ELEN4012 - EMOTION RECOGNITION PROJECT PLAN

Sasha Berkowitz (818737) & Arunima Pathania (1117426)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: The plan for the design and development of a speech emotion recognition classifier is presented. The system is to recognise happiness, sadness, anger and fear in 2 second inputs from English and Afrikaans speakers. This will be done through enhancing a signal with the NPVSS-NLMS, GSVSS-NLMS, RLS and Affine Projection filters, converting them to MFCC spectrograms and using MLP and CNN classifiers. A simple user interface will be developed and all will be carried out using the Python language, as well as Google's TensorFlow library for machine learning aspects. Success will be achieved if the system has an accuracy of 50 % or above. The project's work breakdown has been provided along with inherent risks and their mitigations, as well as socio-economic considerations.

Key words: artificial intelligence, convolutional neural network, emotion recognition, multilayer perceptron, signal enhancement,MFCC.

1. Introduction

Emotion detection from speech is a process that deals with discovering the emotional aspect of speech irregardless of the speech content. This feature would help in providing better services as the machine would be adaptive to human emotion, resulting in a more realistic human-machine interaction. The aim of the project is thus to develop in a system that can detect speech emotion for English and Afrikaans language speakers.

The project employs use of speech enhancement and classification in order to achieve more desirable results. It will also make use of four filters as speech enhancement schemes, namely: Non-parametric variable step size normalised least mean square (NPVSS-NLMS) algorithm, Generalised sigmoid variable step size normalised least mean square (GSVSS-NLMS) algorithm, Normalised lattice recursive least squares filter (NLRLS) algorithm and the fast affine projection algorithm. The speech classification is implemented by neural network algorithms such as Multilayer Perceptron (MLP) and convolutional neural network (CNN).

This project attempts to clearly distinguish between the different emotions in human speech. It will also show comparative results of the developed system in cases where the speech enhancement scheme is absent, between the two languages employed and between the two different classifier algorithms used in order to find out which gives the greatest results.

Section 2. of this report explores previous solutions relevant to the given problem and section 3. the problem specifications. Sections 4. and 5. describe the proposed approach to be taken and how it will be split and managed, respectively. Section 6. lists the project risks and discusses mitigations put into place in order to avoid them and section 7. describes its socio-economic effects. Finally, this report is concluded in section 8..

2. Literature Review

With the increase in the amount of readily available information in the modern world, as well as the vast, growing computational power at our disposal, popularity in the development of artificial intelligence and machine learning is on the rise. This technology is often used to mimic human-like decision making and judgements, such as understanding images and audio. The understanding of images and audio assist in simulating conversations and are useful in cases where chatbots can replace human assistants [1]. In addition to the recognition of spoken words, a new aspect that can be added to these bots is emotional recognition to help them with understanding the human they are conversing with.

For complex problems, such as an automatic emotion recognition in speech, a neural network (NN) is suitable. NNs are part of the representation learning algorithm class, where large and complex problems are broken down into smaller ones, making them more manageable [2]. Within the neural network class, multiple architectures exist and it is important to choose one well suited for a given project.

These NNs, as described above, require precise and valuable data in order to train the network - a process where the network model is fed a large number of sample input cases and their expected outputs in order to 'teach' it how to think for itself [2]. In the case of a speech signal input, often there is unwanted noise present affecting the quality of the information fed into the network. It is therefore important to take steps to minimise the noise present and enhance the signal in the system design by making use of filtering.

Most popular NN architectures employed for audio recognition classifier are recursive neural networks (RNN), convolutional neural networks (CNN) and multi-layer perceptrons (MLP). In the case of emotion recognition, a solution is offered by Ram et al.

where fear is recognised in speech through the use of a MLP classifier and least mean square, normalised least mean square and recursive least mean square filters are applied to input signals [3]. This solution managed a 77% accuracy in recognising fear in speech [3].

An important consideration to take on input data is its representation. In order for the NN, or classifier, to extract and analyse features from each input, the input should be in a form where it is suitable to do so. In order for audio signals to be represented meaningfully often a number of fast Fourier transforms are applied to it and its output frequencies mapped as an array or RGB image. An example of this is demonstrated by Stolar, et al. where real time emotional speech signals are separated into one second blocks, converted to spectrograms and then RGB images, before being analysed using a five convolutional layer CNN[4].

In order to further increase input quality, additional means of filtering acoustic echo are offered by Hamidia, et al. with an improved design on the variable step-size normalised least mean square (VSS-NLMS) algorithm, combining the generalised sigmoid variable step-size NLMS with the estimation error ratio [5].

An ever-growing popular choice for building these neural networks is Google's open-source library, TensorFlow. TensorFlow (mainly implemented in the Python language) uses dataflow graphs with nodes representing mathematical operations and edges tensors - multidimensional arrays [2]. Through the use of the tensor data representation, performing operations on large datasets has become a much more efficient practice. Additionally, the library has support for multiple CPUs and GPUs as well as cross-operating system support [2]. For these reasons, developing neural networks using Python over tools such as Matlab has become well suited and increasingly user-friendly.

3. Problem Specification

3.1 Overview

The aim of this project is to detect an emotion from input speech for a speaker of the English or Afrikaans languages. There are four proposed emotions that the researchers are working on namely, happiness, sadness, anger and fear. The study is intended to assist in the evolving field of artificial intelligence as it will create smarter systems to detect human emotion and enhance the interaction between human and machine. The contextualization section above shows how several methods have been implemented to detect emotion from speech. The main techniques used to perform this project include signal enhancement using adaptive filters , feature extraction using the Mel Frequency Cepstral Coefficients (MFCC) and signal classification using neural network algorithms such as Multi-layer perceptrons and the convolutional neural networks. The following points will motivate the usage of these techniques:

3.1.1 Advantages

- Neural networks are highly accurate when identifying pattern recognition which includes speech recognition.
- Neural networks have high computational power and therefore enable larger data sets to be processed
- Adaptive filters change their filter parameters to get accustomed to the change in signal characteristics in an unknown environment and hence work efficiently for noise cancellation.
- MFCC values give high accuracy in the frequency representation of speech signal.

3.1.2 Disadvantages

- The biggest disadvantages of neural networks is the black box implementation since if something goes wrong you would not know how to fix it.
- Machine Learning problems require very high amounts of data to obtain accurate results which can be very tedious.
- MFCC values are not robust and might need normalization when exposed to a noisy environment.

3.2 Requirements

Listed below are the major requirements for this project:

- The initial requirement entails finding the right algorithms for the signal enhancement and classification processes.
- The designing and developing of the system which would recognize emotions from both English and Afrikaans.
- The system has to be trained and tested with large data sets of different emotions.
- To check the system operations it needs to be validated against a new set of data.
- A user interface is required for real-time application.

3.3 Assumptions

Listed below are the assumptions made for this project:

- The user is effectively able to communicate in either English or Afrikaans
- The neural network would be able to differentiate between emotions with only a couple of thousand data points.

- The validation of the system will be performed in an environment of minimum background noise.

3.4 Constraints

Listed below are the constraints faced during this project:

- The project has a deadline of six weeks.
- The data collection of the project can not be performed without ethical clearance from the University
- The noise cancellation filters do not give a 100% accuracy and so the model will have to be trained with noise in the desired signal.
- Changes in the natural state of voice will affect the system's accuracy i.e the speed of speech or the hoarseness of a voice.

3.5 Success Criteria

Listed below are the success criterion of the project deliverable and management:

- The project must meet all the requirements mentioned above while also adhering to the assumptions and constraints of the project.
- The project is expected to have an accuracy of above 50%.
- The project should successfully be able to distinguish between all the specified emotions in human speech.
- The designed gantt chart should be followed and the desired work load division should be met.

4. Proposed Approach

4.1 Approach Overview

The task of building a system capable of recognising emotion from raw speech input has been separated into a few major subtasks to be performed. These tasks consist of recording an input signal, enhancing that signal through a number of filters, converting that signal into something that could provide meaning to the classifier, classifying the signal through a machine learning model and finally outputting an emotion and displaying it to a user. A breakdown of these tasks are shown in figure 1 and their details will be discussed in sections 4.2 through 4.8.

4.2 Speech Input

Speech signals will be used as an input for both the training model as well as the final application. The inputs will be of a standard two second length, as implemented in [4].

For the model training, speech will be sourced from recordings taken of students. These will be done in a soundproof chamber in order to reduce external interruption. In order to reduce project costs these will be done one of the project partners' iPhones which record at as M4A files, which will be converted to .WAV files for use. This will be done by using the built-in 'Voice Memo' application, which records at frequencies of 44.1 and 48 kHz [6].

In order to elicit emotions, students with an interest in drama or acting will be asked to participate and will be given an excerpt to read, along with an emotion to portray. This will be done in English as well as Afrikaans.

In addition to these recordings taken, sound files may be taken from online databases in order to strengthen the model's accuracy.

4.3 Signal Enhancement

Signal enhancement refers to the refinement of an input signal to meet the requirements of a system for which the input is being used. This project requires an individual to speak into a microphone and the audio signal is then enhanced according to the needs of the project.

As the audio signal will include background noise and channel distortion affecting the speech signal, the signal will be applied to a number of filters. These filters will remove any unwanted frequencies, therefore enhancing the signal to be analysed.

The human speech is typically made up of frequencies ranging from 300 Hz to 3 kHz. Limiting the signal to this frequency band through filtering will reject most part of the noise, as well as the plosive consonants like "p" and "t" which require a higher frequency to be correctly differentiated. This reason contributes to the selection of a higher frequency band which ranges over 300 Hz to 8 kHz[1].

As suggested by [5], two variable step size adaptive filters will be used, being a non-parametric variable step size normalized least mean square (NPVSS-NLMS) algorithm ,a generalized sigmoid variable step size normalized least mean square (GSVSS-NLMS) algorithm and a Recursive least squares filter (RLS). In addition, a fast affine projection algorithm will be applied. These filters will be explained in sections 4.3.1 through 4.3.2.

4.3.1 Adaptive Filters

An adaptive filter is a system with a linear filter which is self-designing and controlled by variable parameters. It uses a recursive algorithm to continuously adjust its tap weights to operate in new environments. It is made up of a digital filter and an algorithm designed to assist in adjusting the tap weights of the filter. The microphone signal is represented by d(n).

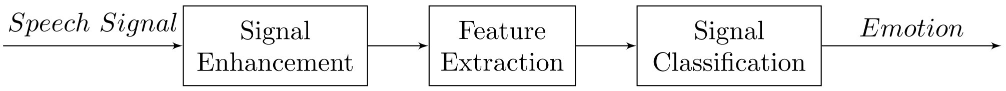


Figure 1: Block diagram of approach overview.

In the single-talk scenario, d(n) is made up of the echo signal y(n), the ambient noise b(n).The resulting error signal e(n) consists of the residual echo signal er(n) and the ambient noise b(n). The output signal of the digital filter is given as:

$$y(n) = X(n) * W(n) \tag{1}$$

where W (n) is the vector of the adaptive filter weights.

The mean square error (MSE) provides the basis of the Least Mean Square algorithm and its adaptations. the error is assumed to be a random variable with a Gaussian distribution. The parameter, step-size in the algorithm is responsible to the rate of adaptation of the filter coefficients.A variable step-size algorithm is used to balance the trade-off between the rate of convergence and the steady-state error. This algorithm uses a large step-size in the initial stages of convergence to increase the rate of convergence and a small step-size to stop the convergence stage corresponding to a small steady-state errors.The weight update recursion of variable step-size parameter is:

$$W(n+1) = W(n) + \mu(n)\frac{X(n)e(n)}{\epsilon + X^T X(n)} \tag{2}$$

where μ(n) is the variable step-size which has a variable positive scalar included to control the filter coefficients update[5].

GSVSS-NLMS

This GSVSS-NLMS uses the variable step size of a adaptive filter on which it's sigmoid function is based . Since this algorithm is a variant of the VSS-NLMS its principle is that a large step size should be used to obtain a better tracking speed while keeping a smaller step size during convergence to maintain the steady-state error. The sigmoid variable step-size(SVSS) in GSVSS-NLMS algorithm is:

$$\sigma_{SVSS}(n) = B(\frac{1}{1 + exp(-A|e(n)|)} - 0.5) \tag{3}$$

where A, 0.001 < A < 0.1, and B, 0 < B < 2, are respectively the shape and the range controlling parameters of the variable step-size function. The generalized sigmoid variable step-size (GSVSS) in GSVSS-NLMS algorithm is:

$$\sigma_{GSVSS}(n) = B(\frac{1}{1 + exp(-A(\sigma_e(n) - \sigma_b)^m)} - 0.5) \tag{4}$$

where m, 0 < m < 5, is the parameter of the generalized sigmoid function.

NPVSS-NLMS

The NPVSS-NLMS algorithm uses an approach where it adjusts the step-size value to attempt and reduce the squared errors at each instant, imposing the condition E[ϵ^2(n)]=E[b^2(n)] where E[$\cdot$] denotes mathematical expectation, and ϵ(n) is an a posteriori estimation error defined by:

$$\epsilon(n) = d(n) - X^T(n) * W(n+1) \tag{5}$$

The optimal variable step-size is given by the following equation:

$$\mu_{NPVSS}(n) = \beta(n) \tag{6}$$

where,

$$\beta(n) = \frac{1}{\epsilon + X^T(n)X(n)}(1 - \frac{\sigma_b}{\epsilon + \sigma_p(n)}) \tag{7}$$

where ϵ is a positive very small number to avoid division by zero, ${\sigma_e}^2$=E[e^2(n)],defines the power of the error signal and ${\sigma_b}^2$=E[b^2(n)],defines the power of the system noise. This condition is only true when σ_e(n) > σ(b), otherwise the value of μ_{NPVSS}(n) is equal to zero [5].

NLRLS

The Recursive least squares filter algorithm is an adaptive filter. It recursively searches for the coefficients that minimize a weighted linear least squares cost function which are related to the input to the filter [11].In this algorithm the filter tap weight vector is updated by:

$$w(n) = w^T(n-1) + k(n)e_{n-1}(n) \tag{8}$$

where

$$k(n) = \frac{u(n)}{\lambda + x^T(n)u(n)} \tag{9}$$

and

$$u(n) = w_{\lambda}^{-}1(n-1)x(n) \tag{10}$$

4.3.2 Affine Projection Algorithm

The affine projection algorithm (APA) works on principles of both NLMS adaptive and RLS adaptive filter algorithms.When speech acts as an input signal the algorithm takes on the low memory requirements of the NLMS and the fast convergence of RLS. Each tap weight vector update of NLMS is viewed as a one dimensional affine projection. The equations for Affine Projection are as follows:

$$e_n = s_n - X_n^t h_{n-1} \tag{11}$$

and

$$h_n = h_{n-1} + \mu X_n \epsilon_n \tag{12}$$

where X$_n$ is the input signal matrix , h$_n$ is the adaptive tap weight vector and e$_n$ consists of background noise [7]. The system output of the algorithm is defined by the equation below:

$$s_n = X_n^t h_{ep} + y_n \tag{13}$$

where X$_n$ represents the input signal, h$_{ep}$ is the echo path impulse response and y$_n$ is the additive system noise.

Fast Affine Projection (FAP) requires the solution to a system of equations involving the implicit inverse of the excitation signal's covariance matrix. The fast affine projection algorithm reduces the cost of the Affine Projection algorithm by N, thus it is suitable for higher projection orders. The equations are similar to the equations mentioned above and the sample period of the system is expanded to implement FAP [7].

4.4 Feature Extraction

Feature extraction is a method to gather the measured data which is intended to be informative and non redundant while working on pattern recognition in machine learning. In theory it is possible to recognize speech from the input waveform. However there is a large variability of the speech signal which makes it necessary to perform feature extraction that will give a good quality input for the system [8]. When designing a neural network the major issue is the method in which the data is to be presented to the neural network. For audio signal several applications convert the input signal to appropriate form including the raw digitized sample stream, machine discovered features, Mel Frequency Cepstral Coefficients and a variety of spectral representations like spectrograms.[12] The feature extraction technique used for this project is the Mel Frequency Cepstral Coefficients.

Mel-frequency cepstrum (MFC) in signal processing represents the short-term power spectrum of an audio signal, based on a linear cosine transform of a log power spectrum on a non-linear mel scale of frequency. Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC.For frequency lower than 1 kHz human ears listen in a linear scale and for above 1 kHz in a logarithmic scale. The mel frequency scale is a linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. The voice signals have most of their energy in the low frequencies.The formula below is used to compute the mels for a given frequency 'f' in Hz:

$$mel(f) = 2595 * log(1 + f/700) \tag{14}$$

In (14) we have a very well known result. Every frequency presented in hertz has a comparable value according to the mel scale. The cepstrum is the spectrum of a spectrum as it is calculated using the forward fourier transform of the spectrum. The advantages of using MFCC is that any repeated patterns in a spectrum is considered as specific components in the cepstrum. A spectrum contains several sets of harmonic series which are confusing because of the overlap. Though in the cepstrum, they will be separated in a way similar to the way the spectrum separates repetitive time patterns in the waveform. Spectrogram have time dependent problems where the same word taken from different audio files would result in two separate spectrograms since they are affected by the speed of the audio and the pauses in the signal. Mel frequency cepstral coefficients are an ideal method for coping with these problems. [10].

Once converted by the MFCC, each sound signal is transformed into an array of numbers which represent the coefficients of the process. This further suits the TensorFlow neural network application as the MFCC's array are a fit to the 1-D array, the ideal input to the neural network, and thus can be processed better than a conventional .WAV signal.

4.5 Signal Classification

The signal classification stage deals with the use of features extracted in order to recognise patterns within inputs and map each to a corresponding output. This will be done by designing a model and then training the model using a large dataset where the inputs and their corresponding outputs are specified. The signal classification will therefore be implemented through the development of a multi-layer perceptron (as suggested in [3]) as well as a convolutional neural network (as suggested in [4]) and their results compared.

The models will be developed by making use of Google's open-source framework, TensorFlow, in the Python language. Both topologies developed will take in equally sized MFCC spectrograms as input. Breakdowns of the two topologies chosen, as well as explanations as to why they were chosen, will be discussed in the following sections, 4.5.1 and 4.5.2.

As the TensorFlow framework is tensor-based, it is suited for a high number of numeric operations and thus machine learning and deep learning, and the use of tensors (multi-dimensional arrays) provide a much higher learning rate than other machine learning systems. Due to time constraints, the TensorFlow library was chosen over using other systems, such as MATLAB, as many models will be trained for comparative reasons which is a lengthy procedure.

4.5.1 Multi-Layer Perceptron

The first neural network to be implemented, the MLP, is a standard NN based classifier often used for speech signal analysis and emotion recognition. The classifier model makes use of a number of decided hidden layers, and the values of which are characterised by weights and biases and determined by equation 15. These weights and biases are altered by the use of back propagation, using equation 16, where $\frac{\delta\lambda}{\delta w}$ is the gradient of λ and δ the learning rate [3].

$$hidden\ layer = (input \times weight) + biases \tag{15}$$

$$\Delta W = -\delta \frac{\delta\lambda}{\delta w} \tag{16}$$

4.5.2 Convolutional Neural Network

Convolutional networks are a type of feed forward neural networks. These are often used for complex operations, such as audio and image recognition, as their convolutional learning structure aid in extracting patterns and features from inputs in order to train itself as to their meanings.

These CNNs consist of input and output layers, as well as a number of hidden layers. These hidden layers are in the form of convolutional layers, pooling layers, fully connected layers and weights, and a typical architecture is shown in figure **??**, where an input V is convolved with a weight matrix W [13]. The resultant weight sharing assists in mapping features of the inputs to their corresponding outputs [13]. This results in n feature maps of size $\frac{t-m+1}{s} \times \frac{f-r+1}{v}$, where t and f are equal to input dimensions of the data, m and r the dimensions of the matrix and s and p the time and frequency of the filter, accordingly [13].

Following, the pooling layer assists in removing distortions in input resulting from noise or speaking styles and speeds. Given a pool size $p \times q$, a resultant output will be of size $\frac{t-m+1}{s \times p} \times \frac{f-r+1}{v \times q}$ [13]. As suggested by [4], five convolutional hidden layers will be implemented for the project.

4.6 Output

Once put through the neural network, an ID will be outputted corresponding to one of the four emotions analysed. A look-up will be performed on this ID and the emotion will be displayed on the user interface, detailed in section 4.7.

4.7 User Interface

A simple user interface will be developed using Python's graphics library, TKinter. The interface will allow for users to start a 2 second voice recording. The recording will then be converted, input to the trained network and the emotional output will be displayed.

4.8 Training and Testing Data

Data gathered (through speech recordings and internet samples) will be separated into three groups, being data used for training, data used for testing and data used for validation. The training data is used to make the system accustomed to the required model and make the designed system respond as accurately to the desired outcome as possible. The validation dataset is used to tune the hyper-parameter of the model while also evaluating the system on it's initial training. The testing data is now used to check the final model along with it's results [14].

It is imperative that the dataset used for training be large, as through the inputs given the network will be 'taught' how to differentiate between different emotions and the more samples fed in at this stage, the higher the systems accuracy will be. Similarly, a large number of inputs should be used for test and validation databases in order to ensure accuracy in the system's success specifications.

In order to take these factors into account, a recommended 80% of data collected will be used for training and the remaining 20% for testing, a ratio implemented in [4] The data obtained from volunteers will be recorded in a sound proof room and then again with a noisy background where the first instance will be used to train the system and the latter to test it.

In addition to a singular model trained, combinations of filters and topologies will be shuffled around and their outputs compared, for example the MLP network with no filters applied to inputs, the CNN with one filter applied and the MLP with all developed filters applied. Finally, accuracy between the English and Afrikaans systems will be compared.

5. Project Management

Throughout the duration of the project, the partners will be in contact with each other regarding their progress or any other obstacles which they may face. In addition, at least one weekly face-to-face meeting will be held between the partners and project supervisor.

Project files will be housed on a private GitHub repository. The subsections following will outline how the project components listed in section Approach will be divided and the estimated time to be taken for each. In addition, a gantt chart of the proposed project timeline is illustrated in appendix B, and its work breakdown structure in appendix A.

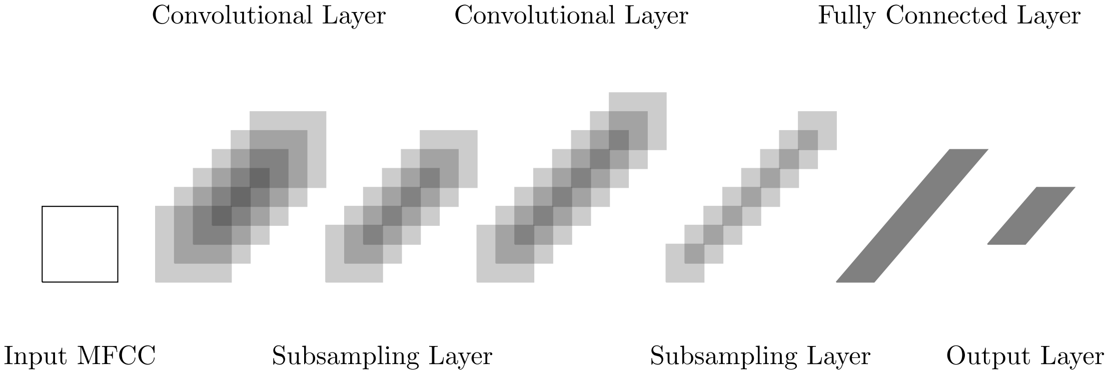


Figure 2: Typical convolutional neural network architecture.

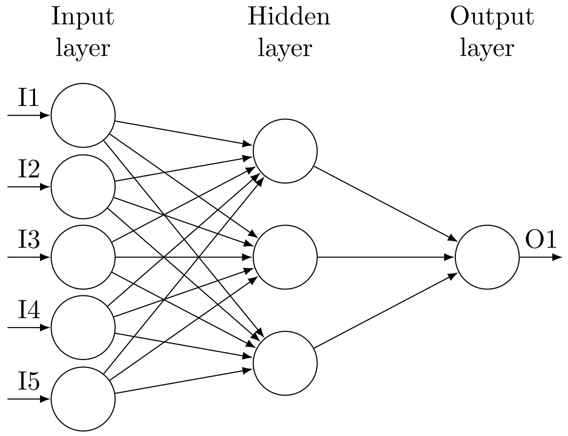


Figure 3: Diagram of a generalised deep neural network.

It is important to note that in project planning, each aspect was timed in order to leave a week free in case of delays experienced during development.

5.1 Project Initiation

At the start of the project technical, set-up is necessary. These tasks are detailed in table 1

Table 1: Table listing initiation task details.

Task	Duration	Group Member
Setup GitHub repo	0 days	S.S. Berkowitz
Install Python 3.5	0.25 days	Collective
Install TensorFlow	0.25 days	Collective

Resources required: Git

5.2 Data Collection

Collection of data will be done through utilising both online sources and recordings made in person and have been listed in the table items below. A full two week period will be used for sourcing and separating data, which will be done in parallel and organised as it is collected. A full breakdown of the tasks is shown in table 2. Organising of data will also include splitting it into 2 second files.

Table 2: Table listing data collection task details.

Task	Duration	Group Member
Setup recording equipment	0.5 days	S.S. Berkowitz
Collect data via recordings	9 days	Collective
Collect data via online sources	9 days	Collective
Organise data	9 days	Collective

Resources required: Microphone, Recording application, access to soundproof chamber

5.3 Signal Enhancement

The signal enhancement phase will include the implementation of the four noise cancellation algorithms and applying them to the datasets. Each partner will be responsible for two separate algorithms and applying them to the collected data. These tasks will be implemented in parallel and their details are listed in table 3.

Resources required: Python 3.5

5.4 Signal Classification

The two partners will work in parallel developing the two neural network topologies for the project, details of which can be seen in table 4

Resources required: Python 3.5, TensorFlow

5.5 User Interface

The final week of the project will focus on the development of a live integration and a user interface. These tasks are separated in table 5.

Table 3: Table listing signal enhancement task details.

Task	Duration	Group Member
Implementation of Algorithm 1	2 days	S.S. Berkowitz
Implementation of Algorithm 2	2 days	A. Pathania
Implementation of Algorithm 3	2 days	S.S. Berkowitz
Implementation of Algorithm 4	2 days	A. Pathania
Application of alorithms 1 & 3	1 days	S.S. Berkowitz
Application of alorithms 2 & 4	1 days	A. Pathania

Table 4: Table listing signal classification task details.

Task	Duration	Group Member
MLP code design	1 days	A. Pathania
Convolution Neural Network code design	1 days	S.S. Berkowitz
MLP code implementation	4 days	A. Pathania
Convolution Neural Network code implementation	4 days	S.S. Berkowitz

Resources required: Python, TKinter

6. RISKS AND THEIR MITIGATIONS

6.1 Data Security

The researchers have access to the voices of several individuals from various sources. This data can be subjected to misuse as it can be stolen from the computers used for the study and cloned to hack into the individual's computer.This personal information taken out of context can lead to data breach and pose a threat to the individual's privacy.

The method of mitigation in this case is to securely keep the voice samples on the researchers computers in a password protected file and destroy these samples as soon as the project finishes, which is after six weeks of the project commencement. This data at no point in time will be shared by any third party candidate for any use.This way the cloning of the samples can be protected.

6.2 Ethical Issues

The access of any individual's personal belonging is a matter of ethical interest. The researchers could take data from students around campus and later use it to for means that are not included in their study.This would expose the private information of volunteers to the public.

Table 5: Table listing signal classification task details.

Task	Duration	Group Member
Python recording & conversion	2 days	A. Pathania
I/O integration	2 days	A. Pathania
User interface	4 days	S.S. Berkowitz

The methods of mitigation for this risk is as follows: The participants are all volunteers and are willingly contributing to the study.They will be provided by an information sheet to explain to the them the process of the study and they can leave the study at any given time if the wish to. The will be given a piece of literature to speak and will thus not provide any personal information. An ethics clearance is applied by the researchers to gather this data from the ethics committee of the University of Witwatersrand and no data acquisition will be started without an acceptance letter from the committee.

6.3 Inaccurate Results

The system is trained with data either obtained from online sources or recorded voices of students willing to participate in the study. Both data sets are prone to a number of background noises or channel distortion which will interfere in the proper training of the system. If the system is not properly trained it might end up giving inaccurate results

The proper mitigation for this system includes using the online data from sources which specify that the data has been used in speech recognition systems and hence has been recorded in a low noise environment with good equipments.The data recording done by the researchers should be performed in a soundproof chamber to minimize the distortion in the recordings. T The system is expected to give an accuracy of 50%. The system then undergoes through tests which will be run on the data to see if the desired outcome is achieved.

6.4 Intellectual Property Risks

Intellectual property risks often causes trouble for innovators or researchers trying to build something new in software. As patenting would only apply to their code implementation and not their design they are at risk.

In the case of Machine Learning this risk is not as evident as it is not as dependent on code implementation as much as the amount of data that is used to train the developed models. Therefore, a third party is less disposed to stealing the resultant system which will have an accuracy dependant on how well the system is trained with gathered data.

7. Social Impacts

With the ever-changing global economic status, the average person's lifestyle is affected, where they are constantly looking for their best suited source for goods and services [9]. As such, businesses need to look for new innovation in order to stay relevant to customers, customer service being a main aspect.

Artificial intelligence (and emotion recognition within it) has become ever more relevant within this field with the automation of the customer-provider experience. Employing such technologies provide efficiency in customer's time, lengthens hours that the provider can be contacted, as well as assists in providing answers going beyond a script or set of questions a human is trained to answer as machines formulate their own solutions with the assistance of deep learning [9]. The implementation of the above not only causes a better customer experience, but also has a positive impact on the business itself as well as the local economy by keeping customers' interests local [9].

With the positive impacts this technology has on an employer, it also provides a negative one to the employee. Due to the increase in technological assistance, a Forrester study estimated that 25% of jobs will be at risk by 2019 [1]. As such, studies in AI and human understanding can have a negative effect and changes, on the part of workers, should be made in order to keep up with the current job market.

8. CONCLUSION

This report discusses the relevance of Artificial Intelligence and more specifically emotion recognition from speech. It introduces the plan for implementing a project which is able to identify happiness, sadness, fear and anger from human speech of both English and Afrikaans speakers. The report further discusses the details of the methodology used for implementing the aforementioned system. Adaptive filters are used to cancel the background noise since they adjust their parameters according to the requirement of the environment, followed by MFCC being used to recognise the relevant aspects of the signal to make them suitable input for a neural network. The Machine Learning algorithms, MLP and CNN, finally give an output which states the emotion detected. The expected accuracy based on reviewed literature is above 50%. The project has a six weeks deadline within which the researchers are expected to collect data, process it, train the system and finally validate its functionality. The report further discusses data breaches and other risks which the study may impose during it's execution, along with mitigations to be taken on each and the social impact of the study on society.

References

[1] Heek, N.F. *How Chatbots are Killing Jobs and Creating New Ones*, 2017. https://venturebeat.com/2017/06/18/how-chatbots-are-killing-jobs-and-creating-new-ones/ [Last accessed: 15-07-2018]

[2] Shaikh, F. *An Introduction to Implementing Neural Networks using TensorFlow*, 2016. https://www.analyticsvidhya.com/blog/2016/10/an-introduction-to-implementing-neural-networks-using-tensorflow/ [Last accessed: 13-07-2018]

[3] Ram, R., Palo H.K., Mohanty, M.N. *Recognition of Fear from Speeach using Adaptive Algorithm with MLP Classifier* ICCPCT, 2016.

[4] Stolar, M.N., Lech, M., Bolia, R.S, Skinner, M. *Real Time Speech Emotion Recognition Using RGB Image Classification and Transfer Learning* School of Engineering, RMIT University, 2017.

[5] Hamidia, M., Amrouche, A. *Improved Variable Step-Size Adaptive Filtering Algorithm or Acoustic Echo Cancellation* USTHB Faculty of Electronics and Computer Science, 2015.

[6] Hill, Andrew R. *Analysis of Voice Recordings Made Using "Voice Memos" Application for iOS* University of Colorado, Denver, p.28, 2014.

[7] Steven L. G., Tavathia A. *THE FAST AFFINE PROJECTION ALGORITHM* ,Acoustics Research Department AT&T Bell Laboratories.

[8] Shrawankar U., Thakare V. *TECHNIQUES FOR FEATURE EXTRACTION IN SPEECH RECOGNITION SYSTEM: A COMPARATIVE STUDY* ,(Computer Science & Engg.), SGB Amravati University .

[9] Bansal, S. *Why is Artificial Intellegence (AI) Relevant Today?*, 2017. https://www.linkedin.com/pulse/why-artificial-intelligence-ai-relevant-today-sanjiv-bansal [Last accessed: 15-07-2018]

[10] Gevaert W., Tsenov G., Mladenov V. *Neural Networks used for Speech Recognition* ,JOURNAL OF AUTOMATIC CONTROL, UNIVERSITY OF BELGRADE,2010.

[11] Singh, K., Gu Y. *Speech Enhancement Based On Noise Reduction* ,Electrical Engineering Department, University Of Rochester, 2016.

[12] Wyse L. *Audio spectrogram representations for processing with Convolutional Neural Networks* ,National University of Singapore, January 2017.

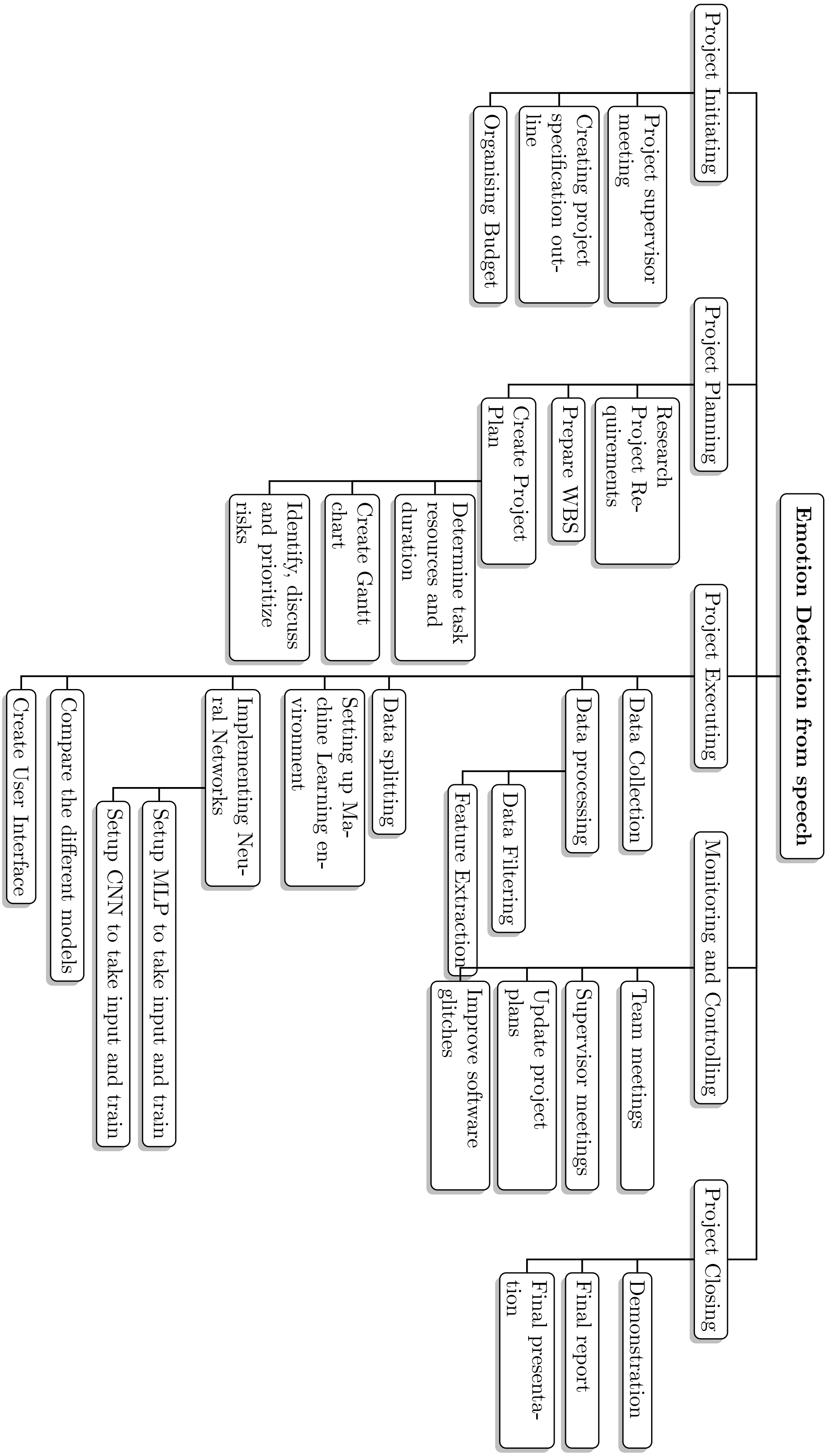
[13] Sainath, T.N., Parada, C. *Convolutional Neural Networks for Small-footprint Keyword Spotting*, Google Inc., New York, 2015.

[14] TARANG SHAH *Train, Validation and Test Sets*Dec 2017. Available at : http://tarangshah.com/blog/2017-12-03/train-validation-and-test-sets/ [Last Accessed on : 15 July 2018]

Appendix

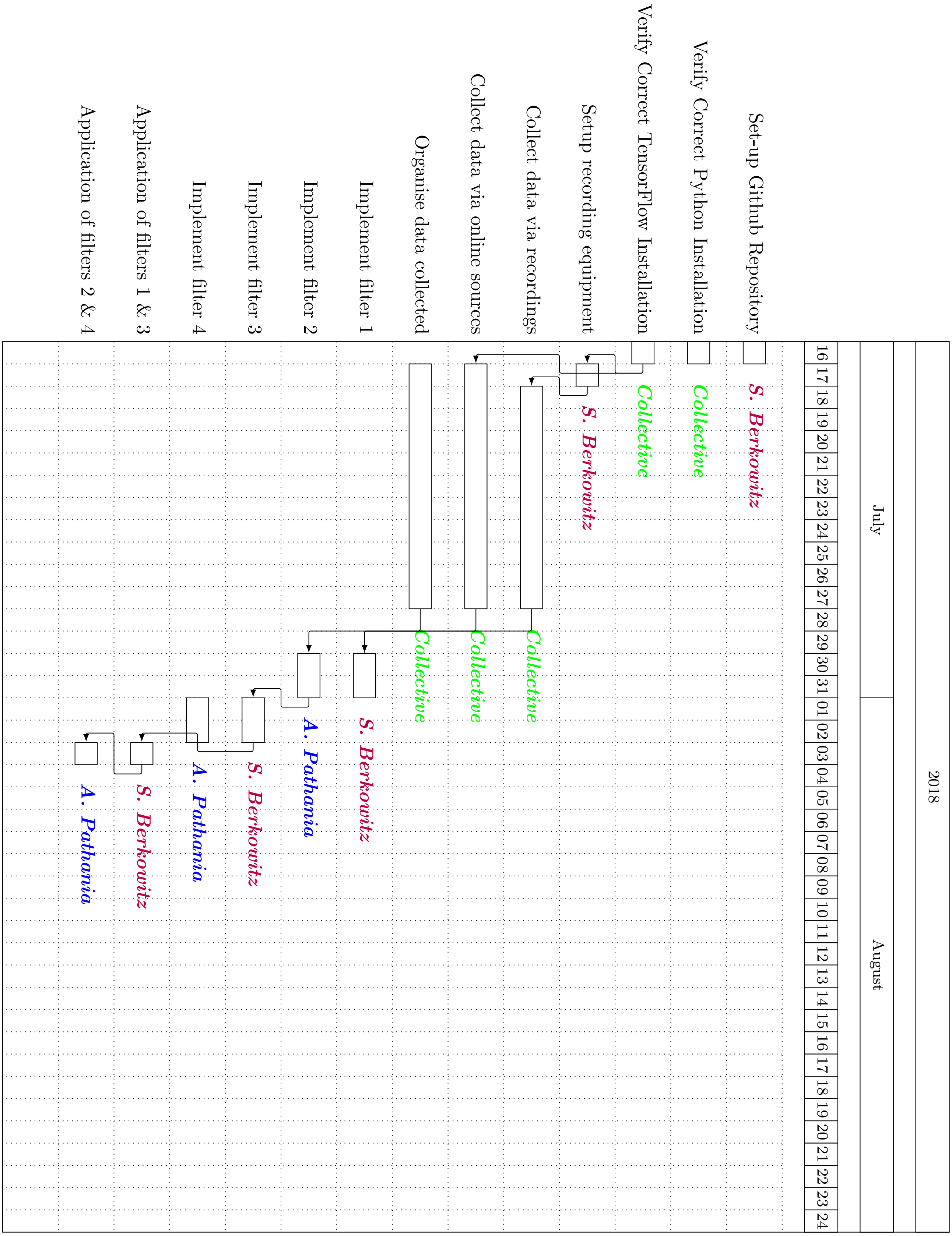
A Work Breakdown Structure

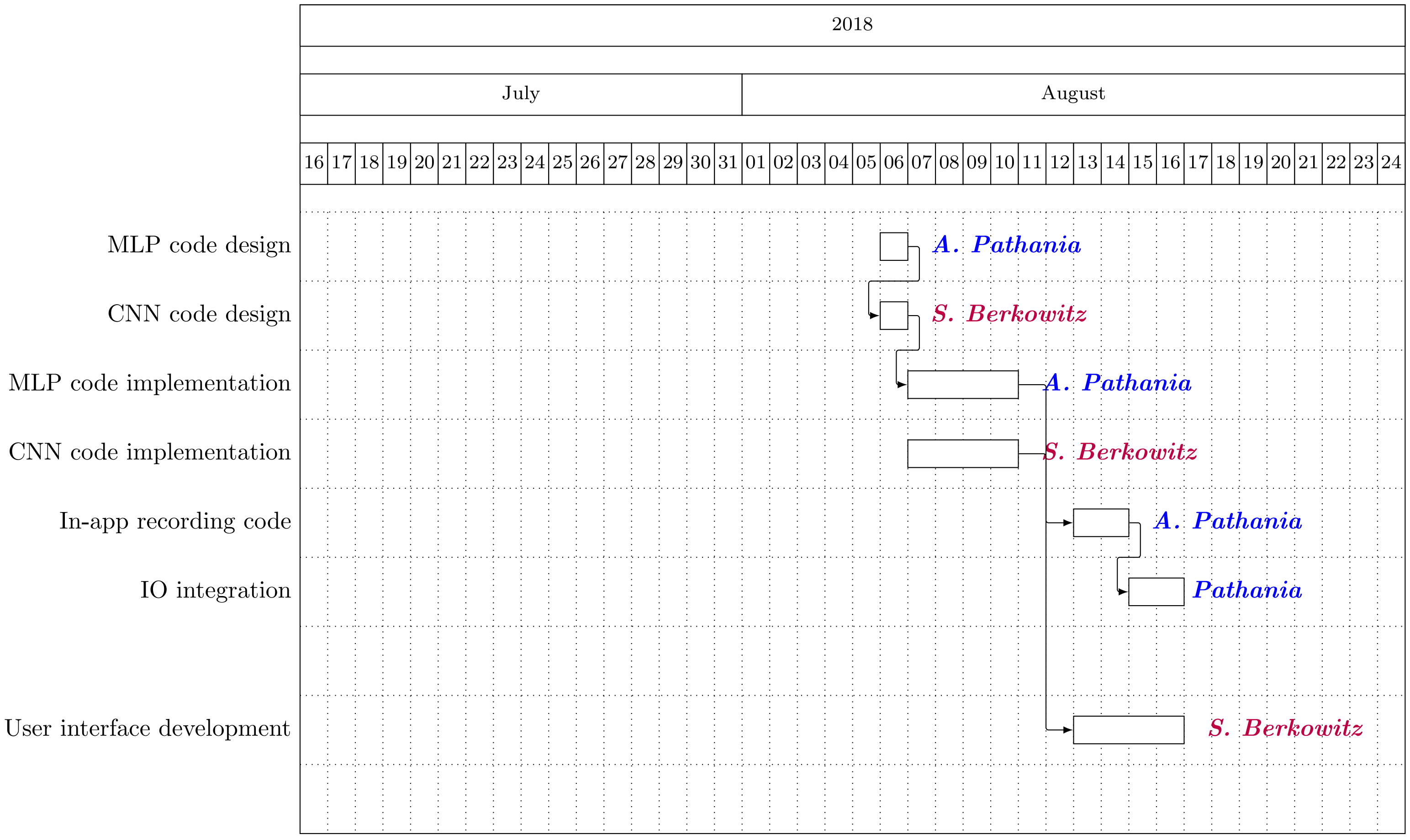
The proposed work breakdown structure for the project follows:


Emotion Detection from speech
Project Initiating
Project supervisor meeting
Creating project specification outline
Organising Budget
Project Planning
Research Project Requirements
Prepare WBS
Create Project Plan
Determine task resources and duration
Create Gantt chart
Identify, discuss and prioritize risks
Project Executing
Data Collection
Data processing
Data Filtering
Feature Extraction
Data splitting
Setting up Machine Learning environment
Implementing Neural Networks
Setup MLP to take input and train
Setup CNN to take input and train
Compare the different models
Create User Interface
Monitoring and Controlling
Team meetings
Supervisor meetings
Update project plans
Improve software glitches
Project Closing
Demonstration
Final report
Final presentation

B Gantt Chart

The gantt chart of the proposed project timeline follows:




2018
July
August
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
MLP code design
A. Pathania
CNN code design
S. Berkowitz
MLP code implementation
A. Pathania
CNN code implementation
S. Berkowitz
In-app recording code
A. Pathania
IO integration
Pathania
User interface development
S. Berkowitz