Project Report

On

CAR LICENSE DETECTION VIA OCR (OPTICAL CHARACTER RECOGNITION)

Submitted for the requirement of

Project

BACHELOR OF ENGINEERING
IN

COMPUTER SCIENCE ENGINEERING

SUBMITTED BY:
BHAVYA THATHERA
20BCS9528

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
CHANDIGARH UNIVERSITY, GHARUAN
November 2022

CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

BONAFIDE CERTIFICATE

Certified that this project report “Car License Detection via OCR (Optical Character Recognition)” is the Bonafide work of “BHAVYA THATHERA (20BCS9528)” who carried out the project work under our supervision.

SIGNATURE **SIGNATURE**

Supervisor **Head of Department**

Submitted for the project viva-voce examination held

on:

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

License Platform Detection is a computer technology that enables us to identify digital images on the platform automatically. Different operations are covered in this system, such as imaging, number pad locations, alphanumeric character truncation and OCR. The final objective of the system is to construct and create efficient image processing procedures and techniques to position a licensing platter on the Open Computer View Library picture. It was used and implemented the K-NN algorithm and python programming language. The technology can be used in different industries such as security, highway speed detection, lighting violations, manuscript documents, automatic charging system, etc.

Auto plate recognition is an integrated technology which identifies the auto license plate. Auto plate auto recognition. Multiple applications include complex safety systems, public spaces, parking and urban traffic control. Automatic Vehicle License Plate Recognition (AVLPR) has undesirable aspects because of many effects, such as light and speed.

This work presents an alternative technique to leverage free software for the implementation of AVLPR systems including Python and the Open Computer Vision (openCV).

TABLE OF CONTENT

Sr no.	Topic	Page No.
1	Use of Modern tools in design and analysis	5 - 6
2	Discussion and report/results analysis	7 - 8
3	Project management and Professionalcommunication (Presentation)	9 - 14
4	Attainment of stated outcomes	14
5	Certificate	15

Use of Modern tools in design and analysis

The Modern Tools used in this Project for Design and Analysis is Python. It is a Programming Language. It is used for web development (server-side), software development, mathematics, system scripting and many more.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

❖ There are Some Libraries Used Here in this Project:

OpenCV :

OpenCV is an open-source library for the computer vision. It provides the facility to the machine to recognize the faces or objects. In this tutorial we will learn the concept of OpenCV using the Python programming language.
There are two common ways to identify the images:

1. Grayscale

Grayscale images are those images which contain only two colors black and white. The contrast measurement of intensity is black treated as the weakest intensity, and white as the strongest intensity. When we use the grayscale image, the computer assigns each pixel value based on its level of darkness.

2. RGB

An RGB is a combination of the red, green, blue color which together makes a new color. The computer retrieves that value from each pixel and puts the results in an array to be interpreted.

Here, In this Project, We are Using GreyScale Technique to identify whether the Space if empty or filled by using pixel approach.

Easy-ORC:

Easy OCR is a font-dependent printed character reader based on a template matching algorithm. It has been designed to read any kind of short text (part numbers, serial numbers, expiry dates, manufacturing dates, lot codes, …) printed on labels or directly on parts.

Numpy:

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely.
NumPy stands for Numerical Python.

CVZone:

Cvzone is the library that develops a bridge between Arduino and python. With the help of the SerialObject module in Cvzone we can connect the arduino port with python as well as send data to arduino and can link any python code withit.

Software Requirements:

- Windows 7 or higher.
- SQL 2008 or higher.
- Visual studio 2010, Python Environment.

Hardware Components:

- Processor – i3(Min).
- Hard Disk – 50 GB.
- Memory – 3 GB RAM.

Discussion and report/results analysis

Overview:

This project which we have made with the help of AI gives us a brief understanding of OCR (Optical Character Recognition). This helps us detect the number of plates of the car and at the same time this prints out the license number and the name of the owner of the vehicle. This will have a huge impact on the upcoming ai generations because we are moving towards a paper-free future and this will surely reduce the paperwork and make things easy and simple for the user with its easy-to-use interface. To operate this program the user should have a slight knowledge of the computer and inputting the image in the program.

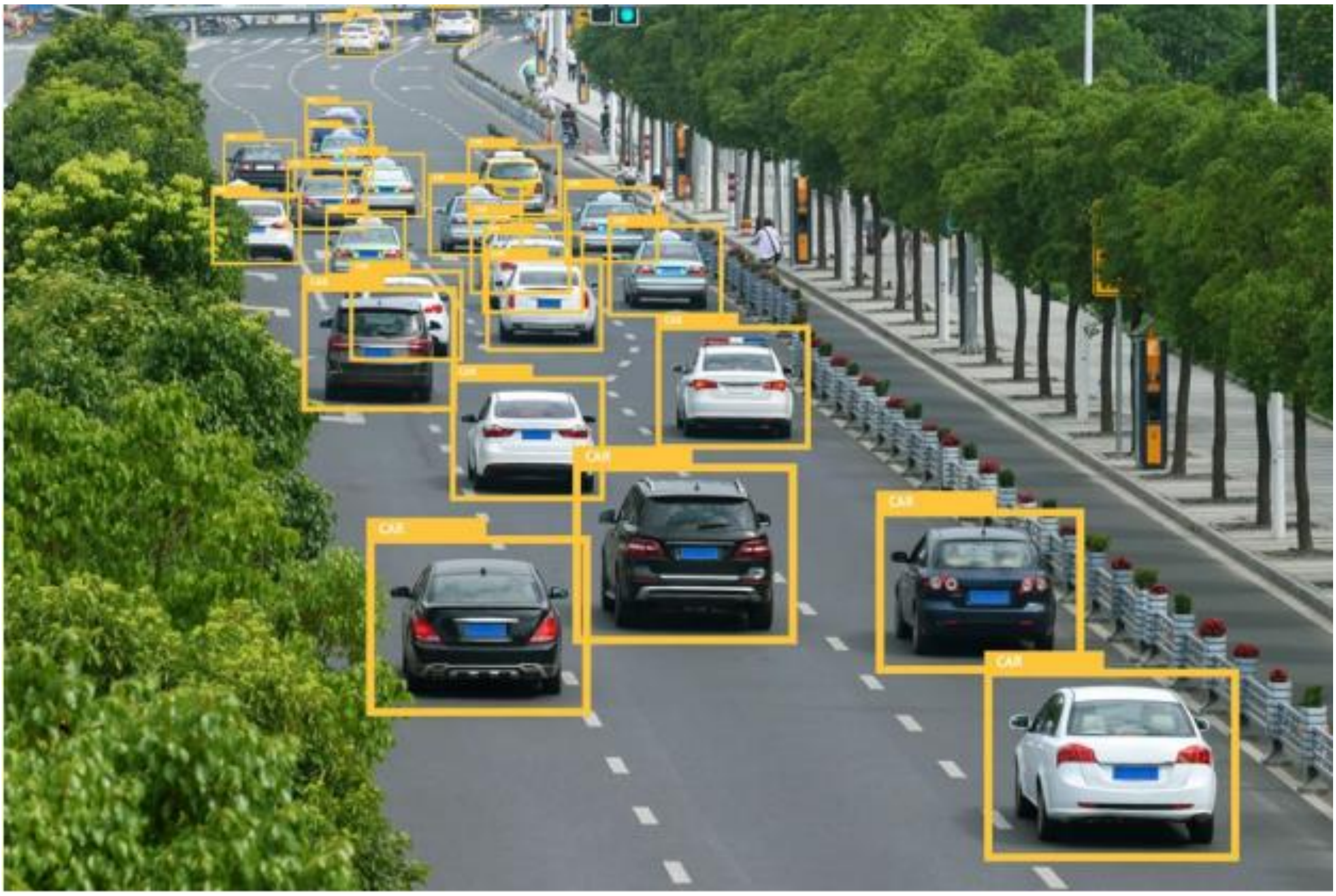
Goal:

1. To make people follow traffic rules.
2. To help the crime handling agencies in finding stolen vehicles.
3. To help the government's pollution checkpoints to keep a proper record of old vehicles which shouldn‘t be used to due to the amount of pollution they are responsible for.
4. To decrease the amount of paperwork done on vehicle-related issues.
5. To make the data processing faster in vehicle-related issues as all the things will be stored in a particular database.

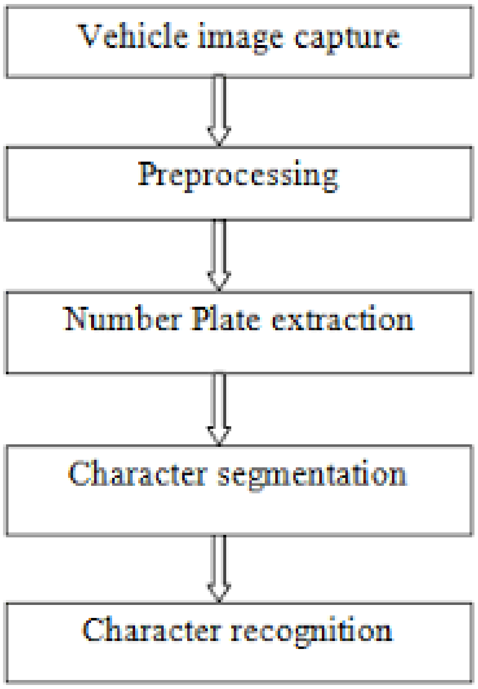
Advantages:

1. This project decreases the time consumption of a person.
2. Easy to use.
3. Easy to access the data.
4. Helps for the benefit of society.
5. Might lead to depression in the stealing vehicles.
6. User friendly.
7. And many more.

IN


CAR
CAR
CAR
CAR
CAR
CAR
CAR

Project management and Professional communication (Presentation)



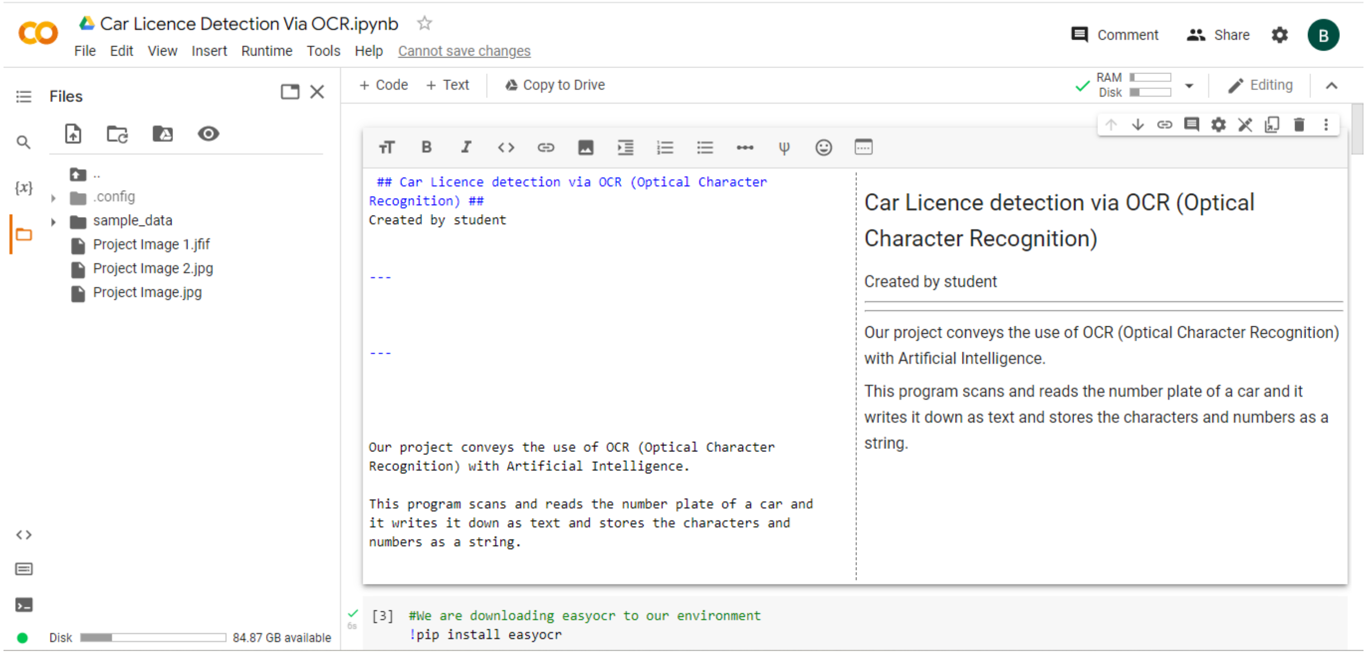
Vehicle image capture

Preprocessing

Number Plate extraction

Character segmentation

Character recognition



```
## Car Licence detection via OCR (Optical Character
Recognition) ##
Created by student


---


---



Our project conveys the use of OCR (Optical Character
Recognition) with Artificial Intelligence.

This program scans and reads the number plate of a car and
it writes it down as text and stores the characters and
numbers as a string.
```

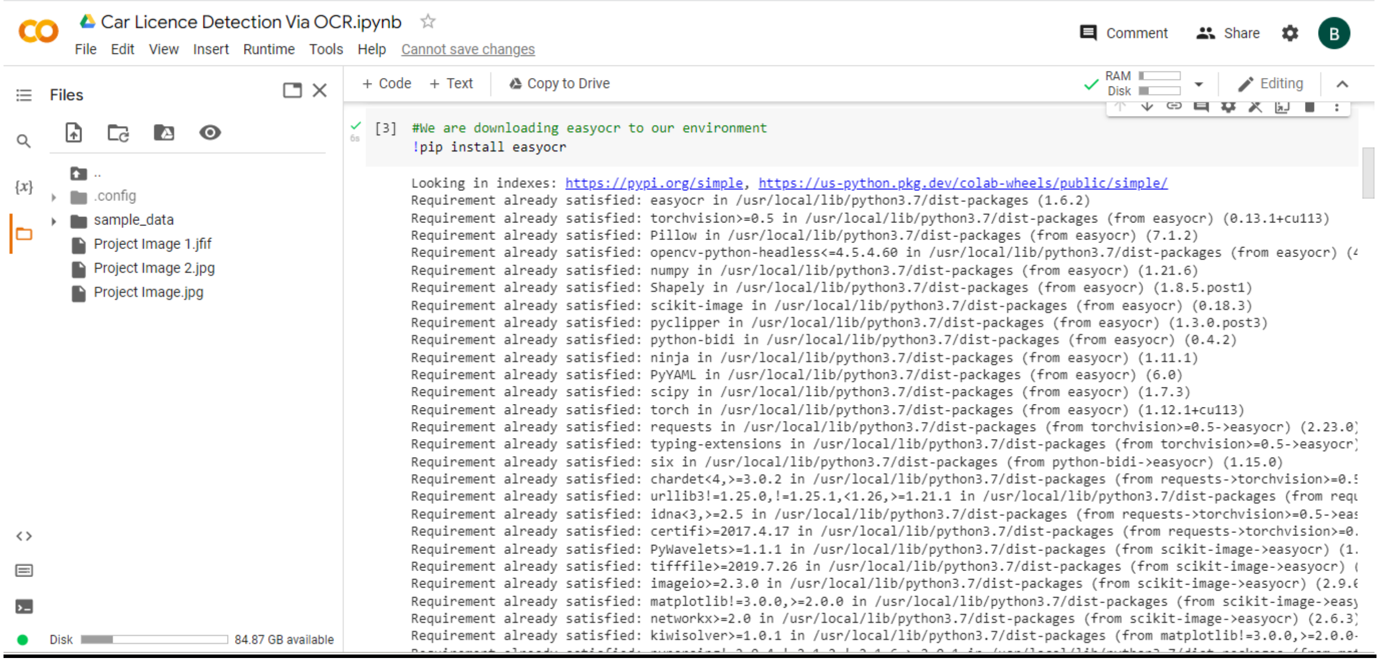
Car Licence detection via OCR (Optical Character Recognition)

Created by student

Our project conveys the use of OCR (Optical Character Recognition) with Artificial Intelligence.

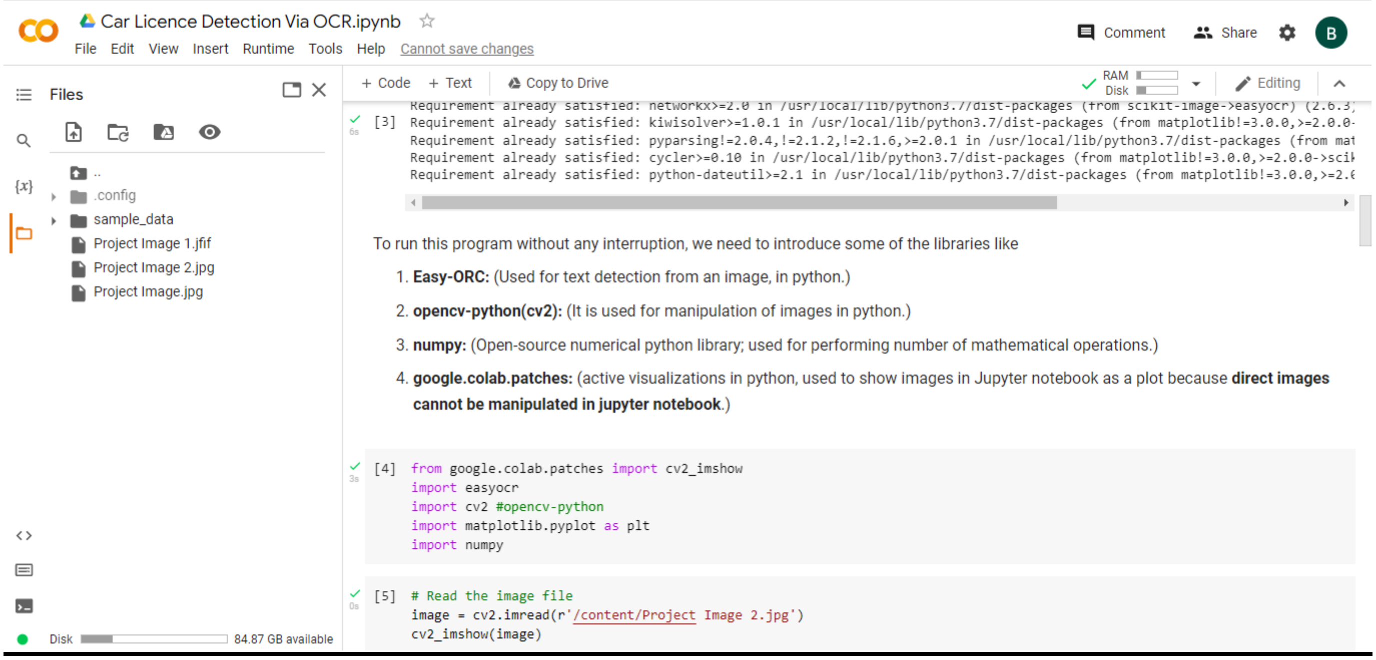
This program scans and reads the number plate of a car and it writes it down as text and stores the characters and numbers as a string.

```
[3] #We are downloading easyocr to our environment
    !pip install easyocr
```



```
[3] #We are downloading easyocr to our environment
    !pip install easyocr

    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Requirement already satisfied: easyocr in /usr/local/lib/python3.7/dist-packages (1.6.2)
    Requirement already satisfied: torchvision>=0.5 in /usr/local/lib/python3.7/dist-packages (from easyocr) (0.13.1+cu113)
    Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from easyocr) (7.1.2)
    Requirement already satisfied: opencv-python-headless<=4.5.4.60 in /usr/local/lib/python3.7/dist-packages (from easyocr) (4
    Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.21.6)
    Requirement already satisfied: Shapely in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.8.5.post1)
    Requirement already satisfied: scikit-image in /usr/local/lib/python3.7/dist-packages (from easyocr) (0.18.3)
    Requirement already satisfied: pyclipper in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.3.0.post3)
    Requirement already satisfied: python-bidi in /usr/local/lib/python3.7/dist-packages (from easyocr) (0.4.2)
    Requirement already satisfied: ninja in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.11.1)
    Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-packages (from easyocr) (6.0)
    Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.7.3)
    Requirement already satisfied: torch in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.12.1+cu113)
    Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from torchvision>=0.5->easyocr) (2.23.0)
    Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from torchvision>=0.5->easyocr)
    Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from python-bidi->easyocr) (1.15.0)
    Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->torchvision>=0.5
    Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requ
    Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->torchvision>=0.5->eas
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->torchvision>=0.
    Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (1.
    Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (
    Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (2.9.0
    Requirement already satisfied: matplotlib!=3.0.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easy
    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-
```



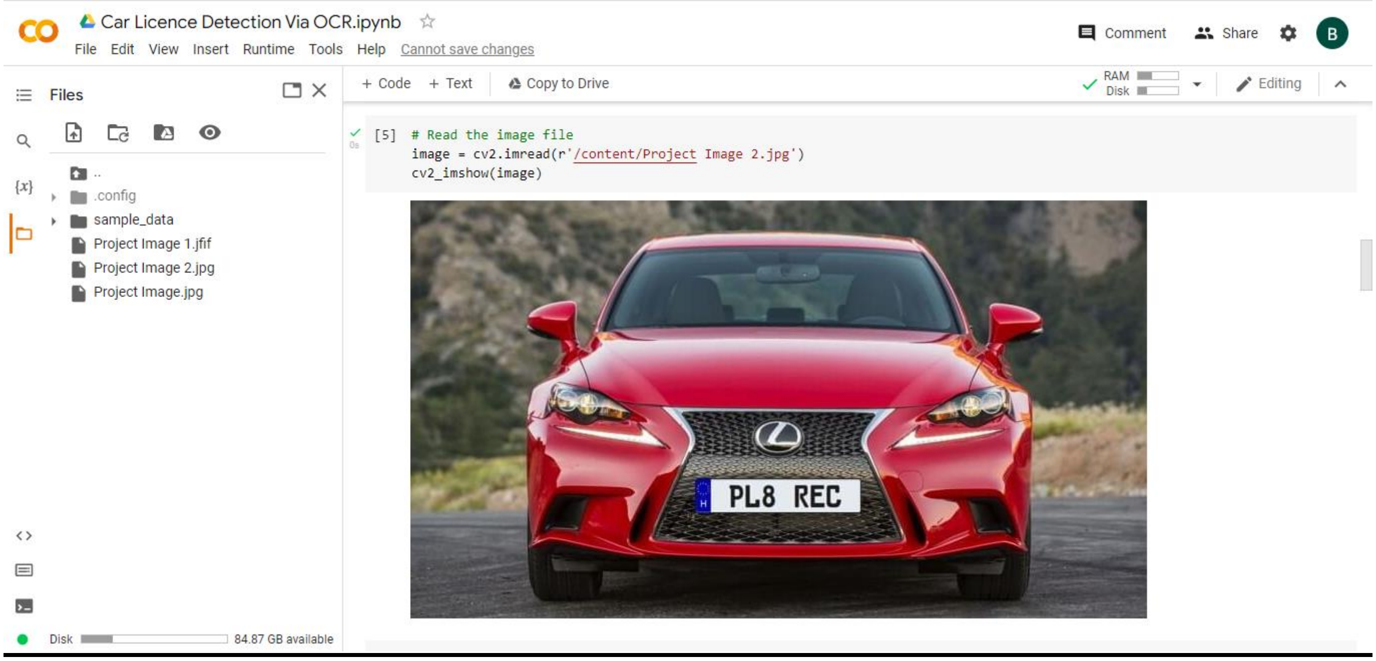
```
    Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (2.6.3)
[3] Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0-
    Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from mat
    Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scik
    Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0
```

To run this program without any interruption, we need to introduce some of the libraries like

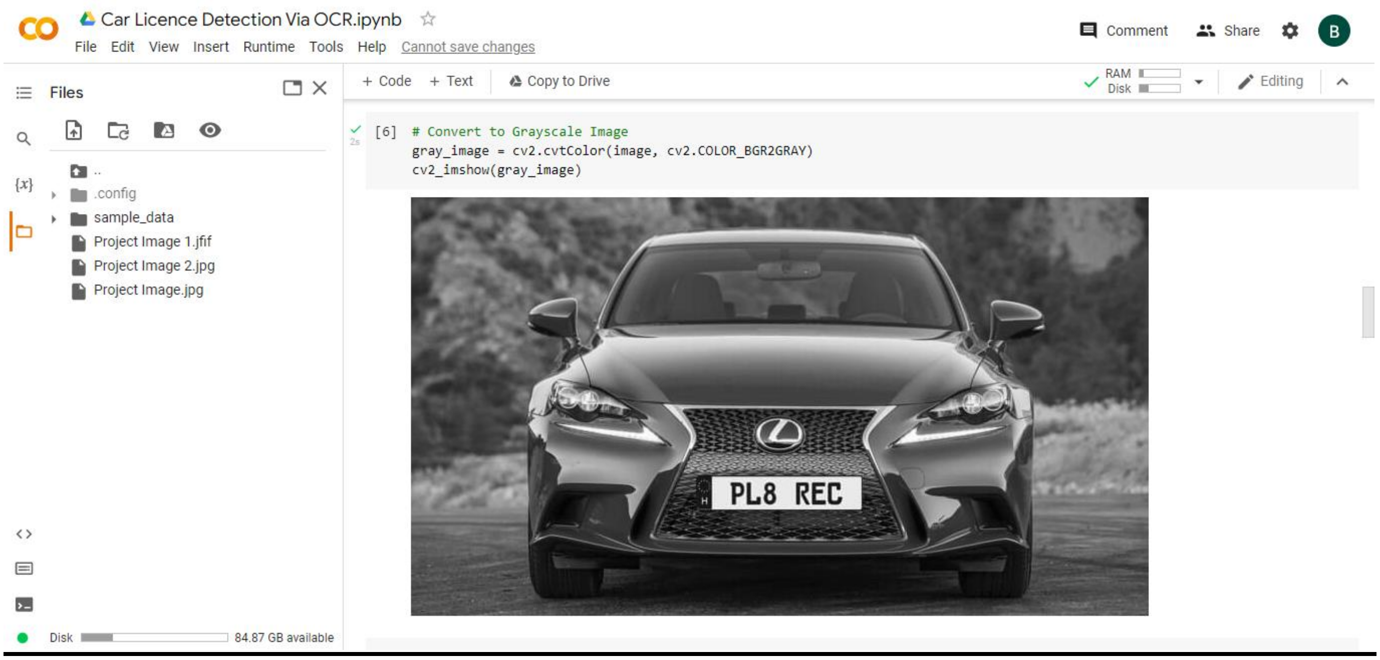
1. **Easy-ORC:** (Used for text detection from an image, in python.)
2. **opencv-python(cv2):** (It is used for manipulation of images in python.)
3. **numpy:** (Open-source numerical python library; used for performing number of mathematical operations.)
4. **google.colab.patches:** (active visualizations in python, used to show images in Jupyter notebook as a plot because **direct images cannot be manipulated in jupyter notebook**.)

```
[4] from google.colab.patches import cv2_imshow
    import easyocr
    import cv2 #opencv-python
    import matplotlib.pyplot as plt
    import numpy
```

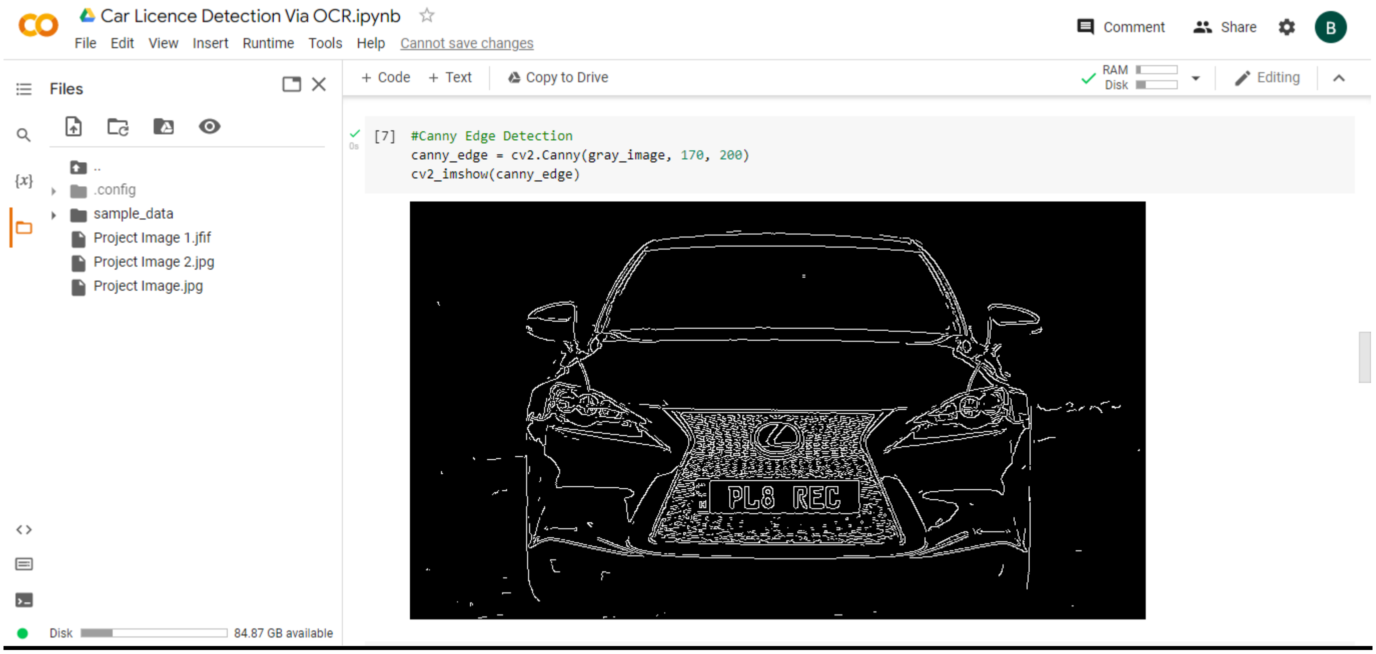
```
[5] # Read the image file
    image = cv2.imread(r'/content/Project Image 2.jpg')
    cv2_imshow(image)
```



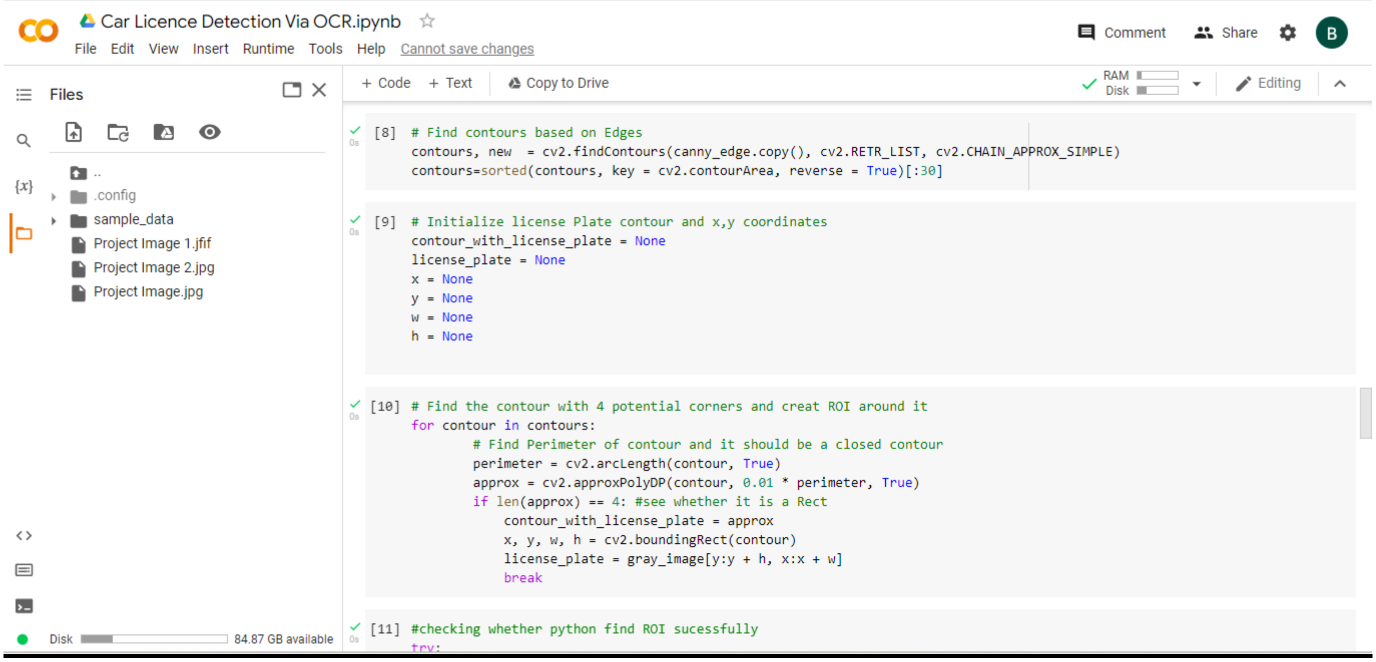
```
# Read the image file
image = cv2.imread(r'/content/Project Image 2.jpg')
cv2_imshow(image)
```



```
# Convert to Grayscale Image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2_imshow(gray_image)
```



```
[7] #Canny Edge Detection
    canny_edge = cv2.Canny(gray_image, 170, 200)
    cv2_imshow(canny_edge)
```

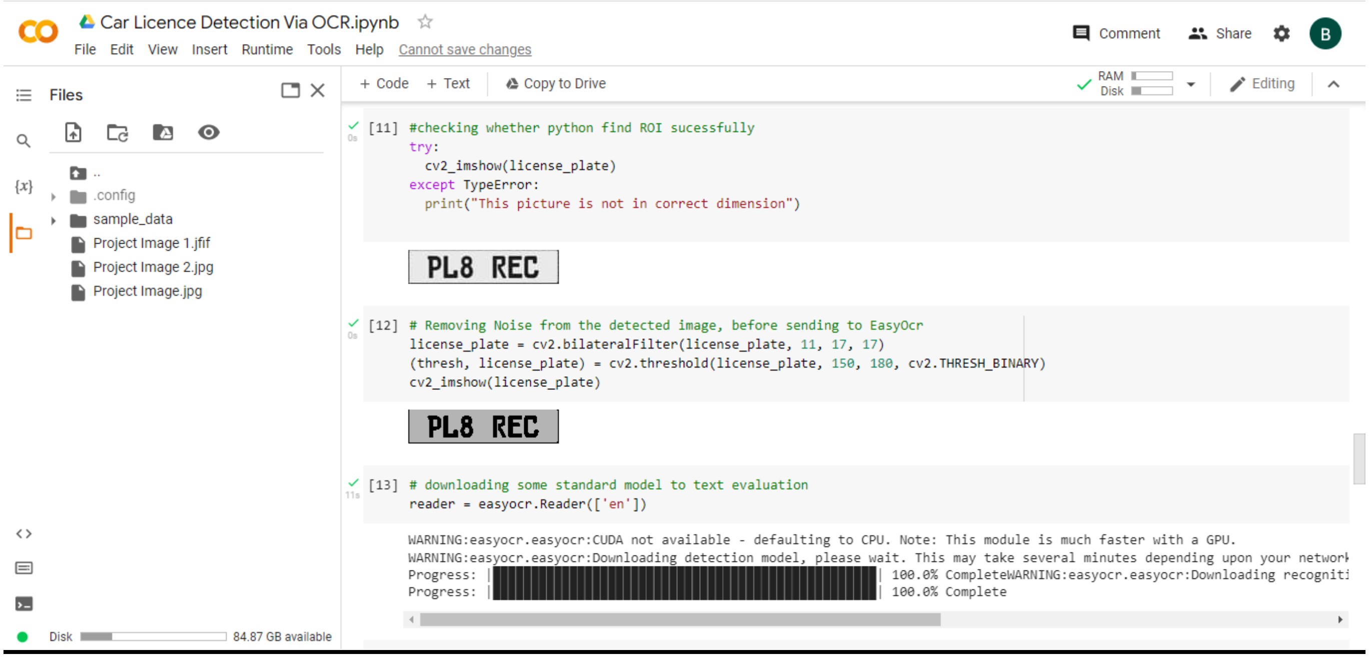


```
[8] # Find contours based on Edges
    contours, new  = cv2.findContours(canny_edge.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
    contours=sorted(contours, key = cv2.contourArea, reverse = True)[:30]
```

```
[9] # Initialize license Plate contour and x,y coordinates
    contour_with_license_plate = None
    license_plate = None
    x = None
    y = None
    w = None
    h = None
```

```
[10] # Find the contour with 4 potential corners and creat ROI around it
     for contour in contours:
             # Find Perimeter of contour and it should be a closed contour
             perimeter = cv2.arcLength(contour, True)
             approx = cv2.approxPolyDP(contour, 0.01 * perimeter, True)
             if len(approx) == 4: #see whether it is a Rect
                 contour_with_license_plate = approx
                 x, y, w, h = cv2.boundingRect(contour)
                 license_plate = gray_image[y:y + h, x:x + w]
                 break
```

```
[11] #checking whether python find ROI sucessfully
     try:
```

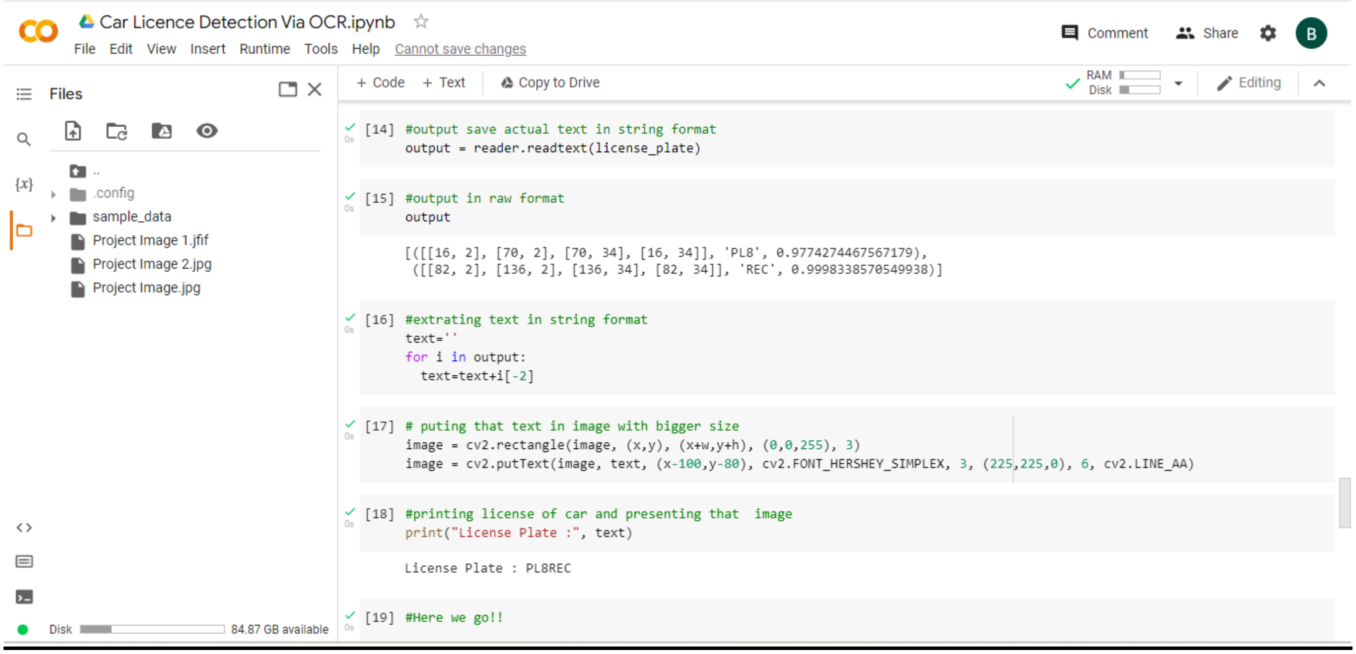


```
[11] #checking whether python find ROI sucessfully
     try:
       cv2_imshow(license_plate)
     except TypeError:
       print("This picture is not in correct dimension")
```

```
[12] # Removing Noise from the detected image, before sending to EasyOcr
     license_plate = cv2.bilateralFilter(license_plate, 11, 17, 17)
     (thresh, license_plate) = cv2.threshold(license_plate, 150, 180, cv2.THRESH_BINARY)
     cv2_imshow(license_plate)
```

```
[13] # downloading some standard model to text evaluation
     reader = easyocr.Reader(['en'])
```

```
WARNING:easyocr.easyocr:CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
WARNING:easyocr.easyocr:Downloading detection model, please wait. This may take several minutes depending upon your networ
Progress: |██████████████████████████████████████████████████| 100.0% CompleteWARNING:easyocr.easyocr:Downloading recogniti
Progress: |██████████████████████████████████████████████████| 100.0% Complete
```



```
[14] #output save actual text in string format
     output = reader.readtext(license_plate)
```

```
[15] #output in raw format
     output
```

```
[([[16, 2], [70, 2], [70, 34], [16, 34]], 'PL8', 0.9774274467567179),
 ([[82, 2], [136, 2], [136, 34], [82, 34]], 'REC', 0.9998338570549938)]
```

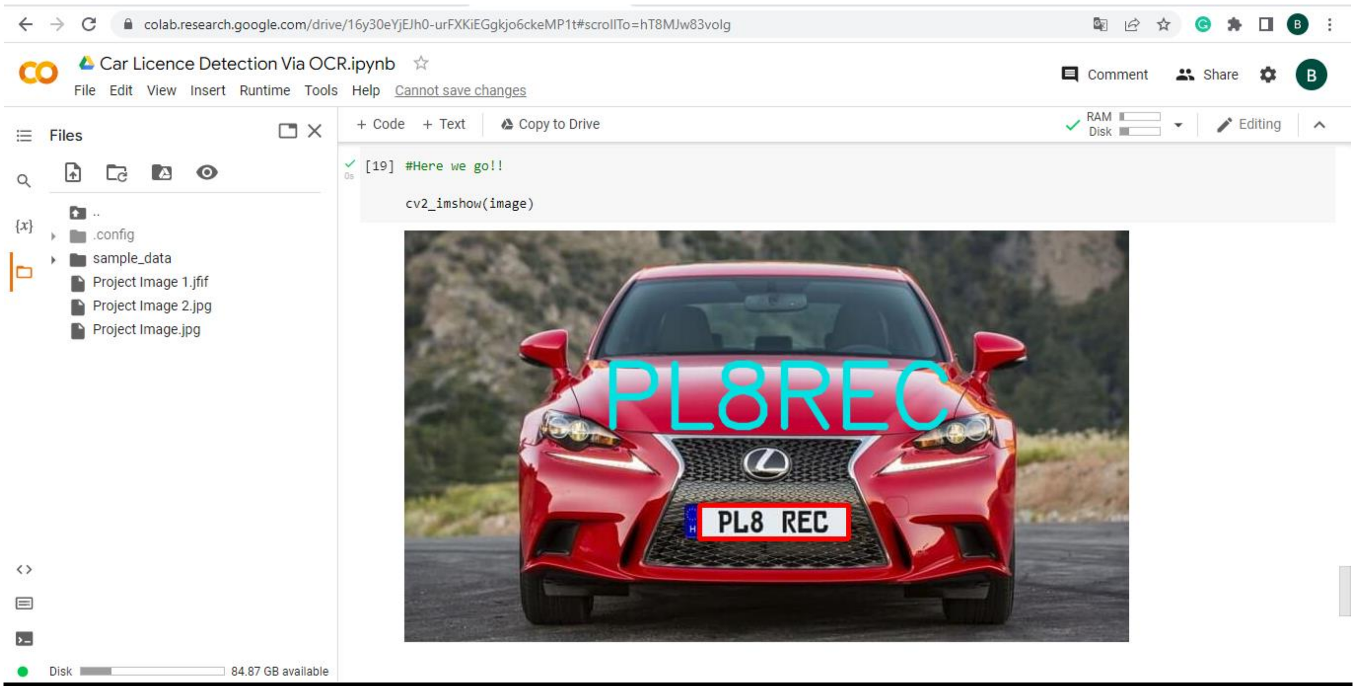
```
[16] #extrating text in string format
     text=''
     for i in output:
       text=text+i[-2]
```

```
[17] # puting that text in image with bigger size
     image = cv2.rectangle(image, (x,y), (x+w,y+h), (0,0,255), 3)
     image = cv2.putText(image, text, (x-100,y-80), cv2.FONT_HERSHEY_SIMPLEX, 3, (225,225,0), 6, cv2.LINE_AA)
```

```
[18] #printing license of car and presenting that  image
     print("License Plate :", text)
```

```
License Plate : PL8REC
```

```
[19] #Here we go!!
```

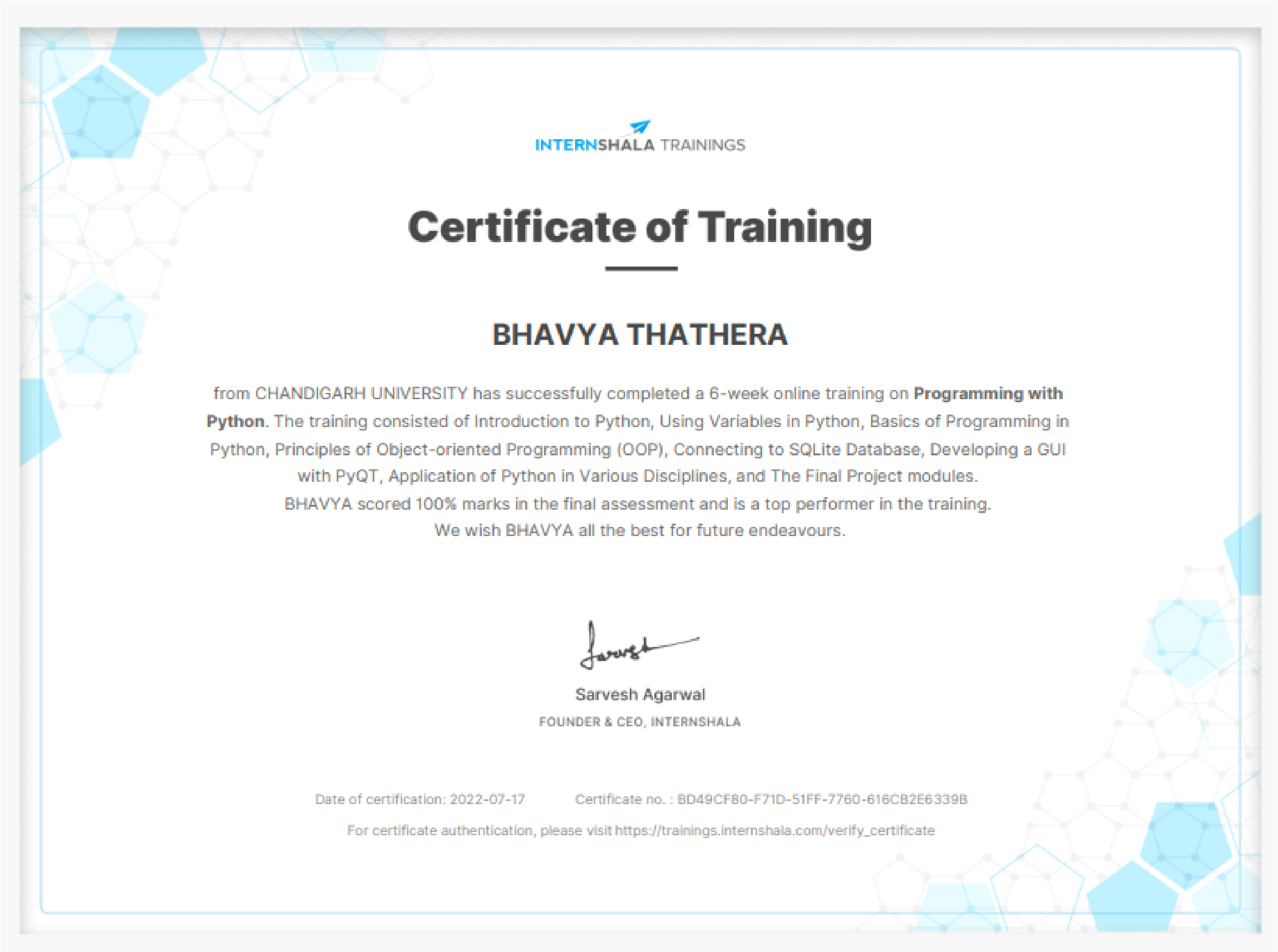

colab.research.google.com/drive/16y30eYjEJh0-urFXKiEGgkjo6ckeMP1t#scrollTo=hT8MJw83volg
Car Licence Detection Via OCR.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Comment Share
+ Code + Text Copy to Drive
RAM Disk
Editing
Files
..
.config
sample_data
Project Image 1.jfif
Project Image 2.jpg
Project Image.jpg
Disk 84.87 GB available
PL8REC
PL8 REC

```
[19] #Here we go!!

     cv2_imshow(image)
```

Attainment of stated outcomes

- Verifying the ownership of the vehicle (Law Enforcement Purposes).
- Used at parking lots for the ease of finding someone’s vehicle.
- Useful for faster processing of plate numbers at the service centres or at the Pollution Check Points.
- Keeping the track of vehicles passed at the toll points at National and Express Highways.
- Security Purposes (Automatic Registration of Vehicle Number Plate via CCTV Footages).
- Reduction of the amount of Paperwork.



INTERNSHALA TRAININGS

Certificate of Training

BHAVYA THATHERA

from CHANDIGARH UNIVERSITY has successfully completed a 6-week online training on **Programming with Python**. The training consisted of Introduction to Python, Using Variables in Python, Basics of Programming in Python, Principles of Object-oriented Programming (OOP), Connecting to SQLite Database, Developing a GUI with PyQT, Application of Python in Various Disciplines, and The Final Project modules.
BHAVYA scored 100% marks in the final assessment and is a top performer in the training.
We wish BHAVYA all the best for future endeavours.

Sarvesh Agarwal
FOUNDER & CEO, INTERNSHALA

Date of certification: 2022-07-17 Certificate no. : BD49CF80-F71D-51FF-7760-616CB2E6339B

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate