Affordable Voice Assistive Intelligent Glass for Visually Impaired People in Perspective of Bangladesh

By

Md. Mahmudul Kabir Peyal
17321014
Quazi Md. Ahnaf Ul Haque
17221012
Tashfia Tahiat
20121083
Sadia Habib
17321041

A thesis submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical & Electronic Engineering

Department of Electrical and Electronic Engineering
Brac University
June 2021

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Mahmudul Kabir Peyal 17321014	**Quazi Md. Ahnaf Ul Haque** 16221032
Tashfia Tahiat 20121083	**Sadia Habib** 17321041

Approval

The thesis/project titled "Affordable voice assistive intelligent glass for visually impaired people in perspective of Bangladesh" submitted by

1. Quazi Md. Ahnaf Ul Haque (17221012)
2. Md.Mahmudul Kabir Peyal (17321014)
3. Tashfia Tahiat (20121083)
4. Sadia Habib (17321041)

of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 9th June, 2021.

Examining Committee:

Supervisor:
(Member)

_____

Dr. A. K. M. Abdul Malek Azad
Professor
Department of Electrical and Electronic Engineering BRAC University, Dhaka.

Program Coordinator:
(Member)

_____

Abu S.M. Mohsin, PhD
Assistant Professor
Department of Electrical and Electronic Engineering,
BRAC University, Dhaka

Departmental Head:
(Chair)

_____

Md. Mosaddequr Rahman, PhD
Professor and Chairperson
Department of Electrical and Electronic Engineering,
BRAC University, Dhaka

Abstract

This paper describes design of affordable intelligent glass for visually impaired. A significant number of people are visually impaired or blind among the total population of Bangladesh. Most of them relied on others directly or indirectly which makes their life challenging. In terms of academic and social perspective, they lag behind from their fellow mates. To help these people and lead a better life, artificial intelligent glasses can play a vital role in today's world. Through this smart glass visual system will convert into an audible system in real time by using a single board computer. Convolutional neural network has been introduced to help visually impaired people to classify objects. Moreover, text recognition will be available for recognizing and reading Bangla as well as English text. With the help of voice command users might be able to control electrical gadgets and be able to communicate with caregivers through mobile app.

Keywords— smart glass; raspberry pi 4; deep learning; Image classification; Speech processing; Convolutional Neural Network; optical character recognition; sensor; app;

Dedication

This work is dedicated mainly to our parents for all their efforts and encouragement they have invested in us up until now. We also dedicate the work to our supervisor who has been more than a pillar in the development of the final work output.

Acknowledgement

We would like to express our gratitude to Dr. A K M Abdul Malek Azad, Professor, BRAC University, Department of Electrical and Electronic Engineering (EEE), for his assistance, guidance, and feedback for the completion of our thesis. We would also like to acknowledge the Control & Applications Research Centre (CARC) of Brac University and research assistants associated with CARC for providing the continuous support needed to accomplish this project with their valuable inputs.

Table of Contents

List of Tables

List of Figures

List of Acronyms

OCR – Optical character recognition

CNN – Convolutional Neural Network

ResNet – Residual Network

API – Application Programming Interface

GPU – Graphics Processing Unit

TPU – Tensor Processing Unit

Eflops – Exa Floating Point Operations Per Second

JSON – JavaScript Object Notation

NoSql – NonStructured Query Language

Chapter 1

Introduction

1.1 Background

Vision is an essential part of our lives. Most of what we perceive of the outside world is through our eyes. A person without it, is partially handicapped. The quality of life drops and assistance of another person is required. Thus, the person becomes dependent. According to a case study of Caring Vision Hospital, the doctor to patient ratio in Bangladesh is currently 1:162.494. Moreover, it also says, the number of patients aged 30 or above is 750,000, due to cataracts. Ophthalmologists say cataracts may also be developed for the increase in exposure of UV light [1].

The evolution of technology has made it possible to come up with glasses which can make it much easier for you to see by the help of artificial intelligence. Through image processing the glasses can make the view more perceivable according to the need of the patient. Additionally, features such as reading out text for you are also available. But all of these do come with a heavy price to pay. In a country where the average annual income was found to be 602.549 USD, one can just imagine how many people will be able to afford even google glasses which are about 999 USD [2]. If we look at previous methods which are more affordable such as Braille books, the restrictions will be unavoidable. Among which would be the lack of availability of teachers and available resources are severely understated.

1.2 Literature Review

Lots of technology have already introduced help visually impaired persons. Lin in his paper [3] proposes a smart glasses application system for visually impaired people based on deep learning. The paper explains the working procedure of the system which can provide voice feedback to visually disabled people about the objects in front of them by uploading images to our backend object recognition device through the camera feature of smart glasses, then downloading the text explanations of the effect and using the text-to-speech function. A system proposed by AlSaid [4] is about an attempt to build a novel smart glass that can extract and recognize text captured from an image and convert it to speech. The paper elaborates about the glasses which are designed to use computer vision technology to capture an image and extract English text and convert it into an audio signal with the aid of speech synthesis based on deep learning techniques to help blind and visually impaired people. Similarly, Fing Lan [5] discusses about giving blind and visually impaired people accessibility with the smart glass system which can detect public signs in cities and gives corresponding voice instructions and hints through wireless bone conduction headphones. The paper also describes the working principle and the implementation of the application which is used for recognizing the public sings and give voice hints to the blindness. In addition, Karthik, V. K. Raja and S. Prabakaran [6] Evaluates and Discusses the use of OCR tool based on raspberry pi for detection of environmental messages. The image captured is processed to improve the quality so that any blur present can be removed. Additionally, TTS technology is used to communicate with the user. Furthermore, S. A. Jakhete [7] Describes how we can even detect objects using a mobile application by using SSD (Single Shot Detector) algorithm. SSD is uses convolutional neural network to classify objects from a dataset of 80 classes. In closing, N.

Sumi and V. Moshnyaga in their paper [8] shows a face recognition system is developed to detect multiple faces in a video stream with people at a specific range in distance.

1.3 Motivation

Living in a society where people with low eyesight vision have difficulties in leading a life and the problem with their eyesight does not even allow them to take a walk alone safely. Moreover, old people without any caregiver have difficulties to do their regular activities. For example, taking their medication on time and using electrical appliances. Family members are always in a tense state about their safety issue wherever they are. Even though they don't have vision, they can overcome it by using their hearing ability. So, we thought of building such a technology where their vision can be alternatively resolved by the help of smart glasses.

1.4 Thesis Organization

The rest of the dissertation is organized as follows –

Chapter 2: *System Design*

In this chapter, the overall concept of the system design of the smart glass was presented. Initially, A CAD general concept was formed about how the smart glass should have looked, then for a test, a practical design was implemented. The Hardware and Software Requirements for establishing the smart glass was also been discussed shortly in this chapter.

Chapter 3: *Image Classification*

In image classification chapter we have showed how we have built image classification model and how can we use it to predict object from real life camera streaming. This is very important feature

for a visually impaired person. Predicting object correctly can make their life lot easier. They will able to do their work by themselves if image classification can be implemented correctly. In this chapter we have also describes how deep learning algorithm works and how can we evaluate model by testing its accuracy. Finally, we have concluded this chapter by showing some simulation and test result.

Chapter 4: *Text Detection and converting it to speech*

In this chapter, the working of text detection algorithm is described. Further we discuss how the text from the picture is extracted then language identification takes place. Later the step of how text is converted to speech is shown. Lastly python libraries being used and their respective functions are discussed briefly. Afterwards, at the end the output results and its discussion have been shown.

Chapter 5: *Home Automation*

In home automation chapter we have briefly discussed about controlling home devices with voice commands. Blind person faces difficulties while moving around and its risky if they try to control ac loads manually. So, home automation can be a game changing solution. Moreover, this chapter tells how can we ensure safety of environment for visually impaired people by using different types of sensors. Sensors can work as a good feedback system while there are any possibilities of causing accident. In conclusion of this chapter shows prototype working model and how we test run that circuit successfully.

Chapter 6: *App Development for Helping Caregiver*

In this chapter, the working principle and features of App named “AI Smart Glass APP (Thesis)

“have been discussed. The app will help to make the blind and visually impaired people ‘s daily life difficulties a little easier. For example, by there are different features like health issues, personal issues by using these different sections the user can give update to other user about his/health or ask to bring his/her emergency stuffs. Moreover, the user can also voice chat with his/her closed one and give his/her updates. Furthermore, the app also gets an alarm if there is any gas leakage or fire near the blind/visually impaired people. Lastly, the working principle and each and every features of the app discussed briefly in this section.

Chapter 7: *Feasibility Analysis*

In this chapter, we have discussed about the project's impact in economy also how our project contribution on economic sustainability. Next, we have discussed about how our project contribute in societal sustainability also the impact of the project in our society. We also discussed about the environmental sustainability and impact of our project. Moreover, we discussed about applicable standards and codes here we talked about what we have planned for our project and how we execute our plans step by step for our project. Furthermore, we have talked about ethics what principle we have followed to maintain ethical policies. Lastly, we have discussed about how we ensure the user’s safety and took care of the risk management of our project.

Chapter 8: *Conclusion*

In this chapter, we go through the experience we had doing this project and to what extent we have met our objective. The socio economic impact of the project in the society has also been analyzed. Besides that, the safety feature of our design has been reviewed. We later elaborate through the future work, stating what are the improvements that can be done and how it can be made suitable for usage.

Chapter 2

System Design

2.1 General Design Concept

The proposed system is consisting of an on-board computer Raspberry Pi 4B+, camera, global positioning system module named NEO6, global mobile communications system module SIM800, node-red development tool, firebase database, earphone, etc. Firebase database is used to send, receive messages and control electrical gadgets. sensors and GPS module help users to measure the distance from the object, detect problems, sending location to a caregiver. Moreover, SIM800 modules are attached to give users online access even WIFI connection is not available. 8 MP camera is attached to take image input and detect an object as well as recognize text in real-time according to the user's need.

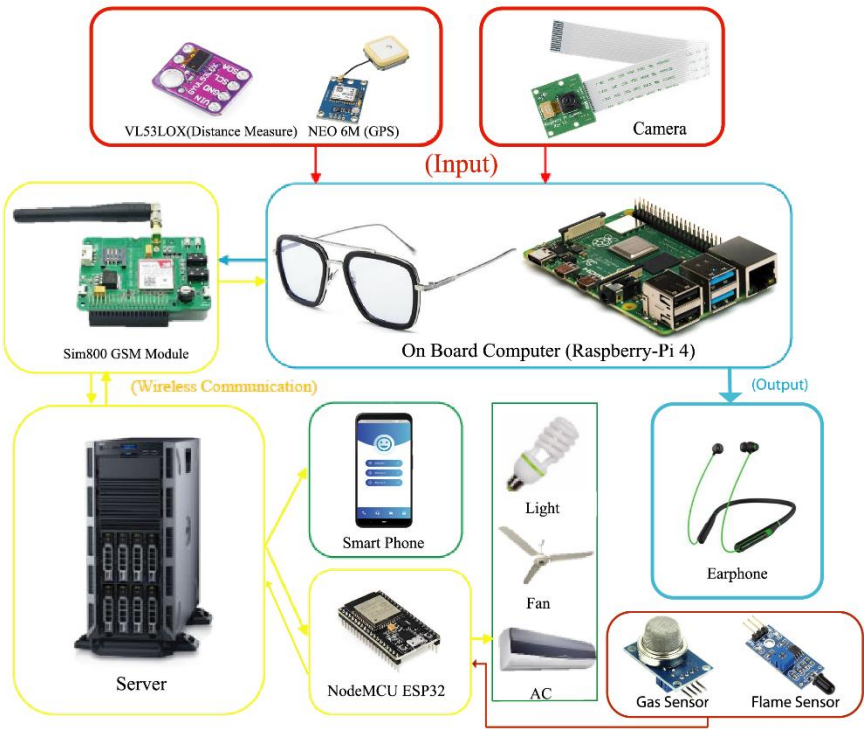


Figure 2. 1: Design Concept of the Smart Glass

Furthermore, to connect glasses with the external environment ESP32 micro-controller is introduced. This microcontroller will talk with glass and caregiver's app through online database. Microcontroller which is connected with electrical gadgets like light, fan, air conditioner, etc. can operate devices according to user command. ESP32 microcontroller is also attached with some sensors to detect environmental hazard like fire, gas.

2.2 Proposed Smart Glass Design

After compiling all the components of the smart glass, the final 3D design was proposed to be something like this –

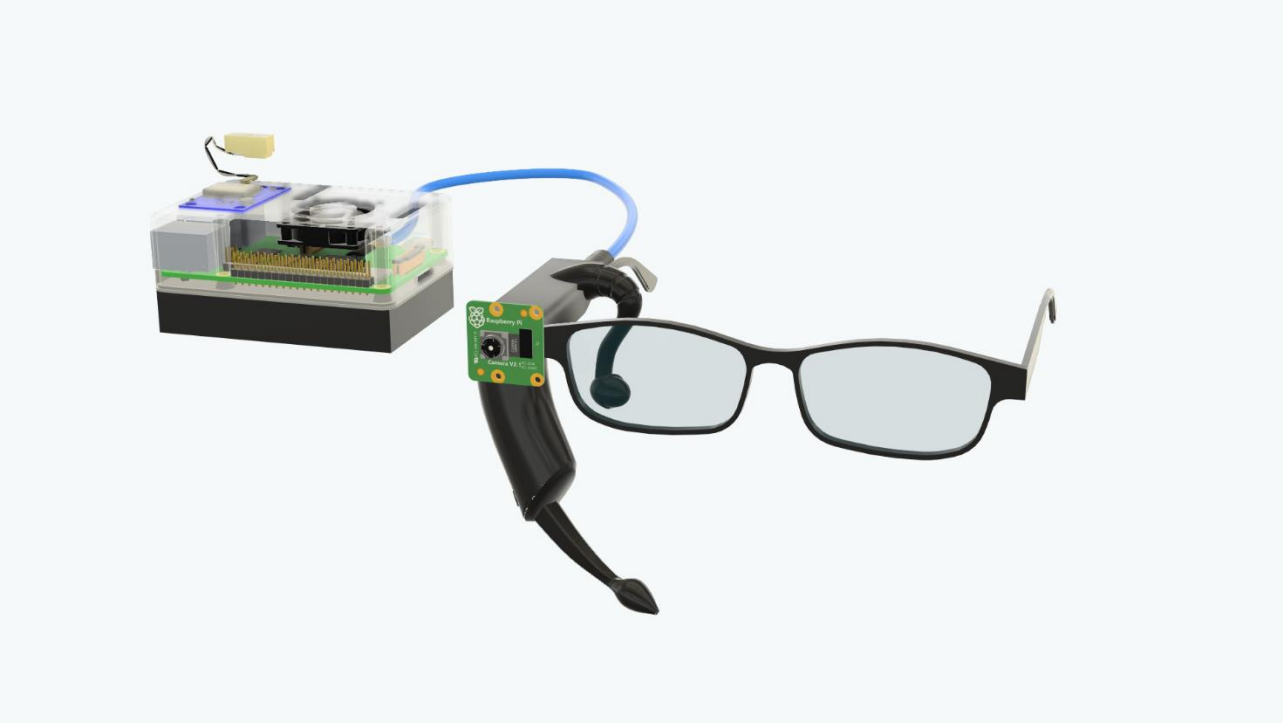


Figure 2. 2: CAD Design of the Smart Glass

Here, the camera is being fitted to one of the frames of the glass as this will act as virtual eyesight for the user, so for the convenience of detecting objects and reading texts the placement of the camera was chosen to be placed in such a position. The earphone and the microphone are also

being placed such that the user can deliver its audio through the microphone attached close to the person's mouth and also in order to hear the output obtained easily the earphone is being attached to the ears. Then afterward, this camera and the earpiece and microphone are being connected to a raspberry pi. The raspberry pi will be wired to the person's pocket. As it would be problematic putting it near the glasses due to the outlook and also because raspberry pi heats up when operated for a long period of time. A sound card is required and is being connected to one of the USB port of raspberry pi in order to connect the earpiece and get the audio output through the earphones as well as for obtaining output from the user through the microphone.

A test design was also done in our project in order to see if everything functions properly and the outputs were seemed to be obtained successfully with much accuracy and in a convenient way.

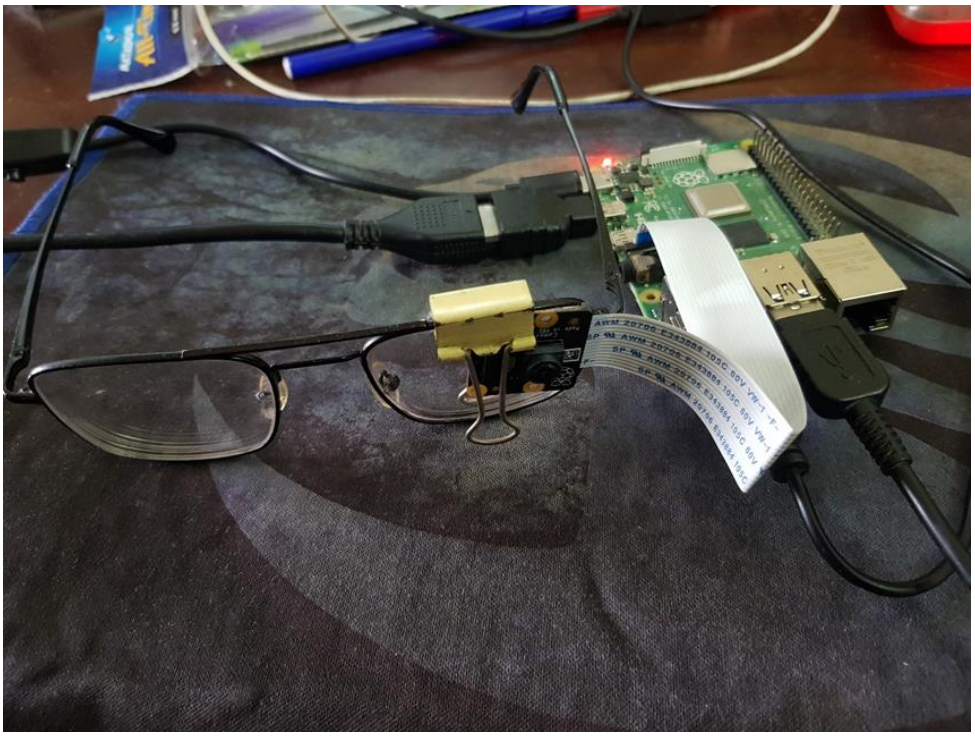


Figure 2. 3: Test Design of the Smart Glass

2.3 Hardware Requirements

Figure 2. 4: Hardware Components

2.3.1 Raspberry Pi 4

Raspberry pi is a single board compact computer that can be easily carried around. Raspberry also supports all kinds of code, which in contrast to Arduino is an advantage as Arduino only supports C, C++. Version 4 is the latest edition and has proven to be very cost-effective. Raspberry pi 4 has a higher processing speed and RAM than version 3. Additionally, it also supports multiple sensors due to the presence of many GPIO pins.

All these features make it suitable for this project as we are using python code and aiming to build a smart glass that is not only cost-effective but also looks appropriate.

Despite all the advantages, there are a few shortcomings such as lack of internal storage makes us require an SD card. An SD card makes the system much slower.

2.3.2 Pi-noir-camera (model-v2):

It is a very light weighing of 3 grams with a still resolution of 8 megapixels. It has a Sony IMX219 sensor with a sensor resolution of 3280 x 2464 pixels. It is compatible with all versions of the raspberry pi.

The higher the resolution of the camera the better the accuracy of object detection results. But in the case of smart glasses, the size of the camera is very important. Having a huge camera attached to the tiny glasses would put the user in an awkward and uncomfortable position. Pi-noir-camera was the best compromise between size, resolution, and budget.

2.4 Software Requirements

2.4.1 Google Colab

Colaboratory, or “Colab” for brief, is a product of Google analysis. Colab permits everyone to write down and implement random python code through the browser and is very fine coordinated to machine learning, data analysis, and research education. The big issue of working with deep learning model was to manage computer with high processing power or which computer has

powerful GPU or TPU access. Google colab solves this problem for us. Using deep learning algorithm with supervised learning needs large number of datasets. Training those datasets and build model with usual computer takes three to four-day times. However, we able to do it in two to three hours using colab GPU or TPU access. Though it has some limitations, it still helps a lot in research or project-based work.

2.4.2 Node-red

Node-RED is a new and fascinating program designing platform for the combination of hardware, APIs, and web properties. The programmer makes it possible to wire nodes in organized way using the broad range of nodes in the palette which can be used with a single click for their runtime. The reason 'Node' is in the name is that the method can be used as a node but is actually just internal implementation detailed from the user perspective. Node-RED has been adopted by the IBM Emerging Technology Organization and is distributed as an open-source [9]. Node-RED is a collaboration of JS Foundation [9]. It allows users to link Web services and hardware through replaced basic coding tasks of low level (like a simple service that speaks to a serial port), with a visual drag-drop interface and different Node-RED components are interconnected to construct a flow [10]. The structures of the Node-RED are: it supports browser constructed editing of flow, diverse flows formed in Node-RED stored in JSON, can be bring in and bring out easily for sharing, it is more simple to use in various devices such as Raspberry Pi, Arduino, Android-based devices. It can also operate in a cloud environment [10]. We use Node-Red for this project because it is very user-friendly. Moreover, by using node-Red, different node can be combined as an individual. To compile all the codes, we use node-RED. Lastly, it is used to combine all the independent codes together.

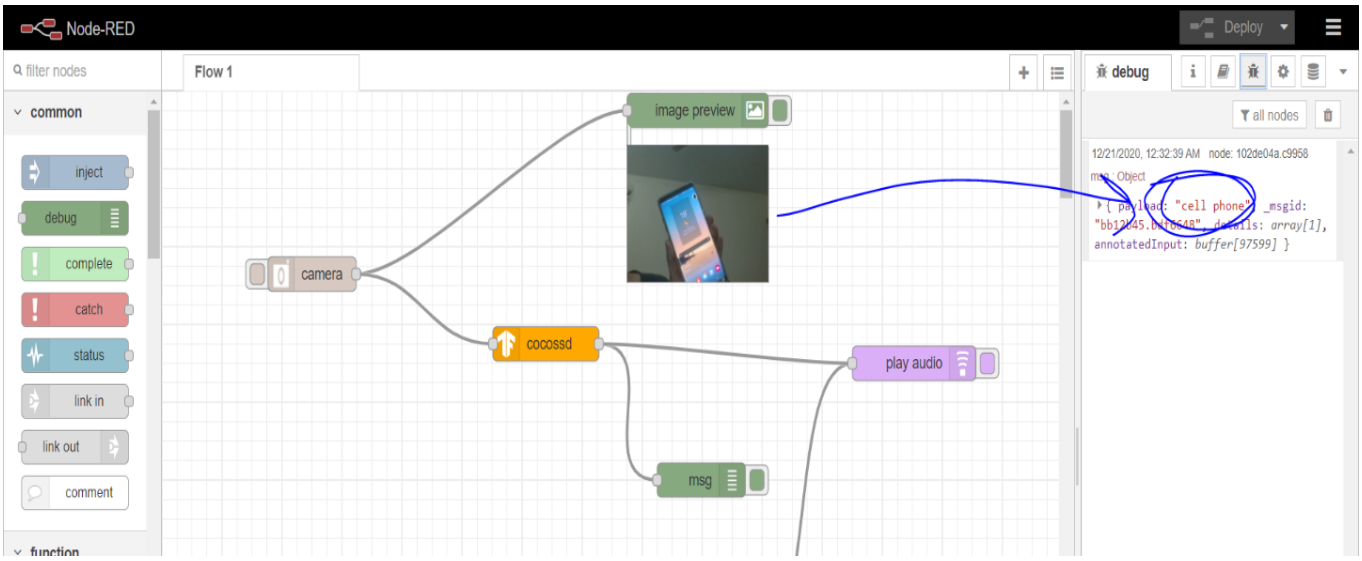


Figure 2. 5: Node red

2.4.3 Firebase Database:

Firebase is a google developed platform released in 2011 and owned by google in 2014 [11]. In our system, we have used two features of the firebase database. It makes application and software development easier. One is a cloud database and another is a real-time database. Cloud database is used to store and synchronize messaging data between smart glass and mobile app and for other features like sync mobile app's emergency status service, glass and app location sharing service, home automation system synchronization with smart glass and app real-time database has been used.

A Cloud database is a set of structured or unstructured information contents on a private, public, and hybrid application network for cloud computing and the behavior of the database is the same, whether you are using direct queries, such as SQL statements or API calls [12]. Two standard deployment model systems are used: operators can whichever operate databases separately on the

cloud by means of a virtual machine image, or they can buy access from a cloud database vendor to a database. Any of the cloud databases are SQL-based and some use a NoSQL architecture [13].

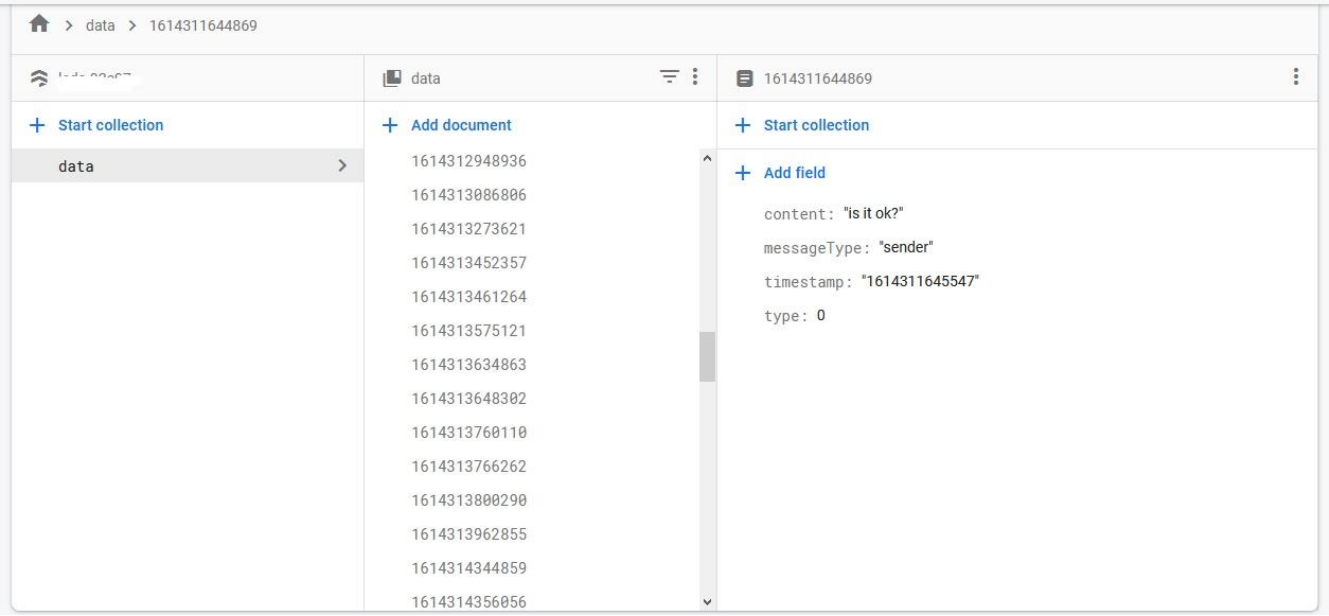


Figure 2. 6: Cloud Database

In this project, cloud database has been used for chatting as the provider manages and operates the database applications and makes users responsible for their own data only also it is cost savings [12].

Realtime Database is cloud hosting database. However, it saves data as a large JSON tree, making it simpler to store basic data, but more difficult to organize complicated hierarchical data whereas cloud database also stores the simple data in same way just like the JSON but complex, structured data is easily organized as database subcollections [14]. Only on iOS and Android offline support is available to Realtime Database, while cloud database offline support is provided for iOS, Android and web clients [14]. Realtime Database allows simple transaction and writing operations whereas atomic writing and activities are facilitated by Cloud Fire store [14]. Realtime Scaling includes sharing, which means spreading the data over many databases over 100,000 concurrent links, and 1,000 writes/seconds in one database whereas Scaling is completely automated in Cloud database (after beta), meaning that it does not take sharing the data across

Figure 2. 7: Realtime Database

Realtime database only charges a higher rate for bandwidth and data storage whereas Cloud database charges for operations in the database (reading, writing, deleting) are performed at a lower rate, bandwidth and storage also it has daily spending limits to avoid over cost [14].

Chapter 3

Image Classifications

This project's basic objective is to predict different types of objects and classify them using a deep learning algorithm. There are various types of algorithms available for training images and building models. However, we are proposing a convolutional neural network [15] algorithm for image object prediction because of the high accuracy result. whole image training workflow is showed in Figure 3.1.

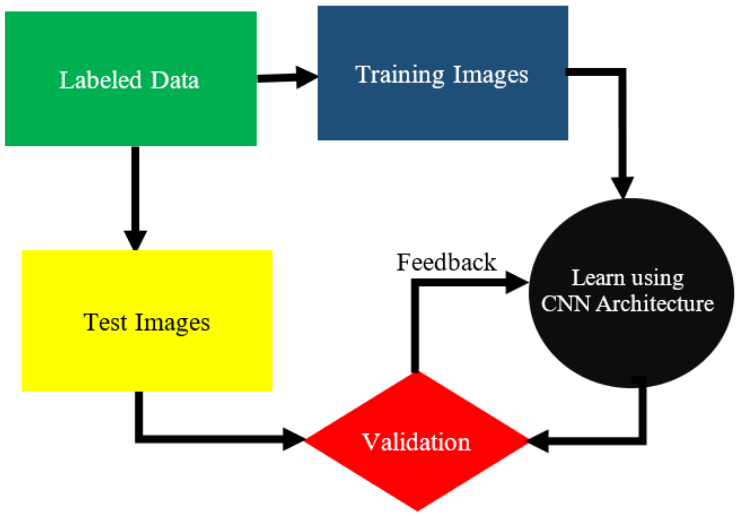


Figure 3. 1: Deep learning algorithm workflow

3.1 Theoretical background

3.1.1 Machine learning algorithm

Deep learning is a subfield of machine learning. The machine learning algorithm can be divided into three learning techniques. One is supervised learning which means learning by pretraining where input and output features are given to the systems. Another technique is supervised learning where the system has to find the feature on its own. The final machine learning technique is

reinforcement learning. This type of algorithm works using system intelligence in the environment with proper positive and negative feedback. We have chosen a neural network algorithm for implantation and this is a supervised learning technique.

3.1.2 TensorFlow

TensorFlow is an end-to-end open-source framework for machine learning. It incorporates a comprehensive, adaptable environment of devices, libraries, and various resources. This is a product of Google released in 2015 which helps managing datasets and it has various pre-trained models with superior polishing. This platform supports various programming languages like C++, C#, Java, Python, etc. This system is able of preparing deep neural systems to perform plentiful tasks that can solve varieties of complex engineering problems, such as object recognition, image classification, video detection, speech recognition, text-based application, and so on [16].

3.1.3 Keras

Keras is python based open-source library which provides a simple and compatible API to work with the artificial neural network. Keras is very user-friendly and runs as an interface for TensorFlow. Keras is ascendible and using the TensorFlow distribution strategy API, we are able to run our models on massive GPU clusters or a whole TPU, representing over one EFLOPS (1018) of computing power. Keras additionally has native support for mixed-precision training on the most recent NVIDIA GPUs likewise as on TPUs, which increase speed up to two times for coaching and illation [17].

3.2 Building Dataset

Deep learning algorithms need large datasets to predict images accurately. Moreover, this training process is time-consuming. It is also hard to manage TPU or GPU for training a large amount of data. So, we have trained our images separately by dividing datasets into four separate models. Models are food model, outfit model, home appliance model, and vehicle model. Each model contains different types of classes. Datasets are selected according to the perspective of Bangladesh. Datasets are collected from some open source websites like Kaggle [18], Mendeley Data [19], and some e-commerce sites Chondralika, Konoklota, etc. Food model contains rice, fish, vegetable, bread, curry classes. Outfit model consists of sharees, panjabi, lungi, shoes, pants, shirts, suits, dress classes. Further model home appliance model contains almirah, bed, chair, basket, door, jug, mug, refrigerator, sofa, switchboard, table, toilet, washroom. Finally, the vehicle model contains images of bus, car, rickshaw, train, motorcycle, truck, and CNG. All datasets according to model and classes are given in Table 3.1.

	Food	Outfit	Home appliance	Vehicles
Total class	5	8	13	7
Training Images	2158	5775	7712	3348
Testing Images	645	2209	2177	1230

Table 3. 1: Dataset Distributions

Some sample images are shown in figure 3.2:

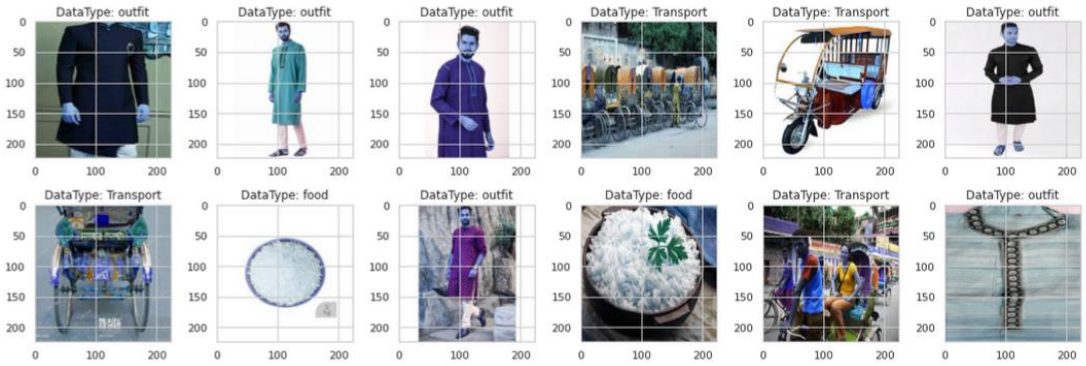


Figure 3. 2: Sample of training images used as dataset

3.3 ResNet50 Architecture

For image classification ResNet50 [20] neural network architecture has been introduced for our system. Residual Network type neural network first presented in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their paper "Deep Residual Learning for Image Recognition" [21]. Other neural networks usually use more additional layers to improve accuracy for the complex dataset. However, it causes degradation of performance and error increases. It also creates a gradient vanishing problem in terms of model accuracy. This problem is removed in ResNet architectural design. The basic structure of the residual block in ResNet architecture is shown in Figure 3.3. Resnet neural networks are made up of Residual Blocks.

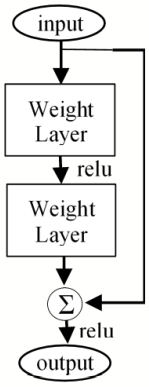


Figure 3. 3: Basic architecture for ResNet Model

From figure 3.3 we notice that one connection connects the input to output directly which is called 'skipped connection'. Except for this connection, the other connection is multiplied by some weights. Moreover, these layers go through some activation functions. If the network's output is W(X) we can write the function as:

$$W(X) = F(X) + X \quad (1)$$

This function sometimes creates a problem that the dimension of input of F(X) and X cannot become the same. To solve this, we can one 1*1 convolutional layer to input-output direct connection side, and the function will become like this:

$$W(X) = F(X) + W_1.X \quad (2)$$

ResNet models were very much successful from the beginning because of good accuracy and it uses less parameter to train images. We have used ResNet50 neural network consists of fifty deep layers. This network takes an image input size of 224*224*3 and gives output as probabilities. Different types of convolutional layers and activation function has been used to get the final output. All layer operations can be done easily to the work by adding layers name provided by the Keras API [15].

3.4 Model Layers

This architecture consists of 176 layers, where 53 nodes are conv2d layers, 16 summation nodes, 53 batch normalization layers, and 50 activation layers. Moreover, it has 1 dense layer, 1 global max-pooling layer, 1 input, and 1 output layer. The CNN layers used in ResNet50 are explained in detail as follows.

3.4.1 Conv2d Layer

This layer filters input continuously and the filtered output send to the relu activation function to get the output for the next layer. In this layer filter, kernel size and strides arguments are used [22]. Kernel size determines the dimension of the 2d convolution window. Strides specify the strides of convolution in terms of heights and width.

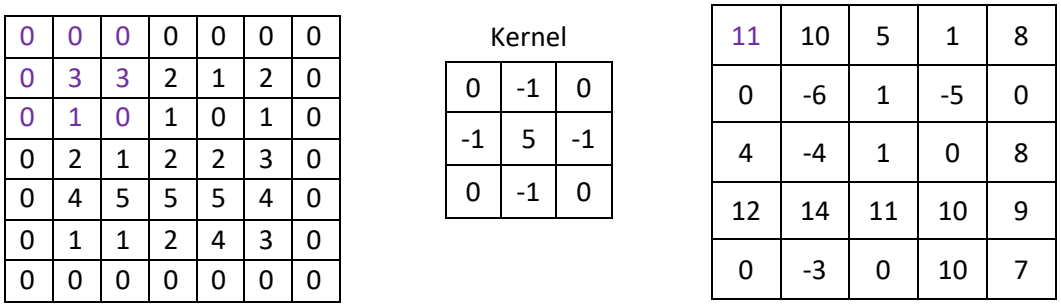


Figure 3. 4: Convolutional layer calculation method

3.4.2 Batch Normalization Layer

These nodes convert input into small batches for good regularization and loss reduction. Batch normalization calculates the mean and standard deviation of each input variable to a layer per batch and by using these results it performs the standardization. This helps the network to become more stable and reduce the number of training epoch results speeding up the training with a faster learning rate. Batch normalization layers are applied as follows for a small batch:

$$Z = g(w, x) + b \tag{3}$$

$$Z_{bn} = \frac{Z-m}{Sd} \times \gamma + \beta \tag{4}$$

Here,

Z_{bn} = Output of Batch normalization

g () = Linear transformation of the neuron

w = Weights of the neuron,

b = Bias of the neurons,

m = Mean of the neurons' output

S_d = Standard deviation of the neurons' output

γ, β = Learning parameters.

γ and β shift the mean and standard deviation, respectively**.** This way outputs of batch normalization over a layer results in a distribution with a mean β and a standard deviation of γ. These values are learned over epochs which helps to decrease the loss of the model.

3.4.3 Pooling Layers

Pooling layers take sample feature from convolutional layers and make new feature maps by finding unique feature from that convolutional image array [22]. The size of filter is smaller than the size of the feature map which is 2×2 pixels applied with a stride of 2 pixels which reduces size of feature map. Max pooling is the mostly used methods to summarize the most active presence of feature. Formula used for max pooling can be shown as:

$$x_{mp} = \frac{I_s - p}{s} + 1 \qquad (5)$$

Here,

X_{mp} = Max pooling output which must be an integer value

I_s = Input shape, P = pooling window size, S = strides

Such as a pooling layer applied to a feature map of 8×8 (64 pixels), 2×2 pooling window size, and strides = 2 will make an output pooled feature map of 4×4 (16 pixels). In our model global pooling layers are also used which downsamples the entire feature to a single value for finalizing the output for predictions.

10	**30**	8	2
8	20	0	**37**
40	80	**32**	2
100	**113**	24	15

→

30	37
113	32

Figure 3. 5: Max pooling layer

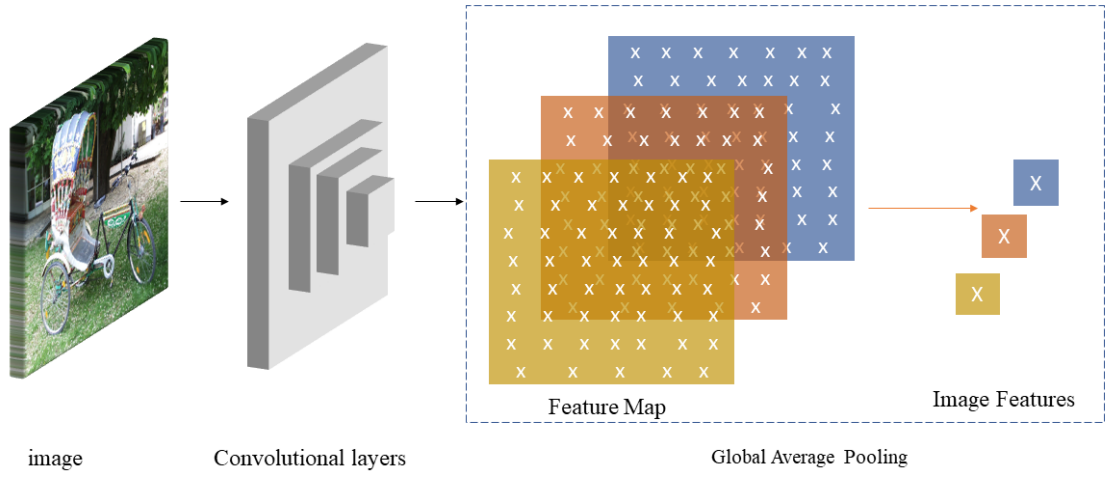


Figure 3. 6: global average pooling layer

3.4.4 Activation Layer

Relu: Relu function is a piecewise linear function, which gives output if and only if the input is positive. Otherwise, it gives zero as output. It solves the vanishing gradient problem that occurred in the model consist of many layers. Relu piecewise activation function can be expressed as follows [23].

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \qquad (6)$$

SoftMax: This activation function is an exponential function, which converts vector numbers into vector probabilities. This function is very much helpful for multiclass classifications. This function can be written as follows [23].

$$f_i(\vec{x}) = \frac{e^{xi}}{\sum_j^k e^{xj}} \qquad (7)$$

Where,

$\vec{x} = input\ vector$

$i = 1,2, \dots \dots \dots, k \qquad here, k = number\ of\ classes\ in\ classifier$

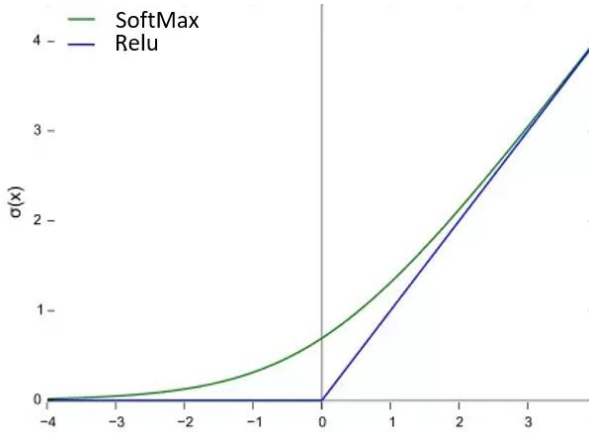


Figure 3. 7: Activation functions graph

3.5 Model Compilation and Image Training

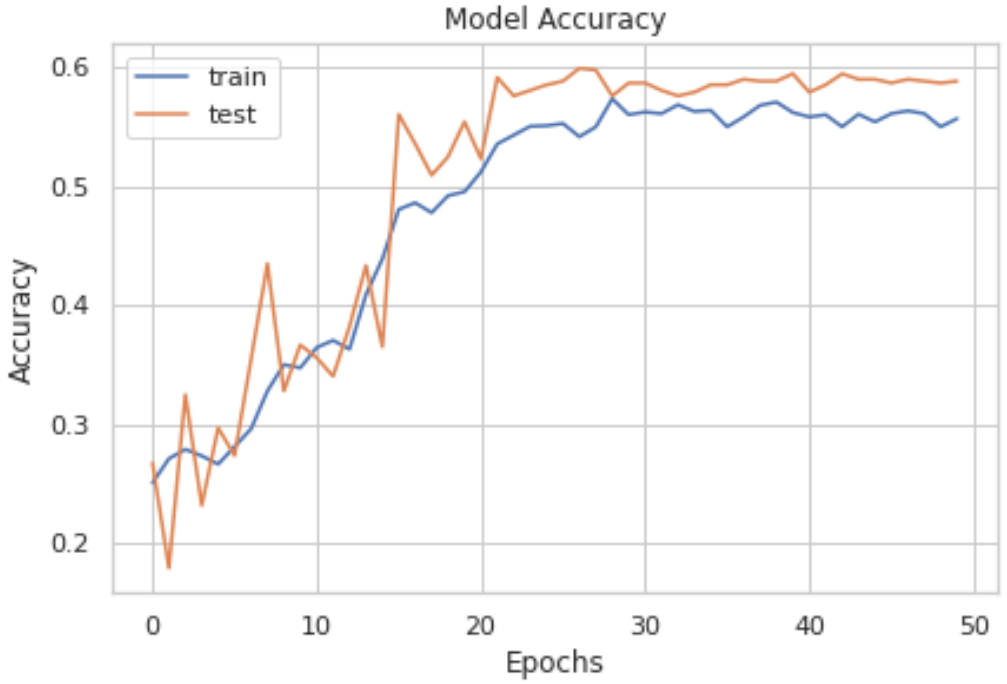
All models are compiled with the ADAM optimizer [24] and categorical cross-entropy is used as a loss function. As the image input size of the ResNet50 network is fixed, images have been resized and cropped accordingly. Images are also converted to RGB format. At the time of training, the network has taken a batch of 32 images and tun experiments 50 times. Finally, the number of outputs defines the number of classes for each model.

After completing the whole training process full model network with weighted value can be saved and use for prediction. After training images. Models are ready to predict the object.

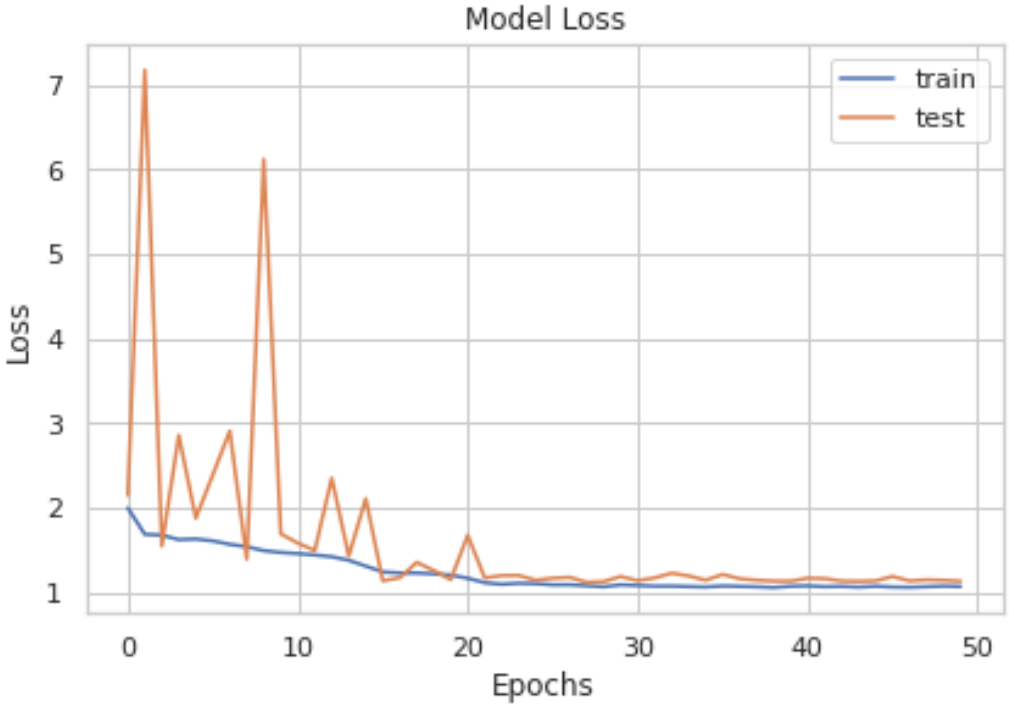
3.6 Result Analysis

For training images, we had used google colab with TPU access the whole time as we did not have a high processing power computer. Google colab had some limitations in terms of timing and computer space. Though colab has paid version that is not available in Bangladesh. However, the output result of our image training is satisfactory.

We run training process on four types of datasets for fifty epochs. Figure 3.8 describes both training and testing accuracy as well as loss history of food classification model. Figure 3.9 shows accuracy and loss history of outfit classification model. In similar fashion, figure 3.10 and figure 3.11 shows the accuracy and loss history graph of home appliance and vehicles classification model with respect to the number of epochs while it was in training and validation stages.

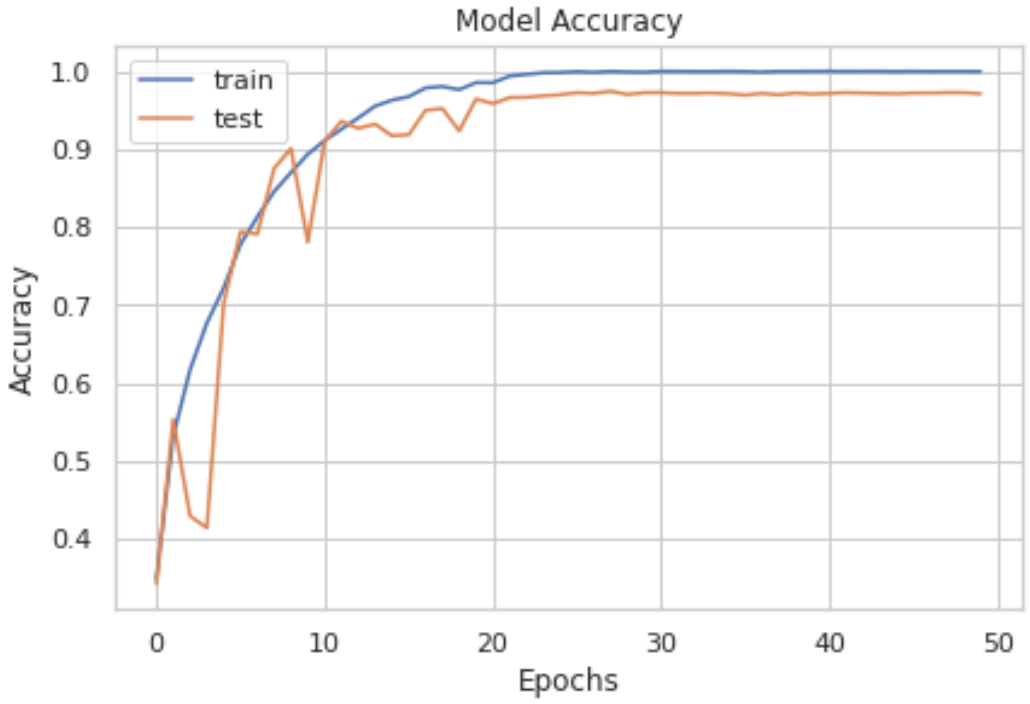


(a)

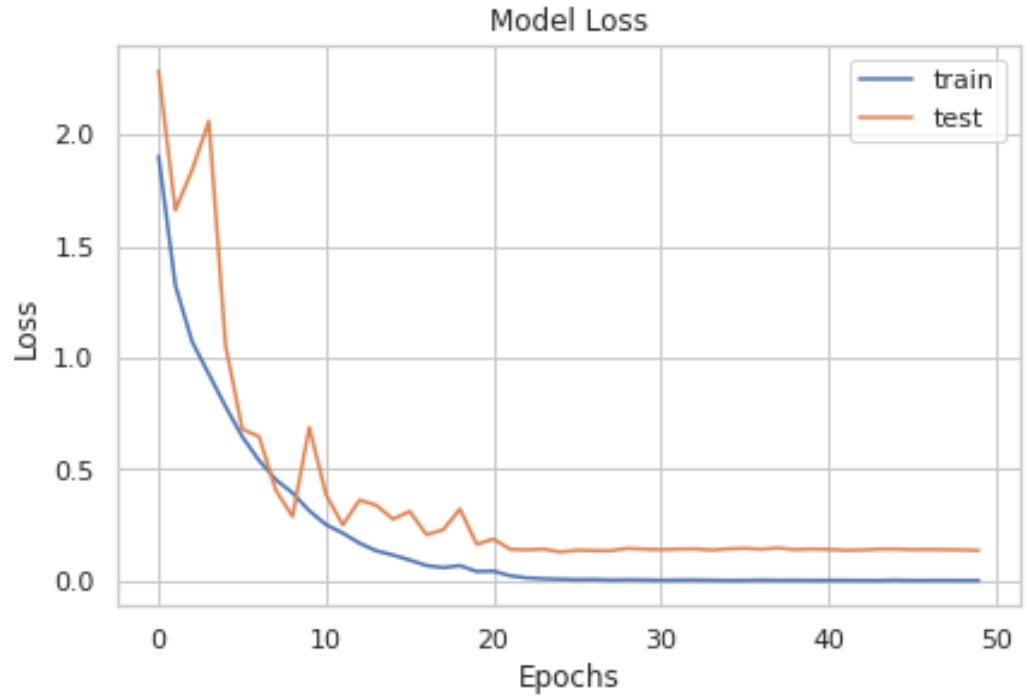


(b)

Figure 3. 8: (a) accuracy and (b) loss graph for food classification Model

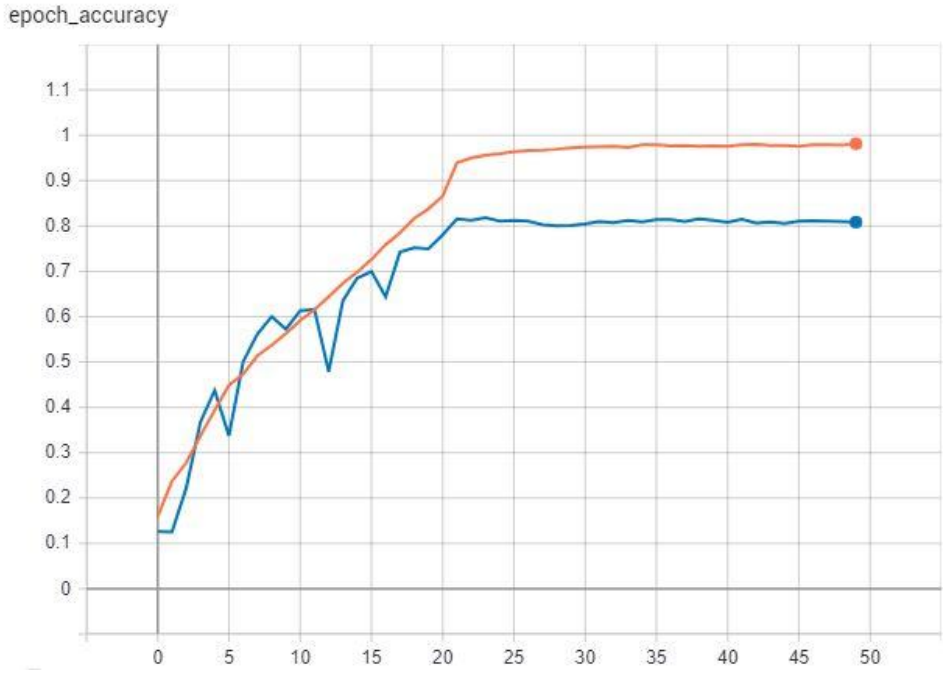


(a)

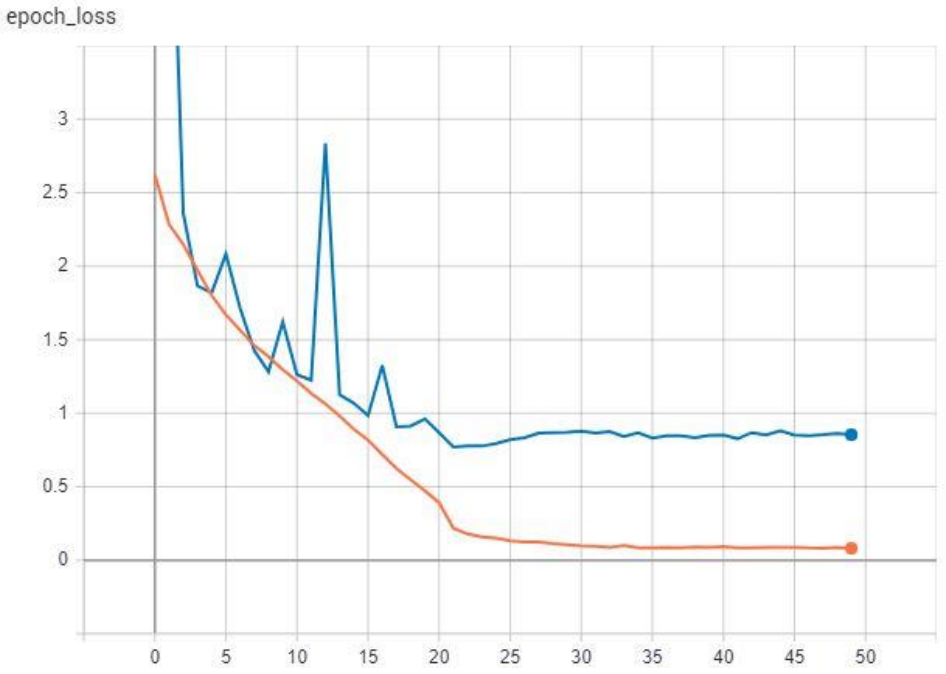


(b)

Figure 3. 9: (a) accuracy and (b) loss graph for outfit classification Model

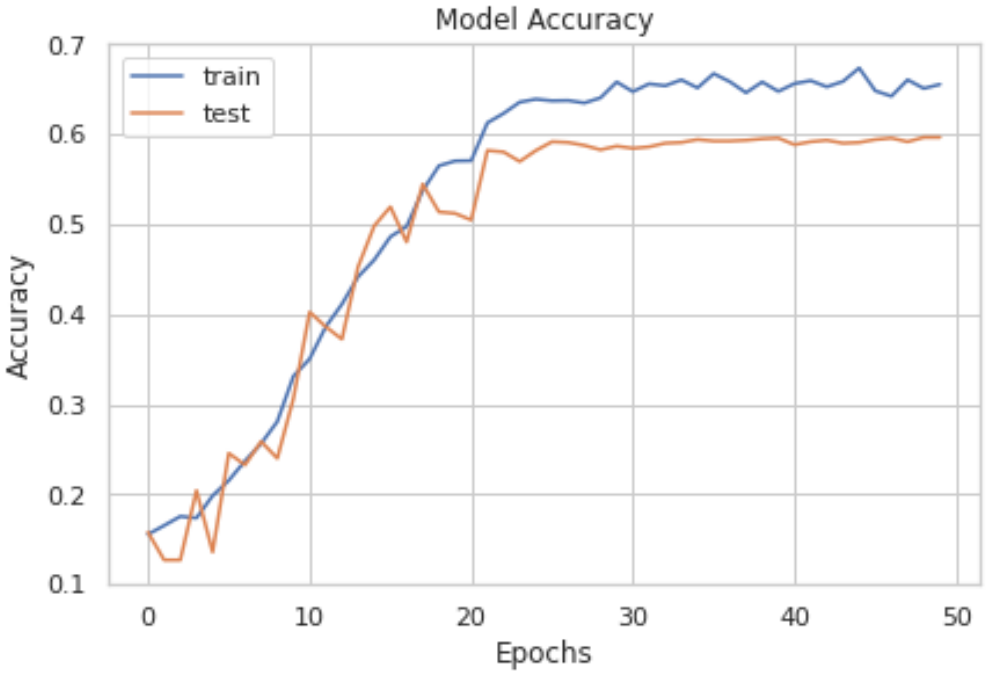


(a)

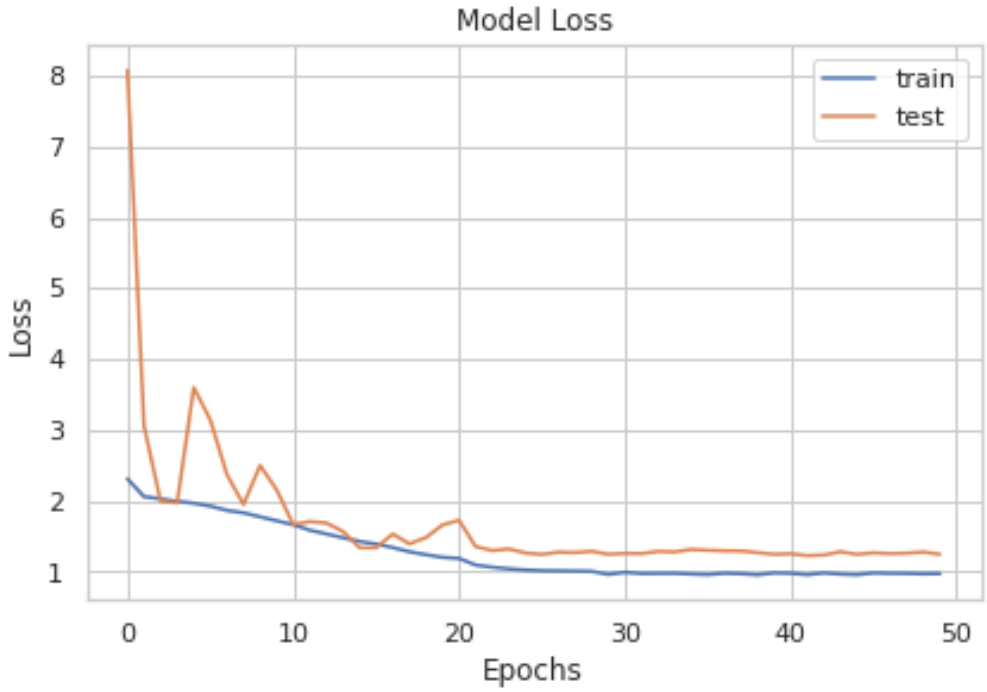


(b)

Figure 3. 10: (a) accuracy and (b) loss graph for home appliance classification Model



(a)



(b)

Figure 3. 11: (a) accuracy and (b) loss graph for vehicles classification Model

From training time history analysis, we have found some model overfits at the middle of training and it continues. To overcome this problem, we programmed this process in such a way that every model for each epoch should be downloaded so that we can use any model which does not overfit with test dataset. So, we have selected four models for image predication. To evaluates these model's testing accuracy, we have measured F1-Score from precision and recall of the test. F1-Score is the harmonic mean of precision and recall. We can find the precision and recall from these equations 8 and equation 9 respectively:

$$\text{Precision} = \frac{True\ positive\ result}{True\ positive\ result + False\ Positive\ result} \quad (8)$$

$$\text{Recall} = \frac{True\ positive\ result}{True\ positive\ result + False\ negetive\ result} \quad (9)$$

In these equations true positive result means where the model correctly predicts the correct objects. False positive is a result where the model incorrectly predicts the positive class and false negative result is the result where the model predicts the negative class incorrectly. After finding precision and recall value we can find the F1-Score easily from the equation 10 which is shown below:

$$\text{F1} - \text{Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

F1-Score result of four models are shown in Table 3.2.

	precision	recall	f1-score	support
bread	0.97	0.95	0.96	103
curry	0.95	0.97	0.96	155
fish	0.99	1.00	1.00	107
rice	0.97	0.96	0.96	134
vegetable	0.99	0.99	0.99	146
accuracy			0.97	645
macro avg	0.97	0.97	0.97	645
weighted avg	0.97	0.97	0.97	645

(a)

	precision	recall	f1-score	support
almari	0.79	1.00	0.88	126
bed	0.93	0.58	0.72	221
chair	0.85	0.81	0.83	106
disposal busket	0.71	0.67	0.69	180
door	0.90	0.57	0.70	257
jug	0.86	0.69	0.77	209
mug	0.69	1.00	0.82	102
refrigerator	0.89	1.00	0.94	209
sofa	0.86	1.00	0.93	198
switchboard	0.96	1.00	0.98	179
table	0.93	0.99	0.96	200
toilet	0.74	1.00	0.85	90
wash besin	0.70	1.00	0.82	100
accuracy			0.84	2177
macro avg	0.83	0.87	0.84	2177
weighted avg	0.85	0.84	0.83	2177

(b)

---

	precision	recall	f1-score	support
dress	0.96	0.96	0.96	342
lungi	0.93	1.00	0.96	100
panjabi	0.98	1.00	0.99	119
pants	0.98	0.95	0.96	363
sharee	0.99	1.00	1.00	101
shirt	0.99	1.00	0.99	356
shoes	0.97	0.95	0.96	479
suit	0.97	1.00	0.99	349
accuracy			0.97	2209
macro avg	0.97	0.98	0.98	2209
weighted avg	0.97	0.97	0.97	2209

(c)

	precision	recall	f1-score	support
bus	0.65	0.64	0.65	160
car	0.83	0.85	0.84	147
cng or battery rickshaw	0.89	0.65	0.75	209
motorcycle	0.86	0.80	0.83	203
rickshaw	0.70	0.85	0.77	199
train	0.61	0.82	0.70	131
truck	0.77	0.69	0.73	181
accuracy			0.75	1230
macro avg	0.76	0.76	0.75	1230
weighted avg	0.77	0.75	0.76	1230

(d)

Table 3. 2: F1-Score result of (a)food (b)home (c)outfit and (d) vehicles classification Model

From Table 3.2 we can conclude the accuracy of our final four model which will be used for image classification. Our food classification model gets 97%, home appliance model gets 84%, outfit classification model gets 97% and vehicles classification model gets 75% test accuracy.

The result of image prediction of four models from python programming simulation is given below:

```
[ ] predictions, probabilities = prediction.predictImage(a, result_count=3)
    frame=cv2.imread(a)
    for eachPrediction, eachProbability in zip(predictions, probabilities):
        print(eachPrediction , " : " , eachProbability)
    c = str(predictions[0]) +" : "+ str(probabilities[0])
    n = str(predictions[1]) +" : "+ str(probabilities[1])
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(frame,c, (10, 50), font, 0.9, (0, 255, 255), 1, cv2.LINE_AA)
    cv2.putText(frame,n, (10,90), font, 0.9, (0, 255, 255), 1, cv2.LINE_AA)
    cv2_imshow(frame)
```

```
WARNING:tensorflow:When passing input data as arrays, do not specify `steps_per_epoch`
shoes  :  99.99101161956787
hoodie  :  0.0058924404584104195
dress  :  0.00185509543371154
```

Figure 3. 12: Simulation setup for predicting images from device

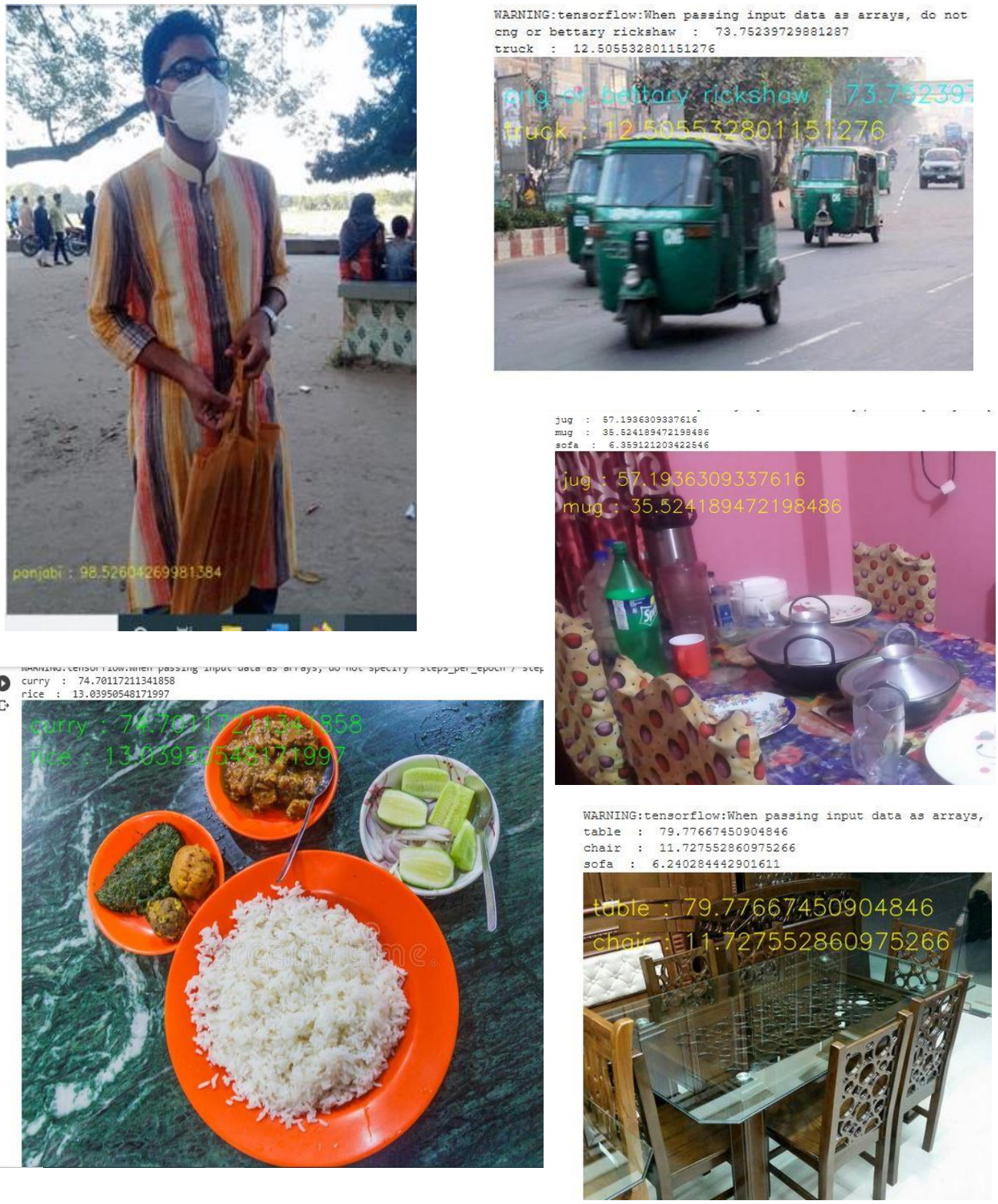
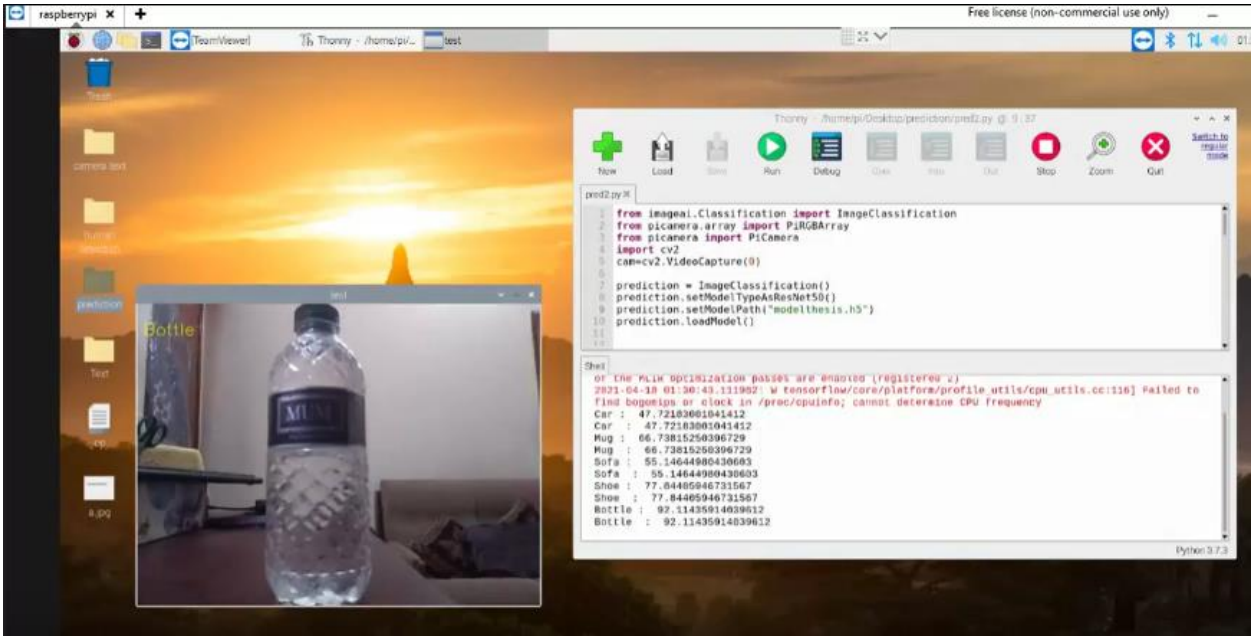


Figure 3. 13: Simulation result of some predicted images in terms of Bangladesh

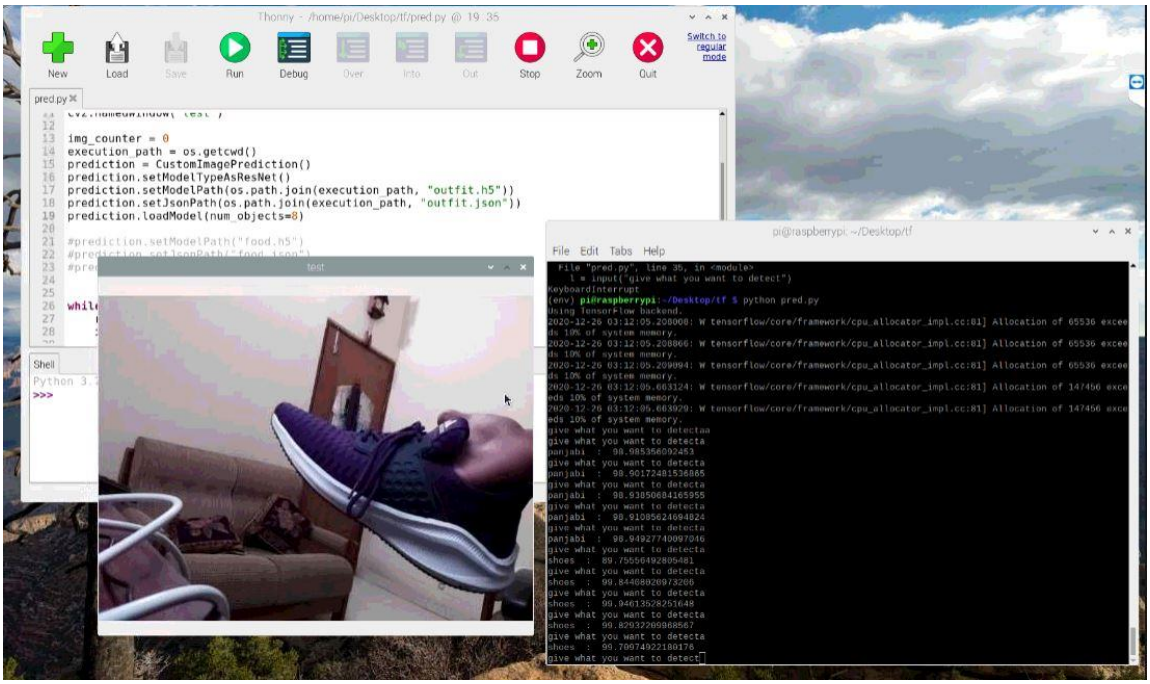
Figure 3. 14: Simulation result of road transport detection in Dhaka City

3.7 Test Run

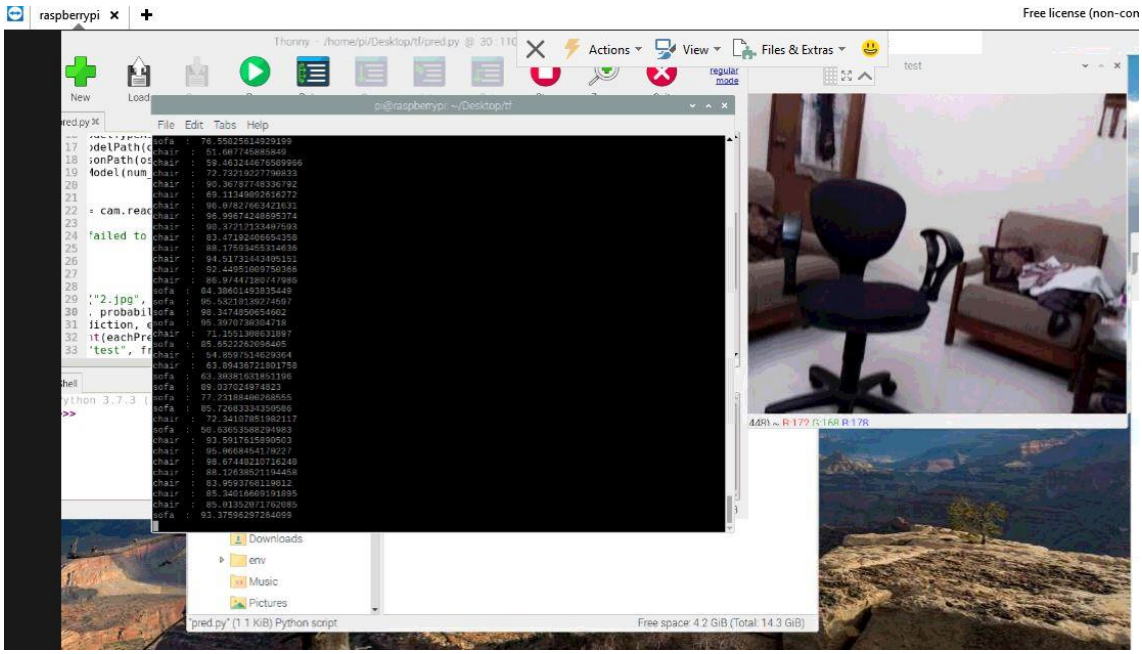
As simulation works successfully, we have implemented the whole thing in Raspberry Pi 4 for real-time testing and find it works effectively in on board computer. However, because of processing power image prediction becomes slow and Raspberry Pi become overheat. To solve this, we have added a cooling fan and problem gets solved. Raspberry Pi testing result in real time situation is given in figure 3.15.


raspberrypi
Free license (non-commercial use only)
Thonny - /home/pi/Desktop/prediction/pred2.py @ 9 : 37
New
Load
Run
Debug
Stop
Zoom
Quit
pred2.py
from imageai.Classification import ImageClassification
from picamera.array import PiRGBArray
from picamera import PiCamera
import cv2
cam=cv2.VideoCapture(0)
prediction = ImageClassification()
prediction.setModelTypeAsResNet50()
prediction.setModelPath("modelthesis.h5")
prediction.loadModel()
Shell
Car : 47.72183001041412
Car : 47.72183001041412
Mug : 66.73815250396729
Mug : 66.73815250396729
Sofa : 55.14644980430603
Sofa : 55.14644980430603
Shoe : 77.84405946731567
Shoe : 77.84405946731567
Bottle : 92.11435914039612
Bottle : 92.11435914039612
Python 3.7.3
test
Bottle

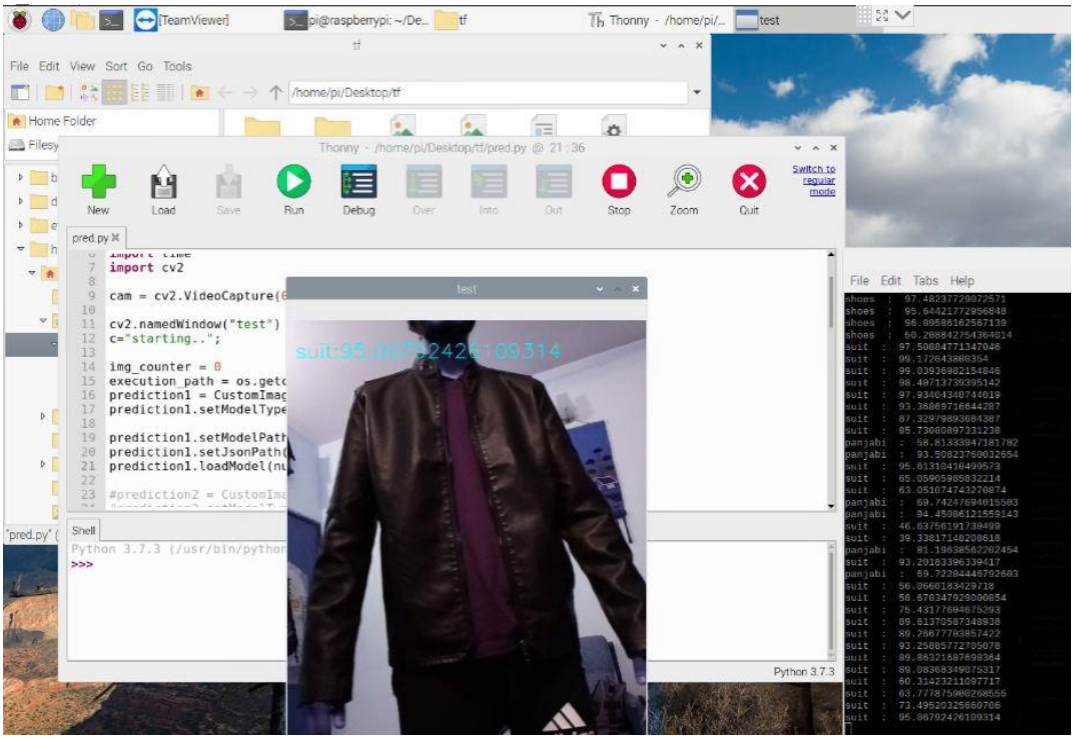
(a)


Thonny - /home/pi/Desktop/tf/pred.py @ 19 : 35
New
Load
Save
Run
Debug
Over
Into
Out
Stop
Zoom
Quit
Switch to regular mode
pred.py
img_counter = 0
execution_path = os.getcwd()
prediction = CustomImagePrediction()
prediction.setModelTypeAsResNet()
prediction.setModelPath(os.path.join(execution_path, "outfit.h5"))
prediction.setJsonPath(os.path.join(execution_path, "outfit.json"))
prediction.loadModel(num_objects=8)
#prediction.setModelPath("food.h5")
while
Shell
Python 3.7
>>>
test
pi@raspberrypi: ~/Desktop/tf
File Edit Tabs Help
File "pred.py", line 35, in <module>
i = input("give what you want to detect")
KeyboardInterrupt
(env) pi@raspberrypi:~/Desktop/tf $ python pred.py
Using TensorFlow backend.
give what you want to detecta
give what you want to detecta
panjabi : 98.985356092453
give what you want to detecta
panjabi : 98.90172481536865
give what you want to detecta
panjabi : 98.93050884165955
give what you want to detecta
panjabi : 98.91085624694824
give what you want to detecta
panjabi : 98.94927740097046
give what you want to detecta
shoes : 89.75556492805481
give what you want to detecta
shoes : 99.84408020973206
give what you want to detecta
shoes : 99.94613528251648
give what you want to detecta
shoes : 99.82932209968567
give what you want to detecta
shoes : 99.70974922180176
give what you want to detect

(b)



(c)



(d)

Figure 3. 15: Realtime Camera Testing of Image classification

Chapter 4

Text Detection and converting it to Speech

4.1 Purpose of the feature

In our daily life we have to read many things, similarly, a partially sighted person has to read such texts as well. We have included this attribute to our smart glasses for enabling the visually impaired individual to be able perceive the words he sees by hearing, even if he is unable to read it.

4.2 Algorithm

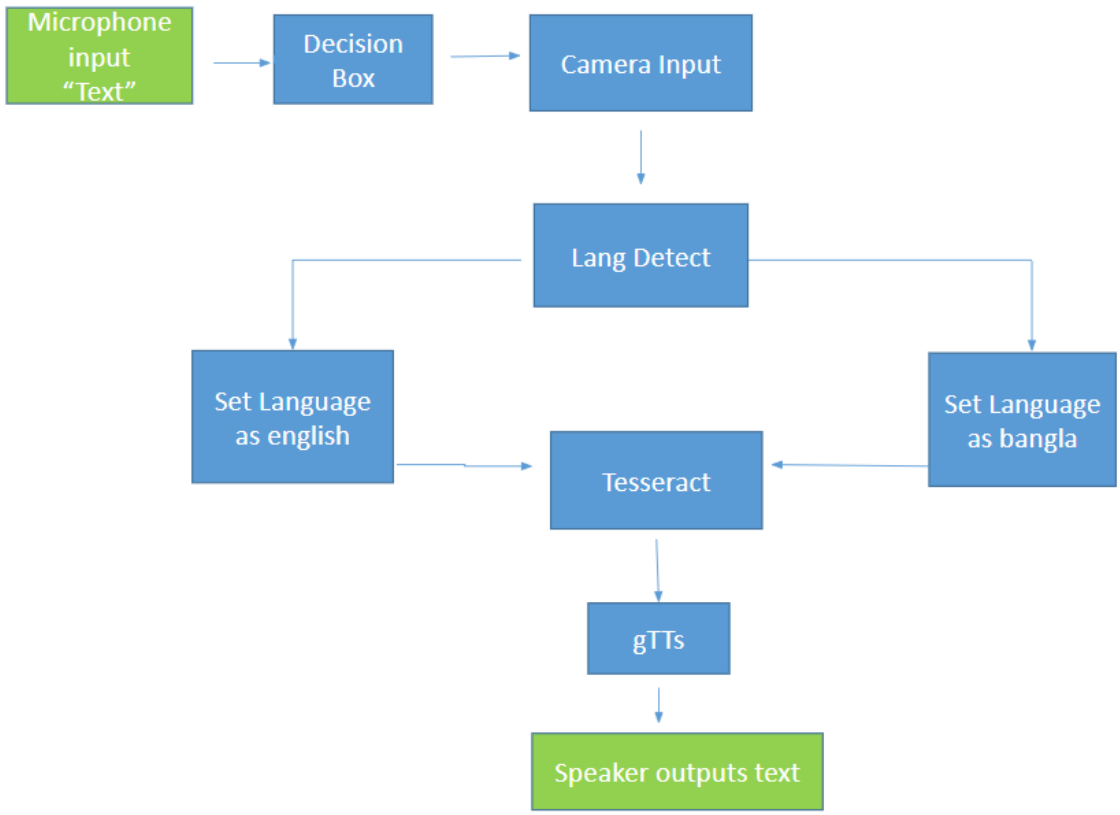


Figure 4. 1: Flowchart of the text detection algorithm

Algorithm 1 shows the process for optical character recognition systems. It is very much straight forward and can be separated into two parts. For recognizing text from image tesseract OCR engine has been used. Another part of the algorithm detects language, reading the text for the user correctly. The text recognition process starts by image processing by the optical character recognition algorithm, that inputs the region containing the images or other noises with the text which need to be recognized. After recognizing OCR, the engine returns a raw text. Then the text is forwarded for language detection to determine whether language is Bangla or English. After detecting language text are made correction using another language processing algorithm and recover sentences if needed. Finally, the result is converted to audio using the text-to-speech algorithm.

4.3 Modules Used

Google Text-to-speech (gTTs) - It is a tool used to interface with Google Translate's text-to-speech API. It can read out unlimited lengths of text. It can read more than thirty languages.

Langdetect - Helps recognise in which language the text is written in. It can recognize 55 different languages.

Speech recognition- it relies on the Hidden Markov Model (HMM) approach. It works on the assumption that a speech signal, when viewed on a short enough timescale. The output of this approach results in many small vectors. With these groups of vectors, phonemes are matched. Phonemes are fundamental units of speech. Phonemes vary from one person to another, so training

is required. Particular algorithm is applied to determine the best suited word that that phoneme will produce.

Tesseract- It is an open source text recognition (OCR) engine. It is mainly used for extraction of text from images. It is a multistep process with steps –

1. Word finding
2. Line finding
3. Character classification

Texts are organized into blobs, then the regions for proportional text are analysed. The lines of text are broken down into words is perspect of the character spacing. Each word is attempted to be recognized by turn. If that can be accomplished then it is passed on to adaptive classifiers for a higher accuracy.

In the Modernization tool the input image is processed in boxes line by line. This is fed to the LSTM model. Additionally, it can be used to recognize text from an image of a single text line. It is trained with several stacks of data with different fonts. But still Tesseract is limited to handwritten pieces of text.

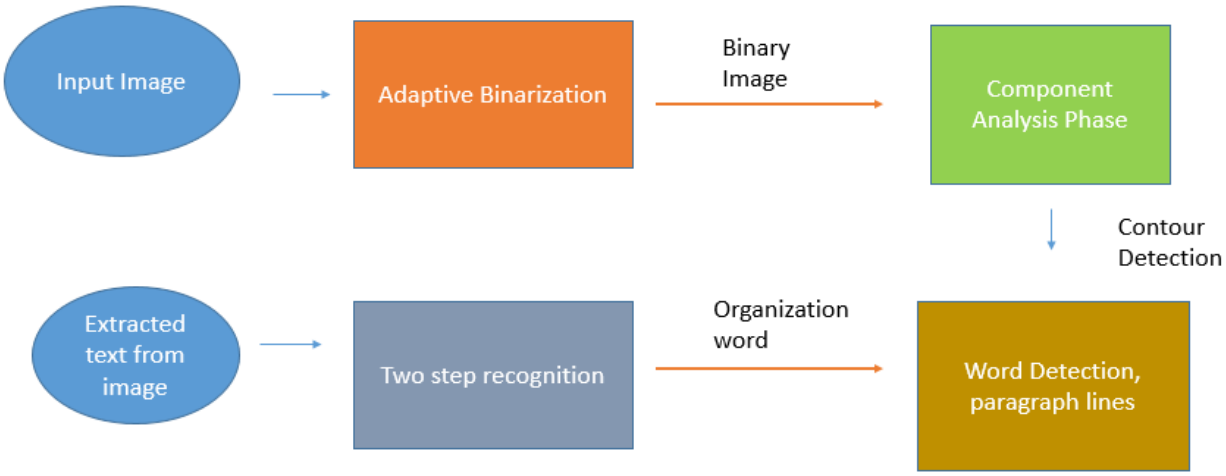


Figure 4. 2: Working Flow Diagram of Tesseract OCR

4.4 Text Detection Outputs and Discussion

We have categorized our text detection into 3 parts –

1. Detecting English texts
2. Detecting English Newspaper
3. Detecting Bangla texts

4.4.1 Detecting English Texts

Any English text we place in front of our camera can easily be detected and read out through audio output. Multiple texts have been tested and the output was successfully obtained with much accuracy. Afterwards, the text extracted from the image was successfully converted to speech and read out.

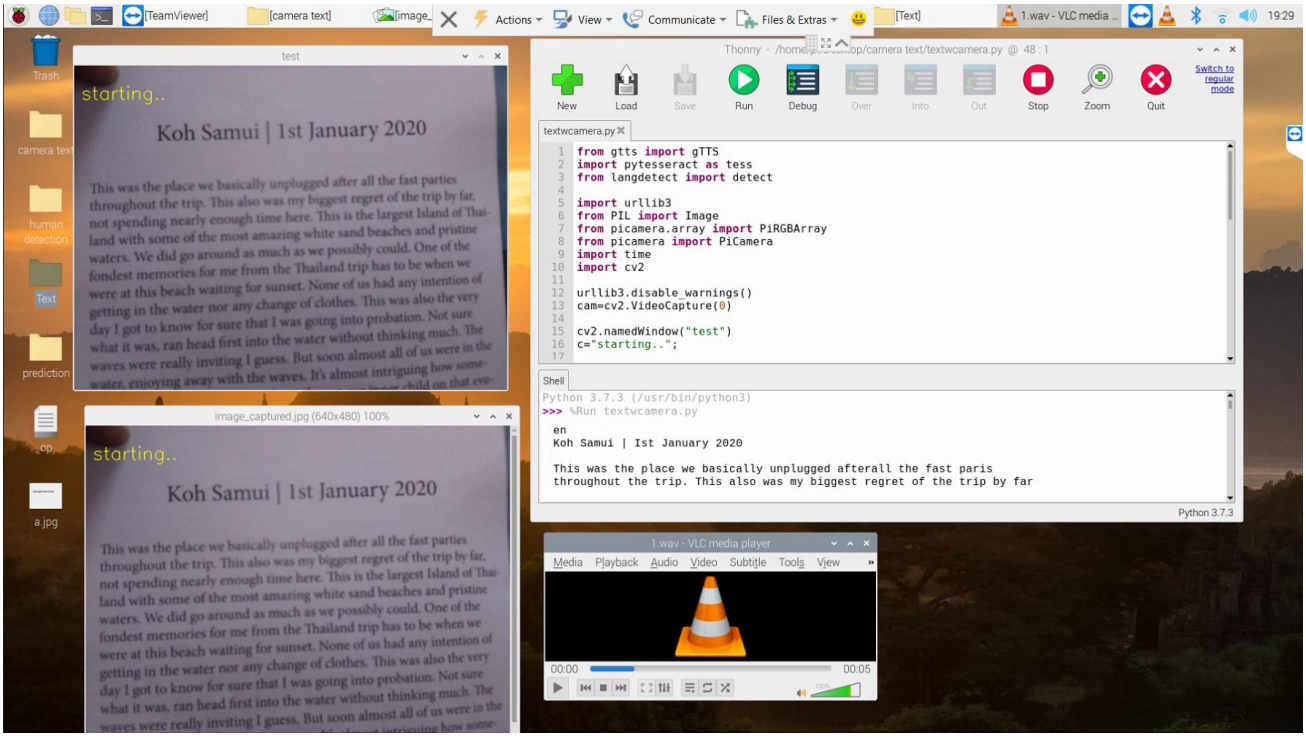


Figure 4. 3: Detecting English Text captured live through Camera

4.4.2 Detecting English Newspaper

Previously, in the text detection part of normal English speech, the camera was successful to extract the words from the live captured image, but in case of newspaper detection as the newspaper article writings are much smaller in fonts and congested, therefore due to our limitations and low resolution in the raspberry pi camera, the live captured image from newspaper article could not be extracted with much accuracy. But, to test our newspaper text detection, we have downloaded a high-quality resolution newspaper article image and tried to extract the text from that newspaper and finally converting it to speech. Afterwards, it was seen that the downloaded newspaper article was successfully detected and was also converted to speech with greater accuracy. Here, in the newspaper as the texts are arranged column wise, it was also observed that the texts were extracted column wise without creating any collision between two columns along with headings and sub-headings. So, we came to a conclusion that, if we had a high-resolution pi

camera, we could also successfully detect newspaper article by capturing live images and finally covert it to speech.

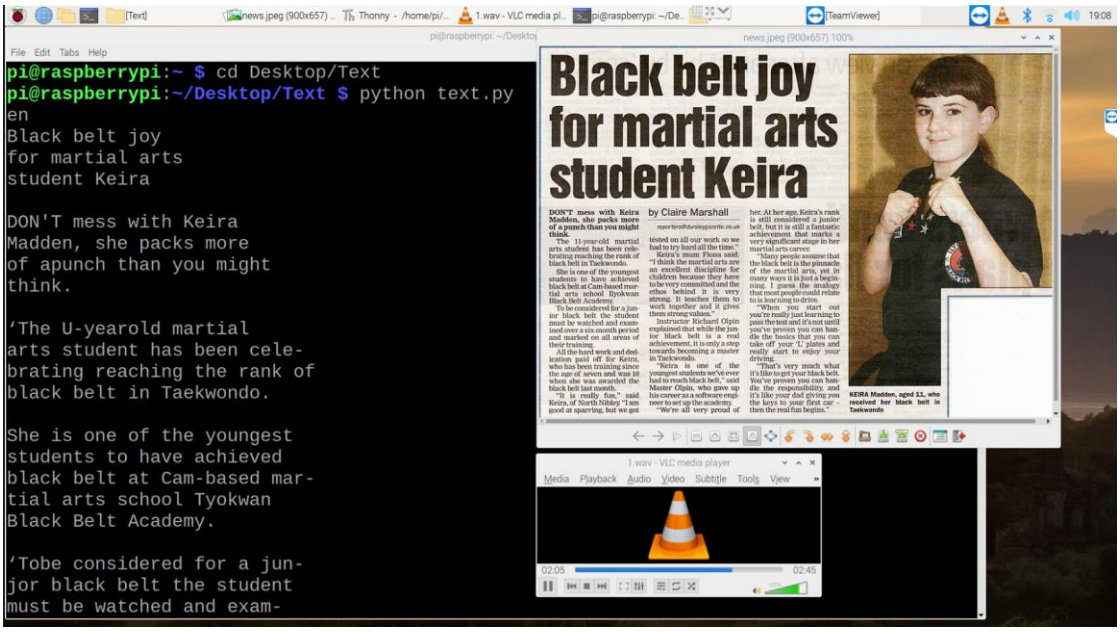


Figure 4. 4: Text to speech detection of English Newspaper

4.4.3 Detecting Bangla Text

In case of Bangla text detection, expected accuracy was not obtained like English text detection. As the open source model which we used for detecting Bangla text was not well trained. Moreover, for getting the outmost accuracy, the Bangla image need to be scaled appropriately with much image contrast as much as possible. Additionally, the text must be horizontally aligned and noise cancellation in the image must be ensured as much as possible. Besides, like newspaper article detection, high quality resolution camera must be assured. Also, it would be better if we could make a custom model for Bangla text detection. Training such models would require a large variation of datasets of Bangla language. Only then, live captured Bangla text can be extracted and detected. But we have tried with a clear simple Bangla text and saw that the text was detected successfully and finally converted to speech.

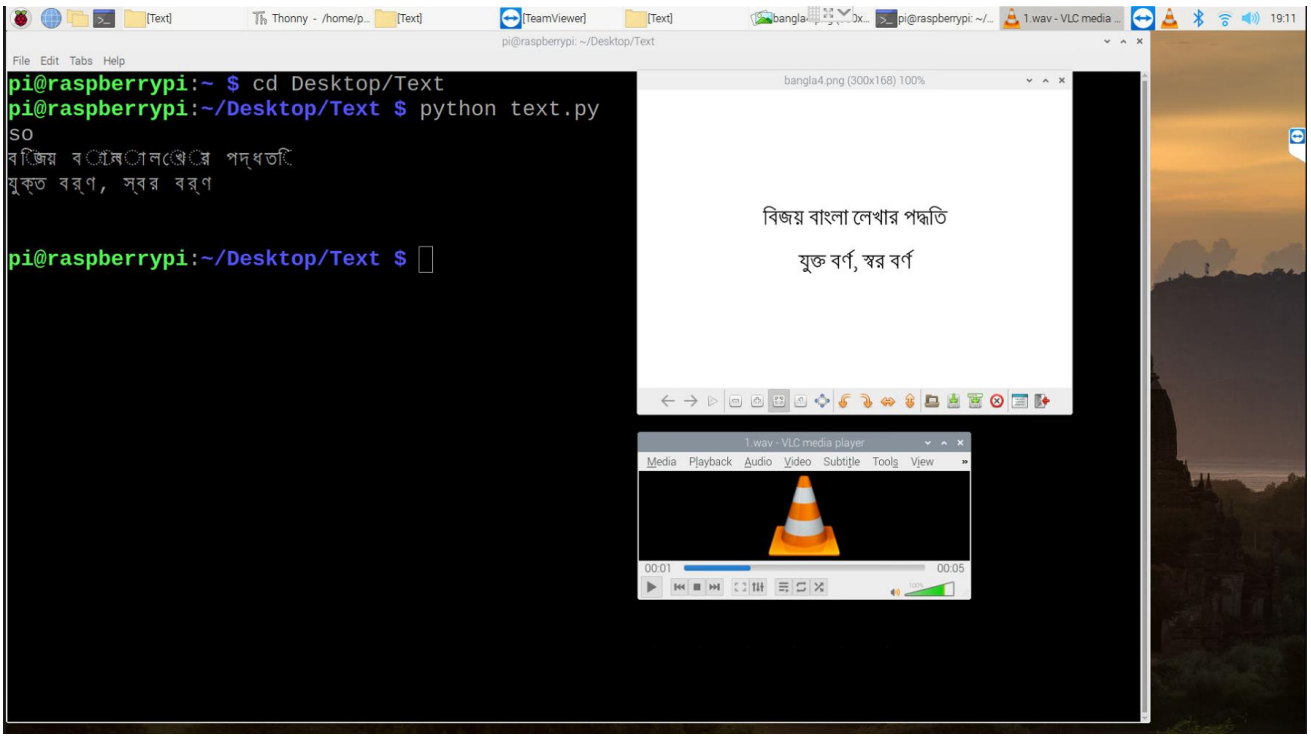


Figure 4. 5: Text to Speech detection of Bangla text

Chapter 5

Home Automation

5.1 Introduction:

One of the largest problems for blind people is to control devices and electrical appliances in home-like light, fan, etc. Using a switchboard or remote is not an efficient solution for blind people even though they can use an object detection method to detect objects. In the modern era of life-controlling devices are now easier. Anyone can control devices through voice commands. However, they are costly and in Bangladesh, almost every device is manually controlled. Moreover, blind persons are not aware of their environment. He may able to detect objects with help of smart glass but not able to know the problem occurring behind him or in another room. It can cause an accident and a person can be injured badly.

5.2 System overview:

So, we have proposed a home automation system where a blind person will able to control any home devices or gadgets through voice command. Input can be sent through the mic attached to spectacles. Sensor setup in the home such as flame sensor, gas sensors may help the visually impaired person and concern person to avoid any accident before it occurs. To control devices and reading data from sensors NodeMCU ESP32 development boards have been introduced. This is a low-cost IoT platform that uses a 32bit ESP32 microcontroller with WIFI and Bluetooth (BLE) capabilities. This microcontroller will directly communicate with smart glass and mobile app through firebase to Store & sync data in real-time. In the ESP32 microcontroller, 34 pins can be used as GPIO and the remaining are input-only pins. Moreover, it has eighteen channels for 12-bit

analog to digital conversion and two channels for 8-bit digital to analog conversion. It also has sixteen channels that can generate PWM signals [23]. Another advantage of this microcontroller is we can easily program this microcontroller using Arduino programming language. Addons and library for ESP32 microcontroller in Arduino allows Arduino IDE to compile and upload the program in NodeMCU development board. In Bangladesh Arduino is cheapest and most popular tool for programming and working with microcontrollers. There are different ways to power up the development board but as we have 5V sensors and relay so it will be better to power up the microcontroller with a 5V power supply.

1. **Micro USB:** Connect the mini USB 5V phone charger through a cable it will draw power required for the board to function

5V or VCC pin: 5V power can be supplied with a Regulated 5V source to board 5V pin, this voltage will be converted to 3.3V by internal voltage regulator as ESP32

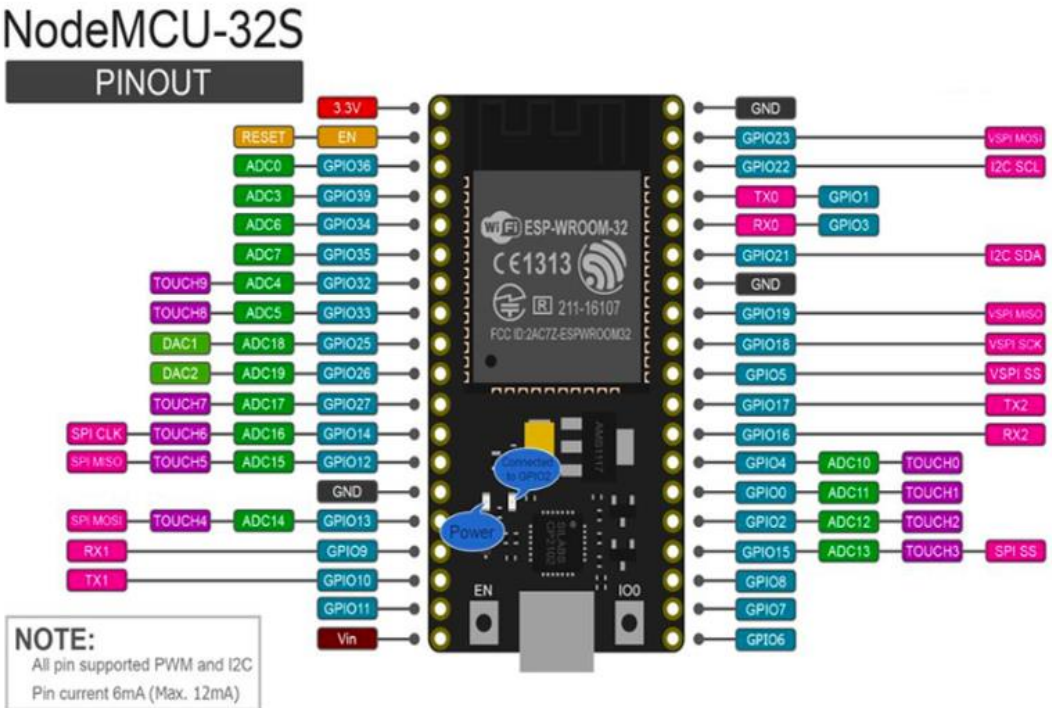


Figure 5. 1: NodeMCU ESP32 Development Board

5.3 Home appliance Control:

All the devices will be controlled by a 5V relay module array which will be operated by a microcontroller according to voice command. The relay module circuit for controlling one device is given in Figure 5.2.

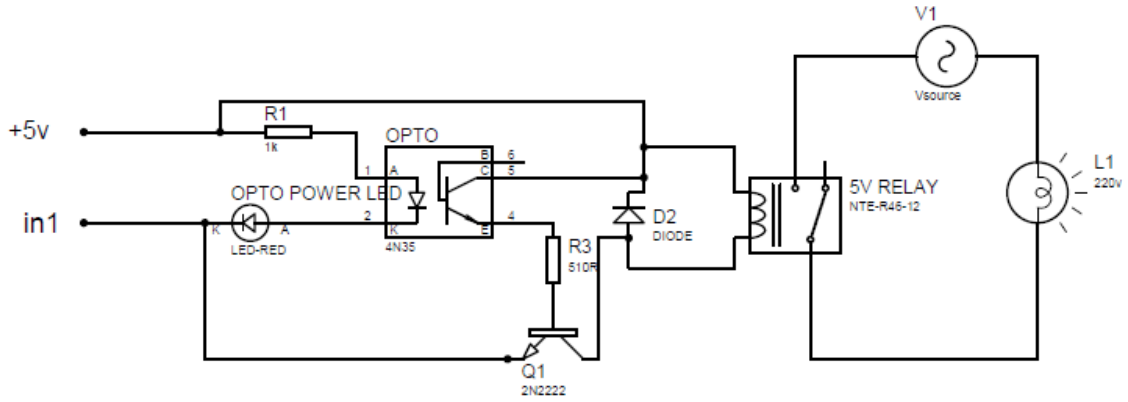


Figure 5. 2: Relay control circuit

In the schematic +5V pin is used as a power supply of the relay controller we can call it VCC pin also. It can be given from the microcontroller 5V voltage regulator. In1 pin is a triggering terminal for the relay. The output side of the relay has three pins labeled with NC (normally closed), COM(Common), and NO (normally open). Normally closed pin usually become short-circuited with a common pin. If the relay coil magnetizes, the common pin becomes short-circuited with a normally open pin. If a signal is given from the microcontroller on the input side relay will be triggered and the lamp will be powered up as from figure 5.2, we can show that one part of AC circuit is connected with a common pin and another is normally an open pin. On the input side, an optocoupler is used to isolate the grounds as well as remove noise between the microcontroller signal circuit and the relay load. Flyback diode eliminates sudden voltage spikes across the relay coil. NPN transistor helps the relay to magnetize properly as a signal from the microcontroller is

given in a range of 3.3V and it has a current limit. On the other hand, the relay operates at 5V. When transistor base voltage is zero transistor acts as an open circuit and the relay does not trigger. If an amount of current flows through the transistor base it goes in saturation mode and current starts to flow from collector to emitter and the triggered relay can switch on the ac device. Relay specifications [26] we have proposed for operating home appliance is given in table 5.1.

Trigger Voltage (V)	**Trigger Current (mA)**	**Max. AC Operating Voltage (V)**	**Max. AC load current (A)**	**Operating Time (mSec)**
5	71	250/125	10	10

Table 5. 1: Relay specification for controlling home appliances

Figure of proposed 5V relay module is given in figure 5.3:

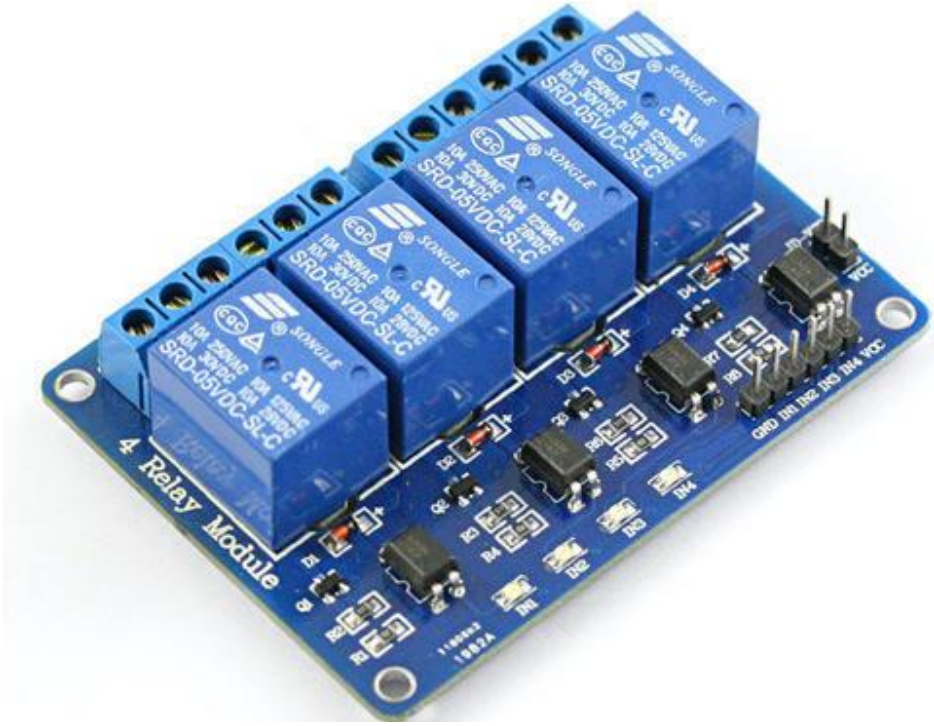


Figure 5. 3: 5V 4 channel relay module able to control four ac devices

5.4 Sensor Setup

We can collect sensor reading using a microcontroller to analyze environmental situations. For this, we have used two sensors: a gas sensor and a flame sensor. The Gas sensor will give the update of gas leakage, air quality, or alcohol concentration in air at home and the flame sensor can detect fire if any accident occurs.

5.4.1 Gas sensor:

Gas sensor basically detects the gas or other air elements. These types of sensors are highly used for air pollution or gas leakage detection. For gas leakage detection, we have used an MQ-6 series gas sensor. MQ-6 is a Metal Oxide Semiconductor type module that detects LPG, butane gas, alcohol using the voltage division rule. When it detects gas particles resistance of the sensor changes accordingly [27]. All MQ sensors work in a similar fashion. This sensor has four pins for powering and sending signals. It can work both as an analog and digital sensor. For operating the sensor voltage (VCC) needs to be in the range of 3.3V to 5V. GND pin needs to be connected with the ground of the microcontroller circuit. For detecting gas leakage here, we have used an analog pin and send the analog reading to the microcontroller as we wanted to set the threshold value using microcontroller. Each gas sensor output gives an analog value between 0 to 4095. To convert this reading into voltage, we have used equation (1).

$$\text{Sensor Voltage } (v_o) = \text{sensor value} * (\text{input voltage} / 4095) \quad (1)$$

All MQ series modules have a built-in analog to digital converter circuit which consists of LM393 dual comparator IC. In that case, the user needs to set the threshold value using a potentiometer

and the digital output pin should be connected with a microcontroller instead of an analog one. When estimating gases, the term concentration is utilized to portray the measure of gas by volume noticeable all around. The two most basic units of estimation are ppm and percent concentration. PPM is the proportion of one gas to another. MQ-6 sensor can detect the gas concentrations anywhere from 200 to 10000 ppm. If we want to detect smoke or other air particles like CO, CO_2 we will have to use another model of the MQ series like MQ7 or MQ6. By using the sensor, we can also know the gas concentrations from the app which is built for helping blind people.

We have chosen MQ series gas sensors as it is cheap and readily available in Bangladesh as well as it easily detects gas leakage and other element's concentrations in air. Table 5.2 shows which MQ sensor modules detect which gases and aerosols mostly [27].

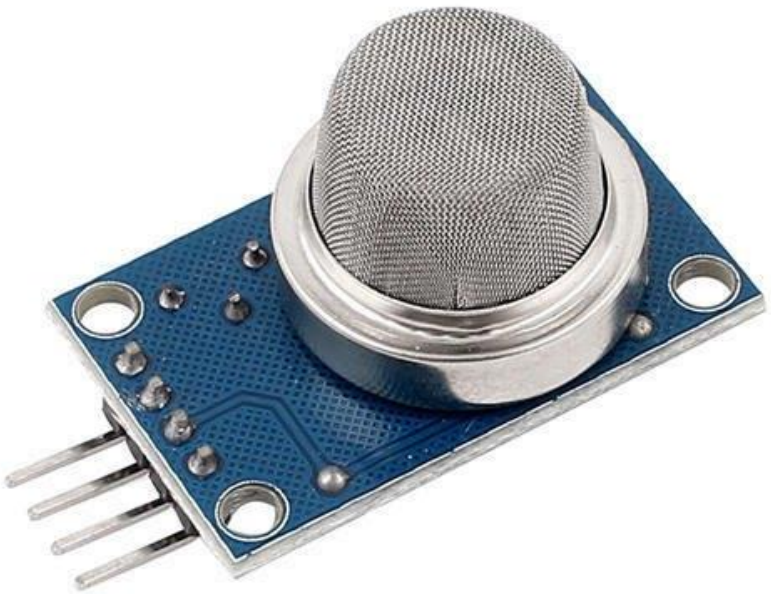


Figure 5. 4: MQ series gas sensor module

Sensors	Most Detecting Elements
MQ2	Methane, Butane, LPG, smoke and flammable elements
MQ3	Alcohol, Ethanol, smoke.
MQ4	Methane, CNG **Gas**.
MQ5	Methane, Natural Gas, LPG.
MQ6	Methane, LPG, butane
MQ7	Carbon Monoxide, Hydrogen Gas
MQ8	Hydrogen Gas
MQ131	Ozone
MQ135	Ammonia gas, benzene, ethyl alcohol and carbonic dioxide
MQ136	Hydrogen Sulfide gas
MQ138	Benzene, Toluene, Alcohol, Acetone, Propane, Formaldehyde gas, Hydrogen
MQ214	Methane, Natural gas

Table 5. 2: MQ series sensor's detected gas elements

5.4.2 Flame Sensor:

For detecting the flame, we have used a flame sensor. A flame sensor is a type of detector designed primarily to detect or respond to a fire or flame. The response to fire detection can rely on its suitability. Actually, this sensor is most responsive to normal light that's why this module is used as flame alarms. This sensor detects flames from the light source in the range of 760 nm to 1100 nm [28]. The high temperature can quickly destroy the sensor. For this, the sensor needs to be placed at a minimal distance from the flame. The observation of flames is possible from a distance of 100cm and the maximum detection angle is 60^{o} [28]. If the flames are lighter in 0.8m, the fire detector may be triggered, if the fire intensity is high, the distance detection will be enhanced. The

flame sensor has 4 pins like gas sensors and they are VCC, Ground, Analog output pin & Digital Output Pin. The operating voltage for this sensor is 3.3 V to 5V. For detecting flame from the sensor, we have used analog pin. We use the flame sensor to detect flame as it gives a very quick response to flame, also responsive to the flame range, has high photosensitivity.

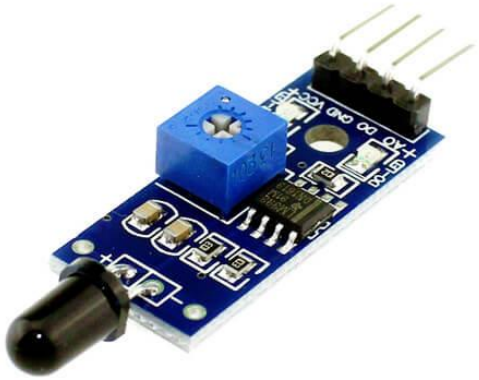


Figure 5. 5: Flame Sensor

5.5 Test Setup and discussion

To simulate future products and for testing the correctness of our design we have come up with a prototype setup as shown in figure 5.6.

Here we have used NodeMCU ESP32 microcontroller to control the whole process and a two-channel 5V relay module for controlling ac devices. For testing purposes, we have chosen an AC-LED lamp and table fan as AC load. A MQ6 gas sensor and a flame sensor are used for reading sensor data and give feedback.

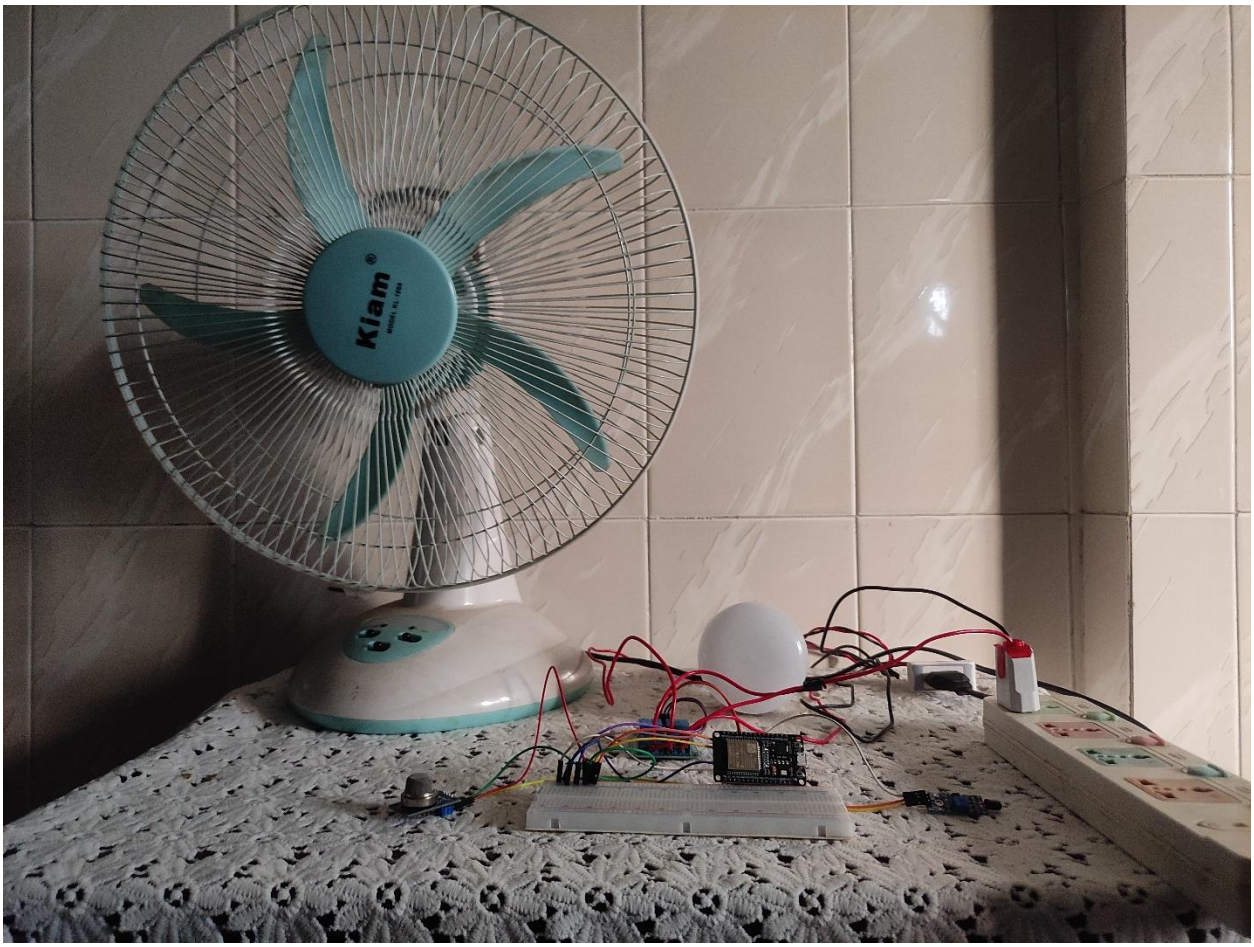


Figure 5. 6: Home automation prototype test setup

Firstly, voice command has given through mic using raspberry pi. To control devices, commands are given like- “Turn on Light”. Here On can be Off and Instead of light we can tell Fan. We have used some programming-based string operations to analyze the sentence and extract the actual command. This command directly writes to the firebase real-time database. From the microcontroller sides, it always read data from the firebase console. When it finds data has been changed in the database it operates those devices accordingly. Likewise, it reads data from the gas sensor and flame sensor continuously. When the microcontroller detects reading from any sensors becomes more the threshold value. It sends this data to the firebase database which can be read from a glass or mobile app for taking proper actions. Changing the value of the sensor was not

easy because working with fire and gas directly is not a good idea. So, we have used perfume that contains alcohol to test the gas sensor and has used gas lighter to test the flame sensor. Controlling light and fan using smart glass is showing in figure 5.7.

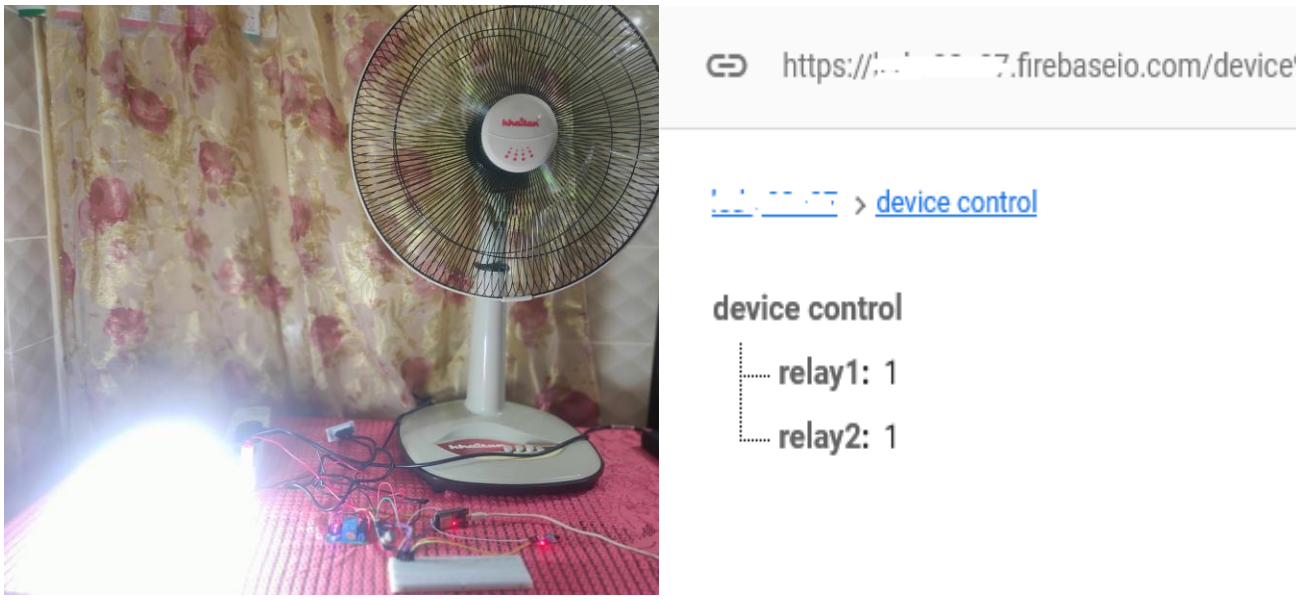


Figure 5. 7: Controlling devices by changing data of database though voice command

While testing data we also observed the sensor reading and relay operation through Arduino serial monitor which has been shown in figure 5.7. It ensures everything is working properly or not. The real-time simulation also helps to debug the code and find the fault of hardware. At the end of the test run, we have found that the whole process needs some time to operate the device after giving the voice command and update sensor data to the database. This is because of firebase database performance and internet latency. Figure 5.8 shows how flame sensor works and figure 5.9 show how microcontroller send data to firebase real-time database when fire has detected.

Circuit setup is a bit risky as it has an AC 220V connection and any loose connection can be a cause of an accident. To avoid this, we have taken some measures. We have used an external multiplug so that circuit becomes isolated from the mainline till the multiplug switch turns on. At

the joining part of two wires, we have ensured insulation by using insulation tape. Safety glasses and gloves are used for extra protection.

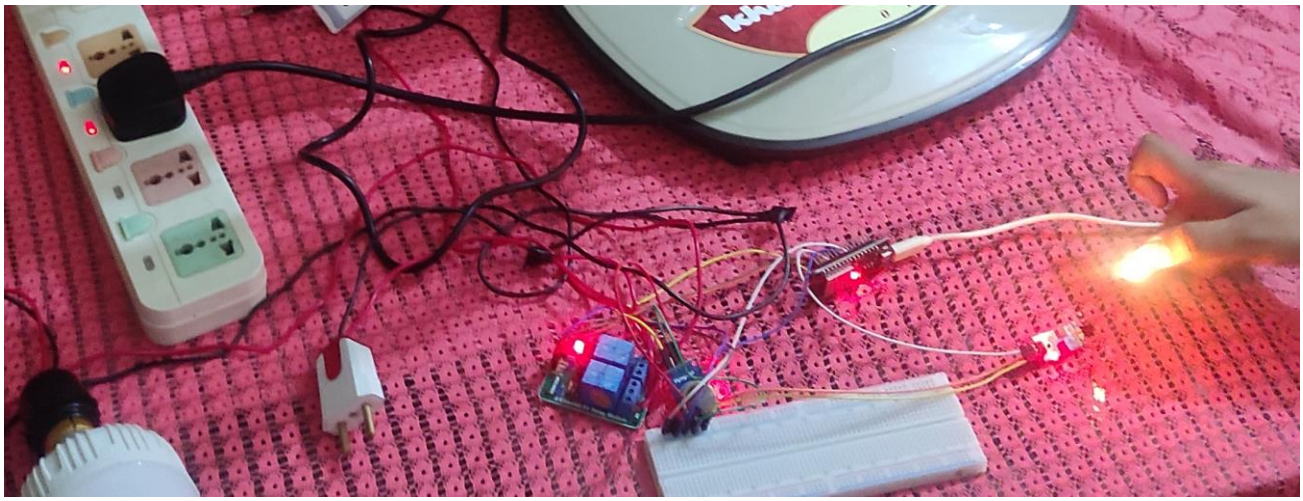


Figure 5. 8: Sensor testing of prototype design (Flame Sensor)

Figure 5. 9: (a) Arduino Serial Monitor shows fire is detected (b) Fire alert is updated in database

Chapter 6

App Development for Helping Caregiver

A visually impaired person has to face a lot of difficulties in daily life. The app, named “AI Smart Glass APP (Thesis)” will make their life a little bit easier in times of need also for the victim’s close ones who always remain tensed about them.

6.1 Working Principle:

For developing the app, we have used flutter language. Flutter which is developed by Google, an open-source mobile UI framework user interface development kit. From a single codebase, it is possible to create apps for Android, iOS, Linux, Windows, etc [29]. This means two separate applications using the same programming language and codebase (for iOS and Android). Flutter contains two main components: A Software Development Kit (SDK) which is a collection of resources that will aid in the creation of applications also includes compilers and tools for compiling the code into native machine code (for iOS and Android), A Framework (UI Library built on widgets) which is a list of the UIs can use to personalize as per the needs (buttons, text inputs, sliders, etc.) [30]. To build flutter app, dart [31] is used as a programming language that mainly focuses on the progress of the front end which can be used by smartphone and web apps. The app is connected with the firebase which is a NoSQL storage software that saves data in JSON documentation. It offers a wide range of tools and resources that support developers in developing quality software, developing a user base and benefit. It is constructed on the infrastructure of Google [32]. Real-time database is one of the features of the Firebase which is used in this app as it is a cloud-hosted database. Its data will be stored as JSON and sync to each linked device in real-time. It helps to create rich, interactive applications by providing safe access to the database

directly from client-side code. Data persist locally, and real-time activities begin to start, providing end-users a reactive experience even though they are offline. The real-time database synchronizes the local modifications with the remote updates that have happened when the system is offline, which automatically merge any discrepancies when a computer regains a link. Both the clients share a Realtime Database instance of the latest updates as they create cross-platform applications with iOS, Android.

In this app, there are features like a real-time status update at emergency, location tracing, emergency contacts and calling service, audio and text chat, medicine reminder, etc. which are not only helpful for the visually impaired people but also for their relatives.

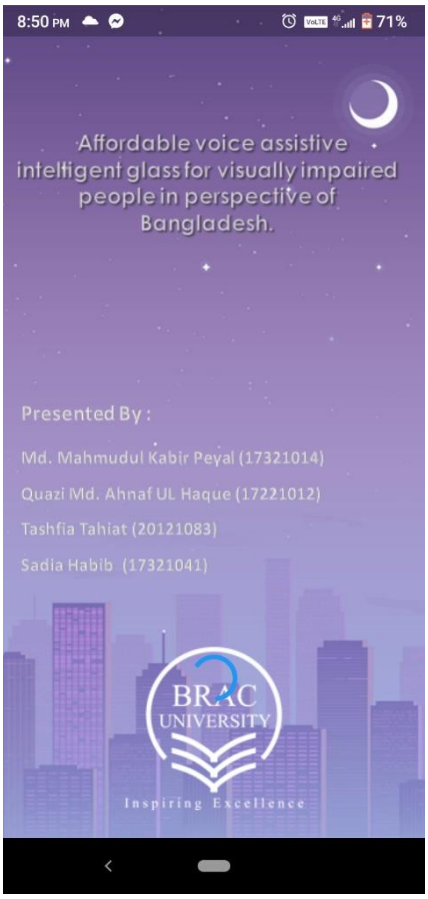


Figure 6. 1: Welcome Screen of App

6.2 Home Screen and Situation Indicator:

We have designed the app home screen as a situation indicator of a blind person. Home screen UI colour changes according to a visually disabled person's condition. If a person is in a safe position and has no problem then UI shows the whole app screen in blue colour, otherwise, the home screen colour becomes red. The human face Icon button (Smiley) at top of the home screen also shows the situation of a person if the user long-press in that icon button. At the bottom side of the app, there is a bottom bar that has options to call at emergency numbers such as, national emergency (999), ambulance service, fire service, doctor. In the middle of the home screen, there are three status receiving options which are divided according to the problem faced by a blind person. They are discussed below:

1. **Personal Issue:** The feature "Personal issue" is basically for the impaired people's personal needs which are emergency for his/her. For example, the person can ask for his/her medicines or his/her personal kinds of stuff to bring to his/her near ones. The impaired person can ask his/her near ones for those things through chat. If the patient faces any problem like he/she does not find out his/her medicines can chat with his/her relative about this problem or when he/she is at risk like he/she is facing unavoidable circumstances outside, he/she let his/her relatives know about the problems he/she is facing by using the feature via chat.
2. **Health Issue:** The feature is basically for the impaired people's health-related updates. When the user (impaired person) feels sick, he/she can send the voice command which will be updated in the "health issue" and can let other persons know about his/her condition. For instance, if the impaired person is facing sickness like high fever, immediately he /she

can let his/her close ones know by saying this command – “heath status fever”. Here instead of fever glass user can use any health issue problem.

3. **Other Situations:** In this feature, we can get sensor value or sensor updates from the environment. In-home automation section we had discussed how we could get the sensor reading and updated on the server. As we have used two sensors: gas sensor and flame sensors to show we will get an update from the gas sensor as there any gas leakage and the flame sensor will detect flame.

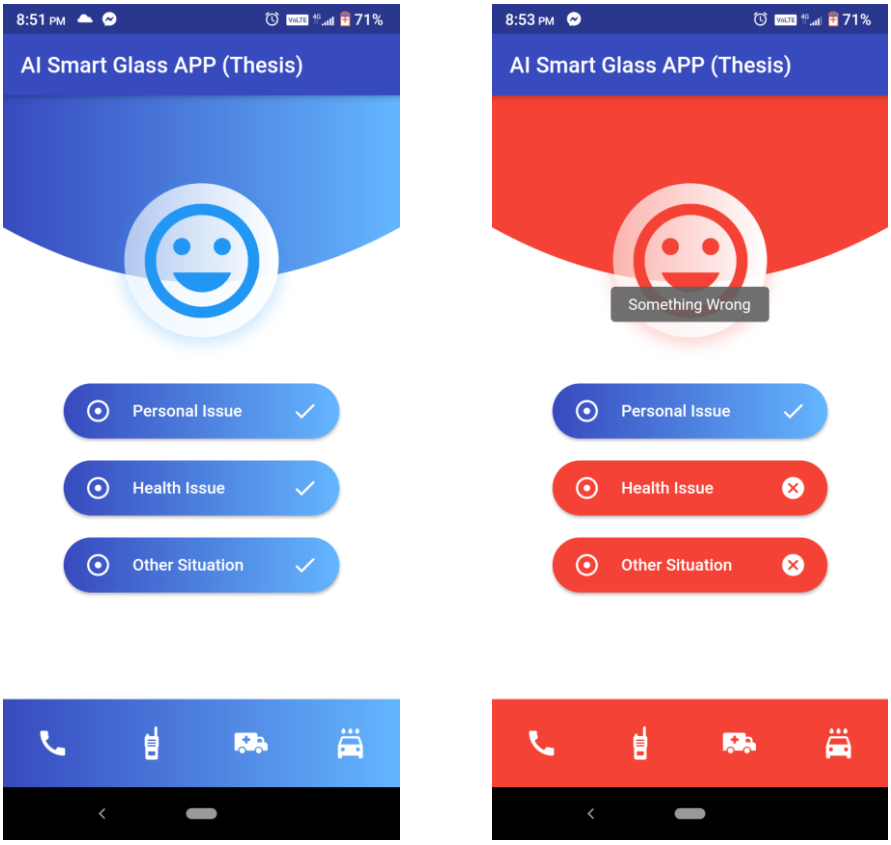


Figure 6. 2: App Home Screen

6.3 Realtime Emergency Status:

This part of the feature basically works for giving the alert as a message. On the home screen, we can detect any problem that occurs or not. However, to know what is the exact problem app user

will have to enter the status page. Such as, if the gas sensor detects gas leakage immediately the app will give the alert and show the message- “Gas Leakage Found” through the app to the other person (the close one of the impaired people). The app works the same way if the flame sensor detects any flame immediately or for others issues. When the issues are gone the app will automatically turn off being red and after removing the status app will go back to the normal mood that can be described as a safe indicator means everything is fine on the other side. The “Status” feature can also be used as an emergency one-way messenger for example the person (visually impaired) can convey a message like he/she is not feeling well or she/he is at risk so immediately after the new status the app will turn red to give a signal to his/her relatives that the person’s condition is not good, he might need some help. When there is new status, the app home screen UI will always in red colour.

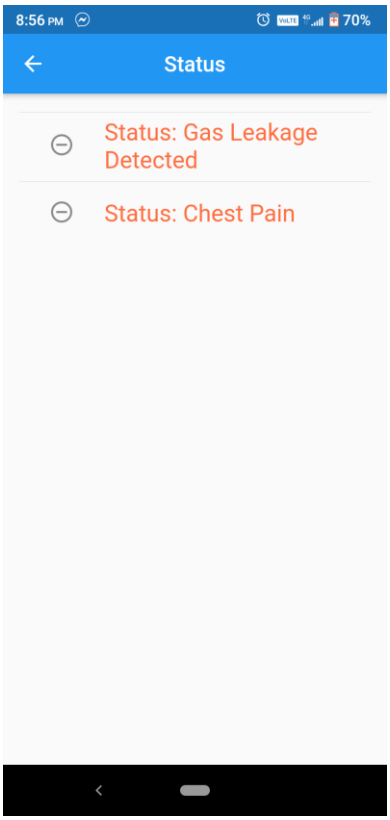


Figure 6. 3: Emergency Status

6.4 Chat Feature:

This section is for a chat with another person. As we know that it is almost difficult for the visually impaired person to read or write any message. By using AI glass and app he/she can do chat with other person using voice and hearing audio. In this section, the sender (the impaired person) can give voice clips which will convert into text and will send it to the recipient (relatives or concerned person). The recipient (relatives or concerned person) will be able to read the message as text which actually a converted version of the sender's voice message and reply by using a phone keyboard. This text will again be converted to speech and will send to glass through a cloud database. An interesting point about this feature is the user will be able to chat with the person using both Bangla and the English language.

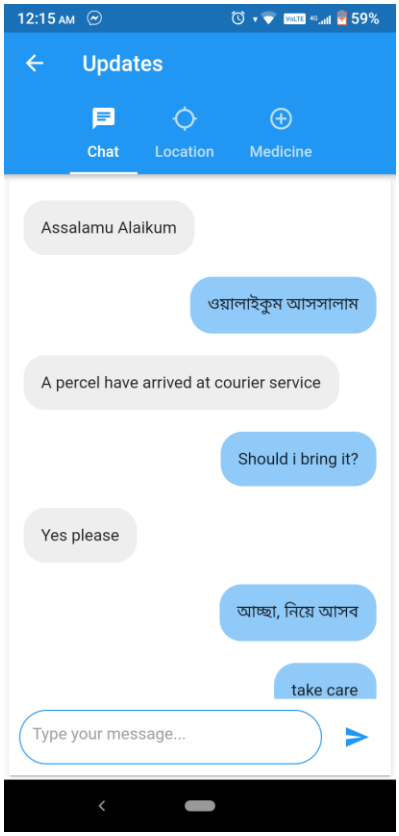


Figure 6. 4: Two-way communication through chatting feature

6.5 Location Tracking:

As per the needs, the impaired person also have to go outside of the house so it is a very stressful time for the close ones. However, with this feature, the relatives can easily track his/her location even if they are not with the person. By using the app relatives who are using the app can know the exact location of the impaired person. So, if the person faces any problem outside, his /her relatives easily trace the location and help the person as early as possible or can give directions if the impaired person needs.

Figure 6. 5: Real-time Location Tracking through AI-glass GPS

6.6 Medicine Adder and Reminder:

This section is for adding medicine according to doctors prescription and reminder for the medicines for the user. The user can be an older person or younger one, some of them might have regular basis medicines because of health issues or some of them have sudden sickness for that disease some medicines need to take for a certain period.” Medicine Reminder” can be helpful for them as there may be some times when they forget to take their medicines but this section can help by giving reminder just in time. As there are some medicines that are just for short times so we can delete the reminder after the time ends or we can add medicine when it needs.

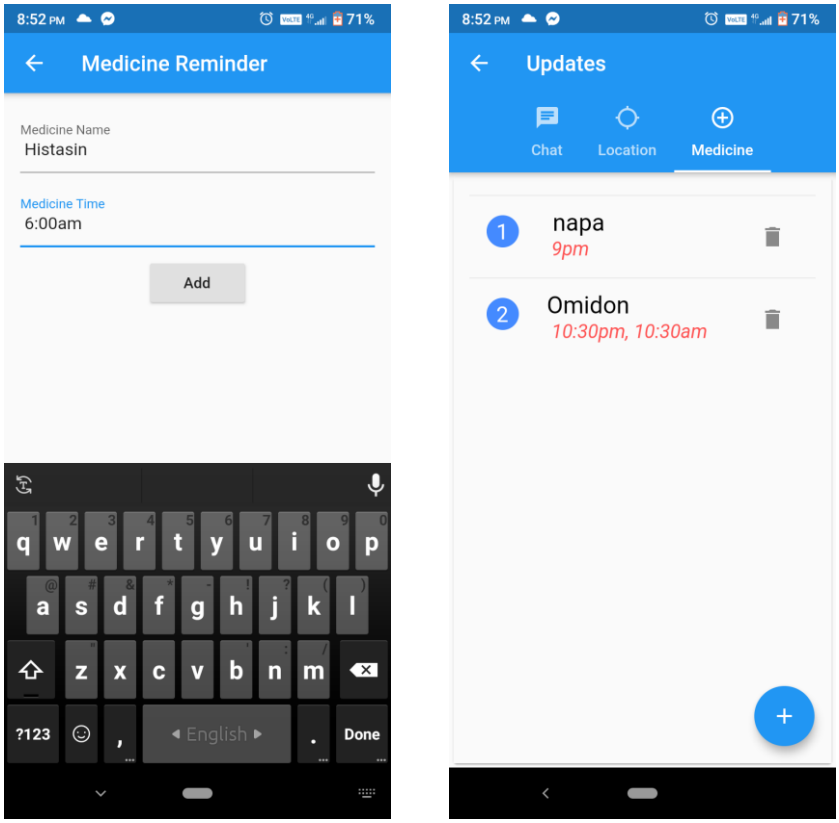


Figure 6. 6: Medicine adding to database for alarm system

Chapter 7

Feasibility Analysis

7.1. Impact and Sustainability

7.1.1 Economic Sustainability and Impact:

Our main motive for this project is to offer affordable smart glass to blind and visually impaired people. As everyone can afford the smart glass it will make the blind and visually impaired people life easier. It is a cost-effective product so people of all classes can buy the product. We all know that Bangladesh is a developing country still now most of the people of our country can not afford expensive products as smart glass is available at a low price so people of all classes can afford it thus the demand for the smart glass will be increased. By doing proper marketing on a large scale, we can not only assure a better life for the blind and visually impaired people but also contribute to our country's economy. As per the demand response, we can produce more smart glass for blind and visually impaired people, and the more we increase the production the more we need more manpower. The need for more manpower can involve many unemployed employees and make contributions to the economic development of our country. We can also export the smart glass to other countries at an affordable price which will contribute to our country's economy.

7.1.2 Social Sustainability and Impact

This project also has a great impact on our society. As we know, blind and visually impaired people have to face a lot of difficulties in their daily life whereas our smart glass can make their life a little bit easier. By using the smart glass can detect objects which helps blind people if they want to know what is in front of them or if something they are finding, they can easily do that by

themselves. For example, they are searching for a water bottle for drinking or clothes for wearing, they can easily detect by the smart glass which is making them more independent for doing their work. Our smart glass can also make a change in the quality of education for blind people. One of the important features is detecting text and convert into speech. As Bangladesh is a developing country most of the blind and visually impaired people can not afford or get the privilege of special teaching but by using the smart glass, they can join the normal students without any hassle as they can detect the text from books or from any site which will convert into speech and read it out for them. This will not only ensure a better education or the blind people also decrease the inequalities as they participate just like normal people.

7.1.3 Environmental Sustainability and Impact

Our project is completely environmentally sustainable. We have used all the components for our project are environment friendly. It has no direct impact on the environment.

7.2. Applicable Standards and Codes

In our design project, we have proposed an experimental design consists of onboard computer like Raspberry Pi, camera, sensors, microcontrollers, web-based server, earphone, etc. In our first target is to detect different types of objects using a machine learning algorithm. As there are many types of machine learning algorithms available, we have used deep learning algorithms for object detection for better performance. Point to be noted that a deep learning algorithm is a complex mathematical process that decreases the performance of microprocessors. Next, we have used an optical character recognition engine named tesseract for text recognition. It is open-source software, released under the Apache License. We used its API directly to detect any kind of text easily. This software can recognize more than a hundred languages including Bangla. It can be

trained to recognize other languages also. For sending output to the user, an earphone will be used where all the feedback from the onboard computer convert to speech and then delivered. Moreover, our target audience can control home appliances on their own like fan and fan, they can turn it on or off by command. The relatives also get the update of the blind people through our App. There are a lot of features in the app which will help the blind people like giving health update, medicine reminder, etc. To fulfill our objective, we took the help of many software and programming languages. For programming an on-board computer and making different kinds of object detection models we have used python programming language as python is the best programming language for scientific and mathematical work. It has a big open-source library for image processing like TensorFlow, Keras, and text to speech conversion library like Pysttx3. Python is an easier and user-friendly programming language than other programming languages like Java, C#, Fortran, etc. After that, for programming microcontroller, we have used Arduino programming language which is nothing but a modified C programming language. Arduino is the perfect programming language because it has a built-in library feature for IoT-based microcontrollers. Arduino has basic built-in functions to keep the program simple and help to complete the task by writing less amount of code. As we design an Android App. For that Dart programming language is the best option as the Dart compiler is so fast and reduces code development time. Google has made an open-source UI toolkit named Flutter which supports dart language for making Android building our custom deep learning model and training we will use Google Colab Notebook which executes code in Google cloud server. Next, for basic python programming, sensors interfacing a server communication with an onboard computer we use basic Python IDE and Node-red programming tool. Additionally, for designing the app we use Android Studio Software. As we are using deep learning algorithms which need a large amount of data to understand it perfectly. For that reason,

we have collected our dataset from online data which are freely available for use. We have also collected datasets by taking pictures by our camera as our target dataset are that kind of data which are available around us and very much needed in our everyday life like mug, brush, bed, table, bus, car, cycle, pet animals, etc.

7.3. Ethical Consideration

All resources and data used for our project have been treated as confidential. All the study and work have been voluntarily done by us. We have carefully observed and examine our data. No payment has been made to any person who will directly or indirectly involve in this project work. We have avoided discrimination against our members. We have also focus on minimizing human and animal harms as much as possible and maximizing benefits.

7.4. Risk Management and Safety Consideration

From the beginning of the project, one of our priorities was to ensure safety to the user of blind people. Keeping that in mind, we have set the main device (raspberry pi) outside of the glass as the user does not feel any heat also remains safe. Secondly, we have used a Li-ion battery instead of a LiPo battery as Li-ion is low cost, has high power density, the battery does not overheat or explode also has no memory effect [33]. We had active the authentication system of Firebase as the communication between database and smart glass can not be controlled or changed by others.

Chapter 8

Conclusion

8.1 Summary

After one year of study and effort, the result we have found is a great accomplishment. The main purpose of our project was to make a low-cost solution to help visually impaired people and make their life secure as well as comfortable. The control systems and a combination of different features have made this artificial spectacle unique. As datasets are collected from Bangladesh and new types of models are built with different classes that can be a great contribution to machine learning work. Text detection with OCR with this glass will enable a visually impaired person to study usual books like us instead of using the brails method. To ensure the safety of blind persons home automation systems can create a great impact. After doing several types of testing we can ensure this project can be a great product and can compete with other products available in the market. This accomplishment will lead us to work further on this project and develop this as a product gradually. To conclude, we can say that this project has a tremendous social and environmental impact as we know visually impaired people face tons of complications in social integration and they might have fears of approaching. Nevertheless, this project can help those persons to overcome that fears.

8.2 Future Work

In the future, we have a plan to improve all features which we have already introduced. Firstly, instead of image classification, we will approach image segmentation so that the object detection process becomes more specific and users can detect in addition find their object according to their needs with proper directions. Secondly, we will build a new model for Bangla text detection as the open-source model's accuracy is not much effective. We will try to use a high-resolution camera and apply advanced image processing algorithms to reduce image noise for getting better results its text detection. In-home automation feature we have the plan to incorporate a control system which will be able to control analog value of devices like fan regulator, ac temperature. We will introduce more sensors to make the environment safer and avoid unwanted circumstances.

Reference

[1] Ahmed JU, Ahmed S, Ahmed A, Abdul Majid M. Caring Vision: Fighting Against Blindness in Bangladesh. FIIB Business Review. 2019;8(1):17-24.

[2] CEICdata.com. (2018, June 1). *Bangladesh Household Income per Capita*. https://www.ceicdata.com/en/indicator/bangladesh/annual-household-income-per-capita

[3] J. -Y. Lin, C. -L. Chiang, M. -J. Wu, C. -C. Yao and M. -C. Chen, "Smart Glasses Application System for Visually Impaired People Based on Deep Learning," 2020 Indo – Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), 2020, pp. 202-206.

[4] H. AlSaid, L. AlKhatib, A. AlOraidh, S. AlHaidar and A. Bashar, "Deep Learning Assisted Smart Glasses as Educational Aid for Visually Challenged Students," 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), 2019, pp. 1-6.

[5] Feng Lan, Guangtao Zhai and Wei Lin, "Lightweight smart glass system with audio aid for visually impaired people," TENCON 2015 - 2015 IEEE Region 10 Conference, 2015, pp. 1-4.

[6] KARTHIK, V. K. RAJA and S. PRABAKARAN, "Voice Assistance for Visually Impaired People," *2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, 2018, pp. 465-468.

[7] S. A. Jakhete, P. Bagmar, A. Dorle, A. Rajurkar and P. Pimplikar, "Object Recognition App for Visually Impaired," *2019 IEEE Pune Section International Conference (PuneCon)*, 2019, pp. 1-4.

[8] N. Sumi and V. Moshnyaga, "A novel face recognition for smart glasses," *2016 IEEE Region 10 Symposium (TENSYMP)*, 2016, pp. 115-118.

[9] Heidloof, N. What is Node-RED? How can it be used for the Internet of Things? . Available: http://heidloff.net/article/21.01.2015081841NHEAL8.htm

[10] Soppin, S. Everything you need to know about Node-RED.

Available: https://www.opensourceforu.com/2017/09/node-red/.

[11] Firebase Realtime Database. April 20,2021(last updated). Firebase. Available: https://firebase.google.com/docs/database?hl=en

[12] Sheldon, R., Shore. Cloud database.

Available: https://searchcloudcomputing.techtarget.com/definition/cloud-database

[13] Cloud Database. In Wikipedia. Available: https://en.wikipedia.org/wiki/Cloud_database

[14] Sharma, A.Realtime Database vs. Cloud Firestore – which Database is suitable for your Mobile App. Available: https://medium.datadriveninvestor.com/realtime-database-vs-cloud-firestore-which-database-is-suitable-for-your-mobile-app-87e11b56f50f

[15] S. Albawi, T. A. Mohammed and S. Al-Zawi, " Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, 2017, pp. 1-6.

[16] Rampasek, L., & Goldenberg, A. (2016). TensorFlow: Biology’s Gateway to Deep Learning?.Cell Systems, 2(1), 12-14.

[17] Keras high level neural network API from: https://keras.io/

[18] Find Open Datasets and Machine Learning Projects | Kaggle. Kaggle.com. (2021). Retrieved 28 April 2021, from https://www.kaggle.com/datasets.

[19] Tabassum, Shaira; Ullah, Md. Sabbir ; Al-nur, Nakib Hossain; Shatabda, Swakkhar (2020), "Poribohon-BD", Mendeley Data, V2.

[20] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[21] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[22] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network, " 2017 International Conference on Engineering and Technology (ICET), Antalya, 2017, pp. 1-6.

[23] Hoon Chung, Sung Joo Lee and Jeon Gue Park, "Deep neural network using trainable activation functions," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 348-352.

[24] D. Kingma and J. Ba, “Adam: A method for stochasticoptimization,”arXiv preprint arXiv:1412.6980, 2014.

[25] ShenzhenAi-Thinker Technology Co., Ltd. (2021). *Nodemcu-32s Datasheet* [Ebook] (1st ed., p. 12). Retrieved 25 April 2021, from https://docs.ai-thinker.com/_media/esp32/docs/nodemcu-32s_product_specification.pdf.

[26] Songle. *Songle Relay* [Ebook] (p. 2). Retrieved 25 April 2021, from https://datasheetspdf.com/pdf-file/720556/Songle/SRD-05VDC-SL-C/1.

[27] *MQ-6 Gas Sensor Pinout, Specs, Equivalent, Circuit & Datasheet*. Components101. (2021). Retrieved 26 April 2021, from https://components101.com/sensors/mq-6-gas-sensor-pinout-equivalent-datasheet.

[28] *Flame Sensor*. Waveshare.com. (2021). Retrieved 26 April 2021, from https://www.waveshare.com/flame-sensor.htm.

[29] Flutter, Flutter GitHub Repository. (2019). Retrieved 26 April 2021, from https://github.com/flutter/flutter

[30] Flutter’s official Website. Retrieved 26 April 2021, from https://github.com/flutter/flutter

[31] Google, Dart’s official Website. Retrieved 26 April 2021, from https://dart.dev/

[32] Firebase Products. Firebase. (2021). Retrieved 26 April 2021, from https://firebase.google.com/products-build.

[33] as the. "Lithium Ion vs. Lithium Polymer Batteries – Which Is Better?" RAVPower.29May, 2021.Available: blog.ravpower.com/2017/06/lithium-ion-vs-lithium-polymer-batteries/.

Appendix A

1. Code for Object Detection

```
from imageai.Classification import ImageClassification
from picamera.array import PiRGBArray
from picamera import PiCamera
import cv2
cam=cv2.VideoCapture(0)
```

```
prediction = ImageClassification()
prediction.setModelTypeAsResNet50()
prediction.setModelPath("modelthesis.h5")
prediction.loadModel()

cv2.namedWindow("test")
c="starting..";

while True:
    rct, frame=cam.read()
    if not rct:
        print("failed to grab frame")
        break
    font= cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(frame,c, (10,50), font, 0.9, (0,255,255), 1, cv2.LINE_AA)
    cv2.imwrite("image.jpg", frame)
    cv2.imshow("test",frame)
    k=cv2.waitKey(2)
    if k%256 == 97:
        predictions, probabilities = prediction.classifyImage("image.jpg", result_count=1 )
        #c= predictions[0]

                c=eachPrediction
                print(eachPrediction," : " , eachProbability)

    if k%256 ==27 :
        print("Escape hit,closing..")
        break
cam.release()
```

cv2.destroyAllWindows()

2. Code for Image Training

```
import os
import keras
import pickle
import numpy as np
def resnet50_model(classes=1000, *args, **kwargs):
# Load a model if we have saved one
if(os.path.isfile('C:\\DATA\\Python-data\\CIFAR-10\\models\\resnet_50.h5') == True):
return keras.models.load_model('model.h5')
input = keras.layers.Input(shape=(None, None, 3))
output = keras.layers.ZeroPadding2D(padding=3, name='padding_conv1')(input)
output = keras.layers.Conv2D(64, (7, 7), strides=(2, 2), use_bias=False,
name='conv1')(output)
output = keras.layers.BatchNormalization(axis=3, epsilon=1e-5, name='bn_conv1')(output)
output = keras.layers.Activation('relu', name='conv1_relu')(output)
output = keras.layers.MaxPooling2D((3, 3), strides=(2, 2), padding='same',
name='pool1')(output)
output = conv_block(output, 3, [64, 64, 256], stage=2, block='a', strides=(1, 1))
output = identity_block(output, 3, [64, 64, 256], stage=2, block='b')
output = identity_block(output, 3, [64, 64, 256], stage=2, block='c')
output = conv_block(output, 3, [128, 128, 512], stage=3, block='a')
output = identity_block(output, 3, [128, 128, 512], stage=3, block='b')
output = identity_block(output, 3, [128, 128, 512], stage=3, block='c')
output = identity_block(output, 3, [128, 128, 512], stage=3, block='d')
output = conv_block(output, 3, [256, 256, 1024], stage=4, block='a')
output = identity_block(output, 3, [256, 256, 1024], stage=4, block='b')
output = identity_block(output, 3, [256, 256, 1024], stage=4, block='c')
output = identity_block(output, 3, [256, 256, 1024], stage=4, block='d')
output = identity_block(output, 3, [256, 256, 1024], stage=4, block='e')
output = identity_block(output, 3, [256, 256, 1024], stage=4, block='f')
output = conv_block(output, 3, [512, 512, 2048], stage=5, block='a')
output = identity_block(output, 3, [512, 512, 2048], stage=5, block='b')
output = identity_block(output, 3, [512, 512, 2048], stage=5, block='c')
output = keras.layers.GlobalAveragePooling2D(name='pool5')(output)
output = keras.layers.Dense(classes, activation='softmax', name='fc1000')(output)
model = keras.models.Model(inputs=input, outputs=output, *args, **kwargs)
print()
print(model.summary(), '\n')
model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.adam(lr=0.01,
clipnorm=0.001), metrics=['accuracy'])
```

```
return model
def identity_block(input, kernel_size, filters, stage, block):
filters1, filters2, filters3 = filters
conv_name_base = 'res' + str(stage) + block + '_branch'
bn_name_base = 'bn' + str(stage) + block + '_branch'
  # Create layers
output = keras.layers.Conv2D(filters1, (1, 1), kernel_initializer='he_normal',
name=conv_name_base + '2a')(input)
output = keras.layers.BatchNormalization(axis=3, name=bn_name_base + '2a')(output)
output = keras.layers.Activation('relu')(output)
output = keras.layers.Conv2D(filters2, kernel_size, padding='same',
kernel_initializer='he_normal', name=conv_name_base + '2b')(output)
output = keras.layers.BatchNormalization(axis=3, name=bn_name_base + '2b')(output)
output = keras.layers.Activation('relu')(output)
output = keras.layers.Conv2D(filters3, (1, 1), kernel_initializer='he_normal',
name=conv_name_base + '2c')(output)
output = keras.layers.BatchNormalization(axis=3, name=bn_name_base + '2c')(output)
output = keras.layers.add([output, input])
output = keras.layers.Activation('relu')(output)
return output
def conv_block(input, kernel_size, filters, stage, block, strides=(2, 2)):
filters1, filters2, filters3 = filters
conv_name_base = 'res' + str(stage) + block + '_branch'
bn_name_base = 'bn' + str(stage) + block + '_branch'
output = keras.layers.Conv2D(filters1, (1, 1), strides=strides, kernel_initializer='he_normal',
name=conv_name_base + '2a')(input)
output = keras.layers.BatchNormalization(axis=3, name=bn_name_base + '2a')(output)
output = keras.layers.Activation('relu')(output)
output = keras.layers.Conv2D(filters2, kernel_size, padding='same',
kernel_initializer='he_normal', name=conv_name_base + '2b')(output)
output = keras.layers.BatchNormalization(axis=3, name=bn_name_base + '2b')(output)
output = keras.layers.Activation('relu')(output)
output = keras.layers.Conv2D(filters3, (1, 1), kernel_initializer='he_normal',
name=conv_name_base + '2c')(output)
output = keras.layers.BatchNormalization(axis=3, name=bn_name_base + '2c')(output)
shortcut = keras.layers.Conv2D(filters3, (1, 1), strides=strides,
kernel_initializer='he_normal', name=conv_name_base + '1')(input)
shortcut = keras.layers.BatchNormalization(axis=3, name=bn_name_base + '1')(shortcut)
output = keras.layers.add([output, shortcut])
output = keras.layers.Activation('relu')(output)
return output
def train():
epochs = 1
batch_size = 32
train_samples = 10 * 5000 # 10 categories with 5000 images in each category
validation_samples = 10 * 1000 # 10 categories with 1000 images in each category
```

```
img_width, img_height = 32, 32
model = resnet50_model(10)
train_data_generator = keras.preprocessing.image.ImageDataGenerator(
rescale=1./255,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True)
validation_data_generator = keras.preprocessing.image.ImageDataGenerator(
rescale=1./255,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True)
train_generator = train_data_generator.flow_from_directory(
'C:\\DATA\\Python-data\\CIFAR-10\\train',
target_size=(img_width, img_height),
batch_size=batch_size,
color_mode='rgb',
shuffle=True,
class_mode='categorical')
validation_generator = validation_data_generator.flow_from_directory(
'C:\\DATA\\Python-data\\CIFAR-10\\test',
target_size=(img_width, img_height),
batch_size=batch_size,
color_mode='rgb',
shuffle=True,
class_mode='categorical')
model.fit_generator(
train_generator,
steps_per_epoch=train_samples // batch_size,
validation_data=validation_generator,
validation_steps=validation_samples // batch_size,
epochs=epochs)
model.save(“model.h5”)
print('Saved model to disk!')
labels = train_generator.class_indices
classes = {}
for key, value in labels.items():
classes[value] = key.capitalize()
with open('C:\\DATA\\Python-data\\CIFAR-10\\classes.pkl', 'wb') as file:
pickle.dump(classes, file)
print('Saved classes to disk!')
def main():
train()
 # Tell python to run main method
if __name__ == '__main__': main()
```

3. Model Evaluatin

```
##Do overall test and get F1 by your model

test_datagen = ImageDataGenerator(
        rescale=1. / 255)
test_generator = test_datagen.flow_from_directory("/content/drive/MyDrive/thesis database/small dataset/food/test", target_size=(224, 224),
                                                  batch_size=1,shuffle=False,
                                                  class_mode="categorical")
# make predictions on the data
print("[INFO] evaluating network...")
test_generator.reset()
from keras.models import load_model
#load model
model1 = load_model("/content/drive/MyDrive/thesis database/small dataset/food/models/model_ex-033_accuracy-0.975000.h5")
predIdxs = model1.predict(x=test_generator)#, steps=(20 // 4) + 1)

    # for each image in the testing set we need to find the index of the
    # label with corresponding largest predicted probability
predIdxs = np.argmax(predIdxs, axis=1)
print(predIdxs)
print(test_generator.classes)
print(len(predIdxs))
print(classification_report(test_generator.classes, predIdxs,target_names=test_generator.class_indices.keys()))
```

Appendix B

1. Code for Text Detection By Camera

```
from gtts import gTTS

import pytesseract as tess

from langdetect import detect
```

```
import urllib3
from PIL import Image
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2

urllib3.disable_warnings()
cam=cv2.VideoCapture(0)

cv2.namedWindow("test")
c="starting..";

while True:
    rct, frame=cam.read()
    if not rct:
        print("failed to grab frame")
        break
    font= cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(frame,c, (10,50), font, 0.9, (0,255,255), 1, cv2.LINE_AA)
    cv2.imwrite("2.jpg", frame)
    cv2.imshow("test",frame)
    k=cv2.waitKey(2)
    if k%256 == 97:
        text1= tess.image_to_string("2.jpg")
        cv2.imwrite("image_captured.jpg", frame)
        language=detect(text1)
        print(language)
        if language=="en" :
            print(text1)
```

```
            audio = gTTS(text=text1, lang=language, slow=False)
            audio.save("audio.wav")
        else:
            text1 = tess.image_to_string("2.jpg",lang='ben')
            language = detect(text1)
            print(text1)
            audio = gTTS(text=text1, lang=language, slow=False)
            audio.save("audio.wav")
    if k%256 ==27 :
        print("Escape hit,closing..")
        break
cam.release()
cv2.destroyAllWindows()
```

2. Code for English Newspaper and Bangla Text Detection

```
from gtts import gTTS
import pytesseract as tess
from langdetect import detect

import urllib3
from PIL import Image
import cv2

urllib3.disable_warnings()

img = Image.open('/home/pi/Desktop/Text/news.jpeg')  #for newspaper detecting
#img = Image.open('/home/pi/Desktop/Text/bangla4.png')  #for bangla detecting
text1 = tess.image_to_string(img)
```

```
language = detect(text1)
print(language)

if language=="en" :
    print(text1)
    audio = gTTS(text=text1, lang=language, slow=False)
    audio.save("1.wav")

else:
    text1 = tess.image_to_string(img ,lang='ben')
    language = detect(text1)
    #if language == "bn" :
    print(text1)
    audio = gTTS(text=text1, lang=language, slow=False)
    audio.save("1.wav")
```

Appendix C

1. App Design

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:firebase_database/firebase_database.dart';
```

```
import 'package:firebase_database/ui/firebase_animated_list.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/widgets.dart';
import 'package:splashscreen/splashscreen.dart';
import 'dart:ui' as ui;
import 'ListView_note.dart';
import 'medilist.dart';
import 'note.dart';
import 'note_screen.dart';
import 'dart:async';
import 'humanitarian_icons.dart';
import 'package:url_launcher/url_launcher.dart' ;
import 'package:flutter_phone_direct_caller/flutter_phone_direct_caller.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:flutter/cupertino.dart';
import 'package:geolocator/geolocator.dart';
import 'package:google_maps_flutter/google_maps_flutter.dart';
import 'package:flutter_polyline_points/flutter_polyline_points.dart';

var _firebaseRef = FirebaseDatabase().reference();

void main() async{
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  WidgetsFlutterBinding.ensureInitialized();
  SystemChrome.setPreferredOrientations([DeviceOrientation.portraitUp]);
  runApp(
     MaterialApp(
     debugShowCheckedModeBanner: false,
```

```
      home: MyApp(),
    ),
  );

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => new _MyAppState();
}

class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return new SplashScreen(
        seconds: 10,
        navigateAfterSeconds: new FirstRoute(),
        imageBackground: AssetImage('images/sp.png'),
    );
  }
}
var _a=true;
var _b=true;
var _c=true;

bool update=true;
class FirstRoute extends StatefulWidget {
  FirstRoute({Key key}) : super(key: key);

  @override
  _Froute createState() => _Froute();
  }
```

```
class _Froute extends State<FirstRoute> {
  List<Note> items;
  StreamSubscription<Event> _onNoteAddedSubscription;
  StreamSubscription<Event> _onNoteValueSubscription;
  StreamSubscription<Event> _onNoteChangedSubscription;

  TextEditingController _numberCtrl = new TextEditingController();


  @override
  void initState() {
    super.initState();
    items = new List();
    _onNoteValueSubscription = _firebaseRef.onChildAdded.listen(_onNoteAdded);
    _onNoteAddedSubscription = _firebaseRef.onValue.listen(_onNoteAdded);
    _onNoteChangedSubscription = _firebaseRef.onChildChanged.listen(_onNoteUpdated);

    super.initState();
    _numberCtrl.text = "01732000831";
  }

  @override
  void dispose() {
    _onNoteValueSubscription.cancel();
    _onNoteAddedSubscription.cancel();
    _onNoteChangedSubscription.cancel();
    super.dispose();
  }
  @override
 Widget build(BuildContext context) {
```

```
  return Scaffold(appBar: new AppBar(title: new Text("AI Smart Glass APP (Thesis)"),backgroundColor: Color(0xff374ABE) ),
    backgroundColor: Colors.white,
    body: ListView(
      children: <Widget>[
        Stack(
          children: <Widget>[
            Padding(
              padding: const EdgeInsets.only(bottom: 50.0),
              child: ClipPath(
                clipper: ClippingClass(),
                child: Container(
                  height: 170.0,
                  decoration: BoxDecoration(
                    gradient: LinearGradient(
                      colors: [(_a&_b&_c) ? Color(0xff374ABE) : Colors.red, (_a&_b&_c) ? Color(0xff64B6FF) : Colors.red],
                      begin: Alignment.centerLeft,
                      end: Alignment.centerRight,
                    ),
                  ),
                ),
              ),
            ),
            Positioned.fill(
              child: Align(
                alignment: Alignment.bottomCenter,
                child: Container(
                  height: 140.0,
                  width: 140.0,
```

```
                child:Container(
                  decoration:BoxDecoration(
                     gradient: LinearGradient(
                       colors: [
                         Colors.white60,
                         Colors.white,
                       ],
                       begin: Alignment.centerLeft,
                       end: Alignment.centerRight,
                     ),
                     borderRadius: const BorderRadius.all(
                       Radius.circular(100.0),
                     ),
                     boxShadow: [
                       BoxShadow(
                         color: (_a&_b&_c) ? Colors.blue.withOpacity(0.2) : Colors.red.withOpacity(0.2),
                         spreadRadius: 4,
                         blurRadius: 10,
                         offset: Offset(0, 3),
                       )
                     ]
                  ),
                  child:IconButton(
                    icon: Icon(Icons.insert_emoticon),
                   iconSize: 125.00,
                   color: (_a&_b&_c) ? Colors.blue : Colors.red,
                   tooltip: (_a&_b&_c) ? 'He is Good':'Something Wrong',
                   onPressed: () {
```

```
                Navigator.push(context, MaterialPageRoute(builder: (context) => SecondRoute()),);
                }
                ),
              ),
            ),
          ),
        )
      ],
    ),
    Container(
      padding: const EdgeInsets.only(top: 32.0),
      child: Column(
        children: <Widget>[
        /*
        if(_a=="not ok"){
          update=false;
        }
         else {update=true;}*/
          Updatestate(color1:_a ? Color(0xff374ABE) : Colors.red, color2: _a ? Color(0xff64B6FF):Colors.red ,updateText:"Personal Issue", icon: _a ? Icons.check : Icons.cancel_rounded),
          Updatestate(color1:_b ? Color(0xff374ABE) : Colors.red, color2: _b ? Color(0xff64B6FF):Colors.red ,updateText:"Health Issue",icon: _b ? Icons.check : Icons.cancel_rounded),
          Updatestate(color1:_c ? Color(0xff374ABE) : Colors.red, color2: _c ? Color(0xff64B6FF):Colors.red , updateText:"Other Situation",icon: _c ? Icons.check : Icons.cancel_rounded),
        ],
      ),
    ),
```

```
      ],
    ),
    bottomNavigationBar:BottomAppBar(
      child: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [(_a&_b&_c) ? Color(0xff374ABE) : Colors.red, (_a&_b&_c) ? Color(0xff64B6FF) : Colors.red],
            begin: Alignment.centerLeft,
            end: Alignment.centerRight,
          ),
        ),
        constraints: BoxConstraints.expand(
          height: 80,
        ),
        alignment: Alignment.center,
        child: Row(
          children: <Widget>[
            Expanded(
              flex: 1,
              child: IconButton(icon: Icon(Icons.call), color: Colors.white, iconSize: 30, tooltip: 'Call Now', onPressed:_callNumber,),
            ),
            Expanded(
              flex: 1,
              child: IconButton(icon: Icon(HumanitarianIcons.walkie_talkie), color: Colors.white, iconSize: 30, tooltip: 'Call 999', onPressed:_callemergency, ),
            ),
            Expanded(
              flex: 1,
```

```
              child: IconButton(icon: Icon(HumanitarianIcons.ambulance), color: Colors.white, iconSize: 30, tooltip: 'Ambulance', onPressed: _callambulance,),

              ),

              Expanded(

                flex: 1,

                child: IconButton(icon: Icon(Icons.local_car_wash), color: Colors.white, iconSize: 30, tooltip: 'Fire Service', onPressed:_callfire,),

              ),

             ]

          ),


        ),

      ),

    );

  }

    void _onNoteAdded(Event event) {

      setState(() {

        _a= event.snapshot.value['personal'];

        _b= event.snapshot.value['health'];

        _c= event.snapshot.value['others'];

      });

    }


    void _onNoteUpdated(Event event) {

      var oldNoteValue = items.singleWhere((note) => note.id== event.snapshot.value['personal']);

      setState(() {

        _a = new Note.fromSnapshot(event.snapshot).id;

      });

    }

    void launchUrl(String url) async {
```

```
      if (await canLaunch(url)) {
        launch(url);
      } else {
        throw "Could not launch $url";
      }
    }
}
_callNumber() async{
  const number = '01732000831'; //set the number here
  bool res = await FlutterPhoneDirectCaller.callNumber(number);
}
_callemergency() async{
  const number = '999'; //set the number here
  bool res = await FlutterPhoneDirectCaller.callNumber(number);
}
_callambulance() async{
  const number = '54321'; //set the number here
  bool res = await FlutterPhoneDirectCaller.callNumber(number);
}
_callfire() async{
  const number = '12345'; //set the number here
  bool res = await FlutterPhoneDirectCaller.callNumber(number);
}
class SecondRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
        length: choices.length,
        child: Scaffold(
          appBar: AppBar(
```

```
          title: const Text('Updates'),
          bottom: TabBar(
          isScrollable: true,
          tabs: choices.map<Widget>((Choice choice) {
            return Tab(
             text: choice.title,
             icon: Icon(choice.icon),
            );
          }).toList(),
        ),
      ),
        body: TabBarView( //to work on tab just add code for each tab under children
        children:[
         // ftab(),
          stab(),
          //GeolocatorWidget(),
          ttab(),
          frtab(),
        ]
      ),
      ),
    );
  }
}
void _createNewNote(BuildContext context) async {
  await Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => NoteScreen(Note(" "," "," "))),
  );
}
```

```
class ThirdRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
        length: choices.length,
        child: ListViewNote(),
    );
  }
}
class ClippingClass extends CustomClipper<Path> {
  @override
  Path getClip(Size size) {
    var path = Path();
    path.lineTo(0.0, size.height - 50);
    path.quadraticBezierTo(
      size.width / 4,
      size.height,
      size.width / 2,
      size.height,
    );
    path.quadraticBezierTo(
      size.width - (size.width / 4),
      size.height,
      size.width,
      size.height - 50,
    );
    path.lineTo(size.width, 0.0);
    path.close();
    return path;
```

```
  }

  @override
  bool shouldReclip(CustomClipper<Path> oldClipper) => false;
}


class Choice {
  final String title;
  final IconData icon;
  const Choice({this.title, this.icon});
}

const List<Choice> choices = <Choice>[
 // Choice(title: 'Call', icon: Icons.call),
  Choice(title: 'Chat', icon: Icons.chat),
  Choice(title: 'Location', icon: Icons.location_searching),
  Choice(title: 'Medicine', icon: Icons.add_circle_outline),
];

class ftab extends StatelessWidget {
  const ftab({Key key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Card(
      color: Colors.white,
      child: Center(
        child: Column(
          mainAxisSize: MainAxisSize.min,
```

```
        crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
          RaisedButton(
            onPressed: () {
              Navigator.pop(context);
            },
            child: Text('Home'),
          ),
          RaisedButton(
              onPressed: () {
                Navigator.push(context, MaterialPageRoute(builder: (context) => ThirdRoute()),);
              },
              child: Text('Testing Mood'),
          ),
        ],
      ),
    ),
  );
 }
}
class ChatMessage{
  String _messageContent="a";
  String _messageType="a";
  String _id="a";

  ChatMessage(this._messageContent, this._messageType);

  ChatMessage.map(dynamic obj) {
    this._id = obj['id'];
    this._messageContent = obj['messagebody'];
```

```
    this._messageType = obj['type'];
  }

  String get id => _id;
  String get messageContent => _messageContent;
  String get messageType => _messageType;

  ChatMessage.fromSnapshot(DataSnapshot snapshot) {
    _messageContent = snapshot.value['messagebody'];
    _messageType = snapshot.value['type'];
    _id = snapshot.key;
  }
}
final chatReference = FirebaseDatabase.instance.reference().child('chatty');
class stab extends StatefulWidget{
  stab({Key key}) : super(key: key);
  @override
  _stabstate createState() => _stabstate();
}
class _stabstate extends State<stab> {
  List<ChatMessage> messages;
  //StreamSubscription<Event> _onNoteAddedSubscription;
  StreamSubscription<Event> _onNoteValueSubscription;
  StreamSubscription<Event> _onNoteChangedSubscription;
  /*
  List<ChatMessage> messages = [
    ChatMessage(messageContent: "Hello,ahnaf", messageType: "receiver"),
    ChatMessage(messageContent: "How have you been?", messageType: "receiver"),
    ChatMessage(messageContent: "Hey quazi, I am doing fine dude. wbu?", messageType: "sender"),
```

```
  ChatMessage(messageContent: "ehhhh, doing OK.", messageType: "receiver"),
  ChatMessage(messageContent: "Is there any thing wrong?", messageType: "sender"),
];*/

final TextEditingController textEditingController = new TextEditingController();
final FocusNode focusNode = new FocusNode();
final ScrollController listScrollController = new ScrollController();
@override
void initState() {
  super.initState();
   messages = new List();
  _onNoteValueSubscription = chatReference.onChildAdded.listen(_chatAdded);
  _onNoteChangedSubscription = chatReference.onChildChanged.listen(_chatUpdated);
}
@override
void dispose() {
  _onNoteValueSubscription.cancel();
  _onNoteChangedSubscription.cancel();
  // Clean up the controller when the widget is removed from the
  // widget tree.
  textEditingController.dispose();
  super.dispose();
}
@override
Widget build(BuildContext context) {
  return Card(
    color: Colors.white,
    child:Stack(
        children: <Widget>[
          ListView.builder(
```

```
itemCount: messages.length,
shrinkWrap: true,
padding: EdgeInsets.only(top: 10,bottom: 10),
physics: const AlwaysScrollableScrollPhysics(),
itemBuilder: (context,int index){
  return Container(
    padding: EdgeInsets.only(left: 14,right: 14,top: 10,bottom: 10),
    child: Align(
      alignment: (messages[index]._messageType == "receiver"?Alignment.topLeft:Alignment.topRight),
      child: Container(
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(20),
          color: (messages[index]._messageType == "receiver"?Colors.grey.shade200:Colors.blue[200]),
        ),
        padding: EdgeInsets.all(16),
        child: InkWell(
          child:Text(messages[index]._messageContent.toString(), style: TextStyle(fontSize: 15),),
        onTap: () {
          final snackBar = SnackBar(content: RaisedButton(
            onPressed: () => _deleteNote(context, messages[index], index),
            textColor: Colors.white,
            color: Colors.transparent,
            child: const Text('Delete', style: TextStyle(fontSize: 20)),
          ),);
          Scaffold.of(context).showSnackBar(snackBar);
        },
      ),
      ),
```

```
        ),
      );
    },
  ),
Align(
  alignment: Alignment.bottomLeft,
  child: Container(
    child: Row(
    children: <Widget>[
    // Button send image
     /* Material(
        child: new Container(
        margin: new EdgeInsets.symmetric(horizontal: 1.0),
          child: new IconButton(
            icon: new Icon(Icons.mic),
            onPressed: () {},
        color:  Colors.blue,
      ),
    ),
    color: Colors.white,
  ),  */
  // Edit text
       Flexible(
        child: Container(
          child: TextField(
        style: TextStyle(color: Colors.blue, fontSize: 15.0),
        controller: textEditingController,
        decoration: InputDecoration(
          disabledBorder: OutlineInputBorder(
            borderSide: BorderSide(
```

```
            color: Colors.grey,
          ),
          borderRadius: BorderRadius.circular(30.0),
        ),
        focusedBorder: OutlineInputBorder(
          borderSide: BorderSide(
            color: Colors.blue,
          ),
          borderRadius: BorderRadius.circular(30.0),
        ),
          enabledBorder: OutlineInputBorder(
            borderSide: BorderSide(
              color: Colors.blue,
            ),
            borderRadius: BorderRadius.circular(30.0),
          ),
          hintText: 'Type your message...',
        hintStyle: TextStyle(color: Colors.grey),
      ),
      focusNode: focusNode,
    ),
  ),
),

// Button send message
   Material(
     child: new Container(
      margin: new EdgeInsets.symmetric(horizontal: 8.0),
       child: new IconButton(
        icon: new Icon(Icons.send),
```

```
              onPressed: () => onSendMessage(textEditingController.text, 0),
              color: Colors.blue,
            ),
          ),
          color: Colors.white,
        ),
      ],
    ),
          width: double.infinity,
          padding: EdgeInsets.only(left: 10,bottom: 10,top: 10),
          height: 70.0,
          decoration: new BoxDecoration(
              color: Colors.white),
        ),
      ),
      ],
    ),
  );
}
void onSendMessage(String content, int type) {
  if (content.trim() != '') {
    textEditingController.clear();
    String _child = 'chatty/'+DateTime.now().millisecondsSinceEpoch.toString();
    final chatReference = FirebaseDatabase.instance.reference().child(_child);
    chatReference.set({
      'messagebody': content,
      'type': "sender",
    });
    var documentReference = FirebaseFirestore.instance
        .collection('data')
```

```
        .doc(DateTime.now().millisecondsSinceEpoch.toString());
    FirebaseFirestore.instance.runTransaction((transaction) async {
      await transaction.set(
        documentReference,
        {
          //'idFrom': id,
          'messageType': "sender",
          'timestamp': DateTime.now().millisecondsSinceEpoch.toString(),
          'content': content,
          'type': type
        },
      );
    });
    listScrollController.animateTo(0.0,
        duration: Duration(milliseconds: 300), curve: Curves.easeOut);
  } else {
    Fluttertoast.showToast(msg: 'Nothing to send');
  }
}


void _chatAdded(Event event) {
  setState(() {
    messages.add(new ChatMessage.fromSnapshot(event.snapshot));
  });
}
void _chatUpdated(Event event){
  var oldNoteValue = messages.singleWhere((ChatMessage) => ChatMessage.id.toString() ==
event.snapshot.key);
  setState(() {
```

```
    messages[messages.indexOf(oldNoteValue)] = new ChatMessage.fromSnapshot(event.snapshot);
   });
 }
 void _deleteNote(BuildContext context, ChatMessage ChatMessage, int position) async {
  await chatReference.child(ChatMessage.id).remove().then((_) {
   setState(() {
    messages.removeAt(position);
   });
  });
 }

}
//initial location(Bracu)
var _lat=23.780081166308584;
var _long=90.4071274967339;
class MapScreen extends StatefulWidget {
 @override
 _MapScreenState createState() => _MapScreenState();
}

class _MapScreenState extends State<MapScreen> {
 GoogleMapController mapController;
 double _originLatitude = _lat, _originLongitude = _long;
 //double _destLatitude = 24.771512739584875, _destLongitude = 90.39769634984269;
 Map<MarkerId, Marker> markers = {};
 Map<PolylineId, Polyline> polylines = {};
 List<LatLng> polylineCoordinates = [];
 PolylinePoints polylinePoints = PolylinePoints();
 String googleAPiKey = "AIzaSyB_5ksvGR5IjxOqSbLzypNt0g_LSBtcHWQ";
```

```
StreamSubscription<Event> _onNoteAddedSubscription;
StreamSubscription<Event> _onNoteValueSubscription;
StreamSubscription<Event> _onNoteChangedSubscription;

@override
void initState() {
  _onNoteValueSubscription = _firebaseRef.onChildAdded.listen(_onNoteAdded);
  _onNoteAddedSubscription = _firebaseRef.onValue.listen(_onNoteAdded);
  _onNoteChangedSubscription = _firebaseRef.onChildChanged.listen(_onNoteUpdated);
  super.initState();

  _addMarker(LatLng(_originLatitude, _originLongitude), "origin",
      BitmapDescriptor.defaultMarker);

  // _addMarker(LatLng(_destLatitude, _destLongitude), "destination",
BitmapDescriptor.defaultMarkerWithHue(90));
  _getPolyline();
}
@override
void dispose() {
  _onNoteValueSubscription.cancel();
  _onNoteAddedSubscription.cancel();
  _onNoteChangedSubscription.cancel();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return SafeArea(
```

```
    child: GoogleMap(
          initialCameraPosition: CameraPosition(
              target: LatLng(_originLatitude, _originLongitude), zoom: 15),
          myLocationEnabled: true,
          tiltGesturesEnabled: true,
          compassEnabled: true,
          scrollGesturesEnabled: true,
          zoomGesturesEnabled: true,
          onMapCreated: _onMapCreated,
          markers: Set<Marker>.of(markers.values),
          polylines: Set<Polyline>.of(polylines.values),
        ),
  );
}
void _onNoteAdded(Event event) {
  setState(() {
    //items.add(new Note.fromSnapshot(event.snapshot));
    _lat= event.snapshot.value['lat'];
    _long= event.snapshot.value['long'];
  });
}
void _onNoteUpdated(Event event) {
 if (_lat!= event.snapshot.value['lat']||_long!= event.snapshot.value['long']){
   _lat= event.snapshot.value['lat'];
   _long= event.snapshot.value['long'];
 }
}
void _onMapCreated(GoogleMapController controller) async {
  mapController = controller;
}
```

```
_addMarker(LatLng position, String id, BitmapDescriptor descriptor) {
  MarkerId markerId = MarkerId(id);
  Marker marker =
  Marker(markerId: markerId, icon: descriptor, position: position);
  markers[markerId] = marker;
}

_addPolyLine() {
  PolylineId id = PolylineId("poly");
  Polyline polyline = Polyline(
      polylineId: id, color: Colors.red, points: polylineCoordinates);
  polylines[id] = polyline;
  setState(() {});
}

_getPolyline() async {
  PolylineResult result = await polylinePoints.getRouteBetweenCoordinates(
      "AIzaSyB_5ksvGR5IjxOqSbLzypNt0g_LSBtcHWQ",
      PointLatLng(_originLatitude, _originLongitude),
      PointLatLng(_originLatitude, _originLongitude),
      travelMode: TravelMode.driving,
      wayPoints: [PolylineWayPoint(location: "Last location")]
  );
  if (result.points.isNotEmpty) {
    result.points.forEach((PointLatLng point) {
      polylineCoordinates.add(LatLng(point.latitude, point.longitude));
    });
  }
  _addPolyLine();
```

```
  }
}
class ttab extends StatelessWidget {
  const ttab({Key key}) : super(key: key);
  @override
  _MapScreenState createState() => _MapScreenState();
  @override
  Widget build(BuildContext context) {
    return Card(
      color: Colors.white,
      child:MapScreen(),
    );
  }
}
class frtab extends StatelessWidget {
   frtab({Key key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Card(
      color: Colors.white,
      child: mediListViewNote(),
    );
  }
}



var _volume = "OK";
var _send =1;
bool pressAttention = true;
```

```
TextEditingController _txtCtrl = TextEditingController();

sendMessage() {
  _firebaseRef.push().set({
    "message": _txtCtrl.text,
    "timestamp": 1
  });
}
deleteMessage(key) {
  _firebaseRef.child(key).remove();
}
updateTimeStamp(key) {
  _firebaseRef
      .child(key)
      .update({"timestamp": DateTime.now().millisecondsSinceEpoch});
}

class MyStatefulWidget extends StatefulWidget {
  MyStatefulWidget({Key key}) : super(key: key);

  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  Widget build(BuildContext context) {
    return Column(
      mainAxisSize: MainAxisSize.min,
      children: <Widget>[
        Ink(
```

```
  decoration: const ShapeDecoration(
    color: Colors.white70,
    shape: CircleBorder(),
  ),
  child: IconButton(
    icon: Icon(Icons.security),
    iconSize: 120.00,
    color: pressAttention ? Colors.blue : Colors.grey,
    padding: EdgeInsets.all(10.0),
    tooltip: 'Tap! Tap! Tap!',
    onPressed: () {
      setState(() {
        pressAttention = !pressAttention;
        _firebaseRef.set({"Situation":_send});
        if (_volume == "OK") {
          _volume = "Critical";
          _send = 1;
        }
        else {
          _volume = "OK";
          _send = 0;
        }
      });
    },
  ),
),
Padding(
  padding: EdgeInsets.all(10.0),
  child: Text(
    'Situation : $_volume',
```

```
            style: TextStyle(
              color: Colors.blueGrey,
              fontSize: 20.0,
            ),
          ),
        ),
      ],
    );
  }
}

class Updatestate extends StatefulWidget {
  final String updateText;
  final IconData icon;
  final Color color1;
  final Color color2;
  Updatestate({Key key, this.color1,this.color2,this.updateText,this.icon}) : super(key: key);
  @override
  _UpdateStates createState() => _UpdateStates();
}

class _UpdateStates extends State<Updatestate> {
  Widget build(BuildContext context) {
    return Container(
      height: 50.0,
      margin: EdgeInsets.all(10),
      child: RaisedButton(
        onPressed: () {
          Navigator.push(context, MaterialPageRoute(builder: (context) => ThirdRoute()),);
        },
```

```
shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(80.0)),
padding: EdgeInsets.all(0.0),
child: Ink(
  decoration: BoxDecoration(
      gradient: LinearGradient(
        colors: [ widget.color1,widget.color2],
        begin: Alignment.centerLeft,
        end: Alignment.centerRight,
      ),
      borderRadius: BorderRadius.circular(30.0)),
  child: Container(
    constraints:
    BoxConstraints(maxWidth: 250.0, minHeight: 50.0),
    alignment: Alignment.center,
    child: Row(
      children: <Widget>[
        Expanded(
          flex: 1,
          child: Icon(Icons.adjust,color: Colors.white),
        ),
        Expanded(
          flex: 2,
          child: Text(
            widget.updateText,
            //textAlign: TextAlign.center,
            style: TextStyle(color: Colors.white, fontSize: 15),
          ),
        ),
        Expanded(
```

```
                flex: 1,
                child: Icon(widget.icon,color: Colors.white),
              ),
            ]
            ),
          ),
        ),
      ),
    );
  }
}

/*
Scaffold(
backgroundColor: Colors.blueGrey,
appBar: AppBar(
leading: Icon(Icons.contacts),
title: Text('Emergency'),
),
body: Center(
child: Column(
mainAxisSize: MainAxisSize.min,
crossAxisAlignment: CrossAxisAlignment.center,
children: <Widget>[
MyStatefulWidget(),
RaisedButton(
child: Text('Next'),
onPressed: () {
Navigator.push(context, MaterialPageRoute(builder: (context) => SecondRoute()),);
},
```

```
),
],
),
),
);
 */


class _ListViewNoteState extends State<ListViewNote> {
  List<Note> items;
  StreamSubscription<Event> _onNoteAddedSubscription;

  @override
  void initState() {
    super.initState();
    items = new List();
    _onNoteAddedSubscription = _firebaseRef.onChildAdded.listen(_onNoteAdded);
  }

  @override
  void dispose() {
    _onNoteAddedSubscription.cancel();
    super.dispose();
  }

  void _onNoteAdded(Event event) {
    setState(() {
      items.add(new Note.fromSnapshot(event.snapshot));
    });
  }
```

```
  @override
  Widget build(BuildContext context) {
  }
}
```

2. Home Automation Code

```
#include "FirebaseESP32.h"
#include <WiFi.h>
int i = 0;
char ssid[] = "TaRaPi";
char password[] = "Mdaa19rafi99";
FirebaseData firebaseData;

#define FlamePin 27// This is for flame sensor pin
#define relay1 26// This is for relay1 pin
#define relay2 25// This is for relay2 pin
#define MQ6 34//this is for gas sensor pin

int Flame=0,val1,val2;
boolean statuss = true;
int sensorvalue = 0;
boolean gasstate = true;
void setup()
{
  Serial.begin(115200);   //use serial0
  Serial.print("start..");
  connectWifi();
```

```
  Firebase.begin("leds-93e97.firebaseio.com",
"SjqzqXLyHYXoDLFqp222DntTagQKpzJ3PVC5nLKF");
   pinMode(FlamePin, INPUT);
   pinMode(MQ6, INPUT);
   pinMode(relay1, OUTPUT);
   pinMode(relay2, OUTPUT);
}
void loop(){
  receivedata();
  senddata();
  }
void receivedata(){
   // get value
 if (Firebase.getInt(firebaseData, "/relay1")) {
     Serial.print("light: ");
    Serial.println(firebaseData.intData());
    if  (firebaseData.dataType() == "int") {
      val1 = firebaseData.intData();
  }
 }
  if (Firebase.getInt(firebaseData, "/relay2")) {
    Serial.print("fan: ");
    Serial.println(firebaseData.intData());
    if  (firebaseData.dataType() == "int") {
      val2 = firebaseData.intData();
   }
  }
  if(val1==1){
    digitalWrite(relay1,HIGH);
```

```
    }
   else{
    digitalWrite(relay1,LOW);
    }
  if(val2==1){
    digitalWrite(relay2,HIGH);
    }
   else{
    digitalWrite(relay2,LOW);
    }
}
void senddata(){
  Flame = digitalRead(FlamePin);
  Serial.println(Flame);
  sensorvalue = analogRead(MQ6);
   if (Flame== LOW){
    Serial.print("Fire In the House!");
    Firebase.set(firebaseData, "/status/fire/status","Fire In the House");
  }
  else{
     Firebase.deleteNode(firebaseData, "/status/fire");
  }
  Serial.print("MQ6 : ");
  Serial.println(sensorvalue);
  if (sensorvalue > 900)
  {
   Serial.println("Gas leakage Detects");
    Firebase.set(firebaseData, "/status/gas/status","Gas leakage found");
```

```
  }
  else
  {
     Firebase.deleteNode(firebaseData, "/status/gas");
  }
  if(Flame==LOW||sensorvalue>900){
     Firebase.setBool(firebaseData, "/others",false);
  }
  else{
    Firebase.setBool(firebaseData, "/others",true);
    }
   delay(1000);
}
void connectWifi() {
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("..............................");
  Serial.println("WiFi Connected....IP Address:");
  Serial.println(WiFi.localIP());
  Serial.println("..............................");
}
```