation required to align an image horizontally and vertically is known as skew. Skew is expressed as a percentage. Deskewing is the process of removing skew from an image by rotating it in the opposite direction by the same amount as the skew. As a result, the text runs across the page rather than at an angle, and the image is horizontally and vertically aligned. This phase is carried out in our project utilising ratio and rotation ()

Step 3.2: Pre-process region of interest: As with the number 'zero,' it is possible for two or more contours to totally overlap. If the inner contour is recognised during the contouring process, it may reside entirely within the outer contour. Both contours may be identified as independent characters throughout the recognition process as a result of this phenomena. Before performing the recognition phase, we resize the image if necessary.

Step 3.3:Number plate text recognition: Python-tesseract is a python-based optical character recognition (OCR) programme. It will detect and "read" text contained in photos, in other words. Finally, we used this programme to extract the text from the filtered, de-skewed contour.



```
The number plate detected is : MH20EJ0365
```

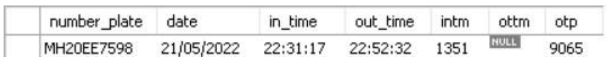
Fig 9 Number plate detected in TEXT

Step 4: Storing the data in portal

Step 4.1: Create database: Using mysql workbench or other sql queries create a database and then using it create a table which contain the data in it. While creating the table use the same variables names for column to connect the database.

Step 4.2: connecting database: First install the libraries require to run the sql in python then import them in python program and using the "mysqlconnector.connect" connect the program to database by initialising the values .

Step 4.3: Storing the data in database : Using the insert insert the data into database and date, time, other details required . Then using the values calculate the total charges as store the value in it.



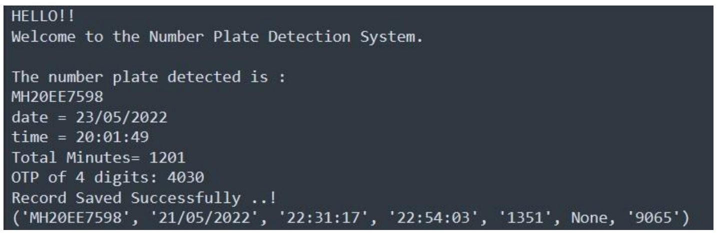
number_plate	date	in_time	out_time	intm	ottm	otp
MH20EE7598	21/05/2022	22:31:17	22:52:32	1351	NULL	9065

Fig 10 Database in portal

Step 5: User interface:

Step 5.1: Online details fetch : User need to open the website and enter his number plate number to fetch his details from the portal like intime and out time and total charges needed to pay.

Outputs:

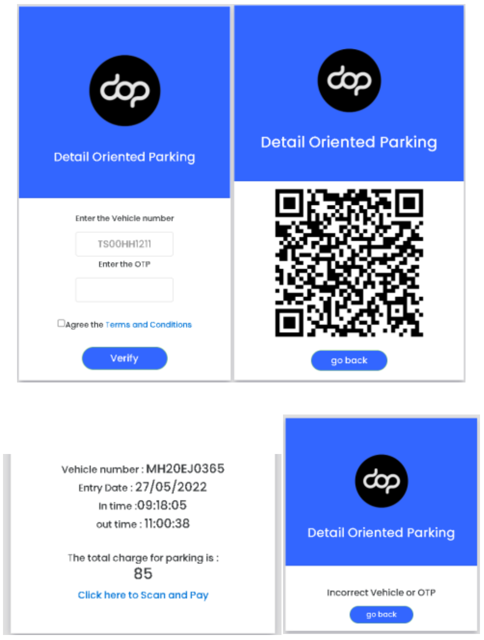


```
HELLO!!
Welcome to the Number Plate Detection System.

The number plate detected is :
MH20EE7598
date = 23/05/2022
time = 20:01:49
Total Minutes= 1201
OTP of 4 digits: 4030
Record Saved Successfully ..!
('MH20EE7598', '21/05/2022', '22:31:17', '22:54:03', '1351', None, '9065')
```

Fig 11 Detected number plate and other details

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified ⁜ Impact Factor 7.39 ⁜ Vol. 11, Issue 7, July 2022
DOI: 10.17148/IJARCCE.2022.117112



Vehicle number : MH20EJ0365
Entry Date : 27/05/2022
In time :09:18:05
out time : 11:00:38

The total charge for parking is :
85
Click here to Scan and Pay

Fig 12 User Interface for details

CHALLENGES AND SOLUTION:

1. Objects that are brilliant and dark.
Solution: When we convert an RGB image to grayscale, many of the features become less noticeable. Converting the image to grayscale is the best way to perceive the change in intensity and thus cope with bright and dark things in the image.

2. Dealing with photos that are noisy.
Solution: To reduce the noise in the image, we convolved it using a Gaussian filter during the pre-processing stage.

3. Dealing with skewed or cross-angled number plates.
Solution: To detect the text on the number plate, de-skewing, or rotating the picture to the proper position, can be done.

4. Dealing with non-standard number plates.
Solution: In the situation of non-standard/partially broken number plates, the code we built will not produce any results. As a result, they will not be saved in the database.

CONCLUSION:

With this application, we can reduce the paperwork and hardwork of securities and direct details entry of vechiles can save the lot of time and implementing this can give an accurate details of a vechile parked in the slot and the charges for the parking is shown.

REFERENCES:

1. M M Shidore, and S P Narote. (2011) “Number Plate Recognition for Indian Vehicles”International Journal of Computer Science and Network Security 11 (2): 143-146
2. Sang Kyoon Kim, D. W. Kim and Hang Joon Kim. (1996) “A recognition of vehicle licenseplate using a genetic algorithm based segmentation,” Proceedings of 3rd IEEE International Conference on Image Processing, Lausanne.
3. https://docs.opencv.org/master/
4. https://www.python.org/about/gettingstarted/
5. https://www.researchgate.net/publication/299858935_Proposal_for_Auto matic_License_and_Number_Plate_Recognition_System_for_Vehicle_Id entification
6. https://medium.com/@heliyahasani/objective-metrics-and-gradient-descent-algorithms-for-adversarial-examples-in-machine-learning-8514de840389