A PROJECT REPORT ON

SMART NAVIGATION FOR THE VISUALLY CHALLENGED

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

Of

SAVITRIBAI PHULE PUNE UNIVERSITY

By

SHUBHAM NAGARKAR	**B150234315**
MAYANK RAMNANI	**B150234350**
SIKANDER SINGH	**B150234384**
SAKSHI TIWARI	**B150234393**

Under the guidance of
Prof. T.P. VAIDYA

SINHGAD TECHNICAL EDUCATION SOCIETY -PUNE

तमसो मा ज्योतिर्गमय

Sinhgad Institutes

DEPARTMENT OF COMPUTER ENGINEERING

SINHGAD COLLEGE OF ENGINEERING, PUNE-41

Accredited by NAAC
2018-19

Sinhgad Technical Education Society,
Department of Computer Engineering
Sinhgad College of Engineering , Pune-41

Sinhgad Institutes

Date:

CERTIFICATE

This is to certify that the project report entitled

“Smart Navigation for the Visually Challenged ”

Submitted by

Shubham Nagarkar	**B150234315**
Mayank Ramnani	**B150234350**
Sikander Singh	**B150234384**
Sakshi Tiwari	**B150234393**

is a bonafide work carried out by him/her under the supervision of Prof. T.P. Vaidya and it is approved for the partial fulfillment of the requirements of Savitribai Phule Pune University, Pune for the award of the degree of Bachelor of Engineering (Computer Engineering) during the year 2018-19.

Prof. T.P. Vaidya
Internal Guide

Prof. M.P. Wankhade
Head
Department of Computer Engineering

Dr. S.D. Lokhande
Principal
Sinhgad college of Engineering

Acknowledgement

Working on this project is one of the wonderful and existing experiences in our lives. This report not only bears the testimony of existing efforts but also reflects the cooperation, help and guidance, which we received time to time. It is obvious but that we acknowledge them for their help without which the report would never have been completed.

We would like to express our deep gratitude to Prof. T.P. Vaidya for her profound guidance and support which helped me understanding the nuances of the project and make it work. We are thankful to her for her timely suggestions which helped us a lot in completion of this report.

Our heart felt thankful for our Head of Department Prof. M.P. Wankhade who has been a consistent motivator throughout the project.

We would also like to thank our review committee members - Prof. P.M. Kamde and Prof. S.H. Sheikh who have helped us all time in one way or the other.

Lastly, and the most important, we extend our thanking to our parents and friends who have constantly guided and motivated us to accomplish the task successfully.

Shubham Nagarkar
Mayank Ramnani
Sikander Singh
Sakshi Tiwari

Abstract

Visually impaired people are often unaware of dangers in front of them,even in familiar environments. Furthermore, in unfamiliar environments, such people require guidance to reduce the risk of colliding with obstacles. This study proposes a simple smartphone-based guiding system for solving the navigation problems for visually impaired people and achieving obstacle avoidance to enable visually impaired people to travel smoothly from a beginning point to a destination with greater awareness of their surroundings. For this purpose neither a specially designed smartphone is needed nor the internet connection.TensorFlow model is used for object detection which is very fast and works offline.It can detect upto 10 objects in a second. Distance of the object from the user is calculated using focal length of the phone camera.

Keywords: TensorFlow; Object detection; Visually Impaired; Machine learning; Smartphone-based; Guiding system; Audio format; Mobile device application

List of Figures

List of Tables

Abbreviations

IDE	Integrated Development Environment
EI	External Input
EO	External Output
EQ	External Inquiry
DFD	Data Flow Diagram
UFP	Unadjusted Function Point
FP	Function Point
VAF	Value Adjustment Factor
SLOC	Source Line of Code
YOLO	You Only Look Once
CUDA	Compute Unified Device Architecture
GPU	Graphics Processor Unit
AWS	Amazon Web Services

Contents

Chapter 1

INTRODUCTION

1.1 Background and Basics

According to a 2012 statistical report from the World Health Organization [1], approximately 285 million people worldwide are blind or have amblyopia, 246 million of whom have serious vision problems. People with amblyopia have less viewable areas than the normal population. Some people with amblyopia have distorted vision or can only sense light and shadows, without being able to discern objects in front of them. Moreover, blind people depend on their experiences or other senses to walk.

Visually impaired people usually have problems walking and avoiding obstacles in their daily lives. Traditionally, such people use guide canes to detect obstacles in front of them. Thus, visually impaired people cannot exactly know what types of obstacles are in front of them and must only depend on guide canes and experiences to walk safely and in the desired path. Furthermore, when in unfamiliar environments, visually impaired people often require assistance in the form of volunteers to guide them through the surrounding environment. Visually impaired people cannot entirely depend on a guide cane to become familiar with their surroundings or react quickly to unforeseen circumstances.

When encountering obstacles, visually impaired people must only rely on their experiences to react because they may not know what the obstacles are. However, some obstacles on the road cannot be predicted, such as a parked bicycle or a resting dog. For visually impaired people, reacting quickly to avoid such obstacles by relying solely on experiences is difcult. Therefore, such obstacles could engender danger for an unaccompanied visually impaired person.

The proposed system is not limited to specific indoor or outdoor environments.This system employs a smartphone to continually capture images of the environment in front of a user and perform image processing and object identication to inform the user of the image results. According to these results, the user can gain a more comprehensive understanding of the surroundings. This system enables visually impaired people to not only know the rough direction and distance to an obstacle, but also know what the obstacle is.

1.2 Literature Survey

1. **Paper-I**

   Title: Let Blind people see: Real time visual recognition with results converted to 3D audio.

   Abstract: This project tries to transform the visual world into the audio world with the potential to inform blind people objects as well as their spatial locations. Objects detected from the scene are represented by their names and converted to speech. Their spatial locations are encoded into the 2-channel audio with the help of 3D binaural sound simulation.

   - **Pros:** Uses computer vision and crowdsourcing to describe a picture captured by blind users in about 10 seconds.
   - **Cons:** Needs an expensive depth detecting camera to perform analysis.

2. **Paper-II**

   Title: YOLOv3: An Incremental Improvement

   Abstract: We present some updates to YOLO! We made a bunch of little design changes to make it better. We also trained this new network that's pretty swell. It's a little bigger than last time but more accurate. It's still fast though, don't worry. At 320×320 YOLOv3 runs in 22 ms at 28.2 mAP, as accurate as SSD but three times faster. When we look at the old .5 IOU mAP detection metric YOLOv3 is quite good. It achieves 57:9 AP50 in 51 ms on a Titan X, compared to 57:5 AP50 in 198 ms by RetinaNet, similar performance but $3.8\times$ faster. As always, all the code is online at https://pjreddie.com/yolo/

   - **Pros:** Faster than YOLOv1, YOLOv2.
   - **Cons:** Requires high processing power.

3. **Paper- III**

   Title: A Deep-learning-based Floor Detection System for the Visually Impaired

   Conference: 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress

Abstract: The American Foundation for the Blind (AFB) has recently reported that over 25 million people in the U.S. suffer from total or partial vision loss. As a result of their visual impairment, they are constantly affected by the risk of a fall and its consequences. Most of this affected population relies on assistive technologies that allow their integration to society. Fall prevention is an area of research that focuses on the improvement of people's lives through the use of pervasive computing. This work introduces a fall prevention system for the blind and its different modules and focuses on the first module: a deep-learning approach for floor detection. A combination of convolutional neural layers and fully connected layers is used to create a network topology capable of identifying floor areas in pictures of multiple indoor environments. This task is remarkably difficult due to the complexity of identifying the patterns of a floor area in different scenarios, the noise added by the movement of the camera while walking, and the real-time nature of the system. This paper provides a general description of the fall prevention system, and a detailed description of the floor detection system. Finally, the evaluation of the proposed floor detection approach is presented: an accuracy of 92.7% , a precision of 90.2% , and a recall of 90.5% were obtained.

- **Pros:** The proposed system uses single image analysis to estimate the location of the floor area.
- **Cons:** Long time required to train model to detect floors quickly.

1.3 Project Undertaken

1.3.1 Problem Definition

Design a system to assist visually challenged users in navigating through their surroundings using machine learning technologies.

1.3.2 Scope Statement

To develop a product for the benefit of the visually challenged population by aiding them in navigation of the environment. This product would enable the visually impaired population to walk freely in the environment without dependency on any other person. It focuses on "seeing with ears" to help the navigation.

1.4 Organization of Report

The report is organized in such a way that it starts with the introduction of Smart Navigation System including related work and project scope. The second chapter includes software requirement specifications which briefly describe the idea of what is to be done. It also includes project process modelling, cost and effort estimates and timeline. The report proceeds with IDEA Matrix, mathematical model, feasibility analysis and UML representations in the third chapter. The fourth chapter discusses the testing methods viz. unit testing, integration testing and acceptance testing. References are mentioned at the end of the report.

Chapter 2

PROJECT PLANNING AND MANAGEMENT

2.1 Detail System Requirement Specification (SRS)

2.1.1 System Overview

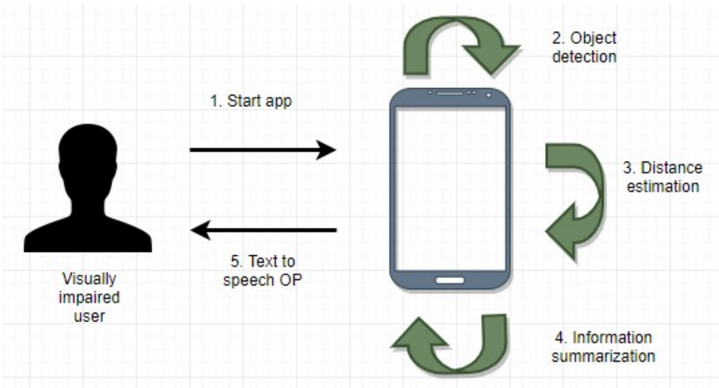

Figure 2.1: Formation of image on camera lens

2.1.2 Functional Requirements

The system should provide three main features:

- Object detection and recognition
- Calculating distance of object
- Calculating direction of object

System Feature 1: Object detection and recognition

Captured images are detected on real time basis by using Tensorflow lite model in the smartphone itself.It detects the objects with the help of bounding boxes and produces the results in text format.

System feature 2: Distance Evaluation

This feature will help in fall prevention. This module calculates distance by taking the focal length of the smartphone camera,its aperture into consideration.

System Feature 3: Direction Evaluation

To make the user aware of the surrounding, directions will be provided along with the obstacle type and distance.

2.1.3 Non-Functional Requirements

- Reliability: There is no or minimal human intervention in the system which will make it highly reliable.
- Availability: An android phone is widely available. Our android app needs to be pre-installed on devices by manufacturers for the visually impaired.
- Privacy: Removal of captured image data after analysis so as to maintain user privacy.
- Performance: Functioning should not be affected even if there are a lot of recognizable objects in the frame. It should still return results within the required time limit.

2.1.4 Deployment Environment

Hardware Requirements:

- Memory: 2GB RAM
- Storage:12 MB

Software Requirements:

- Android Studio

Platform Requirements:

- Java
- Android Version 4.3 above

2.1.5 External Interface Requirements

User Interfaces:

The user will have a basic smartphone app as the interface through which they can start or stop the software using a single click. Also, the user has the privilege to adjust the volume of sound according to his desire.

Hardware Interfaces:

No specific hardware interface required other than network interface.

Software Interfaces:

Ionic mobile app framework to build the mobile app.

Communication Interfaces:

No specific communication interface is required.

2.2 Project Process Modelling

Waterfall model is being used for the development of this project because:

-Requirements are very well documented, clear and fixed.

-Product definition is stable.

-Technology is understood and is not dynamic.

-Clearly defined stages completed one at a time.

The steps involved in the Waterfall Process Model are:

- Requirement Analysis: All the software and hardware requirements are recognized and gathered at this stage.
- Design: The model is actually designed.
- Implementation: The model is programmed at this stage
- Verification: Testing of the program is carried out at this stage
- Maintenance: If any bug is found, they are corrected.

- Deployment: The software is deployed for use.

2.3 Cost & Efforts Estimates

It begins with the decomposition of a project or application into its data and transactional functions. The data functions represent the functionality provided to the user by attending to their internal and external requirements in relation to the data, whereas the transactional functions describe the functionality provided to the user in relation to the processing this data by the application.

1. Internal Logical File (ILF)
2. External Interface File (EIF)

The transactional functions are:

1. External Input (EI)
2. External Output (EO)
3. External Inquiry (EQ)

Analysis of the system as presented in the DFD summarizes the number of various components:

EI = 2

EO = 4

EQ = 1

ILF = 5

ELF = 2 The degree of complexity (simple, average or complex) was evaluated for each component.

From above table we get values of data functions and transaction functions Calculation of Unadjusted Function Point (UFP)

$$UFP = 0 < i < 50 < j < 3\, Counti, j * Weighti, j$$

Record Element Types	Data Element Type		
	1-5	6-15	>15
0-1	Low	Low	Average
0-3	Low	Average	High
>3	Average	High	High

Table 2.1: Degree of Complexity for EO

Record Element Types	Data Element Type		
	1-5	6-15	>15
0-1	Low	Low	Average
2-5	Low	Average	High
>5	Average	High	High

Table 2.2: Degree of Complexity for ILF

Rating	Values				
	EO	EQ	EI	ILF	ELF
Low	4	3	3	7	5
Average	5	4	4	10	7

Table 2.3: Predefined Weights

$$
\begin{aligned}
UFP &= (2 * weightofEI) + (4 * weightofEO) && (2.1)\\
&+ (1 * weightofEQ) + (5 * weightofILF) + (2 * weightofELF) && (2.2)\\
&= (2 * 4) + (4 * 3) + (5 * 7) + (2 * 5) && (2.3)\\
&= 87 && (2.4)
\end{aligned}
$$

Calculation of Final Function Point using formula:

$$FinalFP = UFP * (0.65 + (VAF * 0.01))$$

where VAF is the Value Adjustment Factor

Total VAF = 37

Therefore,

FP = 69.36

Effort calculation using lines of source code (SLOC)

Sr. no	VAFs	Grade
1.	Data Communications	5
2.	Distributed Data Processing	0
3.	Processing	4
4.	Heavily Utilized Configuration	3
5.	Transaction Rate	5
6.	Online Data Entry	0
7.	End-user Efficiency	4
8.	Online Update	0
9.	Complex Processing	1
10.	Reusability	4
11.	Installation Ease	4
12.	Operational Ease	4
13.	Multiple sites/org	0
14.	Facilitate change	3

Table 2.4: Value Adjustment Factor

$$Effort = FP * SLOC(as\ per\ language) \tag{2.5}$$
$$= 69.36 * 46 \tag{2.6}$$
$$= 3190.56 hours \tag{2.7}$$
$$= 5 months \tag{2.8}$$

2.4 Project Scheduling

2.4.1 Time Line Chart

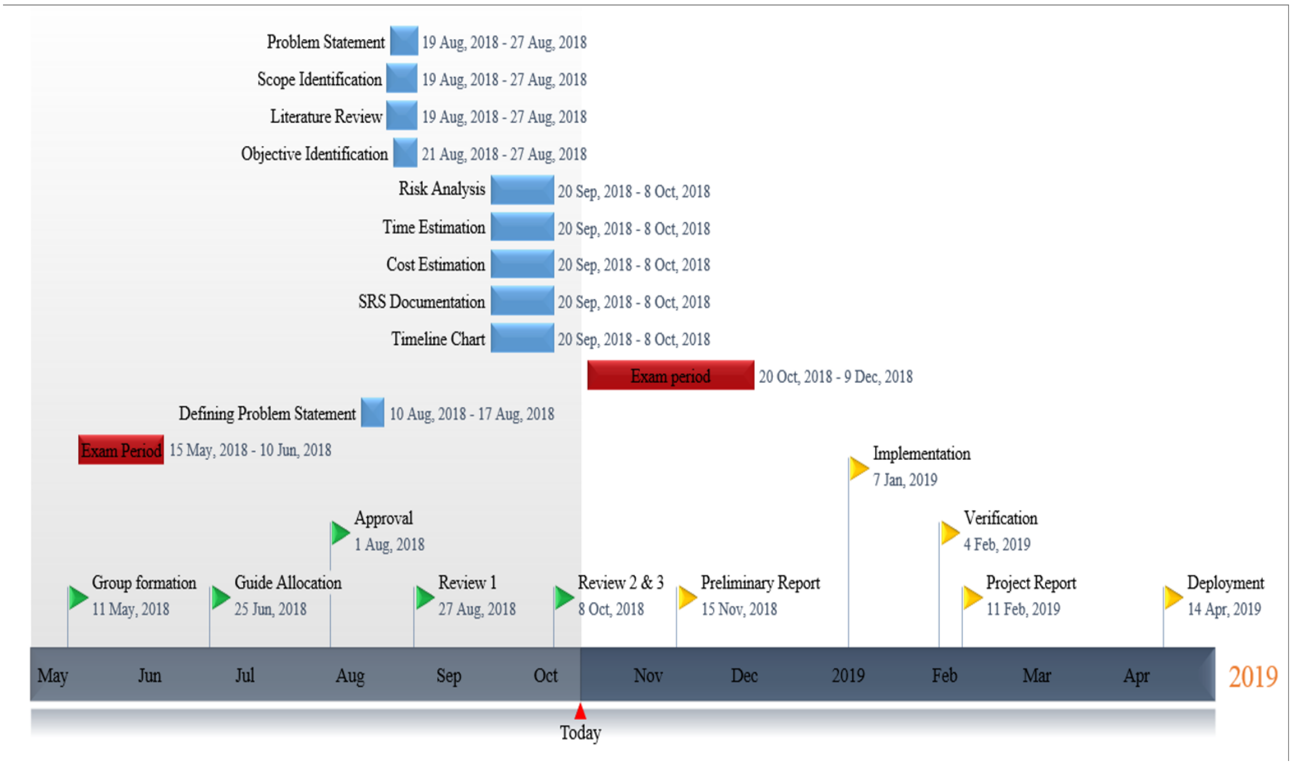

Figure 2.2: Time Line Chart (June-18 - October-18)

Chapter 3

ANALYSIS & DESIGN

3.1 IDEA Matrix

I	**Increase**	Increasing the usability and accessibility of the system
	Improve	Improves navigation experience of the visually impaired
	Integrate	Integrates Object detection, Distance estimation and Audio output
D	**Deduce**	Obstacles are deduced by the System
	Deliver	Delivers safe navigation pathway to the user
	Decrease	Decrease collision of user with the environmental objects
	Educate	Educates visually impaired users about the environmental setup
	Evaluate	Evaluates the position of the objects around the user
	Eliminate	Eliminates the risk of accidents
A	**Accelerate**	Accelerates the process of finding objects and navigation
	Associate	The system associates object detection with audio output of spatial description
	Avoid	Avoids the collision of user with the obstacles

Table 3.1: Idea Matrix

3.2 Mathematical Model

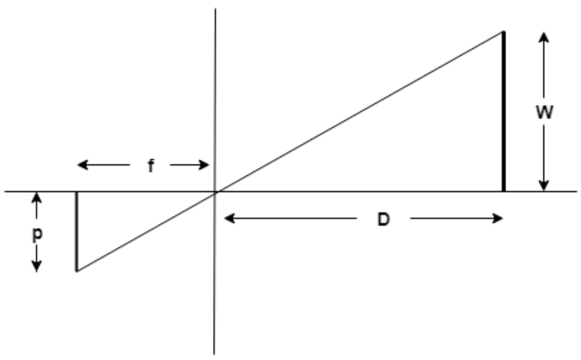

Figure 3.1: Formation of image on camera lens

Thus from above triangle similarity, we get the formula:

$$D = f * W/p$$

where:
D is the distance in meters
W is the width of object in meters
f is the focal length in pixels of camera
p is the width in pixels of the object in image

Pixel width of the object can be obtained by:

$$f = (F/SW) * iw$$

where:
f is the focal length in pixels
F is the focal length in mm, property of the camera
SW is the sensor width in mm, property of the camera
iw is the captured image width in pixels

3.3 Feasibility Analysis

The amount of time required for the model to process images and detect objects can be improved by using high end graphics processor. The depth estimation analysis can be done within milliseconds. With a strong internet connection, sending and receiving speed of data can be improved subsequently. System time can be improved further with high performing CPUs and GPUs.

3.4 UML Diagrams

3.4.1 Use Case Diagram

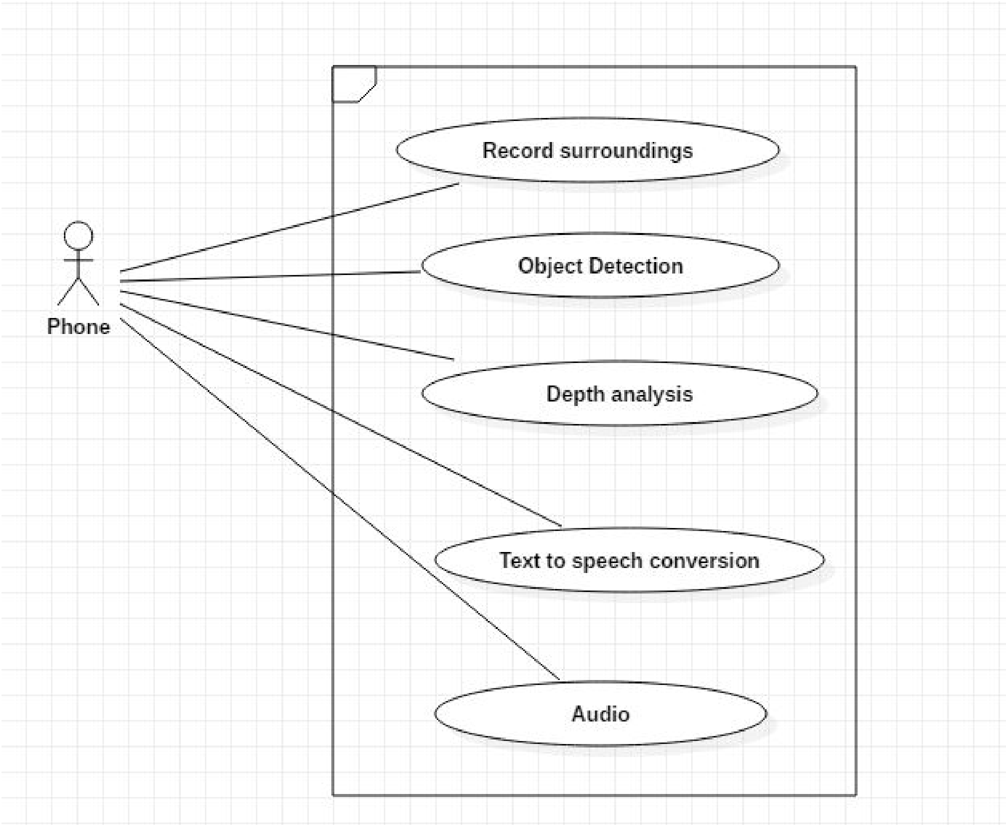

Figure 3.2: Use Case Diagram

3.4.2 Class Diagram

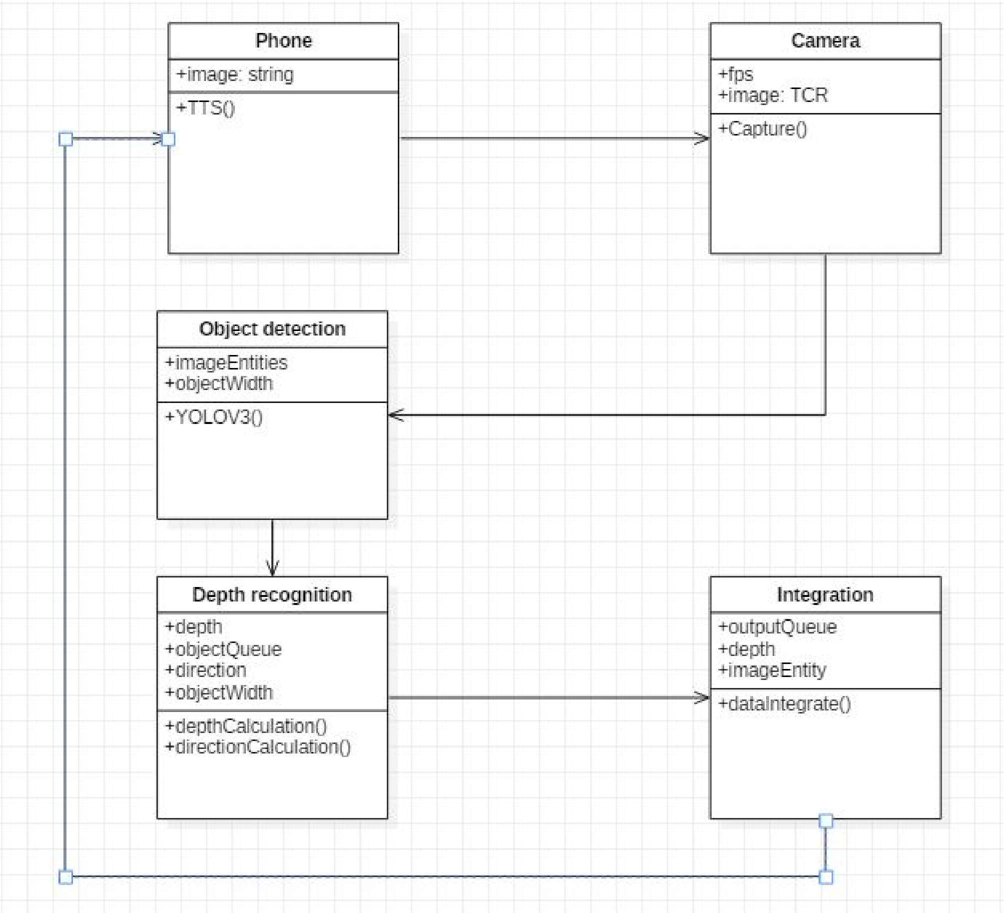

Figure 3.3: Class Diagram

3.4.3 Sequence Diagram

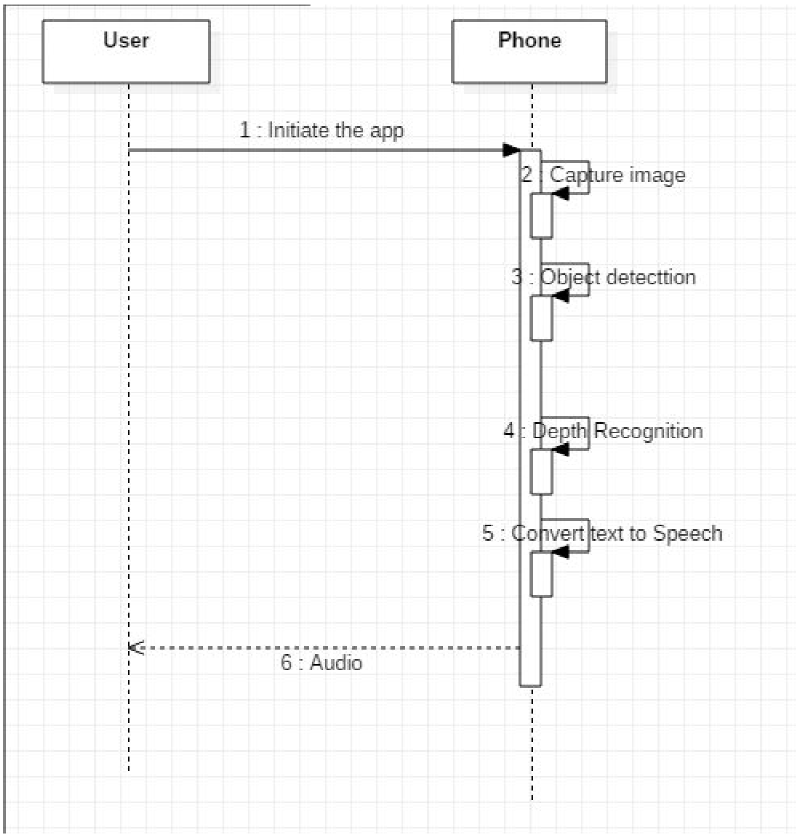

Figure 3.4: Sequence Diagram

3.4.4 Deployment Diagram

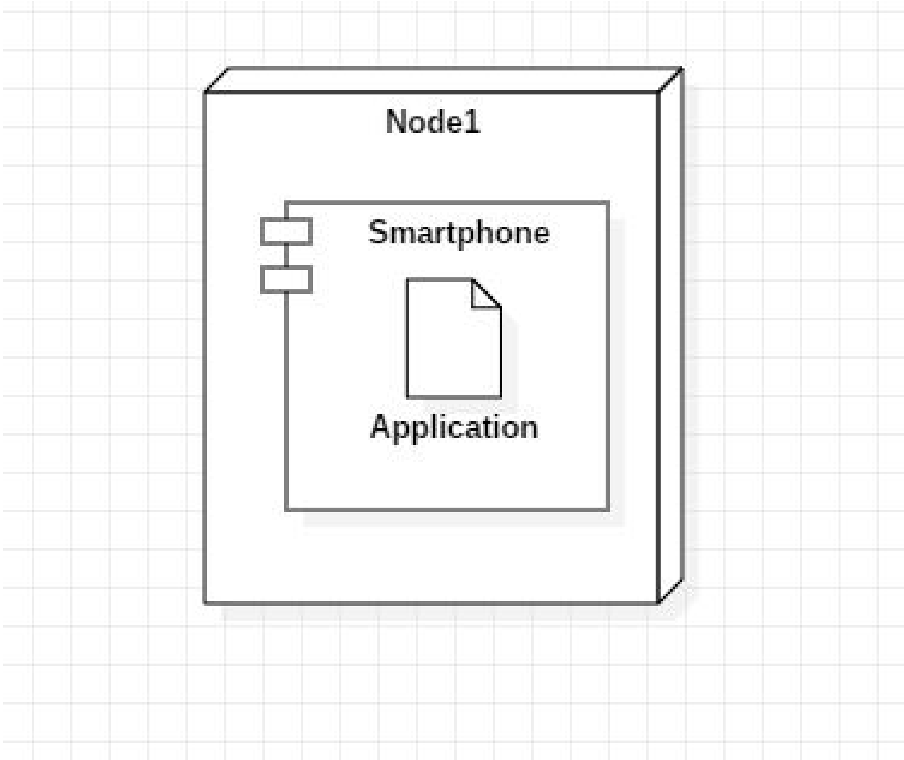

Figure 3.5: Deployment Diagram

Chapter 4

IMPLEMENTATION AND CODING

4.1 Introduction

The proposed system detects all the objects in front of the user and notifies about the ones which are in a radius of 2.5 meters from the user. After a specific interval of time, the system generates alerts about the type of object, the distance of the object from the user and the direction of the detected object. These alerts are in the audio format which helps the user to understand the spatial location of the objects in his surroundings. It will restrict the number of accidents i.e the collisions of the user with the surrounding objects, substantially. As a result, the system will act as a guiding tool for the visually impaired person to navigate in the environment freely.

4.2 Database Schema (if applicable)

The database used for Object detection consists of object names and their corresponding heights.When the image of an object is captured by the phone camera this database is referred by the system to retrieve the object name. Tensorflow model has its inbuilt data set which it uses for object detection.

4.3 Operational Details

- **Object Detection:** This module detects the object on the basis of the bounding boxes. TensorFlow model is used for object detection.For each detected object, the model will return an array of four numbers representing a bounding rectangle that surrounds its position.The object detection model can identify and locate up to 10 objects in an image. It is trained to recognize 80 classes of objects.
- **Distance Estimation:** This module gives the distance of an object from the person.It gives a rough estimate to the user about the spatial position of the object so that he is aware of it beforehand.

4.4 Code Listing

4.4.1 Code snippet for Distance estimation

```
package org.tensorflow.lite.examples.detection;
import android.util.Log;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.HashMap;
import java.util.Scanner;
import java.util.*;

public class Distance_estimation {
    static HashMap<String, Integer> hmap = new HashMap<String,
Integer>();
    static HashMap<String, Integer> hmap2 = new HashMap<String,
Integer>();

    public static void create_hmap()throws IOException
    {
        hmap2.put("dog",40);
        hmap2.put("person",160);
        hmap2.put("bicycle",50);
        hmap2.put("truck",200);
        hmap2.put("chair",70);
        hmap2.put("sofa",70);
        hmap2.put("couch",70);
        hmap2.put("table",50);
        hmap2.put("bed",55);
        hmap2.put("oven",30);
        hmap2.put("suitcase",50);
```

```
            hmap2.put("car",150);
            hmap2.put("motorcycle",70);
            hmap2.put("airplane",700);
            hmap2.put("bus",290);
            hmap2.put("train",250);
            hmap2.put("boat",200);
            hmap2.put("traffic light",40);
            hmap2.put("fire hydrant",38);
            hmap2.put("stop sign",25);
            hmap2.put("backpack",40);
            hmap2.put("umbrella",30);
            hmap2.put("handbag",20);
            hmap2.put("tennis racket",40);
            hmap2.put("bottle",18);
            hmap2.put("wine glass", 17);
            hmap2.put("cup",7);
            hmap2.put("fork",10);
            hmap2.put("knife",10);
            hmap2.put("spoon",10);
            hmap2.put("bowl",7);

            hmap2.put("apple",6);
            hmap2.put("banana",6);
            hmap2.put("orange",6);
            hmap2.put("pizza",2);
            hmap2.put("donut",2);
            hmap2.put("cake",6);
            hmap2.put("potted plant", 30);
            hmap2.put("dining table",60);
            hmap2.put("toilet",36);



        hmap2.put("toilet",36);
        hmap2.put("tv",59);
        hmap2.put("keyboard",10);


        hmap2.put("cell phone",12);
        hmap2.put("microwave",40);
        hmap2.put("sink",30);
        hmap2.put("refridgerator",300);
        hmap2.put("book",20);
        hmap2.put("clock",29);
        hmap2.put("vase",30);
        hmap2.put("teddy bear",30);
        hmap2.put("toothbrush",10);
        hmap2.put("hair dryer",19);
    }


    public static int calculate_distance(float
focal_length,float sensor_height,String obj_name ,int
obj_pix_height,int obj_actual_height) {
        int pixel_FL;
        pixel_FL = (int)((480*focal_length)/sensor_height);
        int distance = ((pixel_FL )* obj_actual_height) /
obj_pix_height;
        hmap.put(obj_name, distance);
        return distance;
    }
```

```
    public static String get_direction(String obj_name, int
width_obj, int left_pix)
    {
        int val = 100;
        int p1=val;
        String direc="";
        int p2=200;
        int mid=((width_obj)/2)+left_pix;

        Log.d("mytag","width is : "+width_obj+" mid is : "+mid);
        if(mid>p1 && mid<p2)
            direc="Front";
        else if(mid<p1)
            direc="Left";
        else
            direc="Right";

        Log.d("mytag","Object: "+obj_name+" Direction :
"+direc);
        return direc;
    }

}
```

Chapter 5

TESTING

5.1 Unit Testing

Unit testing is the first level of software testing where individual units are tested by the developers themselves. Here we have three main modules or units:

1. Object Detection Module
2. Distance Calculation Module
3. Audio Generation Module

Test ID	Test Case	Input	Expected O/P
1	Object detection module, images at the rate of 5 frames per second	Image of Sofa	sofa detected
2		Image of person	person detected
3		Image of a ball	ball detected
4	Distance between user and the obstacle detected	A ball at a distance of 1 meter	1 meters
5		Sofa at a distance of 0.5 meters	0.5 meters
6		Chair at a distance of 3 meters	3 meters
7	Audio generation of the analyzed data	Chair is to the left of user	left
8		Table is to the right	right
9		Ball is in the front	front

Table 5.1: Test Cases Unit Testing

In unit testing we will test all the modules individually.

5.2 Integration Testing

Integration testing is the second level of testing, here first two modules which are distance calculation and object detection will be integrated and tested. The test cases for it are as follows:

Test ID	Test Case	Expected O/P	Actual O/P
1	Inputs: Object detected, Distance calculated Output: Proper object with correct distance is detected or not	Chair at a distance of 1 meter	Table at a distance of 1 meter (fail)
2		Table at a distance of 4 meters	Table at a distance of 4 meters (pass)
3		Window at a distance of 2 meters	Window at a distance of 4 meters (fail)

Table 5.2: Test Cases Integration Testing

5.3 Acceptance Testing

Acceptance testing is the last phase of software testing, it will be performed by the development team which is known as alpha testing and then it will also be performed by the project guide which can be termed as beta testing. The main issues that will be tested in acceptance testing are:

- The system gives proper alerts about the obstacles in the way.
- The system provides alerts in a reasonable time i.e. the alerts should be generated at least 1 second before the collision.
- The alerts generated by the system should be in a user-friendly manner such that the visually challenged person is able to understand the direction of the obstacle easily and be able to avoid it and smoothly walk in the surroundings.

Chapter 6

RESULTS AND DISCUSSION

6.1 Main GUI Snapshots

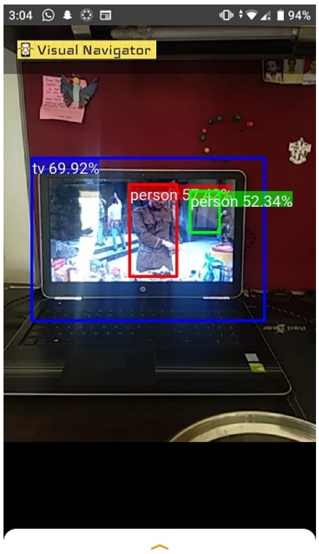

Figure 6.1: GUI Screenshot 1

Figure 6.2: GUI Screenshot 2

6.2 Discussions

The Tensorflow model used for object detection is quite fast and accurate. It detects maximum 10 objects in one frame. Out of those 10 detected objects, the system alerts about the nearest one. The data-set used contains average heights of objects and not the accurate ones, hence the distance which is estimated has an error tolerance of about 0.5 -1 meters. As the objects comes closer to the object the accuracy of the distance increases and the errors decrease. This accuracy can be further increased with the use of Stereo cameras enabled phones, which calculate distance accurately.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This system can not only inform the user of the type of obstacle in front of him or her, but also reveal the approximate distance between the user and the object. Compared with the more traditional guidance method involving a modified guide cane, the proposed system provides more information for the user. As traditional guiding systems are only applicable under specific circumstances, the usage of our system is not limited. The proposed system provides more information than does a conventional guiding system, in addition to being applicable to different environments. Thus, specific areas and special conditions are not necessary.

7.2 Future Work

In the future, to provide information on more types of obstacles and more accurate recognition, a broader range of obstacle images and a stereo camera equipped with a more powerful processing unit could be used to increase the number of recognition categories, the recognition rate and accuracy. On the other hand, the feature recognition has extensibility. More recognition Sensors 2017, 17, 1371 21 of 22 services of other object types can be added when the computing ability of smartphone improves. These improvements could provide more information without needing wireless connections.

References

[1] R. Jiang, Q. Lin and S. Qu *”Let Blind People See: Real time visual recognition with results converted to 3D audio”* Stanford 2016

[2] *”YOLOv3: An Incremental Improvement”* Joseph Redmon, Ali Farhadi, University of Washington.

[3] *“A Deep-learning-based Floor Detection System for the Visually Impaired”*, Yeung Delahoz, Miguel A. Labrador, University of South Florida.

[4] *“Seeing without sight – An automatic cognition system dedicated to blind and visually impaired people”* 2017 IEEE International Conference on Computer Vision Workshops.

[5] https://answers.opencv.org/question/17076/conversion-focal-distance-from-mm-to-pixels/