Bachelor of Science in Computer Science & Engineering

Development of Pedestrian Road Safety Assistance Application for Mobile Phone Users

by

Md. Jakir Hossain

ID: 1504073

Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)

Chattogram-4349, Bangladesh.

May, 2021

Development of Pedestrian Road Safety Assistance Application for Mobile Phone Users

Submitted in partial fulfilment of the requirements for
Degree of Bachelor of Science
in Computer Science & Engineering

by

Md. Jakir Hossain
ID: 1504073

Supervised by

Dr. Mahfuzulhoq Chowdhury
Assistant Professor
Department of Computer Science & Engineering

Chittagong University of Engineering & Technology (CUET)
Chattogram-4349, Bangladesh.

The thesis titled '**Development of Pedestrian Road Safety Assistance Application for Mobile Phone Users'** submitted by ID: 1504073, Session 2019-2020 has been accepted as satisfactory in fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering to be awarded by the Chittagong University of Engineering & Technology (CUET).

Board of Examiners

Supervisor

Dr. Mahfuzulhoq Chowdhury
Assistant Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Head of the Dept.

Dr. Md. Mokammel Haque
Professor & Head
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Member (External)

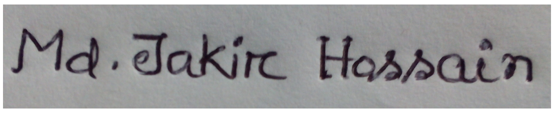
Dr. Md. Iqbal Hasan Sarker
Professor
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology (CUET)

Declaration of Originality

This is to certify that I am the sole author of this thesis and that neither any part of this thesis nor the whole of the thesis has been submitted for a degree to any other institution.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. I am also aware that if any infringement of anyone's copyright is found, whether intentional or otherwise, I may be subject to legal and disciplinary action determined by Dept. of CSE, CUET.

I hereby assign every rights in the copyright of this thesis work to Dept. of CSE, CUET, who shall be the owner of the copyright of this work and any reproduction or use in any form or by any means whatsoever is prohibited without the consent of Dept. of CSE, CUET.



Signature of the candidate

Acknowledgements

I am grateful to almighty Allah who has given me the ability to complete my project and to intend myself in performing the completion of B.Sc. Engineering degree. I am indebted to my supervisor Dr.Mahfuzulhoq Chowdhury,Assistant Professor,Department of Computer Science and Engineering, Chittagong University of Engineering and Technologyfor,his encouragement, proper guidance, constructive criticisms and endless patience throughout the progress of the project. He supported me by providing books, conference and journal papers and effective advices. From the very beginning sir always encouraged me with proper guideline so the project never seemed a burden to me. He motivated me to complete final thesis within time. I would like to thank to my defence committee members, for providing me with their valuable advices and feedbacks.I would like to thank all my teachers of our department for their sincere and active cooperation in completing the project work and invaluable assistance in every phase of my learning process over the course of four years.Finally, I would like to express my gratitude to my parents for their steady love and continuous help during my whole study period.They have always been supportive to me and inspired me become the person I am today.I am always indebted to them.

Abstract

In accordance with the increased number of people the number of various transports is rising day by day in our country. But this increased number of vehicles has also increased the road accidents. These road-accidents has been threatening our lives like many lethal diseases. Pedestrians' distraction is one reason among the various reasons of road accidents. Nowadays, pedestrians are mostly distracted by modern smartphones. Even while crossing busy roads, they remain busy over phone calls.

It would be very nice to have a mechanism by which pedestrians can be made aware of approaching vehicles while crossing roads busy with ongoing phone calls. With the recent technologies it would be preferable to have a portable device for this purpose. One such platform is a smartphone. We have developed an android based application which will entirely help the distracted pedestrians. The system implies a vehicle detection system for pedestrians through which they can easily avoid unexpected accidents and unbearable and miserable consequences due to accidents. Phone back-camera will be used for the detection scheme. Whenever a phone call is in progress, the application will be automatically started. The app will capture images by using the back-camera of the phone. A vehicle detection model is trained in offline. In online scheme this trained model will be then used for testing the captured images. If the captured image is detected as a vehicle the app will instantly give the pedestrian a warning and the phone will start to vibrate simultaneously.

The project has been tested in real life to evaluate the project for real life purpose. The test results showed us that the developed system is functioning well and attaining satisfactory result in meeting our goal. Our proposed system with smartphone's in-built back-camera and their test results can be an exquisite candidate in adaptive, user-oriented system for distracted pedestrians.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Road safety is now a major concern all over the world. Road accident is the main barrier of securing road safety for all and number of road accident is increasing rapidly throughout the world. The situation is more miserable here in Bangladesh and it is getting worse day by day. If it is not reined immediately, the situation will go beyond control. Among various reasons of road accident, pedestrians' distraction is one main reason. Pedestrians are mostly distracted today by smartphone. Even they remains engaged in phone call while crossing a busy road. This immense distraction causes road accidents and various road fatalities. The contribution of modern technology can be more prevalent in reduction of road accidents. As almost all of us bear a smartphone with us today, it would be very nice to have a smartphone-based application that can mitigate the rate of road accidents. Hopefully, our application will fulfill this demand properly. Our developed application will warn those pedestrians instantly who are distracted by ongoing phone call while crossing busy roads. This chapter will cover some introductory discussions on developing pedestrian road safety assistance application, design overview, motivation of the work, our contribution to this work, difficulties of the work, applications etc.

1.1.1 Problem Statement

The rapid development of communication technology has manufactured a variety of smartphones and numerous smartphone-based applications have been developed with the wide expansion of smartphones, such as YouTube, Facebook, Twitter, Gmail and Instagram [1]. At the time of walking people prefer to use mobile phones because they remain busy with works most of the time in office or in house. Though walking gives us physical and mental amenities, talking over mobile phone while walking causes traffic accidents. Scientific research has shown us that smartphone conversations divert its user, which has an important effect on passenger safety; for instance, a smartphone user who is engaged in a phone call while crossing a road is usually more at danger than other passengers who do not join in such conduct. Now road safety issue is being more important day by day. A movement was made by the school children in 2018. As per official data, there were at least 3334 deaths and 3740 injuries were reported in 4114 fatalities in 2018 in Bangladesh. It is predicted that the real deaths could be between 10,000 and 12,000 each year [2].The situation deteriorated in the next year. A statistic shows that total 4702 accidents, 5227 death, 6953 injuries reported in only 2019 [3].

For reducing the road fatalities caused by using smartphone while walking, an android based smartphone application is being developed here that will help individuals talking over phone while walking and improves the safety of pedestrian. This application will definitely mitigate the alarming rate of accidents by providing the opportunity of making pedestrians more conscious of surroundings (incoming vehicles) by a vibration warning while walking and talking to others over mobile phone.

For implementing the system Android platform is required. Android is an operating system created by Google for smartphones, based on the Linux kernel and other open-source software, intended mainly for mobile touchscreen applications such as smartphones and tablets. On the other side, for developers, Android Studio is the official Integrated Development Environment (IDE) for developing Android apps.

We hope that our system will be useful for pedestrians to ensure the safe-walking on road and that will surely reduce the road accidents and consequently, the death rate due to these road causalities will be mitigated day by day.

1.1.2 Objectives

The purpose of this project is to ensure the safety of pedestrians. The main objective and possible results of this work may be set out in the following:

- To design and develop an application of warning pedestrians about approaching vehicles while crossing a road during smartphone conversation.
- To develop an application for the pedestrians who get engaged in staring at phone screen during walking along the road.
- To assess the efficiency of the application in a true setting.

1.2 Framework/Design Overview

The general framework of our system consists of some basic parts as shown in Figure 1.1

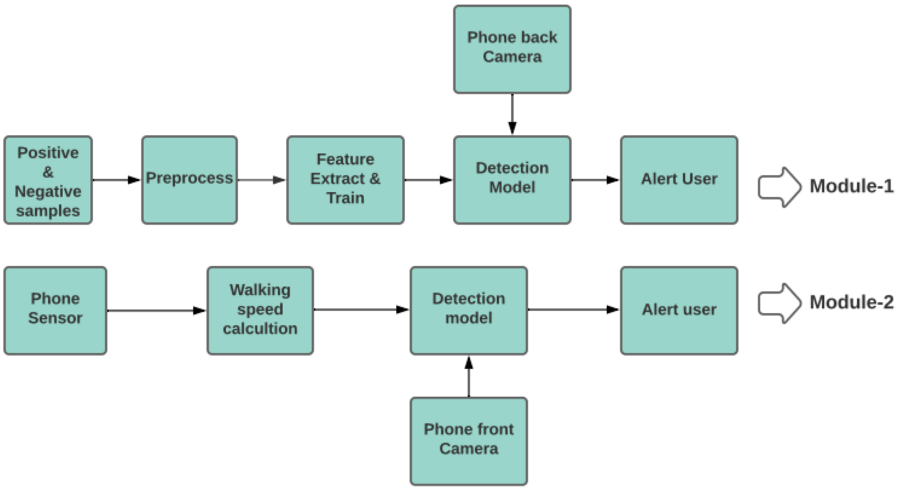


Figure 1.1: Design overview of our application system.

This framework depicts the overall working principle of our application.In our first module,training samples are collected. Then collected positive and negative samples are preprocessed. Next stage is training stage and using the processed samples the desired model is trained with the help of cascade trainer GUI. The trained model is then integrated to the mobile application. The application will open up the phone back-camera conditionally in the background. Finally, if any vehicle is detected then user will get a vibration alert instantly. In the second module, using accelerometer and gravity sensor walking speed of user is calculated. If the speed exceeds the threshold value, then with the help of face eye detection model user's staring at phone screen situation is measured by using phone front camera. If condition satisfies then user get alert notification.

1.3 Difficulties

Difficulties refers to facing any kind of hinderance or hardship while conducting a project. There can be various problems that may arise for technical or environmental issues and so on. We faced several difficulties while doing the whole project.

- We faced difficulties in collecting positive image samples as we did not find our country's vehicles (bus, truck, CNG etc.) dataset.
- We had to ensure that the positive images must contain as little background as possible. For ensuring that we had to crop all the positive images manually which was absolutely a challenging task.
- Due to hardware limitations of laptop I had to faced too much difficulties in implementation.
- We faced difficulties in taking phone sensors (accelerometer, gravity sensor) provided value according to our need.
- Integrating the trained model to an android application also arose difficulties. Internal tasks of android such as- opening the application at the time of phone call, opening the phone's back-camera with the opening of application, closing the application with the termination of phone call also created difficulties.

1.4 Applications

Various smartphone applications are becoming popular among our people day by day. Our application will be able to alleviate the rate of road accidents. Some important applications of our system are mentioned bellow:

- In securing road safety of distracted pedestrians.
- Our model can be used in self-driving vehicle system with some modifications.
- In making people more aware of utilizing mobile phones.
- Will be helpful for traffic police as pedestrians will be aware of incoming vehicles towards the pedestrians.

1.5 Motivation

The motivation that works behind the implementation of this system is reducing the death rate caused by road accidents. The development in various fields of modern technologies paves us the way to finalization of the scheme.

In this era of modern technology, people's living standard is improving day by day through various types of amazing technologies. Smartphone is one of them which has brought a radical change in our daily life. Now almost everyone carries a smartphone with him/her as it is becoming more affordable for all and available in everywhere. According to a statistic, there were 165.8 million mobile connections in Bangladesh in January 2021.The number of mobile connections in Bangladesh increased by 1.7 million (+1.1%) between January 2020 and January 2021 and a huge number of them are smartphone users. So, if we can utilize this opportunity of using smartphones as a road accident reducer that will be great. This thought acted as motivation behind developing the system [4].The BTRC data showed that about 102.3 million users use the internet through mobile operators [5].This indicates that a rich number of people use smartphone today.

Nowadays, smartphones fitted with sensors [6] (accelerometer, gyroscope, GPS and camera) have become cheaper and more widely used, so it can be claimed that smartphones are the ideal "sensor" that can be used to prevent accidents by notifying users of any incoming vehicles during phone conversations.

Texting on phone and walking along the road simultaneously is a new tendency of our generation. Here user stares at phone screen for a certain period of time and consequently faces various troubles. It would be very nice having an application that can prevent this unwanted occurrences.

As soon as we can mitigate the road accidents, that will be better for us. All of us want to move here and there safely without any causalities. Road accidents is the main hinderance to our demand. The accidents, caused by the distraction of pedestrians, can easily be avoided being concerned about approaching vehicles by using modern technologies like smartphone.

1.6 Contribution of the thesis

Some text in Contribution.Every research work project is conducted to gain a specific set of goals, in regards of the accuracy of results or data processing time. The contributions of our system are given bellow:

- Developing a smartphone based pedestrian road safety assistance application utilizing phones own sensor (back camera) without any needing external sensor.
- Creating a 'vehicle dataset' of our country's vehicles and implement it in training the desired model.
- Combining two very much needed modules for securing safety of general pedestrians.
- Applying the trained model in android application for gaining the real-time use reliability.

1.7 Thesis Organization

The entire report is represented in five main chapters. We can depict report structure as follows:

- **Chapter 1** gives an overview of our application development, motivation behind the work. It also covers the project objectives, project applications and difficulties that we faced in developing the application.

- **Chapter 2** describes the related literature reviews and implementation challenges of this project.
- **Chapter 3** provides the overview of the framework, detailed explanation of it and the implementation technique.
- **Chapter 4** gives an overview on the dataset used to train our model as well as impact analysis of the project. This also covers the evaluation of performance and experimental result.
- **Chapter 5** includes the conclusion of our work and provides some future development recommendations as well.

1.8 Conclusion

In this chapter, an overview of this work is provided. The motivation behind this work difficulties, applications and contributions are also stated here.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we will shortly discuss the terminologies related to the project which are important to understand. Our project is related to android application development. So, various terms, techniques of android application development will be discussed in these sections. This chapter will also contain a brief discussion on related previous works. The implementation challenges of our work will also be covered by this section.

2.2 Android

Android is a Google-developed mobile operating system (OS) based on the Linux kernel.Android is primarily intended for touchscreen mobile devices such as smartphones and tablets, but it also includes enhanced user interfaces for televisions (Android TV), automobiles (Android Auto), and wrist watches (Android Watch) (Android Wear). Google released the source code of android and various devices are shipped with a combination of open-source and software [7].Because of its tremendous popularity ,this OS has been a main target for patent disputes as a participant of the "Wars of smartphone" among various companies.The figure 2.1 shows a sample of smartphone.

2.2.1 Java

Java is a concurrent,based on class, object-oriented programming language that was created with the goal of having as few implementation dependencies as feasible. It is developed to let the developers “write once, run anywhere” (WORA),[8].This

Figure 2.1: Android smartphone.

means that generated Java code can run on any platform that supports Java without needing to be recompiled.The Java Development Kit (JDK) provides a programming environment for creating Java applications, applets, and components.The JDK contains tools for designing and testing Java-based programs.

2.2.2 Broadcast Receivers

Similar to the publish-subscribe design pattern, Android apps may transmit and receive broadcast messages from the Android system and other Android applications.When a noteworthy event occurs, these broadcasts are sent out.When the Android system powers up or the device starts charging, when the device's connectivity changes, or when the phone call state changes, for example, the Android system sends out broadcasts.Applications may also send bespoke broadcasts to inform other apps about anything they might be interested in (for example, some new data has been downloaded).

Apps can sign up for particular broadcasts. When a broadcast is sent, the system sends it to applications that have subscribed to receive that sort of broadcast. Broadcasts can be used as a message mechanism across apps and outside of the typical user flow, in general. However, you must be careful not to take advantage of the ability to respond to broadcasts and do operations in the background, since this might cause the system to slow down.There are two types of transmissions that we can receive:

- **Normal broadcasts** (sent with Context.sendBroadcast) are completely incongruent. The broadcast is received by all receivers in an unspecified sequence, typically at the same moment.This is more efficient, but it also implies that receivers will be unable to utilize the result or abort APIs that are included.

- **Ordered broadcast** (sent with Context.sendOrderedBroadcast) are sent to just one recipient at a time.Each receiver can either propagate a result to the next receiver or fully abort the broadcast so that it is not passed on to subsequent receivers as it runs in turn.

There are following two important steps to make BroadcastReceiver works for the system broadcast intents-

- Creating the Broadcast Receiver.

- Registering Broadcast Receiver.

2.3 Related Literature Review

A process is described in [9] where Tung et al. offered a cliched method without having any previous understanding of the user's environment by estimating distances to neighboring objects using phone speakers and microphones. This work doesn't specify the actual position of incoming vehicles and the right direction of the road. A work has been conducted in [10] where Chen et al. have worked on the implementation a collision prevention scheme which instantly tells pedestrians potential hazards. The work of [10] pointed to this and a system 'InfraSee' is implemented which is capable of detecting radical change of floor for pedestrian smartphone users and adds smartphones with a tiny infrared sensor that measures the length from the sensor to the ground. In the work of [11] a new device hase been suggested that helps smartphone users to move outdoors more securely. The suggested scheme is introduced on a mobile and utilizes its rear camera to identify the present external context, e.g. road intersections, sidewalks, and lastly alerts the user to hazardous circumstances using noise from the mobile.In [12] Xue et al. have proposed an Android smartphone-based system that detects the walking behaviour of pedestrians by leveraging the sensors and front camera on smartphones.To assist pedestrians Qianru et al. [13] have suggested 'CrowdWatch', a system that exploits self-powered mobile sensing to fine-grain the field of sensing and offers context-aware warnings to passengers about their risks.In the work of [14], Shikishima et al. concentrated on the issue of texting while walking, have suggested a new technique of tracking of texting without cameras used by the multiple past researches. In that document, they detected texture while walking, using the phone's posture and the impulse of his head. The study trend focuses on the impact of smart phones on passenger safety due to the widespread use of phones as individuals move around in their daily lives.Some apps are being developed that play as a third eye to alert users when they step out from the secure area where no visual flooring is identified [15].

2.4 Implementation Challenges

Every project implementation brings some challenges which have to overcome for the proper completion of that project. In case of our project, there also arose some challenges.

- In model our training, if positive images would contain background (negative images) the result will be very poor. For that case we had to crop all the positive images manually which was absolutely a challenging task.
- When we came to integrate the model to android, it was a challenging task. We faced challenges in opening the application at the time of phone call, opening the phone's back-camera with the opening of application, closing the application with the termination of phone call.
- As we have used Java OpenCV camera view, we had to integrate OpenCV Library to our android project which was really a challenging task.
- Hardware limitations of laptop also threw in challenges as it was not perfect enough for training our model and implementing in android studio properly.

- We could cover only one side (in which side mobile phone is held close to ear),other side is totally blocked by our head.So,we could not implement our first module for that blocked side.It is one kind of limitation of our application.

2.5 Conclusion

A detailed description of the literature review is provided in this chapter. The discussion was divided into two parts-the terminologies and the related previous work section. It concludes that there are only a few works on our project topic. There are lots of opportunities in this specific field.

Chapter 3

Methodology

3.1 Introduction

In this chapter, we will mainly focus on methodology of our proposed system and learn about the module of our system. The overall system architecture of the proposed system will be discussed here.

3.2 Diagram/Overview of Framework

This type of system development necessitates the solution of a detection type problem that will detect vehicles from online (real-time) input in the first module.The following figure 3.1 will demonstrate our system properly.

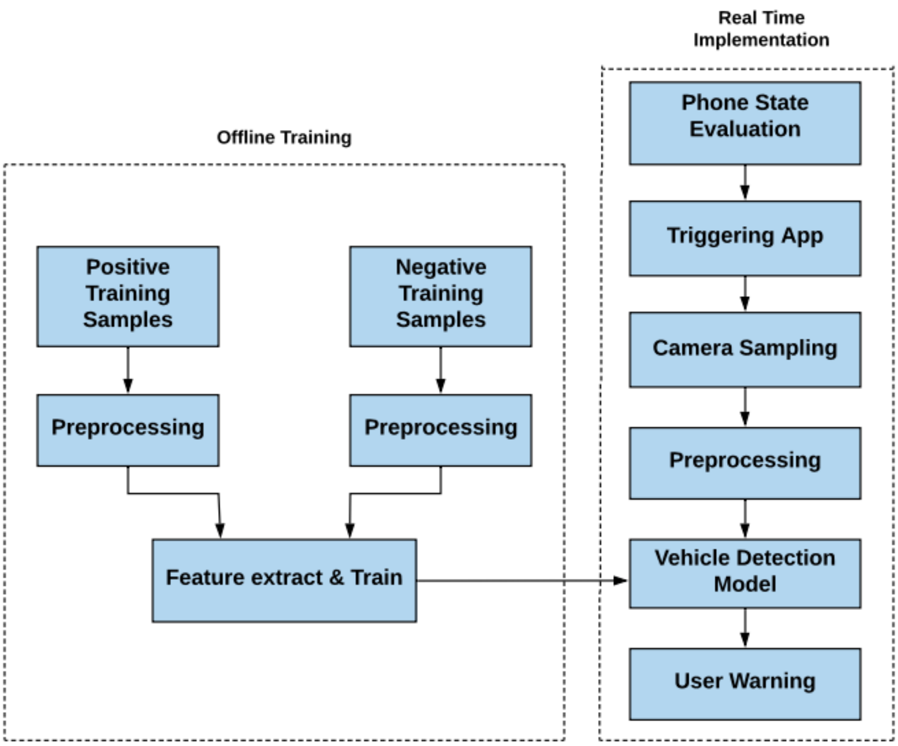


Figure 3.1: Pedestrian road safety architecture during ongoing phone calls.

In our second module, phone user's speed will be used along with face-eye detection model for warning while walking alongside the road with mobile phone in hand. The second module is represented in figure 3.2

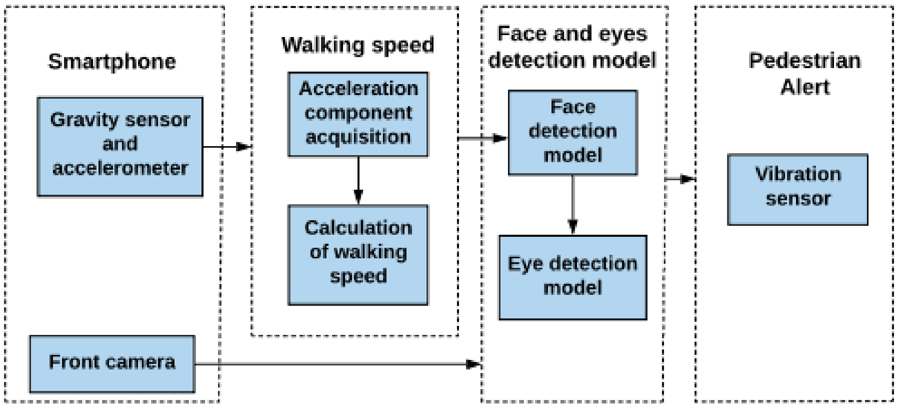


Figure 3.2: Pedestrian road safety architecture while staring at phone screen.

3.3 Module-1: While crossing road

The proposed system protects pedestrians who are distracted by phone conversation while crossing the roads by using the phone's back camera to identify incoming cars, alerting the user through sound notifications and vibrations. The system utilizes the android APIs for executing the vehicle detection module only during ongoing phone calls. The whole procedure consists of offline training and real-time implementation.

3.3.1 Offline Training

By using the dataset, comprised of positive and negative images, a model is constructed and this model is then used for detecting positive matches in real-time on the phone. Here the offline training module can be demonstrated in following four steps:

3.3.2 Dataset building

A dataset is actually a collection of data. The dataset building implies the technique of collecting the relevant data for the training session and this training dataset has huge impact on the final result. Here our dataset consists of (a) positive training sample and (b) negative training sample (side view of vehicles, surrounding of vehicles).

3.3.3 Preprocessing of training images

Vehicle region is properly cropped so that we can take as little background as possible in the positive samples because this background can have a great impact on detection model. The images are resized to same pixel. Color variety of various vehicles may arise problem in the detection scheme. So, we transformed the cropped images into gray scale images. The negative images are created by cropping large background images and converting them to grayscale.The following figure 3.3 represents some preprocessed images of our dataset.

(a) (b)

Figure 3.3: Preprocessed Training samplese:(a) positive, (b) negative.

3.3.4 Feature Extraction

From previously preprocessed images, we extracted haar features for using in training our model. For extracting haar features we have used cascade trainer GUI tool which extracted the most relevant features and trained the model properly.The figure 3.4 shows haar feature extraction from vehicles [16].

3.3.5 Real-Time Implementation

The vehicle detection module runs on Android smartphone in real-time usage. This implementation includes the following steps:

Figure 3.4: Haar-like feature examples for describing a vehicle's appearance.

3.3.6 Capturing video

The application immediately starts capturing video in backgroud by phones back camera during only active phone calls for saving battery life-time. Vehicle is detected by observing this real-time video with the help of offline trained module.

3.3.7 Detecting vehicles

The classification model which was built during offline training session uses the real-time video here to detect vehicles. If an area of interest passes all of the criteria that the model defines, then the app will recognize it as a vehicle.

3.3.8 Dispatching warning

If the particular region is recognized as a vehicle by the trained model then the app will give a vibration warning that will make user aware of the incoming vehicle.Detecting vehicles and dispatching vibration warning feature is shown in figure 3.5.

Figure 3.5: Detecting vehicles and dispatching a vibration alert

3.4 Module-2: While walking along the road

This portion is applicable only when the user is engaged in staring at phone screens during walking. Here the system detects the walking behaviors of pedestrians by using smartphone sensors and the front camera and enhances the safety of pedestrians.

3.4.1 Walking speed estimation

Pedestrian walking speed is estimated by using accelerometer, gravity sensor on phones. The sensor coordinate system of smartphone is shown in figure 3.6. For estimating walking speed the following procedure is followed.

1.Setting hitCOUNT,hitSUM,hitRESULT as 0

2.Setting sample_SIZE as 50 and Threshold as 0.53

3.Assign 0 to Accel [acceleration apart from gravity]

4.Acquire acceleration components (x,y,z) *onSensorChanged()* method.

5.Setting AccelLAST = AccelCURRENT

6. Setting AccelCURRENT as $\sqrt{(x^2 + y^2 + z^2)}$

7.Diff = AccelCURRENT- AccelLAST

8. Setting Accel as Accel*0.9+Diff

9. if (hitCOUNT <= sample_SIZE) then

Increase hitCOUNT by 1 and increase hitSUM by abs(Accel)

else

hitRESULT = hitSUM / sample_SIZE;

10. if (hitRESULT > Threshold) then

Give the vibration alert.

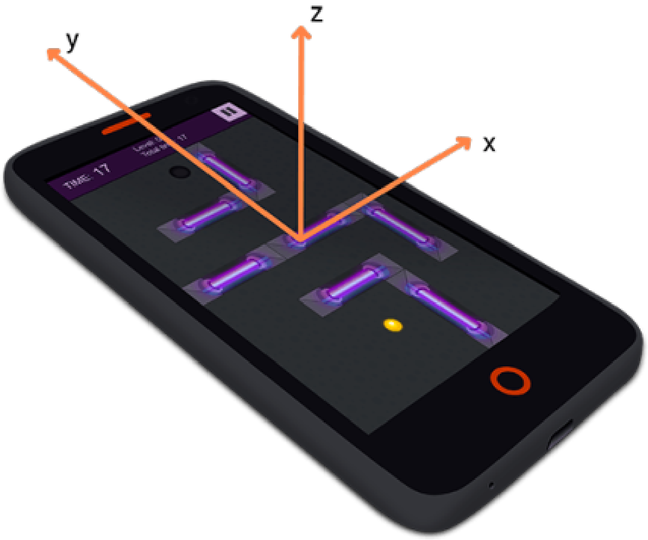


Figure 3.6: Smartphone reference frame.

3.4.2 Face and eye detection

When the walking speed of a pedestrian exceeds the threshold of a particular speed (1.2 mph/0.53 m/s) for a fixed period (such as 7s), the system then begins the next module, face and eye detection scheme. For detecting face and eye we have used pretrained haar cascade classifier offered by OpenCV.

3.4.3 Pedestrian Alert

If the detection model ensures that the pedestrian is in the state of staring at screens more than 7s with over-threshold speed while walking, then the pedestrian will get a alert as a toast message on the phone screen which is shown in figure 3.7.

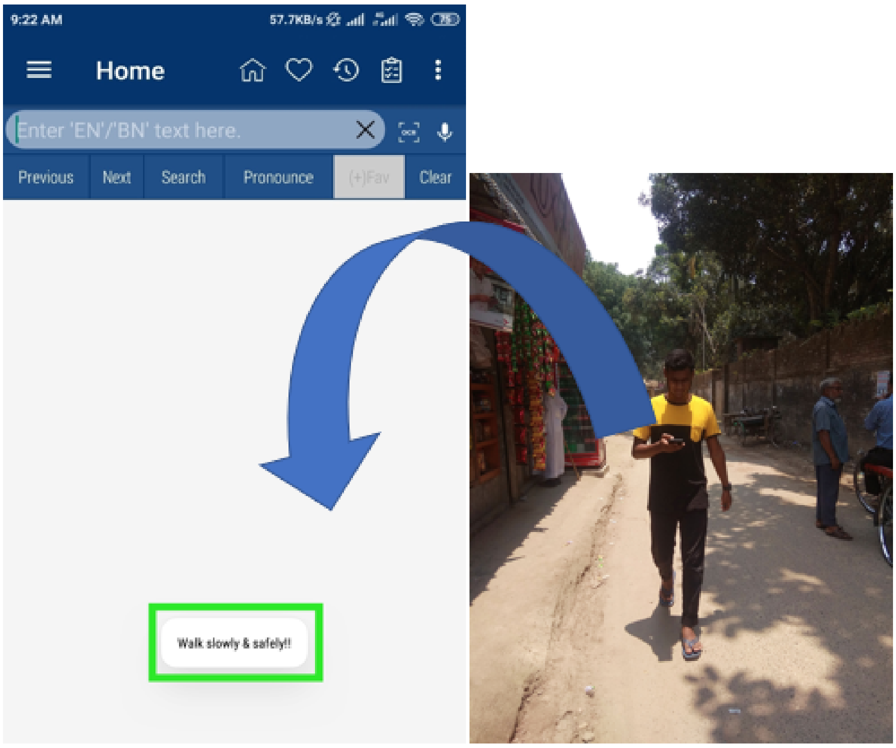


Figure 3.7: Alerting user with a toast message while walking and using dictionary on phone.

3.5 Implementation

The implementation of this system requires designing and development of a smartphone-based software that is able to facilitate the real-world vehicle detection and face-eye detection implementation in a disciplined and convenient way. Various required resources and related modules will be covered in this implementation section.

3.5.1 Required Resources

The required implementation resources can be categorized as follows:

- Hardware Tools:

  1.Personal computer.

  2.Android smartphone.

- Software tools:

  1.Android Studio

  2.Android SDK

- Programming Language:

  1.Java

3.5.2 Camera View

As our application is a detection type application, we had to use the phone camera for capturing real-time video. We have used Java OpenCV camera view for our project. For accessing OpenCV camera and other libraries we had to integrate OpenCVLibrary to our project. Specifically, we integrated OpenCVLibrary3410 which is actually contained in opencv-3.4.1-android-sdk.

3.5.3 Switching Module

As we have two modules, we have introduced a switch widget in our application for switching from one module to another one. The attached switch has two states:

- **Module-1:** When the switch is set to this state, during ongoing phone calls our application will be opened up automatically and do the warning task perfectly while crossing busy road.

- **Module-2:**This module is applicable when the user is walking along the road with using phone (as usual usage of phone such as-facebook, twitter, gmail, dictionary, online news reading etc.) simultaneously. Here if the walking speed of phone user crosses the threshold value and user stares at phone screen for a certain period of time, then the application will be opened up and give warning accordingly.

Figure 3.8: Two modules of our application:(a)Module-1, (b) Module-2.

3.5.4 Phone State Evaluation

For our first module, our application will only be triggered to be opened automatically when the mobile phone is engaged in a phone call (incoming/receiving call), we needed to evaluate the present phone state. With the help of Broadcast Receiver, we determined the phone states. For doing that we created Broadcast Receiver and then registered the receiver.However, there are three available states of a phone:

- **CALL_STATE_OFFHOOK:** At least one call exists that is dialing, active or on hold and no calls are ringing or waiting.
- **CALL_STATE_RINGING:** A new call arrived and is ringing or waiting.
- **CALL_STATE_IDLE:** No phone call exists or a phone call has just terminated.

The following code snipped within the Broadcast Receiver determines the phone states.

```
if (((intent.getStringExtra(TelephonyManager.EXTRA_STATE).equals(TelephonyManager.EXTRA_STATE_OFFHOOK))) ||
        (intent.getStringExtra(TelephonyManager.EXTRA_STATE).equals(TelephonyManager.EXTRA_STATE_RINGING))) {
    Intent i = new Intent(context, MainActivity.class); /// Offhook and ringing states
    i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    context.startActivity(i);
```

Here we see that when our application is in module-1, then the application will only be opened when the phone is in offhook state or ringing state. When the phone is in idle state, there is no need to open the application.

3.5.5 Permissions Setup

There are certain types of permissions required which are declared in the manifest file.

- Camera permission
- Phone State permission
- Vibration permission.

3.6 Conclusion

In this chapter, we focused on dataset used to train the detection model, impact analysis, various types of impact related to the project. Various types of impact defined the actual impact that the application can put in different sectors. Lastly, we discussed about project evaluation.

Chapter 4

Results and Discussions

4.1 Introduction

In this chapter, we will give an overview of the dataset used to train the model as well as various impacts i.e. social, environmental and ethical impact of the project and evaluation of framework and performance.

4.2 Dataset Description

Collecting dataset is a challenging task for creating any detection type model.As our application is related vehicle detection,we needed vehicle dataset.When we came to collect dataset the problem arose before us that all the available vehicle datasets includes the vehicles of foreign country.But ehicles of our country like bus,truck,CNG etc. are not included in any dataset.So, we started collecting the vehicle images of Bangladesh.We have collected more than 1000 vehicle images as positive sample.Roadside surroundings and empty roads are used here as negative sample.Negative sample images collection was comparatively easy.We collected about 1000 images as negative sample.In both cases collected images were preprocessed for using in offline training phase.

4.3 Impact Analysis

Project impact defines how a project affects the matters when it comes in deal with them. Analysing a project impact refers to analysis the positive and negative impacts that the particular project is supposed to have on society, environment, ethics and so on.

4.3.1 Social and Environmental Impact

We have developed a pedestrian road safety application. Every year many pedestrians meet various road accidents due to their immense distraction of remaining busy over phone call while crossing roads. This distraction costs them a lot. With the help of our application the distracted pedestrians can be made more aware of nearer hazards while crossing a busy road. Consequently, road accidents rate will be reduced significantly. This reduction will pave the way of giving our people a lameness-free life. Though this rate won't be noteworthy but this will cause a huge positive change in our society as well as environment.

4.3.2 Ethical Impact

Our project has ethical impact too. If our system is implemented properly, there will arise ethical issues. Every user should bear in mind that ethical usage of anything pays a lot but unethical use costs us a lot also. Proper utilization of this application will cause a positive impact and this is ethical use of application. But if the project is applied in a wrong way that will cause a negative impact also and this is to be considered as unethical use of the application.

4.4 Evaluation of Performance

In this section, we will discuss about the evaluation measure of our system. For evaluating the real-world reliability of first module of our application, we had a 40 minutes real-world experiment.From the experiment we calculated the highest distance of detection. It was observed that 25 meters far away vehicles from the pedestrian can be detected by the application. We noted down the true positive and false positive rate from the experiment. False positive rate was significantly low compared to true positive rate.These false positive results arose due to the presence of side sight of vehicles.Similarly, for the second module we had real-world experiment and observed various accuracy measuring factors.

4.5 Experimental Result

In this section we will discuss about various types of experimental results and analysis for both the module-1 and module-2.Different types of curves will be presented for easily visualising various measured values.

4.5.1 Module-1

Here we will discuss about the real-world experimental results of first module that means while pedestrian is crossing road with ongoing phone calls on his phone.

4.5.1.1 Case Study

We experimented the real-world reliability of our application considering various cases such as:distance between user and vehicle, speed of vehicle,shape of vehicle, brightness of day-light etc.

4.5.1.2 Case-1: Distance between user and vehicle

For experimenting this case, we varied the distance between user vehicle in three phases and noticed the results.

- **First phase:** Here noticed on the vehicles that were 25 meters (approx.) away form the user.Above that distance, the application was hardly detecting vehicles.Thus, we assumed that the maximum detection distance is 25 meters. The following table 4.1 shows this observation.

Table 4.1: Vehicle detection result for phase-1

Vehicle direction	**Front View**	**Rear View**
No. of vehicles detected	18	16
No. of vehicles missed	4	4
True Positive Rate	**81%**	**80%**
No. of false positive	4	
Detection distance	25 (meter)	

- **Second phase:** Here we decreased the distance to 20 meters. With the decrease of distance between vehicles and user, we got better accuracy.The following table 4.2 shows the result for detection distance 20 meters.

Table 4.2: Vehicle detection result for phase-2

Vehicle direction	Front View	Rear View
No. of vehicles detected	32	19
No. of vehicles missed	6	4
True Positive Rate	**84%**	**82%**
No. of false positive	2	
Detection distance	20 (meter)	

- **Third phase:** In this phase the detection distance between vehicle and user was 15 meters and less than that in some cases.As we decreased the distance, we got more accuracy than the previous one. The following table 4.3 shows the related results for this case.

Table 4.3: Vehicle detection result for phase-3

Vehicle direction	Front View	Rear View
No. of vehicles detected	19	17
No. of vehicles missed	3	3
True Positive Rate	**86%**	**85%**
No. of false positive	2	
Detection distance	15 (meter)	

4.5.1.3 Curve Analysis

The graphical comparison among these phases is shown in figure 4.1.

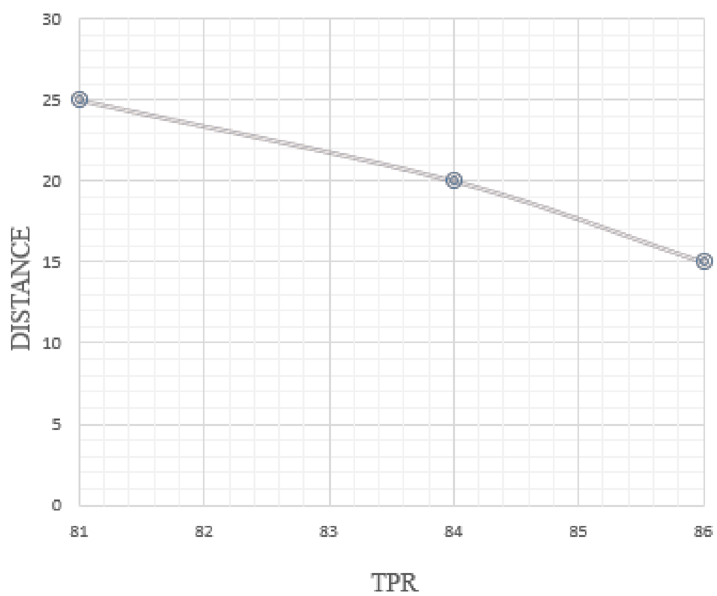


Figure 4.1: Distance versus True Positive Rate (TPR) for front view.

Here we can see that for front view of vehicles with the increase of distance between vehicle and user, resulting accuracy (true positive rate) decreased but at the minimum distance we got better accuracy.

Similarly for rear view, with the decrease of distance between vehicle and user, resulting accuracy (TPR) got improved.The figure 4.2 shows that result for rear view.

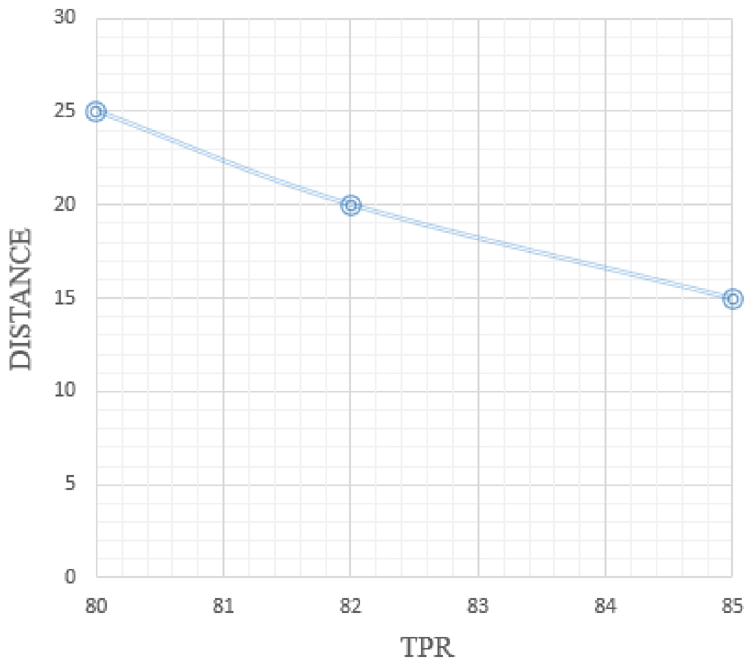


Figure 4.2: Distance versus True Positive Rate (TPR) for rear view.

The following figure 4.3 represents the no. of false positive variation according to distance between user and vehicles.

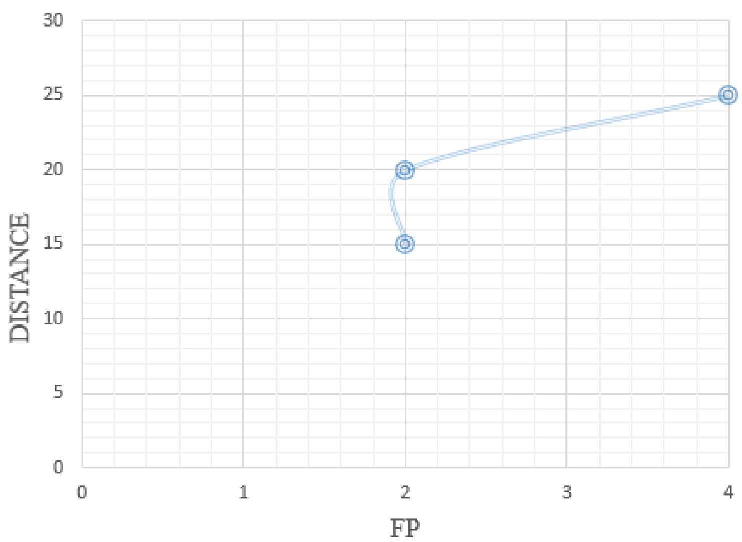


Figure 4.3: Distance versus No. of false positive (FP).

Here we can see that number of false positive is limited from 2 to 4.It starts from 2 for 15 meters,2 for 20 meters and ended with 4 for 25 meters.

4.5.1.4 Case-2: Speed of vehicles

Detection accuracy varied with vehicle speed variation.For different speed of vehicles we got different values of TPR.The following table 4.4 shows the variation of result in terms of vehicle speed.

Table 4.4: Detection varitation in terms of vehicle speed.

Vehicle Speed	**Low speed**	**Medium speed**	**High speed**
Vehicles Passed	22	25	22
Vehicles Detected	19	21	18
Vehicles Missed	3	4	4
True Positive Rate	**86%**	**84%**	**82%**

- For fast speed of vehicles, detection accuracy was low that means true positive rate (TPR) was low as well.
- For medium speed of vehicles, detection performance and true positive rate were better than that of fast speed of vehicles.
- But for slow speed of vehicles, detection accuracy was much more improved and resulted in higher true positive rate.

4.5.1.5 Case-3: Shape of vehicles

The actual shape (small/large) of vehicle played a vital role in detection accuracy also.The following bar-graph 4.4 depicts the related result.

- The larger-shaped vehicles (bus,truck,private car etc.) was being detected more accurately.Hence true positive rate was high in this case.
- The detection accuracy was comparatively low for the small-shaped vehicles (CNG,auto-rickshaw etc.).Thus true positive rate was low here.

4.5.1.6 Case-4: Day-light variation

Day-light also played a vital role while experimenting.In bright and dull day-light detection accuracy was different.Figure 4.5 represents the related results.

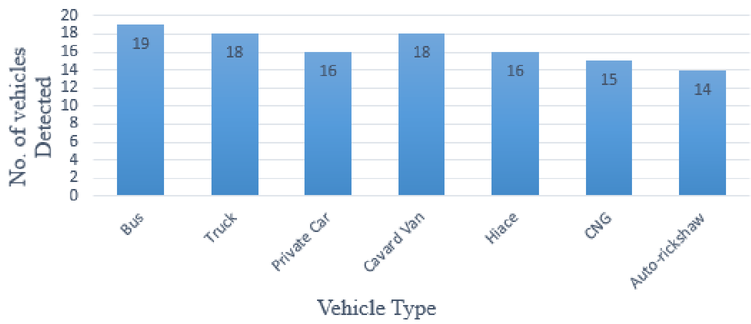


Figure 4.4: Detection comparison in terms of vehicle shape (20 vehicles).

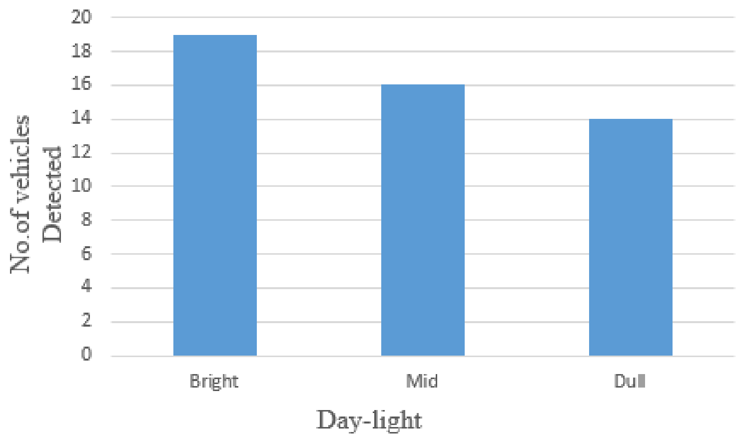


Figure 4.5: Vehicle detection comparison in terms of day-light variation(20 vehicles).

- In dull day-light detection accuracy was low and that caused lower true positive rate.
- In mid-level day-light, detection performance was better than that of dull day-light.
- But in bright day-daylight, the application was detecting vehicles properly and that resulted in higher true positive rates.

4.5.1.7 ROC curve

The Receiving Operating Characteristic (ROC) curve of our framework is shown in figure 4.6.

4.5.2 Module-2

Here we will present about the real-world experimental results of second module that refers while pedestrian is walking along the road and staring at his phone

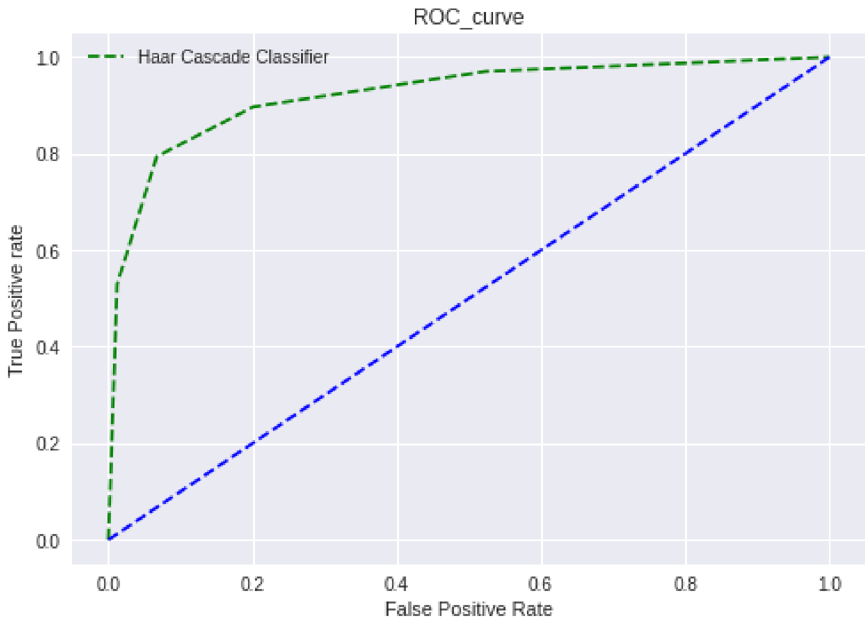


Figure 4.6: ROC curve for HAAR feature-based cascade classifier.

screen.

4.5.2.1 Step Analysis

For estimating walking speed, we analyzed the walking steps of pedestrian on the basis of acceleration of walking variation. The figure 4.7 represents this step observation.

Here we see that acceleration in the Y-axis increases at the time of stepping up but decreases at the time of stepping down.

4.5.2.2 Acceleration and speed observation

We closely observe the acceleration and speed of the user during real-time experiment for our second module. We noted down the value of acceleration speed and plotted them. The following figure 4.8 shows the acceleration along with speed.

4.5.2.3 Accuracy measure

From our experiment we observed the true & false alerts.Alert, when user is in unsafe situation, is referred to as true alert (true positives) and rate of these alert is true positive rate. On the other hand, any alert when actually alert is not

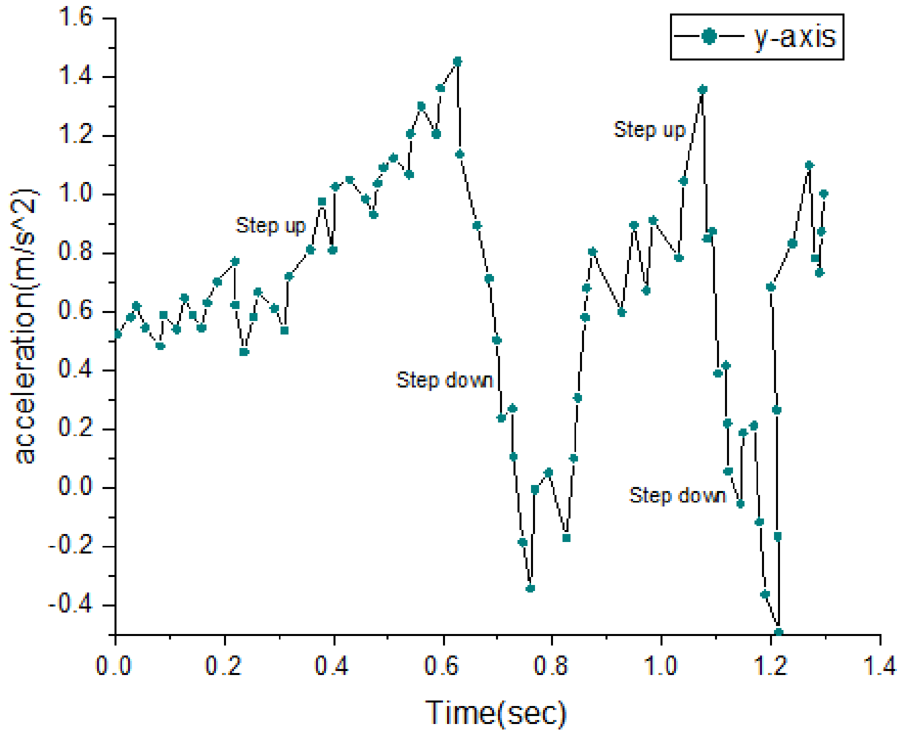


Figure 4.7: Acceleration in the Y-axis

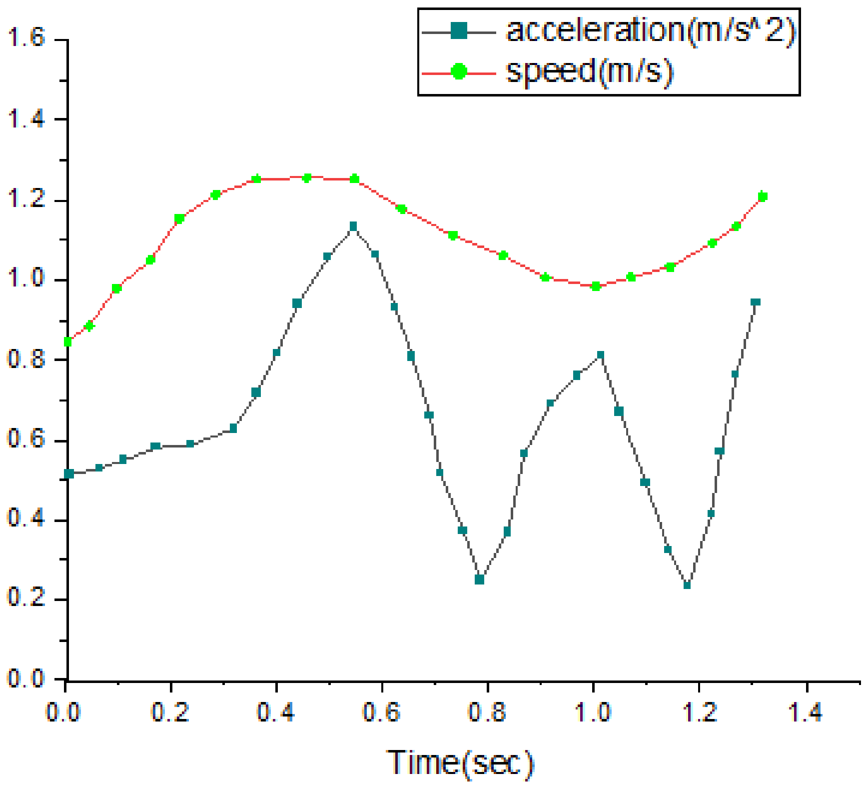


Figure 4.8: Acceleration and speed analysis

necessary is considered as false alert (false positives) and rate of these alert is false positive rate.

From our 30-mins experiment we observed, participant got 96 alerts but actually should have got 104 alerts. Among those 96 alerts, 90 alerts were successful (true positive) and rest 6 alerts were unsuccessful (false positive).These experimental results are shown in table 4.5. Finally, we measured the true positive and false positive rate for observing accuracy. Our application gives a true positive rate of **85%** which is enough for giving user alerts in time and a false positive rate of **6%**.

Table 4.5: Accuracy measure in real-time experiment

User Alert	Measured Values
Number of alert should be given	104
Alerts occurred	96
Successful alerts	91
False alerts	5
True positive rate	**85%**
False positive rate	**5%**

4.6 Conclusion

In this chapter, we focused on dataset used to train the detection model, impact analysis, various types of impact related to the project. Various types of impact defined the actual impact that the application can put in different sectors. Lastly, we discussed about project evaluation.

Chapter 5

Conclusion

5.1 Conclusion

This project demonstrates a discussion on developing a pedestrian road safety assistance smartphone application.The design concept, initial prototyping and evaluation of the application is presented in this project which will helpful for people who speak over phone,walk and cross roads at the same time.Hopefully,this application will pave the way of mitigating the rate of road accidents caused by pedestrians' distraction in our country.Our system has some limitations which will be overcome in future update of the application.

5.2 Future Work

The purpose of this project was to detect vehicles in right time when a distracted pedestrian need to be warned about approaching vehicles and to warn pedestrians who walk and use phone at the same time. Though the application gives proper warning, it can be improved in some areas:

- Enriching the dataset by including more types of vehicles.
- Improving the detection model for increasing accuracy.
- Adding location based (urban and rural) service for saving battery power.
- Including eye movement analysis,eye focusing etc. for improving the accuracy of second module.
- Reduction of power consumption by disabling the app with the help of accelerometer on the phone while the pedestrian is not walking.

References

[1] K. K. Kapoor, K. Tamilmani, N. P. Rana, P. Patil, Y. K. Dwivedi and S. Nerur, 'Advances in social media research: Past, present and future,' *Information Systems Frontiers*, vol. 20, no. 3, pp. 531–558, 2018 (cit. on p. 1).

[2] M. Hoque, S. Mahmud, A. Siddiqui and C. Kawsar, 'Road safety in bangladesh and some recent advances,' in *Proceedings of the Road Safety on Four Continents Conference*, Conference Sponsor, vol. 14, 2017, 12p–12p (cit. on p. 2).

[3] *Monitoring lapses and indifference to blame.* [Online]. Available: `https://www.thedailystar.net/frontpage/bangladesh-road-accidents-in-2019-stats-by-nirapad-sarak-chai-1849588` (cit. on p. 2).

[4] *Digital 2021: Bangladesh.* [Online]. Available: `https://datareportal.com/reports/digital-2021-bangladesh` (cit. on p. 5).

[5] *Number of mobile phone users reaches 170.1 million.* [Online]. Available: `https://thefinancialexpress.com.bd/national/number-of-mobile-phone-users-reaches-1701-million-1610373612` (cit. on p. 5).

[6] S. Ali, S. Khusro, A. Rauf and S. Mahfooz, 'Sensors and mobile phones: Evolution and state-of-the-art,' *Pakistan journal of science*, vol. 66, no. 4, p. 385, 2014 (cit. on p. 5).

[7] L. A. Allison and M. M. Fuad, 'Inter-app communication between android apps developed in app-inventor and android studio,' in *Proceedings of the International Conference on Mobile Software Engineering and Systems*, 2016, pp. 17–18 (cit. on p. 8).

[8] S. Blom, M. Book, V. Gruhn, R. Hrushchak and A. Köhler, 'Write once, run anywhere a survey of mobile runtime environments,' in *2008 The 3rd International Conference on Grid and Pervasive Computing-Workshops*, IEEE, 2008, pp. 132–137 (cit. on p. 8).

[9] Y.-C. Tung and K. G. Shin, 'Use of phone sensors to enhance distracted pedestrians' safety,' *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1469–1482, 2017 (cit. on p. 10).

[10] C.-H. Lin, Y.-T. Chen, J.-J. Chen, W.-C. Shih and W.-T. Chen, 'Psafety: A collision prevention system for pedestrians using smartphone,' in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, IEEE, 2016, pp. 1–5 (cit. on p. 10).

[11] J. Hwang, Y. Ji, N. Kwak and E. Y. Kim, 'Outdoor context awareness device that enables mobile phone users to walk safely through urban intersections.,' in *ICPRAM*, 2016, pp. 526–533 (cit. on p. 10).

[12] Y. Li, F. Xue, X. Fan, Z. Qu and G. Zhou, 'Pedestrian walking safety system based on smartphone built-in sensors,' *IET Communications*, vol. 12, no. 6, pp. 751–758, 2018 (cit. on p. 11).

[13] Q. Wang, B. Guo, G. Peng, G. Zhou and Z. Yu, 'Crowdwatch: Pedestrian safety assistance with mobile crowd sensing,' in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, 2016, pp. 217–220 (cit. on p. 11).

[14] A. Shikishima, K. Nakamura and T. Wada, 'Detection of texting while walking by using smartphone's posture and acceleration information for safety of pedestrians,' in *2018 16th International Conference on Intelligent Transportation Systems Telecommunications (ITST)*, IEEE, 2018, pp. 1–6 (cit. on p. 11).

[15] M. Tang, C.-T. Nguyen, X. Wang and S. Lu, 'An efficient walking safety service for distracted mobile users,' in *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, IEEE, 2016, pp. 84–91 (cit. on p. 11).

[16] *A rapid learning algorithm for vehicle classification.* [Online]. Available: `https://www.sciencedirect.com/science/article/abs/pii/S0020025514010238` (cit. on p. 15).

Chapter 6
Appendix

Source_Code:

- **MainActivity.java**

```java
package com.example.abc;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.SwitchCompat;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.Context;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.os.Bundle;

import org.opencv.android.BaseLoaderCallback;
import org.opencv.android.CameraBridgeViewBase;
import org.opencv.android.JavaCameraView;
import org.opencv.android.LoaderCallbackInterface;
import org.opencv.android.OpenCVLoader;
import org.opencv.core.Mat;
import org.opencv.core.MatOfRect;
import org.opencv.core.Point;
import org.opencv.core.Rect;
import org.opencv.core.Scalar;
import org.opencv.imgproc.Imgproc;
import org.opencv.objdetect.CascadeClassifier;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;

import android.os.Vibrator;
import android.media.MediaPlayer;
import android.telephony.TelephonyManager;
import android.view.View;
import android.widget.CompoundButton;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity implements
    CameraBridgeViewBase.CvCameraViewListener2 {
    JavaCameraView javaCameraView;
    File cascFile;
    CascadeClassifier carDetector;
    private Mat mRgba,mGrey;

    private MediaPlayer mp;

    /// Saving switch widget state
    SwitchCompat switchButton;
    private static String MY_PREFS="switch_prefs"; //Shared prference
        name
    private static String SWITCH_STATUS = "switch_status";

    boolean switch_status;

    SharedPreferences myPreferences;
    SharedPreferences.Editor myEditor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        switchButton = findViewById(R.id.switch1);
        myPreferences = getSharedPreferences(MY_PREFS , MODE_PRIVATE);
        myEditor = getSharedPreferences(MY_PREFS , MODE_PRIVATE).edit();

        switch_status = myPreferences.getBoolean(SWITCH_STATUS,false);

        switchButton.setChecked(switch_status);

        switchButton.setOnCheckedChangeListener(new
            CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton,
                boolean b) {
                if(compoundButton.isChecked()){
                    myEditor.putBoolean(SWITCH_STATUS,true);
                    myEditor.apply();
                    switchButton.setChecked(true);
                }
                else{
                    myEditor.putBoolean(SWITCH_STATUS,false);
                    myEditor.apply();
                    switchButton.setChecked(false);
                }

            }
```

```java
    });
///
    if(ContextCompat.checkSelfPermission(this,
        Manifest.permission.READ_PHONE_STATE)
            != PackageManager.PERMISSION_GRANTED){
        ActivityCompat.requestPermissions(this,
                new
                    String[]{Manifest.permission.READ_PHONE_STATE},1);
    }

    javaCameraView = (JavaCameraView)findViewById(R.id.javaCamView);

    if(OpenCVLoader.initDebug())
    {
        OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_3_0_0,this,baseCallbac
    }
    else
    {
        try {
            baseCallback.onManagerConnected(LoaderCallbackInterface.SUCCESS);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    javaCameraView.setCvCameraViewListener(this);

    }

@Override
public void onCameraViewStarted(int width, int height) {
    mRgba = new Mat();
    mGrey = new Mat();
}

@Override
public void onCameraViewStopped() {
    mRgba.release();
    mGrey.release();
}

@Override
public Mat onCameraFrame(CameraBridgeViewBase.CvCameraViewFrame
    inputFrame) {
    mRgba = inputFrame.rgba();
    mGrey = inputFrame.gray();

    MatOfRect carDetections = new MatOfRect();
    carDetector.detectMultiScale(mRgba,carDetections);

    for(Rect rect:carDetections.toArray())
    {
```

```java
            Imgproc.rectangle(mRgba,new Point(rect.x,rect.y),
                    new Point(rect.x+rect.width,rect.y+rect.height),
                    new Scalar(255,0,0));
            // Get instance of Vibrator from current Context
            Vibrator v = (Vibrator)
                getSystemService(Context.VIBRATOR_SERVICE);
            v.vibrate(1200);
        }
        return null;
    }

    private BaseLoaderCallback baseCallback = new
        BaseLoaderCallback(this) {
        @Override
        public void onManagerConnected(int status) throws IOException {
            switch (status)
            {
                case LoaderCallbackInterface.SUCCESS:
                {
                    InputStream is =
                        getResources().openRawResource(R.raw.cascade);
                    File cascadeDir =
                        getDir("vehicle",Context.MODE_PRIVATE);
                    cascFile = new File(cascadeDir,"vehicle.xml");

                    FileOutputStream fos = new
                        FileOutputStream(cascFile);

                    byte[] buffer = new byte[4096];
                    int bytesRead;
                    while((bytesRead = is.read(buffer))!=-1)
                    {
                        fos.write(buffer,0,bytesRead);
                    }
                    is.close();
                    fos.close();

                    carDetector = new
                        CascadeClassifier(cascFile.getAbsolutePath());

                    if(carDetector.empty())
                    {
                         carDetector = null;
                    }
                    else
                        cascadeDir.delete();

                    javaCameraView.enableView();

                }
                break;
```

```
                default:
                    super.onManagerConnected(status);

            }
        }
    };

        @Override
    public void onResume() {
        super.onResume();
        sensorMan.registerListener(this, accelerometer,
                SensorManager.SENSOR_DELAY_UI);
    }

    @Override
    protected void onPause() {
        super.onPause();
        sensorMan.unregisterListener(this);
    }

      @Override
   public void onSensorChanged(SensorEvent event) {
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER){
            mGravity = event.values.clone();
            // Shake detection
            double x = mGravity[0];
            double y = mGravity[1];
            double z = mGravity[2];
            mAccelLast = mAccelCurrent;
            mAccelCurrent = Math.sqrt(x * x + y * y + z * z);;
            double delta = mAccelCurrent - mAccelLast;
            mAccel = mAccel * 0.9f + delta;

            if(mAccel > .53){
                Toast.makeText(MainActivity.this,"Walk
                    slowly!!",Toast.LENGTH_LONG).show();
            }

        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {
    }
}
```

- **CallReceiver.java**

```
package com.example.abc;
```

```java
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.telephony.TelephonyManager;
import android.view.Gravity;
import android.view.View;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

public class CallReceiver extends BroadcastReceiver {

    Switch aSwitch;

    @Override
    public void onReceive(Context context, Intent intent) {


        aSwitch = (Switch) aSwitch.findViewById(R.id.switch1);
        aSwitch.setOnCheckedChangeListener((buttonView, isChecked) -> {


             if(aSwitch.isChecked()) {
                 if
                     (((intent.getStringExtra(TelephonyManager.EXTRA_STATE).equals(Tele
                     ||
                         (intent.getStringExtra(TelephonyManager.EXTRA_STATE).equals(Te
                             {
                     Intent i = new Intent(context, MainActivity.class);
                         /// Offhook and ringing states
                     i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                     context.startActivity(i);
                 }
             }

        });
    }

     void showToast(Context context,String message){
        Toast toast=Toast.makeText(context,message,Toast.LENGTH_LONG);
        toast.setGravity(Gravity.CENTER,0,0);
        toast.show();
    }
}
```

- **AndroidManifest.xml**

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.abc">
```

```
    <uses-permission
        android:name="android.permission.CAMERA"></uses-permission>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"
        />
    <uses-permission android:name="android.permission.VIBRATE" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/safe"
        android:label="@string/app_name"
        android:roundIcon="@drawable/safe"
        android:supportsRtl="true"
        android:theme="@style/Theme.Abc">
        <activity android:name=".MainActivity"
            android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <receiver android:name=".CallReceiver"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.PHONE_STATE"
                    />
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

- **activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <org.opencv.android.JavaCameraView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/javaCamView"
        android:background="#FF018786"
        />
```

```
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:onClick="QuitApp"
        android:textAlignment="center"
        android:text="QUIT"

        android:textSize="20dp"
        android:textAppearance="@dimen/cardview_default_radius"
        android:textStyle="bold"
        />


    <androidx.appcompat.widget.SwitchCompat
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:thumb="@drawable/thumb"
        app:track="@drawable/track"
        android:textOn="ON"
        android:textOff="OFF"
        android:textStyle="bold"
        app:showText="true"
        android:layout_alignParentRight="true"
        android:id="@+id/switch1"
        android:checked="true"
        />
</RelativeLayout>
```

- **thumb.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
   <item android:state_checked="true">
       <shape android:shape="oval">
           <solid android:color="@android:color/white"/>
               <stroke android:width="1dp"
                   android:color="#34c759"/>
               <size android:width="45dp"
                     android:height="45dp"/>
       </shape>
   </item>


    <item android:state_checked="false">
        <shape android:shape="oval">
            <solid android:color="@android:color/white"/>
            <stroke android:width="1dp"
                android:color="#8c8c8c"/>
            <size android:width="40dp"
                android:height="40dp"/>
```

```
        </shape>
    </item>
</selector>
```

- **track.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_checked="true">
        <shape android:shape="rectangle">
            <solid android:color="#34c759"/>
            <corners android:radius="45dp"/>
            <stroke android:width="1dp"
                android:color="#8c8c8c"/>
            <size android:height="20dp"/>
        </shape>
    </item>


    <item android:state_checked="false">
        <shape android:shape="rectangle">
            <solid android:color="@android:color/white"/>
            <corners android:radius="45dp"/>
            <stroke android:width="1dp"
                android:color="#8c8c8c"/>
            <size android:height="20dp"/>
        </shape>
    </item>
</selector>
```