TRAFFIC CHALLAN SYSTEM USING OpenCV

A

Major Project Report

Submitted in partial fulfillment of the
Academic requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

G PRANATHI POORVA	**(17SS1A0517)**
N HARSHINI REDDY	**(17SS1A0519)**
R SHREYA REDDY	**(17SS1A0542)**
SABAHATH MAHEEN	**(17SS1A0543)**

Under the guidance of
Sri.JOSHI SHRIPAD

Department Of Computer Science and Engineering
JNTUH College of Engineering Sultanpur
Sultanpur(V),Pulkal(M),Sangareddy district,Telangana-502273

June 2021

JNTUH COLLEGE OF ENGINEERING SULTANPUR

Sultanpur(V),Pulkal(M),Sangareddy-502273 ,Telangana

Department of Computer Science and Engineering

Certificate

This is to certify that The Major Project report work entitled **" TRAFFIC CHALLAN SYSTEM USING OpenCV "** is a bonafide work carried out by the team consisting of **G PRANATHI POORVA** bearing Roll No. **17SS1A0517**, **N HARSHINI REDDY** bearing Roll No. **17SS1A0519**, **R SHREYA REDDY** bearing Roll No. **17SS1A0542**, **SABAHATH MAHEEN** bearing Roll No. **17SS1A0543** in partial fulfillment of the requirements of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING discipline to Jawaharlal Nehru Technological University,Hyderabad during the academic year 2020-21.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

Project Guide	**Head of the Department**
Sri.JOSHI SHRIPAD	**Sri.JOSHI SHRIPAD**
Associate Professor	**Associate Professor**

Principal

Sri. B.BALU NAIK Professor

Declaration

We are glad to declare that the Major Project report work entitled **"TRAFFIC CHALLAN SYSTEM USING OpenCV"** is the work done by the team consisting of **G PRANATHI POORVA** bearing Roll No. **17SS1A0517**, **N HARSHINI REDDY** bearing Roll No. **17SS1A0519**, **R SHREYA REDDY** bearing Roll No. **17SS1A0542**, **SABAHATH MAHEEN** bearing Roll No. **17SS1A0543** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Jawaharlal Nehru Technological University Hyderabad College of Engineering Sultanpur.The results embodied in this Report have not been submitted to any other University or Institution for Recognition or for Award of any degree or diploma .

G PRANATHI POORVA	(**17SS1A0517**)
N HARSHINI REDDY	(**17SS1A0519**)
R SHREYA REDDY	(**17SS1A0542**)
SABAHATH MAHEEN	(**17SS1A0543**)

Acknowledgement

We are overwhelmed in all humbleness and gratefulness to acknowledge our depth to all those who have helped us to put these ideas, well above the level of simplicity and into something concrete . It is our pleasure to acknowledge the help of all those individuals who where responsible for foreseeing the successful completion of our technical seminar report .
We would like to Express our Sincere Gratitude to **Prof. B.Balu Naik**, The Principal of JNTUHCES for his Support and Encouragement throughout the course period.
We would like to Express our Gratitude to **Prof. V.Venkateswara Reddy**, The Vice Principal of JNTUHCES for his Kind Help and Consistent Effort.
We are very thankful to our guide **Sri. Joshi Shripad**, Associate Professor and Head of the Department of Computer Science and Engineering of JNTUHCES for his Inspiring words and Guidance in the completion of our Major Project Report. Finally, We express our Gratitude with Great Admiration and Respect to our faculty for their moral and technical support and motivation throughout the course.

G PRANATHI POORVA (**17SS1A0517**)
N HARSHINI REDDY (**17SS1A0519**)
R SHREYA REDDY (**17SS1A0542**)
SABAHATH MAHEEN (**17SS1A0543**)

Abstract

This Project Aims at Implementing Instant Traffic Challan System using OpenCV that recognizes any Digital Image automatically on the Number plate and sends an email to the corresponding owner. It includes various operations such as Taking pictures, Localizing the Number plate, Character Segmentation and Recognition as well as sending the owner the challan (fine) details with Respective Photograph as Proof . The main idea of this system is to Design and Develop Effective Image Processing Techniques and Algorithms to localize the license plate in the Captured image, to divide the characters from that number plate and to identify each character of the segment by using the Open Computer Vision Library. This system is Time Efficient and Automates the process up to a Big Extent. Through this process , The cost of the project is reduced and also human intervention is minimized , which has become important in these Social Distancing Times . Many applications can be implemented by using this system, such as Security, Highway Speed Detection, Identification of Handwritten Text, Discovery of Stolen Cars, Automatic Fee Collection Systems.

List of Figures

List of Abbreviations

OpenCV	Open Source Computer Vision
OCR	Optical Character Recognition
LBPH	Local Binary Pattern History
IDLE	Integrated Development and Learning Environment
GUI	Graphical User Interface
IDE	Integrated Development Environment

Contents

Chapter 1

INTRODUCTION

1.1 Statement of Problem

In the present way of issuing traffic challans, police men after capturing images of the vehicles must manually enter the licence plate number into the database so as to retrieve the owner details. The owner is then informed about the challan through his contact details. This process is time consuming, less efficient and also takes lot of human supervision.

1.2 Scope of Work

The "Traffic Challaning System using OpenCV" is used to automatically detect number plate from the image and then convert it into text format. The relevant vehicle's owner is identified and notified through an email.

1.3 Aim of the Project

This project aims to make an attempt to automate the now existing Challan system(Traffic fines) in our country. Instead of relying on manual work, this project provides us with option to implicitly identify the number-plate of the vehicle, owner of the vehicle and also inform the owner regarding the challan.

Chapter 2

LITERATURE SURVEY

Traffic control and vehicle owner identification have become a major problem in every country. Sometimes it is too difficult to issue challan manually. ANPR or license plate recognition (LPR) has been one of the useful approaches for vehicle surveillance. Most of the ANPR systems are based on common approaches like artificial neural network (ANN), Probabilistic neural network (PNN), Optical Character Recognition (OCR), Feature salient, MATLAB, Configurable method, Sliding concentrating window (SCW), BP neural network, support vector machine(SVM), inductive learning, region-based, color segmentation, fuzzy-based algorithm, scale-invariant feature transform (SIFT), trichromatic imaging, Least Square Method(LSM), online license plate matching based on weighted edit distance and color-discrete characteristics. A number plate can be extracted by using the image segmentation method. There are numerous image segmentation methods available in various kinds of literature. In most of the methods, image binarization is used. Some authors use Otsu's method for image binarization to convert the color image to a grayscale image. Some plate segmentation algorithms are based on color segmentation.

2.1 Common number plate extraction methods:-

1.Image Binarization

Image binarization is a process to convert an image to black and white. In this method, a certain threshold is chosen to classify certain pixels as black and certain pixels as white. But the main problem is how to choose the correct threshold value for a particular image. A threshold can be selected by the user manually or it can be selected by an algorithm automatically which is known as automatic thresholding.

2.Edge Detection

Edge detection is a fundamental method for feature detection or feature extraction. In the general case, the result of applying edge detection of an algorithm is an object boundary with connected curves. Different edge detection algorithms/operators such as Canny, Canny-Deriche, Differential, Sobel, Prewitt, and Roberts Cross are used for edge detection.

3.Hough Transform

It is a feature extraction technique initially used for line detection. Later on, it has been extended to find the position of arbitrary shapes like circles or oval. The original algorithm was generalized by D.H. Ballard.

4.Blob Detection

Blob detection is used to detect points or regions that differ in brightness or color as compared to surroundings. The main purpose of using this approach is to find complementary regions which are not detected by edge detection or corner detection algorithms. Some common blob detectors are Laplacian of Gaussian (LoG), Difference of Gaussians (DoG), Determinant of Hessian (DoH), maximally stable extremal regions, and Principle curvature based region detector.

5.Connected Component Analysis(CCA)

CCA or blob extraction is an approach to uniquely label subsets of connected components based on a given heuristic. It scans a binary image and labels pixels as per connectivity conditions of the current pixel such as North-East, North, North-West, and West of the current pixel (8-connectivity). 4-connectivity is used for only the north and west neighbors of the current pixel. The algorithm gives better performance and it is very useful for automated image analysis. This method can be used in plate segmentation as well as character segmentation.

6.Mathematical Morphology

Mathematical morphology is based on set theory, lattice theory, topology, and random functions. It is commonly applied to digital images but can be used in other spatial structures also. Initially, it was developed for processing binary images and then extended for processing grayscale functions and images. It contains basic operators such as Erosion, dilation, opening, closing.

2.2 Requirements Analysis

2.2.1 Software Requirements:

Operating System	:Windows 64 bit
Pre-requisite Software	:Python 3.x
Software	: OpenCV 3.0.0
IDE	:Visual Studio code
Python Modules	:Numpy,Pandas,Tkinter,Tesseract

2.2.2 Hardware Requirements:

Hard disk space	:500GB or more.
Processor	:preferably core-i3 or above.
RAM	:4GB RAM or above.

2.3 Feasibility Study

Feasibility study is conducted once the problem is clearly understood.Feasibility study is a high level capsule version of the entire system analysis and design process. The objective is to determine quickly at a minimum expense how to solve a problem. The purpose of feasibility is not to solve the problem but to determine if the problem is worth solving.The system has been tested for feasibility in the following points.
1. Technical Feasibility
2. Economical Feasibility

2.3.1 Technical Feasibility

The project entitles "Traffic Challaning System" is technically feasibility because of the below mentioned feature. The project was developed in Python, additionally used a Tkinter Framework for Graphical User Interface.It provides the rich libraries, also high level of reliability, availability and compatibility.All these make Python an appropriate language for this project.

2.3.2 Economial Feasibility

The computerized system will help in automate the selection leading the profits and details of the organization. With this software, the machine and manpower utilization are expected to go up by 80-90 percentage approximately. Time is very precious , this makes this project an apt in every circumstances.

2.4 Requirement Elicitation

2.4.1 Functional Requirements

System functional requirement describes activities and services that must provide.

1. Taking an image of the vehicle.
2. Preprocessing it and extracting the license plate number.
3. Sending an Email directly to the vehicle owner regarding the challan.

2.4.2 Non-Functional Requirements

Nonfunctional Requirements are characteristics or attributes of the system that can judge its operation. The following points clarify them:

1. **Accuracy and Precision:**the system should perform its process in accuracy and Precision to avoid problems.
2. **Modifiability:**the system should be easy to modify, any wrong should be correct.
3. **Security:**the system should be secure and saving one's privacy.
4. **Usability:**the system should be easy to deal with and simple to understand
5. **Maintainability:**the maintenance group should be able to fix any problem occur suddenly.
6. **Speed and Responsiveness:**Execution of operations should be fast.

2.5 Tools Used

Below is the list of tools used.

2.5.1 Visual Studio Code

Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS.[9] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Figure 2.1: Visual Studio Code

Microsoft has released Visual Studio Code's source code on the microsoft/vscode (Code - OSS) repository of GitHub, under the permissive MIT License,while the releases by Microsoft are freeware but not released under an open-source license.As a result, MIT-licensed binaries have been produced by other projects (e.g. vscodium) that disable Microsoft's built-in telemetry and bundle only open-source plugins (some of Microsoft's are not). Some authors have criticised this practice of effectively silently re-leasing two subtly distinct products under distinct licensing arrangements.

Out-of-the-box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.
Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled.In addition, because of the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected.According to Microsoft, the data is shared with Microsoft-controlled affiliates and subsidiaries, although law enforcement may request it as part of a legal process.

2.6 Technologies used

Language:Python
Modules:OpenCV

2.6.1 Python Language

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.Van Rossum led the language community until July 2018.

Figure 2.2: Python

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python features a comprehensive standard library, and is referred to as "batteries included".Python interpreters are available for many operating systems. C-Python, the reference implementation of Python, is open-source software and has a community-based development model. Python and C-Python are managed by the non-profit Python Software Foundation. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta programming and meta objects (magic methods)).Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

2.6.2 OpenCV Library

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real- time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross platform and free for use under the open-source BSD license. OpenCV supports the deep learning frameworks Tensor Flow, Torch/PyTorch and Caffe.

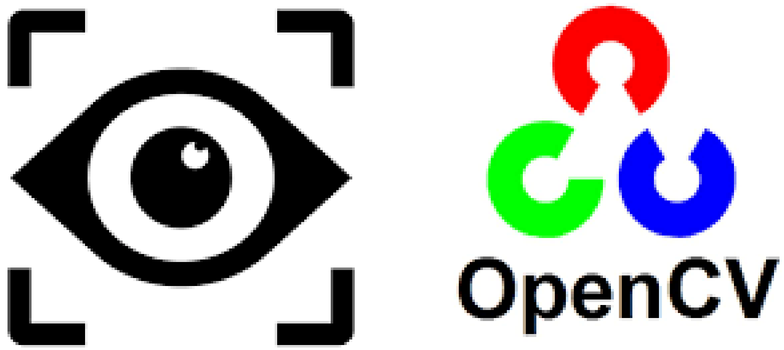


Figure 2.3: OpenCV

Officially launched in 1999, the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls. The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team. In the early days of OpenCV, the goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance optimized code available for free – with a license that did not require code to be open or free itself.

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

2.7 Optical character recognition (OCR) in OpenCV

1.Python-tesseract

Python-tesseract 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. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file.

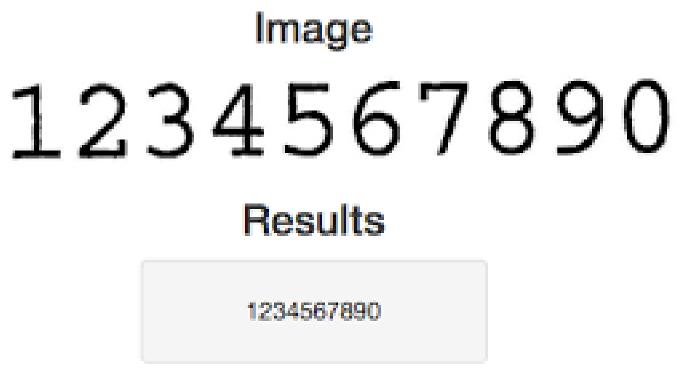


Figure 2.4: Optical Character Recognition

Functions:

1.get_languages Returns all currently supported languages by Tesseract OCR.

2.get_tesseract_version Returns the Tesseract version installed in the system.

3.image_to_string Returns unmodified output as string from Tesseract OCR processing

4.image_to_boxes Returns result containing recognized characters and their box boundaries
5.image_to_data Returns result containing box boundaries, confidences, and other information. Requires Tesseract 3.05+.
6.image_to_osd Returns result containing information about orientation and script detection.
7.image_to_alto_xml Returns result in the form of Tesseract's ALTO XML format.
8.run_and_get_output Returns the raw output from Tesseract OCR. Gives a bit more control over the parameters that are sent to tesseract.

Parameters:
image_to_data(image, lang=None, config=”, nice=0, output_type=Output.STRING, timeout=0, pandas_config=None)

2.8 UML Diagrams

The unified Modeling Language (UML) is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of software-intensive system. The primary goals in the design of the UML were:
1. Provide users with a ready-to-use, expressive visual modelling languages they can develop exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.

2.8.1 Class Diagram

Usage of Class Diagram in Face recognition attendance system:

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.
The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints.
Classes:
- Police
- System
- Dataset
- User

1. It is mainly designed for naive user to understand the details of the project and their Relationships.
2. Improves the understandabilty for a non-techincal user on seeing the class diagram.
3. We provide a description for each use case to show more details about the system.

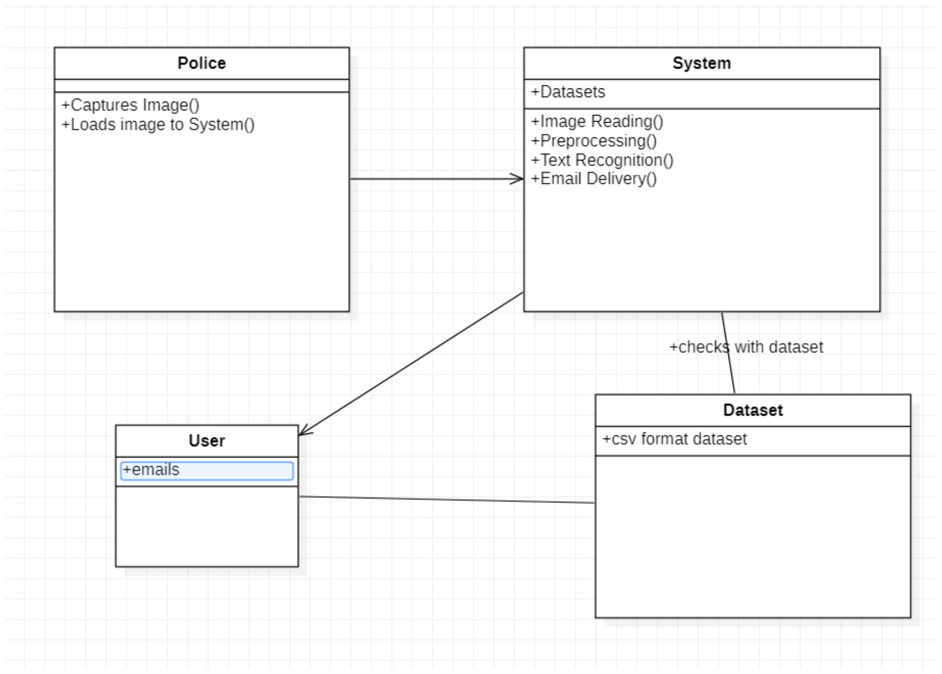


Figure 2.5: Class Diagram

2.8.2 Use case Diagrams

1. The use case diagram represents the pictorial relationship of users and the modules.
2.Using graphics we are defining the entire flow of the project.

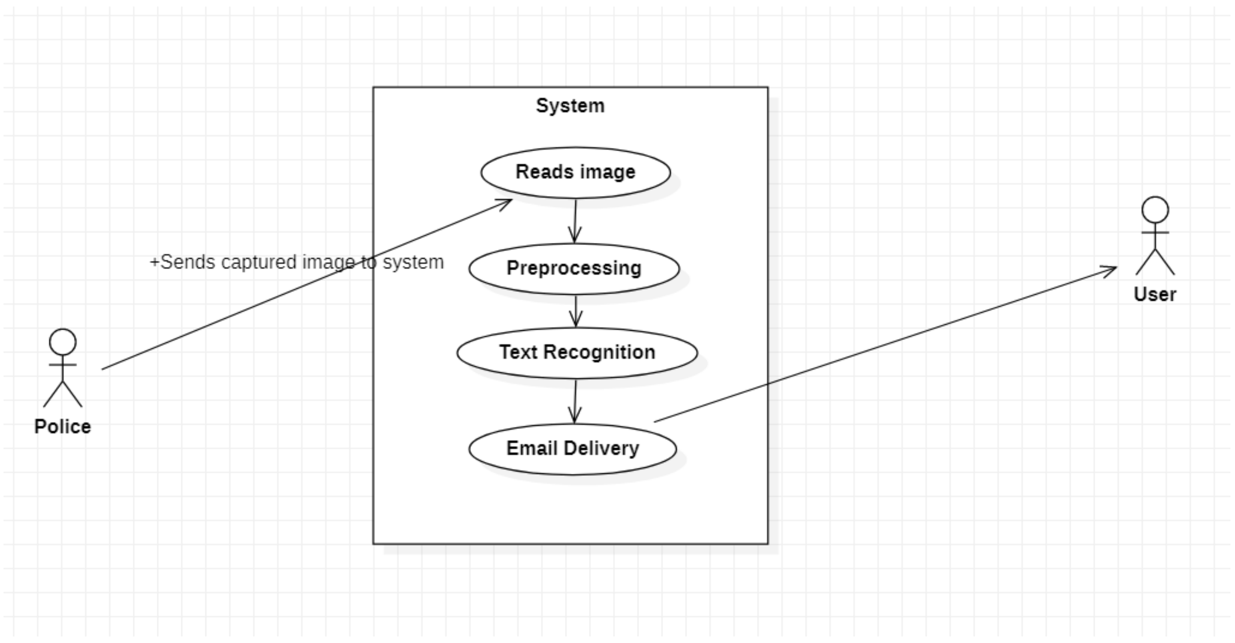


Figure 2.6: Use case diagram

2.8.3 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
The sequence of object interactions are
• Dataset collection
• CSV format of dataset
• Preprocessing
• Feature extraction
• Classification
• Accuracy of result
1. The sequence diagram here shows the interaction logic between the objects of face recognition based attendance system.
2. Sequence diagrams are a popular dynamic modeling solution,because they specifically focus on lifelines, or the processes and objects that live

simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

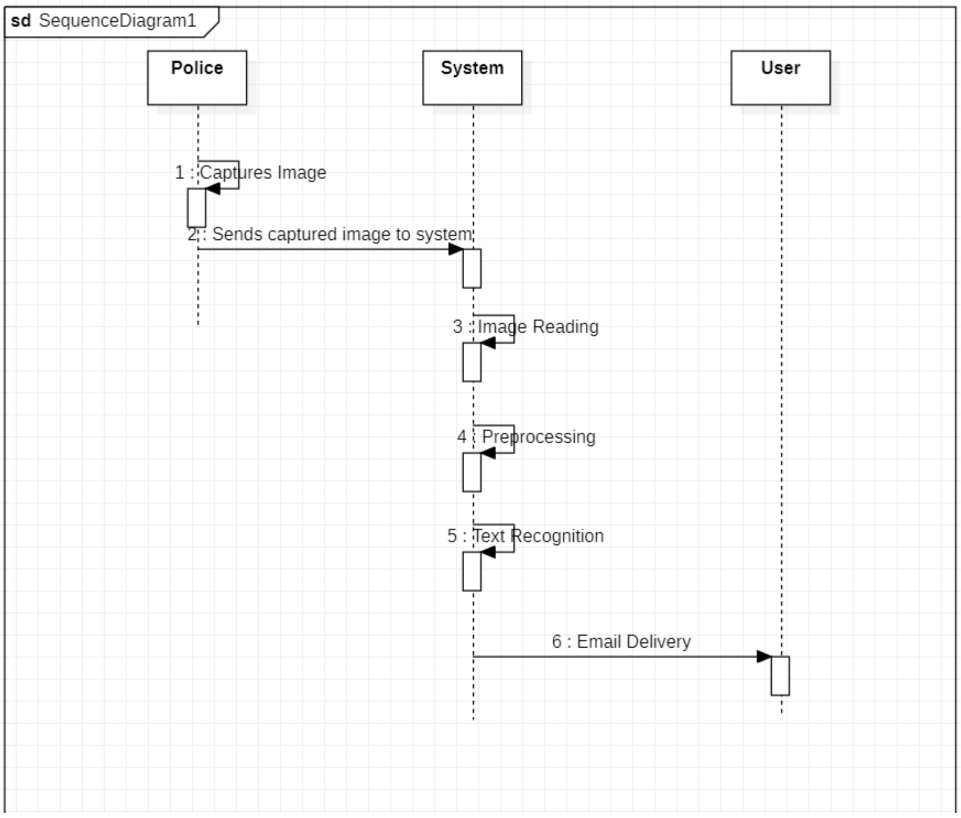


Figure 2.7: Sequence Diagram

2.8.4 Actitvity Diagram

1. Activity diagram represents important behavioral diagram in UML diagram to describe dynamic aspects of the face recognition based attendance system.

2. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

3. Here in our case we will be having three activity diagram each represents a particular flow.

4.The diagrams are

a. Adding new student details to our database

b. Capturing attendance of students

c. Viewing attendance

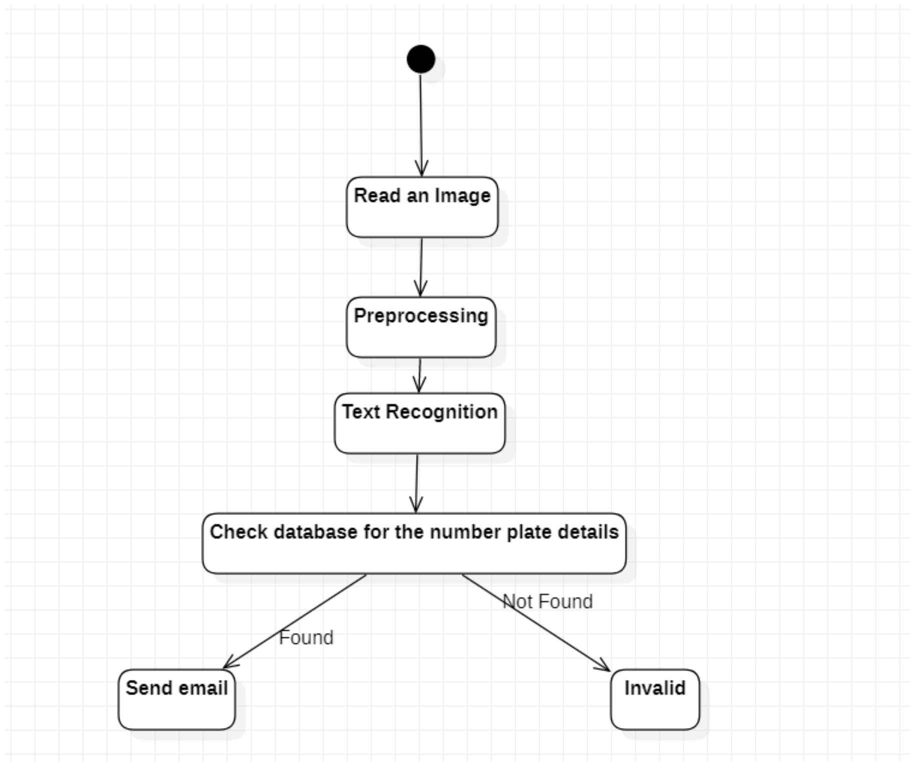


Figure 2.8: Activity Diagram

Chapter 3

IMPLEMENTATION

3.1 Implementing Automatic License Plate Detection using OpenCV

3.1.1 Introduction

This project work is implemented using PYTHON, OPENCV, PYTESSERACT and few Image processing methods. Here we use images from internet and also live images which we capture to identify the numberplate of an individual. We will learn about python, Opencv in the next sections.

3.1.2 Installing OpenCV

First, you need to find the correct setup file for your operating system. I found that installing OpenCV was the hardest part of the task. If you get strange un explainable errors, it could be due to library clashes, 32/64 bit differences, and so on. I found it easiest to just use PyCharm IDE and install OpenCV in settings.

3.1.3 Installing Tesseract-OCR

Tesseract is an optical character recognition engine for various operating systems. We need to install it from Github -Tesseract UB Mannheim.Tesseract should be either installed in the directory which is suggested during the installation or in a new directory. The uninstaller removes the whole installation directory. If you installed Tesseract in an existing directory, that directory will be removed with all its subdirectories and files.

3.2 Writing and running the code(Algorithm)

In this section, we will write code that will take an image as input and return a string i.e The number plate found in the input image.That will be our final answer detecting numberplate from the car as string.

3.2.1 Start by Creating a New File to Hold Our Code

Here we should create a new file to hold our code with .py and to run the code perfectly.
For example:

```
nano app.py
```

3.2.2 Importing Necessary Libraries

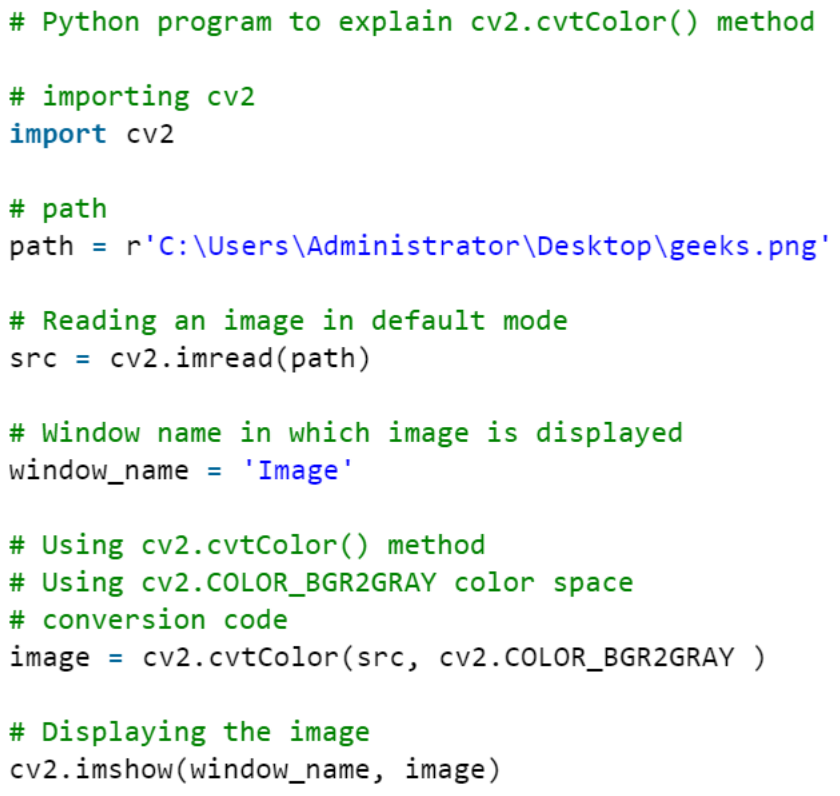
In this new file, start writing code by first importing the necessary libraries. You will import two modules here: cv2 and pytesseract. The cv2 module imports the OpenCV library into the program, and tesseract imports common Pytesseract that our code will use. If you don't get any errors, you can move on to the next part.

3.2.3 Input as Argument

Next, we will specify that the input image will be passed as an argument to the script at runtime. The Pythonic way of reading the first argument is to assign the value returned by sys.argv[1] function to an variable.

3.2.4 Read an Image

A common practice in image processing is to first convert the input image to gray scale. This is because detecting luminance, as opposed to color, will generally yield better results in object detection. Add the following code to take an input image as an argument and convert it to grayscale:



```
# Python program to explain cv2.cvtColor() method

# importing cv2
import cv2

# path
path = r'C:\Users\Administrator\Desktop\geeks.png'

# Reading an image in default mode
src = cv2.imread(path)

# Window name in which image is displayed
window_name = 'Image'

# Using cv2.cvtColor() method
# Using cv2.COLOR_BGR2GRAY color space
# conversion code
image = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY )

# Displaying the image
cv2.imshow(window_name, image)
```

Figure 3.1: Read Image

The .imread() function takes the input image, which is passed as an argument to the script, and converts it to an OpenCV object. Next, OpenCV's .cvtColor() function converts the input image object to a grayscale object.

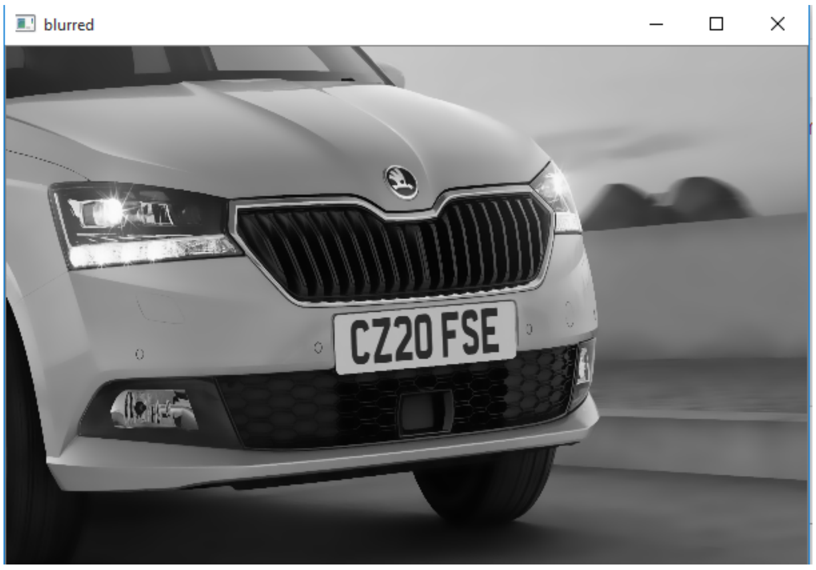


Figure 3.2: blurred

3.2.5 Smoothening the Image

This function detects the actual plate and is the key part of our code, so let's go over the options:
We should eliminate all the noise from the image except car. This helps us to increase the efficiency of the code and also decreases the resolution of the image. So that we can easily locate our license plate.

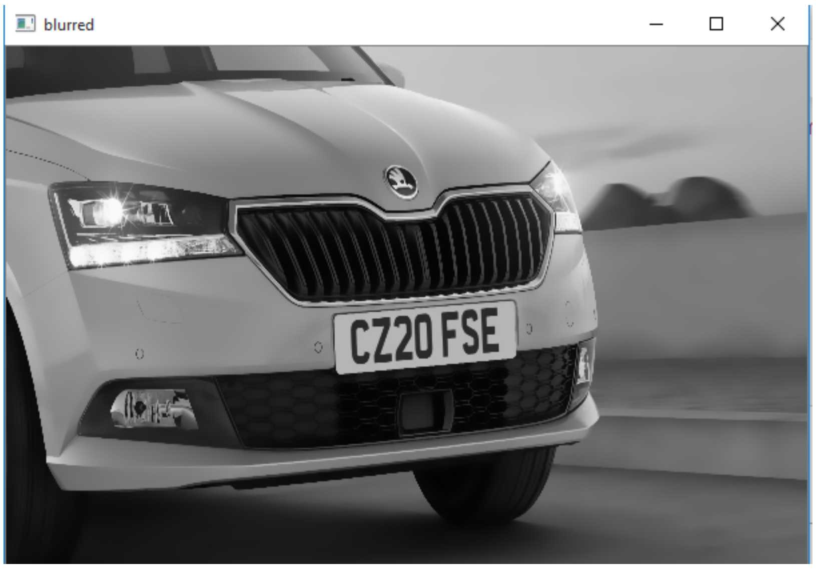


Figure 3.3: blurred

3.2.6 Performing Canny Edge Detection

The popular canny edge detection algorithm is used to detect a wide range of edges in images. OpenCV has in-built function cv2.Canny() which takes our input image as first argument and its aperture size(min value and max value) as last two arguments.

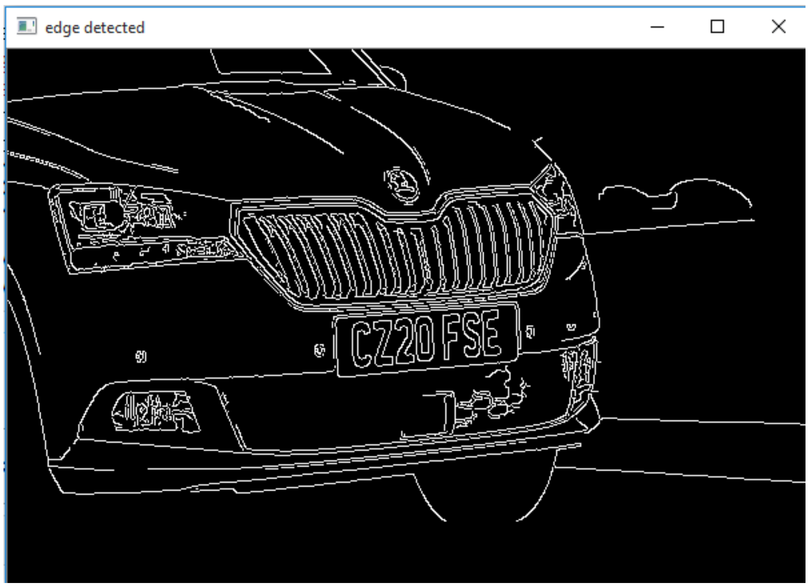


Figure 3.4: Canny edge detection

3.2.7 Detecting License Plate

3.2.8 Identifying contours

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.

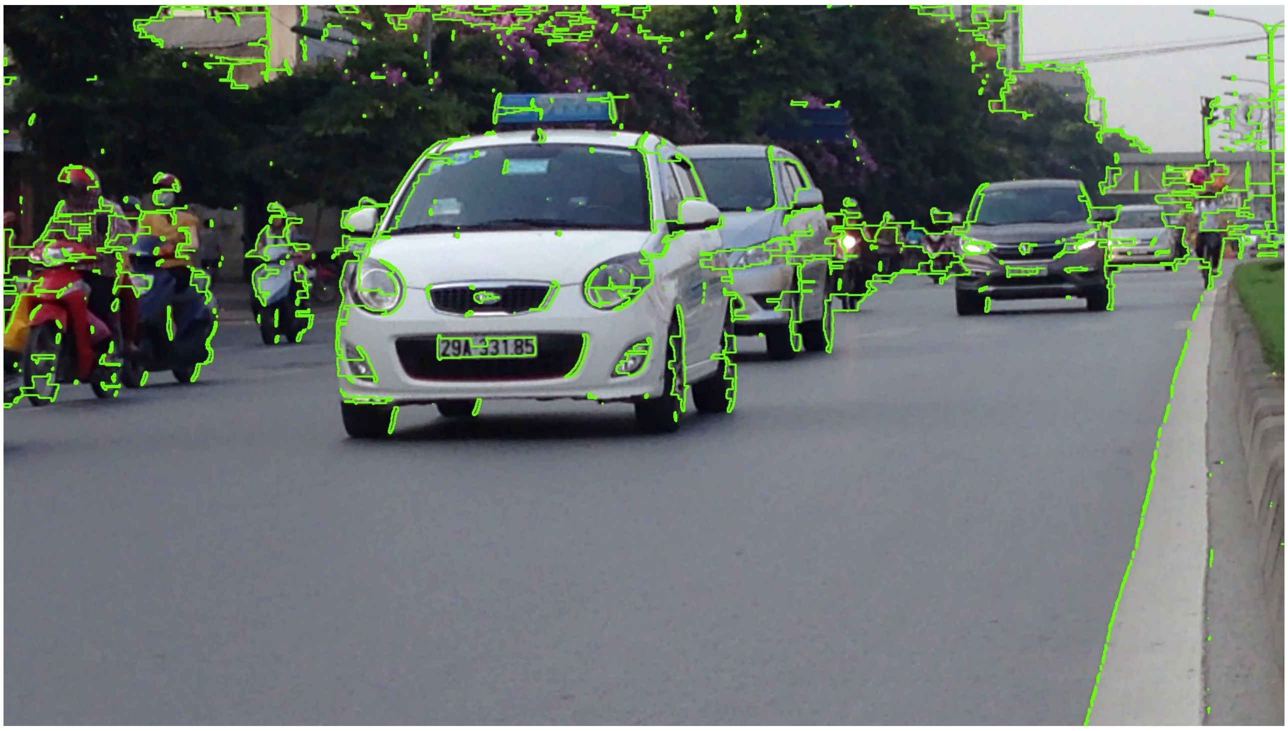


Figure 3.5: ContouredImage

3.2.9 Formation of Rectangle Around Plate

Next, you will use OpenCV's .rectangle() method to draw a rectangle around the detected faces
The function returns a list of rectangles in which it believes it found a face. Next, we will loop over where it thinks it found something. 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 code to crop the roi (Region of interest) image form the main image is shown below.

1.Image tells the code to draw rectangles on the original input image.

2.(x,y), (x+w, y+h) are the four pixel locations for the detected object. rectangle will use these to locate and draw rectangles around the detected objects in the input image.

3.(0, 255, 0) is the color of the shape. This argument gets passed as a tuple for BGR. For example, you would use (255, 0, 0) for blue. We are using green in this case.

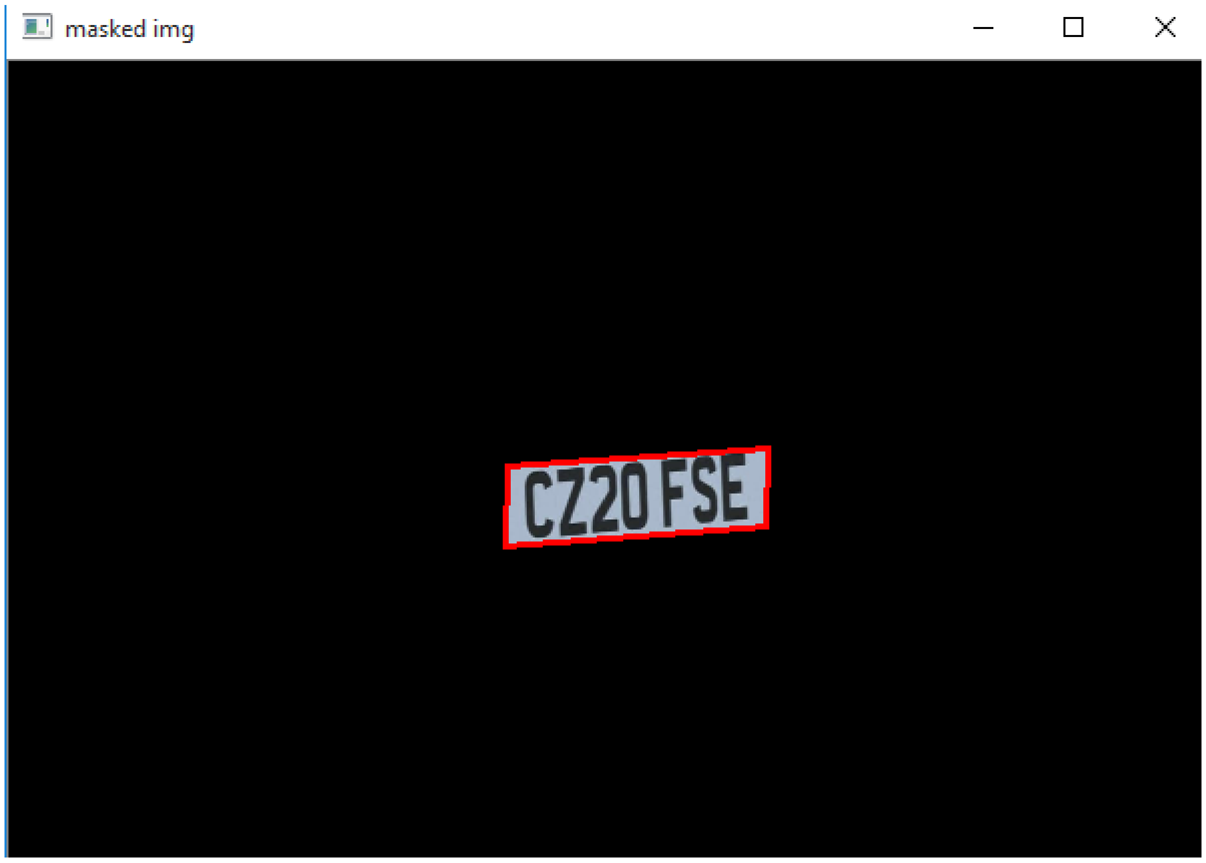
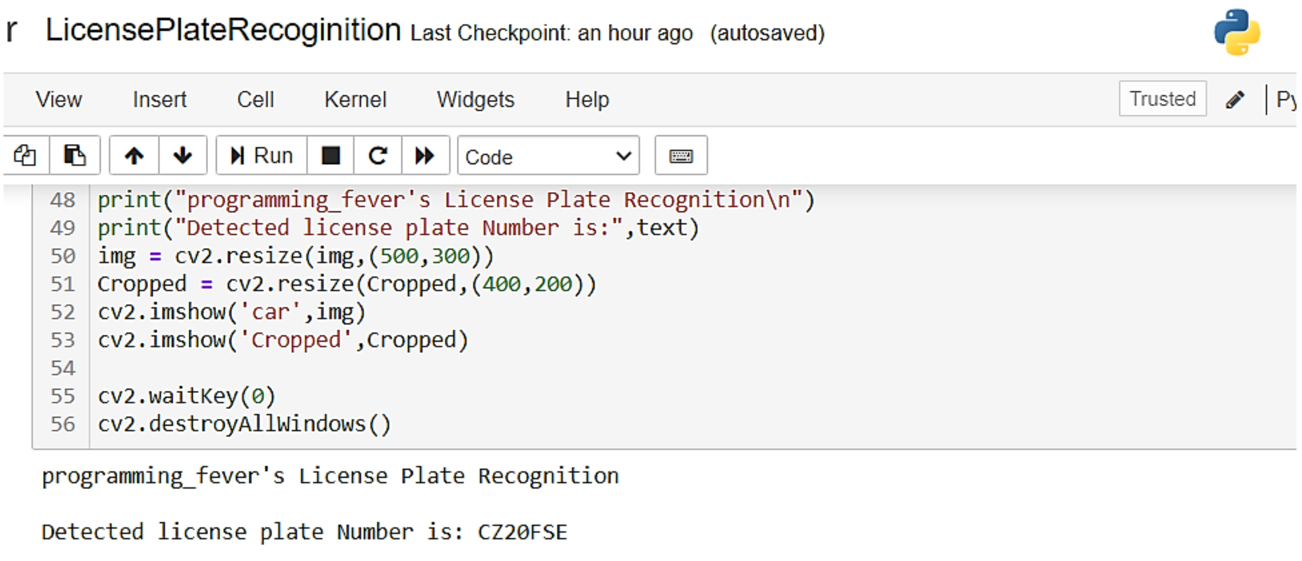


Figure 3.6: License Plate

3.2.10 Extracting text from Image

Now we will earn about Automatic number-plate recognition. We will use the Tesseract OCR An Optical Character Recognition Engine (OCR Engine) to automatically recognize text in vehicle registration plates.
Python-tesseract: Py-tesseract is an optical character recognition (OCR) tool for python. That is, it'll recognize and "read" the text embedded in images. Python-tesseract is a wrapper for Google's Tesseract-OCR Engine.



r LicensePlateRecoginition Last Checkpoint: an hour ago (autosaved)
View Insert Cell Kernel Widgets Help
Trusted
Py
Run
Code

```
print("programming_fever's License Plate Recognition\n")
print("Detected license plate Number is:",text)
img = cv2.resize(img,(500,300))
Cropped = cv2.resize(Cropped,(400,200))
cv2.imshow('car',img)
cv2.imshow('Cropped',Cropped)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
programming_fever's License Plate Recognition

Detected license plate Number is: CZ20FSE
```

Figure 3.7: String Format

3.3 Source Code

3.3.1 main.py

```
import cv2
import pytesseract
import emailDelivery
import UI

#Reading the image file
UI.start()

image=cv2.imread(UI.imname)
#image=cv2.imread('car2.jpg')

# Convert to Grayscale Image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("Gray Scale Image", gray_image)
```

```
gg=cv2.bilateralFilter(gray_image,11,17,17)
cv2.imshow("Smoother Image", gg)
#cv2.waitKey(0)

#Canny Edge Detection
canny_edge = cv2.Canny(gg, 170, 200)
cv2.imshow("Canny edge", canny_edge)
cv2.waitKey(0)

# 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]

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

# Find the contour with 4 potential corners and create 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

"""
Removing Noise from the detected image, before sending to
Tesseract
license_plate = cv2.bilateralFilter(license_plate, 11, 17,
17)
(thresh, license_plate) = cv2.threshold(license_plate, 150,
180, cv2.THRESH_BINARY)
cv2.imshow("Smoother Image", license_plate)
"""



#Text Recognition
pytesseract.pytesseract.tesseract_cmd=r"C:\Program Files\
Tesseract-OCR\tesseract.exe"
text = pytesseract.image_to_string(license_plate)
text=text.lstrip()
text=text.rstrip()
print(text)
```

```
emailDelivery.send_emails(text)
#Draw License Plate and write the Text
image = cv2.rectangle(image, (x,y), (x+w,y+h), (0,0,255), 3)
image = cv2.putText(image, text, (x-100,y-50),
cv2.FONT_HERSHEY_SIMPLEX, 3, (0,255,0), 6, cv2.LINE_AA)
cv2.imshow("License Plate Detection",image)
cv2.waitKey(0)
```

3.3.2 emailDelivery.py

```
import smtplib, ssl
import receiver

port = 465  # For SSL
smtp_server = "smtp.gmail.com"
# Enter your address
sender_email = "testing.majorproj@gmail.com"
# Enter receiver address
receiver_email = "ragireddys@gmail.com"
password = input("Type your password and press enter: ")
message = """\
Subject: Your vehicle is subjected to challan. You need to
pay xxxx amount"""

context = ssl.create_default_context()
with smtplib.SMTP_SSL(smtp_server, port, context=context)
as server:
    server.login(sender_email, password)
    server.sendmail(sender_email, receiver_email, message)
```

3.3.3 receiver.py

```
import pandas as pd


def generate_receiver_email(reg_no):
    records=pd.read_csv("car_dataset.csv")
    email=""
    for row in records.iterrows():
        curr_reg_no=row[1][0]
        if curr_reg_no ==reg_no:
            email= row[1][1]
            break

    return email
```

3.3.4 UI.py

```
import tkinter as tk
from tkinter import font as tkFont
from tkinter import simpledialog
import NewVehicle_Capture
global txt_area
global imname

def capture():
    global imname
    name= simpledialog.askstring(title="Name",
prompt="Enter image name")
    #roll_no= simpledialog.askstring(title="Roll_No",
prompt="What's Your Roll Number:")
    #mail_id = simpledialog.askstring(title="Mail_Id",
prompt="What's your Mail Id:")
    NewVehicle_Capture.captureImage(name)
    imname = name+'.jpg'
    txt_area.insert(tk.INSERT, "Image Captured Successfully")


def start():
    window=tk.Tk()
    window.title("ALPR System")
    window.geometry("600x600")
    helv36 = tkFont.Font(family='Helvetica', size=20,
weight=tkFont.BOLD)
    button1 = tk.Button(window,text="Capture Image",
width=20,height=2,bg="blue",fg="yellow", command=capture,
font=helv36)
    button1.grid(row=1,column=0)

    label1= tk.Label(window,text = "Status").place(x =10,
 y = 250)
    global txt_area
    txt_area= tk.Text(window, height=5, width=50)
    txt_area.place(x =10, y = 275)

    #button1.pack()
    #button2.pack()
    window.mainloop()
```

3.3.5 NewVehicle_Capture.py

```
from cv2 import cv2
import os

path='C:/Users/New/Desktop/ALPR Project'
def captureImage(name):
    vc=cv2.VideoCapture(0)
    while(True):
        check,frame=vc.read()
        #print(frame)
        frame=cv2.flip(frame,1)
        #print("hi")
        gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
        cv2.imshow('New Enrollment( Enter C to capture)',
frame)

        key=cv2.waitKey(1)
        if key==ord('c'):
            cv2.imwrite(os.path.join(path,name+'.jpg'),frame)
            print("success")
            break
    vc.release()
    cv2.destroyAllWindows()
```

Chapter 4

OUTPUT SCREENS

4.1 Testing Results

4.1.1 Test case ID: 1

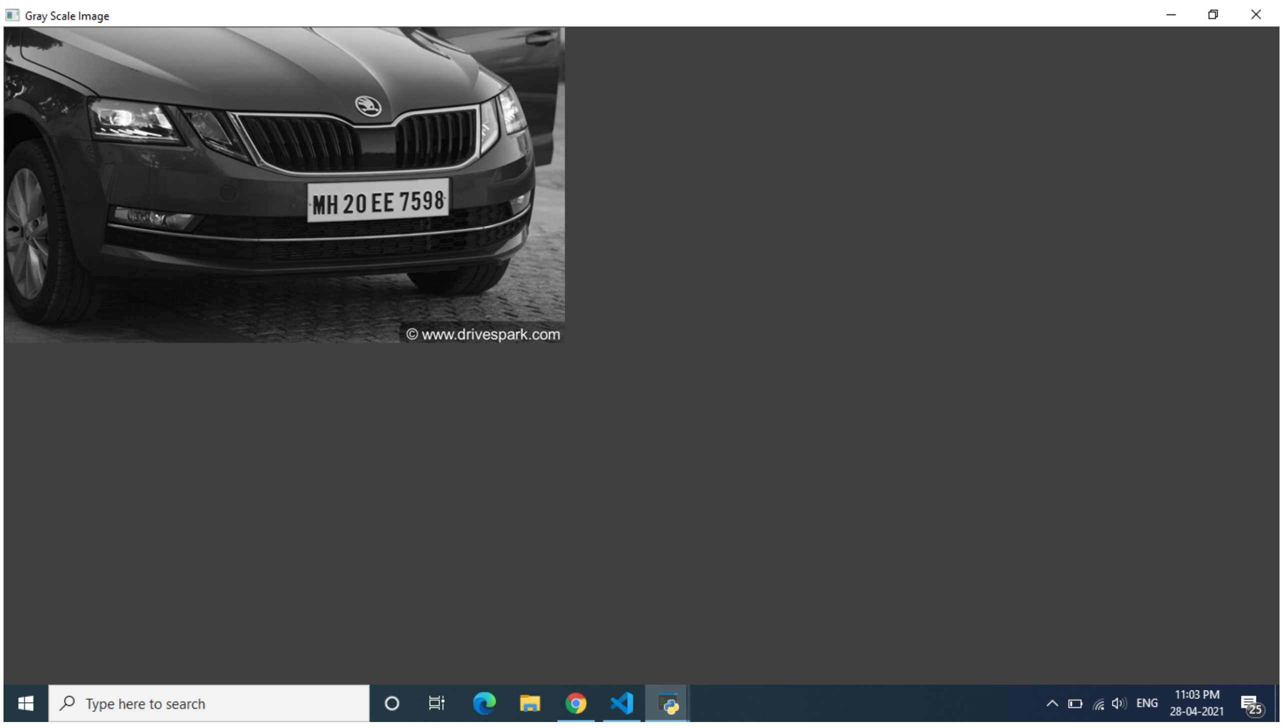
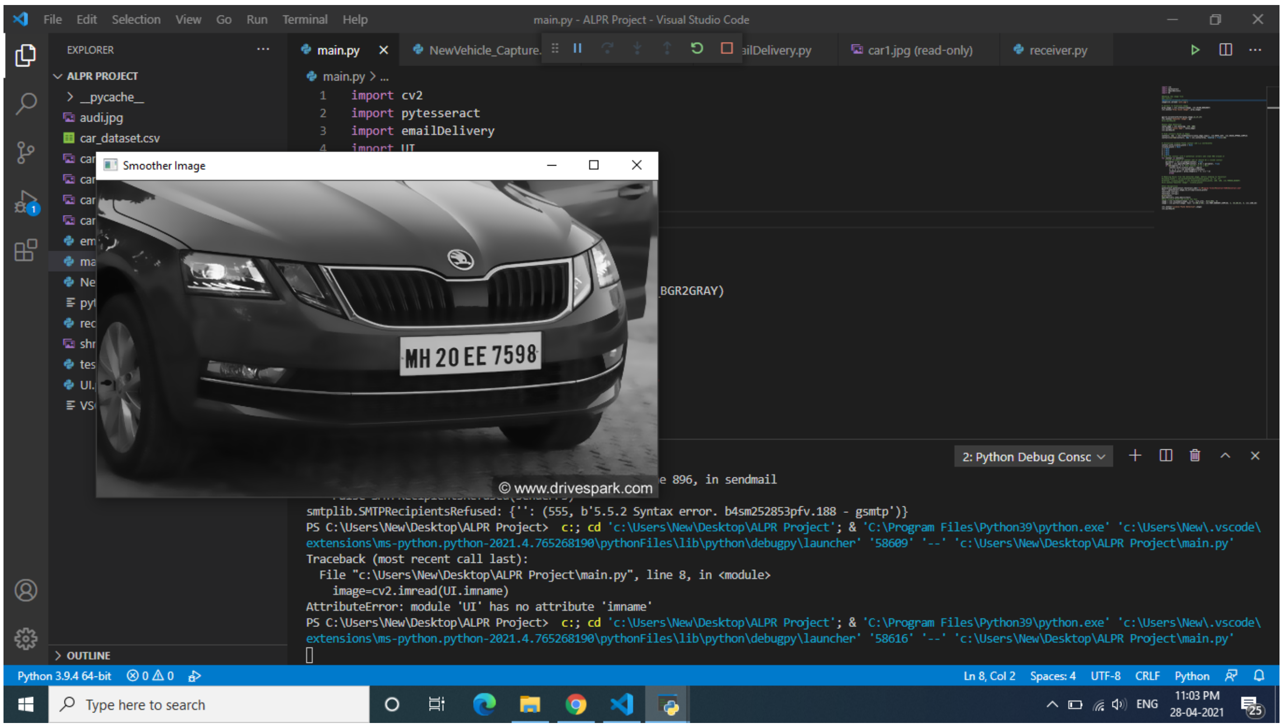


Figure 4.1: Gray scale Image

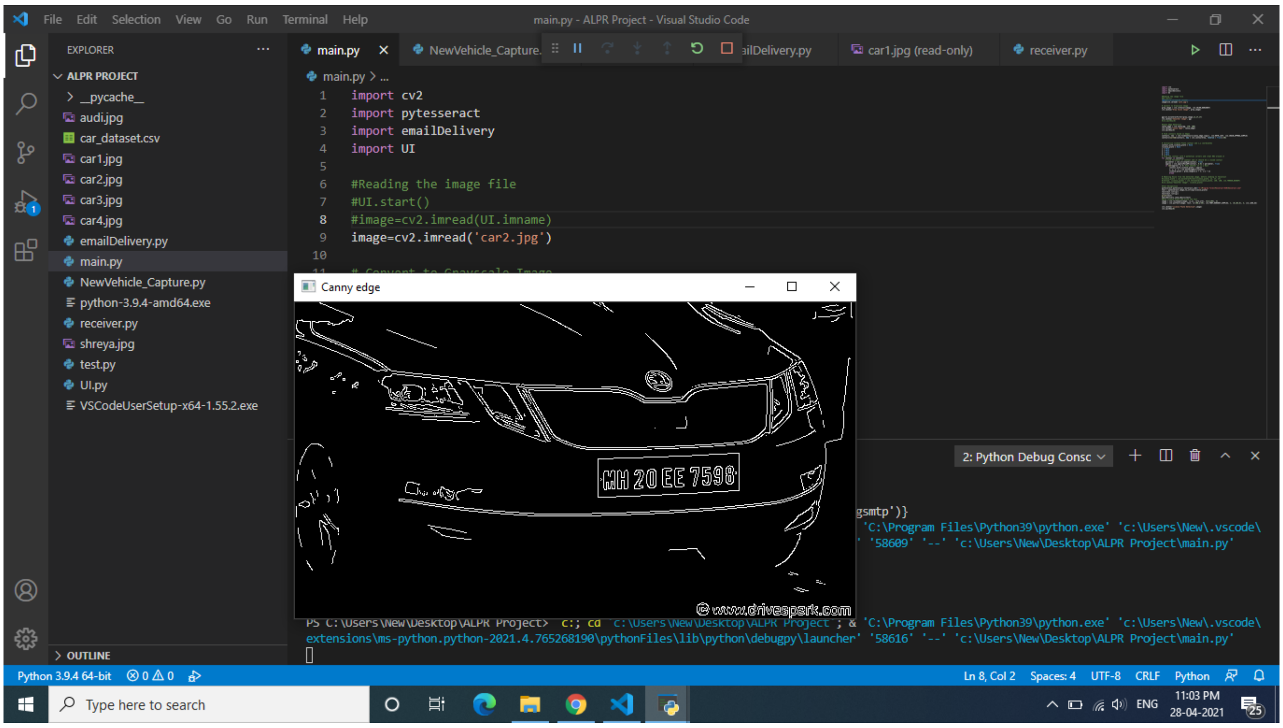


```
import cv2
import pytesseract
import emailDelivery
import UI
```

```
BGR2GRAY)
```

```
e 896, in sendmail
smtplib.SMTPRecipientsRefused: {'': (555, b'5.5.2 Syntax error. b4sm252853pfv.188 - gsmtp')}
PS C:\Users\New\Desktop\ALPR Project>  c:; cd 'c:\Users\New\Desktop\ALPR Project'; & 'C:\Program Files\Python39\python.exe' 'c:\Users\New\.vscode\extensions\ms-python.python-2021.4.765268190\pythonFiles\lib\python\debugpy\launcher' '58609' '--' 'c:\Users\New\Desktop\ALPR Project\main.py'
Traceback (most recent call last):
  File "c:\Users\New\Desktop\ALPR Project\main.py", line 8, in <module>
    image=cv2.imread(UI.imname)
AttributeError: module 'UI' has no attribute 'imname'
PS C:\Users\New\Desktop\ALPR Project>  c:; cd 'c:\Users\New\Desktop\ALPR Project'; & 'C:\Program Files\Python39\python.exe' 'c:\Users\New\.vscode\extensions\ms-python.python-2021.4.765268190\pythonFiles\lib\python\debugpy\launcher' '58616' '--' 'c:\Users\New\Desktop\ALPR Project\main.py'
```

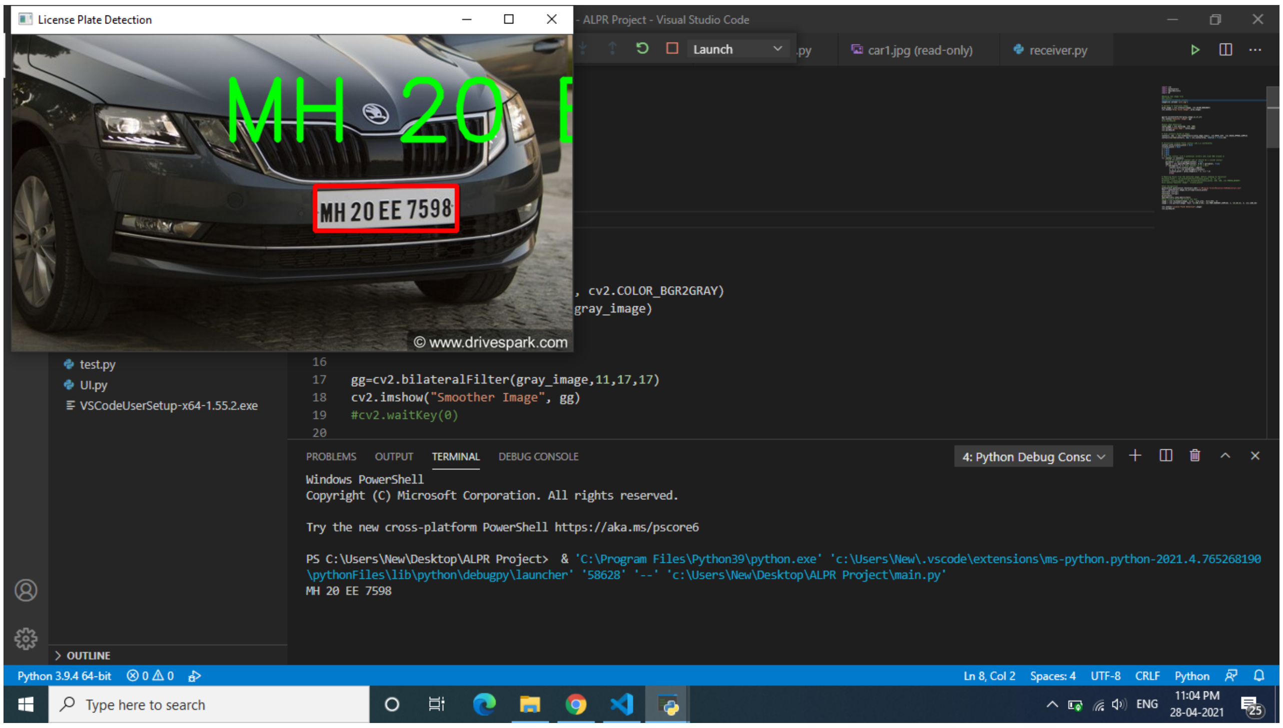
Figure 4.2: Smoothened image



```
import cv2
import pytesseract
import emailDelivery
import UI

#Reading the image file
#UI.start()
#image=cv2.imread(UI.imname)
image=cv2.imread('car2.jpg')
```

```
gsmtp')}
 'C:\Program Files\Python39\python.exe' 'c:\Users\New\.vscode\
' '58609' '--' 'c:\Users\New\Desktop\ALPR Project\main.py'

PS C:\Users\New\Desktop\ALPR Project>  c:; cd 'c:\Users\New\Desktop\ALPR Project'; & 'C:\Program Files\Python39\python.exe' 'c:\Users\New\.vscode\
extensions\ms-python.python-2021.4.765268190\pythonFiles\lib\python\debugpy\launcher' '58616' '--' 'c:\Users\New\Desktop\ALPR Project\main.py'
```

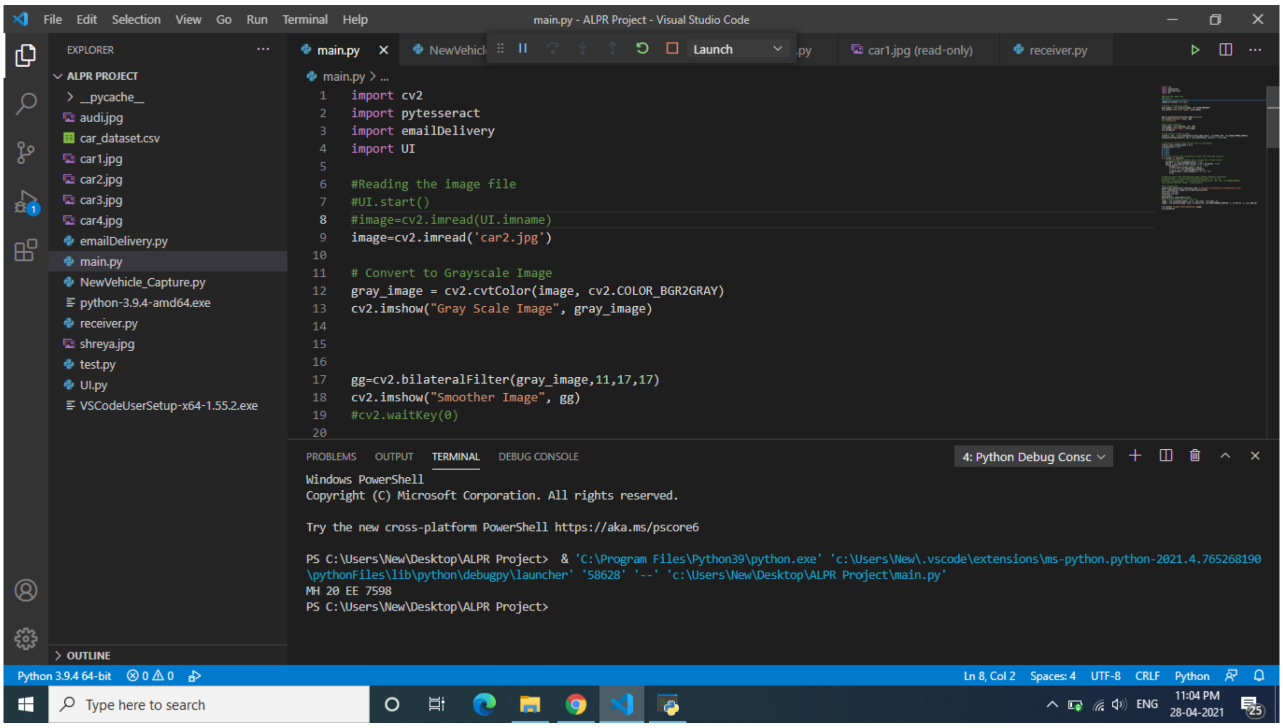
Figure 4.3: Image after Canny Edge detection



```
, cv2.COLOR_BGR2GRAY)
gray_image)

gg=cv2.bilateralFilter(gray_image,11,17,17)
cv2.imshow("Smoother Image", gg)
#cv2.waitKey(0)
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\New\Desktop\ALPR Project>  & 'C:\Program Files\Python39\python.exe' 'c:\Users\New\.vscode\extensions\ms-python.python-2021.4.765268190
\pythonFiles\lib\python\debugpy\launcher' '58628' '--' 'c:\Users\New\Desktop\ALPR Project\main.py'
MH 20 EE 7598
```

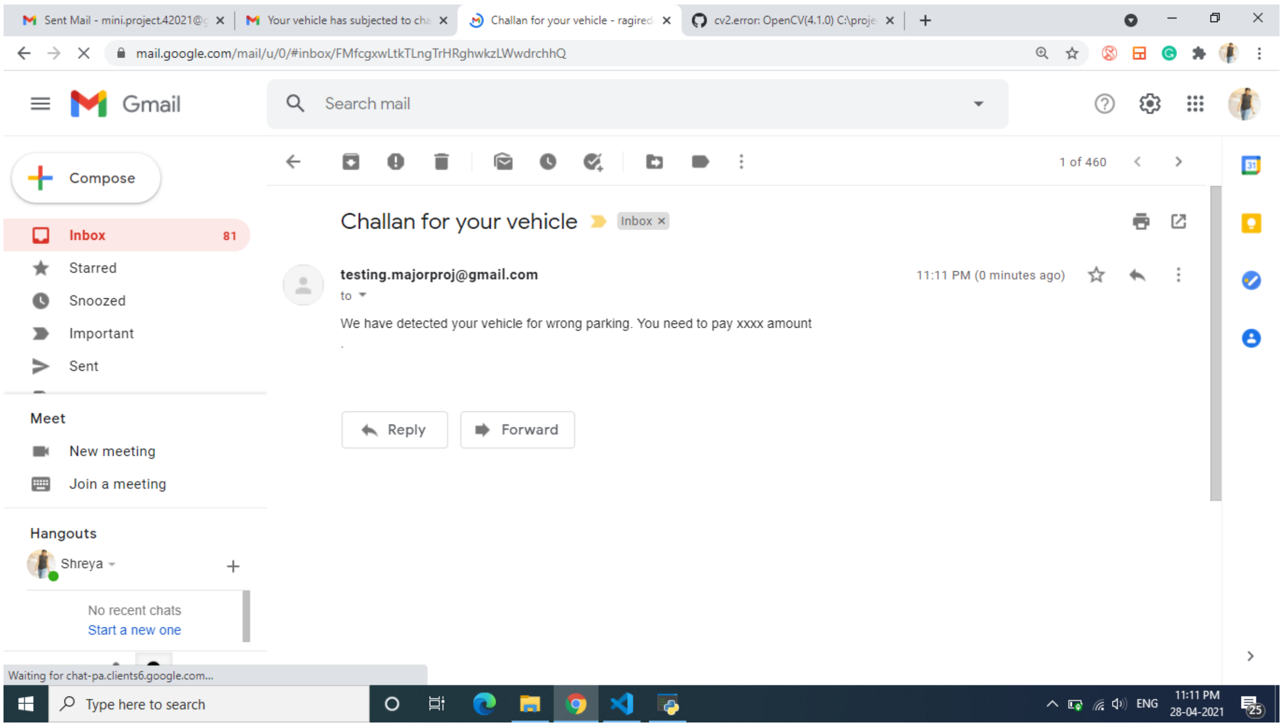
Figure 4.4: Detecting License Plate



```python
import cv2
import pytesseract
import emailDelivery
import UI

#Reading the image file
#UI.start()
#image=cv2.imread(UI.imname)
image=cv2.imread('car2.jpg')

# Convert to Grayscale Image
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("Gray Scale Image", gray_image)



gg=cv2.bilateralFilter(gray_image,11,17,17)
cv2.imshow("Smoother Image", gg)
#cv2.waitKey(0)
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\New\Desktop\ALPR Project>  & 'C:\Program Files\Python39\python.exe' 'c:\Users\New\.vscode\extensions\ms-python.python-2021.4.765268190
\pythonFiles\lib\python\debugpy\launcher' '58628' '--' 'c:\Users\New\Desktop\ALPR Project\main.py'
MH 20 EE 7598
PS C:\Users\New\Desktop\ALPR Project>
```

Figure 4.5: Output in String format


Sent Mail - mini.project.42021@
Your vehicle has subjected to cha
Challan for your vehicle - ragired
cv2.error: OpenCV(4.1.0) C:\proje
mail.google.com/mail/u/0/#inbox/FMfcgxwLtkTLngTrHRghwkzLWwdrchhQ
Gmail
Search mail
Compose
Inbox 81
Starred
Snoozed
Important
Sent
Meet
New meeting
Join a meeting
Hangouts
Shreya
No recent chats
Start a new one
1 of 460
Inbox
Reply
Forward
Waiting for chat-pa.clients6.google.com...
Type here to search
ENG
11:11 PM
28-04-2021

Challan for your vehicle

testing.majorproj@gmail.com

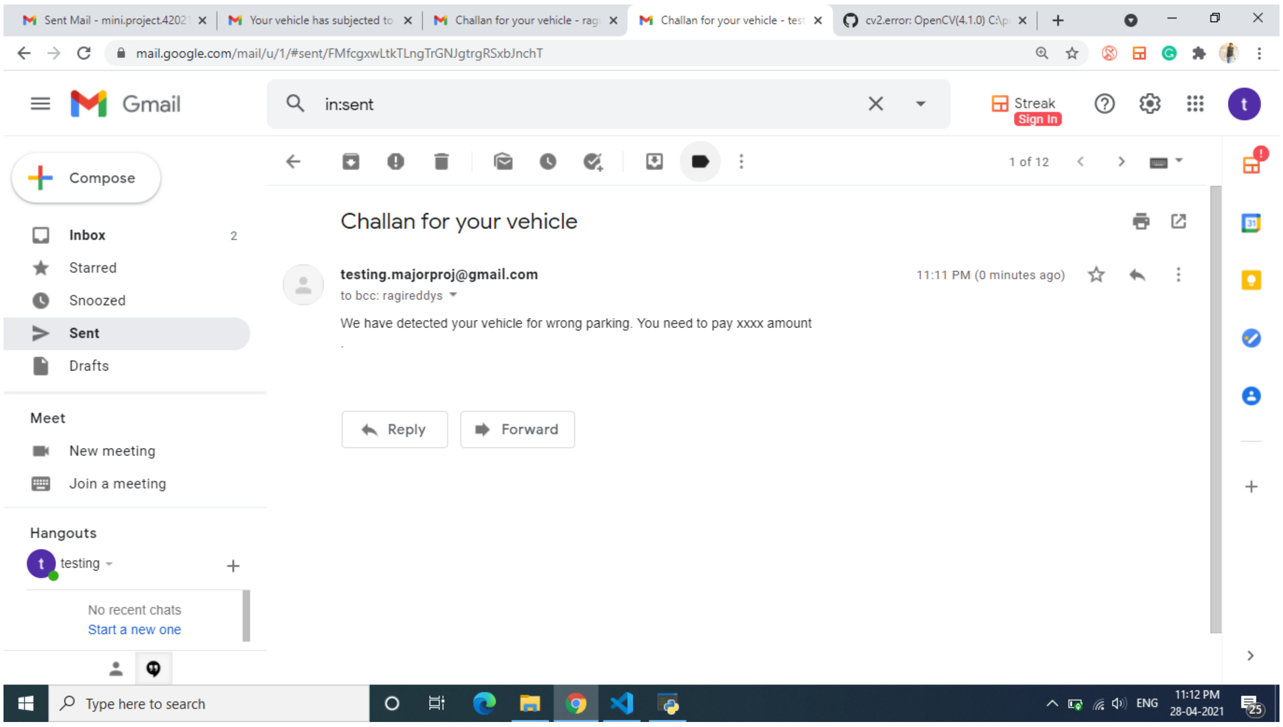
to

11:11 PM (0 minutes ago)

We have detected your vehicle for wrong parking. You need to pay xxxx amount

.

Figure 4.6: Email sent from sender



Challan for your vehicle

testing.majorproj@gmail.com 11:11 PM (0 minutes ago)
to bcc: ragireddys

We have detected your vehicle for wrong parking. You need to pay xxxx amount

Figure 4.7: Email received by receiver

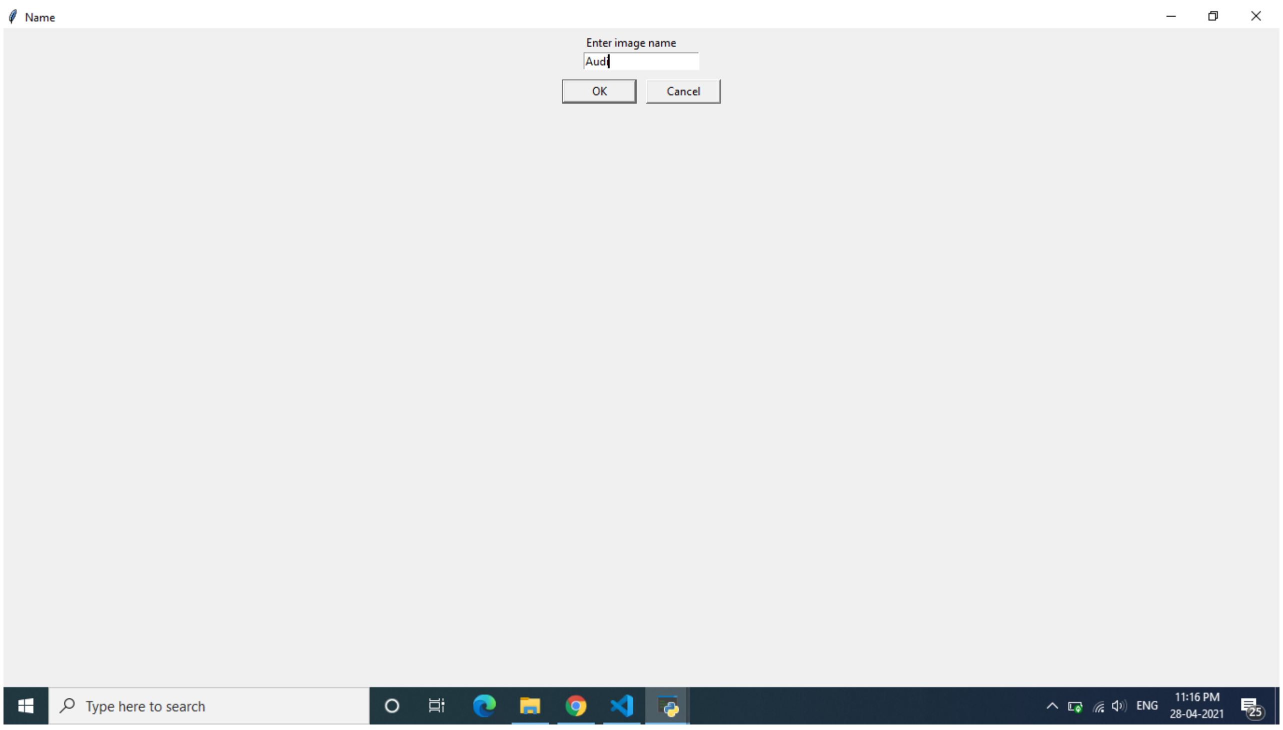


Figure 4.8: Entering car name

CONCLUSION AND FUTURE WORK

To conclude we present a code implemented in python that consists of the following in-built methods like Canny edge detection, Finding Contours, etc. Our experimental results document that there is a significant benefit of making extensive use of the existing algorithms combined in a different way to constrain the people detection problem and to extract relevant scene information. The system is capable of detecting and recognising license plates. Therefore we are, for example, able to detect license plates as well as sending them email notifications . The System proposed in this report works using the most popular software language and most effective image processing tools. Thus, making it even more efficient to work with.Though this model is implemented in a traffic scene, its domain can further be extended to a more higher level. This can be extended to busiest highways ,near movie-theatre. This model based approach will also be able to make effective use of spatial context which will enable the system to detect certain events automatically. Hence, this model has a great scope and potential in future, and will definitely be a win situation when implemented.

REFERENCES

1] Siebel, N.; Maybank, S., “Fusion of Multiple Tracking Algorithms for Robust People Tracking” ECCV 2002, pp.373–387, 2002.

[**2**]OpenCV change logs: http://code.opencv.org/projects/opencv/wiki/ChangeLog

[**3**] OpenCV Developer Site: http://code.opencv.org OpenCV User Site: http://opencv.org/

[**4**] ”Intel Acquires Computer Vision for IOT, Automotive — Intel Newsroom”. Intel Newsroom. Retrieved 2018-11-26.

[**5**]Subject: IDLE 0.2 – Integrated DeveLopment Environment for Python, From: Guido van Rossum, Date: Fri, 8 Jan 1999 17:35:25 GMT Also from the Help About screen

APPENDIX

Python Installation:

Python: Version 3.8.0:

The Python download requires about 25 Mb of disk space; keep it on your machine, in case you need to re-install Python. When installed, Python requires about an additional 90 Mb of disk space.

Downloading:

1. Goto https://www.python.org/downloads/
2. Click the **Download Python 3.8.0** button.

The file named **python-3.8.0.exe** should start downloading into your standard download folder. This file is about 30 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 10 seconds over a cable modem).
The file should appear as:

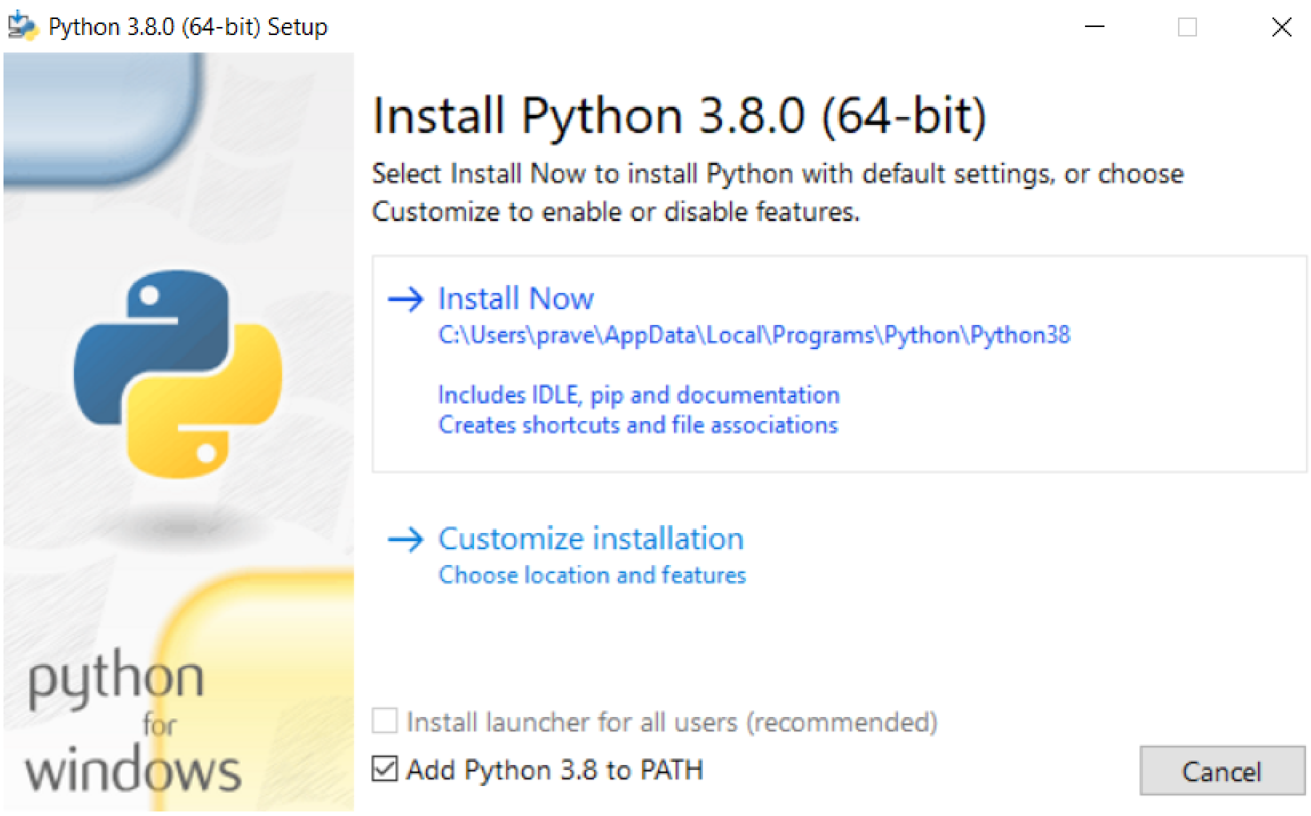
python-3.8.0.exe.

3. Move this file to a more permanent location, so that you can install Python (and reinstall it easily later, if necessary).
4. Feel free to explore this webpage further; if you want to just continue the installation, you can terminate the tab browsing this webpage.
5. Start the **Installing** instructions directly below.

Installing:

1. Double-click the icon labeling the
file **python-3.8.0.exe.**

2.A **Python 3.8.0 (64-bit) Setup pop-up** window will appear.


Python 3.8.0 (64-bit) Setup

Install Python 3.8.0 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:\Users\prave\AppData\Local\Programs\Python\Python38

Includes IDLE, pip and documentation
Creates shortcuts and file associations

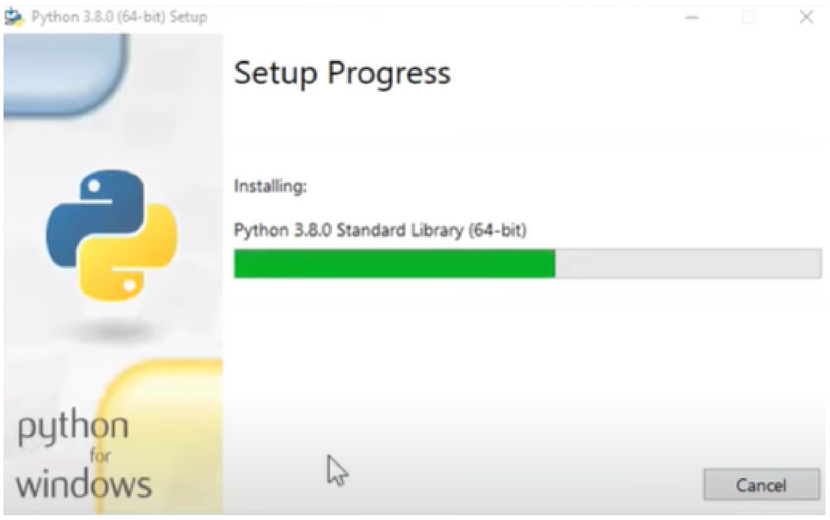
→ Customize installation
Choose location and features

☐ Install launcher for all users (recommended)
☑ Add Python 3.8 to PATH

Ensure that the **Install launcher for all users (recommended) and the Add Python 3.8 to PATH checkboxes** at the bottom are checked. If the **Python Installer** finds an earlier version of Python installed on your computer, the Install Now message may instead appear as **Upgrade Now** (and the checkboxes will not appear).

3. Highlight the **Install Now (or Upgrade Now) message**, and then click it.

4. A new **Python 3.8.0 (64-bit) Setup pop-up** window will appear with a **Setup Progress** message and a progress bar.

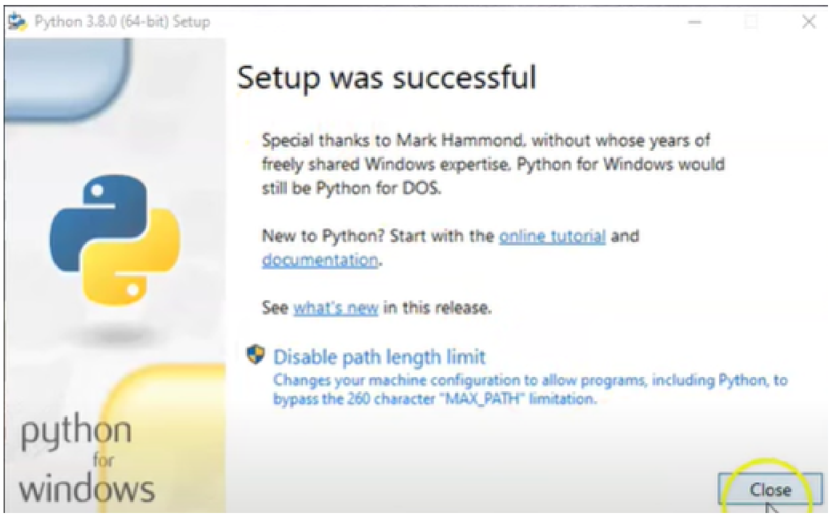


Setup Progress

Installing:

Python 3.8.0 Standard Library (64-bit)

During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new **Python 3.8.0 (64-bit) Setup** pop-up window will appear with a **Setup was successfully** message.



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

5. Click the **Close** button.

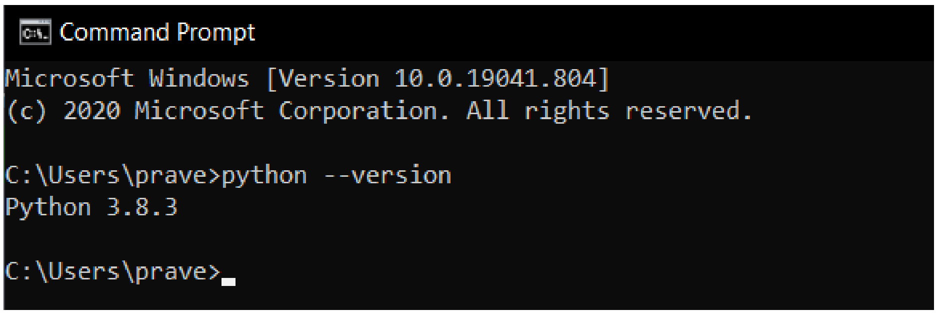
Python should now be installed.

Verifying:

To try to verify installation,

1. Navigate to the command prompt// Run the command "python –

version" . If the installation is successfull, installed python version will be displayed as shown



```
Command Prompt
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\prave>python --version
Python 3.8.3

C:\Users\prave>
```

Required Libraries Installation:

All the required libraries can be installed using pip installer

1. Go to command prompt and run the following commands
2. pip install opencv-python
3. pip install pandas
4. pip install tkinter
5. pip install pytesseract

If the results are printed out without any errors, you have installed all libraries successfully.