“Vehicle Detection and Speed Tracking”

A project report submitted to
Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal
in partial fulfillment for the award of
the degree of
Bachelor of Engineering
in
Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY
INDORE- 453331

2019-2020

“Vehicle Detection and Speed Tracking”

A project report submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

in partial fulfillment for the award of

the degree of

Bachelor of Engineering

in

Computer Science & Engineering

PROJECT GUIDE

Mr. Ankit Upadhyay
(Ass. Professor)

SUBMITTED BY

Animesh Jain (0829CS161020)
Brijesh Rathore (0829CS161035)
Harsh Verma (0829CS161048)
Himanshu Singh Patel (0829CS161049)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY
INDORE- 453331

2019-2020

ACKNOWLEDGEMENT

We express deep gratitude for enthusiasm and valuable suggestions that we got from our guide **Mr. Ankit Upadhyay,** Assistant professor, for successful completion of the Software Design Specification of the project. This was not possible without the invaluable guidance of our project guide

We are also thankful to our project coordinator **Mr. Ankit Upadhyay,** Assistant professor, for his technical guidance, encouragement and support.

We are deeply indebted to **Mrs. Reetu Gupta**, Head Department of Computer Science Engineering, for providing us support and resource for successful completion of the work.

We pay deep regards to our Director **Dr. N. Dagdee** who are instrumental in setting standards for the student to achieve.

Finally, we are thankful to all the people who are related to the project directly or indirectly.

Animesh Jain (0829CS161020)
Brijesh Rathore (0829CS161035)
Harsh Verma (0829CS161048)
Himanshu Singh Patel (0829CS161049)

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY
INDORE, 453331

CERTIFICATE

This is to certify that **Animesh Jain (0829CS161020)**, **Brijesh Rathore (0829CS161035)**, **Harsh Verma (0829CS161048)**, **Himanshu Singh Patel (0829CS161049)** have completed their project work, titled **“Vehicle Detection and Speed Tracking”** as per the syllabus and have submitted a satisfactory report on this project as a part of fulfillment towards the degree of **“BACHELOR OF ENGINEERING” (Computer Science & Engineering)** from **RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL.**

HEAD OF THE DEPARTMENT **PROJECT GUIDE**

DIRECTOR

SUSHILA DEVI BANSAL COLLEGE OF TECHNOLOGY
INDORE, 453331

CERTIFICATE

This is to certify that **Animesh Jain (0829CS161020)**, **Brijesh Rathore (0829CS161035)**, **Harsh Verma (0829CS161048)**, **Himanshu Singh Patel (0829CS161049)** have completed their project work, titled **“Vehicle Detection and Speed Tracking”** as per the syllabus and have submitted a satisfactory report on this project as a part of fulfillment towards the degree of **“BACHELOR OF ENGINEERING” (Computer Science & Engineering)** from **RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL.**

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLES OF CONTENTS

ABSTRACT

Our aim is to build a tracking system which can identify the speed of a vehicle on the road. This information can be used in traffic maintenance, detecting of vehicles with more than specified limit, traffic control, traffic monitoring, traffic flow, security etc. This project will need the concepts of Machine Learning, Video Processing, Image Detection, and some more concept of Computer Vision. It will use data sets for vehicle detection, OpenCV, videos of road, Python-its IDE etc. Finally we will be able to find the speed of vehicle which can be used for traffic maintenance and other things. This will reduce the cost for traffic police and can be used for challan generation and will reduce the accidents on the road.

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Purpose

The purpose behind a project feasibility study is to know the different variables involved with your business venture and how it will be accepted on the open market along with who will be the target audience. The purpose of this study is to determine the factors that will make the business opportunity that was presented in the business case a success.

1.2 Document Convention

Heading

- Font size 16
- Font Style Bold
- Font Times New Roman

Sub Heading

- Font size 14
- Font style Bold
- Font Times New Roman

Content

- Font size 14
- Font Times New Roman

1.3 Intended Audience and Reading Suggestions

The intended audience of this document includes system developer who wants to develop similar project for detection purpose like Traffic Surveillance. Since the document is very accurate to development user can read a precise and effective procedure.

1.4 Scope

The Scope of this project is very broad in terms of traffic surveillance.
The moving vehicle detection, tracking and its speed measurement system is of great significance for present day transport system. Considering limitations of the existing systems such as noise and illumination sensitivity, Traffic controlling, easily detection speed and vehicle details etc.

1.5 Problem Definition

Vehicle tracking is the process of locating a moving vehicle using a camera. Capture vehicle in video sequence from surveillance camera is demanding application to improve tracking performance. This technology is increasing the number of applications such as traffic control, traffic monitoring, traffic flow, security etc. The estimated cost using this technology will be very less.

Video and image processing has been used for traffic surveillance, analysis and monitoring of traffic conditions in many cities and urban areas. Various methods for speed estimation are proposed in recent years. All approaches attempt to increase accuracy and decrease cost of hardware implementation. **The aim is to build an automatic system that can accurately localize and track the speed of any vehicles that appear in aerial video frames.**

1.6 Proposed Solution

- **Video Data Acquisition :**
  In order to examine the performance of the proposed vehicle detection and speed estimation for efficient traffic surveillance the recorded video and some standard vehicle traffic data have been used.

- **Image Pre-Processing :**
  To develop efficient vehicle detection and speed estimation scheme, the appropriateness of input data and its quality is of great significance. Here the input RGB video frame has been converted into the frames that has been followed by extraction of various parameters such as number of frames, frame rate, color format, frame size etc.

- **Background Modeling :**
  It is the matter of the fact that the core of the Background Subtraction approach is to retrieve the background of the moving video. In traffic surveillance system, while recording video on highway it becomes highly intricate to get the image without any moving vehicle. In order to retrieve such image we have implemented Background Subtraction Model.

- **Vehicle Detection :**
  We have detected the vehicle moving in the input video by using some functions provided by OpenCV which gives more accuracy than bounding box generation method.

- **Vehicle Tracking :**
  Our proposed vehicle tracking systems have been made on the basis of the feature tracking concept. The features extracted have been tracked over sequential frames retrieved from input traffic video data. Our proposed system represents an object matching scheme that estimates the distance between vehicle features or the features in the previous frame, which has been stored in track graph metrics and instantaneous frame.

- **Speed Estimation Scheme :**
  The detected moving vehicles possessing its matching ID have been tracked over frames of the video data. In order to calculate the total number of frames having same object has been estimated using following equation-

  Speed = Distance/ (Total Frames*Frame Rate)

- **Predicted Results :**

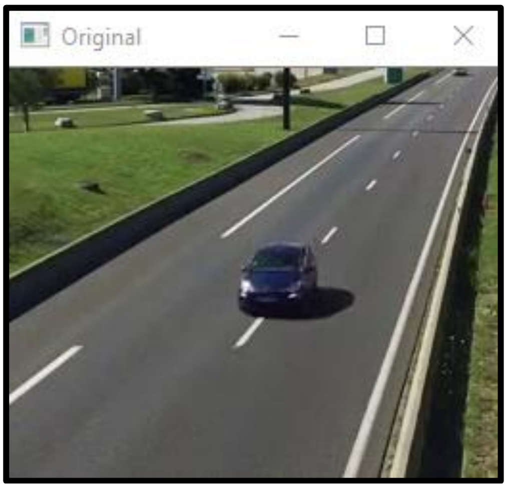


Fig. 1.6.1 Input RGB frame

The above depicted image is an input RGB frame taken from video frames obtained from input video.

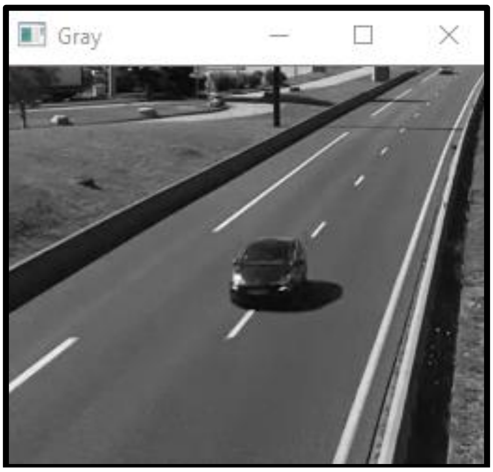


Fig. 1.6.2 RGB to Gray Scale

The input RGB frame is converted to Gray Scale image for performing edge detections and morphological operations.

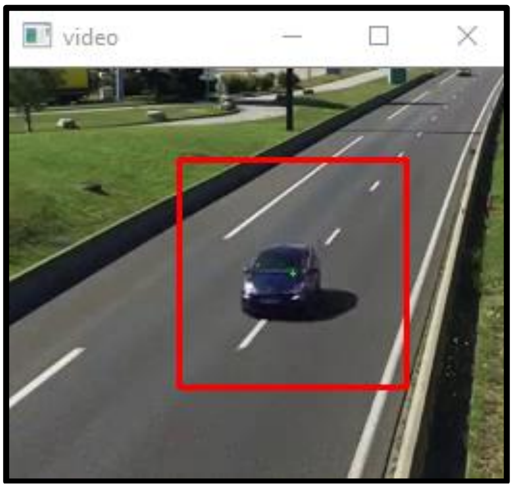


Fig. 1.6.3 Detected Vehicle

The red mark denotes that vehicle is detected.

2. SYSTEM REQUIREMENT ANALYSIS

2.1 Overall Description

In systems engineering and software engineering, requirements analysis focuses on the tasks that determine the needs or conditions to meet the new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.
Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

2.1.1 Product Perspective

One of the significant applications of video-based supervision systems is the Traffic Surveillance. So, for many years of researches have investigated in the vision based intelligent transportation system. The economic and easily availability of hardware have motivated researchers to develop more efficient solution for computer vision based applications. Image Processing based computer vision has become a promising technology for real time supervision, monitoring and control that serves major areas, ranging from civil applications, industries. Especially considering significance of Intelligent Transportation System, the vision based supervision has the significant contribution, as it can facilitate real time monitoring, vehicle tracking and identification. In addition, other parametric identification such as vehicle speeds, Vehicle density, Vehicle Count etc., for efficient traffic monitoring and Control.
However the traditional vehicle systems may be declined and not recognized well due to the vehicles are occluded by other vehicles or by background obstacles such as road signals, trees, weather conditions etc., and the performance of these systems depend on a good traffic image analysis approaches to detect and track the Vehicles.

In this, the traffic image analysis comprises of three parts:-

(1) Motion Vehicle Detection and Segmentation Approaches.

(2) Vehicle Tracking Approaches.

2.1.2 Product Function

- **Vehicle Detection:**

  We detected the vehicle moving in the input video by using region props function which is an inbuilt function in mat lab and gives more accuracy than bounding box generation method.

- **Vehicle Tracking:**

  In this paper the proposed vehicle tracking system has been made on the basis of the feature tracking concept. The features extracted have been tracked over sequential frames retrieved from input traffic video data. Our proposed system represents an object matching scheme that estimates the distance between vehicle features or the features in the previous frame, which has been stored in track graph metrics and instantaneous frame.

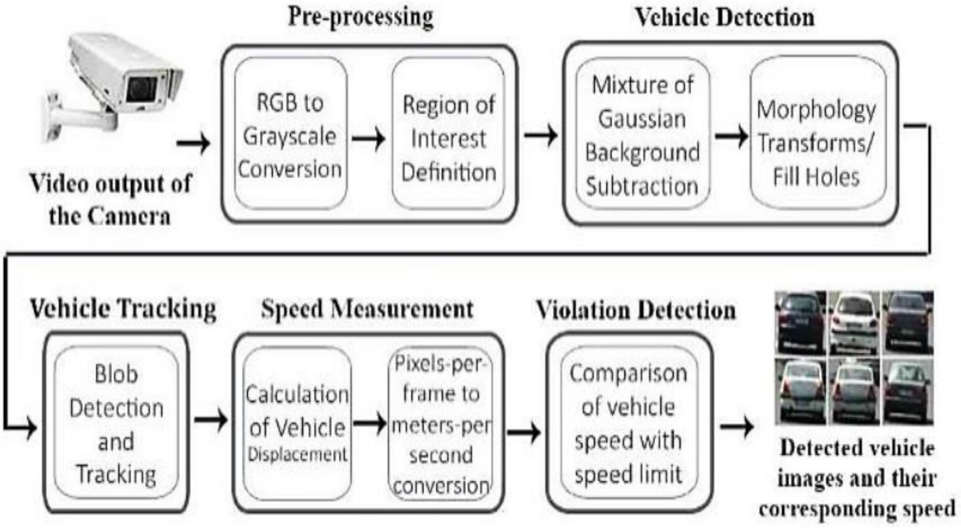

Pre-processing
Vehicle Detection
Video output of the Camera
RGB to Grayscale Conversion
Region of Interest Definition
Mixture of Gaussian Background Subtraction
Morphology Transforms/ Fill Holes
Vehicle Tracking
Speed Measurement
Violation Detection
Blob Detection and Tracking
Calculation of Vehicle Displacement
Pixels-per-frame to meters-per second conversion
Comparison of vehicle speed with speed limit
Detected vehicle images and their corresponding speed

Fig. 2.1.2.1 Traffic Surveillance System

2.1.3 User Classes And Characteristics

This is subsystem of a large model which can be used by any project or mainly for the surveillance purpose. This information can be used in traffic maintenance, detecting of vehicles with more than specified limit, traffic control, traffic monitoring, traffic flow, security etc.

2.1.4 Operating Environment

S.NO.	NAME	HARDWARE
1.	Processor	Intel core i5 (64 bit)
2.	RAM	4 GB
3.	Processor Speed	2.30 GHz

Table 2.1.4.1 Operating Environment

2.1.5 Design And Implementation Constraints

- Language of choice: PYTHON.
- Libraries: OpenCV, Time.
- Platform for deployment: PYCHARM.
- Dataset:https://medusa.fit.vutbr.cz/traffic/research-topics/detection-of-vehicles-and-datasets

2.1.6 Assumptions And Dependencies

We are limited to the dataset which we have taken in terms of detections. The threshold values will change according to the videos. The speed of the vehicle should not exceed beyond certain limits. There should be proper lighting for detection.

2.2 External interface Requirements

2.2.1 User Interfaces

Our System will not contain any user interface, since it is a subsystem which can be used by any other system like traffic surveillance. But user can use it through command line interface or through any python IDE.

2.2.2 Hardware Interfaces

S.NO.	NAME	HARDWARE
1.	Processor	Intel core i5 (64 bit)
2.	RAM	4 GB
3.	Processor Speed	2.30 GHz

Table 2.2.2.1 Hardware Interfaces

2.2.3 Software Interfaces

S. No	NAME	SOFTWARE
1	Platform	Windows 10
2	IDE	PyCharm
3	Libraries	OpenCV Time

Table 2.2.3.1 Software Interfaces

2.2.4 Communications Interfaces

- Windows

2.3 Functional Requirement

2.3.1 System Feature

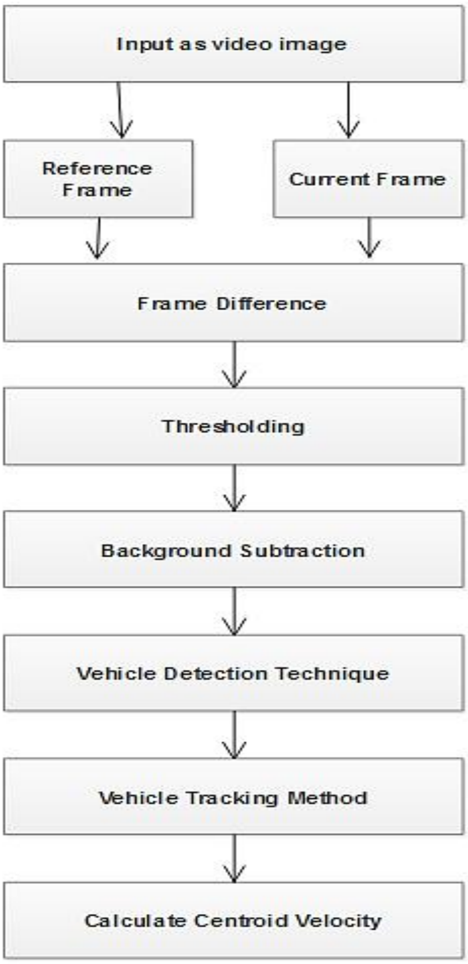


Fig. 2.3.1.1 Architecture

1. Image processing

To get better results, vehicle detection process should be performed in the grayscale image domain. Hence a RGB to grayscale conversion is performed on each video frame. To achieve an appropriate threshold level and make results more suitable than the input image, each frame should be brought in contrast to background.

2. Background Subtraction

Using provided threshold, the static parts of sequential video frames should be cleaned. The main challenge here is that the performance of image analysis algorithms suffers from darkness, glare, long shadows or bad illumination at night, which may cause strong noises. Therefore, the grayscale image might be unspecified in these situations and make the detection task a bit more complex. Edges essentially separate two various regions which are static region (the roadway) and dynamic region (moving vehicles). The static background is then deleted to locate moving objects in each frame. The result zone leaves only vehicles and some details as moving objects in sequential images which are changing frame to frame.

3. Vehicle Detection

As an observation (detection) zone, a region should be defined to display moving vehicle’s edges in a bounding box at the time that the vehicle enters it. This zone is in the middle of the screen and covers 1/3 of its height and 3/5 of its width (considering minimum and maximum available size of detectable passing vehicles in pixels).

4. Vehicle Tacking

The output of detection step i.e. vehicle’s blob inside the ROI- should be tracked in sequential frames to detect motion. Tracking process can be performed using various techniques. In the proposed method, if the vehicle has just entered the ROI, means it has not been tracked yet, it should be labeled by a Vehicle ID.

2.4 Non Functional Requiremet

2.4.1 Performance Requirements

It will change whole system of traffic control and surveillance drastically because of video processing and less influence of human beings.

2.4.2 Safety Requirements

The accident on the road will reduce by using this system.

2.4.3 Security Requirement

There should no dust on the camera’s for detection and the atmospheric condition can change the resultant results.

2.5 Project Plan

2.5.1 Team Members

Name: Harsh Verma

Email: harshverm776@gmail.com

Name: Brijesh Rathore

Email: rathorebrijesh55@gmail.com

Name: Himanshu Singh Patel

Email: himanshu.patel.sdbc@gmail.com

Name: Animesh Jain

Email: animesh3jain@gmail.com

2.5.2 Division of Work

“A team is on success gate if every individual is loyal to his/her responsibility.”

Harsh Verma: Requirement Gathering, Documentation, Programming.

Brijesh Rathore: Requirement Gathering, UML Diagrams, PPT, Programming.

Himanshu Singh Patel**:** PPT, Programming, UML Diagrams.

Animesh Jain**:** Documentation, PPT.

Time Schedule

1. **Requirement Analysis** : Approximately 15 days
2. **Design** : Approximately 10 days
3. **Coding** : Approximately 30 days
4. **Testing** : Approximately 10 days

3 ANALYSIS

3.1 Methodology Used

The programming language used for the development of the project is PYTHON and the software model used is the agile model.

Agile Model

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like −

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.

Advantages of Agile model:

- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Enables concurrent development and delivery within an overall planned context.
- Little or no planning required.

Disadvantage of Agile model:

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is a very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

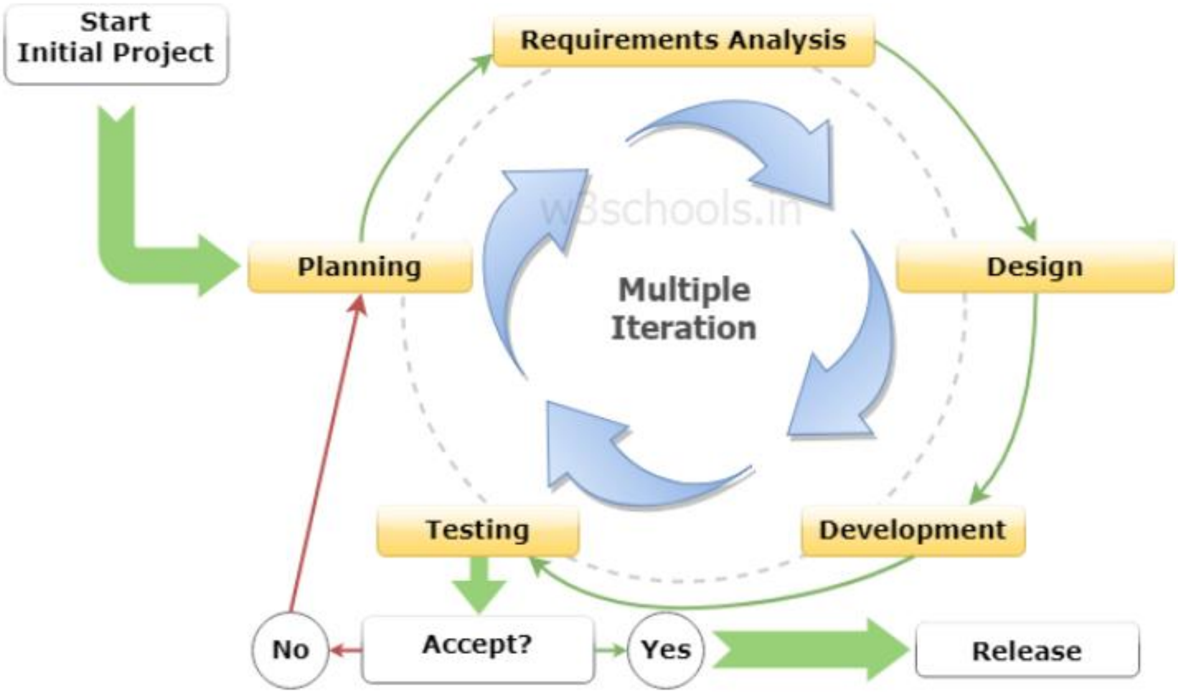


Fig. 3.1.1 Agile Model

3.2 Usecase Diagram

I. Usecase Diagram

A use case diagram at its simplest is a representation of user's interaction with the system that shows the relationship between the user and the different use case in which the users is involved. A use case diagram can identify the different types of user of a system and the different use cases.

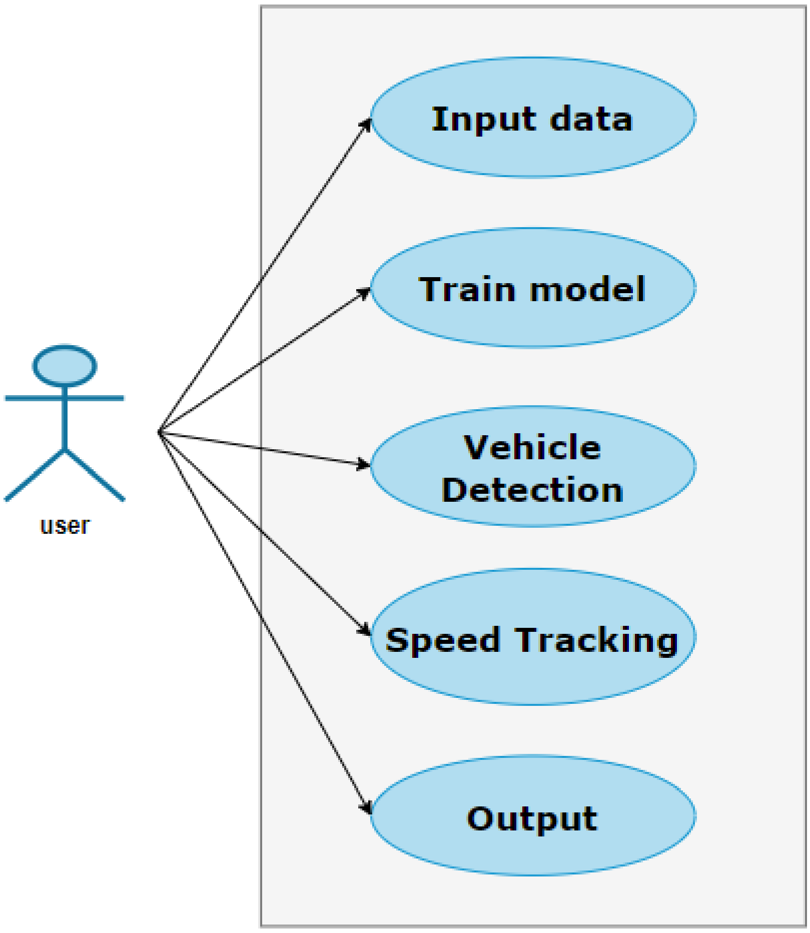


Fig. 3.2.1 Use Case Diagram

II. Use Case Specification

Name of the use case	Train Model
Actor	User
Precondition	Frame should available
Primary flow of events	It should preform vehicle detection.
Alternate flow of events	If frame is not available then the program while terminate.
Post condition	Vehicle detection will take place.
Use case termination	Cancel

Table 3.2.1 Use Case Specification for Train Model

Name of the use case	Vehicle Detection
Actor	User
Precondition	Train Model should work perfectly.
Primary flow of events	It should preform speed tracking.
Alternate flow of events	Vechile detection can’t be performed.
Post condition	Speed Tracking will take place.
Use case termination	Cancel

Table 3.2.2 Use Case Specification for Train Model

Name of the use case	Speed Tracking
Actor	User
Precondition	Vehicle detection should work perfectly.
Primary flow of events	It will show video.
Alternate flow of events	Speed tracking can’t be performed.
Post condition	Show Video.
Use case termination	Cancel

Table 3.2.3 Use Case Specification for Speed Tracking

3.3 Data Flow Diagram

Also known as DFD, Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

I. LEVEL 0 DFD

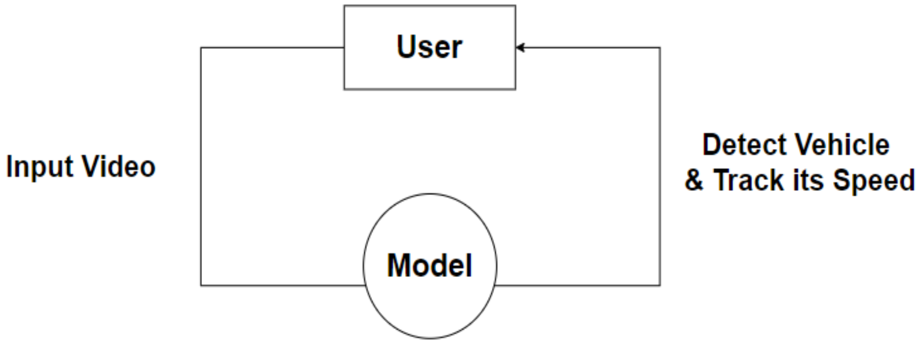


Fig. 3.3.1 Level 0 DFD

II. LEVEL 1 DFD

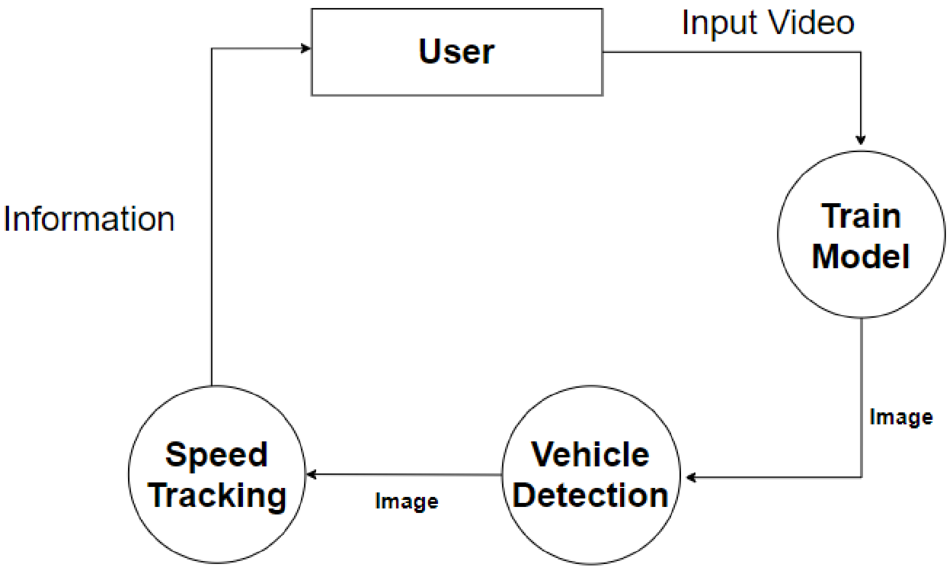


Fig. 3.3.2 Level 1 DFD

3.4 State Transition Diagram

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It’s a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don’t model every class using State diagrams. We prefer to model the states with three or more states.

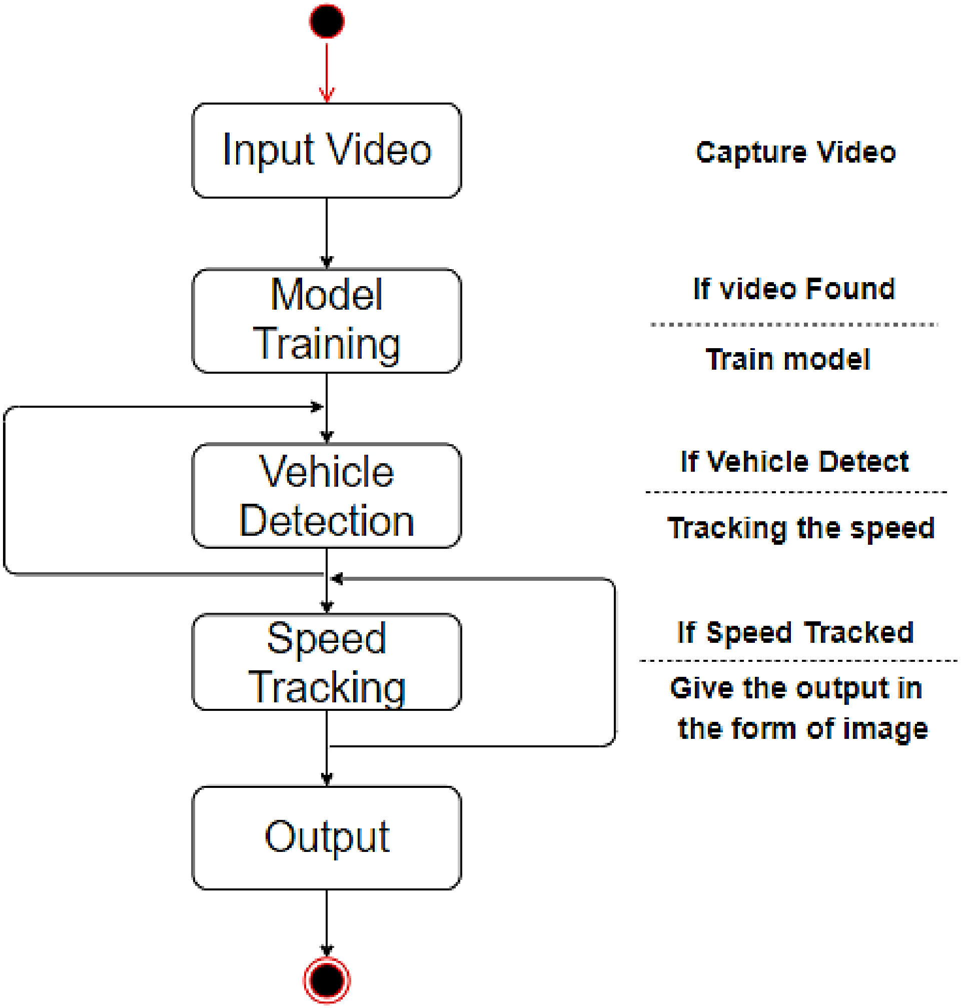


Fig. 3.4.1 State Transition diagram

4. DESIGN

4.1 Architectural Design

4.1.1 System Architectural Design

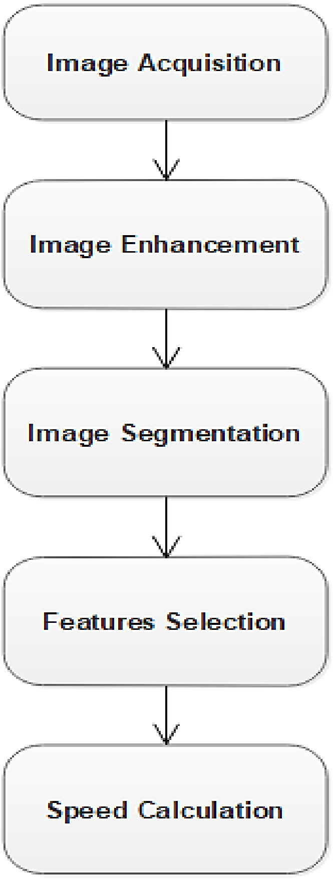


Fig: 4.1.1.1 Architecture of the Vehicle Detection & Speed Tracking system

4.1.2 Description of Architectural Diagram

- **Image Acquisition:**
  In image processing, it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing. It is the first step in the workflow sequence because, without an image, no processing is possible. The image that is acquired is completely unprocessed.

- **Image Enhancement :**
  Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further image analysis. For example, we can remove noise, sharpen, or brighten an image, making it easier to identify key features.

- **Image Segmentation :**
  Segmentation is the technique of dividing or partitioning an image into parts, called segments. It is mostly useful for applications like image compression or object recognition, because for these types of applications, it is inefficient to process the whole image.

- **Feature Selection :**
  Feature Selection is the process where we automatically or manually select those features which contribute most to your prediction variable or output in which we are interested in. Having irrelevant features in our data can decrease the accuracy of the models and make our model learn based on irrelevant features.

- **Speed Calculation :**
  The detected moving vehicles possessing its matching ID have been tracked over frames of the video data. In order to calculate the total number of frames having same object has been estimated using following equation-

  Speed = Distance/ (Total Frames*Frame Rate)

4.2 Database Design

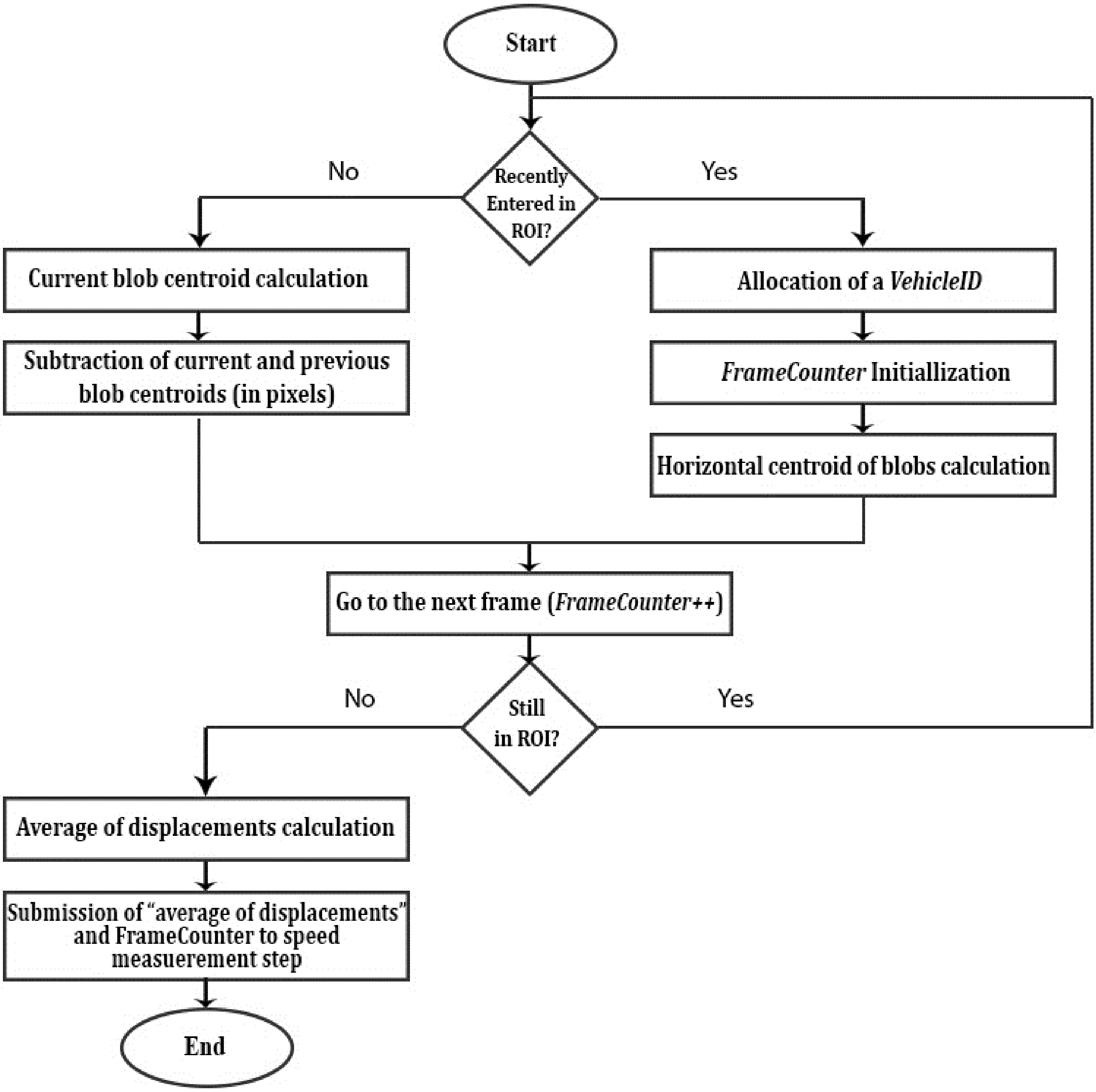
4.2.1 Normalization

Since we are using videos and there is no need for storing data we do not have database So there is no need of normalization.

4.3 Component Diagram

4.3.1 Flow chart

A flow chart is a type of a diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and there order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flow charts are used in analyzing, designing, documenting or managing a process program in a various field.



Start
Recently Entered in ROI?
No
Yes
Current blob centroid calculation
Subtraction of current and previous blob centroids (in pixels)
Allocation of a *VehicleID*
FrameCounter Initiallization
Horizontal centroid of blobs calculation
Go to the next frame (*FrameCounter++*)
Still in ROI?
No
Yes
Average of displacements calculation
Submission of "average of displacements" and FrameCounter to speed measuerement step
End

Fig: 4.3.1.1 A general flowchart of vehicle tracking algorithm.

4.4 Interface Design

User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface.

UI can be graphical, text based, audio-video based, depending upon the under lying hardware and software combination. UI can be hardware or software or a combination of both.

4.4.1 Screen Shots

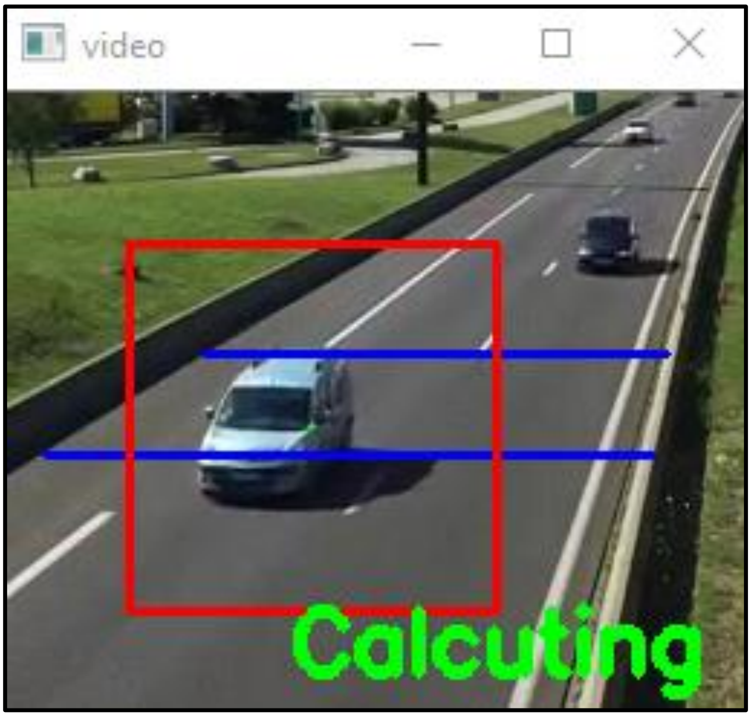
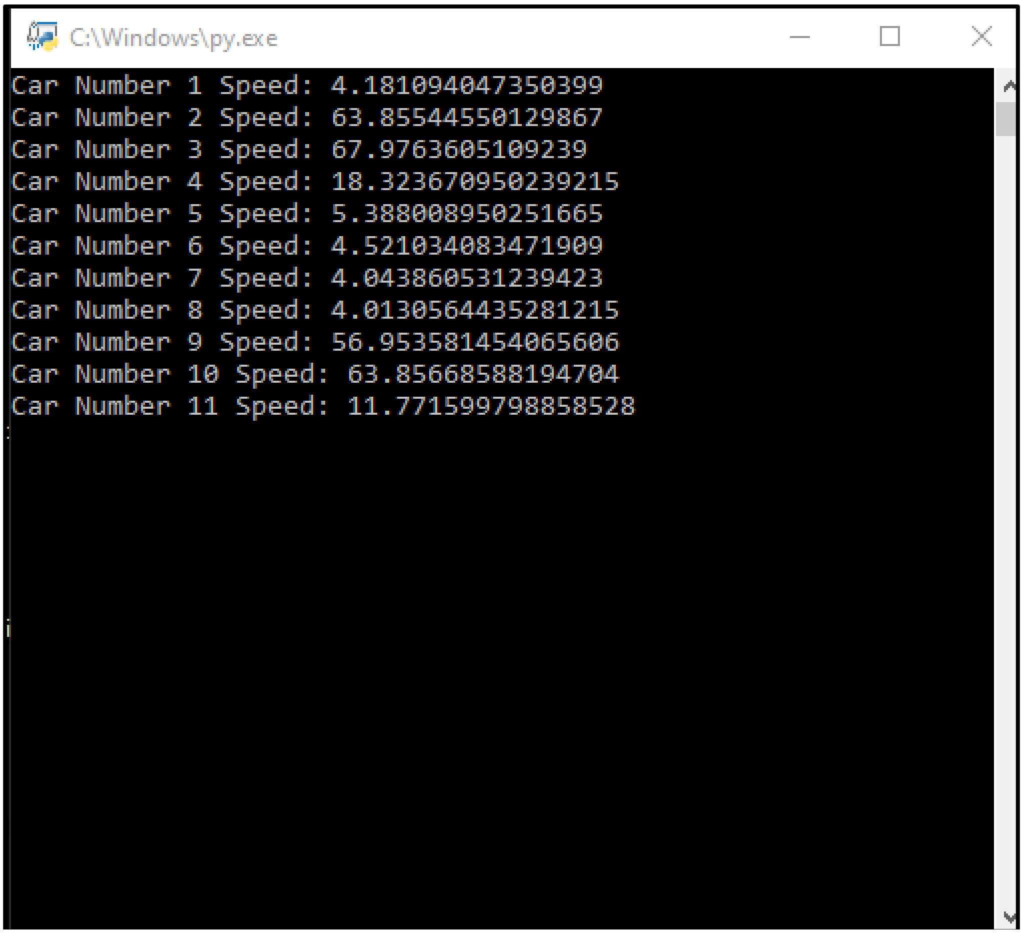


Fig: 4.4.1.1 Output Frame



```
Car Number 1 Speed: 4.181094047350399
Car Number 2 Speed: 63.85544550129867
Car Number 3 Speed: 67.9763605109239
Car Number 4 Speed: 18.323670950239215
Car Number 5 Speed: 5.388008950251665
Car Number 6 Speed: 4.521034083471909
Car Number 7 Speed: 4.043860531239423
Car Number 8 Speed: 4.0130564435281215
Car Number 9 Speed: 56.953581454065606
Car Number 10 Speed: 63.85668588194704
Car Number 11 Speed: 11.771599798858528
```

Fig: 4.4.1.2 Speed of Cars

5. IMPLEMENTATION

5.1 Language and Dataset Used For The Implementation

Python:

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability through use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Dataset:

We have use dataset published by SOCHOR Jakub et al. Comprehensive Data Set for Automatic Single Camera Visual Speed Measurement, IEEE T-ITS.

5.2 Feature Of language & dataset used for the project

Feature of Python:

- Easy to Learn and Use.
- Expressive Language.
- Interpreted Language.
- Cross-platform Language.
- Free and Open Source.
- Object-Oriented Language.
- Extensible.
- Large Standard Library.
- GUI Programming Support.
- Integrated.

Feature of Dataset:

Our dataset contain good quality videos in which each road have its three angles- center, right and left.

- Frame W-H : 1920 x 1080
- Data rate : 8001 kbps
- Bit rate : 8001 kbps
- Frame rate : 50 frames/second

5.3 Description Of Third Party tools used

PyCharm:

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda.

OpenCV:

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 TensorFlow, Torch/ PyTorch and Caffe.

Time:

The Python time module provides many ways of representing time in code, such as objects, numbers, and strings. It also provides functionality other than representing time, like waiting during code execution and measuring the efficiency of your code.

Any Video Converter Ultimate:

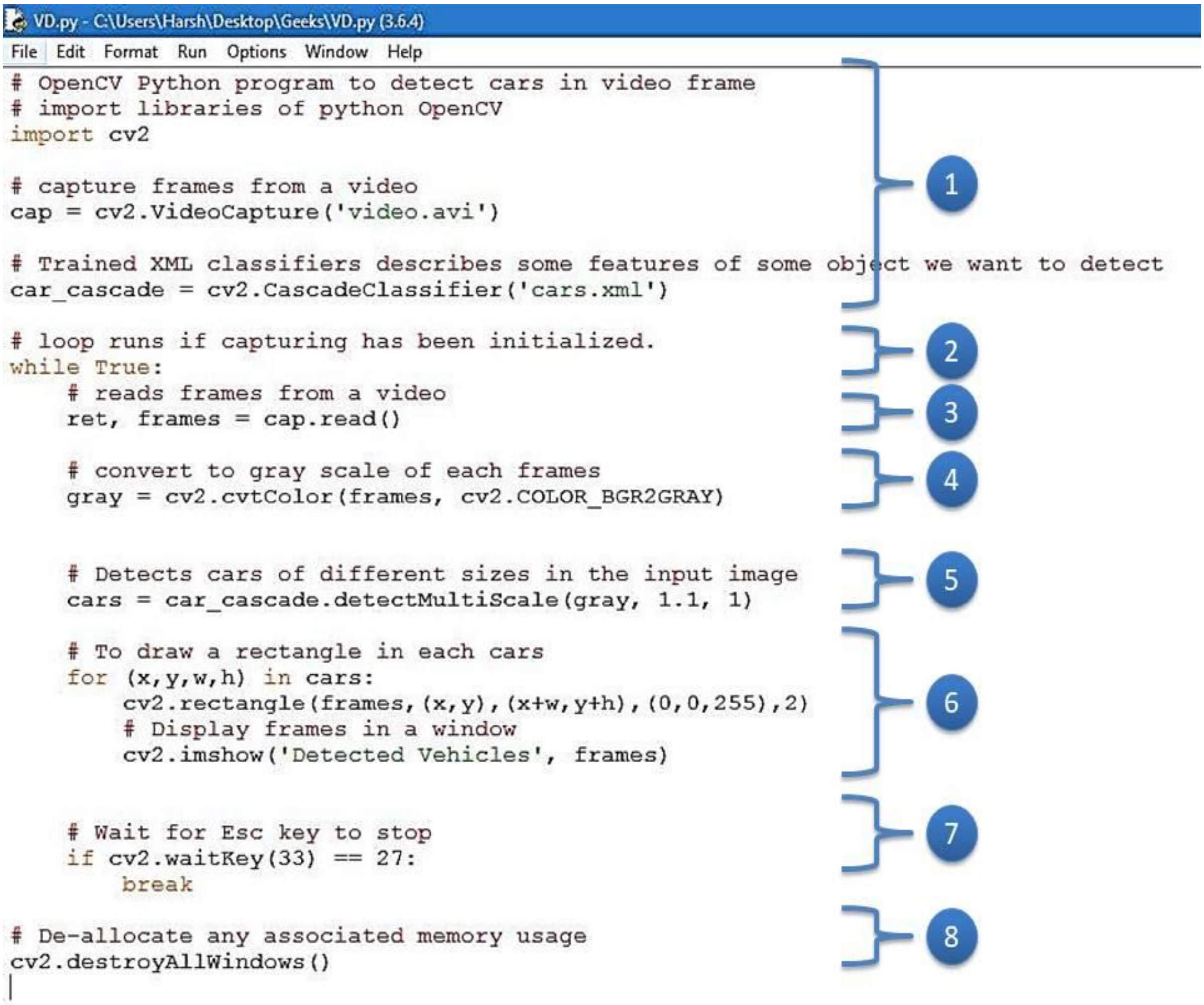
Any Video Converter Ultimate can help us to watch DVD movies and listen to music from CDs anywhere by converting DVD & CD and Blu-ray movies to popular formats like MP4, M4V, AVI, WMV, MOV, WMA, MP3, etc. Also, it can burn media files to blank DVD disc and play it on any DVD player. Any Video Converter Ultimate allows us to backup our DVD by burning content from one DVD to another unused DVD.

6 TESTING

6.1 White Box Testing

White box testing is defined as the testing of software solution’s internal structure, design, and coding. In this type of testing, the code is visible to the tester. It focuses primarily on verifying the flow of inputs and outputs through the application, improving design and usability, strengthening security. White box testing is also called Clear testing, Open Box Testing, Structural testing, Transparent Testing, Code-Based Testing and Glass Box Testing.

Detection Module:


VD.py - C:\Users\Harsh\Desktop\Geeks\VD.py (3.6.4)
File Edit Format Run Options Window Help
1
2
3
4
5
6
7
8

```
# OpenCV Python program to detect cars in video frame
# import libraries of python OpenCV
import cv2

# capture frames from a video
cap = cv2.VideoCapture('video.avi')

# Trained XML classifiers describes some features of some object we want to detect
car_cascade = cv2.CascadeClassifier('cars.xml')

# loop runs if capturing has been initialized.
while True:
    # reads frames from a video
    ret, frames = cap.read()

    # convert to gray scale of each frames
    gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)


    # Detects cars of different sizes in the input image
    cars = car_cascade.detectMultiScale(gray, 1.1, 1)

    # To draw a rectangle in each cars
    for (x,y,w,h) in cars:
        cv2.rectangle(frames,(x,y),(x+w,y+h),(0,0,255),2)
        # Display frames in a window
        cv2.imshow('Detected Vehicles', frames)


    # Wait for Esc key to stop
    if cv2.waitKey(33) == 27:
        break

# De-allocate any associated memory usage
cv2.destroyAllWindows()
```

Fig: 6.1.1 Detection Module Code for White box Testing

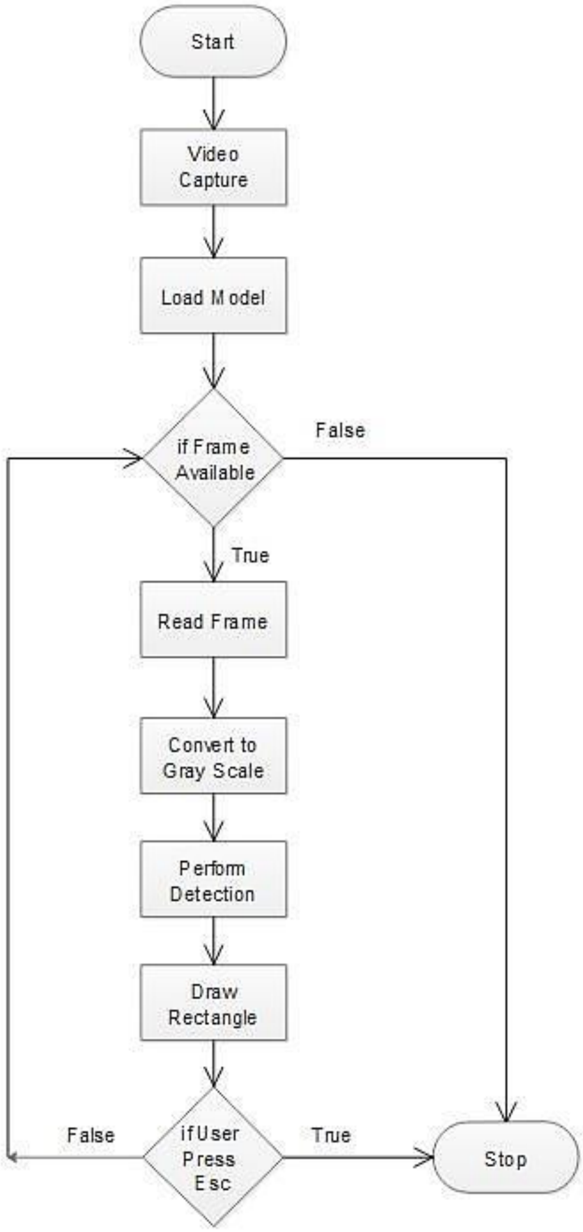


Fig: 6.2.2 Flow Chart

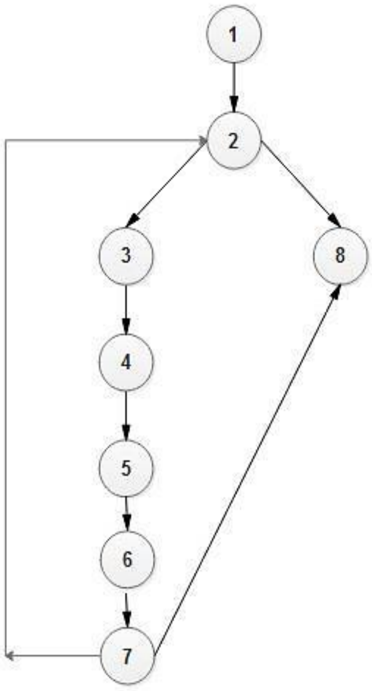


Fig: 6.2.3 Directed Graph

Cyclomatic Complexity:

1. V(G) = P + 1, where P is the number of predicate nodes in the flow graph.

   V(G) = 2 + 1 = 3

2. V(G) = E – N + 2, where E is the number of edges and N is the total number of nodes.

   V(G) = 9 - 8 + 2 = 3

3. V(G) = B + 1, where B is Number of non-overlapping regions in the graph.

   V(G) = 2 + 1 = 3.

Test Cases

- **Test Case Name:** When frame is available in data set.

  Expected Outcome: It should go in a loop for reading a frame & performing detection.

  Actual Outcome: It is going in a loop for reading a frame & performing detection.

  Path: 1 ➔ 2 ➔ 3 ➔ 4 ➔ 5 ➔ 6 ➔ 7 ➔ 2 …

  Result: Pass.

- **Test Case Name:** When frame is available in data set & Esc key is pressed after some time.

  Expected Outcome: It should go in a loop for reading a frame & performing detection & stop after sometime.

  Actual Outcome: It is going in a loop for reading a frame & performing detection & stop after sometime.

  Path: 1 ➔ 2 ➔ 3 ➔ 4 ➔ 5 ➔ 6 ➔ 7 ➔ 8

  Result: Pass.

- **Test Case Name:** When frame is not available in data set.
  Expected Outcome: It should break the loop and show the output.
  Actual Outcome: It break the loop and show the output.
  Path: 1 ➔ 2 ➔ 8
  Result: Pass.

6.2 Black box Testing

Black box testing is also known as Behavioral Testing, is software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, through usually functional.

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot See.

TEST CASE	TEST CASE DESCRIPTION	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
1.	Video containing vehicles	Video	Detect Successfully	Detect Successfully	Pass
2.	Video not containing vehicles	Video	Not Detected	Not Detected	Pass
3.	Video Quality	Video (More Blur)	Not Detected	Not Detected	Pass

Table 6.2.1 Test Cases for Black Box Testing

7 FUTURE SCOPE AND LIMITATION

Scope:-

The Scope of this project is very important to traffic surveillance and control.

Few points are:-

- This can be used for auto generation of challan using more camera and accuracy.
- The stolen vehicle can be detected by comparing with the registered entry of stolen vehicles.
- Can be used as traffic counters to count the number of vehicles plying on a highway.
- The Number plate of vehicle which has crossed over speed limit can be sent to mobile of traffic police in the area using Android application.

Limitation:-

- It will work perfect in proper lighting only.
- It is having less efficiency and accuracy.
- Accuracy can be increased using Neural Networks. Advanced image processing algorithms and libraries could be used so that the system can be used efficiently even during unfavorable lighting conditions and during the night time as well.

8 CONCLUSION

We are able to recognize speed of vehicles from the recorded video frames till a limited extent. The moving vehicle detection, tracking and its speed measurement system is of great significance for present day transport system. Vehicle tracking system makes better fleet management and which in turn brings large profits. Better scheduling or route planning can enable us to handle larger jobs loads within a particular time. Vehicle tracking both in case of personal as well as business purpose improves safety and security, communication medium, performance monitoring and increases productivity. So in the coming year, it is going to play a major role in our day-to-day living.

9 REFERENCE

Reports:

- **“Motion-based Vehicle Speed Measurement for Intelligent Transportation Systems”** by Ali Tourani, Department of Computer Engineering, University of Guilan, Rasht, Iran.
- **“Vehicle Tracking and Speed Estimation from Traffic Videos”** by Shuai Hua, Manika Kapoor, David C. Anastasiu, Department of Computer Engineering, San Jose State University, San Jose, CA.
- **“Vehicle Detection, Tracking and Speed Measurement for Traffic Regulation”** by Pulli Harsha Samhitha, Allu Naga Jyothi, Ramana Vesapogu, Manasa Mannem, S. Sri Harsha, V .R .Siddhartha Engineering College, JNTUK, Vijayawada, Associate Professor, Dept of Information Technology, V. R .Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India

Website:

- UML Diagrams- https://app.diagrams.net/