AI ENABLED CAR PARKING SYSTEM USING OPENCV

A PROJECT REPORT
submitted by

S.SWETHSHREE
E.NARMADHA
VR. PREETHI
SURUTHTHIGAA

K.VHARSHNI

In partial fulfilment for the award of the degree
Of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING

MNM JAIN ENGINEERING COLLEGE,THORAIPAKKAM 600 049.

ANNA UNIVERSITY

MARCH 2023

PROFESSIONAL READINESS FOR INNOVATION ,EMPLOYABILITY, ENTEREPENEURSHIP BY IBM

TABLE OF CONTENT

1.1 Technology used

3. METHODOLGY

i. STEPS

1. Open CV
2. Haar Cascade
3. TesseractOCR

4.OPEN CV and Haar Cascade Detecting Vehicles License Plates with OpenCV

5.Car Number Plate Recognition and Extraction using TesseractOCR

i. Retrive Number from Car Number Plate and Maximize Image
ii. Convert image to gray-scale image
iii. Smoothing the image [de-noising]

6.RESULTS

7.Final step

8.CONCLUSION

9.REFERENCES

ABSTRACT

With today's changing world and huge population we are moving towards a society with higher-rise buildings and towers. Although high-rise buildings and towers solve one of the issues by allowing to accommodate a greater number of people, it tends to create another problem of vehicle parking. We through this experimental project are trying to solve the same problem, by tracking vehicles through their number plates, and allowing residents of the society to give authenticated access to vehicle entry and know if their vehicle has left the society parameters, hence solving/reducing motor vehicle theft. With this 'AUTOMATED PARKING SYSTEM', every motor vehicle entering the society would be registered on the system. And complete data of the resident's vehicle(s) will be maintained. Every vehicle of the resident will be marked locked once entered, and residents can mark it as unlocked through the portal. If a vehicle is being driven out of the parameter when it is marked locked the owner will get an immediate alert through SMS.

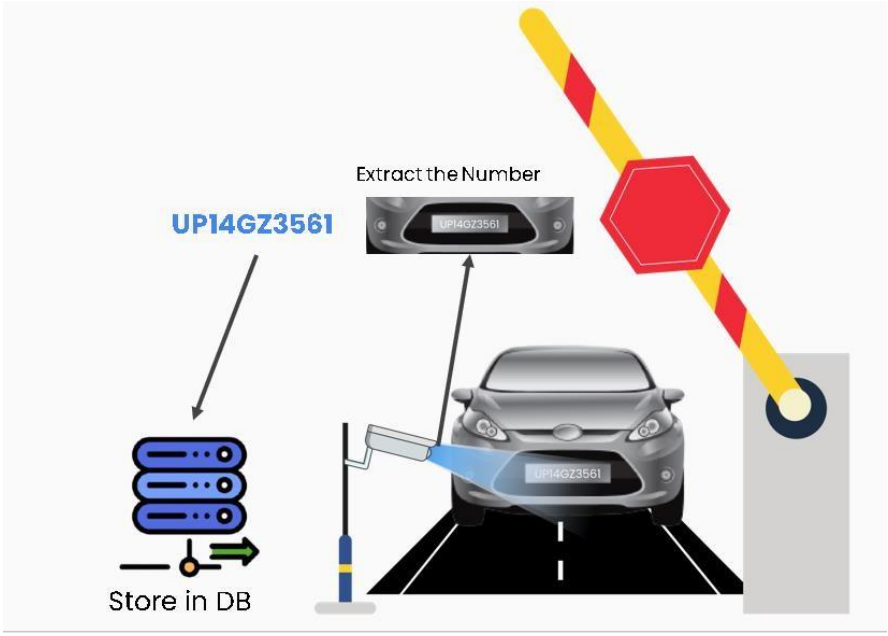
INTRODUCTION

In any surveillance system, we often need to extract numbers from vehicle number plates that enter or leave a parameter. Going through each video and fetching the data is time-consuming and requires a lot of human effort. Computer vision is an intriguing technology because it is like human sight, the most crucial human senses. There are a lot of places where computer vision is used like in face detection, in self-driving cars and one of the cases is car number plate extraction. This car number plate detection and extraction can be used for both real-time video feeds and for old recordings of videos. This can be used in traffic monitoring and automatic challan systems in case any vehicle breaks laws and in automatic car parking systems etc. In an automated parking system, the number plate can be extracted from the CCTV video feed and allowed to enter if that vehicle is authorized.

Technology Used

OpenCV, NumPy, Python, Matplotlib, TesseractOCR.

METHODOLOGY:



Step 1:

The vehicle's number plate will be used to deduce the vehicle number. For this, we will use 2 open-source software: TesseractOCR and OpenCV along with Python Jupyter's notebook for our project. The technologies used are:

(1) OpenCV: OpenCV is a great library for video capturing, image processing, and machine learning, and helps to provide the standard development environment for computer vision applications by including capabilities like face detection and object detection.

(2) Haar Cascade: It's an Object Detection Algorithm which recognizes faces in snapshots and or real-time recordings. Extracts numerical values (e.g., edges, lines) well in the concept of a complete image that enhances the heavy gripping method of extracting pixel figures in multiple regions of the full image. It also uses 'Cascade of Classifiers'. That means rather than using hundreds of class dividers for multiple elements within an image at a time (which does not work properly), the individual classifier is applied.

(3) TesseractOCR: TesseractOCR is a free optical character recognition engine that is open-source and very accurate as well. It is widely regarded as one of the most used and most accurate open-source engines for OCR.

Detecting Vehicles License Plates with OpenCV and Haar Cascade:

1. First, let's import the car images that we'll be working on during the steps. Since BGR format images are imported by OpenCV by default, we'll need to use cv2.cvtColor and convert the image into RGB format before asking matplotlib to show it.
2. Now is the time to use OpenCV's Cascade Classifier function to import the Haar Cascade feature set for Indian automobile plates.
3. Now we will use the Cascade Classifier function in OpenCV to import Haar Cascade feature set for Vehicle License plates.
4. The detection is then carried out using the Cascade Classifier's detect MultiScale function. Let's have a look at the detect MultiScale function in openCV in more detail. The method returns a list of all the rectangle parameters where the items were discovered and allows us to find objects of various sizes in the provided image.

a) At each picture scale, the scale Factor defines how much is the image size decreased. Essentially, the object detection models are trained to detect things of a specific size and may overlook plates that are larger or smaller.

b) MinNeighbors enables us to set the minimum number of neighbors a candidate rectangle must have in order to be kept. With a higher value, fewer detections are made, but the ones that are made are of higher quality and high accuracy.

c) Run the car plate detection function and check if the plate is detected by the model or not. If the plate is detected it will be seen bordered with a red rectangle.

d) The next step is to use OCR to extract the Vehicle License Number from the plate.

Car Number Plate Recognition and Extraction using TesseractOCR

(1) Retrieve Number from Car Number Plate and Maximize Image.

To perform OCR correctly, we need to perform some image processing steps. Let's start by making an image of the Vehicle license plate. We accomplish this by building up a function identical to the one we used earlier to detect automobile plates, except now we will extract and return the area of interest that is the Vehicle number plate as a new picture. In addition, since the vehicle license plate is only a small portion of the input picture, we'll want to enlarge it (using OpenCV's resize technique). Running these two routines will provide us with our area of interest, which is the vehicle license plate.

(2) Convert image to gray-scale image

After that, convert the image to grayscale from RGB color. By doing this essentially we are reducing the number of colors in the image, which might disturb the OCR detection. Since we need to focus more on the edges of the image and using a grey scale image helps us to do that more efficiently.

(3) Smoothing the image (de-noising)

Now, we will apply the slide to the image. The smoothness helps us to eliminate noise in the image and makes the app focus more on the details of the image. This is actually what we call the denoising step, which makes the characters in a picture very different and visible.

We used median blurring to smooth (usingcv2.medianBlur) and after this step image is ready for OCR process.

RESULTS

The car image we have imported to check if our detection model works properly or not:

Enlarging the image and detecting the number plate such that the number plate is bounded by a rectangle.

Extracted area of interest i.e. vehicle license plate, Getting the Actual Number plate through the complete vehicle picture so that we may do OCR exclusively on the car's license plate.

Converting the car plate image to the Grey Scale image.

Smoothed grayscale picture of vehicle license plate

In the end, the output we get is the Number present on the vehicle License plate as a string. The number on the vehicle's number plate match the recognized wording of "**DL7CQ1939**"

DL7CQ1939

Final Step:

After extracting the number from the vehicle image, we will check for parking access and permit if the vehicle's number is whitelisted.

CONCLUSION

After performing these steps we can easily detect the License number from the image/ video of the car through the camera. After that, the information about the vehicle and vehicle owner will be directly saved to the

database. Hence through this, we can automate the car parking process and allow entry only to the authorized vehicles.

REFERENCES

[1] OpenCV: Automatic License/Number Plate Recognition (ANPR) with Python

[2] OpenCV: cv::VideoCapture Class Reference

[3] Graphical User Interfaces with Tk — Python 3.10.1 documentation

[4] Welcome to TesseRACt's documentation! — TesseRACt 0.1.1 documentation

[5] OpenCV Tutorials

[6] Russian Car Plate Detection with OpenCV and TesseractOCR

[7] Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)

[8] License Plate Recognition with OpenCV and Tesseract OCR