e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

SENSOR ENABLED COLLEGE PARKING SYSTEM

Chavi Sehgal*1, Vipin Yadav*2, Jatin Pal*3, Kunwar Abhay Patel*4, Sonelal Patel*5

*1,2,3,4,5Computer Science And Engineering, Inderprastha Engineering College, Ghaziabad, India.

ABSTRACT

Security has always been a priority for humanity. Today we have video surveillance cameras in schools, hospitals and every other public place that make us feel safe. With advances in technology, especially in image processing and machine learning, it is possible to make these cameras smarter by training them to process information from video feeds.

In this article we will learn how to identify and read the license plate number from an automobile using Raspberry Pi and OpenCV. We use some random vehicle images from Google and write a program to locate the number plate using OpenCV contour detection, and then read the number from the plate using Tesseract OCR.

In areas where parking space is allocated for a specific vehicle, an incorrectly parked vehicle can be identified. The number plates of the vehicle come in different shapes and sizes and they also vary in color. This allows the vehicle to be identified by its number plate. Number plate detection helps detect stolen cars, car parking management systems and vehicles in traffic. In this method all the letters and numbers used in the number plate are sorted using the bounding box method. After splitting, the template matching method is used to identify numbers and letters. The decoded number plate is further used for identification, matching and documentation of vehicle details.

Keywords: Smart Parking System, Sensor, Image Processing, Machine Learning, Neural Networks.

I. INTRODUCTION

The "College Parking Certification" was developed to address the problem encountered in the implementation of manual systems. This software supports to remove and, in some cases, to minimize the problems that this current system is facing. Also, this system is designed for the special needs of the organization to work in a smooth and effective manner. Develop a streamlined full functional system that identifies the vehicle number by simply uploading the image or clicking on the image of the vehicle number plate. The API identifies car details such as owner name, vehicle model, registration date vehicle and engine number and vehicle identification number. If the details match the college parking database, send a command to the sensor to use for parking. Every organization, large or small, has challenges in solving and managing vehicles, parking slots, duration information. Each parking system has different parking slot requirements, so we will design specific employee management systems to suit your admin needs. It is designed to assist in strategic planning and to assist in determining your organization.

II. LITERATURE SURVEY

S. No	Name	Year	Author	Methods	Review
1.	Smart Parking using Artificial Intelligence	2019	Jaspreet Kaur	Pattern recognition and computer vision fields	Optical Character Recognition (OCR) Embedded DSP-Platform Pattern match method
2.	Automatic Number Plate Recognition:A Detailed Survey of	2021	M. Hassaballah	Character extraction and recognition	Pattern recognition based on fuzzy logic Did not yield good results for skewed

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

	Relevant Algorithms				plate
3.	A System Design for License Plate Recognition by Using Edge Detection and Convolution Neural Network.	2018	Prashengit Dhar	License Plate Extraction	Edge information Analysis Probabilistic model, Subspace Projection Neural Network
4.	Smart vehicle identification system using OCR	2017	Jain K. Choudhury	Segmentation	Tree of Shapes Hidden Markov Chains
5.	An Iranian license plate recognition system based on color features	2014	Ashtari, A.,H Nordin	Recognition	Gabor transform K-Means Algorithm
6.	Automatic Number Plate Recognition System: A Histogram Based Approach	2016	Dipayan Mitra, Soumit Banerjee	Network Method for Plate Localization	Noisy, tilted plates or other degraded forms of number plate were not considered

III. METHODOLOGY

License plate identification consists of three main steps.

1. License Plate Detection: The first step is to locate the license plate from the car. We use the format option in OpenCV to find rectangular objects to find number plates. Accuracy can be improved if we know the exact size, colour and approximate position of the number plate.

2. Character segmentation: Once we know the license plate, we need to cut it out and save it as a new image. Again, this can be done easily using OpenCV.

3. Character Recognition: Now, certain characters (numbers / letters) are written exactly on the new image we got in the previous step. So, we can do OCR (Optical Character Recognition) on it to find the number.

4. Vehicle Authentication: After character recognition, we check if the identified vehicle number is in our database, and the sensor can provide access to the parking space if our database matches.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

IV. RESULTS AND DISCUSSIONS

In [24]:
```
import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

In [25]:
```
img = cv2.imread('car3.jpg')
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
```

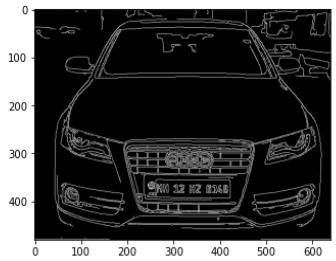
Out[25]:
```
<matplotlib.image.AxesImage at 0x1a74eefe3a0>
```

In [26]:
```
bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
edged = cv2.Canny(bfilter, 30, 200) #Edge detection
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

Out[26]:
```
<matplotlib.image.AxesImage at 0x1a74ef4da30>
```

In [26]:
```
bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
edged = cv2.Canny(bfilter, 30, 200) #Edge detection
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

Out[26]:
```
<matplotlib.image.AxesImage at 0x1a74ef4da30>
```



In [27]:
```
keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
```

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

In [28]:
```
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break
```

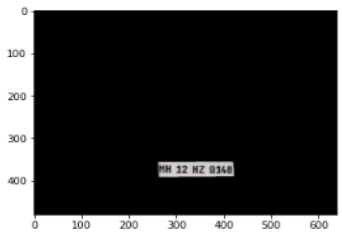
In [29]:
```
location
```

Out[29]:
```
array([[[262, 356]],

       [[420, 355]],

       [[423, 389]],

       [[264, 391]]], dtype=int32)
```

In [30]:
```
mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0,255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)
```

In [31]:
```
plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
```

Out[31]:
```
<matplotlib.image.AxesImage at 0x1a74ef9dc70>
```



In [32]:
```
(x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gray[x1:x2+1, y1:y2+1]
```

In [33]:
```
plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

Out[33]:
```
<matplotlib.image.AxesImage at 0x1a74effa1f0>
```

In [34]:
```
reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

Out[34]:
```
CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
[([[0, 4], [162, 4], [162, 34], [0, 34]], 'MH 12 HZ 0148', 0.8178974409476597)]
```

In [35]:
```
text = result[0][-2]
font = cv2.FONT_HERSHEY_SIMPLEX
res = cv2.putText(img, text=text, org=(approx[0][0][0], approx[1][0][1]+60), fontFace=font, fontScale=1, color=(0,255,0), thickness=2, lineType=cv2
res = cv2.rectangle(img, tuple(approx[0][0]), tuple(approx[2][0]), (0,255,0),3)
plt.imshow(cv2.cvtColor(res, cv2.COLOR_BGR2RGB))
```

Out[35]:
```
<matplotlib.image.AxesImage at 0x1a74cc87b50>
```

V. CONCLUSION

Vehicle parking is a very important issue nowadays and its need is increasing day by day. In India we still use manual vehicle parking system and once we must park our car which needs good light, we've got to waste time and worry about the parking ground. Problems like finding empty space around fuel. Another problem is that the confusion that happens during parking, as there's no special arrangement for anyone to park anywhere,

sometimes causing damage when vehicles move out or within the car parking zone. there's also a security issue. we are introducing new car parking system. The system works as follows: the motive force will place the vehicle ahead of the garage door and there'll be a monitor available where the amount of obtainable parking slots are going to be displayed. The user will must provide his itinerant number and car's license number and also the operator will give command to open the gate, a car parking tray will come & will park the car within the garage.

VI. REFERENCES

[1] Atif, Y ; Ding, J ; Jusfeld ; MA Internet of Things Approach to Cloud-Based Smart Car Parking. Procedia Compute. Science 2016,98, 193-198.
https://www.sciencedirect.com/science/article/pii/S1877050916321603?via%3Dihub

[2] Yang, Z .; Pun-Cheng, L.S.C. Vehicle recognition in intelligent transportation systems and its applications in different environments: A review. Image Vis. Calculation. 2018,69, 143-154
https://www.sciencedirect.com/science/article/abs/pii/S0262885617301592?via%3Dihub

[3] Klappnecker, A .; Lee, H .; Welch, JL made it easy to find available parking spaces. Temporary Net.2014,12, 243-249
https://www.sciencedirect.com/science/article/abs/pii/S157087051200042X?via%3Dihub

[4] Lynn, T .; Rivano, H .; Le Mouel, F. Survey of Smart Parking Solutions. IEEE Trans. Intelligence. Transpist. 2017,18, 3229–3253https://ieeexplore.ieee.org/document/7895130

[5] Geng, Y .; Cassandras, CG. Infrastructure and implementation of the new "smart parking" system. Procidia Society. Behavior. Science 2012
https://www.sciencedirect.com/science/article/pii/S1877042812043042?via%3Dihub

[6] Thomas, D .; Kovur, B.C. Genetic algorithmic approach to autonomous smart vehicle parking systems. Procedia Compute. Science 2018,125, 68-76.
https://www.sciencedirect.com/science/article/pii/S1877050917327758?via%3Dihub

[7] Revathi, G .; Dirt, VRS Smart Parking Systems and Sensors: A Survey. 2012 International Conference Proceedings on Computing, Communication and Applications, Tamil Nadu, India, 22-24 February 2012 https://ieeexplore.ieee.org/document/6179195