Spring 5-14-2020

A Real-Time Internet Of Things (IoT) based Affective Framework for Monitoring Emotions in Infants

Alhagie Sallah
University of Texas at Tyler

A REAL-TIME INTERNET OF THINGS (IOT) BASED AFFECTIVE FRAMEWORK FOR MONITORING EMOTIONS IN INFANTS

by

ALHAGIE SALLAH

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
Department of Electrical Engineering

Prabha Sundaravadivel, Ph.D., Committee Chair

College of Engineering

The University of Texas at Tyler
May 2020

The University of Texas at Tyler
Tyler, Texas

This is to certify that the master's thesis of

ALHAGIE SALLAH

has been approved for the thesis requirements on
April 10th, 2020
for the Master of Science in Electrical Engineering

Approvals:

Thesis Chair: Dr. Prabha Sundaravadivel, Ph.D.

Member: Dr. Premananda Indic, Ph.D.

Member: Dr. Mukul V. Shirvaikar, Ph.D.

Chair: Dr. Hassan El-Kishky, Ph.D.

FOR JK

Dr. Javier Kypuros, Ph.D.,
Dean, College of Engineering.

Acknowledgments

I would like to express my sincerest thanks and appreciation to my thesis and graduate advisor, Dr. Prabha Sundaravadivel for her immense support and guidance. I am very appreciative of her patience, guidance and encouragement that led to the successful completion of this thesis.

I would like to thank my thesis committee members, Dr. Mukul Shirvaikar and Dr. Premananda Indic for their support and counsel despite their busy schedules. To the Department of Electrical Engineering, I say a big Thank You for providing the conducive environment for making this thesis a success.

A special thanks to my wife and daughter for their love, encouragement and support to complete graduate studies. Your sacrifices are immeasurable.

I would like to thank all my family, friends, colleagues in the lab and entire students at the Department of Electrical Engineering for the encouragement and support that led to the success of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A REAL-TIME INTERNET OF THINGS (IOT) BASED AFFECTIVE FRAMEWORK FOR MONITORING EMOTIONS IN INFANTS

Alhagie Sallah

Thesis Chair: Prabha Sundaravadivel, Ph. D.

The University of Texas at Tyler

May 2020

An increase in the number of working parents has led to a higher demand for remotely monitoring activities of babies through baby monitors. The baby monitors vary from simple audio and video monitoring frameworks to advance applications where we can integrate sensors for tracking vital signs such as heart rate, respiratory rate monitoring. The Internet of Things (IoT) is a network of devices where each device can is recognizable in the network. The IoT node is a sensor or device, which primarily functions as a data acquisition unit. The data acquired through the IoT nodes are wirelessly transmitted to the cloud to perform data analytics, thus assisting in remote monitoring. The deployment of IoT in applications such as smart healthcare, smart home, smart cities, smart transportation, and smart agriculture has made this a billion-dollar industry.

Affective computing, also known as emotional artificial intelligence, helps in developing systems that recognize, interpret, process, and simulate human affects. It is an interdisciplinary field of computer science, psychology, and cognitive science. The proposed system will be called Amb-I (Short for Intelligent Ambient Monitoring) deploys affective computing in baby monitoring through the Internet of Things. The proposed system recognizes the mood of the baby through the camera and records the corresponding ambient values through the ambient sensor array, which consists of a humidity sensor and temperature sensor. When the mood of the baby changes i.e. if the baby cries or feels annoyed, with help of the Amb-I sensing unit, the ambient values are checked, and the thermostat is controlled wirelessly, to maintain a desired ambiance for the baby. And if the baby continues to feel annoyed, the parents are notified immediately. The learning model for recognizing the mood of the baby is based on deep learning deployed through MATLAB on local PC or python libraries on Linux based small device environments. The controller for the Amb-I system is built based on the general-purpose computer, Raspberry-Pi 3. This cost-effective,

IoT-based affective ambient monitoring system helps in maintaining an ideal ambiance for babies and improves the quality of life for both parents and babies.

CHAPTER ONE

INTRODUCTION

According to the US department of labor's statistics on labor force participation of women with children age less than three years, there has been a significant increase by almost two folds in the percentages of women in this category in 2016 compared to four decades ago [1]. These statistics reflect the need for more babysitters or the increased demand for baby monitoring systems for such working parents. However, such monitoring devices require high-speed internet, which is very expensive or unavailable in some places.

Without baby monitoring frameworks, people used to tiptoe every night, occasionally to check on their little ones in the nursery. Some will walk through the dark in order not to wake the little one up only to be thwarted by some quaky doors, floorings, or some will even stumble and fall, causing injury or waking the sleeping baby. A baby monitor can help to avoid all this. At the comfort of your couch or bedroom, parents or nursing personnel can watch and monitor the activities of the infant. These baby monitors come in different forms based on requirements. The most common are the ones that fall in the following three categories.

Category-1: *Audio Monitors:* These devices monitor sounds in the environment, that allows you to control the sounds such as the cries of the baby, undesirable noise, and other faults or monitoring alarms in the nursery. Some of these devices often come with two-way communication, where parents can try to interact verbally with their little ones.

Category-2: *Video Monitors:* These devices provide real-time visual monitoring (footage), thus helping parents keeping an eye on their little ones.

Category-3: *Wearable Monitors:* These monitors keep track of the baby's vitals, such as sleeping and breathing patterns, respiratory rates, posture and movements, temperature and crying, etc. [2].

Some present-day monitors in the market may embody one, two, or all the categories. The choice of monitor used depends on different factors such as cost, installation feasibility, and, most importantly, what the parent finds more applicable for their usage. However, one major drawback of these monitors is that it requires occasional or constant monitoring to detect or tract the physical or emotional state of the infant and sudden changes in the ambient conditions. Our proposed framework, Amb-I, addresses this drawback. Figure 1 shows the conceptual overview of the proposed Amb-I system. In this research, we design, test, and prototype a monitoring system that makes use of critical technologies such as IoT, affective computing, and machine learning to address the limitations of current monitors. The Amb-I will not only act as audio, video, and wearable monitor but will use machine learning techniques to detect the emotional state of the infant and use affective computing to try to adjust to the needs of the infants without any direct human interaction. All this will be made possible through the use of the concepts of the Internet of Things (IoT) where connected devices communicate directly with little or no human intervention.

A leading cause of rashes or infections among infants is the untimely change of soiled diapers [3]. Parents or caregivers should regularly check the diapers of infants as frequently to avoid rashes or infections. This is usually difficult, especially for working-class parents. Our proposed system will monitor the diaper moisture condition, and timely notify parents to attend to the infant as soon as possible.

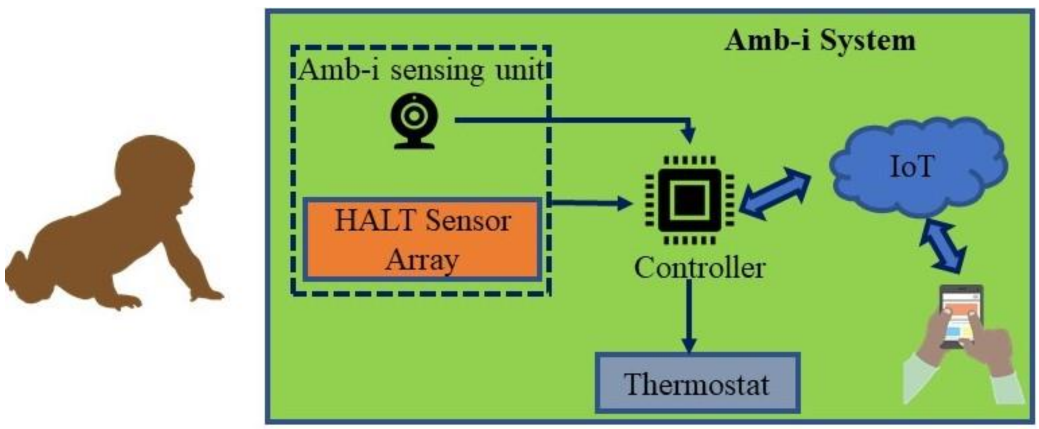


Figure 1: Conceptual Overview of the proposed Amb-I System.

1.1 Aim/Objectives

The aims and objectives of this project address some of the limitations of the current infant monitoring system and introduce a novel idea that will make baby monitoring more comfortable, more efficient, and above all, cost-effective. These objectives will be realized through the following:

1. Develop a monitoring system that can detect infants' key emotions (Happy, Crying, Sleeping, Normal) through deep learning techniques in MATLAB or python libraries.
2. An ambient monitoring system that continuously monitors the ambient conditions surrounding the infant.
3. An intelligent IoT-based affective computing system that makes use of the collected data mentioned above to stimulate or trigger the desired outcome.
4. The use of cheaper devices that will eventually lead to more affordable final products

1.2 Motivation

Peace of mind, security, safety, and critical vitals monitoring are the main benefits of a good infant monitoring system. Most working-class parents struggle through getting an affordable, effective, and efficient monitoring system for keeping up with what is happening around the baby when he/she is in the nursery. During work, travels, or sleep, it's not possible to be glued to baby monitors to monitor our baby's nursery in real-time. Limited broadband Internet access, high cost of continuously streaming the video footage, and having to call someone to attend to the baby's need while away is a nightmare barely solved by existing monitors. The concerns mentioned above mostly prevail among many parents which were the motivation to design a cheaper, more cost-effective, and smarter monitoring system for infants.

1.3 Current Monitoring Devices

There are a lot of different commercially available monitoring devices in the market today. Current monitoring devices that are closely related to our proposed framework include but are not limited to, the following:

1. *Nanit Plus Smart Baby Monitor:* This has audio, video, and sleeping pattern monitoring capabilities. It can provide a review of the sleeping pattern, room conditions to quickly detect changes. It offers real-time sound and video footage that can be viewed on a smartphone. It works with Wi-Fi to easily connect to the internet and provide a two-way communication that allows you to communicate with your little one from afar.
2. *Motorola Halo Baby Monitor and Soother:* This monitor in the market has audio, video, and sleep pattern monitoring capabilities too. Also, it has a built-in speaker to play soothing sounds to promote sleep as well as a two-way communication mechanism to interact with

the infant. It also has infrared night vision to help visually see infants clearly through the dark.

3. *Infant Optic baby Monitor:* This has all the features of a monitor described in the above two. Additionally, it has the innovative lens technology that allows you to customize camera performance by switching different lenses on the camera.
4. *CasaCam Video Baby Monitor*: This camera has a large 5-inch LCD that has a touchscreen display and is very easy to operate. It can be adjusted remotely and supports up to six different languages.
5. *AXBON Wireless Video Baby Monitor:* This has a temperature sensor for detecting if the baby's room is too cold or hot. It also has a built-in alarm that can be enabled to remind parents to change their baby’s diapers or check them periodically.
6. *Summer Infant Video Baby Monitor:* This has a temperature sensor for detecting if the baby's room is too cold or hot. Also, it has a zoom in/out functionality for closer observation of the baby's activities. It can connect up to four baby monitor cameras.
7. *VTech Audio Baby Monitor:* This monitor is strictly for audio monitoring. It has a built-in DECT 6.0 digital technology to reduce background noise to hear the baby’s sound.
8. *Owlet Smart Sock and Cam:* This is one of the most expensive but most spectacular baby monitors out there in the market. It has audio, video monitoring and temperature monitoring capabilities as well as the ability to monitor heart rate and oxygen levels in babies.

1.4 Organization of Thesis

The novel contributions of this research are emotion detection in real-time and the ability to collect, analyze, and react to this data using the concept of IoT. The research in this Thesis is explained in the following five chapters: Chapter 2 looks at a brief overview of previous research in infant monitoring devices or studies. Chapter 3 describes the technical knowledge or concepts involved or used in this project. Chapter 4 describes the method and implementation strategy for this project. Chapter 5 detailed out the discussions, observations, problems faced, conclusions and suggested future works.

CHAPTER TWO

LITERATURE SURVEY

Before delving into the research for our proposed framework, lets discuss the current research work that been done in areas that are foundational for the same. Future IoT projects that will continue to shape our relationship with our surroundings.

2.1 Ambient and Health Monitoring Systems for Infants:

G. Joshi et al. have proposed a system that would supposedly predict the probability of crying of a child, detection of the crying, and the financial viability of this system. The system would target working-class people who don't have the time and resources to attend to their young ones sometimes due to work or certain predispositions. Cheap hardware used was a raspberry pi, pic camera unit, and a microphone and a Microsoft cognitive services capable of image analysis, text analysis, language analysis, and Human Emotion analysis for Emotion detection and prediction [4].

A. Osmani et al. described a Machine Learning Approach for Infant Cry Interpretation. Their paper research focused on reducing infants' distress. They produced real connected objects implementing this solution for smart baby monitors. It proposes a machine learning process that includes a reliable dataset of infant cries and selecting suitable sound features to detect and analyze discomfort (Cries) of infants later automatically. This machine learning process includes a low-level audio features selection method from labeled pre-cry recordings and the high-level features characterizing the envelope of the crying [5].

Chuan Yu. Chang and Fu-Ren Chen use the concept of deep learning (TensorFlow) to detect crying or face covered vomit among infants. The blockage of the nose and mouth by vomit or coverlets are major causes of sudden infant death. They have used Single MultiBox Detector for classification and mobilenetis for prediction. The public face database-WIDER FACE dataset was used for training the neural network and Gaussian filters for image preprocessing. The area of the mouth is estimated so that when there is an object (vomit or coverlets), the new area of the mouth region is different from the initial estimate. This region of interest was used to predict the presence of vomit or object in the mouth region. The same concept was applied to predict crying action amongst infants [6].

Madhuri P. Joshi and Deepak C. Mehetre designed an IoT based smart cradle system for monitoring babies through an Android App. The paper presented a design of a cradle system that supports video monitoring, as well as swing automatically on detecting the baby crying. This is to try to soothe the baby before the baby gets attention from its caregiver. If the baby continues to cry after a certain time, a notification is sent to convey a message that the baby needs personal attention. It also has a wet bed alarm that is useful not only to monitor the comfort of the baby but also help to keep a healthy environment around the baby. All this helps to bridge the gap between parents and children anywhere and anytime [7].

Ananda Mohan Ghosh et al. presented a paper that detailed the design of a health care system that allows relatives or doctors to remotely monitor the health condition of patients over the internet using Arduino and some sensors [8]. However, if the same concept is applied to monitoring children at home during working hours, the flaw if this system is quite apparent. It will require constant surveillance or monitoring of the system for any changes in the patient. This will be nearly impossible for a working-class parent or relative. This is why we propose a system that will do the monitoring, soothing reactions, and human intervention is only required when extremely necessary. Similarly, a raspberry Pi-based patient monitoring system by P Kumar et al. [9] where heartbeat, respiration, and temperature are measured using sensors and displayed using putty software cannot also react to the data automatically without human intervention.

Xiaoting Liu et al. [10] proposed a video-based IoT monitoring system for Sudden Infant Death Syndrome (SIDS). It can reduce response time by using Eulerian Magnification, a video amplification technique to amplify subtle movements by comparing the color difference in frames for breathing detection. An alarm is automatically generated to notify parents or guardians when a breathing abnormality is detected.

A. Archip et al. [11] proposed a system a low-cost modular monitoring system for remotely monitoring patients. This could be neonates or patients in an ICU following surgery or complicated illness. Their proposal offers mobile support to facilitate faster and better medical intervention in an emergency. The design includes sensor arrays for SpO2, temperature and movement capturing, a microcontroller, and interface all to an IoT platform (RESTful based Web Service).

S.P. Patil et al. [12] proposed a body monitoring system that creates peace of mind for parents or caregivers. The proposed system will monitor health conditions such as body temperature, moisture, and movement of the baby and will automatically send out signals during an emergency using GSM networks. The architecture of their prototype includes sensors, GSM Module, LCD screen, and a sound buzzer. However, their prototype does not make use of the internet to be accessible from afar.

M. Leier et al. [13] proposed a miniaturized wireless monitor for long-term monitoring of newborns. The architecture of the proposed infant monitoring prototype consists of aspects critical to long-term use and convenience. The two key aspects included the physical size of the product for convenience and the use of Bluetooth Smart Wireless protocol to increase the monitoring life cycle.

Health-related disease conditions are continuously increasing, and therefore an accurate, cheap, and portable heart rate and body temperature measuring device are essential. And it will enable timely intervention before or during an emergency. The focus of A. Miah et al. research in [14] was the same. Their proposed system will provide health information such as heart rate and body temperature in real-time through a connected Android platform.

N. Indumathy et al. [15] and B. Priya et al. [16] proposed android based systems for monitoring the health condition of patients using different sensors such as temperature, heart rate, and eye blink detection sensors. Data from these sensors are being used to predict or determine the condition of patients.

Hata, Y et al. [17] proposed a human health monitoring system for bedridden patients. The system continuously monitors the patients' health vitals. A specified doctor's cell phone is notified via SMS when a critical condition is detected. Nambu, M et al. [18]proposed a similar system for monitoring health conditions in homecare systems. A 24-Hour health monitoring system aimed at helping independent living of elderly patients in a smart house was proposed by L. Heyoung et al.

[19]. It consists of a biosignal sensing part, a monitoring system for caregivers, and a local PC for processing the data.

F.Guo et al. [20] and S. Brangui et al. [21]designed a prototype for monitoring the environment of a baby's room. Their proposed systems had sensors for temperature and humidity, a wet alarm, infrared alarm and other information collecting systems. This will allow the family of the baby to access, observe, or monitor in real-time the environmental parameters of the baby at home.

N. Zakaria et al. [22] proposed an IoT based infant body temperature monitoring. They pointed out that parents and caregivers are usually not aware of the drastic change in temperature of infants unless a device that can continuously monitor the infant body temperature or environment is used. Due to power requirements. such devices are hard to get, and they proposed a smaller and lower power system to provide continuous monitoring. The proposed system in this project will provide not only continuous monitoring but also the ability to react to the measured abnormal temperature and effectively and efficiently alert parents on the condition of the infant.

A temperature monitored IoT based smart incubator was proposed by A. Ashish [23] to provide a stable and controlled environment in the enormous care of premature babies. Through the use of low-cost devices such as raspberry pi, temperature sensors, relays for thermostat switch control, and IoT, the proposed system sought to provide a cheaper and more eco-friendly alternative to existing high cost, large and complex existing incubators.

E. Saadatin et al. [24] proposed in their paper a low cost, mobile-based monitoring that can continuously monitor the baby and remotely update the parent or caregiver about the status of the baby. The system continuously collects sensor data such as temperature, heart rate, and sends it to a server. This server then immediately notifies the parent as soon as abnormal data is collected. This is a great innovation to infant monitoring, but the drawback is that it cannot react to the given data besides notification. Reactions such as remotely adjusting the thermostat, playing soothing sound, or swinging the cradle to comfort the infant before human intervention is available.

B. Mohammad et al. [25] proposed a portable embedded system that can provide a continuous temperature monitoring system for babies disables or older adults. The includes a single microcontroller that picks data from temperature sensors and feeds this to an LCD or activates a sound buzzer to alert or notify caretakers when an abnormal temperature is recorded. This is quite cheap and simple to implement, but unlike the proposed system in the project, it lacks the intelligence and ability to detect other events surrounding the infant, and also it does not have the ability to react to the abnormal data besides simple notification.

S. Thomas et al. [26] also proposed a system in 2016 for digitally sending heart rate and body temperatures using Arduino and an android platform for display. The Arduino microcontroller was programmed to receive data from the sensors attached to the body, and through the internet, it can relate this data to the user's android application.

A novel continuous infant temperature monitoring and the alerting system was designed by M. Morthi et al. [27]. It was targeted for people with limited literacy or in rural settings. It uses light or sound to call to attention a caregiver whenever a high temperature is detected. Faruk et al., also proposed a similar system for hard of hearing parents. In their system, a vibration or led blink is generated whenever an abnormality is detected.

Pradeep Doss M et al. [29] develop a non-invasive monitoring system using raspberry pi, IoT, and sensors. Sensor data such as temperature, heart rate, respiration are collected and uploaded to an IoT cloud platform for analysis or sending of notifications to parents/caregivers.

A. A Joseph et al. [30] proposed a smart health monitoring system that can be used by parents to monitor the health conditions of their children, especially those under five years. The prototype consists of a temperature sensing unit for detecting fever or other abnormalities and notifying the parents through an alarm system.

2.2 Humidity Monitoring in Infants:

J. Siden et al. [31] described a wet diaper detection system. Their prototype is a paper-based, disposable, and moisture-activated RFID system that can be incorporated into diapers. It transmits radio signals upon contact with moisture that could be used to alert the caregiver. However, even though this might be cheap to build, incorporating it in all diapers may be undesirable or increase the cost of production and may likely not be adopted. The proposed system in this project will be more realistic since a single monitor will be used to measure the change in humidity in the diaper region as well as the surrounding environment.

M.S. Tuma and Y. Kim [32] proposed a system of diaper monitoring using the Impedance Variation of a Dipole Antenna. This antenna will be attached to the diaper to sense the urine and feces of infants by relying on variations in the antenna's input impedance. Similar to other prototypes that require new additional devices to embed into existing diapers, it may increase the cost or will less likely be adopted.

T. Khan [33] proposed a smart wearable gadget for diaper monitoring that is both noninvasive and can send a notification to caregivers or parents. The proposed gadget detects the temperature rise on the outer surface of the diaper due to urination. The gadget is small, low power, and unlike previously proposed gadgets, it is reusable. It also has the functionality to log urination events in a database that could be used to track other disease conditions such as retention and dehydration etc. However, its drawback is that it monitors only one aspect of the infant's health or environment.

P. Sen et al. [34] proposed a low-cost method for detecting the wetness of diapers using hydrogel-based RFID tags. The diapers embedded with RFID tags would communicate with a nearby RFID reader. This reader helps in connecting to the internet would then convey the message to the parent/caregiver if urine or moisture is detected.

M. Y.E Simik et al. [35] proposed an automatic alarm system for wet diapers using GSM. The wet diaper, which comprises an elongated pair of spaced conductors, detects wetness of diaper due to a change in resistance between the conductors. An RF module is then used to transfer the signal from the diaper sensor to a GSM alarm system. Another initial approach towards a paper-based diaper sensor was proposed by M. McKnight et al. [36] to detect not only urine but other critical physiological parameters wirelessly.

Ming-Hui Wen proposed a Goo9 system that comprises a sensor (temperature and humidity) module, a data processing module, and a cloud database. The processing module will analyze sensor data, and output is represented based on wetness levels (dry, slightly wet, wet, and very wet). These wetness levels are translated into emoticons (Happy, Mild, Uncomfortable, and

Crying) and are sent to parents for baby change reminders [37]. A similar design using Goo9 was also proposed by Q. Zhang et al. using SHT21 sensors [38].

2.3 Emotion Recognition in Infant Monitoring Frameworks

On the image processing part, the idea of using existing data to train a system to recognize emotions arise from the studies or prototypes discussed below.

J. Mukhopadhyay et al. [39] did a study aimed at testing the hypothesis that seeks to evaluate the human perception of neonatal cry using a database of 315 neonatal cries. Each newborn cry is associated with one class (hunger, pain, wet diaper, and others). After training participants, different cries are presented to participants to predict the actual type associated with the cry. This is the basic idea behind the image processing part of this project. With the advent of artificial intelligence, this can be implemented in a system quickly and efficiently using neural network techniques. A similar study was done by Aomar Osmani et al. [40] using a machine learning approach.

Another system that was studied to help develop the basis of the project is the paper by K. Kirana et al. [40] that looked at facial emotion recognition based on the Viola-Jones algorithm. Here, facial features were used to detect, differentiate, and recognize emotions.

Extracting facial features from a real-time image capturing system (a camera) and using appropriate classifiers and techniques such as SVMs, decision trees, and random forest, a supposed accurate and efficient emotion detection system was proposed by B.T Nguyen et al. [42].

Suchitra et al. [43] proposed a method for real-time emotion recognition using raspberry pi. In their system, they used three-step face detection using Haar cascade, used Active Shape Model (ASM) for feature extraction, and Adaboost classifier for emotion classification.

CHAPTER THREE

TECHNICAL BACKGROUND

The technical background discussed in this chapter is centered on the three critical components of this research: emotion recognition using cost-effective wearable, ambient monitoring, and the IoT framework. Figure 2 shows an overview of the proposed project. First, let us discuss the application of fundamental concepts of deep learning architecture, in image and/or emotion processing techniques such as this framework. The discussion on deep learning and MATLAB will lead to the implementation of the emotion recognition part of this project; the single board computer, Temperature, and Humidity sensors, as well as the IoT cloud platform (ThingSpeak), will aid the ambient monitoring portion of this project. Further discussion on all these concepts leading to implementation is put together to form Amb-i in the next chapter.

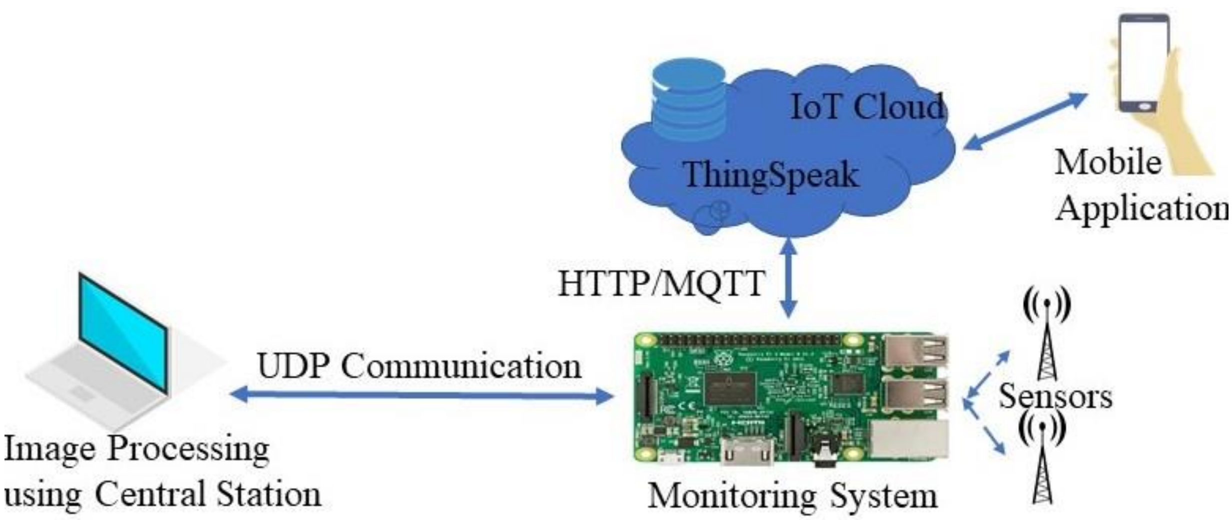


Figure 2: General Overview of the Proposed Project

3.1 MATLAB

MATLAB is a high-performance language that allows easy implementation of algorithms, creation of user interfaces, and interfacing with other languages [44]. It has optional toolboxes specifically designed to accommodate high computing abilities in algebra, simulations, image processing, artificial intelligence, controls, and embedded systems. These toolboxes are expertly built, thoroughly tested and fully documented. This is why we see it as our choice for building the emotion recognition part of this project.

3.2 Neural Network

Neural Networks are techniques used for building artificial systems that can learn from a set of existing data and use what was learned to predict, classify, or process future data. Figure 3 gives a simple overview of a neural network showing input, output along with hidden layers. It works similarly to how the human brain sees, learns, and processes data around us. In order words, they are designed to recognize patterns in data. Neural networks are usually stratified in layers, usually an *Input Layer*, one or several *hidden layers*, and an *output layer*. Each layer consists of *nodes* where computations are done. Between the nodes or layers are a set of coefficients called *Weights* that can amplify or dampen the input or output of a node.

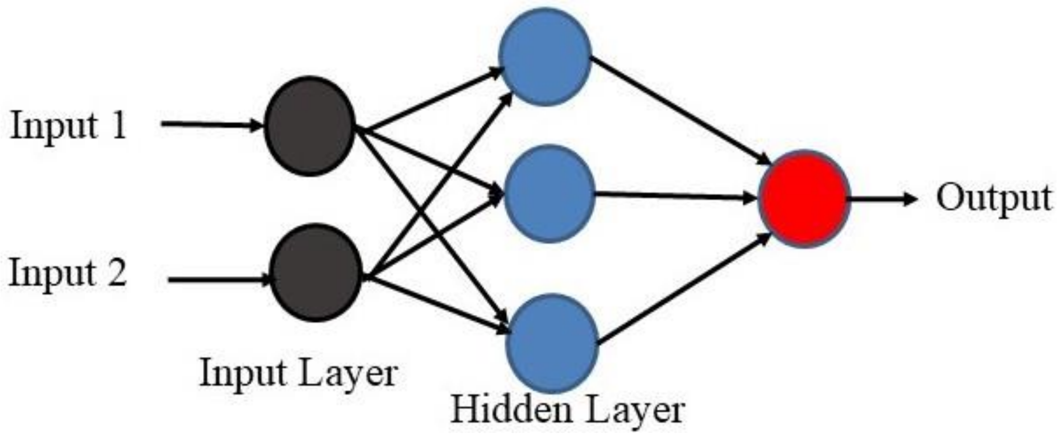


Figure 3: Simple Neural Network

The product of preceding inputs and weights of a neuron are summed and passed on to an *activation function* to determine to what extent such input or weight will be used in training the network.

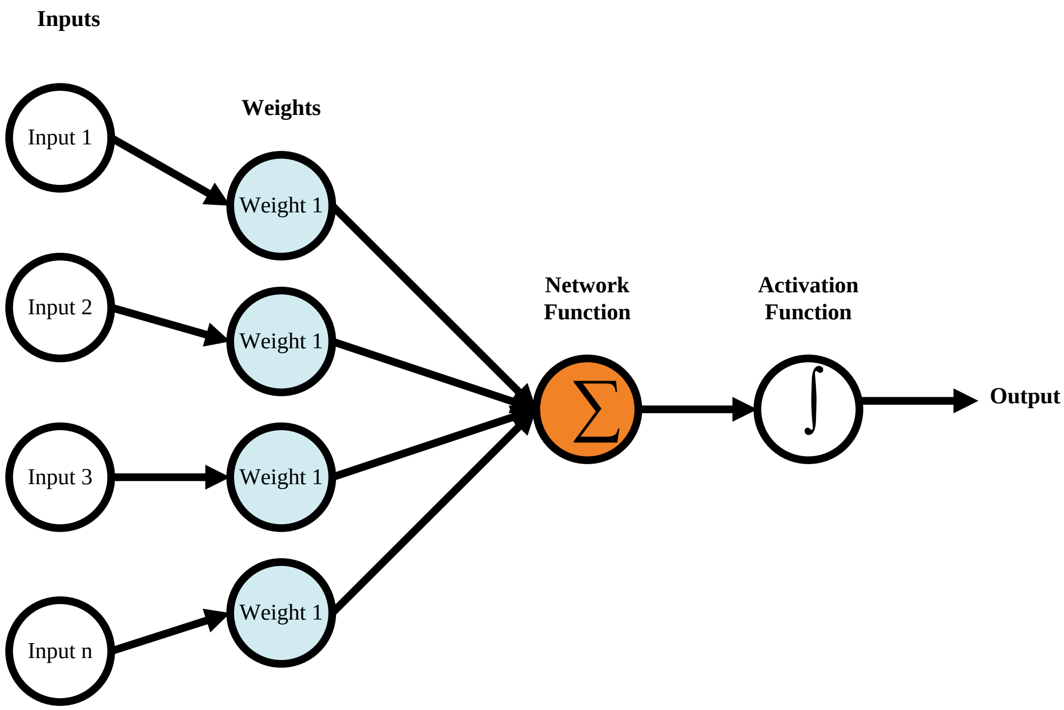


Figure 4: Relationship between Inputs, weights and Activation Function

Depending on the number of hidden layers, a neural network can be classified as a *shallow* or *deep* neural network. The single-layer perceptron is the simplest form of a neural network with no hidden layers. The multilayer perceptron has one or more layers, and they are the ones with most practical applications nowadays.

The type of input data and/or output requirements usually determines the best neural network architectures to adopt. The main idea behind these neural networks is to learn special features in each input data to form a relationship, classify, or learn patterns from the data. This feature extraction can be done manually or automatically by the neural network as data is passed from the input layer through the hidden layer(s) to the output layer. The manual extraction is usually very complicated and tedious to design. The deep learning networks learn high-level features automatically as data is passed through its numerous hidden layers and hence a very suitable choice for my project.

3.2.1 Deep Learning or Deep Neural Networks

These are neural network models or techniques with multiples of layers, that take input and extract several features for training to perform classification, localization, object detection, instance segmentation, or speech recognition.

A convolutional neural network (CNN) is one of the most popular algorithms for deep learning. They are instrumental in finding patterns in images to recognize objects, faces, and emotions. They classify images using patterns learned directly from the image dataset. CNN enables us to avoid the tedious method of manually extracting features. These CNNs, once built, can be trained for recognition of different tasks without necessarily making a new neural network. A CNN, like most deep learning networks, can have tens to thousands of layers that learn to detect a feature in the image with filters applied to each image at various resolutions.

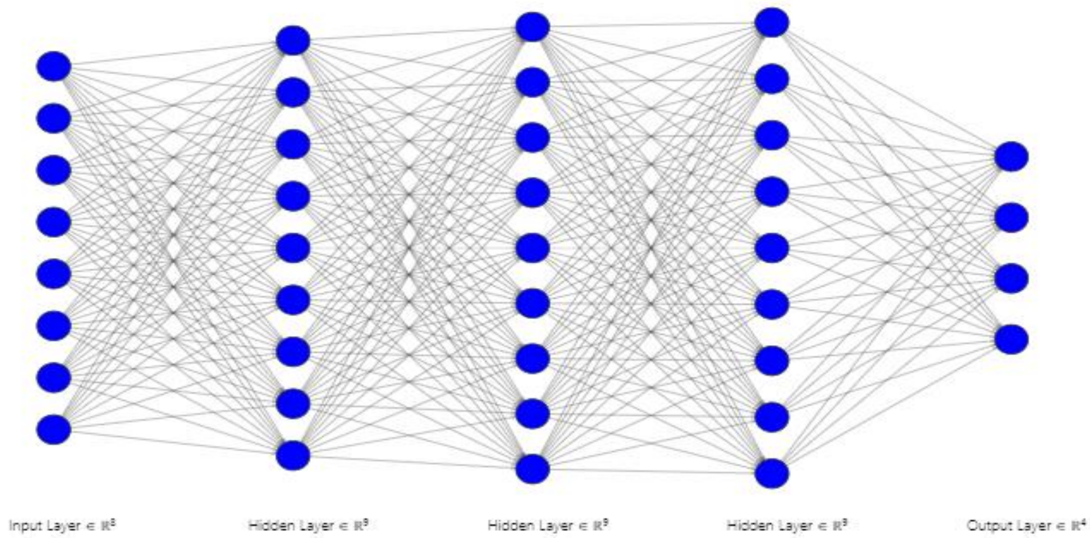


Figure 5: A deep Neural Network Architecture

In terms of functionality, a CNN architecture consists of several components such as convolutional layers, pooling layers, fully connected layers, activation function, input layer, and output layer.

Convolutional Layers

This layer is the building block of convolutional neural networks. The convolutional layers consist of a rectangular grid of neurons whose input is also a rectangular grid of neurons from a previous layer. It takes an input and maps it to a feature map of that input before passing it to the next layer.

A convolutional layer usually has three attributes; Convolutional filters (Kernels), the number of input and output channels and these channels must be equal.

Pooling Layers

Pooling layers are used to minimize the dimensions of a given data by combining the outputs of a layer into a single neuron in the subsequent layer. These layers are crucial in simplifying the outputs of a convolutional neural network by reducing the number of features the network needs to learn. This leads to a reduction in the number of parameters and computations in the network and thus helping to control overfitting. The reduction can be applied on a small cluster (local Pooling), applied on all neurons of that layer (Global pooling), or the use of the maximum value in each of the clusters in that layer (Max Pooling).

Fully Connected Layers

In these layers, neurons between any two adjacent layers are fully connected pairwise but no intralayer connections. This full connectivity often leads to a problem in neural networks known as *overfitting.* Fully connected layers are placed before the classification output to flatten the results before classification.

Activation Functions

The activation functions are included in neural networks to introduce non-linearity, which is so important because most real-world data are nonlinear. These activation functions perform certain mathematical operations on neuron data. There are several types of activation functions, and a few major ones are described below:

Sigmoid: This takes several real-valued inputs and squashes it to the range between 0 and 1. In other words, it sets large negative numbers as 0 and sets large positive numbers as 1. However, it has a few drawbacks. It saturates and kills very small gradients resulting in no leaning by the neurons, especially when initial weights are large. Its output is not zero-centered as a result of setting all output between 0 and 1, which could affect computation in the subsequent layers.

$$S(x) = 1/(1 + exp^{(-x)}) \qquad \text{(Eq.1)}$$

Tanh: This takes several real-valued inputs and squashes it to the range between -1 and 1. Just like the sigmoid, its activation also saturates, but its output is zero-centered, making it the better non-linearity option to sigmoid.

$$T(x) = 2\sigma/(2x - 1) \qquad \text{(Eq.2)}$$

Rectified Linear Unit (ReLU): It is a half-wave rectifier function that is thresholded at zero [45]. It maps all negative values zero. It is effortless to implement, and its convergence is faster compared to the tanh or sigmoid functions. It is given by the equation [3] below.

$$R(x) = \max\,(0, x) \qquad \text{(Eq.3)}$$

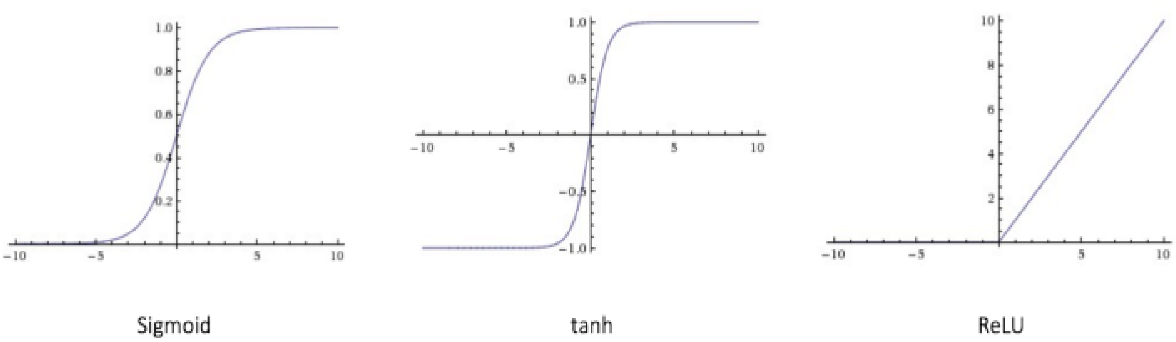


Figure 5: Graphs of activation functions [46] (a) sigmoid (b) tanh (c) ReLU

3.2.2 Training Datasets

In deep learning, one can train the CNN with data from scratch or use pre-trained models with minimal input training data in a new process called transfer learning. The former is not necessary for this project as it requires massive amounts of data (millions of data) and a long time for training. With the latter (transfer learning), minimal data (few hundred) can be used on a pretrained network that had already been trained with millions of images and retrained for our new object classification [47]. Examples of such pretrained networks include AlexNet, GoogLeNet, ResNet-101, ResNet-18, DeepNet, SqueezeNet etc.

3.2.3 Overfitting and solutions (Regularization)

When a convolutional neural network is trained with a model that is so close to a particular set of data that the network fails to fit or predict another set of data, a phenomenon called *Overfitting* arises. This could be as a result of more parameters that are not related or justified by the data.

Regularization is the process of preventing overfitting or mitigating its effect. The following are various types of Regularization used in CNNs:

Dropout: This is a process where certain individual nodes are dropped out of the network, and only some are allowed to progress or trained in the subsequent layer. In CNN, this is usually probabilistic, with dropped nodes (**1-P**) and kept nodes with probability **P**. Reducing the number of nodes trained decreases overfitting, increases the training speed, and allows the active node to learn key features that are specific to the data.

DropConnect: This is similar to the dropout mentioned above, but unlike dropout, individual connections are dropped with probability (**1-P**) rather than the whole output unit or neuron.

Other methods of preventing overfitting include stopping the training before overfitting occurs (*Early Stopping*), reducing the number of parameters, hidden layers or depth of network or filter sizes, or providing more training examples to the CNN (Artificial Data Expansion).

3.2 Ambient Monitoring

3.2.1 Single Board Computer (SBC)

To make this project possible, like most smart device projects, a microcontroller or a single board computer (SBC) is required to put the pieces together. These SBCs represent part or whole computers but in much smaller sizes. They have different interfaces for connecting and/or communicating with other devices or the internet. Some SBCs support the adding of external modules through its numerous slots, while some have all the key components embedded in the device [48]. The former is usually cheaper and offers more flexibility while the latter are usually smaller, with fewer power requirements.

Figure 6: A Single Board Computer: Raspberry Pi 3 B+

The most common interfaces on SBCs are:

- Universal Serial Boards (USB) for interfacing USB devices such as keyboards, mouse, GSM modules, etc.
- Serial Peripheral Interface (SPIs) is a four-wire serial bus used to attach numerous devices to a set of pins.
- Inter-Integrated Circuits (I^2S) are also used to connect different peripheral devices.
- Universal Asynchronous Receiver/Transmitter (UART) is a communication protocol used to control GPIO pins, and control kernel boot messages .
- Networking/Communication interfaces and Protocols: This includes the wireless interface (Wi-Fi), Ethernet, or Bluetooth interfaces for interconnection with other devices or the internet.
- Multimedia interfaces.
- Audio/Video Input/output.
- Serial Data/Multimedia Card (SD/MMC) cards for interfacing storage media.

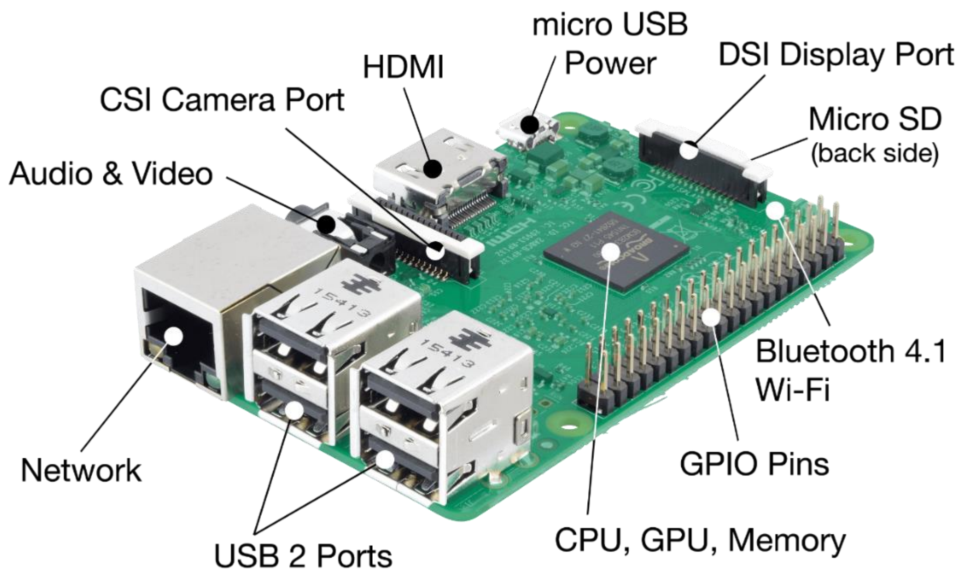


Figure 7: The Major Parts of a popular SBC (Raspberry Pi) from
https://www.vippng.com/preview/hbhbmbo_components-of-raspberry-pi-3/

A power supply unit is usually embedded or externally supplied via a dedicated power port or other interfaces such as the USB. Embedded with power regulator components, voltage converters, transformers, and power distributors. Batteries can also be used to power SBCs.

The Processor architecture comes in different forms with Intel being amongst the most popular. ARM cores are low power processing systems that are very common SBCs used in IoT projects.

The SBCs will require a software platform such as Windows, Linus, Androids or any other variant of the three in order to perform its functionalities. They are usually relatively cheap, can be configured in clusters and can be easily configured as smart IoT devices with high flexibility [49].

3.2.2 Temperature and Humidity Sensors

The Digital Output Relative Humidity and Temperature (DHT22) is used for measuring temperature and humidity simultaneously. They are low-cost sensors with excellent performance capabilities. The supported range for temperature measurement is -40 to 125 degrees Celsius with a +/- 0.5-degree accuracy. Its humidity measuring range is 0 to 100% with 2-5% accuracy. The DHT11 is cheaper but with lesser measuring ranges or specifications, 0 to 50 degrees Celsius for temperature, and 20 to 80% humidity measurement ranges [50].

The sensor consists of three parts, a temperature sensing module(NTC thermistor), a humidity sensing module, and an integrated circuit (IC) at the back of the sensor.

The thermistor is a variable resistor that changes its impedance with the change in temperature, usually made from materials that provide a significant difference in resistance with a slight change in temperature. Negative Temperature Coefficient (NTC) implies resistance is inversely proportional to the changes in temperature.

The Humidity sensing part has two electrodes with a moisture-holding substance in between them. As the humidity changes, so does the conductivity of the material or impedance between the electrodes. This change in conductivity or impedance is what is expressed as humidity values displayed by the sensor.

The DHT22 is an easy-to-connect sensor with 4 pins. Figure 9 shows the example pin diagram of DHT22 sensor. It has 4 pins:

1- VCC- Connects to the power supply
2- Data Out
3- Not Used
4- Ground

Since 1 of the pins remain unused, in figure 9 it is not included. A 10K Ohms pull-up resistor is placed between the data pin and VCC to keep the data line high and enable good communication between the sensor and microcontroller. A DHT library was installed to take care of all the protocols and timings.

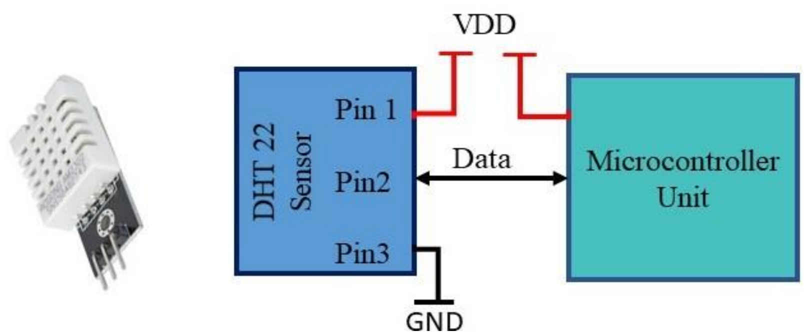


Figure 8: A sample of Humidity and Temperature Sensor (DHT22)

3.3 Internet of Things

IoT is short for the Internet of Things. It is a phrase coined to describe a platform of technology where devices communicate, interact, stimulate, or react to their environment. The word "internet" describes the era where humans control processes or communicate with each other via things (workstations). A new era has emerged where human interaction is limited or made unnecessary to reduce cost, increase productivity, accuracy, and efficiency. This is the era of the Internet of Things. Current rapidly evolving technological trends such as widespread connectivity, increased IP adoption, advanced data analytics, cloud computing, and artificial intelligence have made the adoption of IoT very rapid and extensive [51].

IoT has seen its widespread adoption in different aspects of our lives. Such vital areas are but not limited to, the following:

Smart Health Care: Healthcare IoT is used to connect patients with their health care providers in a manner that would save costs and increase efficiency in a way that would have been difficult or impossible to achieve. A remote patient can continuously collect and share his/her health data with the health care worker in real-time. The health care worker can observe, diagnose, treat, and monitor patients and disease conditions remotely, all partly due to the advent of technologies, including IoT at the center [52].

Smart Agriculture: IoT is also employed in agriculture to improve productivity and cut down costs and increase efficiency. With smart Agriculture, farmers can predict complex weather conditions for higher yield and the application of IoT tools to economize available resources and ensure optimal utilization.

Smart Cities: Modern cities now use IoT for interconnecting resources to better understand the trends of the city and provide or suggest solutions to central problems. Today, most cities have CCTVs installed in almost every corner, traffic lights, and toll roads. This has dramatically helped to improve security, cut costs on human surveillance, reduce accidents, and improve efficiency. Camera footage of offenders are now automatically collected and sent to the correctional department. Toll users or traffic violators are automatically mailed their bills saving time and cost.

Industrial IoT: Nowadays, most industries, especially chemical industries, have smart devices for monitoring oxygen levels, temperature, and toxic gas inside the plant to ensure the safety of workers and goods.

Smart Home: IoT has also found its immense use in most modern homes. It connects our appliances, sensors, and other utilities in a network that enables these objects to complete designated tasks and communicate with each other without any human intervention. A smart bulb or AC unit in a smart home can be controlled from a smartphone. A smart refrigerator that takes inventory of its content and sends a notification when it is due to refill.

In this project, we will make use of the concept of IoT in the direct monitoring of infants and the immediate environment.

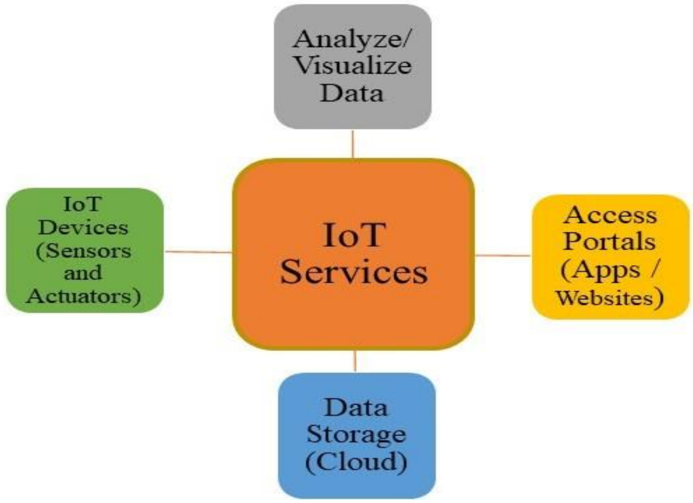


Figure 9: An IoT Ecosystem

3.3.1 ThingSpeak

ThingSpeak is a cloud-based IoT analytics platform that allows users to gather, visualize, and analyze data in real-time [53]. It is an open-source platform and interfaces for storing and retrieving data from things (sensors, etc.) using standard protocols such as HTTP and MQTT over the internet or local area networks. It has the tremendous ability to generate plugins or applications for connecting or communicating with web services, social networks, and other applications.

The core element of ThingSpeak is the ‘ThingSpeak Channel.’ This is what is used to store, isolate, or retrieve data. It encompasses an eight field for data storage of any kind, 3 locations field (longitude, latitude, and elevation) for tracking motion and a one status field for short messaging or description.

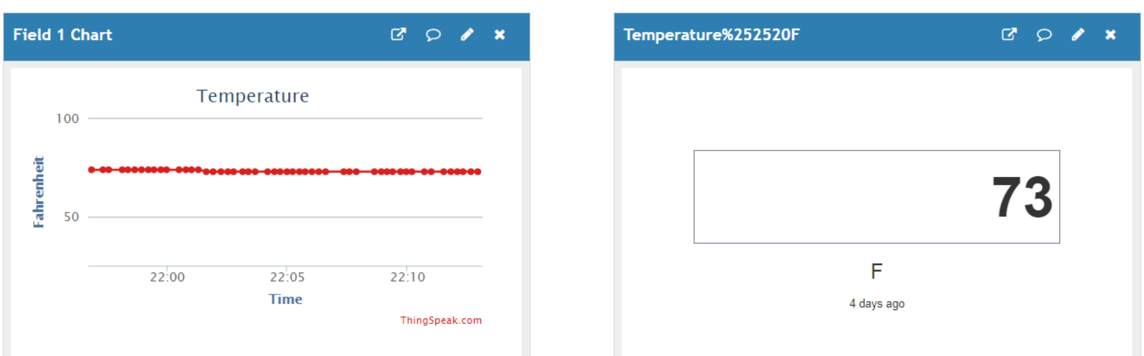


Figure 10: Image of channel sensor data for Temperature

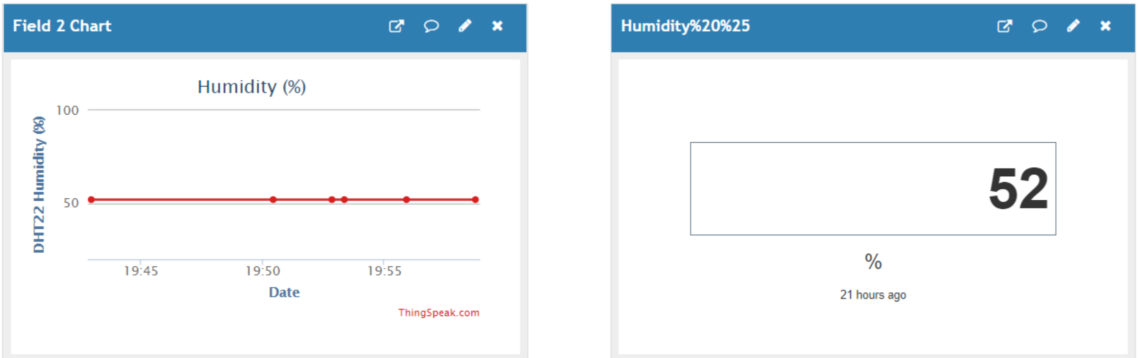


Figure 11: Image of channel sensor data for Humidity

ThingSpeak has contributed tremendously to the growth and rapid adoption of IoT. Students, Engineers, or scientists can now design and build IoT systems quickly without setting up servers or developing web software [53]. With it, IoT enthusiast can make the following critical actions of any IoT system:

Collect data from sensors, instruments, and websites and send them to the cloud for storage in either a private or public channel. Public channels allow the sharing of data stored on its channel.

Analyze and visualize data stored using online analytical tools to discover relationships, patterns or trends in a date. New data can be simulated or generated from the ThingSpeak platform. Acting on data or given results is an essential part of any IoT system. With ThingSpeak, an IoT system can send alerts to describe or prompt reactions to an event.

There are several other options similar to ThingSpeak available for public and commercial use. Still, in this project, we used ThingSpeak because it is straightforward to configure, send data from sensors using popular protocols and real-time visualization of this data. It is highly compatible with MATLAB; in fact, it uses all the powerful tools of MATLAB for data analysis.

CHAPTER FOUR

METHODS AND IMPLEMENTATION PROCEDURES

4.1 Emotion Detection

Emotion, as a general definition, can be described as the natural instinctive state of mine due to someone's mode, circumstance, or relationship to others. In other words, it is the reaction or actions that reflect the significance of a thing, occasion, or state of being. Over time humans have come a long way to learn or associate certain actions or reactions to a particular emotion. Recently machines are being trained through techniques such as convolutional neural networks to recognize, predict, or detect human emotions. This has been applied and continued to be applied in so many aspects of our daily lives. The lie detector machines are a good example. Most normal adult humans can express their emotions verbally and may often call for attention or try to adjust to the situation. However, infants and as well as some adults with certain disabilities, are not able to express their emotions verbally. They require attention or help from others; therefore, it is imperative to detect or predict these emotions as early as possible in order to give timely intervention. Infants express discomforts such as wet diapers, hunger, sleepiness, or sickness by crying, and mothers or caregivers pay a lot of attention to this signal. This is why, in this research, we focused on the four main emotional classes (Happiness, Crying, Sleeping, and Normal) that are useful in predicting or caring for infants. In recent years, several machine learning techniques have been identified to train and predict with high accuracy human emotions using facial recognition techniques. We used MATLAB in-build tools for training and recognition of emotions in this project. Figure 13 shows the type of images used to train the four basic emotions in this research.

In this research, we used the famous AlexNet architecture, which is easier to implement and has better effects than the traditional Support Vector Machines [54]. AlexNet was the first deep convolutional neural network to achieve very high accuracy in 2012 and has since been improved significantly. In general, it has an input database of $227x227x3$ RGB images and an output of 1000x1 probability vectors. A total of more than 60 million variables are trained or learned as the input data moves through the hidden layers to the output layer. Each of these layers performs operations, such that it can learn features that are specific to the data.

Figure 12: Types of Images used to train Amb-I

In the AlexNet architecture, the four most used layers in terms of functionality are Convolution, activation, fully connected (FC), and pooling layers. Figure 15 shows a general AlexNet architecture.

The convolutional layer has a set of convolutional filters that activates certain features from the input image. There are five convolutional layers in the AlexNet architecture. The activation function is usually a Rectified Linear Unit (ReLU), which allows only certain features to be carried to the next layer. The ReLU set all negative values to zero to allow much faster training. The ReLU function does not suffer from the vanishing gradient that activation functions such as the sigmoid and tanh often endure due to their minimal gradient at the saturation regions of the data. This vanishing gradient makes the network very difficult to train. A ReLU layer is placed after each convolutional layer.

The outputs are usually simplified by reducing the number of features the network needs to learn, and this processing is called pooling. Over-fitting, a major problem of previous neural network architecture, is reduced by using the dropout layer approach after each fully connected layer. It stochastically set a number of input or hidden neurons to zero, thereby reducing computation and co adaptations.

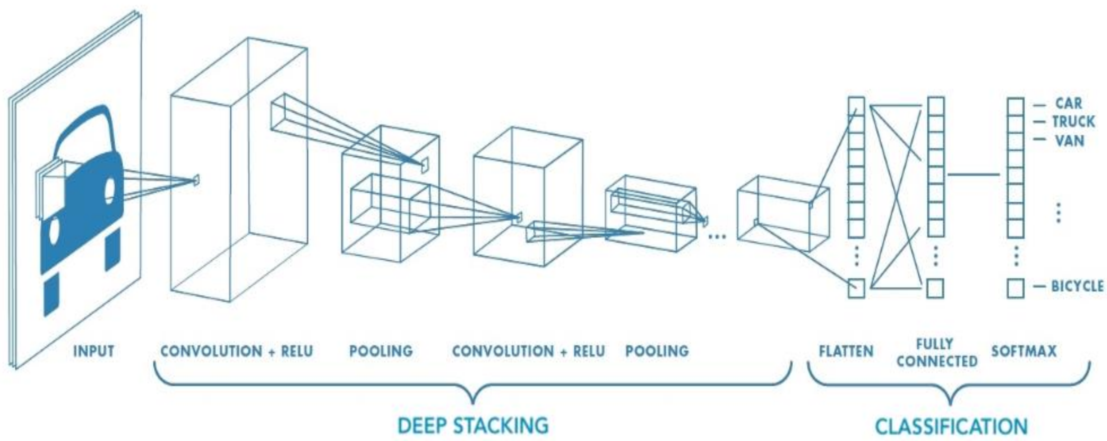


Figure 13: An AlexNet Architecture [55]

Other important layers include the classification layers, which include a vector of K classes to predict and a classification layer such as Softmax to produce the classification output.

The Emotion detection method, as shown in Figure 15., aims to customize and retrain a pre-trained convolutional neural network (AlexNet) to classify emotion on a new dataset (collection of images). The MATLAB deep learning toolbox already has an inbuilt AlexNet that has been trained with more than 1 million images. AlexNet has already learned several features of an image, such as edges, resolution, and contrasts, etc.

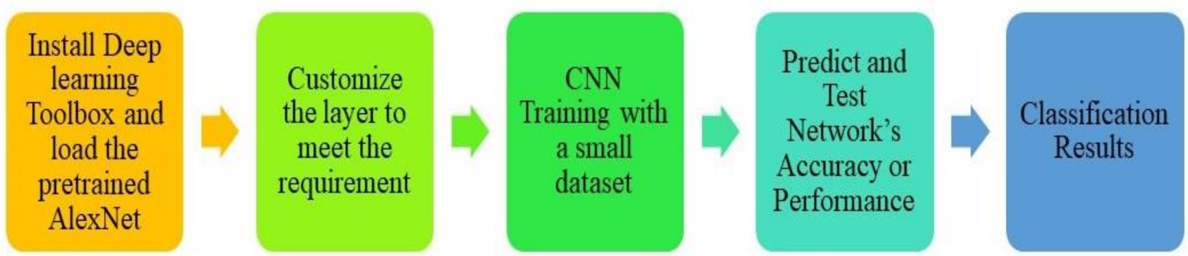


Figure 14: Emotion Detection Method

4.1.1 Dataset Creation

Using my laptop webcam and a small MATLAB code, we can create our desired database, whose number, size, format and other properties can be pre-determined. This will help to reduce or eliminate most of the preprocessing steps required during the retraining of the neural network.

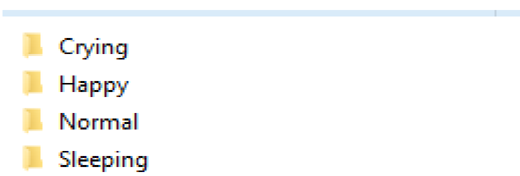
A database of 100 images of each of the class was created and labeled according to four basic expressions. A total dataset of 400 grayscale images was obtained for retraining the pretrained AlexNet using MATLAB. The final image sizes for training will be set to 227x227x1 at the input layer of the AlexNet. The four classes of the data set are *Crying, Happy, Normal, and Sleeping*.

100 Images of a Happy Face

100 images of a Crying face

100 images of a Sleeping face

100 images of a Normal face




```
allImages = imageDatastore('images', 'IncludeSubfolders', true,...
    'LabelSource', 'foldernames');
```

4.1.2 Training of the Neural Network

The images of the created database are loaded using *imageDatastore* function in MATLAB to automatically label the images based on folder names and store the data as an *ImageDatastore* object. The images are resized into the required input image size of 227x227x3 for colored mages or 227x227 for grayscale images.

As mentioned earlier, in this project, a well-studied pretrained convolutional network called AlexNet is selected. In addition, it is a very suitable candidate for low hardware quality and time constraint projects such as this. It has a depth of 8 compared to other CNN architecture out there, making training faster and easier.

The AlexNet architecture comprises five convolution layers, three fully connected layers, and a number of filters or kernels of various sizes for producing the feature maps [56]. The sizes of the filter vary in sizes and are located immediately after the convolution layers. Table 1 shows the layers of the customized AlexNet architecture.

Table 1: Customized AlexNet Architecture as seen in MATLAB

Layer	Layer Name	Layer Type	Layer Details
1	'data'	Image Input	227x227x3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Grouped Convolution	2 groups of 128 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	'pool2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv3'	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	'relu3'	ReLU	ReLU
12	'conv4'	Grouped Convolution	2 groups of 192 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
13	'relu4'	ReLU	ReLU
14	'conv5'	Grouped Convolution	2 groups of 128 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'drop6'	Dropout	50% dropout
20	'fc7'	Fully Connected	4096 fully connected layer
21	'relu7'	ReLU	ReLU
22	'drop7'	Dropout	50% dropout
23	''	Fully Connected	4 fully connected layer
24	'prob'	Softmax	softmax
25	''	Classification Output	crossentropyex

Load AlexNet

Before using the pretrained network, we must download the AlexNet support package in the deep learning toolbox of MATLAB.

Split Data

Our small dataset is further divided into training and validation data. We use 70% of the data for training the neural network and 30% for validation purposes.

```
%% Split data into training and test sets
[trainingImages, testImages] = splitEachLabel(allImages, 0.70, 'randomized');
```

Customize or Modify the Pre-trained network

Based on our classification requirements, such as the number of classes and classification type, we updated the 23 and 25 layers to suit 4 classes of emotions. We made layer 25 as our final classification layer. Table 2 shows the modified layers

Table 2: Modified Layers of AlexNeT to myNET

	AlexNet			myNET		
Layer	Layer Name	Layer Type	Layer Details	Layer Name	Layer Type	Layer Details
22	'drop7'	Dropout	50% dropout	'drop7'	Dropout	50% dropout
23	fc8	Fully Connected	1000 fully connected layer	''	Fully Connected	4 fully connected layer
24	'prob'	Softmax	softmax	'prob'	Softmax	softmax
25	Output	Classification Output	crossentropyex (Tench with 999 other classes)	Output	Classification Output	crossentropyex (Crying with 3 other classes)

```
%% Modify Pre-trained Network
layers(23) = fullyConnectedLayer(4); % change this based on # of classes
layers(25) = classificationLayer
maxEpochs= 50;
miniBatchSize = 128; % lower this if your GPU runs out of memory.
```

Training the Network

The network is now trained to learn the key features of an image, updating the weight correspondingly. We set the epoch (number of full training cycles on an entire dataset) to 50 and the learning rate to a very low figure (0.0001) to slow down the learning in the inherited layers.

```
%% Perform Transfer Learning
opts = trainingOptions('sgdm', ...
    'MiniBatchSize',128, ...
    'MaxEpochs',50, ...
    'InitialLearnRate',1e-4, ...
    'Shuffle','every-epoch', ...
    'ValidationData',testImages, ...
    'ValidationFrequency',3, ...
    'Verbose',false, ...
    'Plots','training-progress');
myNet = trainNetwork(trainingImages, layers, opts);

save myNet myNet
```

After all MATLAB configurations, the training of the CNN is started by a simple press of a handle button created on the *guidemo* figure. This figure is a customized version created using the legacy Graphic user interface development Environment (GUIDE) of MATLAB. Figure 16 shows the MATLAB interface showing the CNN handle/button needed to start the training process.

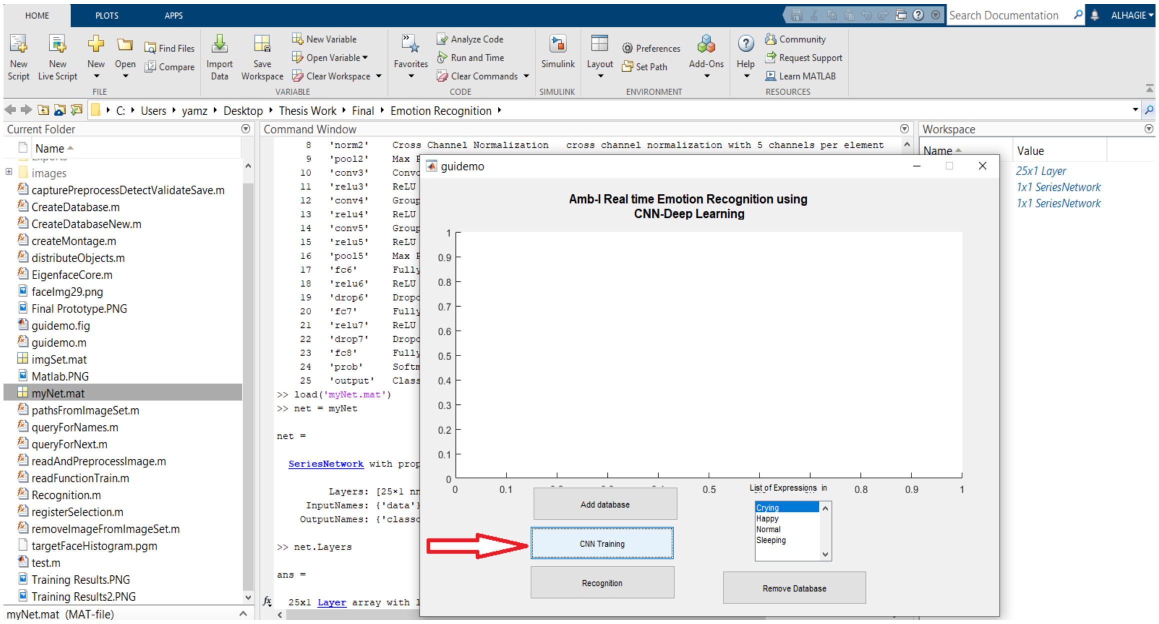


Figure 15: MATLAB Graphical User Interface with the CNN handle/button

4.1.3 Emotion Detection/Recognition

The part too, like the training part is programmed in MATLAB and linked to a hand button in the guide graphic user interface. Figure 17 shows the MATLAB Graphical User Interface showing the CNN handle/button that needs to be used after loading the image for the emotion recognition.

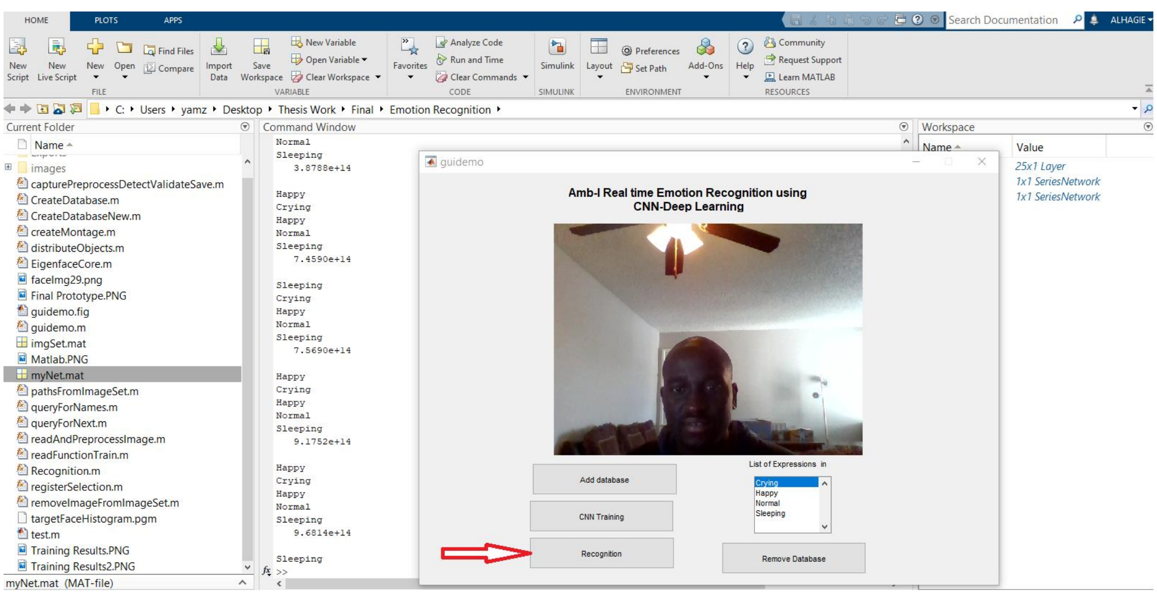


Figure 16: MATLAB Interface with Recognition

The overall detection and recognition phase of this project is broken down into three processes:

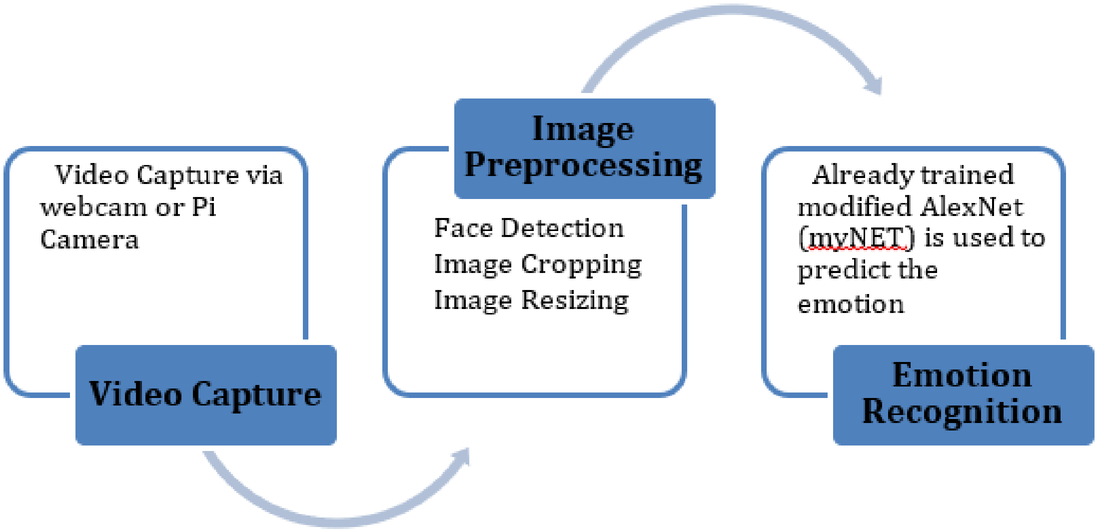


Figure 17: Emotion Recognition process

Video Capture

This is the image acquisition phase of the monitoring system. Video capture is done via a webcam and requires a MATLAB support package for Webcams.

```
   ud.adaptor_name = 'winvideo';
   global framesize
% Check & Init Web-Cam
cam = imaqhwinfo();
nr_adaptors = length(cam.InstalledAdaptors);
linind = 0;
for i=1:nr_adaptors
    if strcmp(cam.InstalledAdaptors{i}, ud.adaptor_name)
        linind = i;
        break;
    end
end
```

Image Preprocessing

The images acquired from the video are preprocessed before being loaded into the trained neural network for emotion detection. For an AlexNet neural network, the input size to the neural network is 227 by 227 for grayscale.

To make the training more accessible to people of different color and race, all the colored images are converted to grayscale before being loaded into the neural network.

Face Detection

To detect the emotions from any video data or image, additional standardization and image processing were performed. Regardless of the image background, size of images or objects in the images, the face which is a key is identified, cropped out, and magnified if necessary and used to detect emotional state associated with the image. The Viola-Jones algorithm [58] was used to detect our interest region (the face) before the emotion detection process is applied. During training, the algorithm ensures that only the right images are used in the training and validation dataset.

Emotion Detection

Once the image is acquired, preprocessed, and the face region detected, this cropped region is projected into the face space. The PCA features are extracted from the image and compared with the feature learned during training. The Euclidean distance between the projected test image and the projections of all training images is calculated. The minimum distance corresponds to the emotion of the image in the training database.

```
load('myNet.mat');
```

4.1.4 Communication with the Single Board Computer

A UDP communication is established between the MATLAB and Raspberry Pi, so that emotion out of the neural network is forwarded to the raspberry Pi for further processing.

```
echoudp('on', 5555)
u = udp('10.11.65.117', 5555);
fopen(u);
```

To transmit data on the communication channel, I decided to translate the image output into a single digit number which is small in size and easily transmitted.

```
label = char(classify(myNet,im)); % classify with deep learning
OutputName=char(label);
        disp(OutputName);
        switch OutputName
           case 'Happy'
               fwrite(u,'0');
           case 'Crying'
               fwrite(u,'1');
           case 'Sleeping'
               fwrite(u,'2');
           case 'Normal'
               fwrite(u,'3');
           otherwise
        end
```

4.2 Ambient Monitoring

The ambient conditions surrounding an infant directly affects their wellbeing and emotion. Infants cry when the surrounding temperatures are too hot or cold. Wet diapers also cause much distress to a lot of young infants and are usually expressed through cries, inability to sleep or restlessness.

In this research, the focus was to monitor the effect of ambient temperature in baby's emotion. To implement the monitoring system, the single board computer, Raspberry Pi 3 B+ was used along with the DHT22 sensor. Raspberry Pi 3 B+ is a low-cost device with low power requirements (2W) and high computing power or performance [59]. It has a Broadcom BCM2837BO, Cortex-A53 (ARMv7) 64-bit quad-core processor capable of running at up to 1.4GHz. It has a 2.4GHz/5GHz wireless LAN, fast Ethernet, and Bluetooth for communication. The temperature and humidity data from the DHT22 sensor is fed into ThingSpeak for future analysis and sent to the Raspberry Pi for event notification and other intelligent processing.

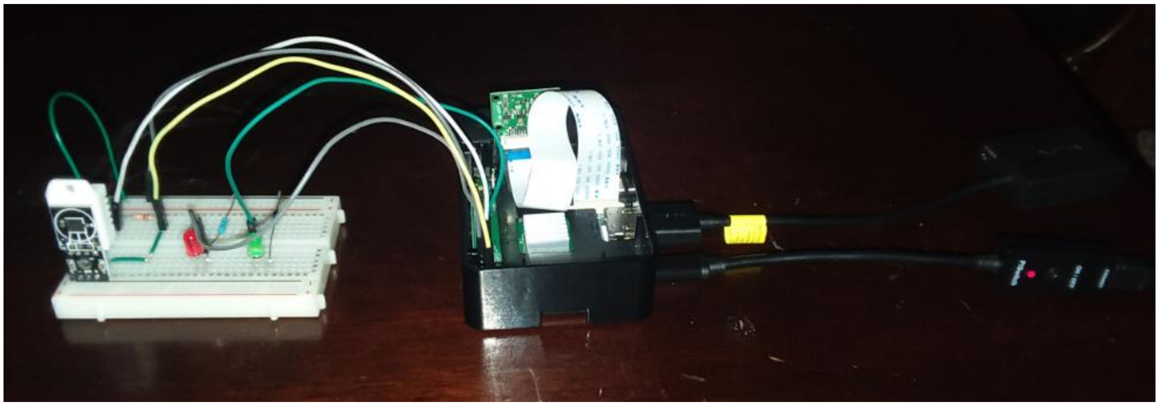


Figure 18: Amb-I system showing sensors and SBC (Raspberry PI)

The applications of our Ambient Monitoring can be described but not limited to the following usage:

The DHT22 sensor can be used to collect temperature conditions in the environment and send it to the microcontroller. Depending on the preset condition (Temperature >80-degree Fahrenheit), it is considered a hot environment; the microcontroller will send a trigger to adjust an attached Thermostat to set the air conditioner on or turn the heater off. When the temperature is low, say below 60 degrees Fahrenheit, the settings are adjusted to readjust the temperature by turning the air conditioner off or turning on the heating system. All these can be accompanied by email notification to the caregiver, stating previous temperature conditions, action taken to regulate it and current temperature in real-time.

4.3 Internet of Things Framework

The use of internet of things framework in this research is 2 folds:

4.3.1 Data Upload, detection, and feedback on ThingSpeak

The ThingSpeak helps in logging all the data from the sensors and image processing unit into the cloud. The data can later be collected and studied for a correlation between emotional events and sensor data. This data can also be accessed by parents who want to look at historical data.

4.3.2 Reactions to emotional or sensor data by ThingSpeak

The Second Part includes the microcontroller, which also receives data from the image processing unit as well as the sensor data. It acts based on the following:

If the emotion data received is Happy and the corresponding sensor data is normal at that time, then no notification messages are forward or no changes in the triggered as in the flow chart below.

4.4 Amb-I System Prototype

In the final prototype, all the three components, communications, data sharing, and analysis, are linked together to develop the Amb-I system.

The emotion detected by the system is converted to single-bit data and sent over the UDP communication like between the MATLAB and the Raspberry Pi. For example, when *crying* is detected, number 2 is sent to the Raspberry Pi which then checks for the ambient conditions surrounding the infant by querying the Humidity and Temperature database. If any of this is above or below a set threshold, a corresponding alarm in the form of a blinking LED and/or send notification (Email) to a preset email address. Table 3 shows the image to data conversion to enable cheaper and faster communication. Figure 20 shows the flowchart of integration of IoT and Affective computing in the proposed Amb-I system.

Table 3: Classifying Images based on emotions

Emotion	**Data Value**
Happy	0
Crying	1
Sleeping	2
Normal	3

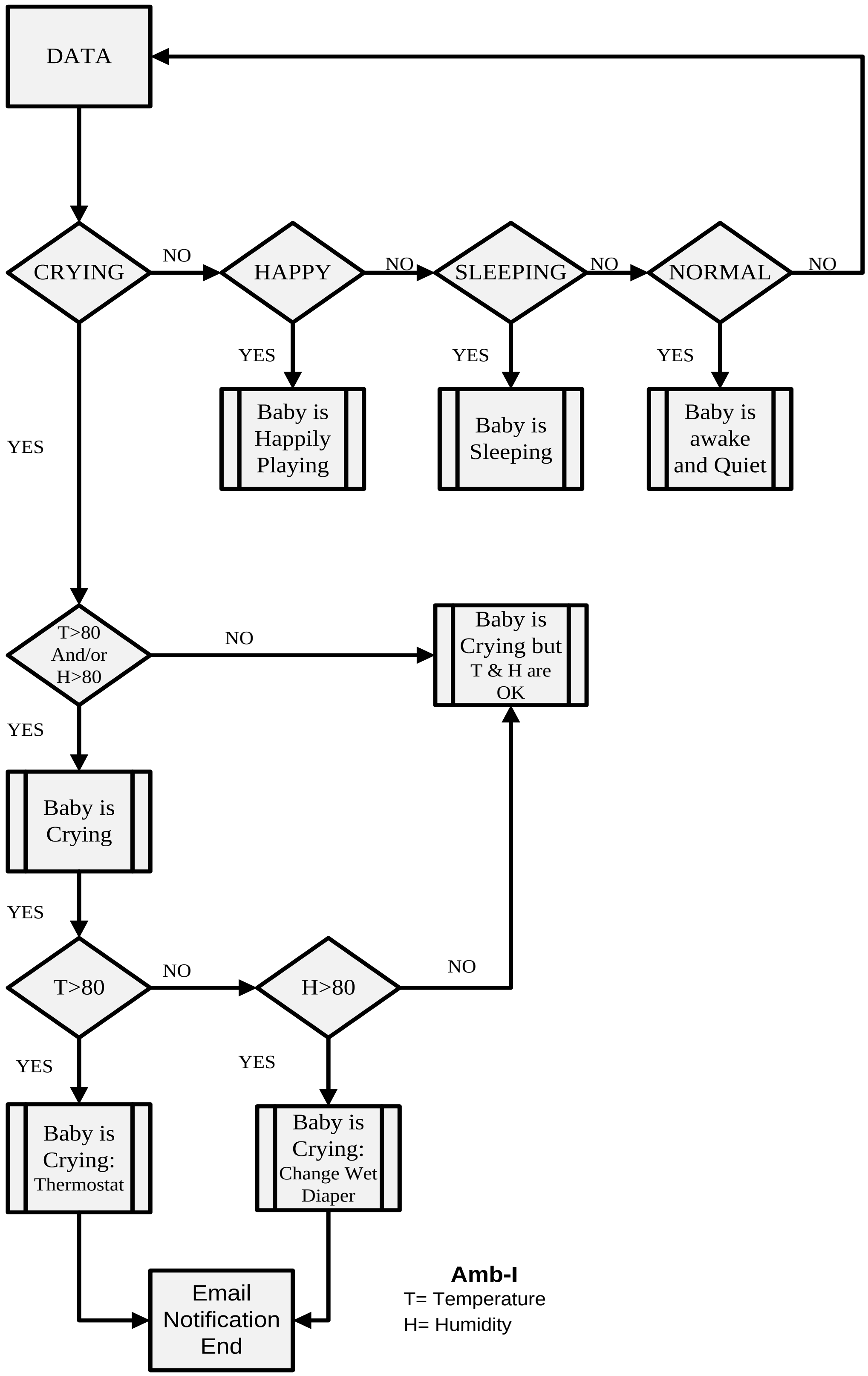


Figure 19: Flowchart of the proposed Amb-I system

CHAPTER FIVE

DISCUSSION AND CONCLUSION

5.1 Discussion

In order to have a quick but rapid classification training to detect or identify emotions, the concept of *Transfer Learning* was used. This concept was applied to the already trained AlexNet with very little data (400 Images) to get the desired or meaningful outcome. The pretrained AlexNet already had 1000 classes at its output, but in this project, this was modified to only four classes: *Crying, Happy, Normal, and Sleeping*. For training and validation, 100 images were used, and obtained a good result with almost 100% training accuracy and 95% validation accuracy.

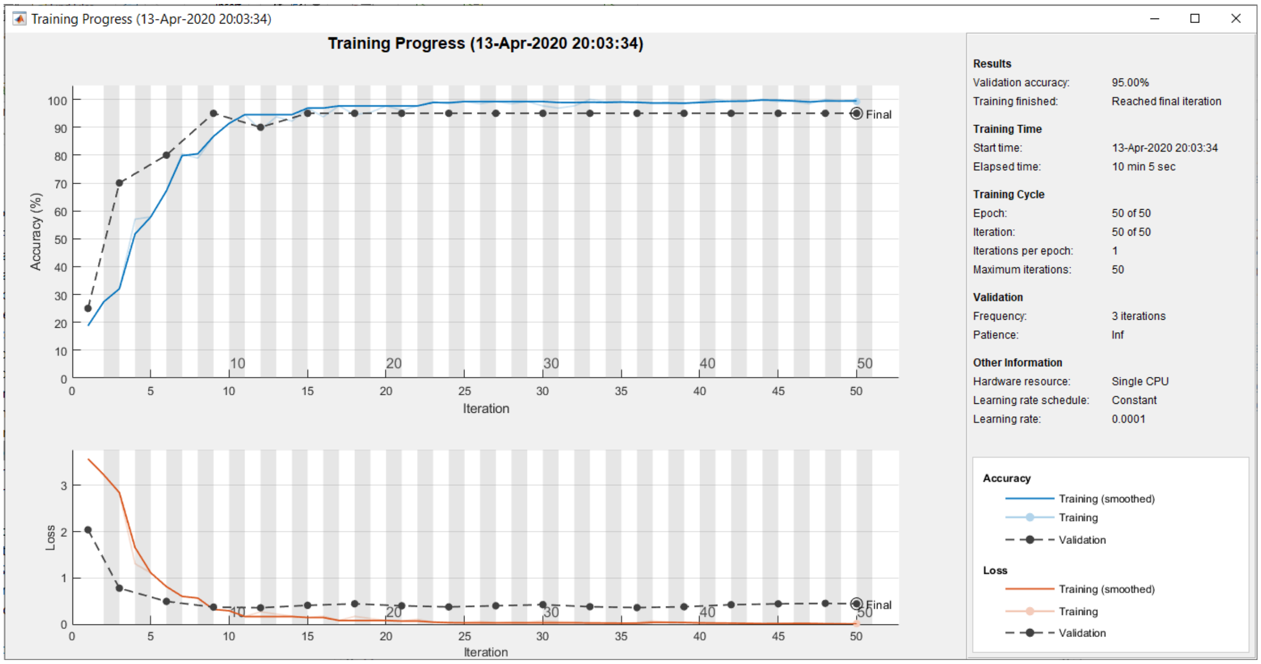


Figure 20: Graph showing training accuracy and loss progress of my customized CNN

Since the hardware resource available for training is just a single CPU (i5 @ 1.95GHz 2.50GHz and 8GB RAM), the following were selected to minimize the time required for training but also not compromise accuracy:

- Maximum of 50 Iterations.
- 60 Iteration per epoch.
- A constant learning rate of 0.0001.

The small learning rate of 0.0001 was chosen to pace out the training, this will help the newly added layers to adjust sufficiently and adapt with the already trained layers in the original pretrained network.

As shown in the figure above, the training started with a low accuracy of around 18% in the first few iterations but progressed almost linearly up to the 10th iteration when it reaches almost 95%. By the 25th iteration, it has already started hitting the 100% accuracy. Such rapid and accurate results would not have been possible without the use of a pretrained network.

The loss ratio shown above reflects the same. It started around 3% but steeped negatively towards zero by the 25th iteration.

Following successful training, CNN is deployed as part of the Amb-i project for video monitoring. When the video is started, frames are obtained from it for analysis at a rate of 20 frames per sec. These are then individually analyzed and forward to the raspberry pi for further processing.

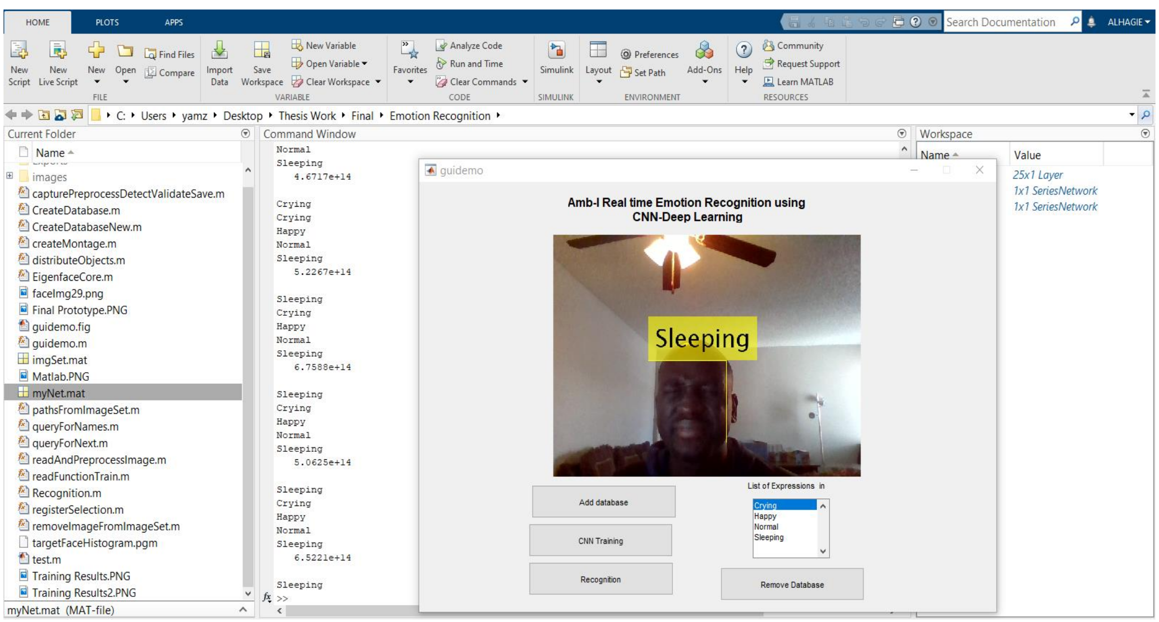


Figure 21: MATLAB Interface showing the Emotion Recognition Process (Sleeping)

At the same time, the sensors are uploading sensor data to the ThingSpeak cloud server and the Raspberry Pi, so that in the event of a change in emotion, temperature or Humidity, the raspberry-Pi is prompted to act on this data based on the set criteria.

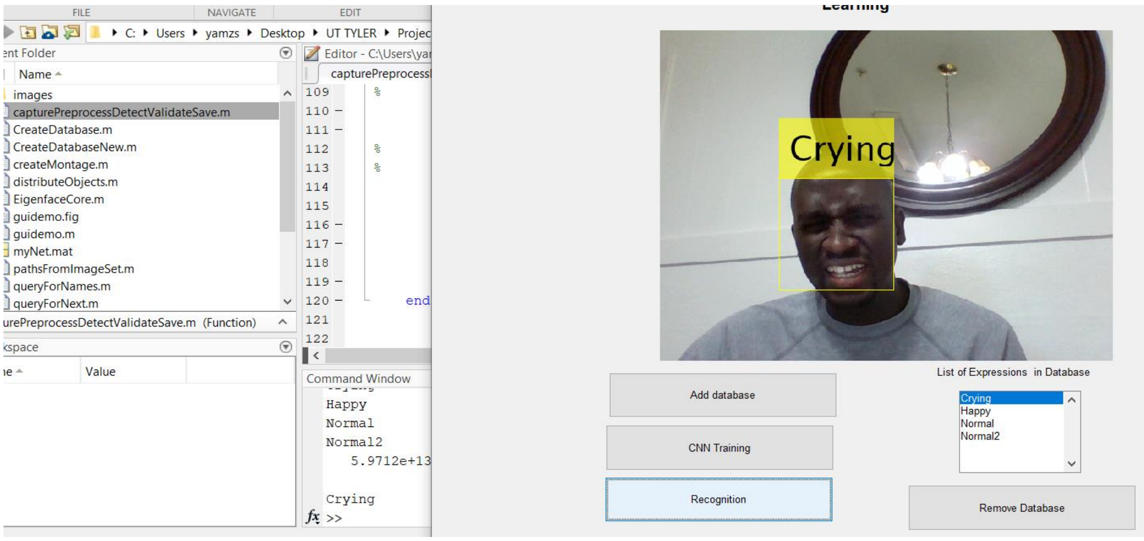


Figure 22:MATLAB Interface showing the Emotion Recognition Process (Crying)

5.2 Validation

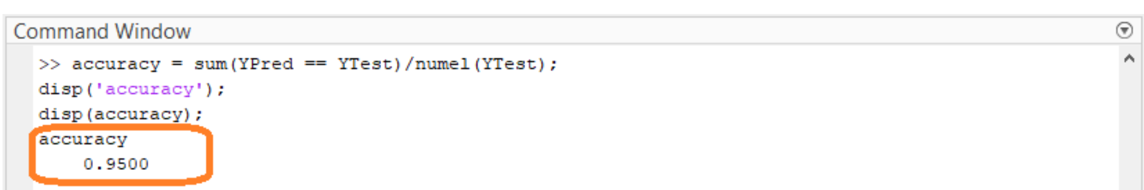
The model is validated using the following two methods: Validation using 30% of dataset, validation using Youtube videos showing emotions,

5.2.1 Validation using 30% of my dataset

During training, we set aside 30% of the data testing and used that as a first step towards validating our model. Figure 21 shows that validation results of 95% accuracy immediately after the training.

```
%Validation Test
YPred = classify(myNet, testImages);
YTest = testImages.Labels;

%Classification accuracy on the validation set
accuracy = sum(YPred == YTest)/numel(YTest);
disp('accuracy');
disp(accuracy);
```



We ran a simple code to test 9 test images and display them with their corresponding labels as shown below.

Figure 23: Display of nine sample validation images with their predicted labels

5.2.2 Validation using YouTube videos of emotional (Crying) infants

Since it was hard to get enough infant data for training and validation, we also used some sample videos and loaded them to our model for validation. However, due to insufficient training data to account for special features on infants' faces such as tears, movements, partial faces, or sizes, the validation was not as good as the previous validation process. A validation score of 17% was recorded when tested on our model. Figures 25 through 28 show the process of loading the videos, validation, and analyzing the accuracy.

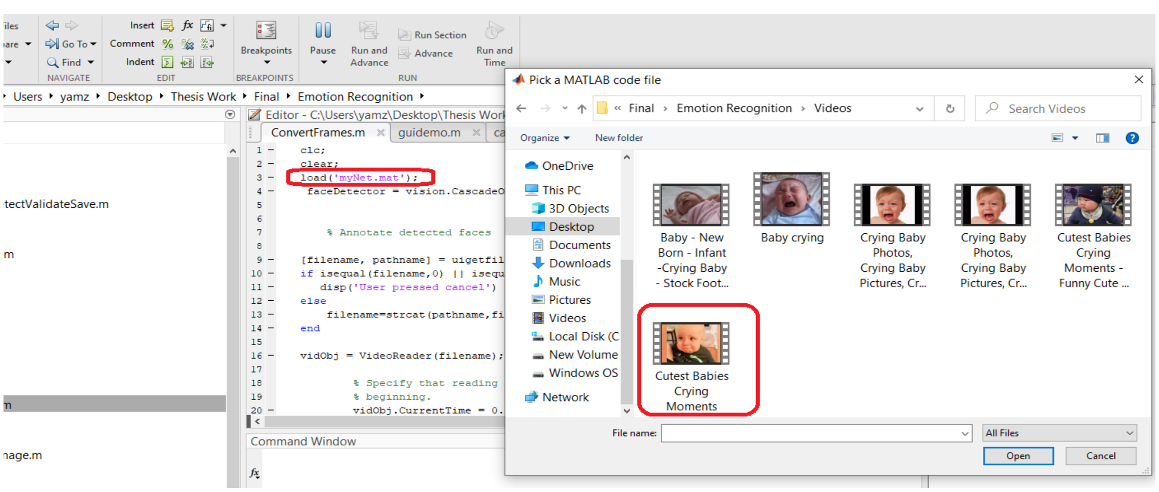


Figure 24: Loading Baby's video for validating emotion recognition

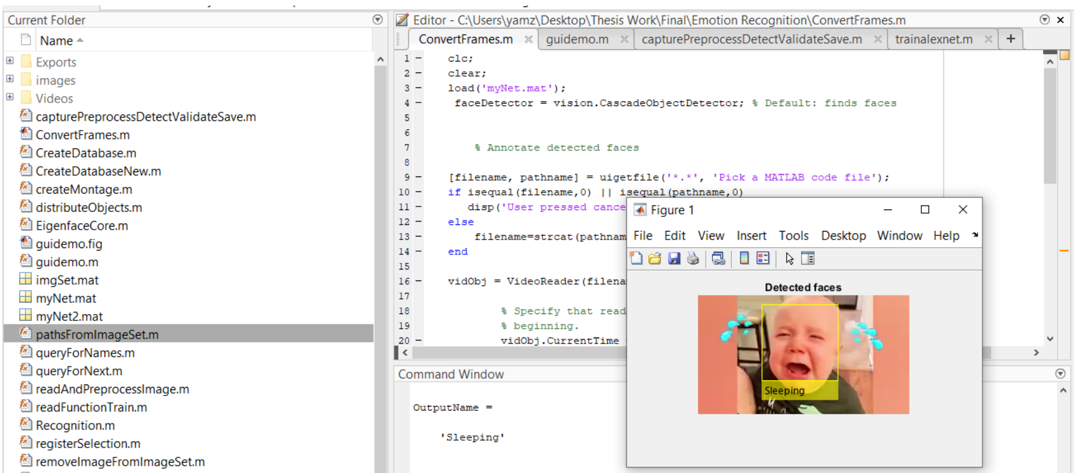


Figure 25: Video loaded being tested in our model for validation

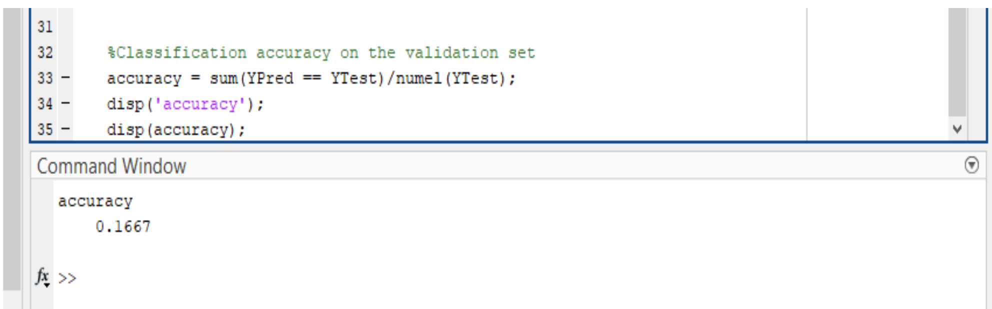


Figure 26: Command window showing accuracy of data

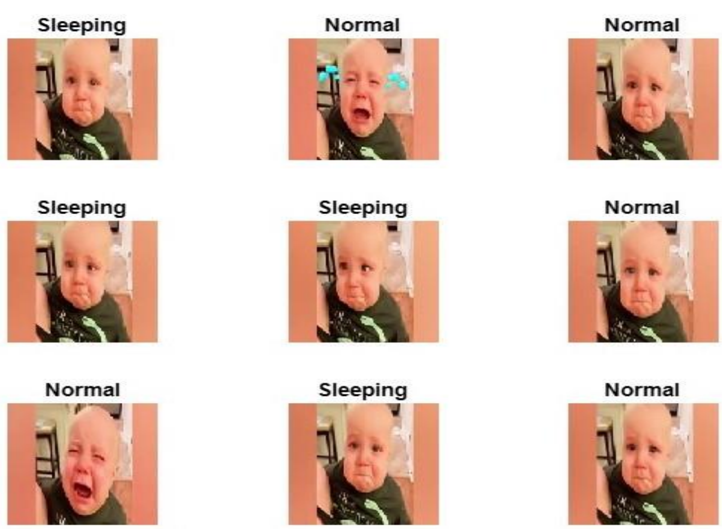


Figure 27: Display of nine sample validation images with their predicted labels from frames extracted from the YouTube video

Table 4: Accuracy of results of validation tests conducted

NO	DATA SOURCE /DATABASE	VALIDATION ACCURACY
1	My Training Dataset	95%
2	https://youtu.be/XlNriVfRBTs	17%
3	https://youtu.be/8RTyyJyHg0w	14%
4	https://youtu.be/MzwAsUlkpjA	20%
5	https://youtu.be/qS7nqwGt4-I	27%
6	https://youtu.be/l0NynUuTPRU	19%

5.3 Conclusion and Future Recommendation

A real-time emotion detection system was built for infant monitoring systems using transfer learning concepts of deep learning. A small database of 400 images was enough to get the desired results. The training time was low, even though it was done on a low CPU processing laptop. The cost involved in both training and implementation was extremely low. The AlexNet was easily customized to recognize the four classes of interest in our project. Realtime time monitoring of the classes of emotion was also achieved using a connected webcam.

However, due to time constraints and limited resources such as low hardware functionality, other convolutional neural network architecture such as GoogleNet, SqueezeNet or DenseNet were not implemented. For further enhancing the accuracy of the proposed system, it is important to train the network with more images in order to validate them.

With the proper sensors, body temperatures can also be measured using attached sensors to monitor the fever and other health conditions that are related to body temperature. With this, parents or caregivers are updated with the conditions of their infants in real-time. The humidity sensor part of the DHT22 sensor will be used to monitor the dampness of the infant's environment. This information is fed to the microcontroller that is connected to a humidifier for autoregulation.

To measure the diaper conditions, a similar humidity sensor can be attached to the diapers to send the moisture condition of the diaper. This same data can be fed into the microcontroller and ThingSpeak. The microcontroller is then able to alert the caregiver or parent in a timely manner.

Monitoring using cameras has some drawbacks, the frequent movement of infants away from the camera, right adjustment of camera settings or position, the posture of infants as well as lighting conditions of the environment should be considered in future studies to further enhance the performance of the system. Security measures should be further studied to ensure the safety and protection of the infant and data collected.

References

1. https://www.bls.gov/news.release/famee.nr0.html Accessed 02/10/2020.
2. S. D. Wipulasundara, "A Novel Concept For Remotely Monitoring Babies," 2017 Moratuwa Engineering Research Conference (MERCon), Moratuwa, 2017, pp. 299-303.
3. M. Turna, Y. Li and Y. Kim, "Estimation of Diaper Condition Based on Antenna Impedance Variation Through Rational Function Approximation," in IEEE Access, vol. 8, pp. 2441-2446, 2020.
4. G. Joshi, C. Dandvate, H. Tiwari and A. Mundhare, "Prediction of Probability of Crying of a Child and System Formation for Cry Detection and Financial Viability of the System," 2017 International Conference on Vision, Image and Signal Processing, IEEE, 2017, pp. 134-141. DOI: 10.1109/ICVISP.2017.33
5. A. Osmani, M. Hamidi, H. and A. Chibani, "Machine Learning Approach for Infant Cry Interpretation," 2017 International Conference on Tools with Artificial Intelligence, IEEE, 2017, pp. 182-186. DOI: 10.1109/ICTAI.2017.00038
6. C. Y. Chang and F. Chen, "Application of Deep Learning for Infant Vomiting and Crying Detection," 2018 32nd International Conference on Advanced Information Networking and Applications Workshops *(WAINA)*, Krakow, 2018, pp. 633-635, doi: 10.1109/WAINA.2018.00158
7. M. P. Joshi and D. C. Mehetre, "IoT Based Smart Cradle System with an Android App for Baby Monitoring," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, 2017, pp. 1-4, doi: 10.1109/ICCUBEA.2017.8463676
8. A. M. Ghosh, D. Halder and S K Alamgir Hossain, "Remote Health Monitoring System Through IoT," 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, 2016, pp. 921-926, doi: 10.1109/ICIEV.2016.7760135
9. R. Kumar and M. P. Rajasekaran, "An IoT based patient monitoring system using Raspberry Pi," 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16) , Kovilpatti, 2016, pp. 1-4, doi: 10.1109/ICCTIDE.2016.7725378

10. X. Liu, K. Tekeuchi, T. Ogunfunmi and S. Mathapathi, "Video-based IoT Baby Monitor for SIDS Prevention," 2017 IEEE Global Humanitarian Technology Conference (GHTC), IEEE, 2018, DOI: 10.1109/GHTC.2017.8239258
11. A. Archip, N. Botezatu, E. Şerban, P. C. Herghelegiu and A. Zală, "An IoT Based System for Remote Patient Monitoring," 2016 17th International Carpathian Control Conference (ICCC), pp. 1-6, 2016.
12. S. P. Patil and M. R. Mhetre, "Intelligent Baby Monitoring System," 2014 ITSI Transactions on Electrical and Electronics Engineering (ITSI- TEEE), vol. 2, no. 1, 2014.
13. M. Leier and G. Jervan, "Miniaturized Wireless Monitor for Long-Term Monitoring Of Newborns," 2014, 14th Biennial Baltic Electronic Conference (BEC), pp. 193-196, 2014.
14. M. Asaduzzaman, M. Mir, H. Kabir, M. Siddiqur, R. Tanveer and M. A. H. Akhand, "Continuous Heart Rate and Body Temperature Monitoring System Using Arduino UNO and Android Device," 2015 2nd International Conference on Electrical Information and Communication Technologies (EICT), pp. 183-188, 2015.
15. N. Indumathy and K. K. Patil, "Medical Alert System for Remote Health Monitoring Using Sensors and Cloud Computing," International Journal of Research in Engineering and Technology, vol. 3, no. 04, pp. 884-888, 2014.
16. B. Priya, S. Rajendran, R. Bala, and R. Gobbi, "Remote Wireless Health Monitoring Systems," *2009 Innovative Technologies in Intelligent Systems and Industrial Applications*, Monash, 2009, pp. 383-388, doi: 10.1109/CITISIA.2009.5224177
17. Y. Hata, H. Yamaguchi, S. Kobashi, K. Taniguchi and H. Nakajima, "A Human Health Monitoring System of Systems in Bed," System of Systems Engineering, 2008. SoSE '08. IEEE International Conference on 2-4 June 2008, pp 1 - 6.
18. M. Nambu, K. Nakajima, A. Kawarada and T. Tamura, "The Automatic Health Monitoring System for Home Health Care," Information Technology Applications in Biomedicine, 2000. Proceedings. 2000 IEEE EMBS International Conference on 9-10 Nov. 2000 pp 79 - 82.
19. L. Heyoug, K.Yong-Tae, J.Jin-Woo, P. Kwang-Hyun, K. Dae-Jin, B. Bang and B. Zeung Nam, "A 24-Hour Health Monitoring System in a Smart House," International Society of Gerontechnology, 2008.ISSN 1569-1101, Gerontechnology Journal, v.7, no.1, pp.22-53.

20. G. Fang, and J. Yong Duan, “Design and Implementation of Intelligent Baby Room Monitoring System Based on the Internet of Things," Applied Mechanics and Materials 513–517 (February 2014): 2484–86.
21. S. Brangui, M. El Kihal and Y. Salih-Alj, "An Ergonomic Comprehensive System for Monitoring and Control of Baby Environments," 2015 6th International Conference on Information and Communication Systems (ICICS), Amman, 2015, pp. 251-256.DOI: 10.1109/IACS.2015.7103184
22. N. Zakaria, F. Saleh and M. Abdul Razak, "IoT (Internet of Things) Based Infant Body Temperature Monitoring," 2018 2nd International Conference on BioSignal Analysis, Processing and Systems (ICBAPS), IEEE, 2018, pp. 148-153. DOI: 10.1109/ICBAPS.2018.8527408
23. B. Ashish, "Temperature Monitored IoT based Smart Incubator," 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), IEEE, 2017, pp. 497-501. DOI: 10.1109/I-SMAC.2017.8058400
24. E. Saadatian et al., "Low-Cost Infant Monitoring and Communication System," 2011 IEEE Colloquium on Humanities Science and Engineering Penang, pp. 503-508, 2011.
25. B. Mohammad, H. Elgabra, R. Ashour and H. Saleh, "Portable Wireless Biomedical Temperature Monitoring System: Architecture and Implementation," 2013 9th International Conference on Innovations in Information Technology (IIT), pp. 95-100, 2013.
26. S. S. Thomas, A. Saraswat, A. Shashwat and V. Bharti, "Sensing Heart Beat and Body Temperature Digitally using Arduino," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, 2016, pp. 1721-1724, doi: 10.1109/SCOPES.2016.7955737
27. M. S. Mothi and U. S. Banu, "Novel Continuous Infant Temperature Alerting System," 2018 9th IEEE Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 2018, pp. 35-38.
28. F. Aktaş, E. Kavuş, and Y. Kavuş, "Real-Time Infant Health Monitoring System for Hard of Hearing Parents," 2016 Medical Technologies National Congress (TIPTEKNO), Antalya, 2016, pp. 1-4. DOI: 10.1109/TIPTEKNO.2016.7863098

29. P. D. M., S. N. Siva, and M. Sudhan, "Noninvasive Health Monitoring System for Infants using IoT," 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), CHENNAI, India, 2019, pp. 1-3.
30. A. A. Joseph, T. H. L. Xuan, K. Kipli, K. L. Chin and N. S. Song, "Smart Body Temperature Monitoring System for Children via Mobile," 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), Kuching, 2018, pp. 1-6.
31. J. Siden, A. Koptioug and M. Gulliksson, "The Smart Diaper Moisture Detection System," Microwave Symposium Digest, June 2004, pp. 659-662.
32. M. S. Turna and Y. Kim, "Monitoring Diaper Condition Using the Impedance Variation of a Dipole Antenna," 2018 International Applied Computational Electromagnetics Society Symposium - China (ACES), Beijing, China, 2018, pp. 1-2, doi: 10.23919/ACESS.2018.8669275
33. Tareq Khan, "A Smart Wearable Gadget for Noninvasive Detection and Notification of Diaper Moiter," 2018 International Conference on Electro/Information Technology (EIT), IEEE, 2018. pp. 240-244 DOI: 10.1109/EIT.2018.8500233
34. P. Sen, S. N. R. Kantareddy, R. Bhattacharyya, S. E. Sarma, and J. E. Siegel, "Low-Cost Diaper Wetness Detection Using Hydrogel-Based RFID Tags," in IEEE Sensors Journal, vol. 20, no. 6, pp. 3293-3302, 15 March15, 2020, doi: 10.1109/JSEN.2019.2954746
35. M. Y. E. Simik, F. Chi, A. M. S. Abdelgader and R. S. I. Saleh, "Automated Alarm System for Diaper Wet Using GSM," 2014 IEEE 17th International Conference on Computational Science and Engineering, Chengdu, 2014, pp. 1799-1803, doi: 10.1109/CSE.2014.330
36. M. McKnight, F. Lin, H. Kausche, T. Ghosh and A. Bozkurt, "Towards Paper-Based Diaper Sensors," 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), Atlanta, GA, 2015, pp. 1-4, doi: 10.1109/BioCAS.2015.7348427
37. M. Wen, "Goo9: A Wireless Sensor Network System for Wet Diaper Detection," 2017 International Conference on Applied System Innovation (ICASI), Sapporo, 2017, pp. 893-895, doi: 10.1109/ICASI.2017.7988580
38. Q. Zhang, T. Guo, A. Bao, "Design Of Real-Time Temperature and Humidity Measurement System based on SHT21 Sensor," *2010 International Conference on*

Computer Application and System Modeling (ICCASM 2010), Taiyuan, 2010, pp. V14-32-V14-35, doi: 10.1109/ICCASM.2010.5622428

39. J. Mukhopadhyay et al., "An Evaluation of Human Perception for Neonatal Cry Using A Database of Cry and Underlying Cause," 2013 Indian Conference on Medical Informatics and Telemedicine (ICMIT), Kharagpur, 2013, pp. 64-67 , doi: 10.1109/IndianCMIT.2013.6529410
40. A. Osmani, M. Hamidi, and A. Chibani, "Machine Learning Approach for Infant Cry Interpretation," 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, 2017, pp. 182-186, doi: 10.1109/ICTAI.2017.00038
41. K. Candra Kirana, S. Wibawanto and H. Wahyu Herwanto, "Facial Emotion Recognition Based on Viola-Jones Algorithm in the Learning Environment," 2018 International Seminar on Application for Technology of Information and Communication, Semarang, 2018, pp. 406-410, doi: 10.1109/ISEMANTIC.2018.8549735
42. B. T. Nguyen, M. H. Trinh, T. V. Phan, and H. D. Nguyen, "An Efficient Real-Time Emotion Detection using Camera and Facial Landmarks," 2017 Seventh International Conference on Information Science and Technology (ICIST), Da Nang, 2017, pp. 251-255, doi: 10.1109/ICIST.2017.7926765.
43. Suchitra, P. Suja, and S. Tripathi, "Real-Time Emotion Recognition from Facial Images using Raspberry Pi II," 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, 2016, pp. 666-670, doi: 10.1109/SPIN.2016.7566780
44. https://www.mathworks.com/products/matlab.html Accessed 02/26/2020.
45. X. Han, Y. Zhong, L. Cao, and L. Zhang, "Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification," 2017 Remote Sens. 9, 848. https://doi.org/10.3390/rs9080848.
46. https://ujwlkarn.files.wordpress.com/2016/08/screen-shot-2016-08-08-at-11-53-41-am.png?w=748 Accessed 03/06/2020.
47. A. A. Almisreb, N. Jamil and N. M. Din, "Utilizing AlexNet Deep Transfer Learning for Ear Recognition," 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), Kota Kinabalu, 2018 IEEE, pp. 1-5, doi: 10.1109/INFRKM.2018.8464769

48. M. Radulovic, M. Pavlovic, N. Nenadic and G. Dimic, "Industrial Single Board Computer based on OMAP5 Processor," 2013 2nd Mediterranean Conference on Embedded Computing (MECO), Budva, 2013, pp. 84-87, doi: 10.1109/MECO.2013.6601324
49. S. Misbahuddin, M. M. Ibrahim, A. M. Alnajar, B. Q. Alolabi, and A. F. Ammar, "Automatic Patients' Vital Sign Monitoring by Single Board Computer (SBC) Based MPI Cluster," 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2019, pp. 1-5, doi: 10.1109/CAIS.2019.8769551
50. https://learn.adafruit.com/dht Accessed 02/27/2020.
51. P.V Dudhe, N.V. Kadam, R.M. Huhangabade and M.S. Deshmukh, "Internet of Things (IoT): An Overview and its Applications," International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017).
52. A. Gipsa, V. Benitta, J.G. Jezna and A.M Abraham, "A Modern Health Care System Using IoT and Android," International Journal for Scientific Research and Development, 2, 2016, pp 98-101
53. https://www.mathworks.com/help/thingspeak/ Accessed 02/04/2020 at 5.38 pm
54. X. Chen, X. Yang, M. Wang and J. Zou, "Convolution Neural Network for Automatic Facial Expression Recognition," 2017 International Conference on Applied System Innovation (ICASI), Sapporo, 2017, pp. 814-817, doi: 10.1109/ICASI.2017.7988558
55. https://www.doc.ic.ac.uk/~jce317/introduction-cnns.html Accessed 03/14/2020 at 2.30 am
56. A. Akula and H. K. Sardana, "Deep CNN-based Feature Extractor for Target Recognition in Thermal Images," TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), Kochi, India, 2019, pp. 2370-2375.
57. P. Viola and M.J. Jones, "Robust Real-Time Face Detection," 2004 International Journal of Computer Vision vol 57 no 2) p 137-154. https://doi.org/10.1023/B:VISI.0000013087.49260.fb
58. M. B. S. Divya and N. B. Prajwala, "Facial Expression Recognition by Calculating Euclidean Distance for Eigen Faces Using PCA," 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2018, pp. 0244-0248, doi: 10.1109/ICCSP.2018.8524332

59. B. Dolwithayakul, P. Boonnasa, S. Klomchit and S. Tuwachaosuan, "Green Public Computer Lab using Single-Board Computer and Interactive Computer Reservation System," 2015 International Computer Science and Engineering Conference (ICSEC), Chiang Mai, 2015, pp. 1-4, doi: 10.1109/ICSEC.2015.7401420

APPENDIX

MATLAB and Python were the programming languages used in this project. The following MATLAB toolboxes, libraries, and function were used:

1. **Deep Learning Toolbox:** provides a platform for designing and implementing deep neural networks with algorithms to perform classification and regression analysis. Neural networks can be built from scratch or one can make use of already pretrained networks for faster deployment.
2. **AlexNet:** is a pretrained convolutional neural network that is already trained with a large set of data (more than a million images from the ImageNet database). Its support package must be downloaded before it can be used.
3. **Instrument Control Toolbox:** This toolbox allows us to connect MATLAB directly with peripheral devices or instruments such as analyzers, power supplies, signal generators, etc. With it, you can write data to or from MATLAB using TCP/IP, UDP, I2C, SPI, and Bluetooth serial protocols.

4. **echoudp:** This is used to start or stop a UDP echo server for enabling communication between a UDP server and a client.

5. **Computer Vision Toolbox:** This toolbox allows us to connect MATLAB directly with peripheral devices or instruments such as analyzers, power supplies, signal generators, etc. With it, you can write data to or from MATLAB using TCP/IP, UDP, I2C, SPI, and Bluetooth serial protocols

The following python libraries or functions were used:

1. **Adafruit_Python_DHT:** is a python library designed to read the sensor data from a DHT sensor on a Raspberry Pi. It can read humidity and/or temperature data for display or transfer to other physical or cloud platforms.
2. **RPi.GPIO:** General-Purpose Input/output is a row of pins on one side of a Raspberry Pi. The purpose of each GPIO pin can be designated by the user using software and can be used for multiple different purposes.

3. **smtplib:** is a python mailing library that includes the SMTP class that is used to connect to a mail server and to send messages.
4. **socket:** is a python library that provides support for low-level networking interface. It provides the option for communication between raspberry pi and other devices using different forms of communications.
5. **Urllib.request:** is a python module that provides a very simple interface for fetching and opening Uniform Resource Locators (URLs) using different protocols.