**LICENSE PLATE RECOGNITION OF CARS**

**TABLE OF CONTENTS**

1. Abstract 3
2. Introduction 4
3. Technologies used 5
4. Modules Information 6
5. Steps involved 7
6. Data Flow Diagram 8
7. Test Case 9
8. Design with screenshots 10
9. Merits 18
10. Conclusion 19

**ABSTRACT**

License Plate Recognition (LPR) is a technology that uses optical character recognition on images to read vehicle registration plates .It can use existing cctv cameras for the surveillance.It can be used for Smart Policing - reducing the time of investigation.

License Plate Recognition (LPR) is a type of technology mainly software that enables computer systems to read automatically the registration number(license number) of vehicles from digital pictures.

License Plate Recognition System uses the concept of Optical character Recognition (OCR) to read the characters of a vehicle license plate.In other words ,LPR takes the image of a vehicle as input and outputs the characters written on its license plate.

Many applications can be implemented by using this system, such as security, highway speed detection, violation of light, identification of handwritten text, discovery of stolen cars, automatic fee collection systems.

**INTRODUCTION**

At present, the use of vehicles is increasing throughout the country. All of these vehicles have a unique vehicle identification number as their main identifier. The ID is actually in the license number that refers to a legal license to participate in the public movement. Each vehicle in the world must have its own number plate that must be installed on its body (at least on the back). They need to Identify the vehicles are increasing in parallel with the number of vehicles.

This identification system helps with safety, automatic switching systems, highway speed detection, light detection, stolen vehicle detection, and human and non-human loss collection systems. The auto license plate recognizing system replaces the manual license plate number writing process in the computer system.

Concern about these systems have centered on privacy fears of government tracking citizens movements,misidentification and high error rates. Critics have described it as a form of mass surveillance.

**TECHNOLOGIES USED**

* **OS - WINDOWS 10**:

Windows 10 is a series of operating systems developed by Microsoft and released as part of its Windows NT family of operating systems.

* **IDE - PYCHARM:**

PyCharm is an integrated development environment used

in computer programming, specifically for the Python language.

* **BACK END - PYTHON :**

Python is an interrupted high-level and general-purpose

programming language. It is meant to be easily and reliable

language.

**MODULE’S INFORMATION**

**PYTHON MODULES** :

opencv ,pytesseract,imutils,xlsxwriter

* **OPENCV**:

OpenCV (Open Source Computer Vision Library) is an open source

computer vision and machine learning software library. OpenCV was

built to provide a common infrastructure for computer vision

applications and to accelerate the use of machine perception in the

commercial products.

* **PYTESSERACT** :

Python-tesserac**t** is an optical character recognition (OCR) tool for

python. That is, it will recognize and “read” the text embedded in

images. Python-tesseract is a wrapper for Google's Tesseract-OCR

Engine.

* **IMUTILS :**

Imutils are a series of convenience functions to make basic image

processing functions such as translation, rotation, resizing,

skeletonization, and displaying Matplotlib images easier with OpenCV.

* **XLSXWRITER:**

XlsxWriter is a Python module that can be used to write text, numbers,

formulas and hyperlinks to multiple worksheets in an Excel 2007+

XLSX file.

# Steps involved : in License Plate Recognition

1. ****License Plate Detection:****

The first step is to detect the License plate from the car. We will use the contour option in OpenCV to detect for rectangular objects to find the number plate. The accuracy can be improved if we know the exact size, color and approximate location of the number plate.

1. ****Character Segmentation:****

Once we have detected the License Plate we have to crop it out and save it as a new image. Again this can be done easily using OpenCV.

1. ****Character Recognition:****

Now, the new image that we obtained in the previous step is sure to have some characters (Numbers/Alphabets) written on it. So, we can perform OCR (Optical Character Recognition) on it to detect the number.

1. **Excel Automation :**

The obtained result from the python code is directly entered into an excel sheet.

**DATA FLOW DIAGRAM**

**OUTPUT SAVED IN AN EXCEL SHEET**

**CHARACTER RECOGNITION**

**CHARACTER SEGMENTATION**

**IMAGE PROCESSING**

**UPLOAD IMAGE**

ADMIN

**OUTPUT LICENSE PLATE NUMBER**

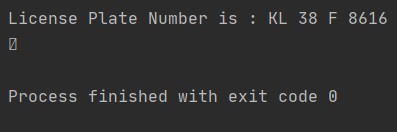
**LICENSE PLATE DETECTION**

**TEST CASES**

**INPUT** :



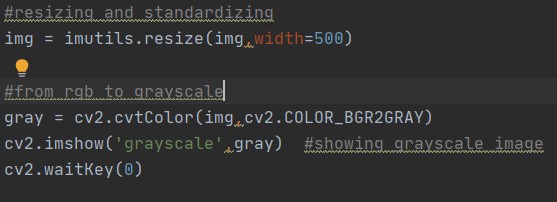
**OUTPUT** :



**DESIGN WITH SCREENSHORTS**

****Step 1 :**** ****Resize the image to the required size and then grayscale it****.

The code for the same is given below

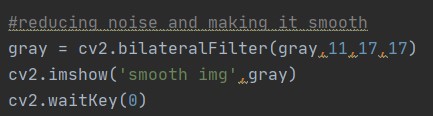


Resizing we help us to avoid any problems with bigger resolution images, make sure the number plate still remains in the frame after resizing. Gray scaling is common in all image processing steps. This speeds up other following process sine we no longer have to deal with the color details when processing an image. The image would be transformed something like this when this step is done



****Step 2 :****

Every image will have useful and useless information, in this case for us only the license plate is the useful information the rest are pretty much useless for our program. This useless information is called noise. Normally ****using a bilateral filter (Blurring) will remove the unwanted details from an image****.

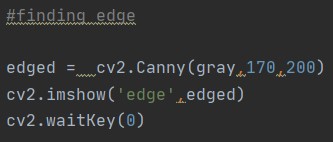


You can increase the sigma color and sigma space from 15 to higher values to blur out more background information, but be careful that the useful part does not get blurred. The output image is shown below, as you can see the background details are blurred in this image. This way we can avoid the program from concentrating on these regions later.



****Step 3:****

The next step is interesting where we perform ****edge detection****. There are many ways to do it, the most easy and popular way is to use the ****canny edge method from OpenCV.****

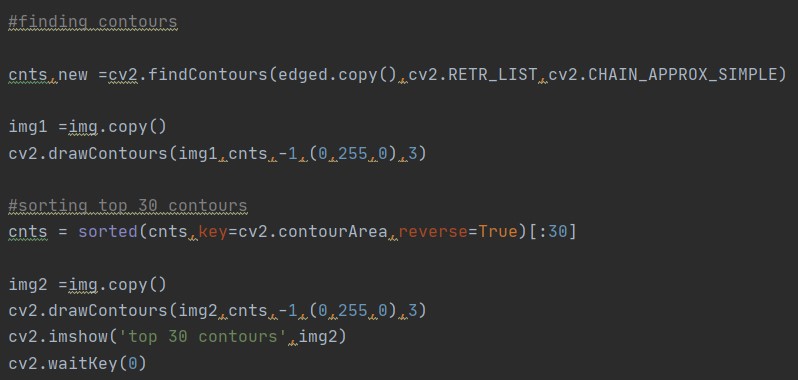
********

 Only the edges that have an intensity gradient more than the minimum threshold value and less than the maximum threshold value will be displayed. The resulting image is shown below



****Step 4:****

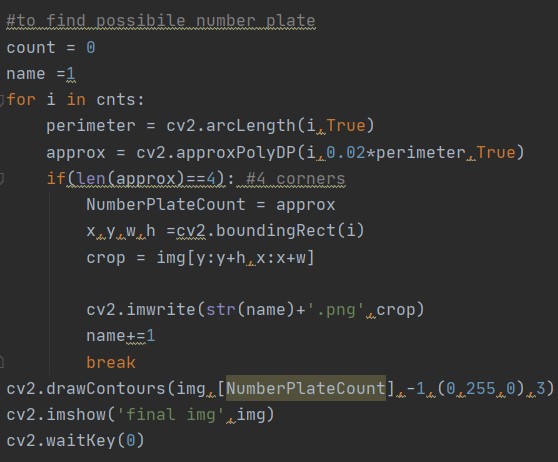
Now we can start ****looking for contours on our image****



****Once the counters have been detected we sort them from big to small**** and consider only the first 10 results ignoring the others. In our image the counter could be anything that has a closed surface but of all the obtained results the license plate number will also be there since it is also a closed surface.

********

To filter the license plate image among the obtained results, we will loop though all the results and check which has a rectangle shape contour with four sides and closed figure. Since a license plate would definitely be a rectangle four sided figure.

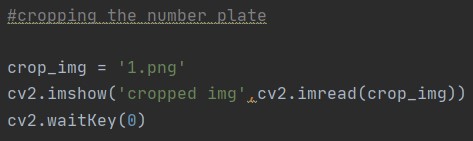
********

****RESULT :****

********

****Step 5:****

The next step in ****Number Plate Recognition**** is to segment the license plate out of the image by ****cropping it and saving it as a new image.****We can then use this image to detect the character in it.

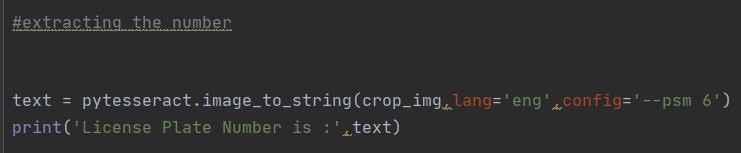


**The resulting image is shown below.**

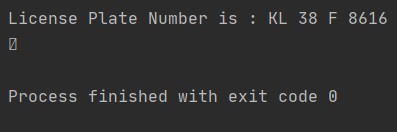
**Annotation 2020-12-09 190055**

****Step 6:****

The Next step in this ****Number Plate Recognition**** is to actually ****read the number plate information from the segmented image****. We will use the***pytesseract*** package to read characters from image.

****

**RESULT :**

****

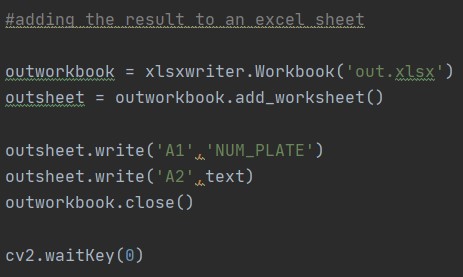
As you can see the original image had the number

“**KL 38 F 8616**”

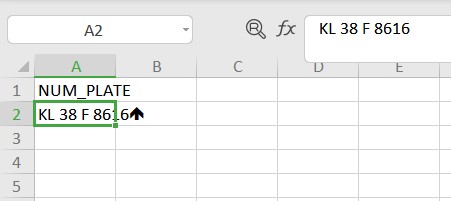
on it and our program has detected it printed the same value.

****Step 7:** Excel Automation**

**The final step is to add the License Plate Number obtained into an excel sheet using ***xlsxwriter*** module for future use.**

****

****RESULT** :**

********

**MERITS**

* **Reduces investigation time**
* **No hardware required**
* **Easy to implement**
* **No need of extra person to control it**
* **Details get stored for future use.**

**FUTURE ENHANCEMENTS**

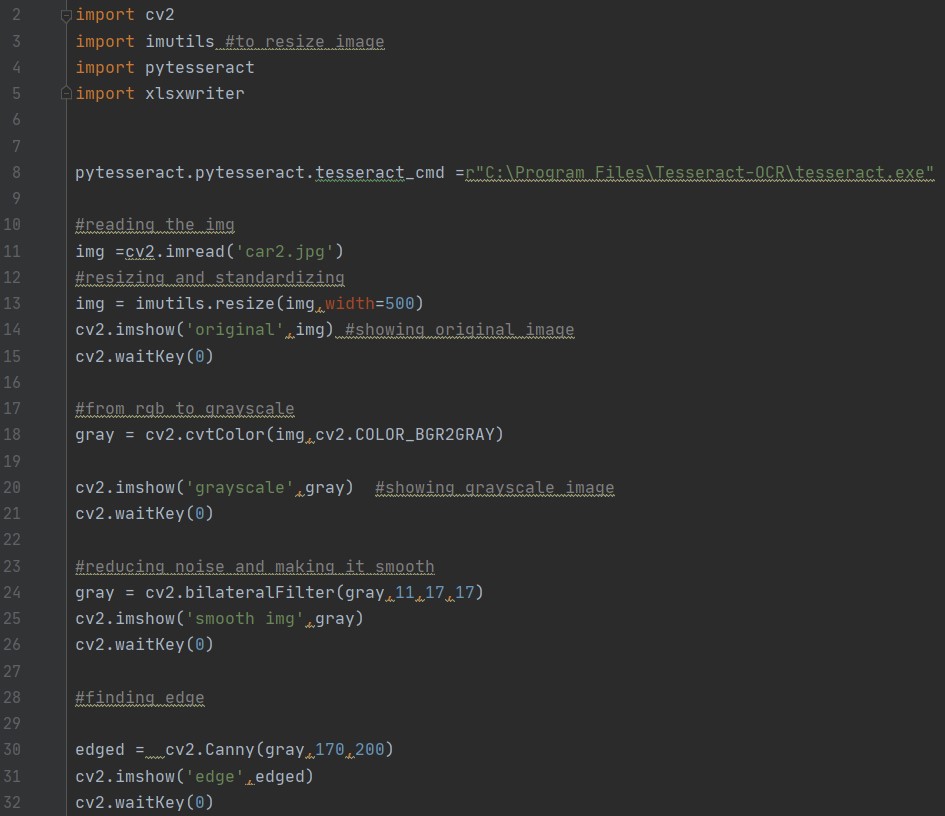
1. The project currently works over still captured images only, and can be modified in future to be implemented to extract license plate information over live video feeds.
2. Efficiency of the project can be increased by improving the character segmentation algorithm so it can be applicable to various types of car’s images.
3. Image Processing speed can be increased by installing faster processors.
4. The project can be implemented with Raspberry-Pie so as to use it for real life conditions.
5. Project currently don’t have a GUI but it can be made much more user friendly and easily navigable by using many other modules.

**CONCLUSION**

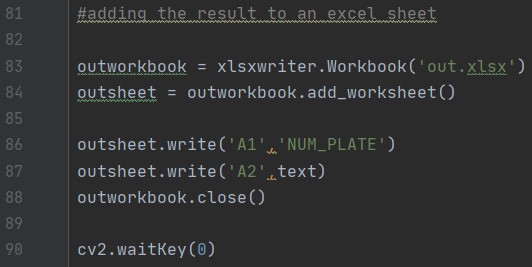
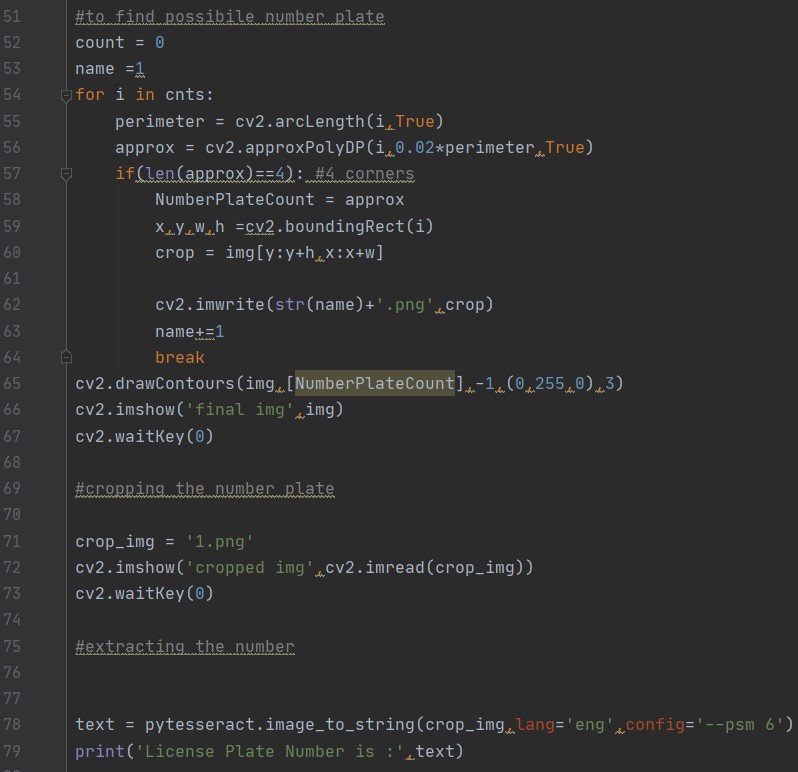
Without being said, it is to be remembered that the results from this method will not be accurate****. The accuracy depends on the clarity of image, orientation, light exposure etc****. To get better results we can try implementing Machine learning algorithms along with this.

Most of the times of the image quality and orientation is correct, the program was able to identify the license plate and read the number from it.

**SOURCE CODE :**

****

****

****