IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# Automatic Number Plate Detection System

**Raghavendra. Puri,**
**School of computer science and engineering,**
**lovely professional university, Jalandhar, Panjab,**

**Harneeth Kaur**
**School of computer science and Engineering,**
**Lovely professional university, Jalandhar, Panjab, India,**
**India**

**Challagundla. Maruthi Manohar School of computer science and engineering,**

**lovely professional university, Jalandhar, Panjab, India**

**Akhil Ganta School of computer science and engineering,**

**lovely professional university, Jalandhar, Panjab, India**

**Basani Chaitanya Reddy School of computer science and engineering,**

**lovely professional university, Jalandhar, Panjab, India**

**Poonam Uday Kumar School of computer science and engineering,**
**lovely professional university, Jalandhar, Panjab, India**

**Abstract** – Automatic number-plate recognition is a technology that makes use of optical character recognition on images to examine automobile registration plates with the use of Tesseract OCR Engine. It may be used on present closed-circuit television, road-rule enforcement cameras, or cameras in particular designed for the task. Using Selenium internet driver, a wide variety of plates recognized is parsed to the authorities' website Vahan. nic.in conjunction with the solved captcha and the automobile, information may be accessed for further Inference and evaluation. The crawled data is transformed into structured and unstructured facts and saved in Firebase and MySQL for data evaluation and stay dashboard. Through the dashboard, the notification triggers can be set if an automobile defaults to any of the rules, and an SMS may be dispatched to the cellular tele cell smartphone of the authority.

A License Plate Recognition system is a real-time embedded environment that robotically recognizes the registration code of vehicles. Many programs are starting from complicated safety structures to common areas and from parking admission to city visitor’s control. Automatic registration code recognition (ALPR) has complicated traits because of numerous results which include light and speed. Most of the registration code recognition structures are built using proprietary tools like MATLAB which takes a protracted manner and time and additionally does have numerous limitations. This concept offers an alternative approach to imposing ALPR systems through the usage of Free Software such as Python and the Open Computer Vision Library.

**Keywords: -** ANIRP automatic number plate detection system, OpenCV, Google COLAB

# Introduction

A Automatic Number Plate Recognition or ANPR is an era that makes use of pattern recognition to 'read' vehicle number plates. In easy phrases ANPR cameras 'photograph' the quantity plates of the motors that pass them. This 'photograph' is then fed into a computer device to discover information about the vehicle itself. ANPR includes cameras related to a computer. As an automobile passes, ANPR 'reads' Vehicle Registration Marks – more usually called quantity plates - from digital images, captured thru cameras placed both in a cell unit, in-constructed in visitors' motors or thru Closed-Circuit Television (CCTV). The virtual picture is transformed into data, that is processed thru the ANPR device. We proposed a technique especially primarily based totally on area detection, OCR operation, and Finding Rectangles in a Vehicle Image.

Owning a car nowadays isn't simply an image of luxury however has emerged as a necessity. However, thinking about motors, any catastrophic scenario can take place. Therefore there's usually a pressing want to set up suitable measures to grow the safety, safety, in addition, to screening the vehicles to keep away from any mishap. It could assist us with inside the-conditions inclusive of Instantaneously gaining vehicle information through the use of picture processing. Allowing a corporation to discover the place of its vehicles. Automatically notify the person if there are site visitor violations registered to the vehicle. One such degree is using a vehicle monitoring system the use of the GPS (Global Position System). Such a monitoring system consists of a mechanized tool this is ready in a vehicle. Using software program gift at an operational bse, enables song the place of the vehicle. This base station is used for tracking purposes. It is observed with the aid of using maps inclusive of Google maps, Here maps, Bing maps, etc. for the illustration of the place.

ANPR may be used to keep the photographs captured through the cameras in addition to the textual content from the license plate, with a few configurable to keep an image of the driver. Systems generally use infrared lights to permit the camera to take the image at any time of the day.

. ANPR technology tends to be region-specific, owing to plate variation from place to place.
The acquisition of digital photos generally suffers from unwanted camera shakes and because of risky random camera motions. Hence photo enhancement algorithms are required to put off those undesirable camera shakes. . Python is used as the primary programming language. We have related to the http://vahan.nic.in with the ANPR gadget to extract all of the automobile and proprietor details. We extract the information and store the information in JSON layout for additional processing and analysis.
The acquisition of digital images usually suffers from undesirable camera shakes and unstable random camera motions. Hence image enhancement algorithms are required to remove these unwanted camera shakes. Python is used as the main programming language.

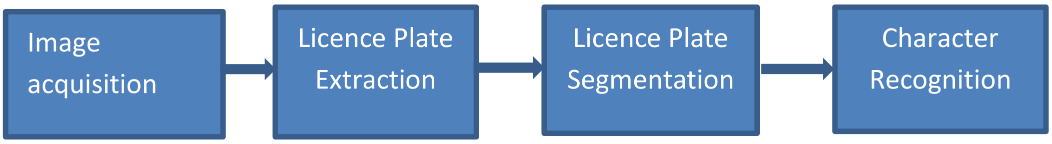


Fig 1. **An Efficient Approach for Vehicle Number Plate Recognition**

## Related work

IMS Engineering collage, et al., (2018) project is to reduce criminal activity like stolen vehicle, road traffic monitoring that involve the use of motor vehicles. In this project, we will be processing the image of the vehicle such that the vehicle number from the image will be extracted. Using that extracted image i.e., the vehicle number we will be comparing that number from our database to check whether that vehicle belongs to the correct person or not and we will also be checking whether the documents belonging to that vehicle are complete or not. Using that detail, we will be giving challan if they are suspected done by using MATLAB for image processing.[1]

1PG Scholar, et al., (2019), accurate number plate detection and real-time identification using raspberry pi. recognizing the characters in the vehicle number plate and the classified characters are used further use in many traffic, security, and access control applications. Accurate car plate recognition (ALPR) has complex features due to diverse effects like light and speed. Most commonly LPR, which comes under image processing uses proprietary tools like MATLAB.[2]

Bharati Vidyapeeth (DU) College of Engineering Pune, et al., (2020), To tackle the problem of illegal encroachment of roads and help in arresting the drastic slide of carbon emissions, this paper aims to carry out the use of a UAV-based system to identify license plates of vehicles in illegal parking spaces. There are numerous closed-circuit cameras for indoor and outdoor lots. However, these systems use hardware that is steeply priced and is implemented using bespoke software. The main goal of this project is to develop an autonomous license plate surveillance system using low-cost hardware and open-source software.[3]

Srinivas Institute of Technology, Mangalore, Karnataka, et al., (2021), Vehicle Number Plate Detection Using CNN Vehicle number plate detection plays an important role in this busy world, due to the heavy rise in vehicles day by day. Passing the tollgates without paying the money, stealing vehicles, breaking traffic rules, and coming into restricted spaces also are increasing linearly, thus, to block these situations vehicle number plate detection is proposed. Among the important process steps such as detection of the number plate, segmentation of characters, and recognition of each character, segmentation plays an important role. To avoid problems like unwanted brightness, and tilt that degrades the segmentation which in turn affects the recognition accuracy, numerous algorithms are developed for this work. We proposed an approach Vehicle Number Plate Detection using the CNN model which is a part of Deep Learning.[4]

RV College of Engineering, Bangalore India, et al., (2020), Vehicle Number Plate Detection using Histogram Analysis: In most recent decades, the quantity of vehicles has expanded rapidly. With this expansion, monitoring the vehicle for law authorization is hard. Tag Recognition is utilized progressively these days for programmed cost assortment, keeping up traffic exercises, and law requirements. There are many strategies with their own preferences and burdens, proposed for plate recognition. The confinement of number plates is the fundamental advancement in license plate detection. The methodology referenced here is a histogram-based methodology.[5]

Aadesh Institute of Technology, Gharuan, Punjab, et al., (2020). Automatic vehicle plate reorganization and detection by image processing and data mining approach Automatic Vehicle Plate Recognition (AVPR) is the extraction of vehicle license plate information from an image or sequence of images. Over the past thirty years, AVPR is becoming a challenging and interesting area of research. AVPR systems include a wide range of applications. Numerous real-world applications such as electronic toll collection, automatic parking management, access control, radar-based speed-control, border control, criminal pursuit, traffic law enforcement,[6] SV University, Tirupathi, et al., (2021), Morphological Operation based Vehicle Number Plate Detection, Segmenting the Region Of Interest (ROI) from an image can be used for Vehicle number Plate detection. In this work, a method based on a convolution neural network is proposed for Vehicle number Plate detection. The accuracy of the method is improved and compared with the existing method.[7]

Warsaw University of Technology, et al., (2021), Institute of Control and Industrial Electronics EC Joint Research Centre, Institute of Environment and Sustainability Automatic car number plate detection using morphological image processing the paper describes a detection method of car number plates on digital images. The method inputs the gray-tone image of a car and extracts from it the characters on the number plate. The proposed method makes use of the morphological operators: morphological reconstruction, filters based on it, and top-hat operators.[8]

University of Asia Pacific, Bangladesh, et al., (2021), Vehicle Speed Estimation using Image Processing, This paper aims at determining vehicle speed which is necessary for traffic surveillance systems. These systems are very much useful to monitor and manage various traffic conditions such as traffic management, prevention of accidents, also secure transportation. This paper describes a comprehensive approach to localizing target vehicles in the video under various environmental conditions.[9]

KLE Technological University. et al., (2021), Vehicle Detection, Tracking, and Counting for Traffic Surveillance using Digital Image Processing: Traffic Surveillance system is being more important with the increasing number of vehicles. Much better ways for traffic analysis are also developed. Traffic analysis is analysing the cloud of vehicles in defined places for specific intervals of time and the vehicle classes. Now a day most of the people involved in sensors are used to detect vehicles. Even though these systems are highly effective and mature and not very budget friendly. These systems need high maintenance and periodic calibration; so, it causes highly smarter computer vision-based systems for traffic surveillance.[10]

# Methodology:-

The total report is composed in the following way:
Literature Survey- The literature survey is an effective tool to research about the current developments in the related field and their drawbacks, in order to incorporate a better mechanism in the proposed system.

Theoretical Background – This contains information about the underlying technologies and algorithms that have been made use of in the architecture and design of the proposed system.

System Requirement Specification – This chapter deals with the various functional and nonfunctional requirements that are to be implemented in the proposed system. It also encloses both hardware and software requirement specifications that are needed to run the software.

System Design – This chapter deals with the architecture and design of the system. Various diagrams have been included that describe the working of the system in a way that canbe understood better.

Implementation – The algorithms that form the proposed system have been listed, along with important modules of code.

Results and testing – This chapter discusses the obtained results and the outcome of the testing process.

Conclusion and future work – This chapter concludes the discussion of the work done so far. Future enhancements that can be made to the work are also discussed in this chapter.

References – This chapter lists the various websites and books that have been referred to, in the making of this project and the project report.

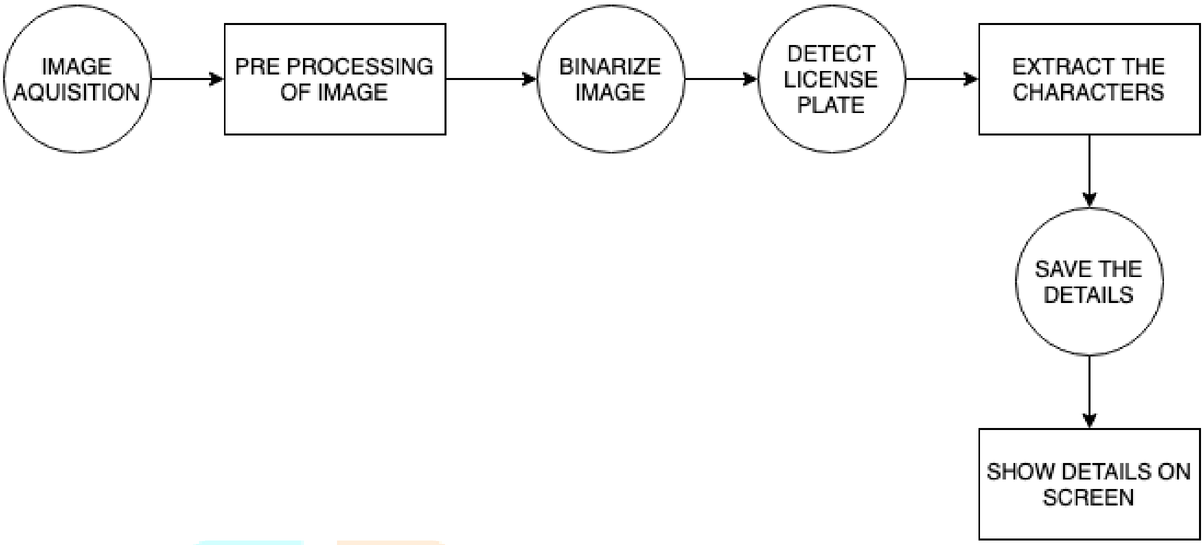


FIG: **DATA FLOW DIAGRAM**

## Applications: -

Through this project it is possible to recognize Vehicle registration numbers through digital image processing. From this system we have effortlessly obtained the various results such as

• Whether the vehicle which is registered is blacklisted or not.

• This also enables one single user to effectively monitor the traffic, and can easily locate the traffic violated vehicle.

• The data can be easily stored and transferred which makes the system more efficient.

The system has been designed using a modular approach which allows easy upgrading and/orsubstituting of various sub-modules thus making it potentially suitable for a large range of vision applications. The performance of the system makes it a valid choice among its competitors especially in those situations when the cost of the application has to be maintainedat reasonable levels. Furthermore, the modular architecture makes it extremely flexible and versatile.

The earlier methodologies which have been implemented have not been as accurate and efficient as the designed Recognition system, this is because of the implementation of digital Image Processing which gives an accuracy of 90% under normal conditions

This Project is based on automatic vehicle license plate recognition, in which it is observed thatthe existing techniques don't pay much attention towards improving the system's efficiency in terms of its power consumption. As the objective in our proposed design is to reduce power consumption of the system, with the successful implementation of the same it will play a very important role in traffic management and security systems such as automobile theft prevention, parking lot management etc. implementations of the software algorithm have shown promisingresults.

The system can be made more robust if high precision cameras can be used to increase overallaccuracy if this system is implemented in real time applications. Also, a sensor can be designedto allow the camera to capture the image only when required to save power.

## Proposed system

Automatic Number Plate Recognition is the use of an efficient OCR engine like Pytesseract in conjunction with the most important and substantial libraries of OpenCV for photo processing. As we've visible to this point ANPR covers an approach to a maximum of the troubles we've posed. We would love to dig a chunk deeper now and spotlight the scope of the assignment and the volume to which we are able to push the boundaries. The essential difficulty this is commonly diagnosed in terms of variety plate detection is the noise this is introduced to the photo with inside the procedure of taking pictures the photo or because of the surroundings around, taking that into attention we are able to say that the use of our system, we are able to put in force it in all environments, be it rain or maybe with inside the dark. Usually, while any new system is proposed to viable clients, their essential difficulty is the addition of the latest functions into their present system. Keeping this in thoughts we are able to say for certain that our system may be incorporated to the pre-present infrastructure of maximum clients.

Using a web crawler, a wide variety of plates identified is parsed to the government internet site Vahan.nic.in together with the solved captcha and the vehicle info may be accessed for in addition Inference and analysis. Also, show off the vulnerabilities inside the protection of the government websites and privacy troubles in the government internet site. Also, offer analytics and answer at the extracted data.

## Implementation

### ALGORITHMS

This section incorporates the algorithms which can be utilized by the system. The device uses those algorithms with the intention to fulfill the various functional and non-functional requirements of the proposed scheme.

### Algorithm to Recognize the Number Plates

The collection of strategies related to Number plate reputation is given below. The record add procedure is initiated with the aid of using the records proprietor entity.

**Input:** Uploading the image file from the camera

**Output:** Vehicle number plate in characters

1) Read the original image or capture the image

2) Resize the image

3) Convert it to grayscale.

4) Apply Bilateral Filter. What is a bilateral filter? A bilateral filter is a non-linear, edge-preserving, and noise-reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels.

5) Identify and store the Canny edges. What are Canny edges? The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

6) Find the contours from the edges detected and sort the top 30contours.

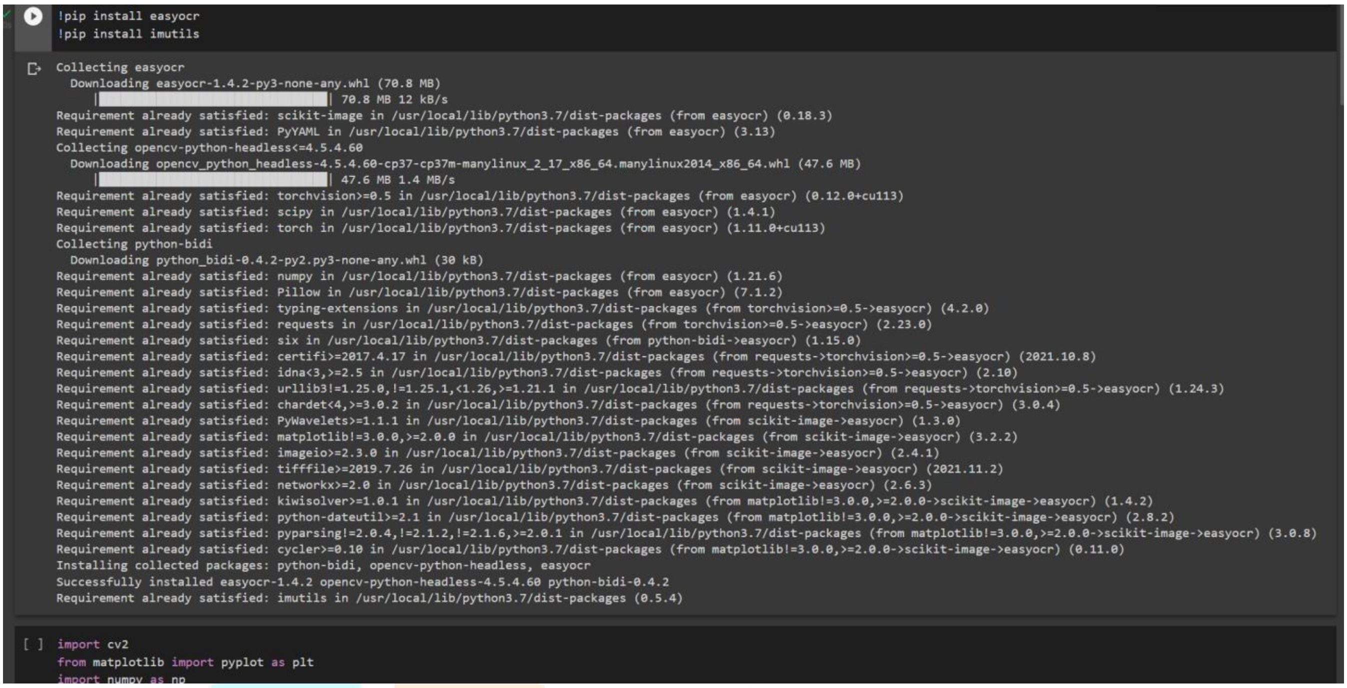
7) Get the perimeter of each contour and select those with 4corners.

8) Mask all other parts of the image and show the final image.

9) Read the text using TesseractOCR

10) Standardize the text to Indian vehicle number plate format

Code



```
!pip install easyocr
!pip install imutils
```

```
Collecting easyocr
  Downloading easyocr-1.4.2-py3-none-any.whl (70.8 MB)
     |████████████████████████████████| 70.8 MB 12 kB/s
Requirement already satisfied: scikit-image in /usr/local/lib/python3.7/dist-packages (from easyocr) (0.18.3)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-packages (from easyocr) (3.13)
Collecting opencv-python-headless<=4.5.4.60
  Downloading opencv_python_headless-4.5.4.60-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (47.6 MB)
     |████████████████████████████████| 47.6 MB 1.4 MB/s
Requirement already satisfied: torchvision>=0.5 in /usr/local/lib/python3.7/dist-packages (from easyocr) (0.12.0+cu113)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.4.1)
Requirement already satisfied: torch in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.11.0+cu113)
Collecting python-bidi
  Downloading python_bidi-0.4.2-py2.py3-none-any.whl (30 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.21.6)
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from easyocr) (7.1.2)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from torchvision>=0.5->easyocr) (4.2.0)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from torchvision>=0.5->easyocr) (2.23.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from python-bidi->easyocr) (1.15.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->torchvision>=0.5->easyocr) (2021.10.8)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->torchvision>=0.5->easyocr) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests->torchvision>=0.5->easyocr) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->torchvision>=0.5->easyocr) (3.0.4)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (1.3.0)
Requirement already satisfied: matplotlib!=3.0.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (3.2.2)
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (2.4.1)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (2021.11.2)
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (2.6.3)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->easyocr) (1.4.2)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->easyocr) (2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->easyocr) (3.0.8)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-image->easyocr) (0.11.0)
Installing collected packages: python-bidi, opencv-python-headless, easyocr
Successfully installed easyocr-1.4.2 opencv-python-headless-4.5.4.60 python-bidi-0.4.2
Requirement already satisfied: imutils in /usr/local/lib/python3.7/dist-packages (0.5.4)
```

```
[ ] import cv2
    from matplotlib import pyplot as plt
    import numpy as np
```

Step:1

**EasyOCR**

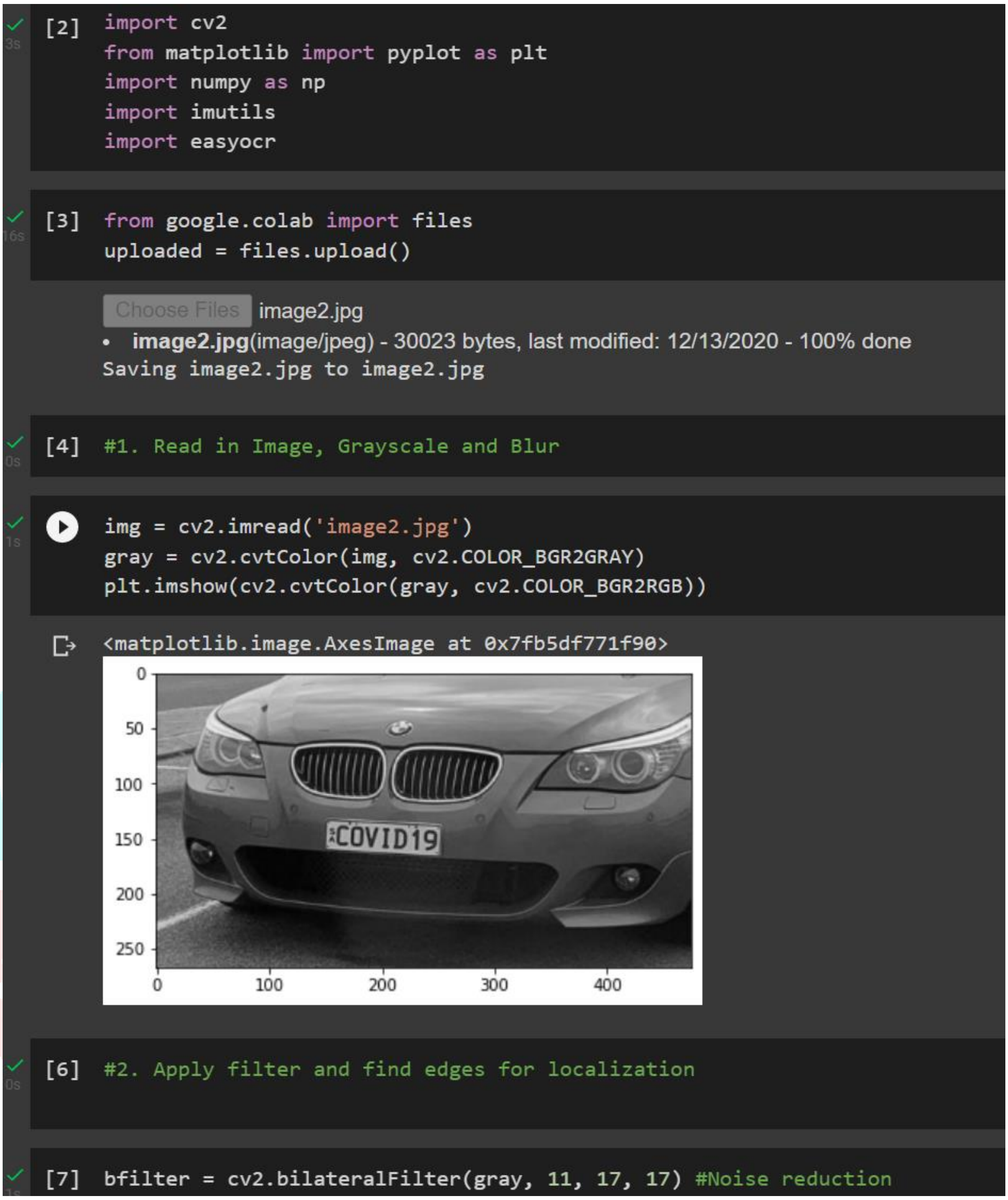
- EasyOCR is a Python package that allows computer vision developers to effortlessly perform Optical Character Recognition (OCR). When it comes to OCR, EasyOCR is by far the most straightforward way to apply Optical Character Recognition
- The EasyOCR package can be installed with a single pip command

**!pip install easyocr**

**IMUTILS**

- IMUTILS is Python Programming basic image processing functional package to do image translation, rotation, resizing, skeletonization, or blur amount detection, displaying Matplotlib images, sorting contours, detecting edges, and much easier with OpenCVand Python 3.
- imutils also check to find functions if you already have NumPy, SciPy, Matplotlib, and OpenCV installed.
- This is the command used to install the imutils package.

**!pip install imutils**



[2]

```
import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

[3]

```
from google.colab import files
uploaded = files.upload()
```

image2.jpg

- **image2.jpg**(image/jpeg) - 30023 bytes, last modified: 12/13/2020 - 100% done

```
Saving image2.jpg to image2.jpg
```

[4]

```
#1. Read in Image, Grayscale and Blur
```

```
img = cv2.imread('image2.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7fb5df771f90>
```

[6]

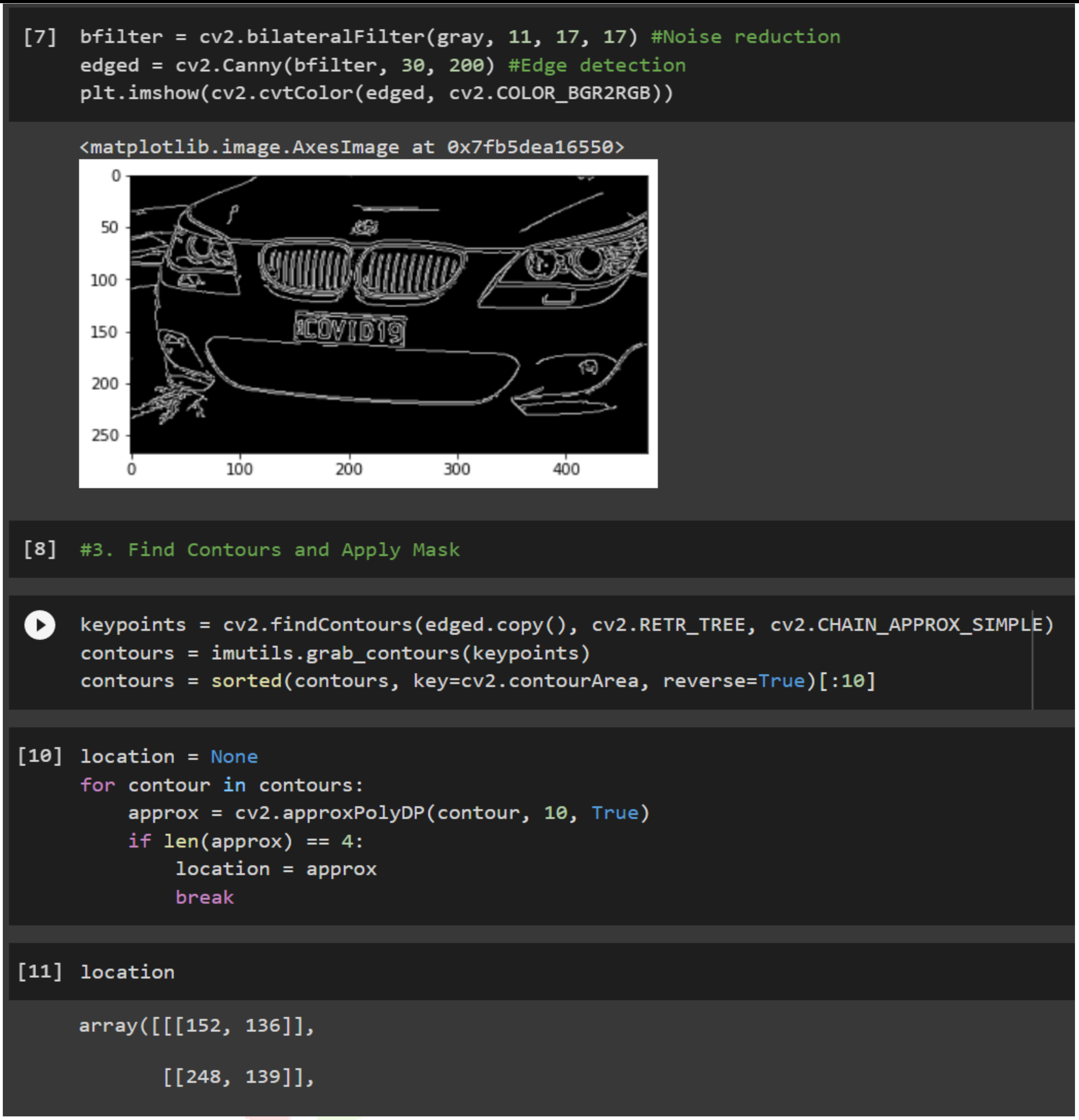
```
#2. Apply filter and find edges for localization
```

[7]

```
bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
```

Step:2

- From google collab we upload the files of the pictures of the vehicle taken and they are uploaded using the function files.upload.
- And we have to choose the file from the same directory.
- Using the function cv2.imread we read the image of the car.
- Using cv2.cvtColor we change the color of the image from BGR to GRAY.

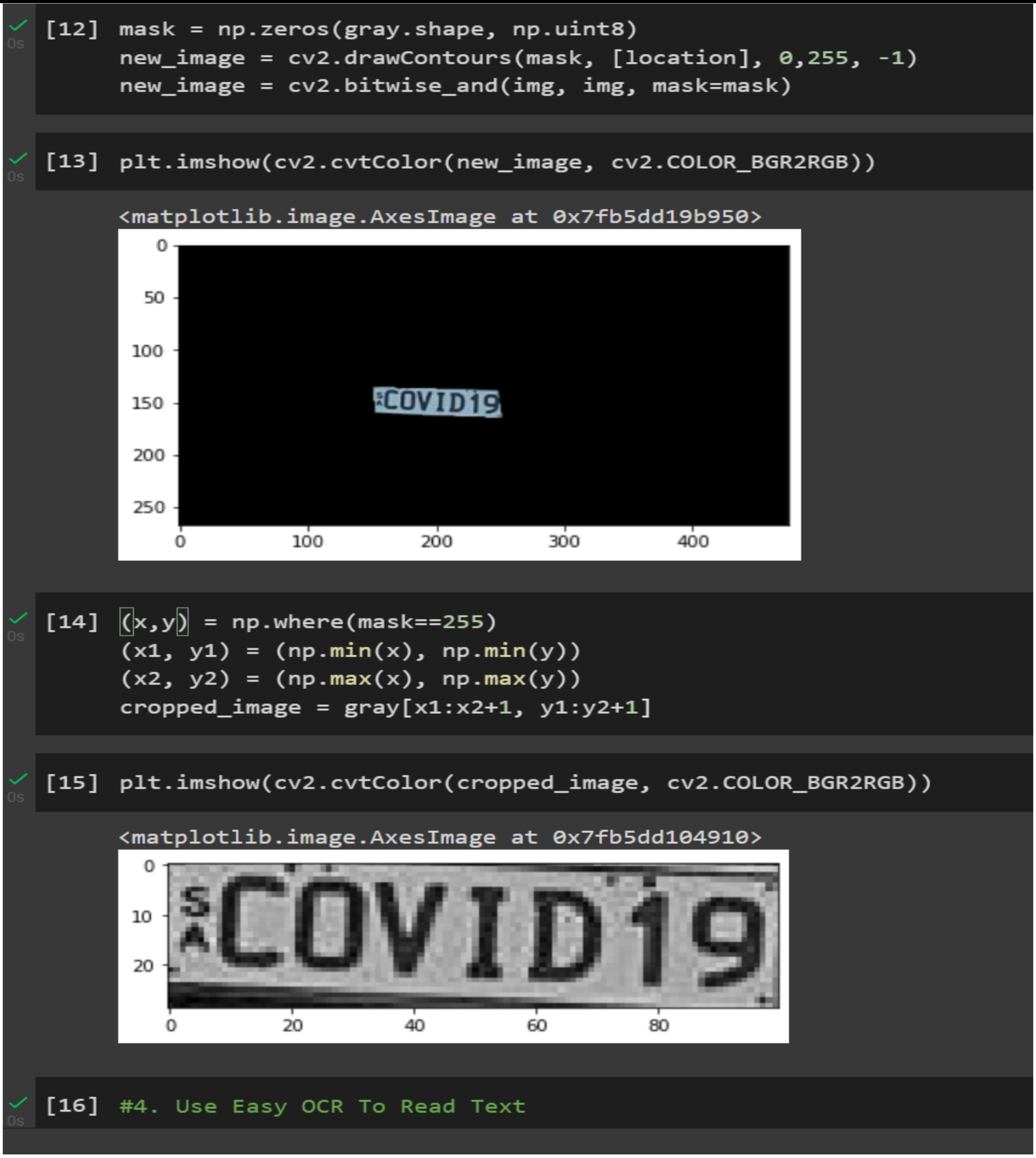


```
[7]  bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
     edged = cv2.Canny(bfilter, 30, 200) #Edge detection
     plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))

     <matplotlib.image.AxesImage at 0x7fb5dea16550>
```

```
[8]  #3. Find Contours and Apply Mask

     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]

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

[11] location

     array([[[152, 136]],

            [[248, 139]],
```

Step:3

- cv2.bilateralFilter is used for noise reduction and cv2.canny is used for edge detection.
- By applying these two filters we can clearly see the edges and shape of the image.
- We can find the counters i.e. in the shape of the rectangle by using the function cv2.findcounters.
- After that we have to find the location of the counter using the function cv2.approxPolyDP, if the length of the counter is 4 then it will return the location of the counter.



```
[12] 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)
```

```
[13] plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))

     <matplotlib.image.AxesImage at 0x7fb5dd19b950>
```

```
[14] (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]
```

```
[15] plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))

     <matplotlib.image.AxesImage at 0x7fb5dd104910>
```

```
[16] #4. Use Easy OCR To Read Text
```

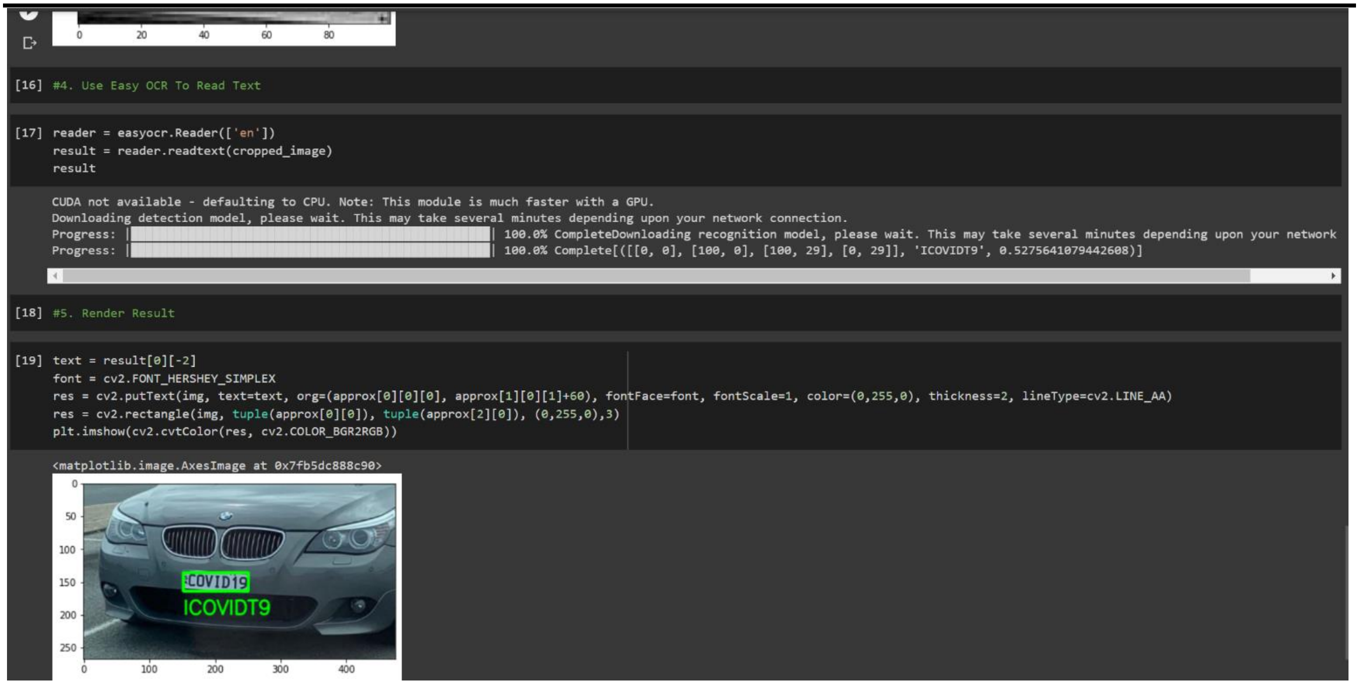
Step:4

## MASKING

- By using np. zero we mask the whole grayscale image.
- We retrieve the counter by using cv2.drawcounters bypassing the argumentsas location.

## EXTRACTING THE CHARACTERS

- Using the functions np.min and np.max of the mask we retrieve the number plate from the masked image, so we get the cropped image of the number plate.



```
[16] #4. Use Easy OCR To Read Text

[17] reader = easyocr.Reader(['en'])
     result = reader.readtext(cropped_image)
     result

     CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
     Downloading detection model, please wait. This may take several minutes depending upon your network connection.
     Progress: |██████████████████████████████████████████████████| 100.0% CompleteDownloading recognition model, please wait. This may take several minutes depending upon your network
     Progress: |██████████████████████████████████████████████████| 100.0% Complete[([[0, 0], [100, 0], [100, 29], [0, 29]], 'ICOVIDT9', 0.5275641079442608)]

[18] #5. Render Result

[19] 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.LINE_AA)
     res = cv2.rectangle(img, tuple(approx[0][0]), tuple(approx[2][0]), (0,255,0),3)
     plt.imshow(cv2.cvtColor(res, cv2.COLOR_BGR2RGB))

     <matplotlib.image.AxesImage at 0x7fb5dc888c90>
```

Step:5

### OCR

- We use EasyOCR to go and grab the text from that image.
- We get the result as the coordinates of the number plateand the characters in the form oftext.

### OVERLAY OUR DETECTION ON THE ORIGINAL IMAGE

- By using tuple the result is stored in the variable text i.e. text = result[0][-2]
- By using cv2.puttext ,cv2.rectangle we highlight the number plate by drawing a rectangle.
- Finally we can display the characters in the number plate on the vehicle.

## Advantages

To perform successful and efficient pre-processing on the raw RGB image

To exploit the high performance and effectiveness of OpenCV and Py-Tesseract framework to detect and recognize LP of vehicles, to improve our system reliability.

To correctly determine the number plate based on Indian Number plate Standards

To Successfully extract the information from the Government vehicle information database

To Show the security vulnerabilities on Vahan. nic.in

## Conclusion

Open CV is the one of the best tool which we are using for image processing which works satisfactorily for the wide variations in illumination conditions and different types of number plates commonly found in India . it is definitely a alternative for the current existing proprietary systems, even though there are known restrictions with high resolution to detect the plate using Open CV and python which is most easy to make changes.

# Reference

[1] Department of IT, IMS Engineering College, Ghaziabad, India 2018.

[2] 1PG Scholar, Department of ECE, Velammal Engineering College, Chennai-600066, 2Associate Professor, Department of ECE, Velammal Engineering College, Chennai-600066.

[3] Suraj Rasal1, Saksham Gupta2, Shashwat Pande3, Shubhankar Gupta4 1Assistant. Professor, Dept. of Computer Engineering, Bharati Vidyapeeth (DU) College of Engineering Pune, Maharashtra, India 2,3,4Student, Dept. of Computer Engineering, Bharati Vidyapeeth (DU) College of Engineering Pune, Maharashtra, India in 2020.

[4] Done by Anvitha K N*, Padmanayana, Abdul Shawwan Department of Computer Science, Srinivas Institute of Technology, Mangalore, Karnataka, India 2021.

[5] Done by Harish S E1, Keshav Priyadarshi2, Ms. Usha Padma3 1Telecommunication Engineering Student, RV College of Engineering 2Telecommunication Engineering Student, RV College of Engineering 3Assistant Professor, Department of Telecommunication Engineering, RV College of Engineering, Bangalore, India 2020.

[6] Done by Rangarajan Chaurasiya rajanchzya@outlook.com Aadesh Institute of Technology, Gharuan, Punjab Dr. Rajat Joshi errajatjoshi@gmail.com Aadesh Institute of Technology, Gharuan, Punjab.

[7] Done by N. Eswar 1M. Tech, Department of Electronics and Communication Engineering, SV University, Tirupathi, Dr. D. Gowri Shankar Reddy2 2Assistant Professor, Department of Electronics and Communication Engineering, SV University, Tirupath.

[8] Done by Warsaw University of Technology, Institute of Control and Industrial Electronics EC Joint Research Centre, Institute of Environment and Sustainability.

[9] Done by Fatima Afifah1, ∗ , Sharmin Nasrin1, Abdul Mukit1 1 University of Asia Pacific, 74/A, Green Rd, Dhaka 1215, Bangladesh Vehicle Speed Estimation using Image Processing.

[10] Done by Roopa Hittanagi1, Dr. Priyatamkumar2 1, 2Electronic and Communication Department, KLE Technological University. Vehicle Detection.