Palestine Polytechnic University

College of Information Technology and Computer Engineering

Department of Computer Engineering

EmoSense: Emotion Recognition For Blind People

Project Team:

Basel Ebido

Omar Aburish

Samir Abuisneneh

Supervisor:

Dr. Alaa Halawani

This project is submitted to fulfill the requirements for a Bachelor's degree in Computer Systems Engineering

Acknowledgment

In the name of "Allah", the most beneficent and merciful who gave us strength, knowledge and helped us to get through this project. To the people that have inspired and supported us into the people that we are today, our families, friends and our supervisor. We would've never been able to reach this achievement without their support, care, and encouragement. We want to thank them all and we would like to express our gratitude to our graduation project supervisor Dr. Alaa Halawani for his guidance, support, and encouragement throughout the project.

Moreover, we owe an immense debt of gratitude to our families, whose unwavering encouragement and continuous support have been the cornerstone of our journey. Their generosity, both in spirit and action, has shaped the very fabric of who we are. Mom, Dad, and all our family members, your belief in us has been a guiding light, and for that, we are profoundly thankful.

At last, we acknowledge the collective effort that has propelled us forward. Each person who touched our lives, leaving an imprint of care and encouragement, has played a vital role in our story. As we celebrate this milestone, we do so with gratitude for the shared moments and the countless individuals who have left an indelible mark on our hearts.

Abstract

EmoSense is a technology-driven solution designed to help visually impaired individuals better understand the emotions of those around them. Social interactions can be challenging for individuals with visual impairments, and emotions play a crucial role in how people communicate and interact with one another. Understanding the emotional states of others can be especially difficult for visually impaired individuals. Therefore, EmoSense aims to leverage the power of artificial intelligence to provide either haptic or audio feedback of the emotions of person they are interacting with.

The EmoSense system consists of a motor-driven camera and an AI system that detects human speech then tracks, captures and analyzes the facial expressions of individuals in the wearer's surroundings. The system then categorizes the emotions of the facial expressions into one of five categories: happiness, sadness, anger, surprise, and neutral. The system provides this information as haptic vibration or audio that the wearer can use to navigate social interactions more effectively.

EmoSense represents a significant step forward in improving the quality of life for visually impaired individuals by providing them with a better understanding of the emotional states of those around them, enabling them to participate more fully in social events and interactions. The EmoSense project aims to help visually impaired individuals better understand the emotions of those around them. It utilizes a head-mounted device equipped with a camera and an AI system to capture and analyze the facial expressions of individuals in the wearers surroundings. The system categorizes emotions into happiness, sadness, anger, surprise, and neutral, and provides this information as haptic vibration or audio feedback to the wearer.

Keywords: Real-time emotion recognition meeting experience for the visually impaired using Raspberry Pi, webcam, vibration motor, servo motor, 2-Axis Pan and Tilt Mount Kit, microphone, Bluetooth earphones, OpenCV, machine learning, CNN, facial detection, and haptic/audio feedback.

Contents

1 Chapter 1: Introduction 9

1.1 Overview 9

1.2 Problem Statement 9

1.3 Aims 9

1.4 System Description 10

1.5 System Requirements 10

1.5.1 Functional Requirements 10

1.6 Objectives 11

1.7 Expected Results 11

1.8 Constraints 12

1.9 Outline 12

2 Chapter 2: Background 13

2.1 Overview 13

2.2 Theoretical background 13

2.2.1 Machine Learning 13

2.2.1.1 Convolutional Neural Network (CNN) 14

2.2.2 Facial Detection 15

2.2.2.1 Identifying Facial Detection Algorithms 15

2.2.2.2 Best Algorithm for this project 17

2.2.3 Emotion Detection 17

2.2.4 Speech Detection 18

2.3 Literature Review 19

2.3.1 An Ear Wearable Device System for Facial Emotion Recognition Disorders 19

2.3.2 Emotion Recognition - A Tool to Improve Meeting Experience for Visually Impaired 20

2.3.3 Real-time Stage Tracking Camera using Raspberry Pi 21

2.4 Hardware Components 22

2.4.1 Raspberry Pi 3 22

2.4.1.1 Hardware Comparison between Raspberry Pi 3 and Nvidia Jetson Nano 23
2.4.2 Webcam 24
2.4.3 Vibration Motor 24
2.4.4 Servo Motors 25
2.4.5 Two Axis Servo Pan Tilt 25
2.4.6 Microphone 26
2.4.7 Transistor BC547 26
2.4.8 Bluetooth earphones 27
2.5 Software Components 29
2.5.1 Programming languages 29
2.5.2 Libraries 29

3 Chapter 3: Design 31
3.1 Overview 31
3.2 Detailed Design and Hardware Setup 31
3.3 Block Diagram 31
3.4 Schematic Diagram 32
3.5 Software Design 34

4 Chapter 4: Implementation 37
4.1 Overview 37
4.2 Hardware Implementation 37
4.3 Software Implementation 37
4.3.1 pigpiod 37
4.3.2 Tensorflow Lite 37
4.3.3 Yet Another Mobile Network (YAMnet) 38
4.3.4 Async IO for multi threading 38
4.3.5 Coding 38
4.3.5.1 Setup 38
4.3.5.2 Loop 38

5 Chapter 5: Results and Validation 42
5.1 Overview 42

5.2 Hardware Testing 42
5.2.1 Testing Raspberry Pi 42
5.2.2 Testing Servo Motor 42
5.2.3 Testing Webcam 42
5.2.4 Testing Vibration Motor 43
5.3 Software Testing 43
5.3.1 Raspberry Pi OS installation 43
5.3.2 Testing face detection 43
5.3.3 Tensorflow Lite 43
5.3.4 Testing pigpio 43
5.4 Testing Emotion Detection Model 44
5.4.1 Emotion Detection Model Results 44
5.5 Biggest Face Priority Testing 47
5.6 Testing Vibration Patterns 47
5.7 Test Scenarios 48

6 Chapter 6: Conclusion and Future Work 50

6.1 Conclusion 50
6.2 Future Work 51

List of Figures

1.1 Simple Block Diagram 10
2.1 Convolutional Neural Network Architecture. 15
2.2 Raspberry Pi 3. 23
2.3 Webcam. 24
2.4 Coin Vibration Motor 24
2.5 Servo Motor 25
2.6 Pan Tilt. 26
2.7 Microphone. 26
2.8 Transistor 27
2.9 Transistor 27
2.10 Bluetooth earphones 28
3.1 Block Diagram 32
3.2 Schematic 33
3.3 State Diagram 36
5.1 Model Accuracy chart. 44
5.2 Model Loss chart. 44
5.3 Neutral Emotion. 45
5.4 Happy Emotion. 45
5.5 Sad Emotion. 46
5.6 Angry Emotion. 46
5.7 Surprise Emotion. 46
5.8 Bigger Face Priority. 47

List of Tables

5.1 Test Scenarios 48

Acronyms

ML Machine Learning

CNN Convolutional Neural Network

HOG Histogram of Oriented Gradients

RNN Recurrent Neural Network

GPU Graphics Processing Unit

YAMnet Yet Another Mobile Network

1 Chapter 1: Introduction

1.1 Overview

Blindness is a visual impairment that affects a person's ability to perceive the world through sight. In 2015, the global estimate for the number of people with visual impairment was approximately 253 million. Out of this total, around 36 million individuals were classified as blind, while an additional 217 million people experienced moderate to severe visual impairment (MSVI) [1]. Blind people possess unique strengths, talents, and abilities that enable them to lead fulfilling and independent lives. Blind individuals face unique challenges in perceiving and understanding the emotions of those around them, as they heavily rely on non-visual cues such as tone of voice, body language, and context. EmoSense is a technology-driven initiative that aims to assist visually impaired individuals to better understand the emotions of those around them.

1.2 Problem Statement

Visually impaired individuals face challenges in understanding the emotions of those around them, which can lead to difficulties in social interactions and contribute to social isolation. The EmoSense project aims to address this issue by using technology, specifically artificial intelligence and haptic / audio feedback, to provide users with information about the emotional states of those around them.

1.3 Aims

- To assist visually impaired individuals in understanding the emotions of those around them.
- To leverage artificial intelligence to provide haptic or audio feedback of the emotions of people around the wearer.
- To improve the quality of life for visually impaired individuals.
- To reduce social isolation for visually impaired individuals.
- To make the visually impaired individuals feel more included.

1.4 System Description

EmoSense is a system designed to help visually impaired individuals better understand the emotions of people around them. It uses a head mounted device equipped with a camera to capture video of the wearer's surroundings. The video is then sent to an AI-powered emotion recognition system that analyzes the images and uses machine learning algorithms to detect the emotions of people in the images. It tracks the face of the opposing person using motors to rotate the camera. Figure 1.1 shows the components of the system and how they act with each other.

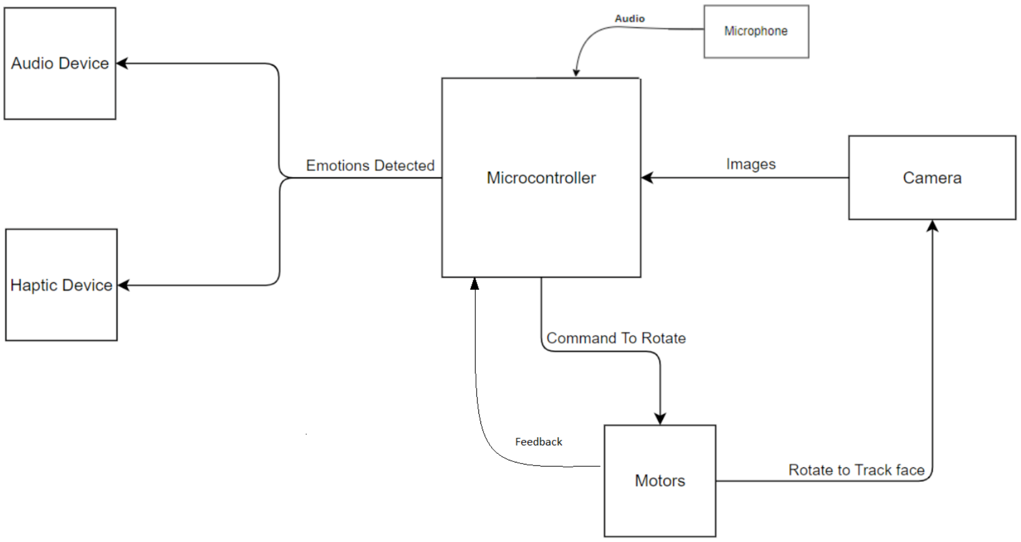

Figure 1.1: Simple Block Diagram

1.5 System Requirements

1.5.1 Functional Requirements

- The EmoSense system shall capture and analyze facial expressions of people around the wearer using a camera.
- The EmoSense system shall recognise human speech to start looking for a face to analyze.
- The EmoSense system shall translate the facial expression analysis into haptic or audio feedback of the emotions of people around the wearer.

- The EmoSense system shall provide feedback on five emotions: happiness, sadness, anger, surprise, and neutral.
- The EmoSense system shall be compatible with different facial structures and skin tones.
- The EmoSense system shall be able to track the movement of the nearest face to the wearer.

1.6 Objectives

- To develop a head mounted device equipped with a camera and an AI system to capture and analyze the facial expressions of individuals in the wearer's surroundings.
- To develop a motor-driven camera in this device that tracks the nearest face around the wearer.
- To translate the information into an haptic vibration or audio that the wearer can use to navigate social interactions more effectively.
- To provide greater participation in social events and an overall improved quality of life for visually impaired individuals.

1.7 Expected Results

We expect this project to be able to do the following:

- Camera tracks the opposing face of the wearer.
- The system accurately classifies the detected emotions into predefined categories such as happiness, sadness, anger, neutral and surprise.
- The system accurately recognises human speech to start the recognition part.
- Output the result as a vibration pattern or audio to the blind person.

1.8 Constraints

These are the conditions that the system to ideally work under. If not satisfied, it will not have the best results.

- Lighting Conditions: Lighting conditions within a closed room can vary, affecting the quality and visibility of facial features.
- Distance between the camera and the opposing face: The opposing face cannot be too far nor too near the camera.
- Emotion detection for one person at a time.

1.9 Outline

This is the outline for this document:

Chapter 1 provides an introduction to the EmoSense project, outlining its objectives and system requirements. Chapter 2 presents a comprehensive literature review, including existing emotion detection projects and their outcomes. It also delves into the theoretical underpinnings of the project, covering hardware and software components. Chapter 3 details the conceptual design of the EmoSense system, encompassing block diagrams, pseudo-code, and in-depth hardware connections. Chapter 4 discusses the implementation of the project on the hardware and software sides, while delving into the various software algorithms used in Emosense. Chapter 5 delves and analyses the results of testing both the hardware and software components. Chapter 6 presents the conclusion, summarizing the key findings, implications, and potential impact of the EmoSense project on the lives of visually impaired individuals alongside any future improvements that can be implemented in Emosense.

2 Chapter 2: Background

2.1 Overview

This chapter mainly provides a quick background about the important components in our project. First, we will talk about the theoretical background, we will explain general terms like machine learning, deep learning, and computer vision. Secondly, we will present a literature review to show how previous studies achieved such objectives. Thirdly, we will look over the main hardware devices we need and why we need them. Finally, we will talk about the software background, generally about the programming languages and the algorithms we will use.

2.2 Theoretical background

The first step the system shall do is to detect the faces that are seen by the camera. Then, it shall detect the emotion for that face. If there are multiple faces it will take the emotion of the largest one it sees. Finally, it will give the user the output using haptic or audio feedback.

In this section we'll discuss the algorithms needed for each part and how they work.

2.2.1 Machine Learning

Machine Learning (ML) is a branch of artificial intelligence (AI) that focuses on the development of algorithms and models that enable computers and systems to learn from data and make predictions or decisions without explicit programming. ML algorithms learn patterns and relationships in data by analyzing and processing large datasets, allowing them to automatically improve their performance over time. [2]

The core idea behind ML is to enable machines to learn from experience and data, rather than being explicitly programmed for specific tasks. ML algorithms can recognize complex patterns, make predictions, classify data, and discover insights from large and complex datasets.

Supervised machine learning (ML) is a subfield of ML that focuses on developing algorithms and models capable of learning patterns and making predictions based on labeled training data. The key theory underlying supervised ML is the concept of learning from input-output pairs provided in the training dataset.

In supervised ML, the goal is to train a model to map input features to corresponding output labels or target values. The training data consists of examples with known inputs and their corresponding correct outputs. The model learns from these examples to generalize and make predictions on unseen or future data [3].

2.2.1.1 Convolutional Neural Network (CNN)

A CNN, or Convolutional Neural Network, is a type of deep neural network commonly used in computer vision applications such as image and video recognition [4].

CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers extract features from the input images by applying a set of learnable filters. Pooling layers downsample the feature maps, reducing spatial dimensions and extracting dominant features. Fully connected layers combine the learned features and make final predictions.

In a CNN, the input data (such as an image) is fed through a series of convolutional layers as shown in figure 2.1 [5], which extract features from the input data by applying convolutional filters. These filters detect edges, shapes, and patterns in the image, and the resulting feature maps are then passed through pooling layers to reduce the spatial size of the output.

After several convolutional and pooling layers, the output is flattened and fed through one or more fully connected layers, which use the extracted features to classify the input image into one or more categories. Figure 2.1 shows how the architecture of the CNN model and how it operates

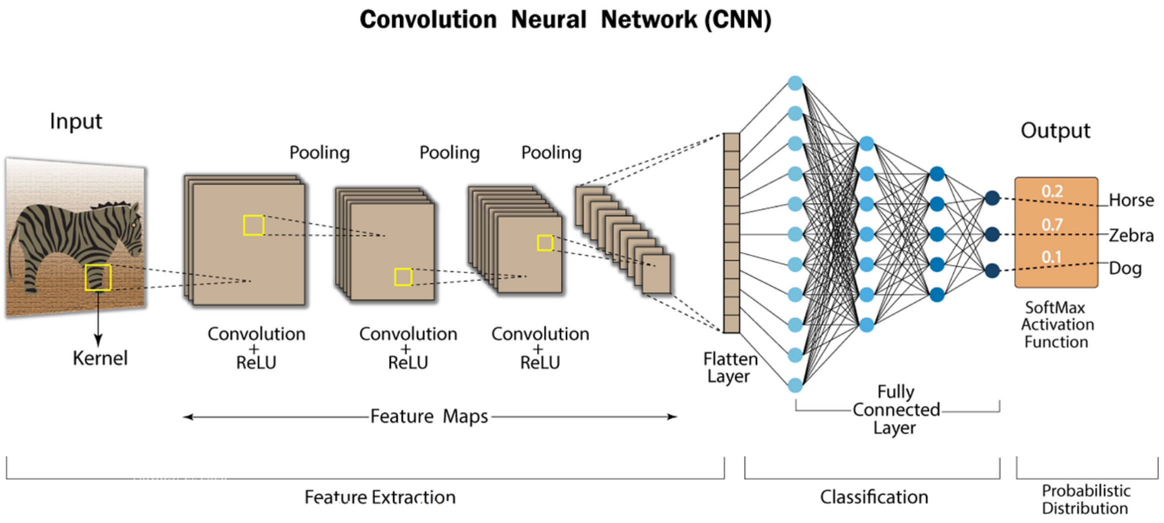

Figure 2.1: Convolutional Neural Network Architecture.

2.2.2 Facial Detection

This will be the first step in our workflow. We need an algorithm that detects the faces with high response time.

2.2.2.1 Identifying Facial Detection Algorithms

Facial detection is the process of identifying faces in a particular image or video. It is a very important process since it is used in many fields. Due to that, there are many algorithms that do this. This will be the first step in our system. The most important algorithms for this task are:

1. Viola-Jones Algorithm: This is one of the most widely used algorithms in this field. It uses Haar-like features to identify facial features such as eyes, nose, and mouth. The algorithm is known for its efficiency and speed, making it suitable for real-time applications, as it uses a simple cascade classifier with Haar-like features.

   Haar-like features are simple rectangular filters that act as local image intensity patterns. These features encode information about changes in contrast, which are useful for detecting facial features.

   To efficiently compute Haar-like features, the algorithm utilizes an intermediate representation called the integral image. The integral image allows for rapid calculation of rectangular sum operations over an image, which is crucial for evaluating Haar-like features efficiently.

The Viola-Jones algorithm uses a cascaded classifier composed of multiple stages. Each stage consists of a set of weak classifiers, which are simple binary classifiers trained on Haar-like features. The cascade structure enables efficient filtering and quickly rejects regions of the image that are unlikely to contain faces, reducing the computational load.

The weak classifiers within each stage are trained using the AdaBoost algorithm. AdaBoost iteratively selects the most informative features and assigns weights to the weak classifiers to create a strong classifier that combines their decisions. This boosting process helps improve the overall accuracy of the algorithm.

The original implementation reported a processing speed of up to 15 frames per second (fps) on a 700 MHz Pentium III processor [6].

2. Convolutional Neural Networks (CNNs): CNNs are a deep learning approach that have become very popular for facial detection. As we have mentioned in section 2.2.1.1 that this algorithm is trained using large datasets of labeled images. We use it to learn to detect faces with high accuracy.

   The processing speed of CNNs varies widely depending on the network architecture, input image size, and hardware used. For example, a popular CNN-based approach for facial detection, the Single Shot Detector (SSD) network, can achieve real-time processing speeds of up to 30 fps on a high-end GPU [7].

3. Histogram of Oriented Gradients (HOG): HOG is another popular feature-based approach for facial detection.The HOG algorithm works by capturing local gradients or changes in intensity and their orientations within an image. It represents the distribution of gradient orientations using a histogram-based descriptor.

   The algorithm begins by computing the gradients of image intensity in both the horizontal and vertical directions. This can be done using techniques such as the Sobel operator.

   The image is divided into small cells, and for each cell, the gradient orientations are quantized into predefined bins. This quantization encodes the dominant orientations within each cell.

Within each cell, a histogram is constructed, where each bin represents the count or magnitude of gradient orientations falling within that bin. The histogram summarizes the distribution of gradients within the cell.

To account for variations in lighting conditions and contrast, neighboring cells are grouped together into blocks. Block normalization is applied to each block, which normalizes the histograms within the block and enhances the algorithm's robustness.

The resulting histograms from all cells are concatenated into a single feature vector, which represents the image's HOG descriptor. This feature vector captures the spatial distribution of gradient orientations in the image.

The HOG algorithm is also known for its fast detection speed. Dalal and Triggs (2005) report that their implementation of HOG achieves a detection rate of 4 frames per second on a 3 GHz Pentium 4 processor [8].

2.2.2.2 Best Algorithm for this project

With these results, it seems that Viola Jones will be the most appropriate algorithm since it will be run on a low power device.

2.2.3 Emotion Detection

For emotion detection, we will need to use a neural network for that. There are many different architectures like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and other architectures. We will use CNN in this project.

Convolutional Neural Networks (CNNs) mentioned in section 2.2.1.1 can be better suited for emotion detection than other machine learning algorithms for several reasons:

- CNNs can learn hierarchical representations of the input data: CNNs can learn to identify patterns and features at different levels of abstraction by applying a series of convolutions and pooling operations. This allows them to capture complex relationships between the input data and the emotional states [9].

- CNNs are robust to variations in the input data: CNNs can handle variations in the input data, such as changes in lighting, pose, or facial expression, which can be important in emotion detection tasks where the emotional states are often expressed in subtle ways [10].

- CNNs can learn from large amounts of data: CNNs can be trained on large datasets of facial images or videos, which can improve their ability to generalize to new data and improve the accuracy of their predictions.

- CNNs can incorporate spatial and temporal information: CNNs can be used to process both spatial and temporal information in the input data, which can be important for emotion detection tasks that involve facial expressions in videos.

2.2.4 Speech Detection

We propose using Recurrent Neural Networks (RNNs) for speech detection, as they are highly effective in processing sequential data like speech or text. RNNs are commonly utilized for temporal or ordinal problems such as language translation, natural language processing (NLP), speech recognition, and image captioning. They have achieved state-of-the-art performance in various speech and language processing tasks and have been widely adopted in real-world applications like Siri, voice search, and Google Translate.

RNNs consist of recurrent layers that extract features by leveraging recurrent connections, allowing them to capture information from previous time steps in the sequence. These extracted features are then classified into one or more categories using fully connected layers. One key advantage of RNNs for speech detection is their ability to learn hierarchical representations, enabling the identification of patterns and features at different levels of abstraction. This capability allows RNNs to capture complex relationships between input data and speech patterns.

Another advantage of RNNs is their robustness to variations in the input data, such as changes in pitch or speaking rate, which is particularly important for detecting subtle speech patterns. RNNs can effectively handle these variations, enhancing the accuracy of speech detection tasks. Additionally, RNNs excel in learning from large datasets, enabling them to generalize well to new data and improve prediction accuracy.

Lastly, RNNs are capable of incorporating temporal information, making them suitable for speech detection tasks that involve recognizing speech patterns over time. This temporal awareness enables RNNs to process and interpret the sequential nature of speech data accurately.

In conclusion, RNNs are well-suited for speech detection due to their ability to learn hierarchical representations, handle variations in the input data, learn from large datasets, and effectively incorporate temporal information. With their impressive performance in speech recognition and language processing, RNNs offer promising capabilities for accurate and robust speech detection applications. [11].

2.3 Literature Review

Studies related to emotion detection for the visually impaired have proposed various approaches, including wearable devices, smartphone-based systems, and multimodal approaches.

2.3.1 An Ear Wearable Device System for Facial Emotion Recognition Disorders

Lian et al. [12]. introduce a system that utilizes ear wearable devices for facial emotion recognition. With an overall accuracy rate of 88.7%, this system integrates sophisticated sensors into the ear wearables to capture physiological signals and facial expressions indicative of different emotional states. By monitoring parameters like heart rate, skin conductance, and temperature, valuable insights into individuals' emotional states are obtained.

The study employs a Convolutional Neural Network (CNN) algorithm to achieve high accuracy in emotion recognition. By training on a diverse dataset that pairs facial expressions with physiological signals, the algorithm learns to recognize complex visual cues associated with various emotions. Leveraging a large and diverse training dataset allows the CNN algorithm to effectively learn complex relationships and generalize its understanding of emotions.

During the testing phase, the system accurately classifies emotions by leveraging the knowledge acquired during training. The CNN algorithm successfully identifies and labels emotions based on the captured physiological signals and facial expressions.

Ear wearable devices offer advantages such as unobtrusiveness and convenience, allowing for continuous and non-invasive monitoring in various real-world settings. The integration of sensors into these wearables enables the capture of precise physiological signals and facial expressions, even in dynamic environments.

The combination of the CNN algorithm and physiological signal analysis forms a robust framework for real-time emotion recognition. This technology has potential applications in diagnosing and monitoring individuals with neurological disorders that affect facial emotion recognition in clinical settings. It also holds promise for advancing research on emotional processing and its relationship with physiological responses.

This project is suitable for our purposes as it shares similarities with our own project in terms of detecting emotions using facial features.

2.3.2 Emotion Recognition - A Tool to Improve Meeting Experience for Visually Impaired

Lutfallah et al. [13] addressed the challenge faced by blind and visually impaired individuals in perceiving facial expressions during meetings, which are crucial for non-verbal communication. They proposed an emotion recognition system that aims to enhance the meeting experience for visually impaired people by providing real-time detection of emotions relevant to lively discussions: ”agree,” ”neutral,” and ”disagree.”

To achieve this, the researchers developed a system based on neural networks that can extract emotional states from facial expressions captured in videos. They emphasized the importance of considering the temporal information in the video frames to improve emotion recognition accuracy. The system consists of a combination of a convolutional neural network (CNN), a recurrent neural network (RNN), and a multi-layered perceptron (MLP).

The researchers discussed the challenges of emotion recognition, including variations in emotional expressivity across cultures and the ability of individuals to control their facial expressions. They highlighted existing datasets such as the Acted Facial Expressions In The Wild (AFEW) and CK+ datasets ((Cohn-Kanade Plus) is a widely used facial expression database that consists of images and corresponding emotion labels), which were used for training and fine-tuning their model.

To adapt the existing emotion categories in these datasets to the three emotion classes (neutral, positive, and negative) relevant to meetings, the researchers clustered and summarized the original emotion categories. They used pre-training with the AFEW dataset, followed by fine-tuning with the CK+ dataset, which better represented the meeting context.

The proposed system achieved an accuracy of 88.8% on the CK+ dataset, with high accuracy for predicting positive and neutral emotions. However, the classification of negative emotions proved to be more challenging, with a lower accuracy and a tendency to misclassify them as neutral.

The researchers discussed the potential application of their tool in providing real-time feedback on facial expressions during net-based meetings. They presented a graphical interface that displays emotion scores in the form of a histogram and a smiley face, which can be translated into a three-stage signal using a Braille display for visually impaired individuals.

Our project aligns closely with this research project in the emotion detection using facial features, and we can greatly benefit from leveraging the knowledge gained from it.

2.3.3 Real-time Stage Tracking Camera using Raspberry Pi

Suhair Shareef et al. [14] proposed a project that aims to develop a cost-effective and fast model for detecting and tracking objects using computer vision techniques. The authors of the project propose a solution that utilizes a Raspberry Pi 4, coupled with the Raspberry Pi Camera Module, as the hardware platform for image acquisition and processing. The project focuses on real-time object detection and tracking, with the goal of keeping the detected object centered within the camera's field of view.

To achieve this objective, the project employs various algorithms and techniques. Two different detection algorithms are used: the default object detection algorithm, which detects the entire object, and a face detection algorithm, which is activated when the object gets closer to the camera. These algorithms leverage the power of OpenCV, a popular computer vision library, and are implemented in C++ to ensure efficient processing on the Raspberry Pi.

The system utilizes a pan and tilt system that allows the camera to follow the detected object by adjusting its orientation in both the horizontal (pan) and vertical (tilt) directions. This enables the camera to track the object's movement and keep it centered in the image frame. The authors have also considered the constraints of cost, power consumption, robustness, processing time, and ease of deployment in designing the system.

By combining the capabilities of object detection, face detection, and the pan and tilt system, the project aims to achieve reliable and real-time object tracking. The low-power and low-cost embedded vision platform, represented by the Raspberry Pi, offers an affordable and accessible solution for implementing such tracking systems. The authors expect the system to successfully detect and track moving objects, providing a cost-effective alternative to traditional camera setups for various computer vision applications.

In our project, we will employ a face tracking system that bears resemblance to the one proposed by them.

In conclusion, emotion detection technology has the potential to enhance the quality of life for the visually impaired. The studies reviewed in this literature review have proposed various approaches to emotion detection. Further research is needed to improve the accuracy and usability of these systems and to ensure they meet the needs and preferences of visually impaired users.

2.4 Hardware Components

This part describes the components and devices used in the system.

2.4.1 Raspberry Pi 3

Raspberry Pi 3 is a low cost, small weight and credit-card sized computer. It can be used as the main processing unit for the EmoSense project. It can handle the camera input and run the AI algorithms for facial detection and emotion recognition. The Raspberry Pi's GPIO (General Purpose Input/Output) pins can be utilized to connect and control the haptic feedback device. Its compact size and low power consumption make it suitable for a portable and wearable device like EmoSense, figure 2.2 shows the raspberry pi 3 [15].

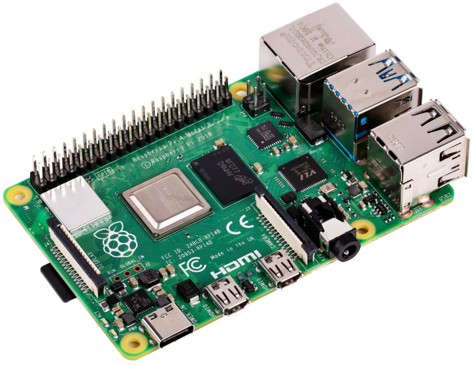

Figure 2.2: Raspberry Pi 3.

2.4.1.1 Hardware Comparison between Raspberry Pi 3 and Nvidia Jetson Nano

While the Jetson Nano is specifically designed for AI and deep learning tasks. As it features an NVIDIA GPU with CUDA cores, which allows for accelerated computing and efficient neural network inference. We decided to use a Raspberry Pi for the EmoSense project for these reasons:

- Affordability: The Raspberry Pi 3 is generally more affordable compared to the Jetson Nano. While the Raspberry pi 3 costs around 35$, the Jetson Nano retails for 149$.

- Availability and community support: The Raspberry Pi has a massive community of enthusiasts and developers worldwide, which translates into a vast amount of resources, tutorials, and community support. Unlike the Jetson Nano which has a much smaller community with less online resources available.

- Connectivity: Raspberry Pi boards have support for Wi-Fi and Bluetooth, making it easier to connect and communicate with other devices. This will prove useful in the EmoSense system for connecting the Bluetooth earphones for the audio feedback option. Unlike Jetson Nano which does not support Bluetooth or Wi-Fi natively.

- Space: For the EmoSense project the space the micro controller can occupy is limited due to its portable nature. The Raspberry Pi 3 takes up considerably less space than the Jetson Nano due to the it having a large heatsink for cooling

2.4.2 Webcam

A webcam will be used for capturing footage and sending it to the Raspberry Pi for analysis. We chose to use a webcam due to its compact and affordable nature. Additionally, webcams typically support resolutions such as 1080p30fps (frames per second) or 720p60fps for video, which are acceptable for our project requirements, figure 2.3 show the webcam [16].

Figure 2.3: Webcam.

2.4.3 Vibration Motor

A vibration motor is a small electric motor with an eccentric weight attached to its output shaft, which creates centrifugal force to produce vibration or oscillation. They are used in mobile phones, game controllers, and other devices for haptic feedback. It will be used in our project to provide haptic feedback to the user regarding the emotional state of the person they are interacting with. For our project a Coin Vibration Motor will the best fit due to its compact size since it will be used in a limited space, figure 2.4 shows the vibration motor [17].

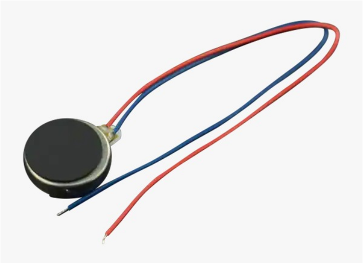

Figure 2.4: Coin Vibration Motor.

2.4.4 Servo Motors

Servo Motor is an actuator that allows an accurate degree of control in position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

In this project we will need two servo motors to rotate the camera horizontally and vertically, figure 2.5 shows the servo motor [18].

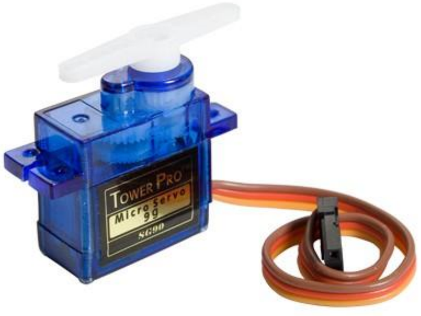

Figure 2.5: Servo Motor.

2.4.5 Two Axis Servo Pan Tilt

Two Axis Pan Tilt Brackets for the camera which is based on 2 axis pan and tilt mechanism for mounted camera. Panning, rolling and tilting are achieved by controlling Servo motors using PWM pulses, figure 2.6 shows the pantilt [19].

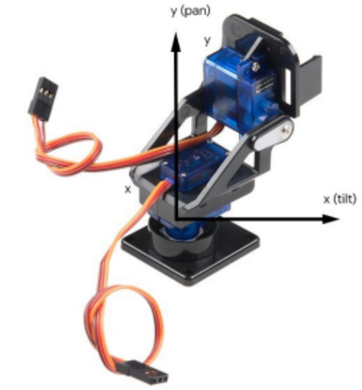

Figure 2.6: Pan Tilt.

2.4.6 Microphone

The microphone will detect if there is a human speaking around the wearer. If there is, the camera will rotate to find their and detect their emotions, figure 2.7 shows the microphone [20].

Figure 2.7: Microphone.

2.4.7 Transistor BC547

BC547 is used To control the current amplifier inside the vibration motor, figure 2.8 shows the transistor [21].

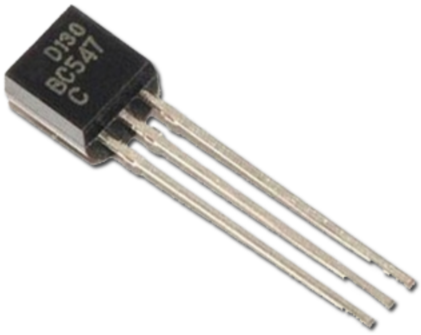

Figure 2.8: Transistor.

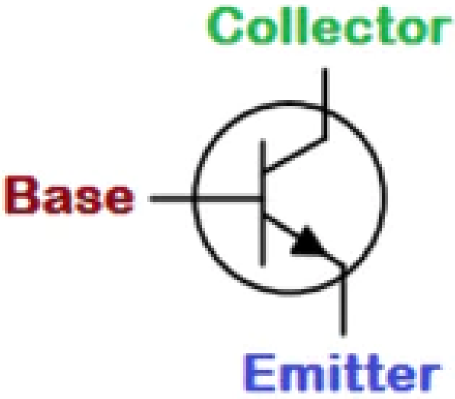

Figure 2.9: Transistor Schematic.

2.4.8 Bluetooth earphones

Bluetooth earphones are wireless headphones that connect to a device through Bluetooth technology. They come in various styles and feature a built-in battery that can be recharged with a USB cable. They provide a convenient and wireless way to listen to audio and communicate with others on the go. They will be used to provide audio feedback to the user, letting them choose between haptic or audio. The user is free to choose his/her preferred brand of earphones and connect them to the device, figure 2.9 shows the bluetooth earphones [22].

Figure 2.10: Bluetooth earphones.

2.5 Software Components

2.5.1 Programming languages

Primarily, Python is used as our programming language since it deals with the complex algorithms in machine learning efficiently and it is also widely used by the machine learning community.

2.5.2 Libraries

- **OpenCV**.

  OpenCV, short for open-source Computer Vision, is a cross-platform open-source programming library that includes a numerous number of Computer Vision algorithms and functions [23].

- **Keras**

  Keras is a high-level neural network library that is built on top of TensorFlow, CNTK, or Theano. It provides a simple and easy-to-use interface for building and training neural networks. Keras allows users to quickly prototype and iterate on neural network models, making it a popular choice for both researchers and industry practitioners [24].

- **Pigpio**

  Pigpio is a Python library for controlling the GPIO (General Purpose Input/Output) pins on a Raspberry Pi. It facilitates communication with the pigpio daemon, a background process that manages the low-level GPIO operations. This library allows users to interface with various sensors, actuators, and other electronic components connected to the Raspberry Pi's GPIO pins [25].

- **TensorFlow Lite**

  TensorFlow Lite is a lightweight solution developed by Google that allows developers to deploy machine learning models on mobile and embedded devices with constrained resources. It is a part of the broader TensorFlow ecosystem, designed specifically for mobile and edge computing scenarios. TensorFlow Lite enables efficient execution of machine learning models on devices such as smartphones, IoT devices, microcontrollers, and other embedded systems. [26].

3 Chapter 3: Design

3.1 Overview

This chapter discusses the overall design of the system and how its components are integrated, showing the block diagram, schematic and state diagram for the system. In addition, we will discuss the basic algorithms we will use.

3.2 Detailed Design and Hardware Setup

This is a general overview of how the main components are put and how they work with each other:

- The system will be attached on a helmet (the raspberry pi, the motors and the camera).
- The motors and the camera will be set up in tilt so that the motors rotate the camera 180 degrees in the X-axis and 45 degrees in the Z-axis.
- The wearer will wear the vibration motor on their arm.

3.3 Block Diagram

The block diagram in Figure 3.1 represents what the main components of the system.

- Once the system detects human voice it will start the searching mode
- If a person is seen, the motors will start tracking them. The system starts to detect their emotions and sends it to the user through the vibration motors.
- If a person is not seen, the system will return to the idle mode.

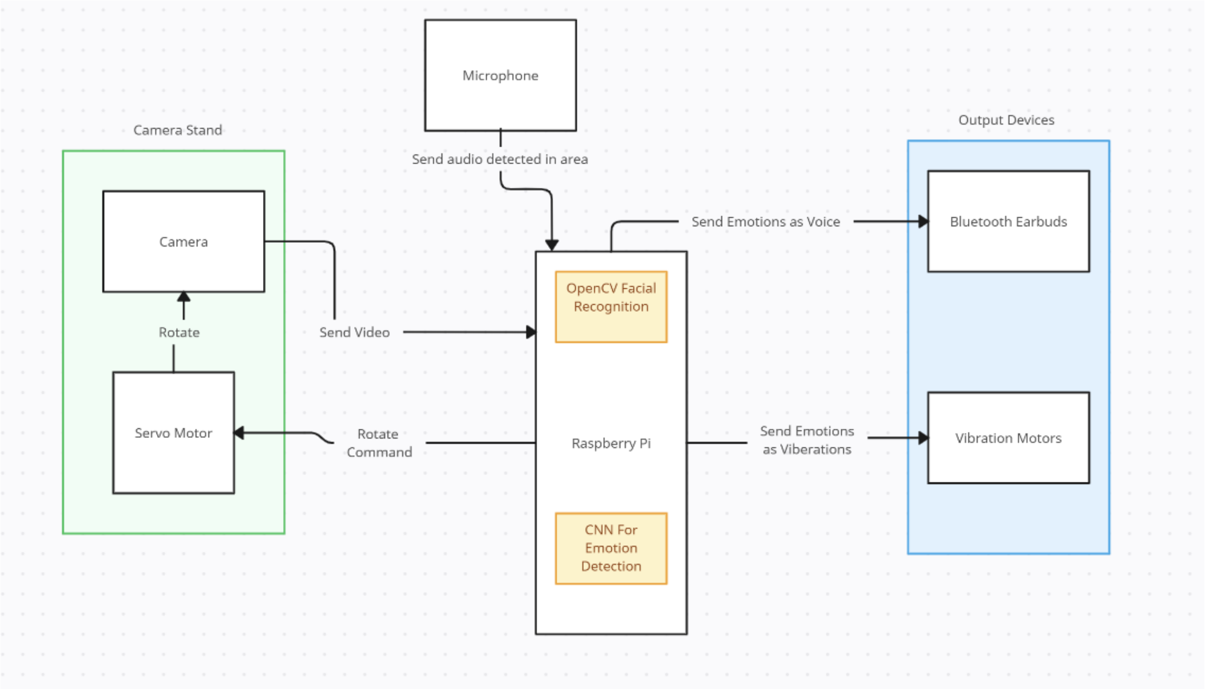

Figure 3.1: Block Diagram

3.4 Schematic Diagram

Figure 3.2 shows the schematic diagram of the components of the system and how they are connected. The main component in it is the Raspberry PI 3 which controls all the other components. The camera will be connected through the USB ports in the Raspberry PI. Then there is the two servo motors and the vibration motor, these will be connected through the GPIO pins which are the General Purpose Input/Output for the Raspberry.

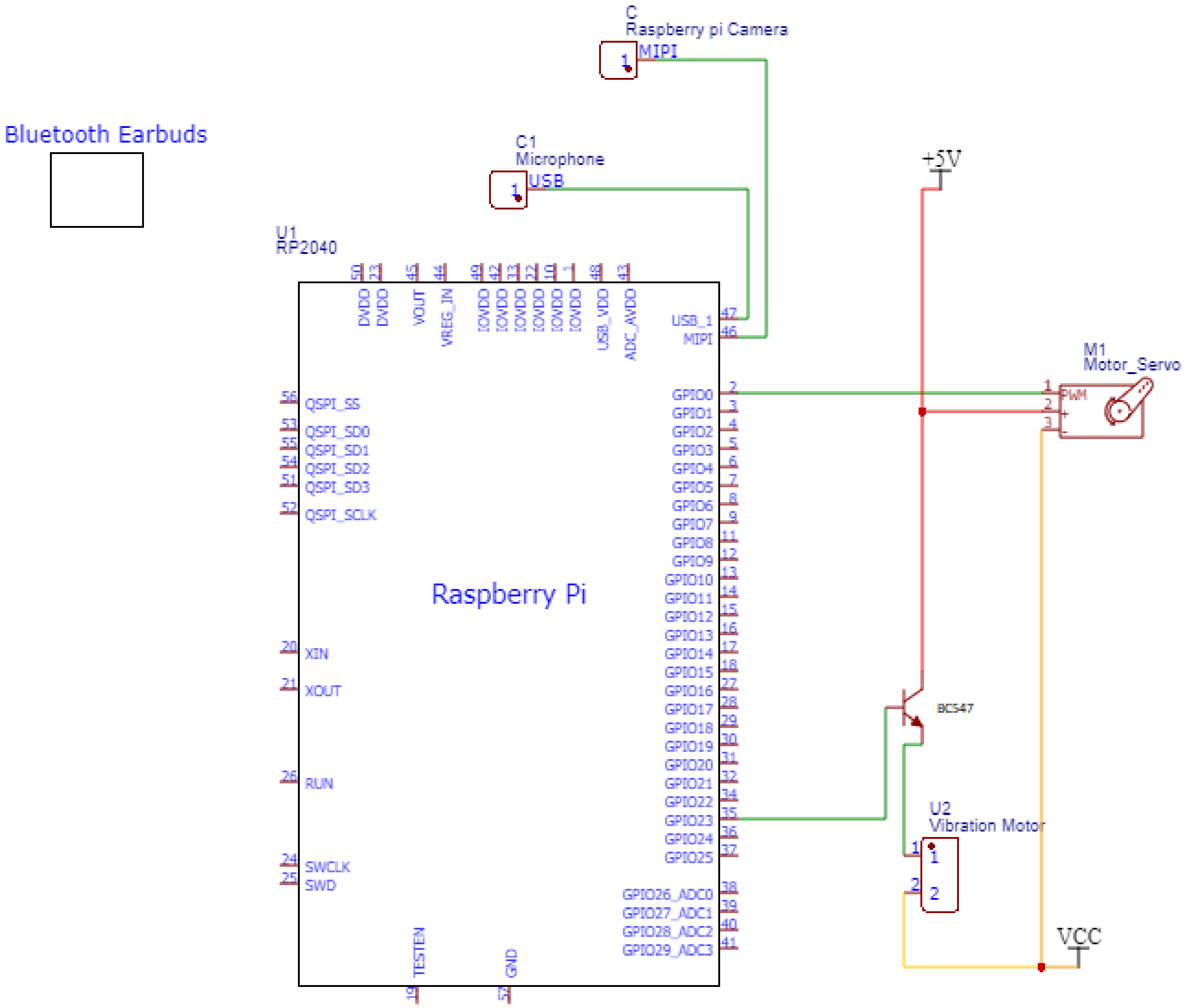

Figure 3.2: Schematic

3.5 Software Design

The algorithm shown below explains how the system will be initialized and triggered

There will be three main modes (states) for the system. The first state is the idle mode, which is the default state, and motors will be idle, this is the state where the system is checking for human speech, once it detects it it goes into the second state. The second state is the searching mode where the model will actively look for a face to analyze. The third state is the Tracking and Emotion Detection Mode. In this mode, the camera will track the face in front of it, and the system will detect the emotions of that face. After that, it will send the emotions to the user as haptic feedback.

We chose to employ face detection followed by CNN in two separate steps, rather than relying solely on CNN for both face detection and emotion recognition. This decision was made to optimize performance, as using CNN for every frame, even those without faces, would significantly impact computational resources. Compared to Viola-Jones, CNNs are more computationally intensive, hence utilizing them solely for face detection across all frames could lead to decreased efficiency.

Figure 3.3 shows how transitions between the modes. Algorithms 1 and 3 show what happens in each state.

Algorithm 1 Idle Mode

```
while Idle Mode is on do
    if Voice is detected using RNN or Face is detected using CNN then
        if Audio is detected using RNN then
            Feed the voice into the speech detector.
            if Human Speech is detected then
                Enter Searching Mode.
            end if
        end if
        if Face is detected using CNN then
            Stop Motors.
            Enter Tracking and Emotion Detection.
        end if
    end if
end while
```

Algorithm 2 Searching Mode

```
Rotate the camera to the left to it is maximum
if Face is found then
    Stop Motors.
    Enter Tracking and Emotion Detection Mode.
end if
Do the above steps but rotate the camera to the right
if Face is not found then
    Return to Idle Mode.
end if
```

Algorithm 3 Tracking and Emotion Detection

```
while Tracking and Emotion Detection Mode is on do
    while number of emotions detected $< 5$ do
        Detect Faces using Viola Jones
        Select the largest face detected as the target face
        Capture the emotions of the largest face detected using CNN.
        Store the emotion detected.
        number of emotions detected += 1
    end while
    $CurrentEmotion \leftarrow$ The most frequent Emotion in that period.
    Convert the $CurrentEmotion$ classification into feedback to the user.
    if time passed without a face detected then
        Enter Idle Mode
    end if
end while
```

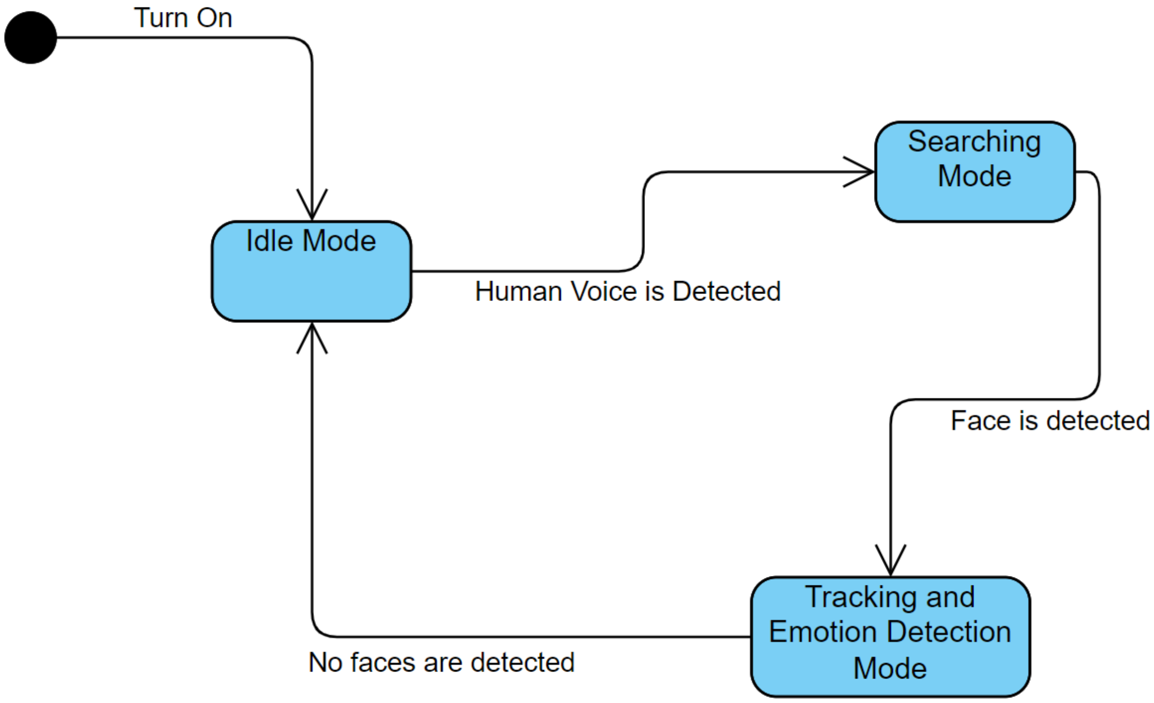

Figure 3.3: State Diagram

4 Chapter 4: Implementation

4.1 Overview

This chapter explains the implementation part for the hardware components and the software algorithms. It dives deep into the different hardware components of the system and its software with all of its modules.

4.2 Hardware Implementation

In the previous chapter we viewed the schematic diagram of the system which shows how the components interconnect with each other. In this section we will present how we assembled these components and arranged them.

1. **Assemble the tilt.** We only need the tilt part since we will only have one degree of freedom.

2. **Install the Raspberry Pi camera on the board and attach it to the base.**

3. **Install the vibration motor.** Raspberry Pi 3B+ does not put enough current to the vibration motor to work, so we need to make the raspberry pi output as a base for a transistor which gives current from a 3.3V source to the motor.

4.3 Software Implementation

In order for the system to start working, we needed several libraries to be installed on the OS of the raspberry pi. This required writing a Raspbian OS image to an SD card.

4.3.1 pigpiod

We noticed that without using this library that motor produced a lot of jitter. This library aims to fix this problem thus ensuring stable motor operation. The pigiod runs as a daemon in the background.

4.3.2 Tensorflow Lite

In order for the CNN model to work, we needed to install tensorflow Lite library. We noticed that we need the Operating System to be 64 bit in order for it to work.

4.3.3 Yet Another Mobile Network (YAMnet)

YAMnet is a lightweight model designed to classify audio to various classifications including human voice. By using it we are able to detect when human speech occurs to start the searching mode. [27]

4.3.4 Async IO for multi threading

With the help of Async IO we were able to run the voice detection and face detection models on two seperate threads. allowing us to always check for human speech when no face is currently in view.

4.3.5 Coding

In this section, we'll describe setup, detection and tracking functions on the raspberry pi.

4.3.5.1 Setup

First, we initialize several parts:

- Face Detection Model (Viola Jones with OpenCV).
- Connect the pigpio daemon.

4.3.5.2 Loop

Voice Detection: Using the model YAMnet the system actively checks for human speech around it, when it detects such it goes into the facial detection part of the system.

Facial Detection:

Using the face detection model ”Viola Jones”, the algorithm detects faces for every captured frame. this allows the model to send the face data to the emotion recognition part of the system and to use facial tracking to keep the face in the center

Facial Tracking:

After getting the face position from the facial detection model we calculate the angle that both the servo motors shall rotate to in the X and Y axis to keep the face in the center.

Emotion Recognition:

After acquiring face data, we provide it to the CNN algorithm that was specially trained to recognize emotions. It outputs the most likely emotion of that face and sends it as audio or vibration pattern to the end user.

The Convolutional Neural Network (CNN) model used for this emotion recognition task is a 2D CNN, which is particularly effective for image processing due to its ability to capture spatial features. The model is composed of several layers, including convolutional layers, pooling layers, and fully connected layers.

The convolutional layers use the Rectified Linear Unit (ReLU) activation function, which introduces non-linearity into the model, enabling it to learn complex patterns. ReLU is often preferred in CNNs due to its computational efficiency and its ability to mitigate the vanishing gradient problem. [28]

After the convolutional and pooling layers, the data is flattened and passed through fully connected layers. The final layer uses the softmax activation function, which outputs a probability distribution over the target classes. In this case, each class represents a different emotion. The model then predicts the emotion with the highest probability. [29]

This combination of ReLU and softmax, along with the 2D CNN architecture, allows the model to effectively learn from the facial data and accurately predict emotions. The predictions are then communicated to the end user through audio or vibration patterns, providing a real-time, intuitive understanding of the detected emotion.

Model Architecture

Below describes the architecture of the CNN model used.

1. **Convolutional Layers:**

   - **1st CNN layer:**

     $\text{Filters} : 64, \text{Kernel Size} : (3, 3), \text{Padding} : \text{'same'}$

     Activation: ReLU, Batch Normalization, MaxPooling : $(2, 2)$

     Dropout Rate : 0.25

- **2nd CNN layer:**

  $$\text{Filters} : 128, \text{Kernel Size} : (5, 5), \text{Padding} : \text{'same'}$$

  Activation: ReLU, Batch Normalization, MaxPooling : $(2, 2)$

  Dropout Rate : 0.25

- **3rd CNN layer:**

  $$\text{Filters} : 512, \text{Kernel Size} : (3, 3), \text{Padding} : \text{'same'}$$

  Activation: ReLU, Batch Normalization, MaxPooling : $(2, 2)$

  Dropout Rate : 0.25

- **4th CNN layer:**

  $$\text{Filters} : 512, \text{Kernel Size} : (3, 3), \text{Padding} : \text{'same'}$$

  Activation: ReLU, Batch Normalization, MaxPooling : $(2, 2)$

  Dropout Rate : 0.25

2. **Flatten Layer:** Converts the output of the convolutional layers into a one-dimensional array.

3. **Fully Connected Layers:**

   - **1st Fully Connected Layer:**

     $$\text{Neurons} : 256, \text{Activation: ReLU}$$

     Batch Normalization, Dropout Rate : 0.25

   - **2nd Fully Connected Layer:**

     $$\text{Neurons} : 512, \text{Activation: ReLU}$$

     Batch Normalization, Dropout Rate : 0.25

4. **Output Layer:**

   $$\text{Neurons} : 7, \text{Activation: Softmax}$$

5. **Optimizer:**

   Adam Optimizer, Learning Rate : 0.0001

6. **Loss Function:**

   Categorical Cross-Entropy

Dataset

The dataset employed for training this model comprises approximately 36,000 face images, distributed across various emotional expressions. Specifically, it includes 5,000 images of anger, 550 of disgust, 5,000 of fear, 9,000 of happiness, 6,300 of neutrality, 6,300 of sadness, and 4,000 of surprise. To ensure model generalization, 20% of the images were reserved for validation purposes.

5 Chapter 5: Results and Validation

5.1 Overview

In this chapter we will discuss the testing of all components of the system and the results obtained. We tested all the parts to ensure that all of the functions work as expected and without errors.

5.2 Hardware Testing

This section discusses the testing process of each of our hardware components.

5.2.1 Testing Raspberry Pi

We did point testing on the Raspberry Pi by checking every point so it can help with troubleshooting hardware issues. It was not taking enough voltage and the Raspberry was giving us a warning, so we had to change the charger.

5.2.2 Testing Servo Motor

Emosense requires one servo motor to operate correctly, once the motor was connected to the Raspberry pi some tests were conducted to make sure it was up to standard:

- To make sure the motor was fully functional we made it do a simple 360 degree turn.
- Once we made sure it was working we strapped the Raspberry camera to it and run the facial tracking code, the outcome was as expected. The servo motor rotated in a way to allow the camera to always keep the subjects face in the center of its field

5.2.3 Testing Webcam

We connected the webcam to the Raspberry Pi using a USB port. It captured the photo correctly.

5.2.4 Testing Vibration Motor

At first we tried using vibration motor directly to the Raspberry PI GPIO port, but it did not give it enough current to vibrate. So, we connected it through a transistor to make it work and it did.

5.3 Software Testing

This section discusses the testing process of each of our software testing.

5.3.1 Raspberry Pi OS installation

We installed and tested various versions of "Raspbian OS" until we found the one that fits our requirements. The list below represents each test of the versions we used:

- **Raspberry Pi OS (32-bit)**
  We first tried to use it but we encountered a problem when trying to run Tensorflow.
- **Raspberry Pi OS (64-bit)**
  We used it in the project since it provided us what we need.

5.3.2 Testing face detection

The detection model we used is "Voila Jones". We used it because it provides a lightweight and fast face detection model for edge devices. We installed the model and tested it with OpenCV. It resulted with a fast detection speed around 5-7 FPS without adding the emotion recognition model.

5.3.3 Tensorflow Lite

We tested it by trying to connect it to OpenCV and the emotion recognition model. We used its detection API to run the model. It worked as expected

5.3.4 Testing pigpio

This library was an alternative to the Rpi.GPIO because it solved the jittering problem. pigpio is a special library from gpiozero library that allows you to use different subsystems for controlling the pins. It's main feature that it's hardware based timing for PWM and for servo pulses. We tested this library using python's environment and the response had no delay. It also requires the library's demon to run in the background, but it did not show any compatibility issues. The pigpio library controls the servos by running an instance of pigpio and it controls all the servos.

5.4 Testing Emotion Detection Model

Testing this model involved evaluating it on various faces, During 45 epochs of training we achieved an accuracy of 0.6. Identifying emotions proved challenging because the model relies on a single image to generate results, rendering it less accurate than desired. Consequently, we gathered five results and displayed the most frequently occurring ones, thereby enhancing result accuracy, it become close to 80%. While training the model we tested the accuracy of the model on both the training and validation data sets, which when plotted to a graph produced the figures 5.1 and 5.2.

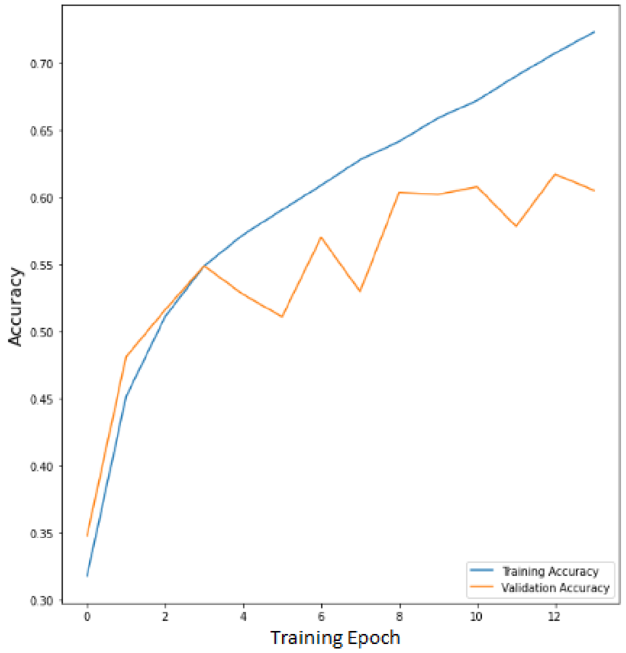

Figure 5.1: Model Accuracy.

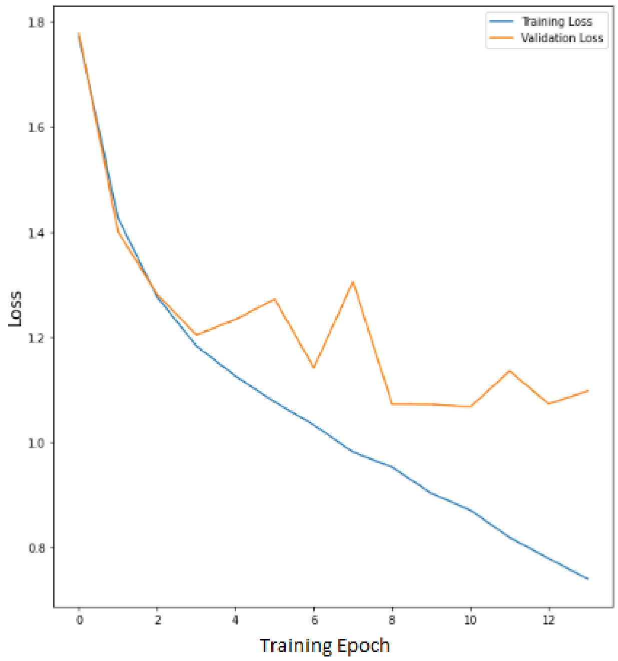

Figure 5.2: Model Loss.

5.4.1 Emotion Detection Model Results

After testing the model with the correct conditions (good lighting, one face) and taking the most prevalent emotion from 5 captures the results of our testing were accurate to the emotion portrayed by our model.

To validate our findings, we utilized a robust set of validation data consisting of 5,937 images. This validation set comprised 960 images depicting anger, 1,825 images of happiness, 1,216 images of neutrality, 1,139 images reflecting sadness, and 797 images conveying surprise.

The classifier's accuracy rates for detecting different emotions are as follows: for happy, it achieves a high accuracy of 90%, while neutral also has a strong accuracy of 90%. However, distinguishing sad poses a slightly lower accuracy at 80%. Surprise exhibits a decreased accuracy of 70%, and angry demonstrates the lowest accuracy of 50%. There is notable confusion between angry and sad making their differentiation challenging.

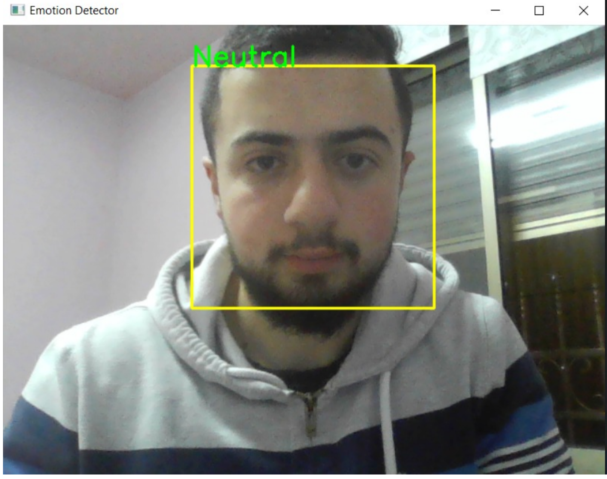

Figure 5.3: Neutral Emotion.

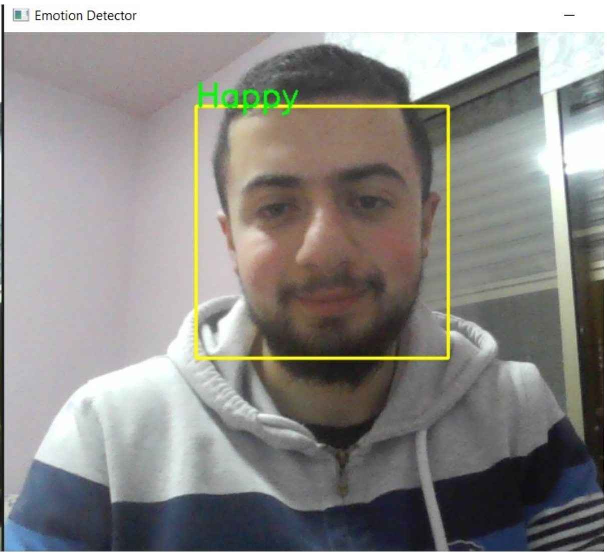

Figure 5.4: Happy Emotion.

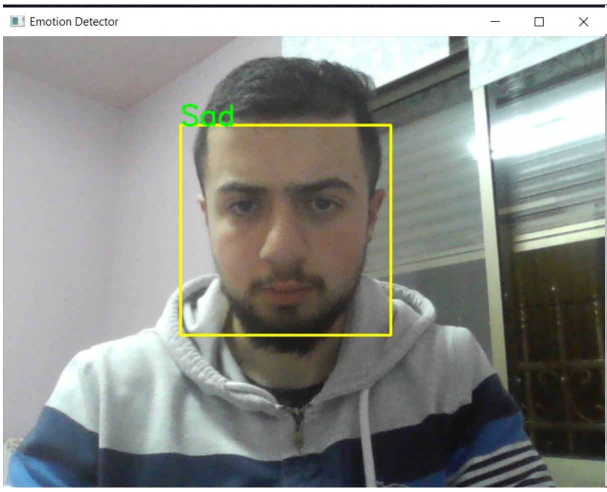

Figure 5.5: Sad Emotion.

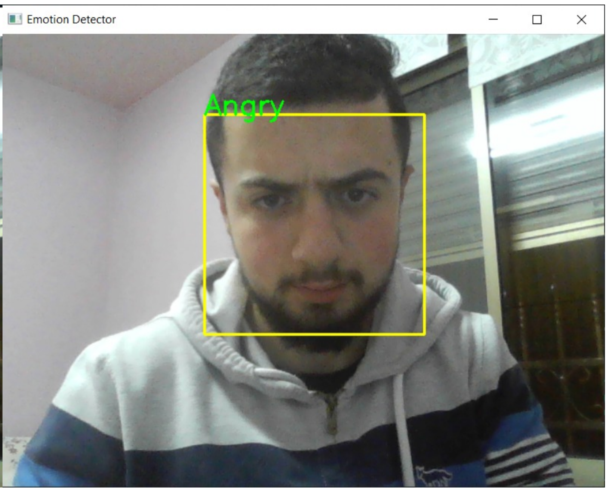

Figure 5.6: Angry Emotion.

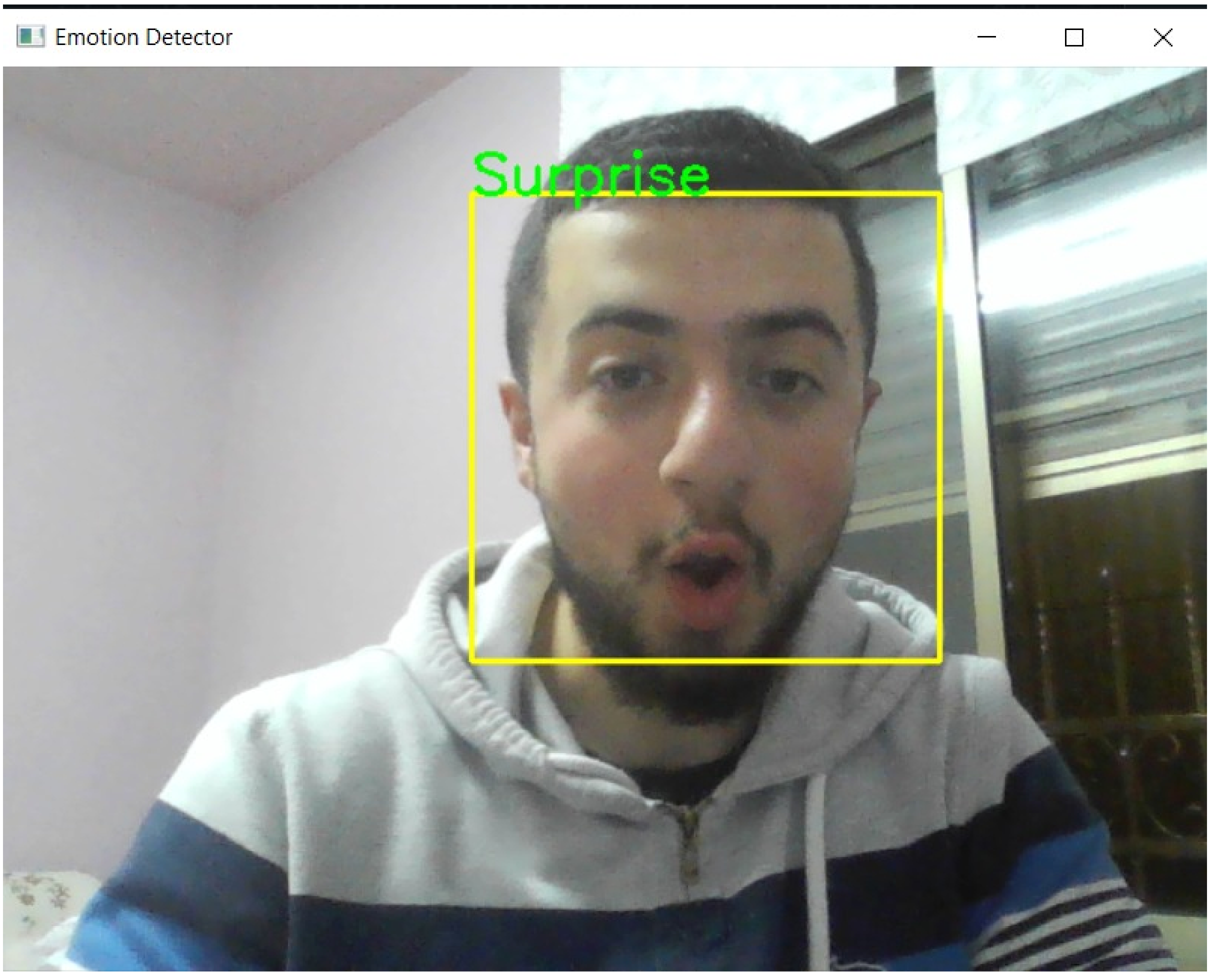

Figure 5.7: Surprise Emotion.

5.5 Biggest Face Priority Testing

As we mentioned above the Emosense system only works with one face, and it prioritizes the bigger face in its view to detect and recognise its emotions. Through testing the software in a variety of scenarios we see that the system is most likely to prioritise the closest face to it.

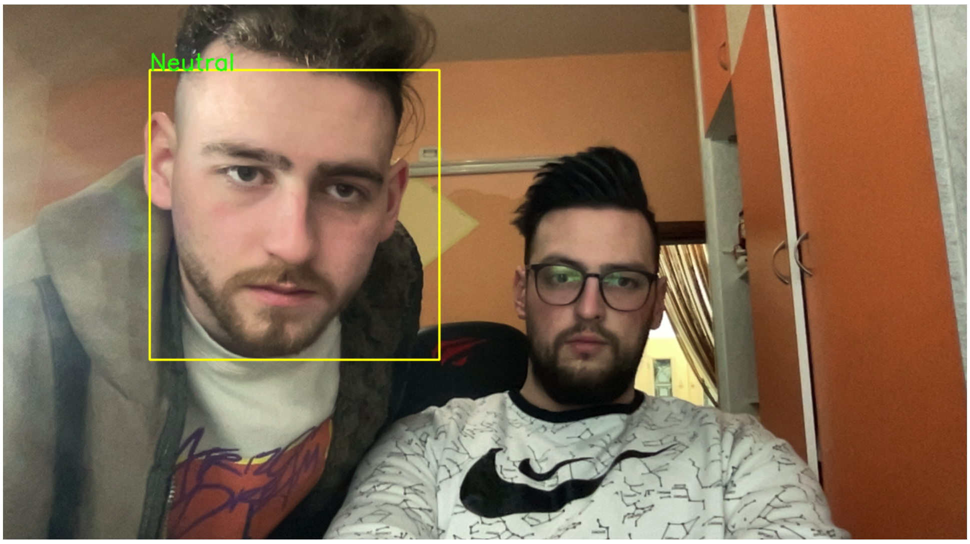

Figure 5.8: Bigger Face Priority.

5.6 Testing Vibration Patterns

Emosense relies on two feedback options, with the less intrusive one being haptic feedback through a vibration motor. Conveying complex emotions through a limited medium like vibration patterns poses a significant challenge. Our approach involves attempting to emulate the visceral sensations associated with various emotions in a digital format through the vibration motor.

- Neutral: Represented by one vibration to indicate neutrality.
- Sadness: Represented by two vibrations to indicate that it is an emotion more sever than neutrality.
- Happiness: Represented by three vibrations to indicate the human feeling of happiness which is followed with a faster heartbeat and an excited feeling.

- Surprise: Represented by four vibrations since it is an emotion than occurs less frequently than the past ones so it deserves special attention.
- Anger: Represented by five vibrations which represent the intense emotion a person feels when angry.

5.7 Test Scenarios

Table 5.1: Test Scenarios

People	**Environment**	**Expectations**	**Results**
One person	Bright, no obstacles, standing in front of the camera and not moving	The camera should not move and the emotions of that person should be given back the user	The system passed the test
One person	Bright, no obstacles, standing in front of the camera and moves left and right	The camera should move according to the person's movement the emotions of that person should be given back the user	The system mostly passed the test but it sometimes was moving too fast.
No people	Camera has no faces in front of it, No human voice in the area	The system shall do nothing	The system passed the test

Continued on next page

Table 5.1 – continued from previous page

People	Environment	Expectations	Results
No people	Camera has no faces in front of it, Human speech detected	The system shall go into searching mode	The system passed the test
No people	System is in searching mode	go back into idle mode since no faces are found	The system passed the test
Two people	Bright, no obstacles, standing in front of the camera and not moving	The camera shall recognize the closest face to it and detect it's emotion	The system passed the test

6 Chapter 6: Conclusion and Future Work

6.1 Conclusion

EmoSense is a technology-driven solution that addresses the challenges faced by visually impaired individuals in understanding the emotions of those around them during social interactions. Emotions play a vital role in human communication, and visually impaired individuals often struggle to perceive and interpret emotional cues. EmoSense leverages the power of artificial intelligence to provide haptic or audio feedback, enabling visually impaired individuals to better understand the emotional states of people in their surrounding environment.

By utilizing a Raspberry Pi 3 as its computational platform, EmoSense combines advanced algorithms to facilitate emotion detection. The system employs the Viola Jones algorithm for efficient face detection, enabling the identification of individuals in the vicinity. Subsequently, a Convolutional Neural Network (CNN) is utilized for accurate emotion detection, having been trained on a diverse dataset of facial expressions. This allows EmoSense to recognize emotions such as happiness, sadness, anger and surprise.

To enhance the overall perception of emotions, EmoSense incorporates two motors that enable the camera to track the person directly in front of the wearer. This feature ensures that the camera remains focused on the relevant individual, allowing for precise analysis of their facial expressions. The system is designed to be head-mounted, providing a user-friendly and intuitive experience for visually impaired individuals.

EmoSense presents a promising solution for visually impaired individuals, empowering them to engage more effectively in social interactions by providing real-time haptic or audio feedback of emotions. By leveraging artificial intelligence, face detection, and emotion recognition technologies, EmoSense enhances emotional understanding and fosters more meaningful connections. This technology-driven system has the potential to significantly improve the quality of life for visually impaired individuals, enabling them to perceive and respond to the emotional states of those around them.

6.2 Future Work

- **More processing**

  A proposed way to improve Emosense would be through introducing a more powerful processing unit. Thus enabling Emosense to do video analysis instead of individual photos ensuring a more accurate emotion classification.

  Another improvement that will utilise the increased resources in introducing emotion detection through the tone of the voice of the subject.

- **Low accuracy in low light conditions**

  Emosense does not have good accuracy while working in low light conditions, we thought of two improvements that could improve low light performance:

  - Installing a better camera with a bigger light sensor for its ability to capture more light thus enabling Emosense to produce more accurate results.
  - Training the emotion detection algorithm on a bigger data set containing more low light photos to better prepare.

References

[1] P. Ackland, S. Resnikoff, and R. Bourne, "World blindness and visual impairment: despite many successes, the problem is growing," *Community Eye Health*, vol. 30, no. 100, pp. 71–73, 2017.

[2] IBM, "What is machine learning (ml)?" https://www.ibm.com/topics/machine-learning, 2024.

[3] T. M. Mitchell, *Machine Learning.* McGraw-Hill, 1997.

[4] S. University, "Convolutional neural networks," Online, Accessed Apr. 10, 2023. [Online]. Available: http://cs231n.github.io/convolutional-networks/

[5] "convolution online photo." [Online]. Available: https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5

[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. I–511–I–518.

[7] J. Zhou, Z. Cao, and Q. Yin, "Overview of deep learning-based face detection," *Journal of Electronic Imaging*, vol. 28, no. 3, pp. 1–18, 2019.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 886–893.

[9] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *arXiv preprint arXiv:1808.03314*, 2018.

[12] Z. Lian, Y. Guo, X. Cao, and W. Li, "An ear wearable device system for facial emotion recognition disorders," *Frontiers in Bioengineering and Biotechnology*, vol. 9, p. 703048, 2021.

[13] M. Lutfallah, B. Käch, C. Hirt, and A. Kunz, “Emotion recognition - a tool to improve meeting experience for visually impaired,” in *Proceedings of the Conference Name*, 2022, p. 35.

[14] S. Shareef, L. Zahdeh, and B. Atawna, “Real-time stage tracking camera using raspberry pi,” College of IT and Computer Engineering/Palestine Polytechnic University, 2022, graduation Project.

[15] “raspberry pi 3 online photo.” [Online]. Available: https://www.azdelivery.de/en/products/raspberry-pi-3

[16] “webcam online photo.” [Online]. Available: https://www.konga.com/product/usb-webcam-camera-6024388

[17] “vibration motor online photo.” [Online]. Available: https://www.mouser.sg/new/dfrobot/dfrobot-fit0774-mini-vibration-motor/

[18] “servo motor online photo.” [Online]. Available: https://www.electronicwings.com/arm7/servo-motor-interfacing-with-lpc2148

[19] “2-Axis Pan and Tilt Mount Kit.” [Online]. Available: https://protosupplies.com/product/2-axis-pan-and-tilt-mount-kit/

[20] “microphone online photo.” [Online]. Available: https://www.amazon.com/Microphone-Gooseneck-Universal-Compatible-CGS-M1/dp/B08M37224H

[21] “Bc547 transistor.” [Online]. Available: https://www.circuits-diy.com/bc547-npn-transistor-datasheet/

[22] “bluetooth earphones online photo.” [Online]. Available: https://www.apple.com/shop/product/MME73AM/A/airpods-3rd-generation-with-magsafe-charging-case

[23] “OpenCV Online Documentation.” [Online]. Available: https://docs.opencv.org/4.1.0/

[24] “Keras Documentation.” [Online]. Available: https://keras.io/

[25] “pigpio online Documentation.” [Online]. Available: https://abyz.me.uk/rpi/pigpio/

[26] "tensorflow online Documentation." [Online]. Available: https://www.tensorflow.org/

[27] L. G. Martins, "Transfer learning for audio data with yamnet," https://blog.tensorflow.org/2021/03/transfer-learning-for-audio-data-with-yamnet.html, 2021.

[28] A. F. Agarap, "Deep learning using rectified linear units (relu)," 2019.

[29] G. Developers, "Multi-class neural networks: Softmax — machine learning crash course," 2022. [Online]. Available: https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax