OVER SPEED VEHICLE DETECTION

A Major Project report submitted in partial fulfilment of the requirements for the award of the Degree of **Bachelor of Technology** in
Computer Science & Engineering (CSE)

By
K. Deepashikha (18011P0503)
D. Vamshi (18011U0501)
P. Sai Bhargav (18011U0509)

Under the guidance of
Dr. O. B. V. Ramanaiah
Professor

Department of Computer Science and Engineering,
JNTUH University College of Engineering,
Kukatpally, Hyderabad - 500 085.

Department of Computer Science and Engineering,
JNTUH University College of Engineering,
Hyderabad - 500 085.

DECLARATION BY THE CANDIDATE

We**, Kadaru Deepashikha (18011P0503), Dantoori Vamshi (18011U0501)** and **Poosa Sai Bhargav (18011U0509)**, hereby declares that the project report entitled OVER SPEED VEHICLE DETECTION**,** carried out by us under the guidance of Dr**. O. B. V. Ramanaiah,** is submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**. This is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced/copied from any source.

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

K.Deepashikha(18011P0503),
D.Vamshi(18011U0501),
P.Sai bhargav(18011U0509).

Department of Computer Science and Engineering,
JNTUH University College of Engineering,
Hyderabad - 500 085.

CERTIFICATE BY THE SUPERVISOR

This is to certify that the project report entitled **Over Speed Vehicle Detection,** being submitted by **Kadaru Deepashikha(18011P0503), Dantoori Vamshi (18011U0501) and Poosa Sai Bhargav (18011U0509)** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of bonafide work carried out by them. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

Date:

Dr.O.B.V. Ramanaiah

Professor

Department of Computer Science and Engineering,
JNTUH University College of Engineering,
Hyderabad - 500 085.

CERTIFICATE BY THE HEAD OF THE DEPARTMENT

This is to certify that the project report entitled **Over Speed Vehicle Detection,** being submitted by **Kadaru Deepashikha (18011P0503)**, **Dantoori Vamshi (18011U0501) and Poosa Sai Bhargav(18011U0509),** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of bonafide work carried out by them.

Dr. D. Vasumathi,
Professor & Head.

Date:

ACKNOWLEDGEMENTS

I would like to express sincere thanks to our guide **Dr. O.B.V.Ramanaiah** , Professor of Computer Science and Engineering Department, JNTUH University College of Engineering Hyderabad for his admirable guidance and inspiration both theoretically and practically and most importantly for the drive to complete the project successfully. Working under such an eminent guide was our privilege.

I owe a debt of gratitude to **Dr. D.Vasumathi**, Professor & Head of the department of Computer Science & Engineering, for her kind considerations and encouragement in carrying out this project successfully.

I express thanks to our parents for their love, care and moral support without which we would have not been able to complete this project. It has been a constant source of inspiration for all our academic endeavours. Last but not the least, we thank the Almighty for making us a part of the world.

Kadaru Deepashikha (18011P0503),
Dantoori Vamshi (18011U0501),
Poosa Sai Bhargav(18011U0509).

ABSTRACT

Road accidents have been very common in the present world with the prime cause being Over speeding. The necessity to check this has been very essential and different methods have been used so far . One of them is using handheld guns by police personnel that allow them to check car speed and then inform authorities about the vehicle manually.There has to be a system which automates the task in detection of overspeed vehicles.

Vehicle tracking is the process of locating a moving vehicle using a camera. Capturing vehicles in video sequence from surveillance cameras is a demanding application to improve tracking performance. As Over speeding is a major cause for high rate of road accidents, Vehicle tracking is a must to detect overspeed automobiles.

This project “OVERSPEED VEHICLE DETECTION” is aimed to build an automatic system that can accurately localise and track speed of any vehicle that appears in areal video frames and detect over speed vehicles, which does not need any human intervention. The tool is designed using python and opencv.

CONTENTS

Abstract I
Contents II
List of Figures III

Chapter 1 Introduction 1

1.1 Objective 2
1.2 Motivation 2
1.3 Problem Definition 3
1.4 Report Organization 3

Chapter 2 Requirements Specification 5

2.1 Functional Requirements 6
2.2 Non-Functional Requirements 6
2.3 System Requirements 7
2.3.1 Hardware Requirements 7
2.3.2 Software Requirements 7
2.4 Main concepts 7

Chapter 3 Technologies Used 11

3.1 Python 12
3.2 Flask 13
3.3 OpenCV 13
3.4 Dlib 15

Chapter 4 Implementation 17

Chapter 5 Design 19

5.1 Flow chart of the process 20
5.2 UML Diagrams 21
5.2.1 Use case Diagram 21
5.2.2 Sequence Diagram 22
5.2.3 Activity diagram 23

Chapter 6 Results **24**
6.1 Screenshots 25

Chapter 7 Conclusion **30**
7.1 Work Carried Out 31
7.2 Scope for Further Work 31

References **32**

LIST OF FIGURES

Figure 4.1 Flow Chart Diagram 20
Figure 4.2 Use Case Diagram 21
Figure 4.3 Sequence Diagram 22
Figure 4.4 Activity Diagram 23
Figure 5.1 Initial Screen 25
Figure 5.2 Speed Detection(Standard data) 26
Figure 5.3 Speed Detection 27
Figure 5.4 Details of Over speed Vehicle 28
Figure 5.5 Alternative Output Screen 29

Chapter 1: Introduction

1.1 Objective

Over speeding is the major cause for high rate of road accidents.Hence vehicle tracking is must to detect over speed automobiles.This project is aimed at proposing a system to detect the vehicle which are being driven above the given maximum speed limit of that particular area or zone.

1.2 Motivation

Road accidents are the most unwanted thing to happen to a road user, though they happen quite often. Main cause of road accidents and crashes are due to human errors, especially due to Over speeding. Faster vehicles are more prone to accidents than slower one and the severity of accidents will also be more in case of faster vehicles than the slower ones. The reason behind this is that slower vehicles come to halt immediately while faster one take a long way to stop and also skids a long distance and can lead to a major accident.

So, detection of over-speeding vehicles is a must to avoid road accidents.Current speed detection systems are handheld guns held by police personnel that allow them to check car speed and then manually inform authorities about the vehicle.But there needs to be a system which records the overspeed vehicles by tracking them continuously.

This project “OVER SPEED VEHICLE DETECTION” is one such thing which can accurately localize and track speed of any vehicle that appears in aerial video frames.

1.3 Problem Definition

Road accidents occurrences have increased recently and the major cause for this is over speeding of vehicles.The necessity to check this has been very essential and different methods have been used so far. However with the advancement in technology, different governing bodies are demanding some sort of computerized technology to control this problem of overspeed driving.

The aim is to build an automatic system which can track speed of any vehicle that appears in video frames and detect over speed automobiles. The tool is designed using python and opencv.

1.4 Report Organization

Chapter 1: Introduction deals with presenting the objective of our application along with the motivation behind this idea and giving the problem definition.

Chapter 2: Requirements Specification gives an in depth look into the functionalities both functional and non-functional, expected of the application that is proposed to develop.

Chapter 3: Technologies Used gives a brief description of the various hardware and software tools used in order to develop the application of interest with all the features as stated as part of functional requirements.

Chapter 4: Design gives an outlook on the software modularity and architecture being followed while developing the application.

Chapter 5: Implementation gives detailing about the code developed in order to reach the requirements specified.

Chapter 6: Results are populated with the screenshots which portrait significant features of our application.

Chapter 7:Conclusion gives the closing note on the work done and gives information about the inspirational areas for carrying further work.

Chapter 2: Requirements Specification

2.1 Functional Requirements

Our application is developed to detect over speeding vehicles. It eases the task of charging penalties to the public so that they can be more responsible towards traffic rules.

Once the application starts, the user needs to input the CCTV video and speed limit value.

The tool detects all the cars in the video and calculates the speed of all detected cars.

The Number plate of the car will be detected, if the speed exceeds the specified speed limit.

Finally, the user can see the list of detected number plates of over speeding cars on the result page.

2.2 Non-Functional Requirements

Usability

Flexibility

Reliability

Availability

Maintainability

2.3 System Requirements

2.3.1 Hardware Requirements

Processor: Intel® Core™ i5-2430M CPU @2.40GHz
RAM: 4GB

GPU : 2GB(optional)

2.3.2 Software Requirements

Python, Web browser, Operating system(OS), Pycharm

2.4 Main Concepts

a. Object Detection :

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched Domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

Cascading is a particular case of ensemble learning based on the concatenation of several classifiers, using all information collected from the output from a given classifier as additional information for the next classifier in the cascade. Unlike voting or stacking ensembles, which are multi expert systems, cascading is a multistage one.

Cascading classifiers are trained with several hundred "positive" sample views of a particular object and arbitrary "negative" images of the same size. After the classifier is trained it can be applied to a region of an image and detect the object in question.[1]

To search for the object in the entire frame, the search window can be moved across the image and check every location with the classifier. This process is most commonly used in image processing for object detection and tracking, primarily facial detection and recognition.

By using the Cascade-training-GUI developed by Amin Ahmadi, a XML file containing features is generated (features include number of vehicles, the bounding rectangle coordinates, size of vehicle). CascadeClassifier is built using this XML file.Finally, detectMultiScale(image,scaleFactor,minNeighbours,(minSize,maxSize)) is called, which returns the list of bounding rectangles of each detected object.[1]

b.Object tracking :

Object tracking is an application of deep learning where the program takes an initial set of object detections and develops a unique identification for each of the initial detections and then tracks the detected objects as they move around frames in a video.In other words, object tracking is the task of automatically identifying objects in a video and interpreting them as a set of trajectories with high accuracy.

Object tracking is used for a variety of use cases involving different types of input footage.Whether or not the anticipated input will be an image or a video, or a real-time video vs. a prerecorded video, impacts the algorithms used for creating object tracking applications.The kind of input also impacts the category, use cases, and applications of object tracking.

Video tracking is an application of object tracking where moving objects are located within video information. Hence, video tracking systems are able to process live, real-time footage and also recorded video files.The processes used to execute video tracking tasks differ based on which type of video input is targeted.

Algorithms for tracking objects are supposed to not only accurately perform detections and localize objects of interest but also do so in the least amount of time possible.Enhancing tracking speed is especially imperative for real-time object tracking models.

Object tracking refers to the ability to estimate or predict the position of a target object in each consecutive frame in a video once the initial position of the target object is defined. To do that, we use **correlation_tracker()** class defined in the **DLIB library.** The overloaded method **start_tracker(image, prev_bounding rect)** is called to keep track of each and every detected object.

Using **correlation trackers in dlib**, you can track any object in a video stream without needing to train a custom object detector.The algorithm tracks the object by correlating filters on the search window in the next frame. The new position of the object is represented by the maximum value of associated output. Then performs an online update in the new location.

c. Speed Calculation:

- The distance travelled by the car is calculated by using Euclidean distance.
- Comparing the two successive frames in the video, the initial and final locations of the cars are determined.

- Distance travelled is calculated by applying Euclidean distance to the location coordinates.
- The pixel distance is converted into meters by dividing the value with pixel-per-metre(ppm) value.
- The time taken by each car is measured by using the python time module.
- Finally, the speed is calculated by dividing the distance and time values and then converted to kilometers per hour..

Chapter 3: Technologies Used

3.1 Python:

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

3.1.a Python Math module:

It is the standard math module in Python,which includes basic operations along with trigonometric,representation and logarithmic functions. **In this project pow,sqrt are used for distance calculations.[4]**

3.1.b Python Time Module:

It allows you to work with time in Python,i.e it allows functionality like getting the current time,pausing the program from executing,etc. **Time module time() function is used to determine the starting and ending time of each vehicle in a video frame.**[4]

3.1.c Python EasyOCR Module:

EasyOCR is a python module for extracting text from an image that comes with trained models for numerous languages. This makes it easy for the end-user to analyse individual pieces of texts, available as free and open-source applications.[4]

3.2 Flask:

It is a **small and lightweight python web framework** that provides useful tools and features that make creating web applications in python easier. It gives developers flexibility and is a more accessible framework for new developers since you can build a web application quickly using only a single Python file.[5]

It's a microframework, but that doesn't mean your whole app should be inside one single Python file.You can use many files for larger programs, to handle complexity.

Micro means that the Flask framework is simple but extensible. You may make all the decisions: which database to use, do you want an ORM etc, Flask doesn't decide for you. Flask is one of the most popular web frameworks, meaning it's up-to-date and modern. You can easily extend its functionality. You can scale it up for complex applications.

3.3 OpenCV:

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them.OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems.

By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To

Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.[2]

OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency.

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it. If we talk about the basic definition of image processing then "Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality".

An image may be defined as a two-dimensional function f(x, y), where x and y are spatial(plane) coordinates, and the amplitude of any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.

3.4 Dlib- Davis Library:

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge.[3]

Major features of dlib are machine learning algorithms(Structural SVM tools for object detection in images as well as more powerful (but slower) deep learning tools for object detection), numerical algorithms, image processing, Data Compression and Integrity Algorithms and General Utilities.

Chapter 4: Implementation

Methodology:

1. The user inputs the video and specifies the speed limit of the area.
2. The tool breaks the video into frames and stores it in the variable named image.

   ```
   rc, image = video.read()
   ```

3. The tool detects the cars by using a Cascade Classifier object named carCascade. Firstly, The image is converted to gray scale and then detectMultiScale() is used to detect the cars.

   ```
   gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

   cars = carCascade.detectMultiScale(gray, 1.1, 13, 18, (24, 24))
   ```

4. The tool keeps track of all the cars on the screen. It is done using the dlib library. The start_track() is used to keep track of all the bounding box coordinates of the cars

   ```
   tracker = dlib.correlation_tracker()

   tracker.start_track(image, dlib.rectangle(x, y, x + w, y + h))
   ```

5. The speed of the detected cars is calculated using Euclidean distance.

   ```
   d_pixels=math.sqrt(math.pow(location2[0]-location1[0],2)+math.pow(l
   ocation2[1] - location1[1], 2))

   ppm = 8.8

   d_meters = d_pixels / ppm

   fps = 18

   speed = d_meters * fps * 3.6
   ```

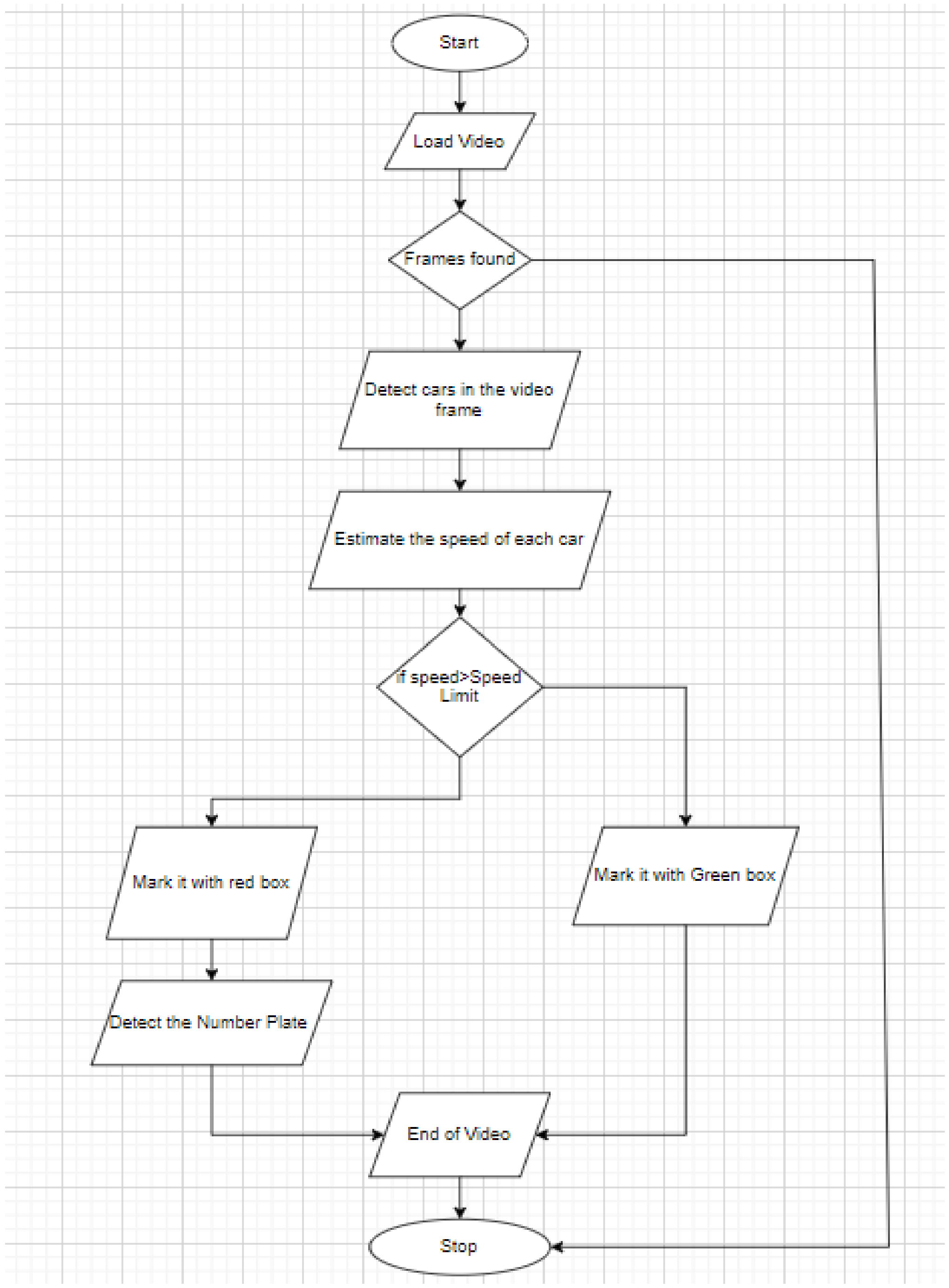
6. Finally, the Number plates of the overspeeding cars are detected and converted to text.

```
car_plates=carPlatesCascade.detectMultiScale(gray,scaleFactor=1.2,mi
nNeighbors = 5, minSize=(25,25))

reader = easyocr.Reader(['en'])

text[i] = reader.readtext(car_plates)
```

Chapter 5: Design

5.1 Flow chart of the process

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and here it is used to describe processes.



5.1 flow chart

5.2 UML Diagrams

The application can be diagrammatically depicted using the UML diagrams as below:

5.2.1 Use-Case Diagram:

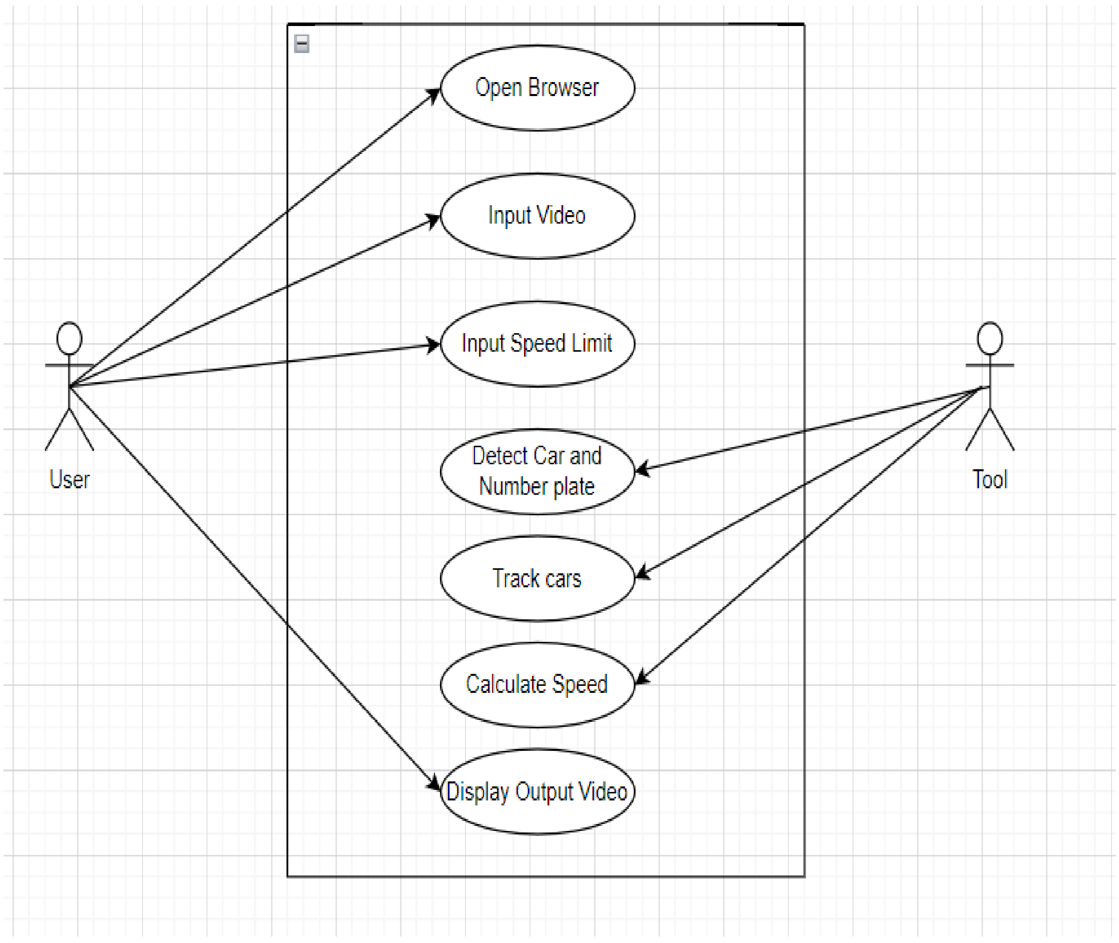


Figure 5.2 Use case Diagram

The use-case diagram depicts the set of actors involved and the set of actions they perform on the developed application. These elements can be represented as shown in Figure 5.2.

5.2.2 Sequence Diagram:

The sequence diagram depicts the message ordering of a set of messages exchanged between the actors and the modules. The sequence diagram of the message ordering of all the messages that could possibly be issued when using the application to illustrate all the modules Figure 5.2.

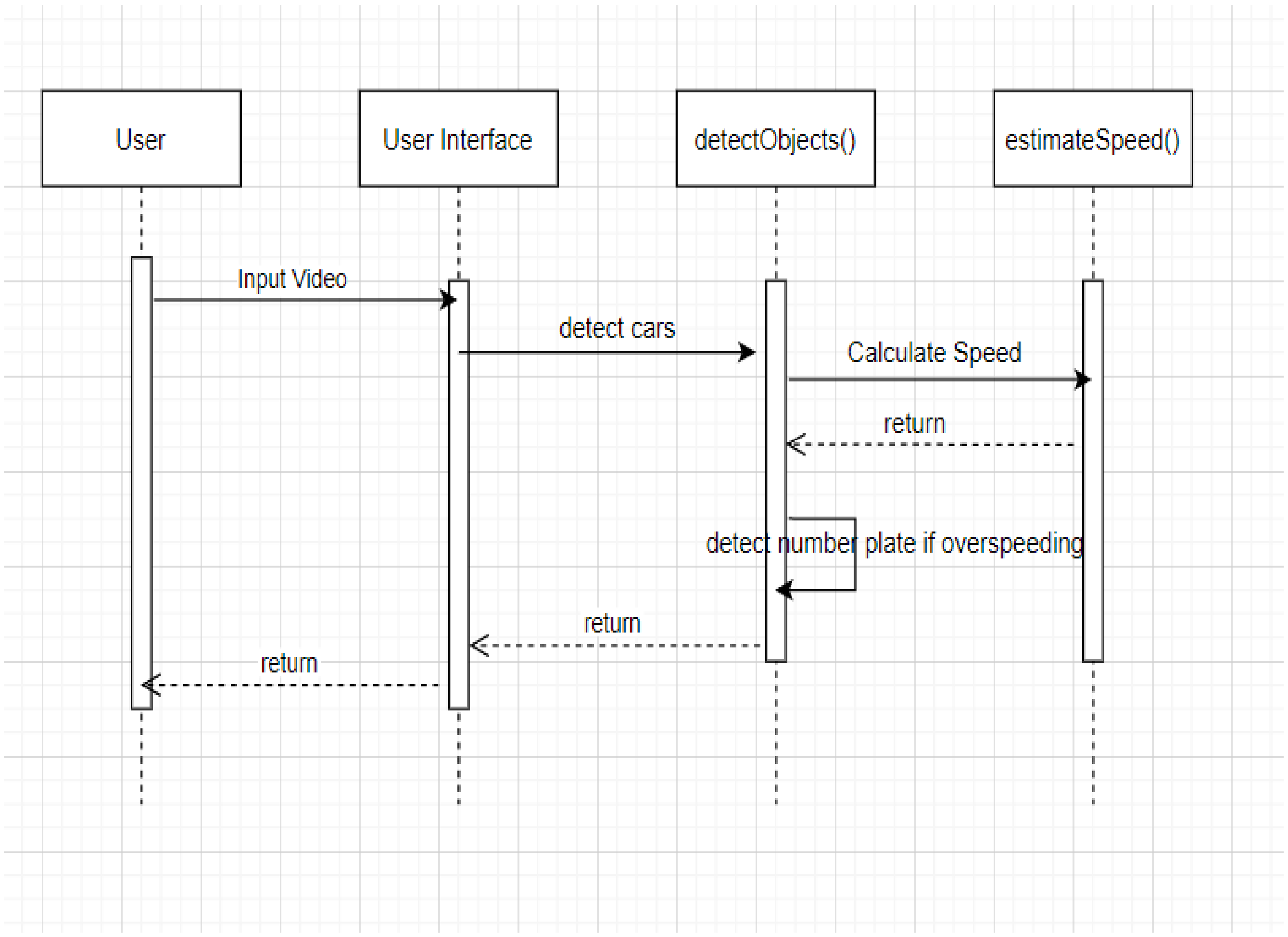
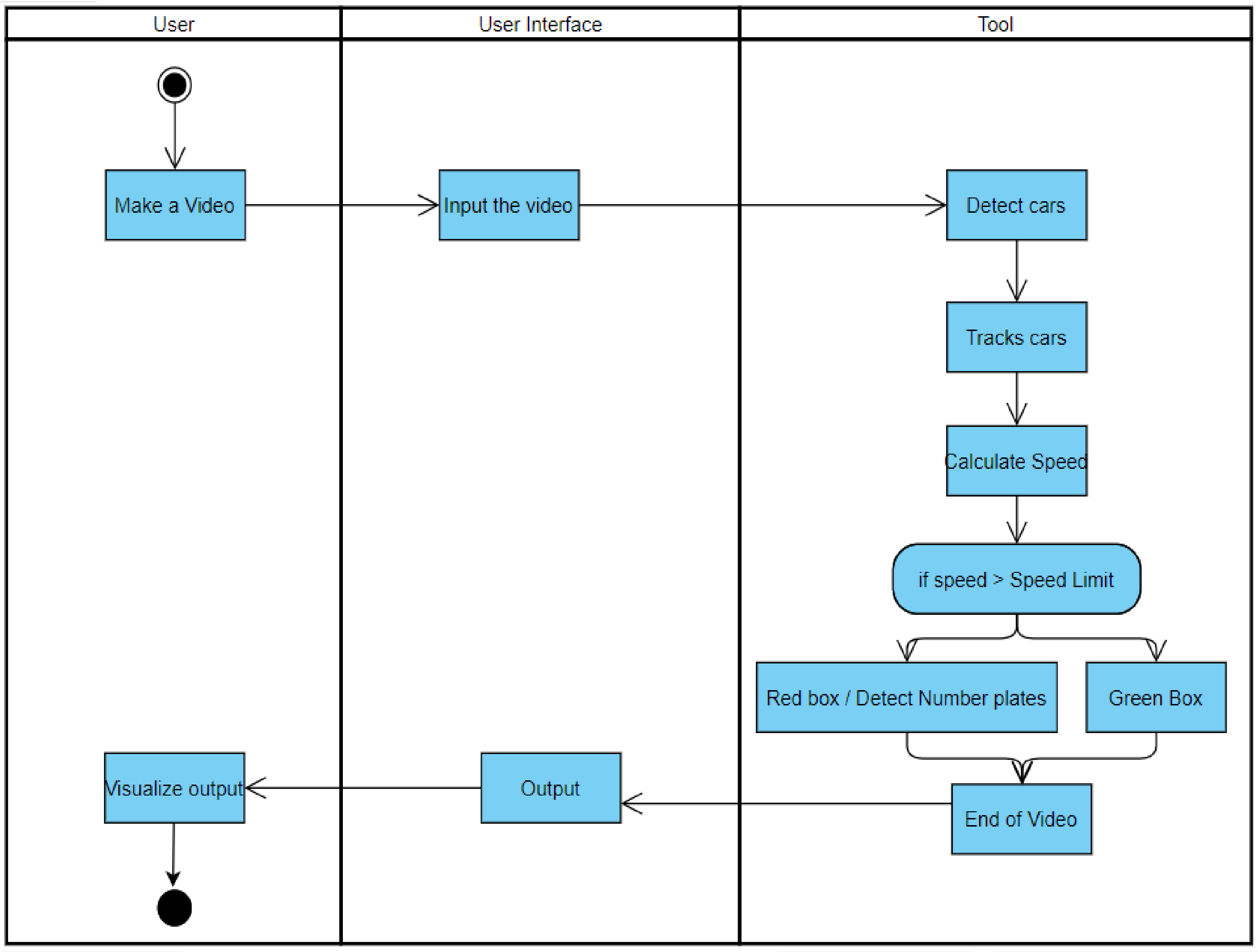


Figure 5.3 Sequence Diagram

5.2.3 Activity Diagram:

The activity can be described as an operation of the system.The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Here captures the dynamic behavior of the system.

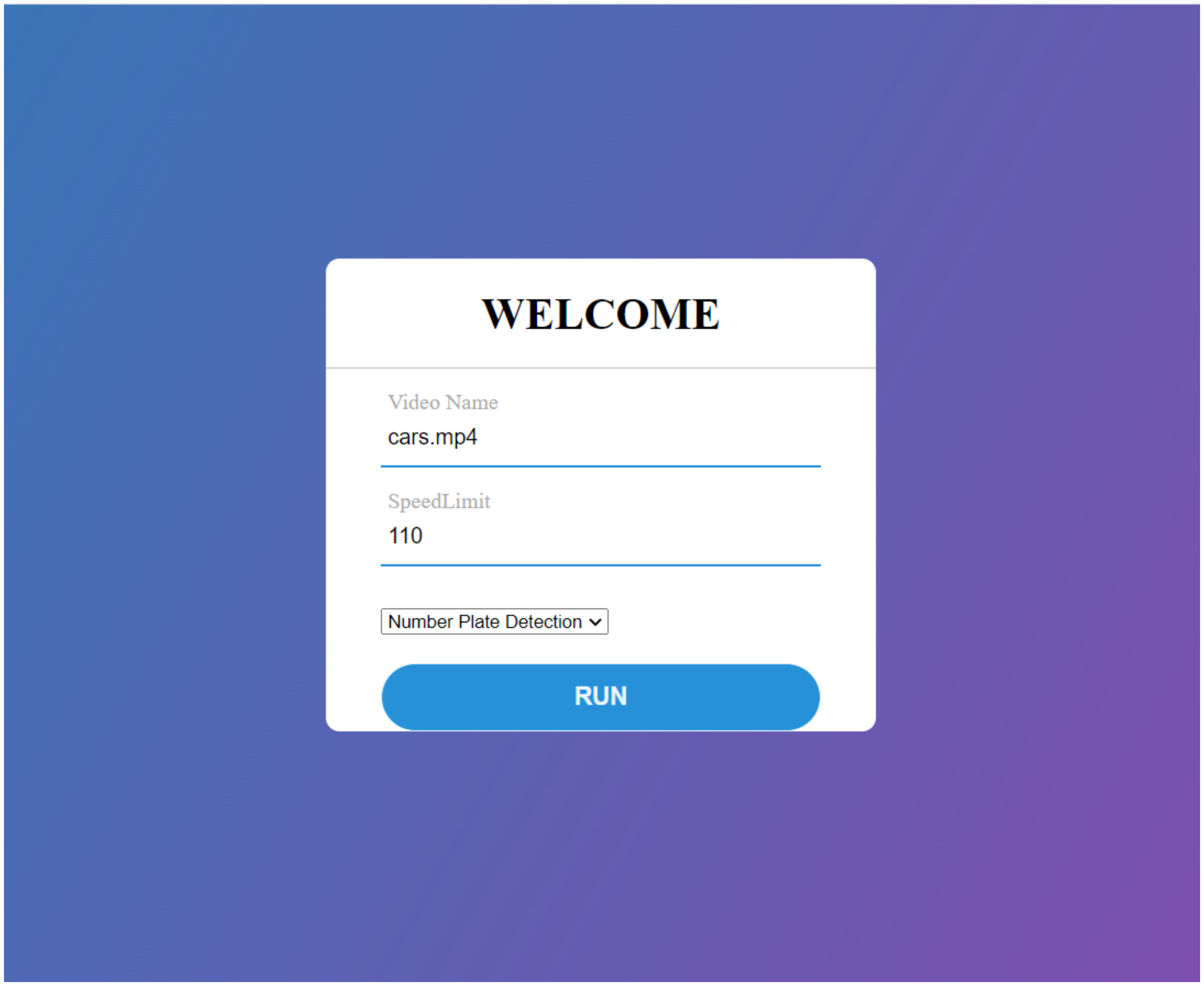


5.4 Activity diagram

Chapter 6: Results

6.1 Screenshots

When the application is started, it looks as shown in Figure 6.1.



Video Name
cars.mp4

SpeedLimit
110

Number Plate Detection

Figure 6.1 Initial Screen

The user interface contains video name i.e the file containing the vehicle input and speed limit of a particular area , provided by the user.

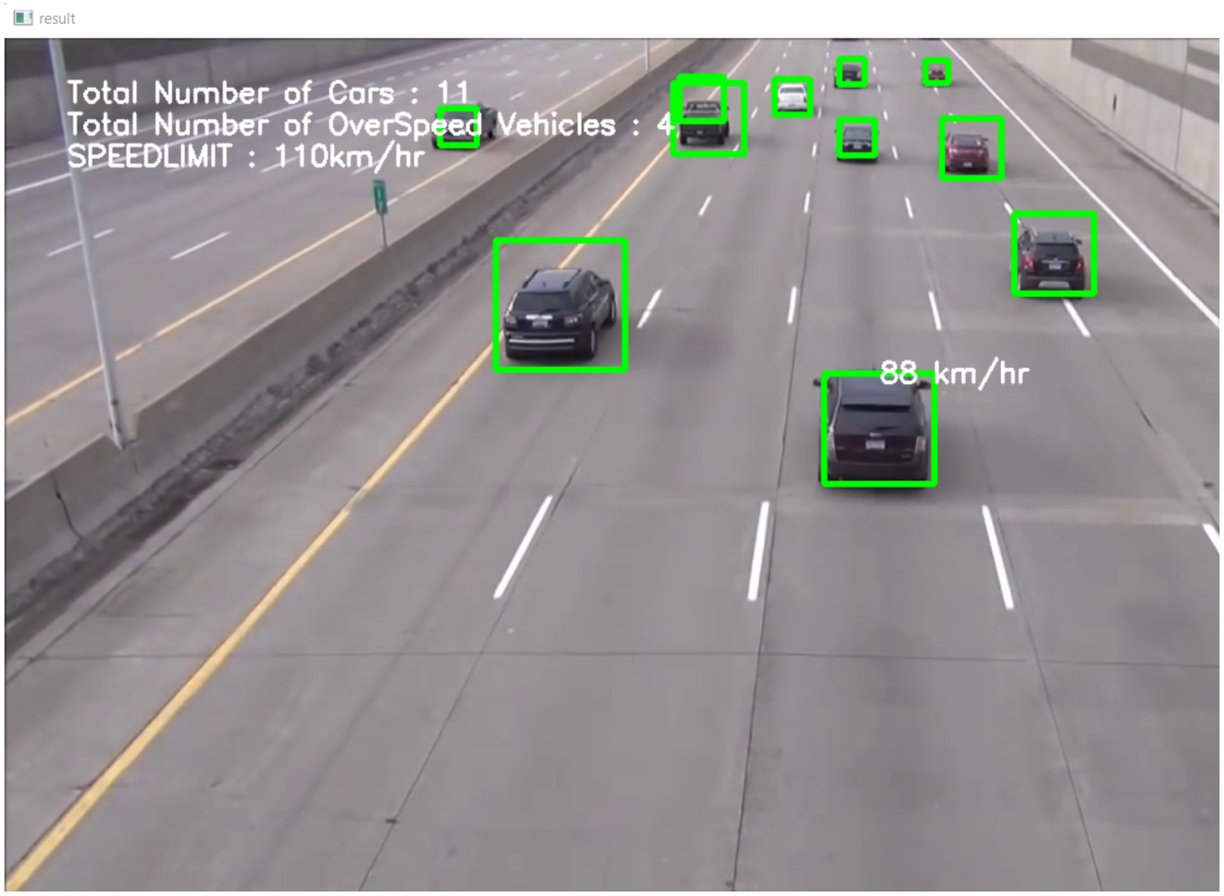

result
Total Number of Cars : 11
Total Number of OverSpeed Vehicles : 4
SPEEDLIMIT : 110km/hr
88 km/hr

Figure 6.2 overspeeded cars detection (standard data)

In the figure 6.2 , we can see detected cars and their speeds in green boxes. At the top left corner we see the total number of cars detected, total number of overspeed vehicles and speed limit of the area.

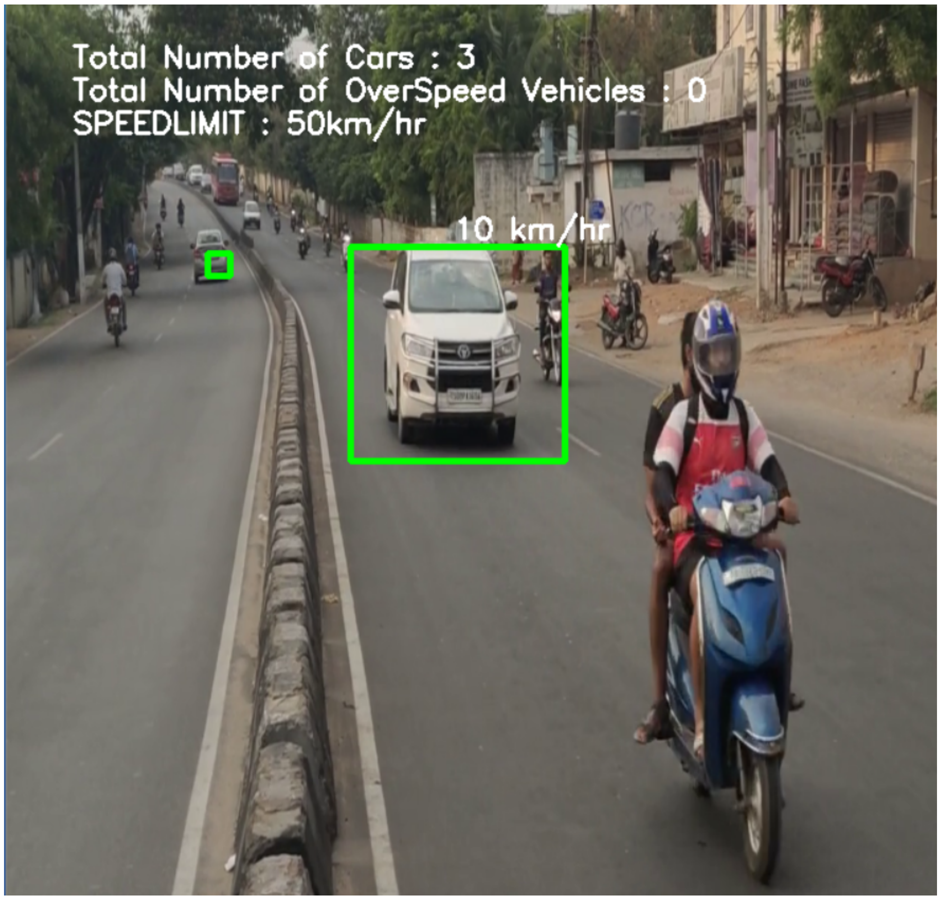
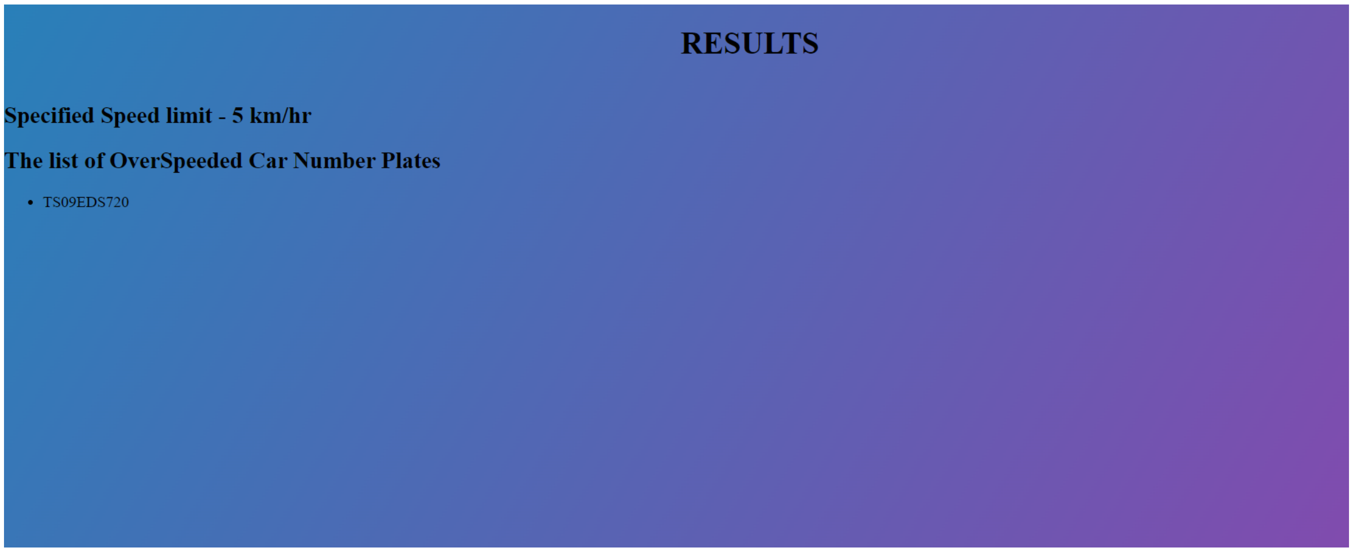


Figure 6.3 overspeeded cars detection(at our local area)

In the figure 6.3 we can see that the total count of overspeeding vehicles with a given speed limit as input is shown on the screen.

The results for the given input video are shown below.

1.



RESULTS

Specified Speed limit - 5 km/hr

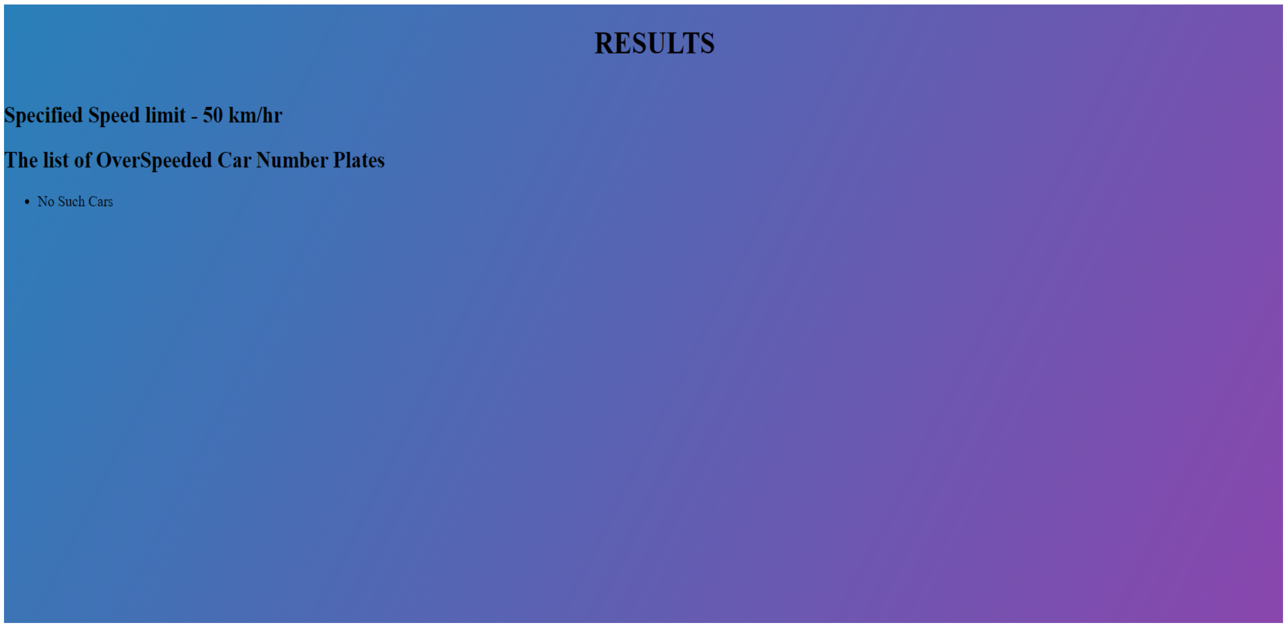
The list of OverSpeeded Car Number Plates

- TS09EDS720

Figure 6.4 Details of overspeeded vehicles

In the figure 6.4 we can see the Number plates of Over-speeded cars for the given speed limit of area are detected and are displayed on the screen.

2.



RESULTS

Specified Speed limit - 50 km/hr

The list of OverSpeeded Car Number Plates

- No Such Cars

Figure 6.5 Alternate output screen

In the figure 6.5 we can see the number plates of overspeed cars ,for the given speed limit of the area it shows that no such cars exceeding the speed limit are found.

Chapter 7: Conclusion

7.1 Work Carried Out

This application was built on the basis of requirements specified and the design. It aims at easing the tasks of the detection of over speed vehicles. It eases the task of charging penalties to the public. It has got required features which come handy and interest in the application.

7.2 Scope for Further Work

The application just forms a model exemplifying the importance of technology in detection of vehicles. This application can be further extended to various kinds of vehicles like motorbikes, trucks etc.

References:

[1] Cascade Classifier
https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html **- verified on 05-07-2022.**

[2] Opencv
https://docs.opencv.org/4.x/d9/df8/tutorial_root.html **- verified on 05-07-2022.**

[3] Dlib
http://dlib.net/ **- verified on 05-07-2022.**

[4] Python Modules
https://www.w3schools.com/python/python_modules.asp **- verified on 05-07-2022.**

[5] Flask
https://www.tutorialspoint.com/flask/index.htm **- verified on 05-07-2022.**