A Project Report on

Voice Based Smart Assistive Device for the Visually Challenged

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Technology

by

Sameer Dev(16104030)
Sudama Jaiswal(16104015)
Yogendra Kokamkar(16104016)

Under the Guidance of

Prof. Kiran Deshpande
Prof. Kaushiki Upadhyaya

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road, Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2019-2020

Approval Sheet

This Project Report entitled ***"Voice Based Smart Assistive Device for the Visually Challenged "*** Submitted by ***Sameer Dev(16104030), Sudama Jaiswal(16104015), Yogendra Kokamkar(16104016)*** is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

(Prof. Kaushiki Upadhyaya)
Co-Guide

(Prof. Kiran Deshpande)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Voice Based Smart Assistive Device for the Visually Challenged "*** Submitted by ***Sameer Dev(16104030), Sudama Jaiswal(16104015), Yogendra Kokamkar(16104016)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2019-2020.

(Prof. Kaushiki Upadhyaya)
Co-Guide

(Prof. Kiran Deshpande)
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(Sameer Dev, 16104030)
(Sudama Jaiswal, 16104015)
(Yogendra Kokamkar, 16104016)

Date:

Abstract

Blind, the dictionary defines it in one simple word, sightless. The life for a visually challenged person is extremely hard for obvious reasons. In this era of cutting-edge technology, it is still extremely difficult for visually challenged people to carry out day to day chores or enjoy the simple pleasures of life such as going for a walk, socializing, and so on. Hence, developing new solutions that allow those individuals to interact with sighted people, and the sighted world, in a way that lessens any of the problems that can arise from being visually impaired is becoming increasingly important. This paper presents a Smart Device built using a Raspberry Pi that can be controlled via Voice Commands and carry out various tasks such as Object Detection, Navigation, and notify the user through Audio feedback. The device will also take help of Image Recognition and Image Processing in order to convey information about specific places to the user as soon as the user is in that particular vicinity, hence allowing the person using the device know their surrounding environment in a better way.

Contents

1 Introduction **1**

2 Literature Review **3**

3 Problem Statement **5**

4 Objectives **6**

5 Project Design **7**
5.1 Proposed System 7
5.2 Flow of System 10

6 UML Diagrams **12**
6.1 Use Case Diagram 12
6.2 Activity Diagram 13

7 Proposed Technology Stack **14**

8 Project Implementation **15**
8.1 Client 15
8.2 Server 16

9 Testing **17**
9.1 Unit Testing 17
9.2 Integration Testing 17
9.3 System Testing 18
9.4 Acceptance Testing 18

10 Result **19**
10.1 Object Detection 19
10.2 Environment Description 19
10.2.1 Correct Results 20
10.2.2 Partially Correct Results 20
10.2.3 Incorrect Results 20

11 Conclusion and Future Scope **21**

Bibliography **22**

Publication **26**

List of Figures

5.1 Block Diagram of the Proposed System 7
5.2 Object Detection using the Camera module 9
5.3 Environment Description Model in action 9
5.4 Flowchart of the proposed system 10

6.1 Use Case Diagram 12
6.2 Activity Diagram 13

7.1 Technology Stack 14

8.1 Client Declaration 15
8.2 Trigger Conditions 15
8.3 Server Declaration 16
8.4 Training Data for Environment Description Model 16

10.1 A few correct results given by the model 20
10.2 Examples of partially correct results given by the mode 20
10.3 Examples of incorrect results given by the model 20

List of Tables

10.1 Confusion Matrix of the Object Detection Module 19
10.2 BLEU Scores for the Description Model 19

List of Abbreviations

CNN:	Convolutional Neural Network
ConvNet:	Convolutional Neural Network
API:	Application Program Interface
REST:	Representational State Transfer
RFID:	Radio-Frequency Identification
GPS:	Global Positioning System

Chapter 1

Introduction

According to the October 2018 article of World Health Organization (WHO) globally, it is estimated that approximately 1.3 billion people live with some form of distance or near vision impairment. With regards to distance vision, 188.5 million have a mild vision impairment, 217 million have moderate to severe vision impairment, and 36 million people are blind. With regards to near vision, 826 million people live with near vision impairment [12].

Population growth and aging will increase the risk that more and more people acquire vision impairment in the near future. In this high-tech era, technology has made it possible for everyone to live a comfortable life. From entertainment to sports, studies to arts, technology has had a huge impact on our lives which has certainly made everybody's life enjoyable as well as easy. And yet, even after so many technological advancements and breakthroughs physically challenged people still need to depend upon other people for day to day functioning which ultimately makes them less confident in unfamiliar settings.

There are many definitions of Assistive Technology in common use, they range from formal technical definitions maintained by organizations such as the WHO to informal definitions often popularized by users themselves. As outlined by Wikipedia, Assistive Technology is an umbrella term that includes assistive, adaptive, and rehabilitative devices for people with disabilities or the elderly population while also including the process used in choosing, identifying, and using them [13].

A growing variety of special devices are readily available for use by visually impaired people. They vary in cost from solely a couple of Hundred Rupees to Thousands of Rupees for a single device. Hardly ever does a man-made device, by itself, make the difference in whether or not a visually impaired person can do a job. Devices do, however, provide added independence and adaptability to visually impaired persons in numerous positions.

Visually impaired people have limited scope of reading and understanding text and images; hence a Voice interface will be a very important medium of communication for them. With the help of the voice interface, they will be able to get various kinds of information from the device. This interface will be enabled by the increasingly available Speech Recognition as well as Speech-to-Text APIs which allow a person to extract relevant features from audio signals.

With the advent of computing power and the humongous amount of data being generated every day, Deep Learning has come into much prominence in the past few years giving rise to powerful models. This, in turn, has made way for Transfer Learning which allows people to use the same models, with minor readjustments, for other purposes thus reducing their overhead of having to train a model from scratch. A pre-trained image classifier, coupled with new domain-specific data can do wonders while prediction new unseen images hence enabling Object Detection at amazing speeds.

This combination of Voice Recognition, Speech-to-Text and Object Detection when coupled into a single device and optimized in such a way so as to provide real-time analysis, as well as feedback, can benefit the needy in a plethora of ways. The device will act as a medium of communication and assistance, connecting visually impaired people to the outside world.

Chapter 2

Literature Review

Yeong-Hwa Chang et al. [1] proposed an intelligent walking stick for visually challenged people. The stick was mounted with a Raspberry Pi that had multiple modules such as an ultrasonic sensor, a water sensor, a vibration motor. It also contained a GPS in order to track the location of the user. An additional Programmable Interrupt Controller (PIC) was attached to the Raspberry Pi to increase response speed and decrease computational complexity. A vibration motor was used to vibrate the device gently, in an event where the ultrasonic sensor detected an object and the water sensor was used to detect small puddles or moist surfaces that may come in the walking path of the person.

Ashwini B Yadav et al. [2] presented a cheap, userfriendly smart stick to improve the mobility of visually impaired folks in a specific area. The device was able to help a person navigate, which was possible due to the use of RFID technology, with the help of voice output. The advantage of using RFID was that it was also possible to get the location of the person very easily. A push button was also built which helped a person locate the stick.

D. Munteanu and R. Ionel [3] came up with an experimental micro-controller-based device which had an audio as well as a haptic feedback option that the user could select from. The device could be controlled by an application running on a Smartphone. The device could also be controlled using predefined voice commands. The application connected to the device using Bluetooth connectivity. Distance measurements were performed using ultrasonic echolocation.

Rohit Agarwal et al. [4] proposed a device which included a pair of glasses and an ultrasonic sensor fitted on the center of the glasses, along with a processing unit and a beeping component. The ultrasonic module detects objects in front of the user and on detecting one it sends the information to the control unit which in turn sets off the buzzer, thus alerting the user. It was a lightweight, cheap, easy to use and portable device.

Chien-Nan Lee et al. [5] designed Lazy Susan, a device aimed at increasing the dining conveniences and safety of visually impaired people. The device included electric motors and gears, a voice integrated circuit module, an RFID module, an Android application, a speaker, and a set of buttons with Braille. The addition of Braille helped a person understand the functions of the buttons. On pressing a button, the device brings the corresponding dish in front of the user. The information about the various dishes is conveyed to the user through the speaker.

Kasthuri R. et al. [6] developed a smart android application to guide visually impaired people. The application helped the user open any app on the phone, call any contact and so on, all with the help of Speech Recognition. Users could control the mobile device through voice commands. The app also had a Selendroid interface which enabled the person to fetch the latest bits of information from various internet based web-servers. The fetched information included live weather, news updates as well as transport related queries.

Chapter 3

Problem Statement

Almost all of the works of literature mentioned have a fundamental requirement or dependency, i.e. every smart device that has been proposed needs to be connected to an android application for proper functioning. This gives rises to the question as to what should a visually impaired person do if he/she has no access to a smartphone to the run required application? Should they suffer for not having a smartphone?

In this paper we propose a novel device that is standalone in nature and can function without having to connect to an Android application. Although there will be an Android app for the device to connect with, it will not be a required dependency as is the case with most smart devices. The main purpose of the app will be to change certain system settings, debug the device and view system generated logs and stats.

The proposed assistive device for visually impaired people will let them get a better sense of the surroundings and environments of our college and similar premises, it will also give them audio feedback regarding the obstacles that come in their path. The device can also classify nearby objects, thanks to the camera module and the power of Deep Learning, which will help give the user a basic idea about the objects surrounding him/her. The device will also be fully operable with the help of voice commands that will be enabled thanks to the various Text-to-Speech and Speech-to-Text APIs available for development purposes.

Chapter 4

Objectives

A. Overcome Environmental Challenges
Physical movement is one of the biggest challenges for visually impaired people, our device can help them get an overview of the surroundings so they can be more confident about the path and surroundings they are travelling in

B. Overcome Social Challenges
Blindness can cause significant social challenges, typically because there are activities in which blind people can't easily participate. Our device can assist the visually impaired in regards to other people, things, animals and surroundings by which the he/she can decide a way of social interaction

C. Overcome Technological Challenges
The small touch screens of many tablets and smartphones may be particularly difficulty for the visually impaired, hence our device can be fully controlled by voice or by an app with the subtle and simple user interface, overcoming the technology usage barrier

Chapter 5

Project Design

5.1 Proposed System

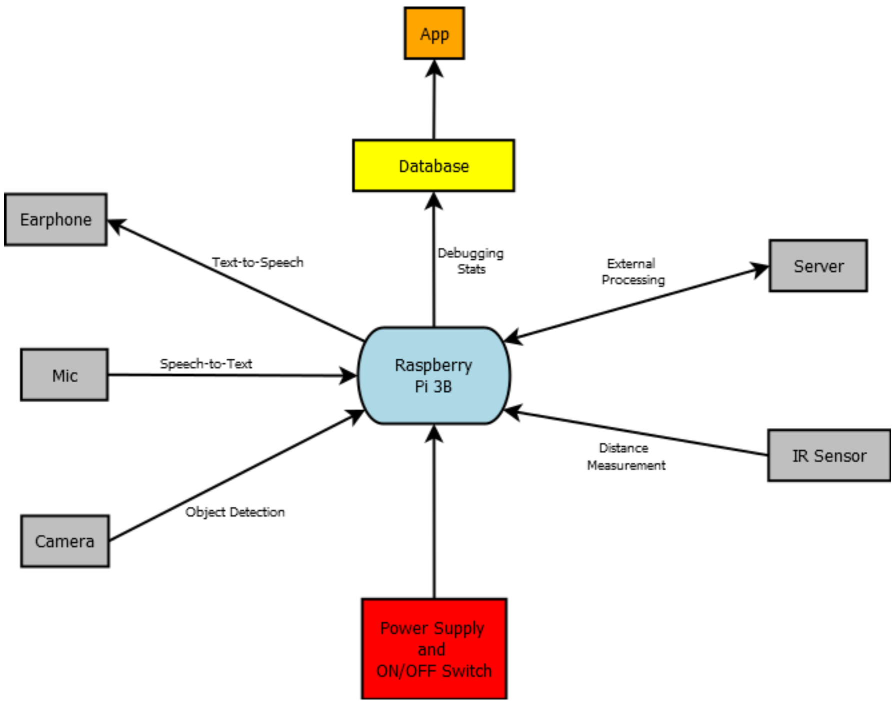

Figure 5.1: Block Diagram of the Proposed System

The proposed device will help visually impaired people get a better sense of their surroundings and overcome the various challenges they face daily for doing the simplest of tasks. The device is portable and the purpose of its usage is to warn the user when objects are present in the walking path so a collision can be avoided and also guide the user in the right directions while walking.

Since the Raspberry Pi has limited processing power, running complex Deep Learning models on it can be very time consuming. Hence, in order to tackle this, we use a separate server for processing images and running our models. For any task that is compute intensive and could take a lot of time, we send requests to the server using REST APIs. This additional step is necessary because if the response of the device to stimuli and changes is not real-time then the user may face difficulties while performing tasks. Since the device and the server will be on the same network, the task of establishing communication between them becomes easy.

Distance measurements, between the user and possible obstacles, are done using an IR Sensor. This module is chosen instead of an ultrasonic sensor because its maximum range is considerably higher than that of an ultrasonic sensor. Object Detection and Image Classification are performed using Deep Learning models based on Convolution Neural Networks (CNN) implemented via OpenCV and the data provided by the modules is processed by the Raspberry Pi, which also handles the audio feedback part.

In practice, very few people train an entire Convolutional Network or ConvNet from scratch (with random initialization), because it is relatively rare to have a dataset of sufficient size. Instead, it is common to pretrain a ConvNet on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then use the ConvNet either as an initialization or a fixed feature extractor for the task of interest. [14]

This technique is known as Transfer Learning wherein we take a pre-trained, state-of-the-art image classifier such as VGG16, InceptionV3, ResNet50, DenseNet, NasNet, etc. and instead of training the whole model we only train the fully connected layers with our data while keeping the weights of the previous layers unchanged. This technique, while being faster, also proves to be more accurate than creating a CNN from scratch.

Delving a little bit more into the topic of transfer learning, the three major transfer learning scenarios look as follows:

1) ConvNet as fixed feature extractor: Take a ConvNet pretrained on ImageNet, remove the last fully-connected layer and then treat the rest of the ConvNet as a fixed feature extractor for the new dataset

2) Fine-tuning the ConvNet: The second strategy is to not only replace and retrain the classifier on top of the ConvNet on the new dataset, but to also fine-tune the weights of the pre-trained network by continuing the back-propagation

3) Pre-trained models: Since modern ConvNets take 2-3 weeks to train across multiple GPUs on ImageNet, it is common to see people release their final ConvNet checkpoints for the benefit of others who can use these networks for fine-tuning

Using OpenCV we can perform a frame by frame, real-time video analysis with the help of which we will be able to detect objects in front of us. For illustration purposes, in Fig. 2 we have shown bounding boxes along with the class/category to which an object belongs. Also included is a numerical value. This value is the confidence with which our model is able to identify an object in an image.

Once an object has been detected the device will then calculate the distance between the object and the user using the IR Sensor and the information will be conveyed to the user via the audio feedback system. In case multiple objects are detected, the information about the nearest object will be conveyed first followed by information about the rest of the

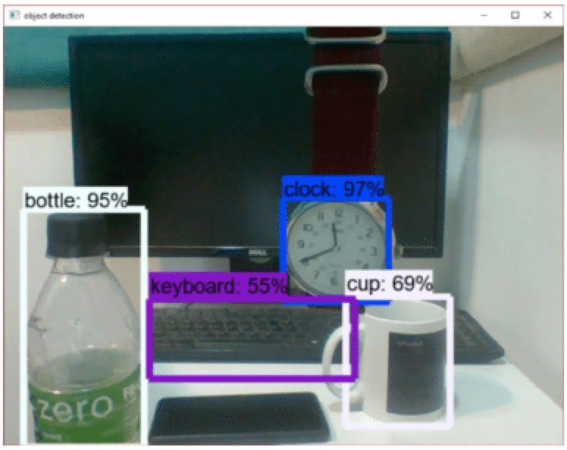

Figure 5.2: Object Detection using the Camera module

objects. The program will also analyze the direction in which the user must go in order to avoid the obstacle. The information about the directions to be taken will be conveyed via audio feedback.

Another feature of this device would be its ability to describe the surrounding environment at the click of a button. By using yet another Deep Neural Network and training it on the Flickr8K dataset [15] we can allow this device to generate sentences that can describe where the user is and what is going on around him. This model is based on the PhD thesis of Andrej Karpathy [7].

Since we are using OpenCV, we can also perform minor Image Processing activities in

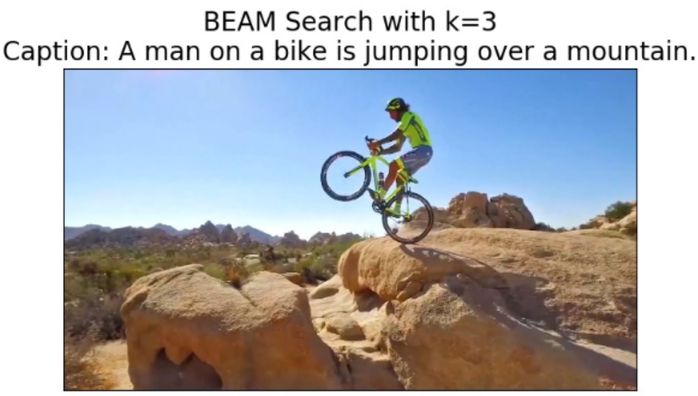

Figure 5.3: Environment Description Model in action

order to parse sign-boards, hoardings, etc. and extract useful information from them. This parsed information can then be conveyed to the user pretty conveniently.

The advantage of adding Speech Recognition to the device also opens up the possibility of creating a Conversational AI or chat-bot which can carry out certain tasks based on the needs of the user, the input for which will be available via the Speech-to-Text module. We will be using the SpeechRecognition library in Python in order to extract text from the user's voice input and send the extracted text to Dialogflow, via REST APIs, for intent recognition and based on the recognized intent perform specific actions.

5.2 Flow of System

Once the device is turned on the scripts will first check system and device health and then the status of the sensors and peripherals. If the device is being turned on for the first time, a demo or introduction script will execute. This script will walk the user through all the functionalities on the devices, its features, the modules, use of each button and so on. There will be a dedicated button to run this demo script since it may so happen that a user may forget what a certain button does or what a certain output means. Hence the user may be able to hear the demo/introduction as many times as he/she wants. Once the walk-through is finished, the device will start executing the actual program which will be a collection of multiple functions that will be called as and when needed by the system.

The camera module, with the help of the Object Detection model running on the

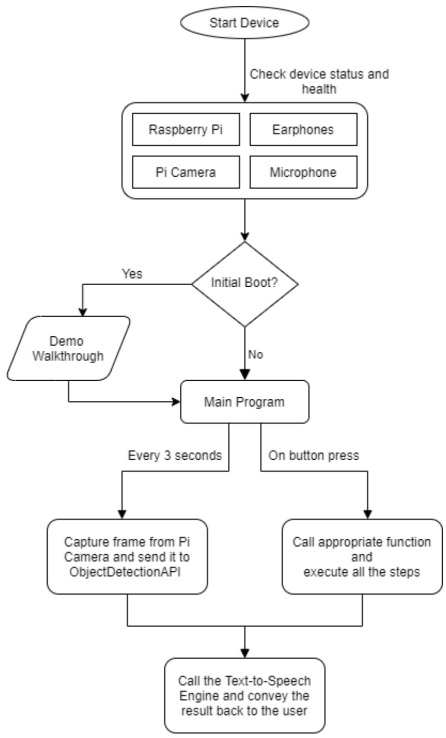

Figure 5.4: Flowchart of the proposed system

server, will continuously try to detect obstacles in the user's path and on successfully detecting one it will generate a sentence that is now to be conveyed to the user after which the Text-to-Speech Engine will convey the information to the user in an audio format.

A dedicated button will be present which when pressed will trigger the camera module to capture the current frame and send it to the Environment Description Module, running on the server, which will generate a description of the frame and send it back to the Raspberry Pi which will in turn convey it to the user using the Text-to-Speech Engine.

There will also be another button to start the Voice Recognition module. It will listen for a period of 5 seconds and if it is unable to capture any voice or command then the mic will be shut and the rest of the program will continue its normal execution. The captured text will then be sent to Dialogflow via REST APIs, where we already have an agent running, for intent recognition. Based on the output of Dialogflow we then understand what the request of the user is and give the appropriate output for any given query.

Every 5-10 seconds data such as run count of sensors, frequency of sensor activation, object detection results will be sent from the Raspberry Pi to the database on the cloud. This data will then be queried by the application on the Android phone to create various dashboards and debug the device in case of some unforeseen or abnormal event.

Chapter 6

UML Diagrams

6.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

Figure 6.1: Use Case Diagram

6.2 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.Activity diagram is basically a flowchart to represent the flow from one activity to another activity.The control flow is drawn from one operation to another.Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. Activity is a particular operation of the system.Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques

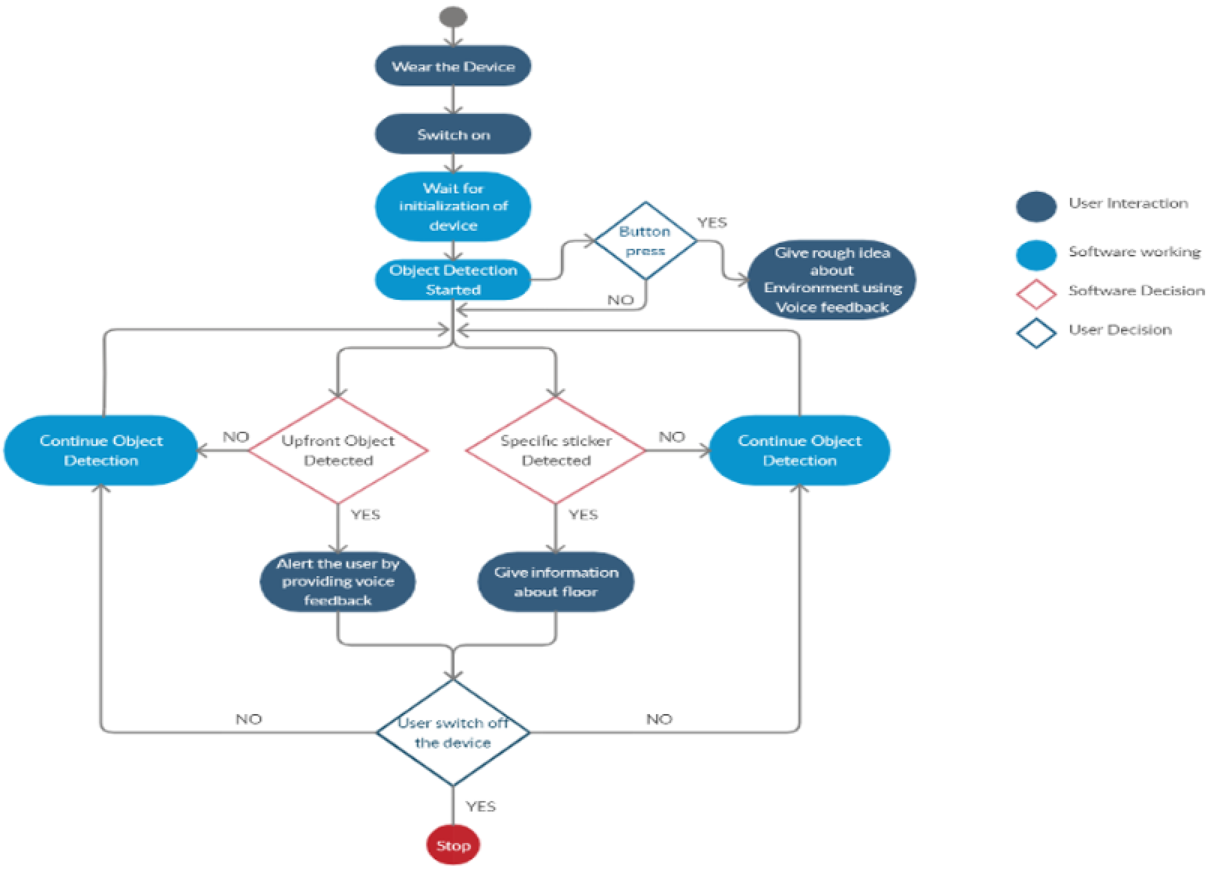

Figure 6.2: Activity Diagram

Chapter 7

Proposed Technology Stack

Figure 7.1: Technology Stack

Chapter 8

Project Implementation

8.1 Client

We start the program by declaring constants such as the server URLs as well as initialising the camera module. We also set the number of frames to be captured at any instance

```python
# API URLs
objectDetectionURL = 'http://'+IP_ADDRESS+'/api/objectDetection'
environmentDescriptionURL = 'http://'+IP_ADDRESS+'/api/environmentDescription'
dialogflowBotURL = 'http://'+IP_ADDRESS+'/api/dialogflowBot'

# Loading the TTS Engine
engine = pyttsx3.init()
engine.setProperty("rate", 150)
print('=== TTS Loaded ===')

# Initializing Camera Module
cap = cv2.VideoCapture(0)

# Setting resolution of images
cap.set(3, HEIGHT)
cap.set(4, WIDTH)

# Setting buffer size
cap.set(cv2.CAP_PROP_BUFFERSIZE, 1)
```

Figure 8.1: Client Declaration

We then go on to declare the trigger condition, which basically tell the program which function is to be called or what action is to be taken based on the users input

```python
if char == 'v' or char == 'V':
    # initialize recognizer
    r = sr.Recognizer()
    r.dynamic_energy_threshold = False
    # Mention source, it will be either Microphone or audio files.
    with sr.Microphone() as source:
        engine.say("Please speak after the beep")
        engine.runAndWait()
        playsound(BEEP_AUDIO)
        try:
            audio = r.listen(source, timeout=5.0)
            # use recognizer to convert our audio into text part.
            text = r.recognize_google(audio)
            result = DialogflowAPI(text).split(";")
            if (result[0] == "WELCOME"):
                WELCOME()
            if (result[0] == "ABOUT_ME"):
                ABOUT_ME()
            if (result[0] == "ABOUT_COLLEGE"):
                ABOUT_COLLEGE()
            if (result[0] == "DATETIME"):
                DATETIME()
            if (result[0] == "FEATURES"):
                FEATURES()
```

Figure 8.2: Trigger Conditions

8.2 Server

On the server side, we first load both the models using two different graphs as well as two different sessions

```
# Max sequence length (from training)
max_length = config['max_length']

# Load the RNN model with new TF Session and new Graph
caption_model = ''
graph1 = Graph()
with graph1.as_default():
    session1 = Session()
    with session1.as_default():
        caption_model = load_model(config['model_load_path'])
print('=== Caption Model Loaded === \n')


# Load the InceptionV3 model with new TF Session and new Graph
image_model = ''
graph2 = Graph()
with graph2.as_default():
    session2 = Session()
    with session2.as_default():
        image_model = InceptionV3(
            weights='model_data/object_detection/inception_v3_weights.h5')
        image_model.layers.pop()
        image_model = Model(inputs=image_model.inputs,
                            outputs=image_model.layers[-1].output)
print('=== InceptionV3 Model Loaded === \n')
```

Figure 8.3: Server Declaration

This is a snippet of the data that was used to train the Environment Description Model. Every line contains a caption for every image in the data-set

```
1000268201_693b08cb0e A child in a pink dress is climbing up a set of stairs in an entry way .
1000268201_693b08cb0e A girl going into a wooden building .
1000268201_693b08cb0e A little girl climbing into a wooden playhouse .
1000268201_693b08cb0e A little girl climbing the stairs to her playhouse .
1000268201_693b08cb0e A little girl in a pink dress going into a wooden cabin .
1001773457_577c3a7d70 A black dog and a spotted dog are fighting
1001773457_577c3a7d70 A black dog and a tri-colored dog playing with each other on the road .
1001773457_577c3a7d70 A black dog and a white dog with brown spots are staring at each other in the street .
1001773457_577c3a7d70 Two dogs of different breeds looking at each other on the road .
1001773457_577c3a7d70 Two dogs on pavement moving toward each other .
1002674143_1b742ab4b8 A little girl covered in paint sits in front of a painted rainbow with her hands in a bowl .
1002674143_1b742ab4b8 A little girl is sitting in front of a large painted rainbow .
1002674143_1b742ab4b8 A small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it .
1002674143_1b742ab4b8 There is a girl with pigtails sitting in front of a rainbow painting .
1002674143_1b742ab4b8 Young girl with pigtails painting outside in the grass .
1003163366_44323f5815 A man lays on a bench while his dog sits by him .
1003163366_44323f5815 A man lays on the bench to which a white dog is also tied .
1003163366_44323f5815 a man sleeping on a bench outside with a white and black dog sitting next to him .
1003163366_44323f5815 A shirtless man lies on a park bench with his dog .
1003163366_44323f5815 man laying on bench holding leash of dog sitting on ground
1007129816_e794419615 A man in an orange hat starring at something .
1007129816_e794419615 A man wears an orange hat and glasses .
```

Figure 8.4: Training Data for Environment Description Model

Chapter 9

Testing

For the testing purpose we opted to go for the functional testing methods. Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. Function testing includes:
1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance Testing

9.1 Unit Testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing also makes debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process.

Therefore, opting for the unit testing method in our project played a crucial role in assessing each module of the application separately. This testing method best suited our project as we had various modules at the start which were to be tested and verified. It made the testing process easier by helping us discover the minute errors in each module and therefore we could rectify them efficiently

9.2 Integration Testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

Integration testing was a necessity to check whether each individual module/unit was working well in synchronisation with one another. There were multiple problems while integrating the various modules which were only discovered with the help of integration testing methodology.

9.3 System Testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production. Finally, the entire system was tested as a whole using the System testing Methodology.

Here, the functional requirements of our applications include connecting to the server, generating descriptions for images, detecting objects from images, recognizing users voice and building conversation. The device was given to other professionals (faculty) who checked and verified the proper working of all the modules and the system as a whole.

9.4 Acceptance Testing

Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user's needs. This requires the product be tested both internally and externally, meaning you'll need to get it into the hands of your end users.

Acceptance Testing method was adopted for our application to get a final review of our application by the actual end users. The application was given to fellow peers to try hands on and get their opinions and recommendations which were worked upon. Also, it was seen that all end user needs were met to their complete satisfaction after using the application.

Chapter 10

Result

10.1 Object Detection

	Person	**Chair**	**Table**	**Laptop**	**Car**	**Total**
Person	20	-	-	-	-	20
Chair	2	15	3	-	-	20
Table	-	5	15	-	-	20
Laptop	-	-	3	17	-	20
Car	-	-	2	2	16	20
Total	22	20	26	16	16	100

Table 10.1: Confusion Matrix of the Object Detection Module

The Object Detection Module of the device is able to detect common campus objects with an accuracy of 83% after being trained on a set of only 50 images of each object. The data-set was split into two parts: 70% of it was used for training the model and the rest 30% of it was used to evaluate the model

10.2 Environment Description

BLEU-1	0.605109
BLEU-2	0.356089
BLEU-3	0.356089
BLEU-4	0.129909

Table 10.2: BLEU Scores for the Description Model

Since the Environment Description Module has been trained on only 8000 images that were available in the Flickr8K data-set, it is able to accurately generate text for simple images but its accuracy decreases as the complexity of the images increase.

10.2.1 Correct Results

Figure 10.1: A few correct results given by the model

10.2.2 Partially Correct Results

Figure 10.2: Examples of partially correct results given by the mode

10.2.3 Incorrect Results

Figure 10.3: Examples of incorrect results given by the model

Chapter 11

Conclusion and Future Scope

The main motive of our work is to create a trustworthy, efficient and real-time device for the visually impaired so that they can roam around their surroundings freely, without the need of being helped or relying on others even for the simplest pleasures of life. We hope to remove the drawbacks of all the previously mentioned illustrations wherein an Android Application is a required dependency for even setting up the device. The powerful model that we built using a pre-trained model gives us higher accuracy due to the concept of Transfer Learning. Since we are relying on the model to predict the objects in every frame of the captured videos, the task of predication is computationally very expensive which means that most of the resources of the Raspberry Pi would be consumed by the Object Detection and Environment Description modules itself. Hence, the introduction of an additional server or processing unit helps reduce the response time and achieve near real-time processing. .

Bibliography

[1] Chang, Y.-H., Sahoo, N., & Lin, H.-W. (2018). An intelligent walking stick for the visually challenged people. 2018 IEEE International Conference on Applied System Invention (ICASI)

[2] Yadav, A. B., Bindal, L., Namhakumar, V. U., Namitha, K., & Harsha, H. (2016). Design and development of smart assistive device for visually impaired people. 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)

[3] Munteanu, D., & Ionel, R. (2016). Voice-controlled smart assistive device for visually impaired individuals. 2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC)

[4] Agarwal, R., Ladha, N., Agarwal, M., Majee, K. K., Das, A., Kumar, S., Saha, H. N. (2017). Low cost ultrasonic smart glasses for blind. 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)

[5] Lee, C.-N., Chu, Y.-T., Cheng, L., Lin, Y.-T., & Lan, K.-F. (2017). Blind assistive device - Smart Lazy Susan. 2017 International Conference on Machine Learning and Cybernetics (ICMLC)

[6] Kasthuri, R., Nivetha, B., Shabana, S., Veluchamy, M., & Sivakumar, S. (2017). Smart device for visually impaired people. 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM)

[7] Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[8] Patil, P., & Sonawane, A. (2017). Environment sniffing smart portable assistive device for visually impaired individuals. 2017 International Conference on Trends in Electronics and Informatics (ICEI)

[9] Pawluk, D. T. V., Adams, R. J., & Kitada, R. (2015). Designing Haptic Assistive Technology for Individuals Who Are Blind or Visually Impaired. IEEE Transactions on Haptics, 8(3), 258–278.

[10] Aymaz, S., & Cavdar, T. (2016). Ultrasonic Assistive Headset for visually impaired people. 2016 39th International Conference on Telecommunications and Signal Processing (TSP).

[11] Rehan, M., Kumar, A., Sree Sai Sharan P, & Hegde, R. (2015). Design and analysis of a collision avoidance system for the visually impaired. 2015 International Conference on Communications and Signal Processing (ICCSP)

[12] World Health Organization (WHO), https://www.who.int/newsroom/fact-sheets/detail/blindness-and-visual-impairment, last accessed on 29.07.19

[13] Wikipedia, https://en.wikipedia.org/wiki/Assistive_technology, last accessed on 29.07.19

[14] CS231n, Stanford University, http://cs231n.github.io/transferlearning/, last accessed 16.08.19

[15] Flickr8k Dataset, https://forms.illinois.edu/sec/1713398

Appendices

Appendix-A: Installation of Client and Server Requirements

1. Clone the repository containing the code
git clone https://github.com/sameer-m-dev/voiceEnabledSmartDeviceForVisuallyChallengedPeople

2. Run the following commands to create separate virtual environments for client and server
virtualenv client
virtualenv server

3. Run the following commands to install all the dependencies for the client
cd client
source bin/activate
pip install -r requirements.txt

4. Run the following commands to install all the dependencies for the server
cd server
source bin/activate
pip install -r requirements.txt

5. Use the following command to start the client
python mainFile.py

6. Use the following command to start the server
python run.py

Acknowledgement

We have great pleasure in presenting the report on **Voice Based Smart Assistive Device for the Visually Challenged** We take this opportunity to express our sincere thanks towards our guide **Prof. Kiran B. Deshpande** & Co-Guide **Prof. Kaushiki Upadhyaya** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student Name1: Sameer Dev
Student ID1: 16104030

Student Name2: Sudama Jaiswal
Student ID2: 16104015

Student Name3: Yogendra Kokamkar
Student ID3: 16104016

Publication

Paper entitled **“Voice Based Smart Assistive Device for the Visually Challenged ”** is presented at **“IEEE International Conference on Convergence to Digital World – Quo Vadis (ICCDW 2020) ”** by **“Sameer Dev”**