Project Report

On

License Plate Recognition System

Submitted By

S VIVEK (R170851)

U GOVARDHAN (R170856)

T BABJI (R170854)

Under the guidance of

P Santhosh Kumar

Department of Computer Science Engineering

Rajiv Gandhi University of Knowledge and Technologies (RGUKT),

R.K.Valley, Kadapa, AndhraPradesh.

Rajiv Gandhi University of Knowledge and Technologies (RGUKT),

R.K.Valley, Kadapa, Andhra Pradesh.

CERTIFICATE

This is certify that the project work titled "**License Plate Recognition System**" is a bonafied **project work** submitted by S.VIVEK (R170854), T.BABJI (R170851), U.GOVARDHAN (R170856) in the department of COMPUTER SCIENCE ENGINEERING in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering for the academic year 2021-2022 carried out the work under the supervision

GUIDE

P SANTHOSH KUMAR

Head of the department

P HARINADHA

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director, **Prof. K. SANDHYA RANI** for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department **Mr. P.HARINADHA** for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college **Mr. P SANTHOSH KUMAR** for his guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement

INDEX

Abstract

License plate recognition System (LPRS) is an image processing technology which uses license plate to identify the vehicle. The objective is to design an efficient automatic number plate identification system by using vehicle number plate on pre-trained dataset.

The developed system takes the vehicle image as input, the vehicle number plate region is extracted using the image segmentation in the given image. By using the Optical Character Recognition (OCR) technique the system recognizes the characters from the segmented image.

By using the Pytesseract, a python library, which is used to recognize the characters in the image. The extracted characters are displayed using the Tkinter GUI interface in the image format. The system is capable of producing the sequence of characters in the string format too.

The system is implemented using the vscode Jupiter extension, and those are performed on pre-trained real time images. It is observed from the project that the developed system successfully detects and recognizes the vehicle number plate on pre-trained real images.

INTRODUCTION

Advanced machine vision technology called license plate recognition is utilized to identify automobiles by their license plates without the direct involvement of a human. The data of vehicle numbers provided by this Intelligent Transportation System development can be used for monitoring, analyzing, and follow-up. This technique is becoming more and more used in a variety of applications besides traffic monitoring. This technology is becoming more and more common in a number of other applications besides traffic monitoring, including highway toll collection, border and customs checkpoints, and parking access control systems. ANPR is projected to play a significant role in global security as a result of the recent rise in terrorist attacks.

In this LPR mainly there are 3 steps:

- Taking picture Input
- Analyzing the picture
    - Original image to gray image
    - Applying bilateral filter and canny edge detection
    - Counters from edged image
    - Rectangular shape detection
    - Cropped image of a license plate
    - Applying OCR
- Displaying the output

PURPOSE

License plate detection and recognition is the technology that uses computer vision to detect and recognize a license plate from an input image of a vehicle. This technology applies in many areas. On roads, it is used to identify the cars that are breaking the traffic rules.

Police departments use License Plate recognition to identify vehicles involved in traffic offences, verify whether a vehicle is registered, and enforce the law. Authorities can identify vehicles and determine their locations if the extracted output is an authorized/registered user in real-time.

SCOPE

License Plate Recognition system is used in various areas nowadays such as automatic toll collection, Border crossings, parking system, Traffic control, stolen cars tracking, maintaining traffic activities and law enforcement etc.

As a future work the developed system would be concentrated upon increasing the accuracy of text localization and graphics removal in caption text images. It can be evaluated using various other available image data bases and using various other classifiers. The proposed methods can be further improvised and applied for automatic mixed mail sorting.

In this system only one image is taken as input in that image only one vehicle is identified if there is multiple vehicles contains. In future it develops to detect two or more vehicles at a time and taking video also as an input and recognizing the text

Requirement Specifications

Hardware Specification:

Processor	AMD A9-9420 RADEON R5, 5 COMPUTE CORES 2C+3G 3.00 GHz or i3 & above
RAM	4GB
Hard disk	512GB or More

Software Specifications:

Language:	Python
Tools:	VS Code, Jupyter Notebook
GUI:	Tkinter
Modules:	▪ Pip ▪ Numpy ▪ pytesseract ▪ easyocr ▪ opencv ▪ matplotlib ▪ imutils

Implementation

It is seen that the security forces and authorities face problems whenever security forces chase a vehicle or they can't catch a vehicle which broke traffic rules. Authorities find it very hectic on a busy day to log the vehicle numbers manually in a parking lot. So, in order to make the entire process autonomous, we can install this system so as to automatically detect the vehicle which breaks the traffic rules.

They will give vehicle image containing license plate as input and the output is displayed in the monitor.

Step 1: Installing required modules.

- Python

  Ubuntu:

  sudo apt update
  sudo apt upgrade
  sudo apt install python3

- **pip3:**

  sudo apt – get update
  install pip3
  sudo apt-get-y install pip3 python3-pip

  - **verification:**
    - pip3—version
- **Install open CV library with pip**
  - Pip3 install open cv-python
- **Install matplotlib library with python**
  - sudo apt-get install python3-matplotlib
- **Install Numpy library through python**
  - pip3 install numpy
  - pip3 install numpy opencv-contrib-python
- **Install pytesseract library through python**
  - pip3 install pytesseract
- **Install imutils library through python**

- ❖ pip3 install imutils

➢ **Install easyocr through python**

- ❖ pip3 install easyocr

➢ **Install tkinter through python**

- ❖ sudo apt-get install python-tk

Note:

1. Incase pip3 is not worked properly run the command using “pip”.

2. Download VS Code or jupyterNotebook or Pycharm IDE’s.

Step2:

Now open the text editor or the IDE’s create a file with **.py** extension and mention the path to the pytesseract and import the import modules as shown below,

```python
import cv2
from matplotlib import pyplot as plt
import numpy as np
import pytesseract
pytesseract.pytesseract.tesseract_cmd = 'C:\Program Files\Tesseract-OCR\\tesseract'
```

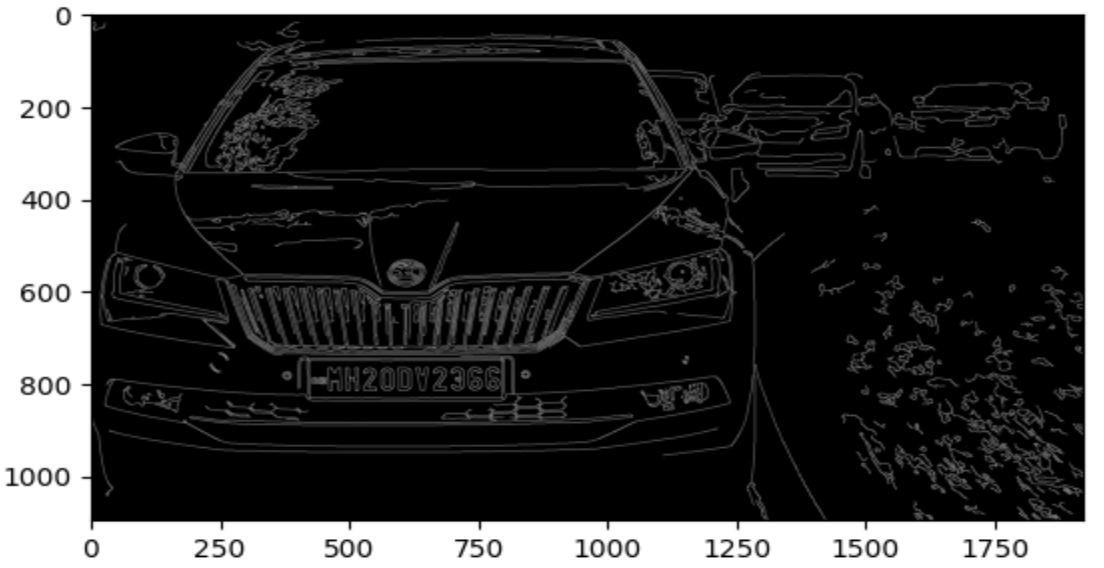
Python

Now insert the image name along with .format for processing the image, and we will get a converted gray image of the given input

```
img = cv2.imread('4.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
plt.show()
```
✓ 1.4s

After getting the gray image it process the image and noise reeduction is applied and we get bilateral filtered image , and the resulted image is shown below,

```
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))
plt.show()
```
✓ 1.5s



Imutils is a python library which contains series of convenience functions to make basic image processing functions such as translation rotation, resizing and displaying matplotlib images easier with OpenCV and both Python 2.7 and python 3

By this imutils, finds the contours from the edged image i.e. it fixes the license plate co-ordinates

```
import imutils
```

```
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]
```

+ Code + Markdown

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

And those fixed co-ordinates are stored in an array in matrix form as shown in below

```
location
```

Python

```
array([[[421, 753]],

       [[421, 832]],

       [[794, 831]],

       [[791, 750]]], dtype=int32)
```

Then it masks the unwanted area in the image and just crops the license plate as shown below

```
(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]
```

✓ 0.1s

```
plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
plt.show()
```

✓ 0.8s

After the license plate is cropped by using easyocr displays the coordinates of the cropped image and also it recognize characters in the license plate

```
import easyocr
reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
print(result)
```

✓ 6.1s Python

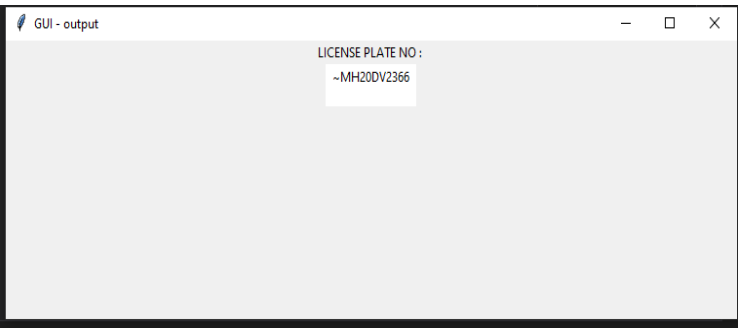
```
CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.

[([[0, 4], [376, 4], [376, 78], [0, 78]], '"HHZODV2366', 0.3316291923945292 6)]
```

Finally the characters in the image is converted into string and displayed in the GUI which is created through tkinter. And the string is displayed as shown below,

```
plate=pytesseract.image_to_string(cropped_image,lang='eng')
print("vechile plate number = ",plate)
from tkinter import *
root=Tk()
root.geometry("750x250")
root.title("GUI - output")
label=Label(root,text="LICENSE PLATE NO : ")
label.pack()
our_msg=plate
messageVar=Message(root,text=our_msg,width="400")
messageVar.config(bg='white')
messageVar.pack()
root.mainloop()
```

7.8s

```
vechile plate number =  ~MH20DV2366
```



Project input:

Project output:

CONCLUSION

This project performs some tasks. The first task is to input an image of the car and this will given by the user. When the image is fed the image is enhanced in quality. The enhancement is done in the resolution and the image is constraint to a fixed image frame size.

After the enhancement the image is processed to segment the number plate from the full picture based on the mathematical model of the rectangle. The segmented plate is shown in a new window. The enhanced segmented plate is then processed for OCR or Optical Character Recognition to segment all the characters in the picture in the form of Text and then it can be stored in a database or can be displayed as in this prototype.

The project is designed so that we can understand the technology used in now-a-days Automatic license plate systems and OCR systems used in most of the developed countries like Germany, France, Singapore, Japan, etc.

It is seen that security forces all over the world face problem to locate or register vehicle number to track any culprit. It is also seen that technology can greatly help us in this situation by solving it.

REFERENCES

- Digital image processing - Wikipedia
- (PDF) Proposal for Automatic License and Number Plate Recognition System for Vehicle Identification (researchgate.net)
- https://en.wikipedia.org/wiki/Automatic_number-plate_recognition
- https://docs.opencv.org/4.x/d9/df8/tutorial_root.html
- https://numpy.org/learn
- Python For Beginners | Python.org
- pytesseract · PyPI