Seer - A Computer Vision and Machine Learning Based Device for Visually Impaired

2015-FYP-01

Submitted by:

Muhammad Abdullah	2015-EE-166
Muhammad Awais Ismail	2015-EE-178
Muhammad Mehmood Ahmed	2015-EE-185
Saad Ali	2015-EE-190

Supervised by: Dr. Kashif Javed

Department of Electrical Engineering

University of Engineering and Technology Lahore

Seer - A Computer Vision and Machine Learning Based Device for Visually Impaired

Submitted to the faculty of the Electrical Engineering Department
of the University of Engineering and Technology Lahore
in partial fulfillment of the requirements for the Degree of

Bachelor of Science

in

Electrical Engineering.

Internal Examiner	External Examiner

Final Year Project
Coordinator

Department of Electrical Engineering

University of Engineering and Technology Lahore

Declaration

I declare that the work contained in this thesis is my own, except where explicitly stated otherwise. In addition this work has not been submitted to obtain another degree or professional qualification.

Signed: _____
Date: _____

Signed: _____
Date: _____

Signed: _____
Date: _____

Signed: _____
Date: _____

Acknowledgments

First of all, we would like to thank Allah Almighty, who gave us the strength and courage to accomplish this feat and complete the project. We would also like to thank the Department of Electrical Engineering, University of Engineering and Technology, Lahore for providing workplace to develop and test the ideas. We would also like to thank Dr. Kashif Javed, of the Department of Electrical Engineering, for his continuous support throughout the project.

Dedicated to
our beloved parents,
our honorable teachers,
and our trustworthy friends,
without whom,
this would not have
been possible.

Contents

List of Figures

List of Tables

Abbreviations

FYP	**F**inal **Y**ear **P**roject
VI	**V**isually **I**mpaired
AI	**A**rtifical **I**ntelligence
ML	**M**achine **L**earning
DL	**D**eep **L**earning
CV	**C**omputer **V**ision
NN	**N**eural **N**etworks
CNN	**C**onvolutional **N**eural **N**etworks
OD	**O**bject **D**etection
FR	**F**acial **R**ecognition
OCR	**O**ptical **C**haracter **R**ecognition
SS	**S**peech **S**ynthesis
TTS	**T**ext **T**o **S**peech
RPi	**R**aspberry **Pi**

Abstract

Vision is the most important and primitive tool for mankind to learn and interact with the environment. The significance of vision has skyrocketed in this current era of information technology. Sadly, there are millions of people in the world who have to live their lives in eternal darkness or with some sort of visual impairment. They rely on their family to fulfill their daily needs. We are trying to come up with a solution which can make the visually impaired people more independent in their daily chores. Visually challenged people use their sense of touch or someone elses help to identify everyday objects. Our proposed device will help the people with visual disabilities to recognize common objects in their line of sight. We want to allow them to identify familiar faces, everyday objects and recognize text that they come across in their daily life. We are using models based on machine learning and computer vision to input image through a camera and get the information about various objects in the image. The obtained information about the object is conveyed to the user in the form of audio. For object detection, we are using a pre-trained model which is trained on hundreds of thousands of images and we have fine-tuned it with our own collected dataset. The model being used is MobileNet-SSD which is based on Convolutional Neural Networks. The data collected by us spans around 30 categories with 40 to 50 images per category. With a train/test split of 80/20, we've achieved an accuracy of around 80% for object detection. For text recognition, the object containing text is first identified using a model called EAST Detector and then an OCR software called Tesseract is used to convert the image into machine recognizable text. The text detector is based on a deep neural network architecture and gives an accuracy of around 90%. In case of facial recognition, a combination of HAAR and HOG Classifier is being used to detect the faces while Nearest Means Classifier employing the vector embeddings created from our own custom dataset is being used to recognize them. The data collected by us spans around 10 persons with 50 images per person. With a train/test split of 80/20, weve achieved an accuracy of around 85% for face recognition. The major tools being deployed are Python, Numpy, Pandas, Scikit-Learn, Matplotlib, Tensorflow, Keras, OpenCV and ImUtils.

Chapter 1

Problem Statement

In our planet of 7.4 billion humans, 285 million are suffering from some form of visual impairment. Out of those people, 39 million people are those who are totally blind, meaning that they are void of any vision, and 246 million have mild or severe visual impairment. According to a prediction, the number of blind people will reach to 75 million and that of visually impaired will reach to 200 million by the year 2020.

Visual impairment is one of the biggest limitations for humanity, especially in these days and age, as this is the age of information. Information is communicated more often through text (electronic and paper media) as compared to voice. We highly rely upon our vision as compared to other senses for our daily life tasks. So it is quite hard for visually impaired people to perform simple tasks [1].

The device we have proposed aims to help people with visual impairment. As it is hard for visually impaired people to identify different objects, our device will help them by using speech to identify these objects. In this project, the device will help the user to convert object from image to speech by using deep learning and voice synthesis helping them identifying the object, text or face. The device will have features like text-to-speech, text recognition, facial recognition and object recognition.

Chapter 2

Literature Review

On Earth, 285 million people are visually impaired, out of which 39 million people are living in complete darkness i.e are totally blind and 246 million people on earth are suffering from some sort of vision deficiency which may be mild or severe. These numbers will keep on increasing, up-to nearly 75 million blind and 200 million people with visual impairment by the year 2020. As it is hard to identify different things for visually impaired people like objects, text and faces, our device will help them by using speech to identify these products.

The trend of machine learning is growing in these past few years. New machine learning algorithms are being researched and they are getting better and more efficient each passing day. Deep Learning stems from machine learning. Here algorithms which are inspired by the working of neurons in brain, are used. These algorithms simulate the working of brain by means of different layers of neurons and constructing structure of neurons into a neural network. An input is provided which passes through neural pathways and an output is returned. Convolutional neural networks are used for the image input. These neural nets are designed with the assumption that the input will be image which results in a more efficient and accurate classifier.

We will be using these algorithms for our device to detect objects through images. After detection we will use several deep neural network techniques to identify the detected objects.

2.1 Machine Learning

Machines are getting smarter everyday with machine learning which is a branch of artificial intelligence. Statistical techniques are used here to give machines the ability to learn tasks which are hard for machines to perform with traditional programming. Data is used to improve the working of the machine on a specific task. Computational statistics

is used in machine learning, which focuses on prediction-making.

There are two types of machine learning algorithms.

2.1.1 Supervised Algorithms

These algorithms are supervised by a data analyst with machine learning skills. Both input and desired output is provided by the analyst, in addition during algorithm training the feedback is refined by observing the accuracy of predictions. The features or variables analyzed by the model are determined by data scientists, which are used to develop predictions. The algorithm will apply what was learned in the training process to the new data after completing its training.

2.1.2 Unsupervised Algorithms

These algorithms are not trained with desired output data. Instead, an iterative approach is used to improve the predictions by reviewing the relevant data. Unsupervised learning algorithms are also usually called neural networks. More complex processing tasks such as image recognition, speech-to-text and natural language generation etc are performed by using unsupervised machine learning algorithms. After processing through the large amount of training data subtle correlations between many variables are determined. Accurate predictions can be made on new data by using the trained model. These algorithms are required to be trained by using massive training data.

There are a lot of machine learning algorithms out there and new algorithms are being researches every day. There are complex as well as some simple algorithms available. Let us look at the most commonly used models here:

- **Correlation Matrix:** There is a class which finds correlation between variables generally and use these patterns to classify new data.
- **Decision Trees:** Observations about certain actions are observed and an optimal path is identified to arrive at a desired outcome.
- **K-Means Clustering:** In this model, specified number of data points are grouped according to their characteristic.
- **Neural Networks:** These deep learning models are trained using large amount of training data until the model is optimized for the processing of data in the future.
- **Reinforcement Learning:** A process is completed by the model, iterating various times over it. Steps producing correct outcomes are rewarded and steps that generate undesired outcomes are penalized until the algorithm achieves the optimal process.

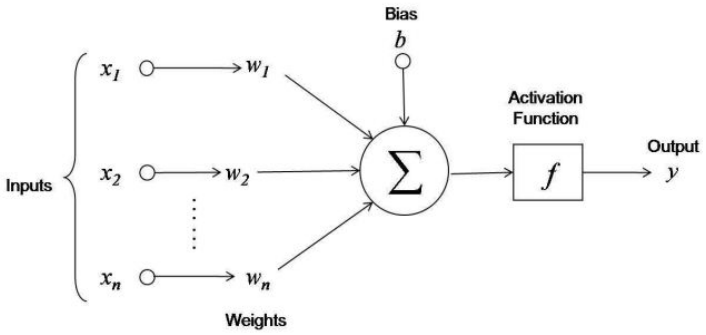

FIGURE 2.1: Neuron in Neural Net

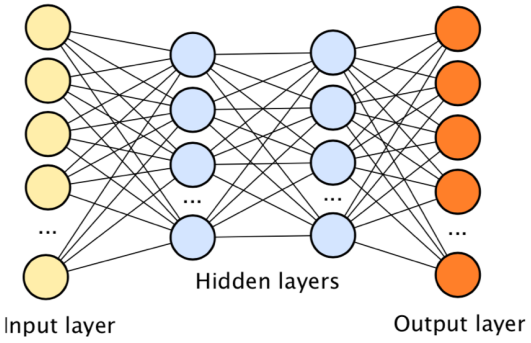

FIGURE 2.2: Neurons Hidden Layers

2.2 Deep Learning and Neural Networks

Deep Learning is a branch of machine learning which uses algorithms which are structured to simulate the functioning of brain. These algorithms simulate the working of brain by means of neurons and constructing neural networks. Therefore they are also referred to as artificial neural networks. Artificial neural networks learn to classify by processing records one at a time, and improve themselves by comparing their classification with the known actual classification of the record and making improvements accordingly. The errors from the classification of the first record are fed back and the model is altered in the second cycle and this process goes on until the model is able to provide optimal outcomes.

Roughly speaking, a neuron in an artificial neural network is

1. A set of some inputs (xi) and their corresponding weights (wi)

2. An active function whose job is to add up the weights and map the obtained results to some output (y) (Figure 2.1)

3. Neurons are organized into layers (Figure 2.2)

There are only two visible layers in the neural network i.e input layer and output layer.

Hidden layers are layers which are in between the input and output layer. These layers take input data at the first input layer and transform it into an output class at the final output layer. When the input data passes through the neural network, a specific value is assigned to each of the output nodes and the result will be the node with the highest value.

In the training phase, we know the correct class of each test data. Hence, we can assign correct value 1 to the output node with correct class and 0 to the node with incorrect class. In this way, the correct value is compared with the calculated value of output node of neural network. An error term is calculated for every node. These values are used to adjust the weights in the hidden layers of the neural network and values of the weights are adjusted such that the output values may approach the correct value with minimum error.

Neural Networks learn by means of an iterative process in which each test data case passes through the net at a time and the output is compared with the correct value and weights and adjusted accordingly. This cycle repeat for millions of times to adjust the weights and to get to the correct answer.

The neural networks after being trained for a long time become able to put a correct label on each input class. The advantage of neural networks is the ability to handle noisy data and being able to classify the data which they have not encountered in the training phase.

2.3 Convolutional Neural Networks

The field of machine learning has taken a dramatic twist in recent times, with the rise of the Artificial Neural Network (ANN). These biologically inspired computational models are able to far exceed the performance of previous forms of artificial intelligence in common machine learning tasks. One of the most impressive forms of ANN architecture is that of the Convolutional Neural Network (CNN). CNNs are primarily used to solve difficult image-driven pattern recognition tasks and with their precise yet simple architecture, offers a simplified method of getting started with ANNs [2].

Convolutional Neural Networks are neural networks designed with the assumption that the input will be an image. They are similar to normal neural network with learnable weights and basis. The received inputs, by the neuron, passes through dot product and may follow by a non-linearity. The first layer of the convolutional neural network is comprised of pixels values of the input image and the output layer has class scores. The whole network produces a single output score: from the image pixels at input layer and class scores at output.

Conv Net only receive inputs as images, allowing us to design the architecture specified for images. The amount of parameters in the network is vastly reduced and the forward function becomes more efficient and accurate.

Image are the inputs in Conv Net and they allow us to structure the model in a more appropriate and efficient way. Unlike a regular neural network, neurons in a Conv Net are three dimensional i.e having width, height and depth (depth here means the third dimension of vector) In case of CIFAR-10 the output layer has 1x1x10 dimensions, because at the end of Conv Net model the output layer has a vector containing the class scores in a vector.

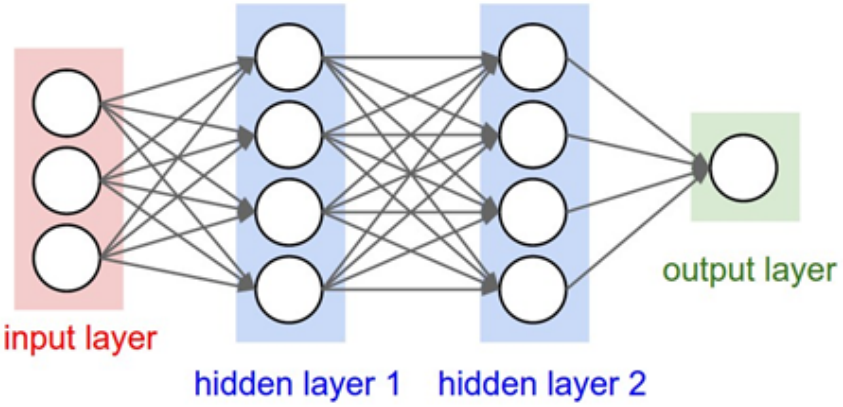

FIGURE 2.3: 3-Layer Neural Net

A regular 3-layer neural network is shown in the Figure 2.3 and 2.4. Neurons are arranged in 3D (width, height, depth) in a Conv Net as shown in the Figure 2.4. A 3D input volume transforms to a 3D output volume by a differentiable function. In this example, the red input layer holds the image which will be a three dimensional volume have width and height consisting of the position of the pixel on the screen while the depth of the layer will represent the color of the image which will be intensity of the three colors: Red, Green, Blue.

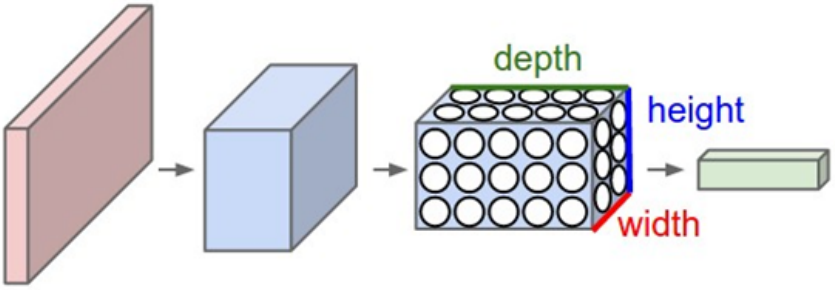

FIGURE 2.4: Three Dimensional ConvNet Visualization

Differentiable function is used to transform one volume to another in a Conv Net layer. There different type of layers are used to build a Conv Net architecture: Convolutional

Layer, Pooling Layer, and Fully-Connected Layer. These three layers are stacked to form a Conv Net architecture.

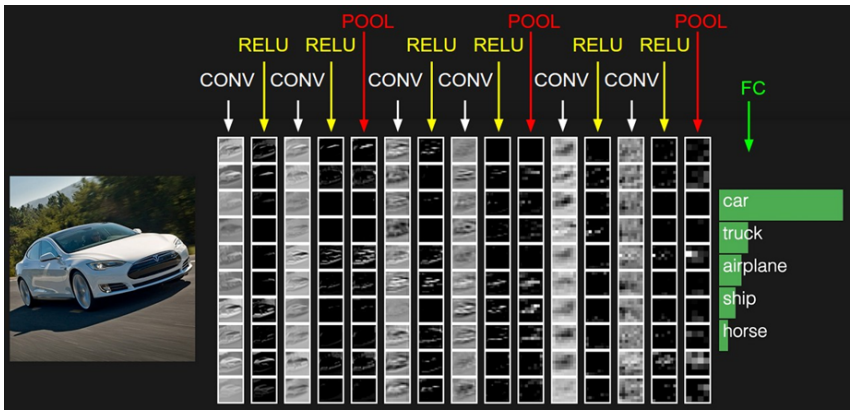

FIGURE 2.5: Data Passing Through Layers and Output

There is an example of Conv Net model in the Figure 2.5. The raw image is pixel data is forming the input layer at the left and final volume stores the class score at the right. Each column represents a volume. Each volume has been laid out as slice in rows to allow us to easily visualize the 3D volume of layer. Here only sorted top 5 scores are displayed but the last layer contains score for each class in the form of a vector. Each label is printed in the figure. A small VGG Net is shown in this architecture.

In summary:

- A Conv Net model converts an input image into an output class score.
- There are few types of layers in Conv Net (e.g. CONV/FC/RELU/POOL etc).
- Through each layer, a 3D volume is passed from input side and output is produced as well in 3D volume after passing through a differentiable function.
- There may be parameters in some layers.
- Additional hyper parameters are part of some layers (e.g. CONV/FC/POOL possess hyper parameters, RELU doesnt have them).

2.4 Image Processing

Various algorithms and methods are used for the purpose of Image processing and detection [3, 4] as mentioned previously some of them are given here.

2.4.1 Histogram of Oriented Gradients (HOG)

In computer vision and image processing for object and human detection, Histogram of oriented gradients (HOG) is used quite often. The method is based on evaluating well-normalized local histograms of image gradient orientations in a dense grid [5]. The basic idea is that local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions. For training and testing the dataset, the main processes applied are given separately.

2.4.1.1 Data Training Method

The method followed for training data through HOG is illustrated in Figure 2.6.

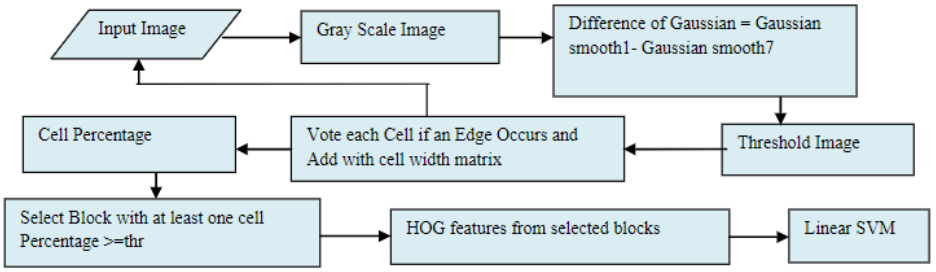

FIGURE 2.6: Data Training for HOG

2.4.1.2 Data Testing Method

Machine learning has to deal with big and uncertain data [6]. The method followed for training data through HOG is illustrated in Figure 2.7.

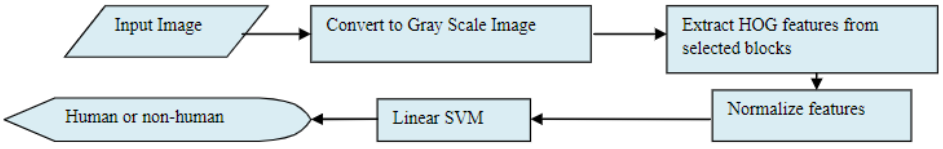

FIGURE 2.7: Data Testing for HOG

2.4.2 Deep Residual Learning

Deep neural networks are not easily trained. A residual learning framework is used to ease the procedure of data training. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs [7].

Different accuracies have been measured on different datasets using deep residual learning. Some of them are on ImageNet datasets obtained by varying the number of layers. Also

it has shown great results on CIFAR-10 by varying the number of the layers to 100 and 1000 layers.

2.5 Speech Synthesis

Speech is one of the oldest and most natural means of information exchange between humans. Over the years, attempts have been made to develop vocally interactive computers to realize voice/speech synthesis. Obviously such an interface would yield great benefits. In this case a computer can synthesize text and give out a speech. Text-to-speech synthesis is a technology that provides a means of converting written text from a descriptive form to a spoken language that is easily understandable by the end user [8].

A text-to-speech system (TTS) outputs speech if a text of common language is fed into it. Recorded speech for different texts is stored in a database and used to generate speech by comparing them. There are different systems which differ by the size of their database of speech units. For high quality output, words or entire sentences are stored in certain domains. On the other hand, a synthesizer can be used to store model of human audio track and different human voice characteristics to output a synthetic voice output.

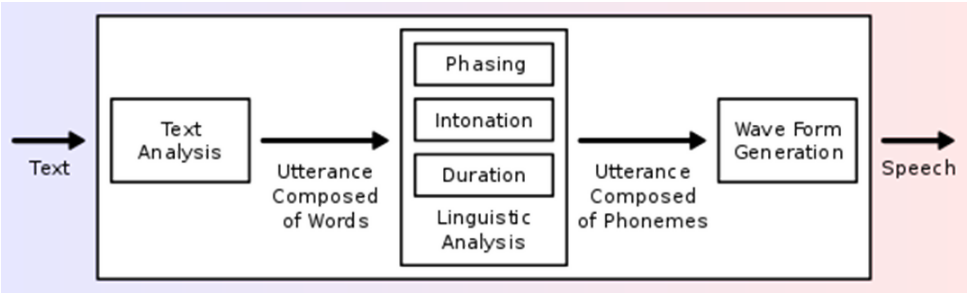

FIGURE 2.8: Overview of A Typical TTS

A typical text-to-speech system (Figure 2.8) can be divided into two types:

1) Front-end performs two major tasks. The raw text is converted which may contain symbols into written out words. This process is usually called pre-processing or text normalization. Then phonetic transcription is assigned to each word by the front-end system. The text is then divided into prosodic units such as sentences, phrases and clauses. Transcriptions are assigned to words in a Text-to-phoneme process is the process. The front end outputs the prosodic information and phonetic transcription which makes up the symbolic language. This becomes input to a speech synthesizer.

2) The back-end is a synthesizer which is responsible to convert the symbolic language produced by the front end into sound. This part may contain computation of target prosody which is applied to the output.

2.5.1 PyTTSx

A Text-to-Speech API is a speech synthesizer that can convert normal language text into speech. The process of synthesized speech involves construction using pieces of recorded speech stored in a database. The biggest challenge with speech synthesis is text normalization. Texts are full of abbreviations, numbers and heteronyms. The ability of a speech engine to recognize these patterns define its efficiency. The engines like PyTTSx for python based applications offers a wide range of features such as different voices, changing voice rates, understanding the numeric system and punctuations. Thus, by bundling this TTS engine with the image processing module, a complete system can be made [9].

The working of a TTS engine is shown in Figure 2.9.

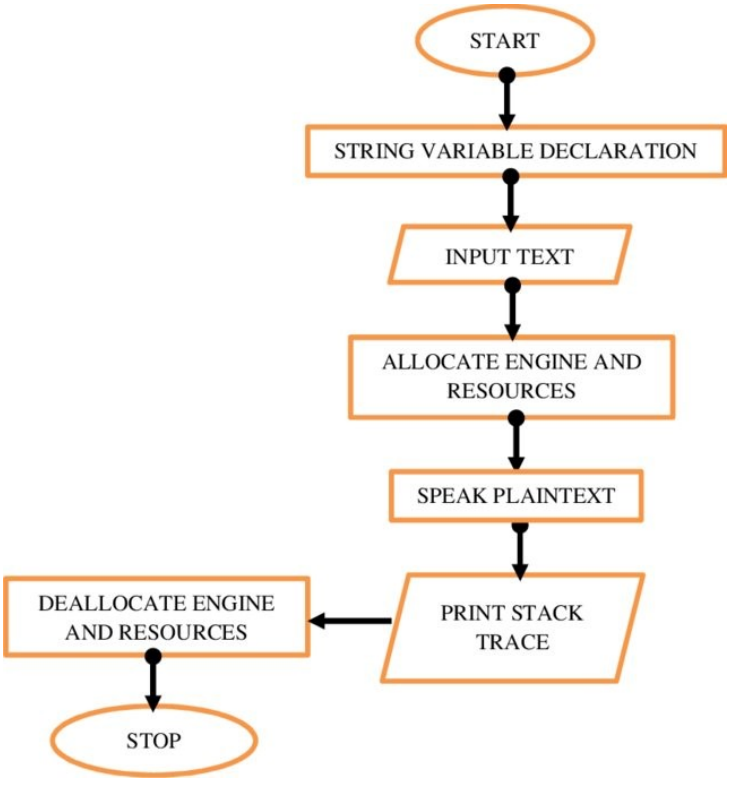

FIGURE 2.9: Working of TTS Engine

2.6 Raspberry Pi

The Raspberry Pi (Figure 2.10) is comprised of a set of small single board computers. A complete computer shrinks to a circuit board in single board computer. It includes a microprocessor, memory, I/O and required features for a functional computer. Single-board computers were designed for educational purposes and embedded system as demonstration and development systems.

FIGURE 2.10: Raspberry Pi

All the computer functions incorporated on a single printed circuit board in home PC and portable computers.

The Raspberry Pi is a small low cast computer. It can be interfaced with a keyboard, monitor and other peripherals to function as a portable computer. RPi is widely used in projects as a portable computer. It is low cost and easily available. We can interface camera with the microcontroller and connect an external SD card to increase the storage. RPi comes with Linux as an operating system but many other operating systems can be used on it. Most of the software RPi uses are open source or free. So overall using RPi is really cost effective.

Chapter 3

Methodology

The basic methodology deployed is as follows :

- *Step 1*: Image Input
- *Step 2*: Image Processing
- *Step 3*: Machine Learning
- *Step 4*: Speech Synthesis
- *Step 5*: Audio Output

3.1 Image Input

Selecting the right module to get the images consists of discrete options. Quality of camera decides the quality of image captured. We are using A4Tech Webcam with a resolution of 16 MP to take pictures and video on the Raspberry Pi and using suitable operating system (Linux) in setting the properties of the acquired image. Another alternative is to use Raspberry Pi camera module which is available in both 5 MP and 8 MP version. A sample image after input is shown in Figure 3.1.

3.1.1 Working

An image is captured with the help of the webcam, which is then fed into the machine learning model running on Raspberry Pi.

3.1.2 Constraints

The constraints for this portion are:

- Captured image should not be blurred.

- Image should be well-lit.
- There should be distinction between foreground and background.
- The target object should be well-focused.

FIGURE 3.1: Post Image Input

3.2 Image Processing

If we capture images from a sensor (camera) then without using any software we have to go through some fundamentals steps for image processing and extracting some useful information. In image pre-processing, image data recorded either by a sensor or some other source identified errors related to geometry and brightness values. Appropriate mathematical tools are used for correction of errors. Visual impact of image is improved by image enhancement which is done by changing the brightness value of pixels of the image. A sample image after processing is shown in Figure 3.2.

3.2.1 Working

The captured image is pre-processed before being fed into the machine learning model. Processing can consist of segmentation, color conversion, binarization, filtering, normalization, mean subtraction, scaling and resizing.

3.2.2 Constraints

The constraints for this portion are:

- Image size
- Processing and storage

FIGURE 3.2: Post Image Processing (Grey Scaling)

3.3 Machine Learning

Machine learning consists of generic algorithms that do manipulation on set of data. Instead of writing or compiling a lengthy code, machine learning algorithms deduce its own logic depending upon the data fed. Example of machine learning algorithm is in emails.

3.3.1 Working

The core subsystem of the project uses machine learning and deep learning. The models used in this subsystem must have good precision, accuracy, speed, finiteness and generality. Three different types of detection/recognition are being done: Object Detection, Facial Recognition and Text Recognition.

3.3.2 Object Detection

In this part, objects from around 30 different categories are first detected and then recognized. The model being used for object detection is MobileNet SSD [10].

3.3.2.1 Algorithms

The most popular algorithms for object detection and recognition are:

- RCNN (Region-based Convolutional Network)
- SSD (Single Shot Detectors)
- YOLO (You Only Look Once)

3.3.2.2 Constraints

The constraints for this portion are:

- Size of objects
- Closeness and speed of objects
- Limited number of objects
- Processing power and speed
- Training time

3.3.3 Facial Recognition

In this part, faces of around 10 different are first detected and then recognized. The model being used is a combination of HOG and HAAR algorithms [11].

3.3.3.1 Algorithms

The most popular algorithms for facial detection and recognition are:

- HAAR Cascade (by Viola Jones)
- DNN (Deep Neural Network by OpenCV)
- HOG (Histogram of Oriented Gradients)
- MMOD (Maximum Margin Object Detection)

3.3.3.2 Constraints

The constraints for this portion are:

- Orientation and motion of faces
- Size of faces
- Limited number of faces data
- Processing power and speed

3.3.4 Text Recognition

In this part, text appearing in English language is first detected and then recognized. The model being used is a combination of EAST and Tesseract OCR.

3.3.4.1 Algorithms

The most popular algorithms for text detection and recognition are:

- EAST (Efficient and Accurate Text Detector)
- CRNN (Convolutional Recurrent Neural Network)
- CTPN (Connectionist Text Proposal Network)
- OCR (Optical Character Recognition)

3.3.4.2 Constraints

The constraints for this portion are:

- Orientation and movement of text
- Viewing angle
- Text size and font
- Language and script

3.3.5 Working of CNNs

Convolution neural networking is the smarter way of dealing with images in machine learning. It consists of following steps as compared to simple neural networking or deep neural net. A sample image after detection is shown in Figure 3.3.

- **Break The Image Into Overlapping Image Tiles**
  Mostly a square section is passed over and the image is broken into several sections and each section is saved as separate image.
- **Input An Image Section Into A Small Neural Network**
  We will then feed each subsection of broken image into a small neural network.
- **Creating New Arrays**
  New arrays are created mimicking the original image data set array.
- **Down Sampling Using Max Pooling**
  Size of the array is reduced using max-pooling. The biggest number from a 2x2 array is kept.

- **Make A Prediction**
  The data from each array is used as an input and is feed to the next neural network. The final neural network obtained from cascading smaller neural networks will decide about resemblance of image with original image.

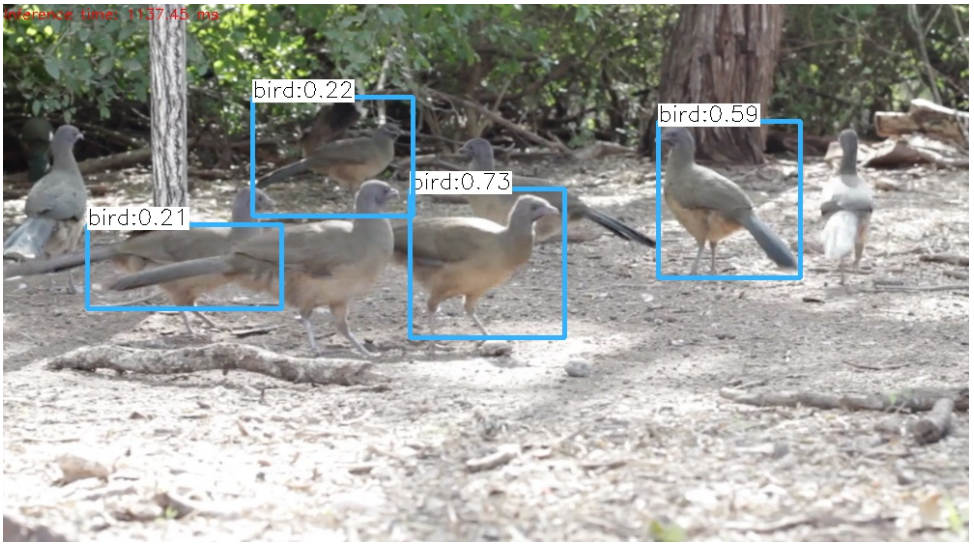

FIGURE 3.3: Post Machine Learning (Object Detection)

3.4 Speech Synthesis

There are different approaches for speech synthesis or TTS (Text to speech). Both open-source and proprietary softwares for TTS are available, but we went with open-source one.

3.4.1 Working

The result obtained from the previous subsystem is in the form of text. The text is to be converted to speech using some form of TTS. Popular TTS systems are eSpeak, gTTS, pyTTSx3 and Festival.

3.4.2 Constraints

The constraints for this portion are:

- Text in broken words/phrases
- Non-English text
- Text-to-speech conversion speed
- Robotic voices

3.5 Audio Output

After object detection through image processing and text to speech synthesis, the final output has to be conveyed to the user.

3.5.1 Working

The voice generated from the previous subsystem is fed to the user via some audio device such as earphones.

3.5.2 Constraints

The constraints for this portion are:

- The volume should be audible.
- The speed should be comprehensible.
- The sound be somewhat natural.

Chapter 4

System Architecture

4.1 System Architecture

The system architecture of the project is quite unified and streamlined. The architecture consists of five major portions of the project. Each of these portions come one after another. Each portion has an input and an output. The output of one portion becomes the input to the next one and so on.

The architecture consists of following five major portions:

- Image Input
- Image Processing
- Machine Learning
- Speech Synthesis
- Audio Output

4.1.1 Image Input

This is the first portion of the architecture. In this portion, the image is taken as input with the help of a camera. The output of this portion is the raw image.

4.1.2 Image Processing

This is the second portion of the architecture. The input of this portion is the raw image. In this portion, the image is processed by image segmentation and feature extraction. The output of this portion is the processed image.

4.1.3 Machine Learning

This is the third portion of the architecture. The input of this portion is the processed image. In this portion, machine learning algorithms are applied to detect objects from images. The output of this portion is the detected object.

4.1.4 Speech Synthesis

This is the fourth portion of the architecture. The input of this portion is the detected object. In this portion, the resultant data in the form of text is converted into speech. The output of this portion is the speech from text.

4.1.5 Audio Output

This is the last portion of the architecture. The input of this portion is the speech generated. In this portion, the detection result in speech form is spoken via some audio device. This marks the end of the architecture.

The system architecture is illustrated by means of a block diagram in Figure 4.1.

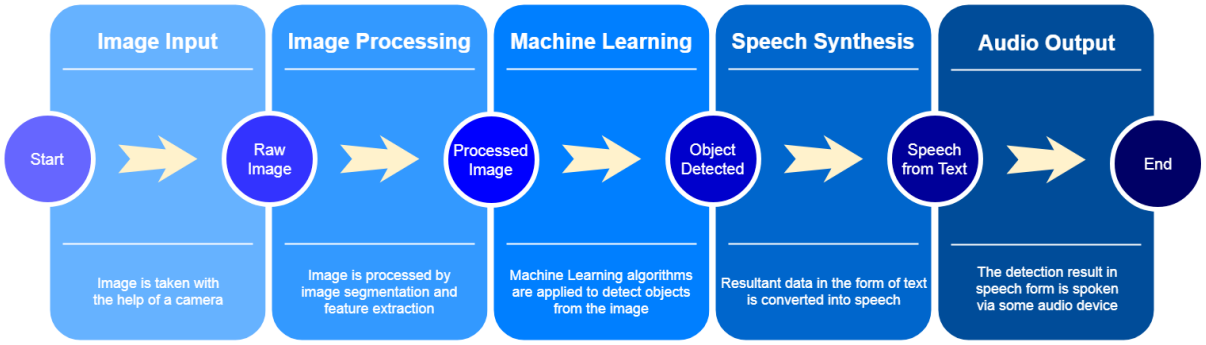

FIGURE 4.1: System Architecture (Block Diagram)

4.2 Subsystem Architecture

The system architecture is further broken down into smaller elements to give the subsystem architecture. Just like the system architecture, the subsystem architecture is also quite unified and streamlined. Each of the major portions of the architecture is further subdivided into smaller portions.

The input image taken via camera after some processing is fed to the machine learning models inside the microcontroller to generate detection result which is converted into speech and fed to the user in the form of audio via a pair of earphones. Besides that, some training is also done on the collected dataset to generate custom models for detection and recognition.

The subsystem architecture is illustrated by means of a flow chart in Figure 4.2.

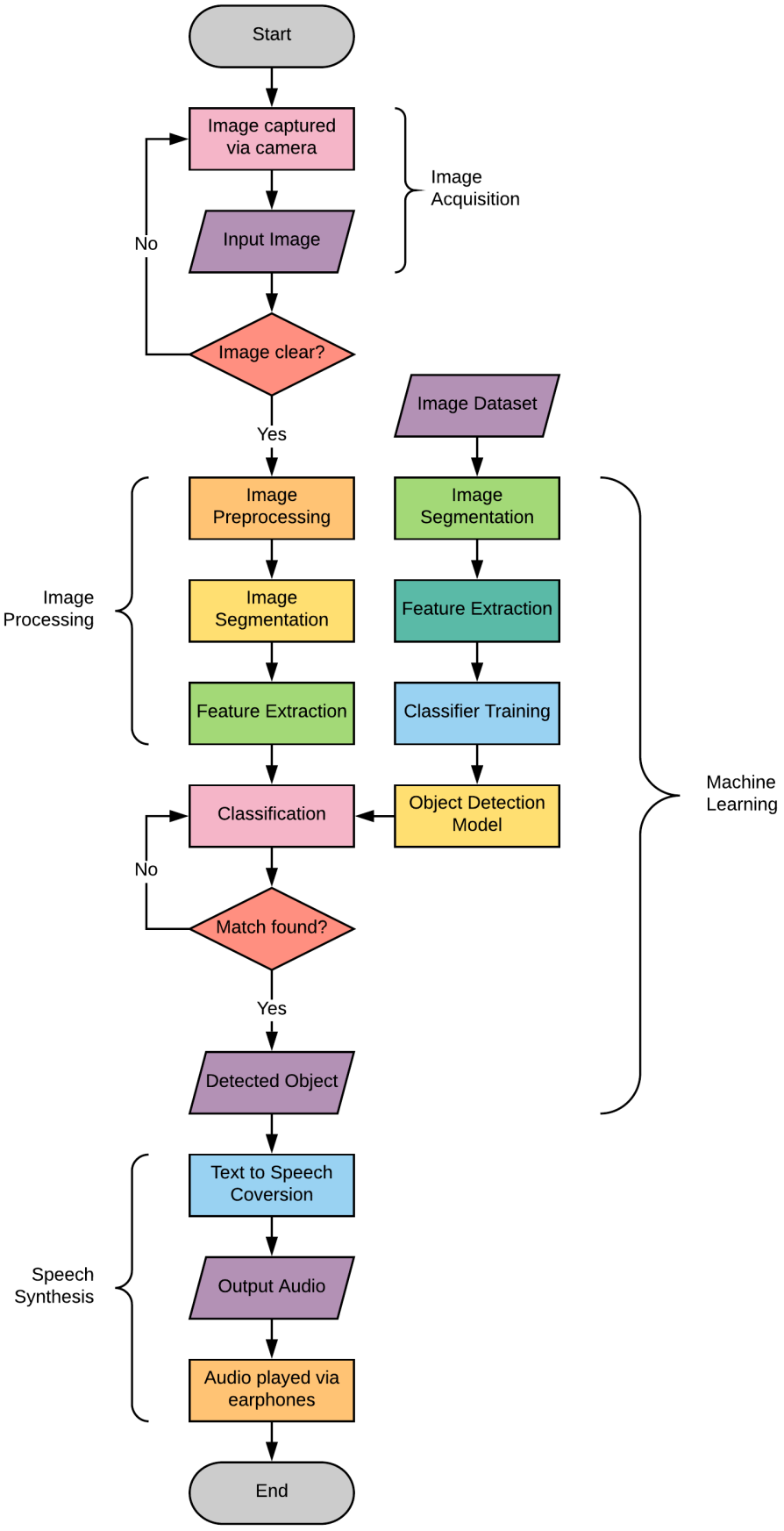

FIGURE 4.2: Subsystem Architecture (Flow Chart)

4.3 Functional Description

The basic functionality of the project is quite easy to understand.

- The system consists of one power switch and three push buttons.
- When the power switch is pushed, the device is turned on. This is indicated by the status LED. Right after boot, the code starts running.
- The three push buttons are mapped to the three different modes for object, facial and text recognition respectively.
- When button 1 is pushed, the objects infront of the camera are detected and recognized, and the result is spoken via earphones.
- When button 2 is pushed, the faces infront of the camera are detected and recognized, and the result is spoken via earphones.
- When button 3 is pushed, the text infront of the camera is detected and recognized, and the result is spoken via earphones.

Chapter 5

Detailed System Design

5.1 System Design

The system design for the project is quite evident from its functionality. A simple yet comprehensive design has been made for the project and it has been followed throughout the completion of the project. The basic design is shown in Figure 5.1.

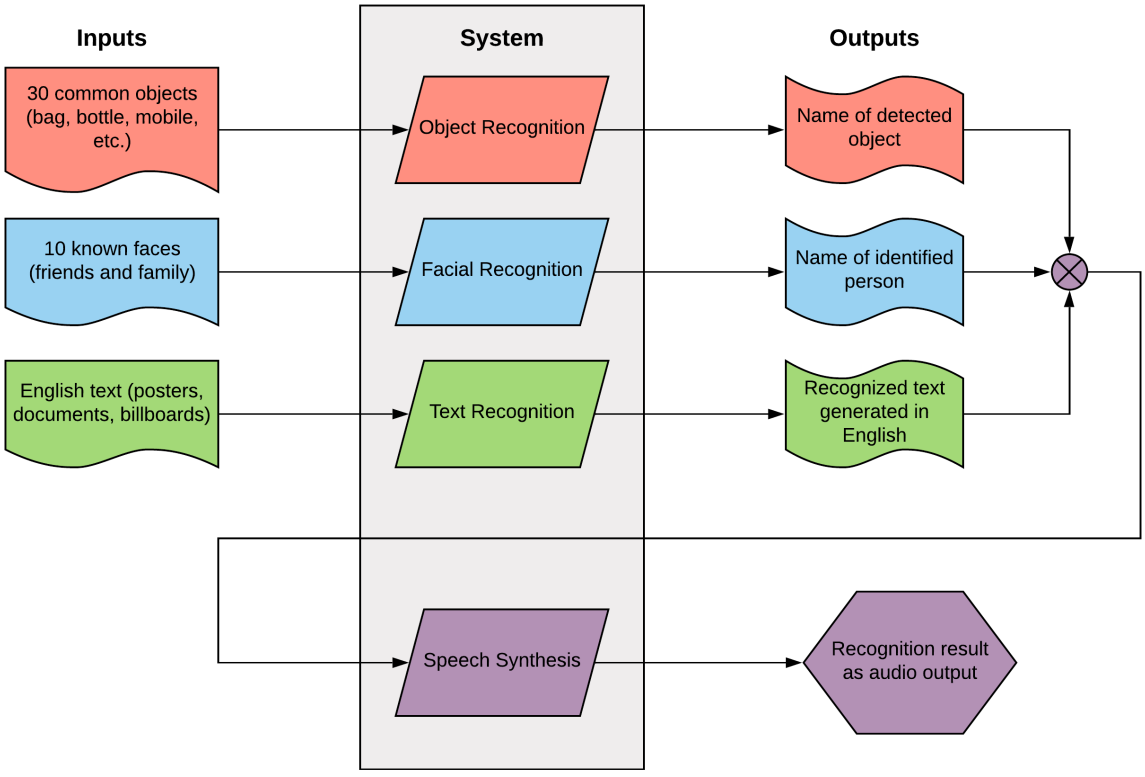

FIGURE 5.1: System Design

Since this is a machine learning project, huge amount is data is being manipulated in it. The different types of data included in the database of the project and their relationships and the flow of data are represented in the entity relationship diagram of Figure 5.2.

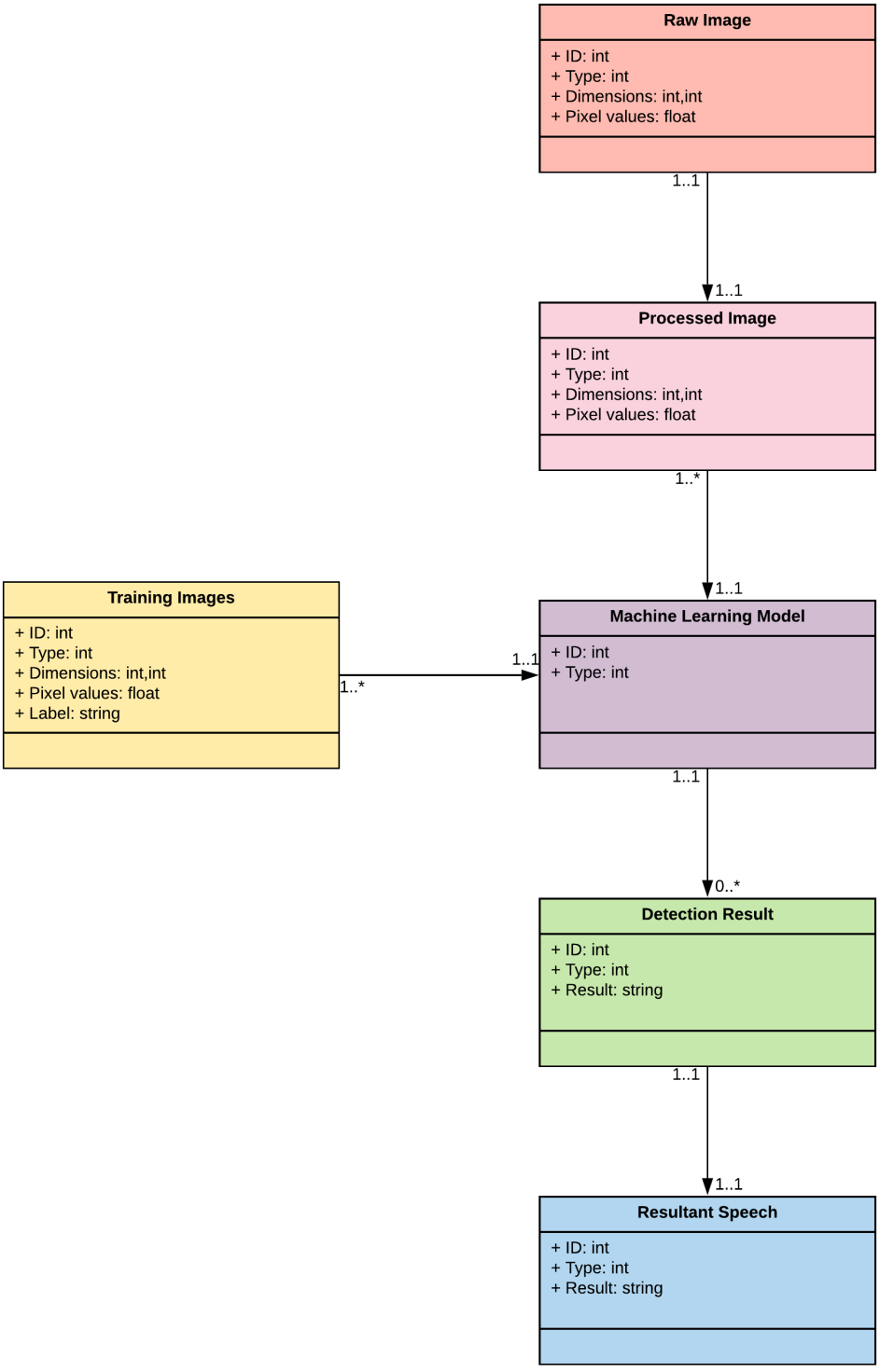

FIGURE 5.2: ER Diagram

The complete design of the system is made up of three major components:

- Object Recognition
- Facial Recognition
- Text Recognition

Each of these components is explained further below.

5.2 Object Recognition

The first component of our project is object recognition. In this component, the objects belonging to 30 different classes are first detected and then recognized as well. The input image is captured via camera and then passed to the machine learning model after some processing. The model being used for object detection and recognition is MobileNet SSD, which is a state of the art model for object detection developed by Google. This model has been pre-trained on more than a 100,000 images from the Microsoft COCO dataset which spans over 80 classes. This model has been re-trained by us on custom collected dataset of around 1500 images spanning 30 categories with 50 images per category. The data was divided into a train/test split of 80/20. After the object is recognized by the model, the result is converted into speech and conveyed to the user via earphones in the form of audio.

5.3 Facial Recognition

The second component of our project is facial recognition. In this component, the faces belonging to 10 different persons are first detected and then recognized as well. The input image is captured via camera and then passed to the machine learning model after some processing. The model being used for facial detection and recognition is a combination of HAAR and HOG algorithms. The HAAR model is used for detecting the faces and has been pre-trained on numerous faces of different types. The HOG algorithm used for recognizing faces is trained by us on custom collected dataset of around 500 images spanning 10 persons with 50 images per person. The data was divided into a train/test split of 80/20. After the face is recognized by the model, the result is converted into speech and conveyed to the user via earphones in the form of audio.

5.4 Text Recognition

The third component of our project is text recognition. In this component, the text belonging to English language is first detected and then recognized as well. The input image is captured via camera and then passed to the machine learning model after some

TABLE 5.1: List of classes for object detection

Object Classes		
Bag	Bed	Bike
Billboard	Bird	Book
Bottle	Bowl	Building
Car	Cat	Chair
Cup	Dog	Door
Fan	Keys	Laptop
Pen	Person	Phone
Plant	Rickshaw	Shoes
Switchboard	Table	Tree
TV	Wallet	Watch

processing. The model being used for text detection and recognition is a combination of EAST and Tesseract. The EAST model is used for detecting the text and has been pre-trained on various datasets for English and Chinese language. Tesseract is a state of the art OCR software developed by Google and is used for recognizing the text. After the text is recognized by the model, the result is converted into speech and conveyed to the user via earphones in the form of audio.

5.5 Dataset and Training

Custom local dataset was collected for object and facial recognition and the selected models were trained on our own collected dataset.

5.5.1 Object Detection

5.5.1.1 Dataset

The dataset collected for object detection is around 1500 images. This dataset spans over 30 classes of objects which are common in our daily life. All of these categories have around 50 images each.

The list of classes for object detection is given in Table 5.1.

5.5.1.2 Training

After the collection of dataset, the next step was to train our model on this dataset. The model selected for object detection was MobileNet SSD. If the model was retrained completely, it would have taken a lot of time and resources, and still wouldn't have produced satisfactory results due to the limited amount of training data. So instead of training the model from scratch, we used the model pre-trained on Microsoft COCO dataset and retrained it on our own custom dataset. The data was divided into a train/test

Table 5.2: List of persons for facial recognition

Persons
Abdullah
Awais
Mehmood
Saad
Usman Ali
Arham
Usman Jutt
Sohaib
Sabih
Zirsha

split of 80/20, giving a total of 1200 images for training, and 300 images for testing. The framework used for training was Tensorflow by Google. All of the training was done in the cloud, using Colaboratory, which is Google's platform for deploying machine learning models online.

5.5.2 Facial Recognition

5.5.2.1 Dataset

The dataset collected for object detection is around 500 images. This dataset spans over 10 persons belonging to the family or friends of the user. The data contains around 50 images per each person.

The list of persons for facial recognition is given in Table 5.2.

5.5.2.2 Training

After the collection of dataset, the next step was to train our model on this dataset. The model selected for facial recognition was a combination of HAAR and HOG algorithms. For facial detection, the pre-trained model of HAAR cascade was used. While for facial recognition, the HAAR algorithm was used to train the model. The data was divided into a train/test split of 80/20, giving a total of 400 images for training, and 100 images for testing. The training was done in Anaconda, which is an environment for machine learning and data science applications.

Chapter 6

Implementation and Testing

6.1 Software

The final software consists of a python code written specifically for Raspberry Pi which runs automatically when the device boots up.

All the three models for object, facial and text recognition are initially loaded into the memory. Then the camera is turned on. Finally an infinite loop is started in which the system waits for either of the three interrupt calls, and enters the respective code portion.

Each code portion takes the current frame and passes it through the respective model to get the detection results. Those results are then converted to speech and fed into the earphones.

6.2 Hardware

The complete hardware is divided into two distinct portions.

The first portion of hardware (Figure 6.1) consists of a Raspberry Pi 3B+ microcontroller enclosed in an acrylic case. The casing is further embedded with a cooling fan on the top and heat sinks on controller chips for the purpose of cooling. Furthermore, a single power switch for the purpose of turning the hardware on and off is installed. Three push buttons are also installed with the help of jumpers on for the purpose of interrupt generation in the context of mode switching. An indicator LED for power is also installed. Connection between power bank and microcontroller is established through a USB cable and a switch which work as a power source.

In the second portion of hardware (Figure 6.2), a small webcam and a set of earphones are fitted on a standard pair of glasses, and further connected to microcontroller in the first hardware portion via cables.

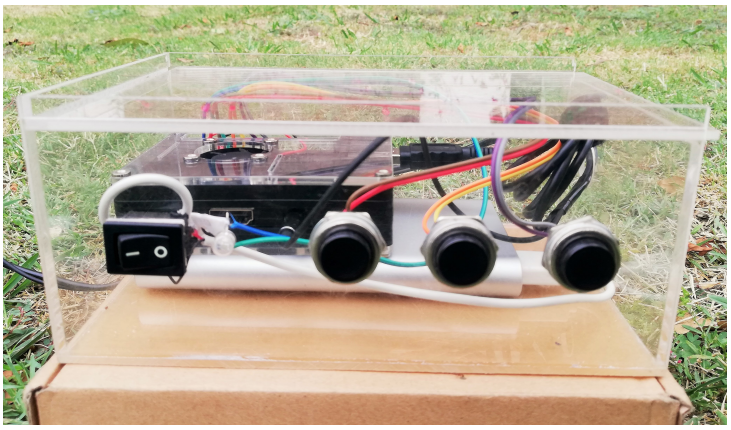

FIGURE 6.1: Hardware Portion 1

FIGURE 6.2: Hardware Portion 2

The final hardware consists of an acrylic box containing the microcontroller and power bank along with three push buttons and a power switch, and a pair of glasses with webcam and earphones attached to it.

The complete list of components used in the hardware is given in Table 6.1.

6.3 Testing

Different models for object, facial and text detection and recognition employing various algorithms were run both in simulation mode. The simulation and testing was being done on Laptop/PC running either Windows or Ubuntu (Linux) operating system. The models that performed best in simulation were then run on standalone hardware.

The criteria for performance was both speed as well as accuracy. All the models were tested on both still images as well as live video stream. The accuracy results were calculated by using confusion matrix to figure out the amount of true positives and false negatives. The speed results were calculated by noting the time taken for each image or video frame to be processed and recognized.

TABLE 6.1: List of components

Components	Quantity
Raspberry Pi 3B+	1
Xiaomi Power Bank (10,000 mAh)	1
A4Tech Webcam PK-930H	1
Xiaomi In-ear Earphones	1
Standard Pair of Glasses	1
Acrylic Case for Raspberry Pi	1
Acrylic Case for Hardware	1
Cooling Fan	1
Heat Sinks	3
Male to Female Jumpers	10
Push Buttons	3
Switch	1
USB Cables	2
LED	1

6.4 Simulation

Simulation was done for object recognition, facial recognition and text recognition as well.

6.4.1 Object Recognition

Quite a few models were tried for object recognition.

6.4.1.1 YOLO V3

The accuracy achieved for YOLO v3 was around 80% while the speed reached around 1 fps. A sample image is shown in Figure 6.3.

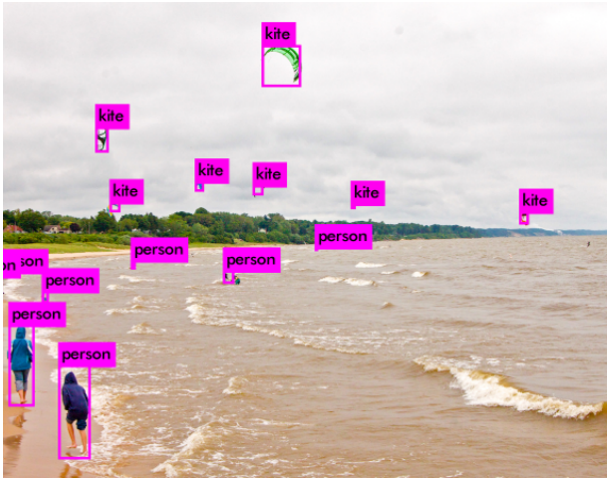

FIGURE 6.3: YOLO V3 Results

6.4.1.2 MobileNet-SSD V2

The accuracy achieved for MobileNet-SSD v2 was around 85% while the speed reached around 5 fps. A sample image is shown in Figure 6.4.

FIGURE 6.4: MobileNet-SSD V2 Results

6.4.2 Facial Recognition

Quite a few models were tried for facial recognition.

6.4.2.1 DLib

The accuracy achieved for DLib was around 90% while the speed reached around 1.1 fps. A sample image is shown in Figure 6.5.

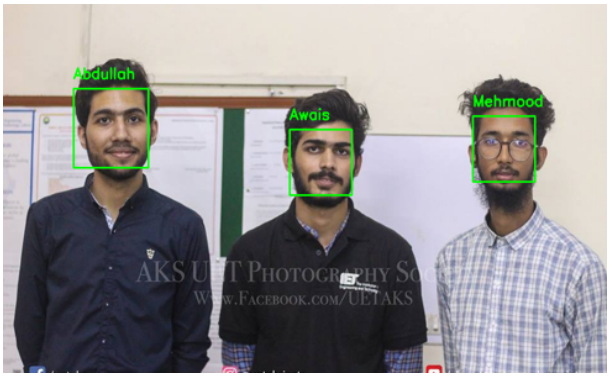

FIGURE 6.5: DLib Results

6.4.2.2 DNN

The accuracy achieved for DNN was around 60% while the speed reached around 6.1 fps. A sample image is shown in Figure 6.6.

6.4.3 Text Recognition

Some models were tried for text recognition as well.

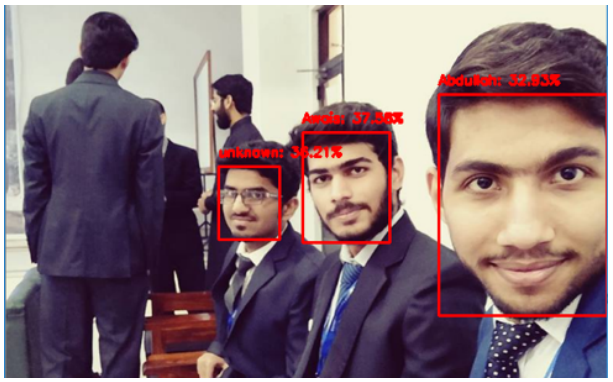

FIGURE 6.6: DNN Results

6.4.3.1 EAST and Tesseract

The accuracy achieved for EAST was around 95% while the speed reached around 2.9 fps. The accuracy achieved for EAST and Tesseract was around 85% while the speed reached around 1.4 fps. Sample images are shown in Figure 6.7 and Figure 6.8.

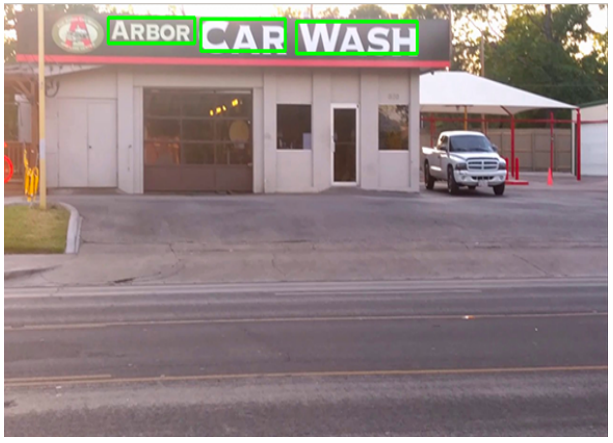

FIGURE 6.7: EAST Results

FIGURE 6.8: EAST and Tesseract Results

Chapter 7

Results

After all the models had been tested in simulation mode, the selected models were then transferred onto the standalone hardware. These models were also retrained on our custom collected dataset. Afterwards, these models were tested once again on the standalone hardware. The metrics were once again speed and accuracy. All the models were tested on both still images as well as live video stream. The accuracy results were calculated by using confusion matrix to figure out the amount of true positives and false negatives. The speed results were calculated by noting the time taken for each image or video frame to be processed and recognized.

7.1 Object Recognition

The model deployed on hardware for object detection and recognition was MobileNet-SSD v2. The accuracy achieved was around 80% while the speed reached around 1.8 fps. A sample image is shown in Figure 7.1.

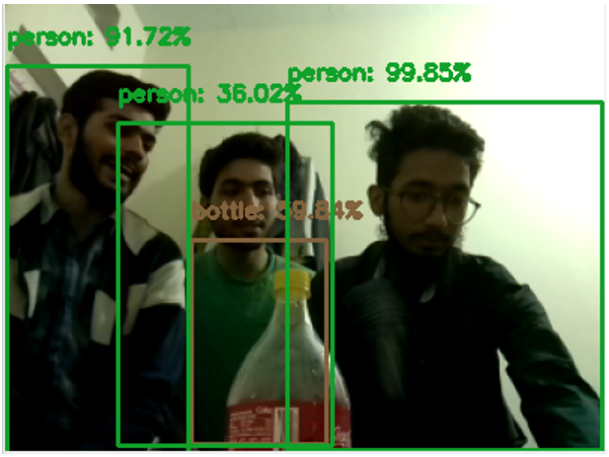

FIGURE 7.1: Hardware Results for Object Recognition

7.2 Facial Recognition

The model deployed on hardware for facial detection and recognition was HOG and HAAR combination. The accuracy achieved was around 85% while the speed reached around 1.6 fps. A sample image is shown in Figure 7.2.

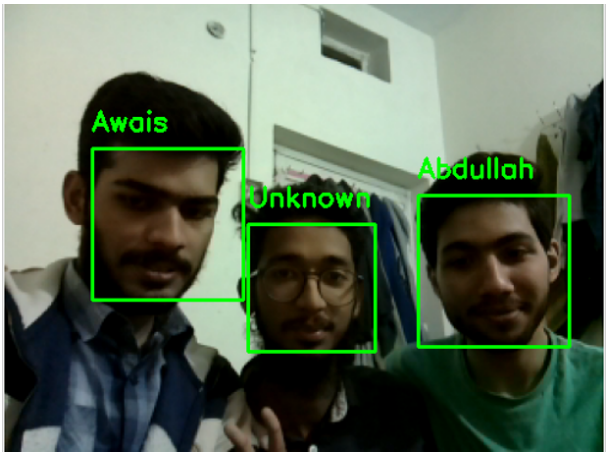

FIGURE 7.2: Hardware Results for Facial Recognition

7.3 Text Recognition

The model deployed on hardware for text detection and recognition was EAST and Tesseract combination. The accuracy achieved was around 90% while the speed reached around 1.4 fps for just detection while it took around 4 to 5 seconds to complete recognize the text in the image and this time depends upon the amount of text in the image. A sample image is shown in Figure 7.3.

FIGURE 7.3: Hardware Results for Text Recognition

7.4 Results Summary

The major results of the project can be summarized in the following points:

- The results show promising accuracies and substantial speeds for all three domains.
- The accuracy achieved for object detection and recognition using MobileNet SSD model on custom dataset comprising of 30 objects with 50 images per category is around 80%.
- The accuracy achieved for facial detection and recognition using a combination of HAAR and HOG algorithms on custom dataset comprising of 10 persons with 50 images per person is around 85%.
- The accuracy achieved for text detection and recognition using a combination of EAST and Tesseract models pre-trained on dataset for English language is around 90%.

These results are illustrated in the form of graph shown in Figure 7.4.

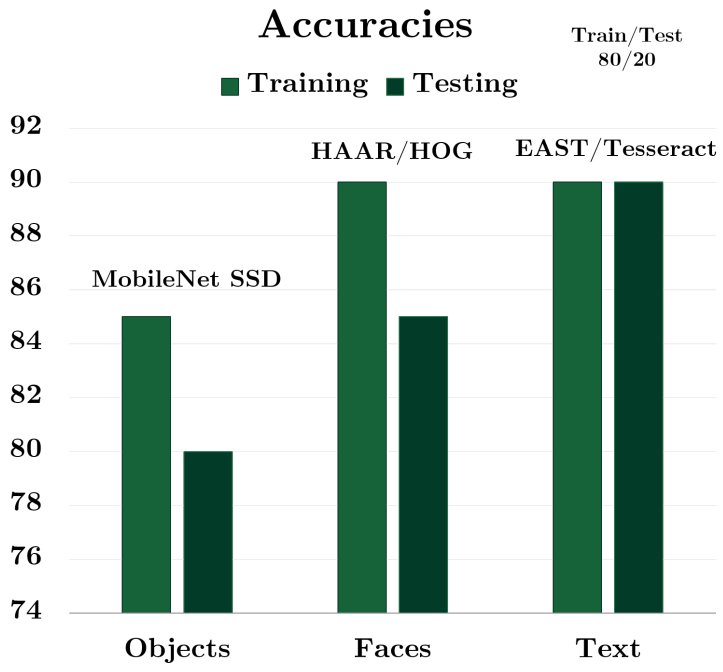

FIGURE 7.4: Graph for Accuracies

Chapter 8

Conclusion and Future Work

8.1 Summary

The purpose of the project is to help the people with visual disabilities to recognize everyday objects, faces and textual information in their line of sight to make them more independent in their daily chores. The accuracy achieved for object recognition is around 80%, for facial recognition is around 85%, and for text recognition is around 90%. The solution is cheap, compact and wearable being fairly quick and accurate. However, the speed and accuracy could be improved even further.

8.2 Applications

The technology can be useful in many other applications aside from helping the blind.

- The technology can be used in self-driving cars to detect other cars on the road, identify pedestrians and read road signs.
- The technology can also be useful in surveillance procedures to monitor some individuals and their activities as well.

8.3 Future Prospects

Following are some of the recommendations for future work:

- The number of objects or known faces can be increased by using more data.
- The ability to recognize languages other than English can be added as well.
- The accuracies can be increased by using a larger dataset comprising of more data.
- The speed can be increased by using a more powerful microcontroller or computer.

Table 8.1: Cost analysis

Items	Cost (PKR)
Raspberry Pi	6000
Powerbank	2000
Webcam	3000
Earphones	400
Glasses	200
Cases/Boxes	1000
Misc	400
Total	13000

8.4 Cost Analysis

A comprehensive final cost analysis of the project and the product is given in Table 8.1.

References

[1] H. Jabnoun, F. Benzarti and H. Amiri, "Object Detection and Identification for Blind People in Video Scene", In: 15th International Conference on Intelligent Systems Design and Applications (ISDA), 2015.

[2] K. OShea and R. Nash, " An Introduction to Convolution Neural Networks", 2015.

[3] K.M.M. Rao, "Overview of Image Processing", In: Readings in Image Processing Fundamentals Of Digital Image Processing, Prentice-Hall, 1989.

[4] E.L. Hal, "Computer Image Processing and Recognition", Academic Press, 1979.

[5] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005.

[6] M. Jamshed and A. Patwary , "Significant HOG-Histogram of Oriented Gradient Feature Selection for Human Detection", In: International Journal of Computer Applications, 2015, 132(17) , 0975-0987.

[7] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", In: IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[8] N. Ifeanyi, O. Ikenna and O. Izunna, "Text To Speech Synthesis (TTS)", In: IJRIT International Journal of Research in Information Technology, 2014, 2(5), 154-163.

[9] Hariprasad, K.C., et al., "Information awareness for the visually-impaired using machine-vision", In: Advances in Natural and Applied Sciences, 2017, 11(6), 220-225.

[10] W. Liu, D. Anguelov, et al., "SSD: Single Shot MultiBox Detector", In: Proceedings of the European Conference on Computer Vision (ECCV), 2016.

[11] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", In: Conference on Computer Vision and Pattern Recognition, 2001.