VEHICLE CLASSIFIER

By

Saira Rasheed

Wajeeha Noor Haroon

Submitted to the Faculty of Computer Science, Military College of Signals
National University of Science and Technology, Rawalpindi in partial fulfillment for the requirements of a B.E Degree in Computer Software Engineering, May 2019

In the name of Allah, the Most Beneficent, the Most Merciful

ABSTRACT

Vehicle Classifier

The rapid increase in number of vehicles has caused many problems; traffic accidents, congestion and security issues, making it almost impossible for humans to accurately classify vehicles and store its data manually. It has become a necessity to develop smart solutions for this problem, due to advancements in technologies. Vehicle Classifier will help minimize the number of persons required to monitor incoming vehicles and will reduce time and money. Also, it will be helpful for security purposes.

The idea of the project is to recognize any vehicle entering specific areas (Cantonment) through security cameras to keep record in the images extracted from the video cameras. Vehicle Classifier will detect, count and classify different types of vehicles in real-time traffic. On detection the result will be shown on the screen and the classification result will be stored in database.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis –Vehicle Classifier carried out by Saira Rasheed and Wajeeha Noor Haroon under supervision of Col. Adil Masood Siddique for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

_____

Approved By

Col. Adil Masood Siddique

Department of CSE, MCS

Dated:

_____

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To Our Parents and Teachers for their continuous support

AKNOWLEDGEMENTS

There is no success without the will of ALLAH Almighty. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are also grateful to our parents and family and well-wishers for their admirable support and their critical reviews.

We are grateful to our parents for their support and prayers. Their faith kept us going.

We are extremely grateful to our project supervisor Col. Adil Masood Siddique and Lec. Ayesha Naseer, for their support and guidance without which we could not have completed the project.

In the end, we would like to acknowledge our teachers and instructors here at Computer Science Department, MCS, all of them have guided us to make it possible for us to complete both this degree and the project.

Table of Contents

List of Figures

List of Tables

Chapter 1. INTRODUCTION

This chapter gives an overview and Introduction of the Vehicle Classifier system. It gives a background and reason to develop the system as well as the solution we have managed to develop to the respective problem.

1.1 Overview

The idea of the project is to recognize any vehicle entering specific areas (Cantonment) through security cameras to keep record in the images extracted from the video cameras. Vehicle Classifier will detect, count and classify different types of vehicles on the road in real-time. On detection the result will be shown on the screen and the classification result will be stored in database.

1.2 Problem Statement

The increase of vehicles has caused many problems; traffic accidents, congestion and security issues. It is impossible for humans to accurately classify vehicles and store its data manually. This project will minimize the number of persons required to monitor incoming vehicles and will reduce time and money. This project will be helpful for security purposes. Therefore, it has become a necessity to develop smart solutions for this problem, due to advancements in technologies.

1.3 Approach

The project involves implementation of desktop application using python. Object classification and counting is done using TensorFlow. The application is integrated with MS SQL Server which provides for reliably storing the data in the database. The camera feed used is form already deployed security cameras.

1.4 Scope

Vehicle Classifier will help to detect and classify vehicles. Firstly, we will acquire vehicle images from videos, secondly we will perform image processing techniques on

the image to classify the vehicle and thirdly we will display the results and store them in database.

1.5 Objectives

The main objective of the system is to provide a system which performs following functions:

1. Detecting moving vehicles on road.
2. Count vehicles incoming on the road.
3. Classify vehicles as car, truck, bus, motorcycle.
4. Provide time and cost effective solution.
5. Reduce workforce for manual surveillance on roads.

During the course of this project, all the aspects of software engineering are covered i.e. survey and feasibility analysis, requirement gathering, architectural and detailed design, implementation and testing along with documentation (SRS, SDS, Test Document, Final Report and User manual). Students are also expected to develop extensive knowledge and technical skills in the fields of image processing, python programming and databases.

1.6 Deliverables

SR.	**Tasks**	**Deliverables**
1.	Literature Review	Literature Survey
2.	Requirements Specification	Software Requirements Specification Document(SRS)
3.	Detailed Design	Software Design Specification Document(SDS)
4.	Implementation	Project demonstration
5.	Testing	Evaluation plan and Testing plan
6.	Training	Deployment plan
7.	Deployment	Complete application with necessary documentation

Table 1.1: Deliverables

Chapter 2. LITERATURE REVIEW

Numerous researchers have proposed approaches for detecting vehicles and classifying different types of vehicles. However, it is a very compelling process because of increasing number of vehicle models and sizes, even within a single class. The classification task gets even more challenging, due to noisy background subtraction, change in the size of regions and uncontrollable environment conditions (e.g. fog, rain, lighting, and haze).

In [4] the authors proposed an appearance-based method for classifying moving objects such as cars, vans, trucks, people, and bikes in real time. The classified objects are represented by utilizing multi-block local binary patterns.

In [5] blob features followed by linear discriminate analysis, hidden markov model, fuzzy C-means clustering, and weighted k-nearest neighbor classifiers were used to classify vehicles and path modelling. The author presents two modules one for classifying and other module for behavioral analysis.

In [6] the author combined geometry-based and appearance-based approaches to classify vehicles in multi-classes such as small, medium, and large size vehicles and intra-classes such as pickup, sport utility vehicle (SUV), and van. In intra class classification medium sized vehicles are classified further.

In [7], a vehicle-detection and tracking system was developed to detect vehicles entering. This system measured optical flow and tracked vehicles by classifying their headlights, bonnet, front window, and roof area.

In [8], the authors proposed a vehicle model recognition system based on SIFT of an image of the vehicle's headlights and the homogeneity, which were calculated based on the distribution of features.

In [9], the author proposed a method to extract license plates from ROIs and corner templates based on edge detection.

In [10] the authors built a three-layer neural network and trained it with texture features.

These texture features were computed from frontal images of vehicles. Thus, the neural network was able to recognize the make and model of the moving vehicles.

In [11] Nam and Nam (2018), established the unique vehicle detection and classification ways based on images from thermal cameras and visible light. Physical features obtained from pictures were used to classify moving vehicles. However, several other characteristics like contrast, entropy, homogeneity and energy were also studied from images. They experimentally proved their methods and established that their vehicle classifier like visible and thermal images were 92.1% and 65.8% accurate respectively while vehicle classified of six types like RV, Sedan, SUV.

In [12] the authors present and algorithms for vision-based detection and classification of vehicles in monocular image sequences of traffic scenes recorded by a stationary camera.

In present study, we propose a smart surveillance system to detect and classify vehicles using tensorflow.

Chapter 3. SOFTWARE REQ. SPECIFICATION(SRS)

3.1 Introduction

Software Requirements Specification (SRS), is a document describing the expected behavior of a software system. The aim of this document is to present detailed description and requirements of project Vehicle Classifier which uses image processing techniques to classify vehicles.

3.2 Purpose

This document covers the software requirements and specifications for project Vehicle Classifier. The idea of the project is to detect and classify vehicles in the images extracted from the video cameras. This document describes the system development requirements and features of the Vehicle Classifier, which can serve as a guide to developers and as a software validation document for the prospective client.

3.3 Document Conventions

This section describes standards followed while writing this document.

3.3.1 Headings

Headings are prioritized using numbers. Highest priority has a single digit and subsequent headings having numbers per their level.

Main headings have a single digit which is followed by a dot and name of the section having font size 18pt and font style Times New Roman.

For second level subheadings each subheading has its respective main heading number followed by a dot and subheadings own number followed by the name of the sub heading. It has font size 16pt and font style Times New Roman.

Further subheadings follow the same convention as second level subheadings but with font size 14pt.

3.3.2 Basic Text

Basic text has font size 12pt and font style Times New Roman

3.3.3 References

References are provided where necessary. They are written using IEEE-Citation Style and numbered.

3.4 Intended Audience and Reading Suggestions

This section describes the different audiences for which the document is intended.

The intended readers for Vehicle Classifier are project supervisor, UG project evaluation team, BESE 21 FYP group (developers, testers and documentation writers) and potential users of this system.

3.4.1 Project Supervisor

This document will help the supervisor to supervise the project and provide better guidance to the group. It will help the supervisor, to check whether all the requirements have been understood and implemented properly.

3.4.2 UG Project Evaluation Team

This document will help the evaluation team to evaluate the progress of the project, its implementation and final project. This document will provide scope and requirements of the project to the evaluators.

3.4.3 BESE 21 FYP Group (developers, testers and documentation writers)

This document will help the group in better development of the project and it is important because it will help them to understand the project. As testers, it will help to understand system features to set up test cases for system testing.

3.5 Overall Description

3.5.1 Product Perspective

Main idea of the project Vehicle Classifier is to detect and classify vehicles with high accuracy. It is impossible for humans to accurately classify vehicles and store its data manually. This project will minimize the number of persons required to monitor incoming vehicles and will reduce time and money. This project will be helpful for security purposes.

3.5.2 Product Functions

The main functions of the product are:

1. **Video Acquisition:** System will obtain videos from camera.
2. **Video Processing:** Vehicle images will be acquired by the system.
3. **Classify Vehicle:** The system will classify the vehicles.
4. **Display Vehicle Classification Results:** The system will display the results on screen
5. **Count:** The system will count the detected vehicles.
6. **Store in database:** The system will store results in a database.

3.5.3 User Classes and Characteristics

This section describes the type of users for the Vehicle Classifier:

1. **Security Agency (Regular user)**

   The user is basically the security personnel who is using the Vehicle Classifier to detect and classify vehicles on road/ entrance. When the vehicle is classified the system will display results.

2. **Project Supervisor (Occasional user)**

   To evaluate the project i.e. to find accuracy and error in it the project supervisor will use the system.

3. **Tester (Occasional user)**

   To check that the project is developed in accordance to the Software Requirement Specification Document.

3.5.4 Operating Environment

3.5.4.1 Hardware Requirements

The Vehicle Classifier operates with following external hardware:

1. **Camera:** The camera (already installed) is used to get video of vehicles on road.

3.5.4.2 Software Requirements

1. OpenCV 3.0.0
2. Python 3.6
3. MS SQL Server 2014
4. Anaconda 3

5. Visual Studio Code
6. TensorFlow

3.5.5 Design and Implementation Constraints

1. Vehicle Classifier will only process the video when camera is at a certain height.
2. Input may contain noise along with data.
3. Vehicle Classifiers performance will be low on bad weather conditions.
4. Client is responsible for maintenance after delivery of software.

3.5.6 User Documentation

User manual in text for using the software.

3.5.7 Assumptions and Dependencies

1. For Vehicle Classifier normal weather conditions have been assumed.
2. Performance of the system will depend upon the hardware infrastructure of the system on which Vehicle Classifier is running.

3.6 System Features

System feature are organized by use cases and functional hierarchy so that the main functional hierarchy so that main functions are understandable. The major modules of the product are:

- **Video Acquisition:** System will obtain videos from camera.
- **Video Processing:** Vehicle images will be acquired by the system.
- **Classify Vehicle:** The system will classify the vehicles.
- **Display Vehicle Classification Results:** The system will display the results on screen
- **Count:** The system will count the vehicles.
- **Store in database:** The system will store results in a database.

3.6.1 Video Acquisition

This feature allows the system to acquire video.

3.6.1.1 Description

This is where the application initiates. This feature enables the system to acquire the videos as input for further operations.

This feature has high priority with low risk.

3.6.1.2 Stimulus/Response Sequences

3.6.1.2.1 Basic Data Flow

This function begins when the video is sent to the system for processing.

3.6.1.2.2 Alternate Data Flow

The system notifies when the video is not acquired correctly.

3.6.1.3 Functional Requirements

3.6.1.3.1 The system shall be able to acquire real time video.

3.6.2 Video Processing

System divides the video into frames for preprocessing.

3.6.2.1 Description and Priority

This feature identifies best pictures from obtained video as input for further operations.

System shall be able to divide video into frames and select the best one.

This feature has high priority with medium risk.

3.6.2.2 Stimulus/Response Sequences

3.6.2.2.1 Basic Data Flow

System divides the video into frames.

Frames are used for identification of moving vehicles

3.6.2.3 Functional Requirements

3.6.2.3.1 The system shall be able to divide video into frames effectively.

3.6.2.3.2 The system shall be able to select the best image from divided frames.

3.6.3 Classify Vehicle

The system will classify the detected vehicles into car, truck, motorcycle, bus.

3.6.3.1 Description and Priority

The images obtained previously will be used to classify the vehicles. This feature has high priority and high risk.

3.6.3.2 Stimulus/Response Sequences

3.6.3.2.1 Basic Data Flow

The image obtained contains a vehicle in it.

Vehicle is classified into its type.

System will continue to process more images.

3.6.3.2.2 Alternate Flow

The image obtained is not suitable for classification.

System will extract image form video again.

3.6.3.3 Functional Requirements

3.6.3.3.1 The system shall be able to classify the vehicles based on the features extracted from the image.

3.6.4 Store in Database

This will store the classification results in the database.

3.6.4.1 Description and Priority

The classification results obtained from previous stage will be stored in the database. This feature has medium priority and low risk.

3.6.4.2 Stimulus/Response Sequences

3.6.4.2.1 Basic Data Flow

Classification results are obtained correctly.
The data is store in the database tables.

3.6.4.3 Functional Requirements

3.6.4.3.1 The system shall be able to store the classification results in the database.

3.6.5 Display Vehicle Classification Results

This will visualize the classification and counting results on the screen for the user.

3.6.5.1 Description and Priority

This feature displays the classification results: i.e. type of vehicle and the number of vehicles on the screen. This feature has high priority and medium risk.

3.6.5.2 Stimulus/Response Sequences

3.6.5.2.1 Basic Data Flow

The results are displayed on screen.

3.6.5.2.2 Alternate Flow

System notifies when it is not able to display results.

3.6.5.3 Functional Requirements

3.6.5.3.1 System displays correct results on the screen.

3.7 Count

This will count the number of vehicles detected and classified.

3.7.1 Description and Priority

This feature counts the vehicles detected and classified in the video frame.

This feature has high priority and medium risk.

3.7.2 Stimulus/Response Sequences

3.7.2.1 Basic Data Flow

The number of vehicles passing ROI line are counted.

3.7.2.2 Alternate Data Flow

The vehicle is not counted.

The vehicle is counted twice due to camera position.

3.7.3 Functional Requirements

3.7.3.1 The system shall be able to count the number of vehicles.

3.7 External Interface Requirements

3.7.2 User Interfaces

Responsive GUI is provided for the user . User will be able to view result video and database. When the user access the system they view an interface as shown below:

1. Main Window()

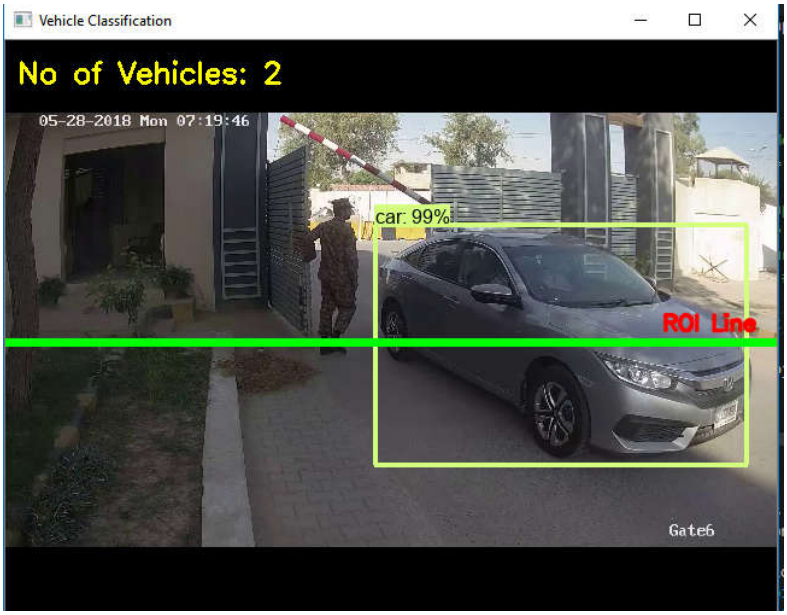

Figure 3.1 Mian Window

2. Database View Window

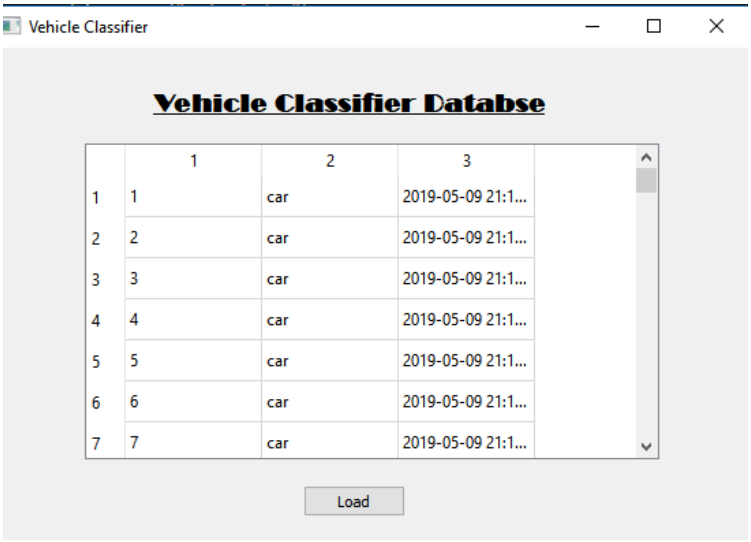

Figure 3.2 Database Window

3.7.3 Hardware Interfaces

- Video input will be taken from camera in real time.

3.7.4 Software Interfaces

- Windows for running the system on PC.
- Anaconda for Python and its libraries.
- MS SQL Server for database.

3.7.5 Communication Interfaces

Cable will be used as medium of communication between camera and software.

3.8 Non-Functional Requirements

3.8.1 Performance Requirements

The system should be fast in terms of performance. To judge the systems performance, we check its response time and efficiency.

3.8.2 Safety Requirements

The major types of failure of software systems which may damage the system are programming errors, software support errors and hardware failures. We had done following safety requirements to avoid above mentioned problems.

1. Careful while coding
2. Testing after coding

3.8.3 Security Requirements

There is no connection with any device other than the camera so the data will not be misused.

3.8.4 Software Quality Attributes

1. **Accuracy**

   The system shall provide 85% accuracy to make the project more useful for security purposes and in real time smart surveillance system.

2. **Ease of use**

   The system shall have a graphical user interface for easy use.

3. **Reliabillity**

   The system should provide reliability to the user. The system will run steadily with every one of the features mentioned above available and executing perfectly. The system will be tested completely and all exceptions should be taken care of.

3.9 Other Requirements

3.9.1 Database

We are using MS SQL Serve as our database.

Database implementation is as follows:

```
Create table Vehicle
(
S_NO          NUMERIC(6) PRIMARY KEY IDENTITY(1,1) NOT NULL,
VEHICLE_TYPE  VARCHAR(15) NOT NULL,
C_TIME        DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL
);
```

Vehicle

S_NO	VEHICLE_TYPE	C_TIME

Figure 3.3Database Schema

Chapter 4. DESIGN AND DEVELOPMENT

4.1 Introduction

4.1.2 Purpose

This software design document describes the architecture and system design of Vehicle Classifier. This document serves as a guide for the developers and as a software validation document for the client. Document includes classes and their inter-relationships, use cases with elaborate descriptions, sequence diagrams and various flow charts.

4.1.3 Scope of Project

Vehicle Classifier will help to detect, count, extract number plate and classify vehicles. Firstly, we will acquire vehicle images from videos, secondly we will perform image processing techniques on the image to classify the vehicle and thirdly we will display the results. This project is based on safe city idea.

4.1.4 Definitions, Acronyms, Abbreviations

OpenCV: Graphics library for image processing

ROI: Regions of Interest

4.2 Overview of Document

This document is about the detailed architectural design of Vehicle Classifier. The document is divided into various sections. Section 1 introduces the document and provides overview for executive purposes. Section 2 includes detailed description of the system with various diagrams and charts. This section includes all the architectural details of system under development. Section 3 describes all the modules and components of the system in detail. Section 4 compares this product to various other similar products available in the market. Section 5 throws light on the design decisions and tradeoffs. Section 6, pseudo code of all the components is provided.

4.3 System Architecture Description

This section provides detailed system architecture of Vehicle Classifier. Overview of system modules, their structure and relationships and user interfaces are discussed.

4.3.1 Overview of Modules/Components

Following is the brief overview of all the modules for Vehicle Classifier. Detailed description of these modules is presented in section 3.

1. **Image Acquisition Module**

   This module initiates the start of Vehicle Classifier i.e. it takes a frame from the video and passes this frame to image preprocessing module.

2. **Image Preprocessing Module**

This module takes the frame from image acquisition module and then this frame is processed to remove noise and background and enhance its quality.

3. **Identify Vehicles Module**

This module after taking image form the image preprocessing module identifies the vehicle (*Regions Of Interest*) in the image.

4. **Classify Vehicle Module**

This module extracts features from the vehicle identified before. On the basis of these features vehicles are classified into their types i.e car (Suzuki, Toyota etc), truck etc. These classification results are sent to the database module.

5. **Database Module**

This module will save the classified vehicle results and number plate in the database to keep record.

6. **Counting Module**

This module simply counts the number of vehicles passing in the video.

4.4 Structure and Relationships

This section covers the overall technical description of Vehicle Classifier. It shows relationships between different components.

4.4.1 System Block Diagram

System block diagram shows the high-level description of Vehicle Classifier. It shows the modules of the system, their associations and flow of data between modules.

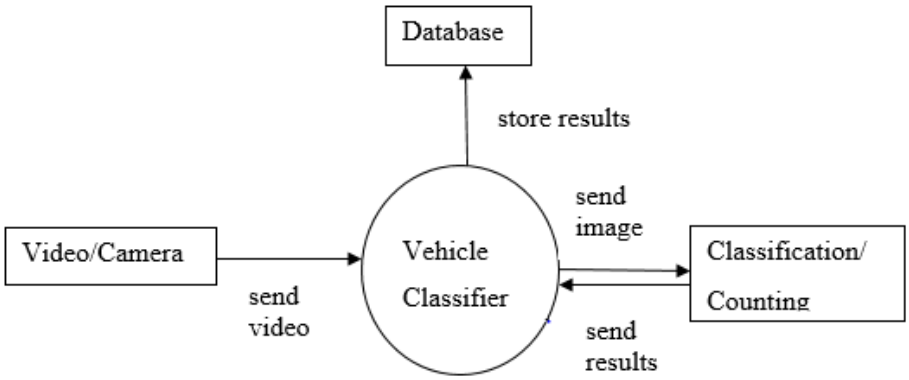

Figure 4.1 System Block Diagram

Image acquisition module would acquire the video from camera already installed. It will then divide the video into frames and extract required frame and then apply preprocessing steps on this frame. From this frame image of vehicle will be identified and its features will be extracted. Finally, form these features vehicle will be classified into its type, vehicle's number plate will also be identified and no of passing vehicles will be counted.

4.4.2 Use Case Diagram

Use case diagram shows different ways in which a user interacts with the system.

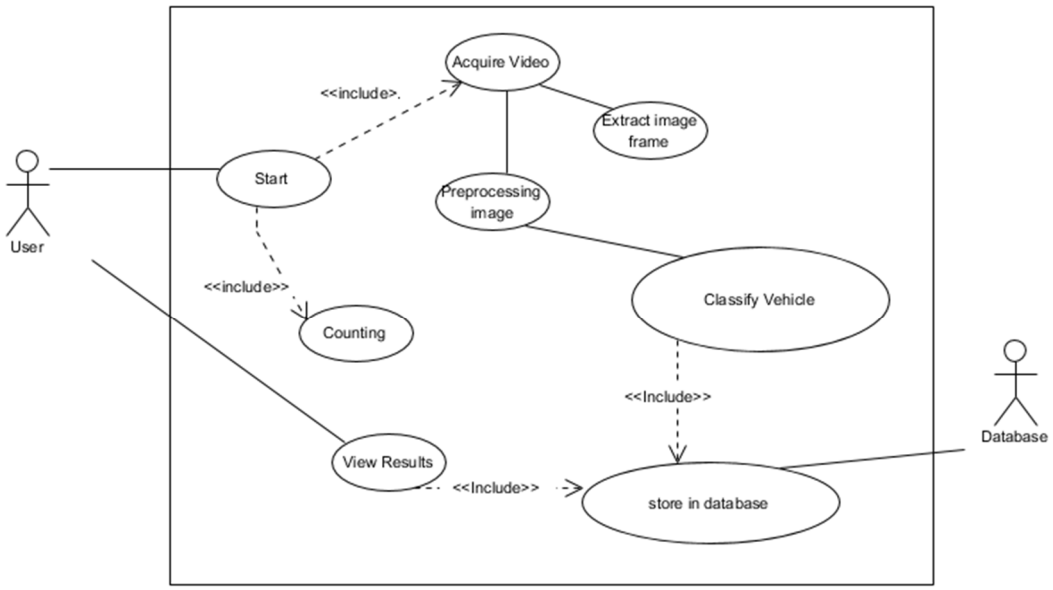

Figure 4.2 Use Case Diagram

Description of use cases in the Fig 4.2 is given below:

Use case name	Start
Primary actor	User
Secondary actor	N/A
Normal course	- Acquire video successfully - Count the no. of vehicles passing. - Divide video into frames and select best frame. - Vehicle is identified in the frame. - Features of identified vehicle are extracted. - Vehicle is classified successfully. - Number plate of vehicle is identified and saved.
Alternate course	- Error occurred due to misconfiguration or wrong calibration of camera consequently producing a beep.
Pre-condition	Cameras are already installed.
Post-condition	Vehicle is classified and its number plate is extracted.
Include	Acquire Video, Extract image frame, Preprocess image, Classify Vehicle, Number Plate, Counting
Assumptions	Already installed camera is properly aligned.

Table 4.1 Use Case 1-Start

Use case name	View Results
Primary actor	User
Secondary actor	LCD
Normal course	- Displays results
Alternate course	- N/A
Pre-condition	Results are in database.
Post-condition	Display results on screen.
Include	N/A
Assumptions	Classification results are in database.

Table 4.2 Use Case 2- View Results

Use case name	Store in Database
Primary actor	Database
Secondary actor	N/A
Normal course	- Store results from classification and in database.
Alternate course	- Data is not in correct format.
Pre-condition	Classification results are available.
Post-condition	Data is stored successfully and is available to be viewed.
Include	N/A

Assumptions	N/A

Table 4. 3 Use Case 3- Store in Database

Use case name	Count
Primary actor	User
Secondary actor	N/A
Normal course	- Count the number of vehicles passing ROI line.
Alternate course	- Vehicle is not counted
Pre-condition	Vehicle is detected.
Post-condition	Number of counted vehicles is shown on screen.
Include	N/A
Assumptions	N/A

Table 4.4 Use Case 4- Count

Use case name	Acquire video
Primary actor	User
Secondary actor	N/A
Normal course	- Video is acquired form system.
Alternate course	- Video not available.
Pre-condition	Vehicle is successfully acquired.
Post-condition	After processing video is shown on screen.
Include	N/A
Assumptions	N/A

Table 4.5Use Case 5- Acquire Video

Use case name	Extract image frame
Primary actor	User
Secondary actor	N/A
Normal course	- Best frame is acquired form video.
Alternate course	- N/A
Pre-condition	Video acquisition was successful.
Post-condition	Best frame is extracted and sent for preprocessing.
Include	N/A
Assumptions	N/A

Table 4.6 Use Case 6- Extract Image Frame

Use case name	Preprocess image
Primary actor	User
Secondary actor	N/A
Normal course	- The frame extracted is preprocessed.
Alternate course	- N/A
Pre-condition	Frame extraction was successful.
Post-condition	Processed frame is sent for classification.
Include	N/A
Assumptions	N/A

Table 4.7 Use Case 7-Preprocess Image

4.4.3 Sequence Diagram

Following sequence diagram shows the sequence in which events occur for the use case in Fig 4.3.

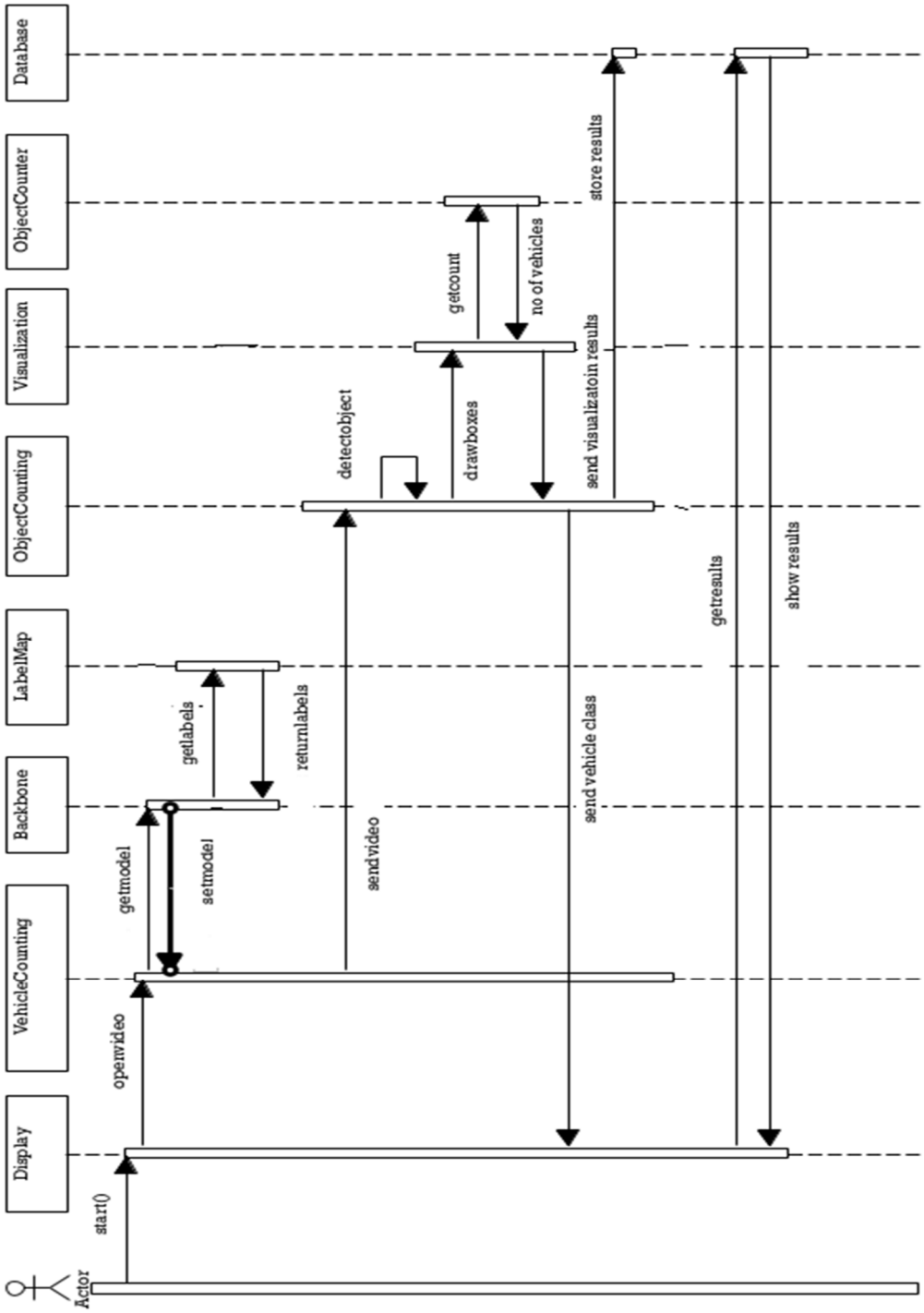

Figure 4.3 Sequence Diagram

4.4.4 Class Diagram

Following class diagram shows the classes of Vehicle Classifier and their relationships.

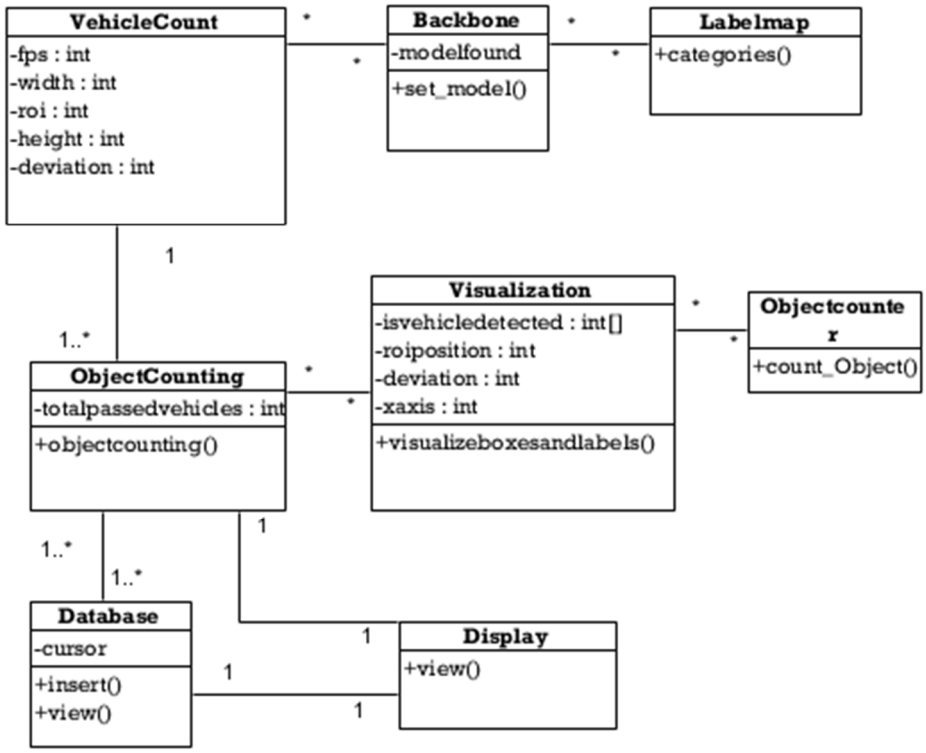

Fig 4.4: Class Diagram of Vehicle Classifier

Class	Description
VehicleCounting	This is main class of the System. It starts the system.
Backbone	This class loads the model.
LabelMap	This class gets the labels for the detected vehicles.
Object_counting	This class detects the vehicles in the video feed and sends them to visualization_utils class for drawing boxes.
Visualization_utils	This class draws bounding boxes around detected vehicles.
Object_counter	This class counts the number of vehicles detected. If the vehicle passes the ROI then the counter is incremented.
Database	This class provides the connection to the database. It stores and retrieves data form the database.
Display	This class initiate the GUI of the system.

Table 4.8 Class Description Table

4.4.5 Activity Diagram

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently with their respective start and end states.

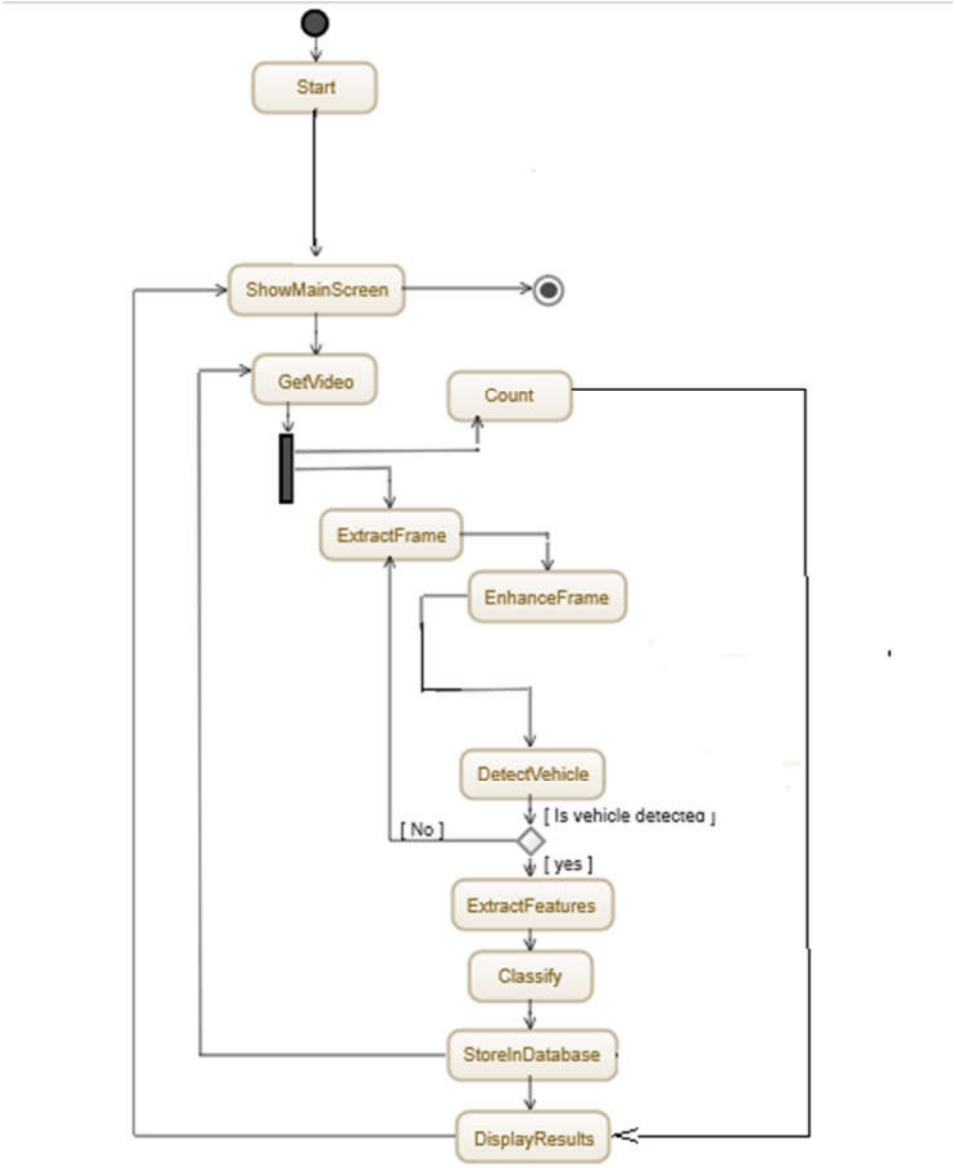

Figure 4.5 Activity Diagram

4.5 Detailed Description of Components

This section describes the modules of Vehicle Classifier. These modules are further sub classified into components.

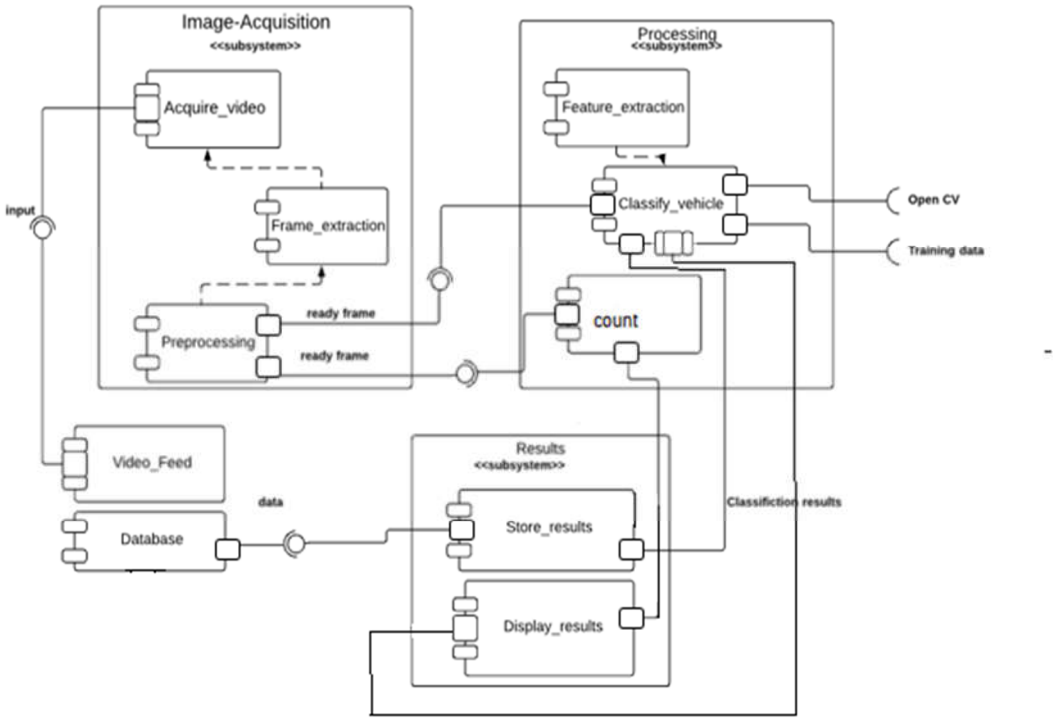

Figure 4.6 Component Diagram

4.5.1 Image-Acquisition Module

This module performs all the preprocessing steps for Vehicle Classifier which includes receiving footage from camera, frame extraction and frame enhancement techniques. It provides base for successful working of other modules.

Identification	Name: Acquire_Video Location: Image-Acquisition Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Video Acquisition** **Requirement** The system shall be able to acquire real time video. **Description** This feature enables the system to acquire video from cameras already installed for surveillance purpose. This video will be fed into the system for further processing.
Function	This component interfaces with camera to obtain footage for further processing.
Subordinates	It has two subordinate: Frame Extraction Preprocessing

Dependencies	This component is independent module and runs in parallel to entire application.
Interfaces	This component has following interfaces: **Camera Interface:** For getting video input as live feed from camera.
Resources	**Hardware:** Camera already installed for surveillance purposes **Software:** Camera interface
Processing	This component will receive real time camera feed which will be used for further processing.
Data	This component uses following information of the application: - Time of video recording as sequence identifier for further videos.

Table 4.8 Acquire Video

Identification	Name: Frame_Extraction Location: Image-Acquisition Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Video Processing Requirement** The system shall be able to divide video into frames effectively. **Description** This feature identifies best images from the acquired video as input for further processing.
Function	This component divides the video into frames i.e. sequence of images.
Subordinates	It has one subordinate: Preprocessing
Dependencies	This component is dependent on Acquire_Video module.
Interfaces	None
Resources	**Hardware:** Camera already installed for surveillance purposes
Processing	This module transforms the video into a set of frames as a video is actually a sequence of frames coming at a certain number of frame per second.
Data	This component uses following information of the application: - Time of video recording as sequence identifier for further videos.

Table 4.9 Frame Extraction

Identification	Name: Preprocessing Location: Image-Acquisition Module

Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Video Processing** **Requirement** System shall be able to select best image from divided frames. **Description** This feature identifies best images from the acquired video as input for further processing.
Function	This component of system performs preprocessing steps like image enhancement after choosing the best suitable frame for further processing.
Subordinates	It has two subordinates: Classify_Vehicle Count
Dependencies	This component is dependent on Frame_Extraction module.
Interfaces	None
Resources	**Hardware:** Camera already installed for surveillance purposes
Processing	Video is sequence of frames coming at a certain number of frame per second. We need to enhance the frames for better results.
Data	This component uses following information of the application: - Time of video recording as sequence identifier for further videos.

Table 4.10 Preprocessing

4.5.2 Processing Module

This module performs all the processing related to identification and feature extraction of slots from an image. Feature extraction is the main input for classification module.

Identification	Name: ROI_Identifier Location: Processing Module
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **Video Processing** **Requirement** System shall be able to identify all ROI (front of car areas) from the selected frame. **Description**

	The images obtained from previous stage will be used to identify ROI (front of the car areas).
Function	This component of system will identify the objects in the image which, in our case, are no plate, headlights, grill areas of the car.
Subordinates	It has one subordinate: Feature_Extraction.
Dependencies	This component is dependent on preprocessing component because feature rich image would lead us to better results.
Interfaces	None
Resources	Hardware: Camera Software: OpenCV
Processing	This component will identify objects in an image by using intensity based/feature based edge detection techniques.
Data	This component uses following information of the application: Time of video recording as sequence identifier for further videos, information of features in an image

Table 4.11 ROI Identifier

Identification	Name: Feature_Extraction Location: Processing Module
Type	Component
Purpose	This component helps in fulfilling the following requirement from Software Requirements Specification Document: **Video Processing** **Requirement** System shall be able to extract features correctly from the selected ROI. **Description** Features for classification will be identified.
Function	This component of system will find and calculate the features of Regions of Interest.
Subordinates	It has two subordinates: Classify_Vehicle Count
Dependencies	This component is dependent on ROI_Identifier component.
Interfaces	None
Resources	Hardware: Camera Software: OpenCV
Processing	Features are the characteristics which are used to classify vehicles. This component will get these for every selected frame.
Data	This component uses following information of the application: Time of video recording as sequence identifier for further videos, information of features in an image

Table 4.12 Feature Extraction

Identification	Name: Classify_Vehicle Location: Processing Module
Type	Component
Purpose	This component helps in fulfilling the following requirement from Software Requirements Specification Document: **Classify Vehicle** **Requirement** The system shall be able to classify the vehicles based on the features extracted from the image. **Description** Identified features from previous stage will be classified as different types of cars.
Function	The system shall be able to classify the vehicles based on the features extracted from the image.
Subordinates	It has one subordinate: Store_Results
Dependencies	This component is dependent on Feature_Extraction component.
Interfaces	None
Resources	Hardware: Camera Software: OpenCV
Processing	Features are the characteristics which are used to classify vehicles. This component will use training data to classify vehicles based on these features.
Data	This component uses following information of the application: Time of video recording as sequence identifier for further videos, information of features in an image.

Table 4.13Classify Vehicle

Identification	Name: Count Location: Processing Module
Type	Component
Purpose	This component helps in fulfilling the following new requirement added in Software Requirements Specification Document: **Count Vehicle** **Requirement** System shall be able to count the number of vehicles. **Description** Vehicle will be counted when it crosses ROI line.
Function	This component of system will count the number of the vehicles.
Subordinates	N/A
Dependencies	This component is dependent on Feature_Extraction component.
Interfaces	None

Resources	Hardware: Camera Software: OpenCV
Processing	Features are the characteristics which are used to classify vehicles. This component will count number of the vehicles.
Data	This component uses following information of the application: Time of video recording as sequence identifier for further videos, information of features in an image.

Table 4.14 Count

4.5.3 Results Module

Identification	Name: Store_Results Location: Results Module
Type	Component
Purpose	This component helps in fulfilling the following requirement from Software Requirements Specification Document: **Store in Database Requirement** The system shall be able to store the classification results in the database. **Description** Classification results will be stored.
Function	The system shall be able to store the classification and no plate results in the database.
Subordinates	None
Dependencies	This component is dependent on Classify_Vehicle component.
Interfaces	None
Resources	Software: MS SQL
Processing	This component will take classification results and store them in the database.
Data	This component uses following information of the application: Time of video recording as sequence identifier for further videos and classification results

Table 4.15 Store Results

Identification	Name: Display_Results Location: Results Module
Type	Component
Purpose	This component helps in fulfilling the following requirement from Software Requirements Specification Document: **Display Vehicle Classification Results Requirement**

	System displays correct results on the screen. **Description** Classification results will be displayed on screen for the user.
Function	System displays correct results on the screen.
Subordinates	None
Dependencies	This component is dependent on database module.
Interfaces	None
Resources	Software: MS SQL
Processing	This component will display classification results from the database.
Data	This component uses following information of the application: Data saved in the database.

Table 4.16 Display Results

4.6 Reuse and Relationship in Other Products

Vehicle Classifier implements the safe city project idea. It first registers the image for regions of interest and then extracting features for classifying vehicles by their type. Various systems for vehicle classification have been proposed using sensor based systems but these systems are difficult to install and maintain.

Vehicle Classifier is simple and install to go system that can reduce the effort required to detect and classify vehicles.

4.8 Design Decisions and Tradeoffs

Vehicle Classifier is component based system that is driven by demand. Every component has been assigned with the responsibility to do a task. Mainly there are three modules; image acquisition, processing and results module. Image acquisition module will acquire the footage and divide into frames. These frames are then needed to be enhanced to be processed. Next, processing module will register the regions of interest. Finally, features are extracted from ROIs and then classified. Results module will display the classification details.

Clearly, components can do their work independently, but, in a certain flow (data as well as control). That lead us to **high cohesion.**

Moreover, component don't have much interaction, once a component has completed its work system will generate an event for further action, consequently, the component registered for that event will come into action. This leads us to **low coupling**.

Chapter 5. PROJECT TEST AND EVALUATION

5.1 Introduction

The objective of test document is that it is required for the successful execution of testing process for a project. The test plan describes the strategies, process and methodologies which will be used to plan and execute the testing of Vehicle Classifier. By providing detailed test information, we hope to reduce the probability of overlooking items and improve test coverage. Testers will be able to use each test case provided in this document to move forward and begin testing.

Manual testing will be done by the tester taking the role of end-user to test the system and identify any unexpected behavior.

The test document includes all the plans, approach and methods to test Vehicle Classifier. The pass/fail for each test item is also defined.

5.2 Test Items

Based on the requirements of project Vehicle Classifier following are the major modules/ functionalities that should be taken into account during the testing process: -

- Video Processing
- Detect Vehicle
- Classify Vehicle
- Counting
- Database
- Display

5.3 Features to be Tested

Following are the features being tested:

1. The system shall be able to acquire video in real time for further processing.
2. The system shall be able to divide video into frames.
3. The system shall be able to classify vehicle by applying classify vehicle module.
4. The system shall be able to count vehicles in the video.
5. The system shall be able to store vehicle type into the database.

5.4 Detailed Test Strategy

Testing strategy comprises of unit testing using white box testing and black box testing. Integration testing is done in order to check successful integration of the modules of the system.

5.4.1 Unit Testing

In unit testing, individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. It is done at code level for specific programming errors.

5.4.2 White Box Testing

In white box testing, tester bypasses the UI. The tester chooses the inputs and outputs and they are tested directly at code level and results are compared according to requirements. In this type of testing the code and structure of the program are known to the tester. The test cases generated shall cause each condition to be executed at least once, so for this to happen we are Alternative Path Testing. As the functionality of program is simple, this method will be easy to apply

5.4.3 Black Box Testing

In black box testing test cases are derived from system requirements and specifications. It involves running through every possible input/output to verify its results.

5.4.4 Integration Testing

In integration testing we test all the previous modules after their integration. It is done to ensure that the modules are functionally normally when combined together.

5.4.5 Incremental Testing

There are four basic modules that are to be integrated. After integration these modules will form the complete application. Incremental testing will be done to complete the

integration. Integration testing will be performed by the developers. The following describes the modules as well as the steps to be taken to achieve the integration.

Video Acquisition

This module initiates the major functioning of the Vehicle Classifier system. It divides the video into frames after it has been acquired. Each frame passes further preprocessing techniques to detect the vehicle. After detection frame is passed to the

classify vehicle module.

Classify Vehicle Module

After video has been acquired by the system, system will select an enhanced frame and apply image processing on these frames to detect the vehicle. After detection this module classifies the vehicle into its type i.e. car, motorcycle, truck, bus. This module also counts the vehicles in the video. This module is integrated with display and count module.

Count

After the vehicle has been detected this module counts them in the video feed. We have drawn a ROI line on the video frame, it helps in counting. Vehicle counter is only incremented after the vehicle crosses the ROI line.

Display Module(GUI)

This module will display the classification results and vehicle count on the user interface screen. GUI is designed independently and then integrated with whole system.

Database Module

This module will make sure that number plate extraction data is successfully being stored in the database. This module will be integrated with the number plate extraction module.

5.4.6 System Testing

System testing ensures that all modules are working, separately and together combined.

Performance Testing

This test evaluates the fulfillment of a system with specified performance requirements. It is done using black-box testing. It will be performed by

- Checking the response time of the system
- Check database read/write time

5.5 Item Pass/Fail Criteria

Test case details are specified in section Test Deliverables. Following are the item pass/fail criteria:

1. Preconditions are met
2. Inputs are carried out as specified
3. The results are as specified in output => Pass
4. The results are not as specified in output => Fail
5. The system does not work => Fail

5.6 Suspension Criteria and Resumption Requirements

Whenever a defect is found/introduced, testing will be suspended. Testing will be resumed after removal of defect.

5.7 Test Deliverables

Following are the test cases:

Test Case Name	Frame Extraction
Test Case Number	1
Description	Frames are extracted from the video
Testing Technique Used	White Box Testing
Preconditions	Video from camera is acquired properly.
Input	Video.
Steps	Frame Extraction algorithm applied.
Expected Output	Multiple frames

Alternative Path	N/A
Actual Output	Confirmed

Table 5.1 Test Case 1- Frame Extraction

Test Case Name	Enhance Image
Test Case Number	2
Description	Image is enhanced for better processing.
Testing Technique Used	White Box Testing
Preconditions	Test case 1 is satisfied
Input	Frame to be enhanced
Steps	Enhance image by removing blur and increase contrast
Expected Output	Enhanced image
Alternative Path	N/A
Actual Output	Confirmed

Table 5.2 Test Case 2- Enhance Image

Test Case Name	Segmentation
Test Case Number	3
Description	Removes background from the enhanced image
Testing Technique Used	White Box Testing
Preconditions	Test cases 1 and 2 are satisfied
Input	Enhanced image
Steps	Background removal
Expected Output	Segmented image
Alternative Path	N/A
Actual Output	Confirmed

Table 5.3 Test Case 3- Segmentation

Test Case Name	Feature Extraction
Test Case Number	4
Description	Segmented image is compared with the data in model.
Testing Technique Used	White Box Testing
Preconditions	Test cases 1,2 and 3 are satisfied
Input	Segmented image
Steps	Features are compared with the model.
Expected Output	Features are extracted

Alternative Path	N/A
Actual Output	Confirmed

Table 5.4 Test Case 4- Feature Extraction

Test Case Name	Classify Vehicle
Test Case Number	5
Description	Segmented images are classified as particular vehicle types.
Testing Technique Used	White Box Testing
Preconditions	Test cases 1-4 are satisfied
Input	Features of the segmented image
Steps	Classification is done by comparing with the model used.
Expected Output	Segmented image is classified
Alternative Path	N/A
Actual Output	Confirmed

Table 5.5 Test Case 5 Classify Vehicle

Test Case Name	Count
Test Case Number	6
Description	Vehicles in the video feed are counted
Testing Technique Used	White Box Testing
Preconditions	Test case 1 is satisfied
Input	Video
Steps	Vehicles are counted as they are tracked for classification
Expected Output	Count on display screen
Alternative Path	N/A
Actual Output	Confirmed

Table 5.6 Test Case 6- Count

Test Case Name	Send data to database
Test Case Number	7
Description	Classification result is sent to database
Testing Technique Used	Black Box testing
Preconditions	Database schema and query is correct
Input	Data from classify vehicle
Steps	Data is sent to database
Expected Output	Data is correctly received in database module

Alternative Path	N/A
Actual Output	Not Confirmed

Table 5.7 Test Case 7- Send Data to Database

Test Case Name	Store in database
Test Case Number	8
Description	Storage of data in database
Testing Technique Used	Black Box testing
Preconditions	Database is correct and the query is correct
Input	Data from store in database
Steps	Data is added in to the database
Expected Output	Data is stored in the database
Alternative Path	N/A
Actual Output	Confirmed

Table 5.8 Test case 8- Store in Database

Test Case Name	GUI
Test Case Number	9
Description	This module is related to user interface of the system.
Testing Technique Used	Black Box testing
Preconditions	System is correctly working/installed
Input	Program initiation
Steps	Create and display the main screen
Expected Output	Data is displayed
Alternative Path	N/A
Actual Output	Confirmed

Table 5.9 Test Case 9 - GUI

Test Case Name	Show database
Test Case Number	10
Description	Retrieval of data in database
Testing Technique Used	Black Box testing
Preconditions	Database is correct and the query is correct
Input	Data from database
Steps	Data is retrieved form the database
Expected Output	Data is shown on the screen
Alternative Path	N/A
Actual Output	Confirmed

Table 5.10Test Case 10-Show Database

5.8 Environmental Needs

5.8.1 Hardware

- Camera (already installed surveillance cameras)

5.8.2 Software

- Python 3.6
- Anaconda
- Tensorflow
- OpenCV
- SQLServer 2014

5.9 Responsibilities, Staffing and Training Needs

Responsibilities

For the completion of unit and integration testing tasks, all developers of the project are responsible.

Staffing and Training Needs

Basic knowledge of testing techniques such as black box testing, white box testing, integration testing should be known by the developers. All developers will check each other's work and actively participate in development and testing process.

5.10 Risk and Contingencies

Efforts have been made to remove all failures but there are certain unpredictable factors such as incorrect input data, network issues between camera and the system or system failure that may lead to some issues in the system. To cover all these issues error handling will be done but there may still be unforeseeable circumstances that may happen.

Schedule Risk

In order to complete the project on time, as the project might get behind schedule, we will increase the no.-of- hours/day to work on the project.

Budget Risk

By using less costly alternatives the budget will be reduced, i.e. we are using already installed surveillance cameras.

Chapter 6. FUTURE WORK

This project can be used as a basis to understand and add features to make it into an even bigger and complex system. It could also be commercialized. In future accuracy could be improved further and scope could be extended to make it more comprehensive. Additional features could be added like more functionalities such as:

1. Number plate detection
2. A login method
3. An alarm system for suspicious vehicles for which data has already been fed into the database.
4. The project can be further explored on the idea of intelligent transport systems.

Chapter 7. CONCLUSION

7.1 Overview

In conclusion, this project provides solution for detecting and classification problem. Vehicle classifier has 75% accuracy in classifying vehicles and counting them. This helps to easily perform traffic surveillance through the use of cameras which have already been installed.

7.2 Objectives Achieved

1. Reduce manual surveillance personnel.
2. Easy and efficient monitoring.
3. Providing cost and time effective solution.

APPENDICES

APPENDIX A

Glossary

ROI:	Region of interest
Tensor flow:	An open source framework for object detection.
OpenCV:	An open source library for real time programming

APPENDIX B (USER MANUAL)

Table of Contents

1. General Information

General Information section explains the system and the purpose for which it is developed.

1.1 System Overview

Vehicle Classifier processes the videos form the surveillance cameras in real time to detect, count and classify vehicles into their types. After classification the results are shown on screen and also stored in database. The project is developed to reduce manual surveillance workload.

1.2 Organization of the Manual

The User Manual is divided in three sections.

- General Information
- System Summary
- Interfaces and Functionality

General Information section explains the system and the purpose for which it is developed.

System Summary section gives you information about the system configuration components and different modules of the system.

Interface and Functionality introduce you to the system interface and its functionality.

1.3 Acronyms and abbreviations

GUI: Graphical user interface

ROI: Region of interest

2. System Summary

System summary provides the general overview of the system. The summary outlines the uses of the system's software requirement, system's configuration, user access levels and system's behavior in case of any contingencies.

2.1 System Configuration

Vehicle Classifier operates on Windows based OS initially it can be modified to be used on Linux based systems.

2.1.1 System Components

Vehicle classifier is developed without any external hardware but it will work better with a GPU installed in the system.

2.2 User Access Levels

- Administration/Organization using the system has full access over the system.

2.3 Contingencies and Alternate mode of Operations

If the system crashes then the database will not be effected.

3. Getting Started

Getting started section explains how to configure the system and install it for first time.

3.1 Installation

1. Camera (already installed) must be aligned properly.
2. The window based system must have MS SQL server and python libraries (anaconda which comes with preinstalled libraries)

3.2 System Interface

On starting the system it will start taking the video feed form and show results on screen.

1. Main window

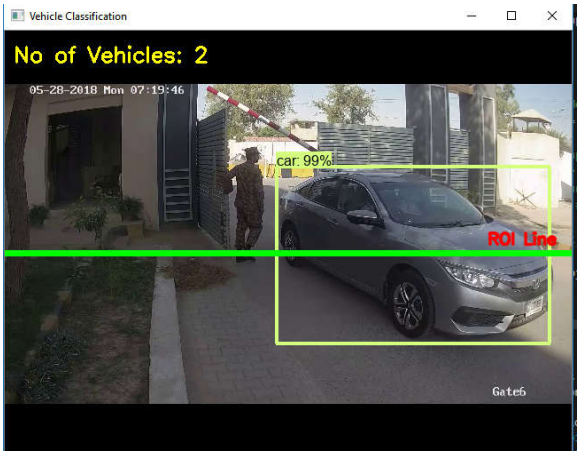

Figure 3.1 Main Window

2. Database view

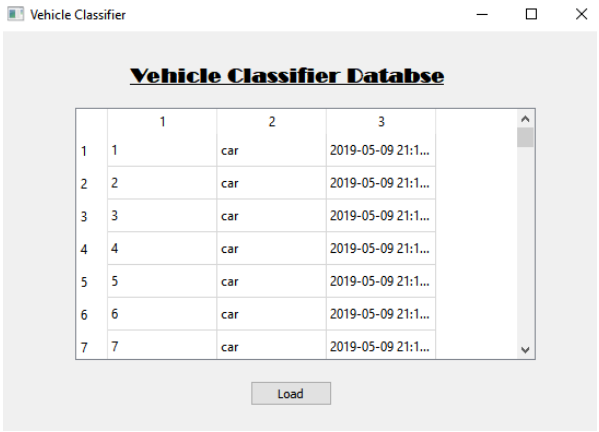

Figure 3.2 Database view

4. Using the System

4.1 Start the application

To use the Vehicle Classifier just start the application after successful installation.

The system will start processing the video feed and the results will be shown on the output window as shown in Fig 3.1.

BIBLOGRAPHY

[1] http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=787548

[2] http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=15571&arnumber=720574&punumber=5841

[3] http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&isnumber=16016&arnumber=741940&punumber=5982

[4] Zhang, L, Li, SZ, Yuan, X. Real-time object classification in video surveillance based on appearance learning. In: Proceedings of IEEE conference on computer vision and pattern recognition, MN, USA, 17–22 June 2007, pp. 1–8. IEEE.

[5] Morris, B, Trivedi, MM. Learning, modelling and classification of vehicle track patterns from live video. IEEE Trans Intell Transp Syst 2008; 9(3): 425–437

[6] Moussa, GS . Vehicle type classification with geometric and appearance attributes. Int J Civil Struct Construction Archit Eng 2014; 8(3): 273–278.

[7] Sung-Ho Bae, Jun-Eui Hong. "A vehicle detection system robust to environmental changes for preventing crime." J Korea Multimedia Soc, 13.7 (2010.07): 983-990

[8] Min-Ho Kim, Doo-Hyun Choi. “A vehicle model recognition using car’s headlights features and homogeneity information.” J Korea Multimedia Soc, 14.10 (2011.10): 1243-1251.

[9] Jong-Hwa Kim, Doo-Hyun Choi. "License plate extraction through the searching area reduction and corner templates." Proceeding of Korea Multimedia Society, (2010.5): 31-3

[10] Hyo Jong Lee. "A study on the model recognition of moving vehicles using a neural network." Ins Electron Eng Korea - Signal Processing, 42.4 (2005.7): 69-78.

[11] Nam, Yunyoung & Nam, Yun-Cheol. (2018). Vehicle classification based on images from visible light and thermal cameras. EURASIP Journal on Image and Video Processing. 2018. 10.1186/s13640-018-0245-2.

[12] Gupte S., Masoud O., Martin R. F. K. and Papanikolopoulos N. P., proposed―Detection and Classification Vehicles‖ in the March, 2002,