Convolutional Neural Network for Emotion Detection in Audio Recordings.

Project Report

School of Computer Science & Applied Mathematics
University of the Witwatersrand

Group Members:
Samuel Oladejo
2441199
Phola Bavuma
1848739
June 18, 2022

A Project report submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Bachelor of Science with Honours

Declaration

We, Samuel Oladejo (2441199) and Phola Bavuma (1848739)hereby declare the contents of this project report to be our own work. This project report is submitted to the ADAPTIVE COMPUTATION AND MACHINE LEARNING (COMS4030A) course, in partial fulfilment of the requirement for the degree of Bachelor of Science with Honours in Computer Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

Abstract

Human expression recognition from a number of sources, such as speech, text, and photographs, has always been an interesting field. Face and voice recognition have experienced gradual improvements in accuracy and latency as deep learning technology advances, with a range of articles investigating various implementation approaches being published on a regular basis. We hope to contribute to the body of knowledge by investigating the application of Convolutional Neural Networks for speech recognition and emotion recognition in this research. We did this by comparing chroma, Mel-Spectrogram, and MFCC on eight different emotional cues, including happy, sad, startled, neural, furious, disgust, and fear, and selecting the best feature to train a 13-layer convolutional neural network to predict human emotion.%.

Contents

1 Introduction

Speech emotion recognition is the job of identifying the emotion expressed by an individual through examination of how one has verbally conveyed a message. This task is done effortlessly by humans through the use of communication skills that have been developed over their lives. Though this ability may come easy for humans it is a more strenuous task for machines. This project aims to develop a model to efficiently predict the speaker's emotion through the use of the Convolutional Neutral Network model. Each audio file differs in terms of pitch, frequency, intensity, and speaking rate. The different emotions classified in this paper are neutral, anger, happy, sad, disgust, fear, and surprise. In Section 2 describes the dataset adopted to train, validate and test our model. Section 3 describes all the processing techniques used on the dataset to ensure the correct training and testing of the model and details the architecture of the model as well as its implementation. Lastly, Section 4 presents the results achieved by our model.

2 Dataset

2.1 Dataset Description

The Toronto Emotional Speech Set (TESS) [Lok 2019], The Ryerson Audiovisual Database of Emotional Speech and Song Dataset (Ravdees) [Livingstone 2019], Surrey Audio-Visual Expressed Emotion (SAVEE)[Lok 2019], and Crowd-Sourced Emotional Multimodal Actors Dataset (CREMA-D) [Lok 2019] were all used in this work. TESS dataset comprises 2800 audio recordings produced from 200 target words spoken by two female actors, aged 26 and 64, conveying eight different emotions: anger, contempt, fear, happiness, pleasant surprise, sadness, and neutral. RAVDESS contains 1440 audio files derived from 60 trials in which 24 professional actors (12 females, 12 males) vocalize two lexically matched statements in a neutral North American accent, expressing eight different emotions: anger, contempt, fear, happiness, pleasant surprise, sadness, and neutral. SAVEE contains 480 audio files recorded by four English-speaking males aged from 27 to 31 years, each having 120 utterances expressing 7 emotions: neutral, anger, disgust, fear, happiness, sadness, and surprise. CREMA-D consists of 7442 audio files from 91 actors (48 male and 43 female) aged 20 to 74 from various races and ethnicities expressing six (6) different emotions: anger, disgust, fear, happy, neutral, and sad. There are a total of 12,162 audio files in these datasets, with 10,913 useable (non-nan) files (6,578 females and 4,335 males). It also has the same number of emotions as the audio files (angry-1650, calm-192, disgust-1650, fear-1650, happy-1650, neutral-1400, sad-1649, surprise-1072). All loaded audio recordings were 2.5 seconds with a sample rate of 44100 in a mono(1-channel) format. Training, Validation, and Testing dataset were split into 75-15-10 respectively with each having 8 columns corresponding with one of the following labels:

0: 'Neutral',
1: 'Happy',

2: ‘Sad’,
3: ‘Calm’,
4: ‘Disgust’,
5: ‘Anger’,
6: ‘Fear’,
7: ‘Surprise’.

3 Methodology

We explored concepts related to the project’s core in previous sections. In this section, we formally state the project question and provide a more detailed description of the learning system. In Section 3.1, we outline the project’s goals and the problem we attempt to tackle. In Section 3.2, we look at the dataset’s relationships and trends. In Section 3.3, we go deeper into understanding the audio dataset by explaining all of the strategies used to convert the audio dataset into a format suitable for our model. Section 3.4 covers the mechanics of how we trained the model and gives the baseline implementation of our model.

3.1 Motivation

Emotional Recognition of auditory information is a difficult task for computers, yet it is very simple for humans. With developments in deep neural networks, CNN has proven to be well-suited to learning patterns that are translation invariant and feature spatial hierarchies (F. Chollet, 2018)in computers. As a result, we implemented CNN with MFCCs,Chrome and Mel as features to predict emotions. Utilizing these features we aim to find patterns and correlations within the dataset to produce a satisfactory emotion recognition model.

3.2 Data Exploration and Visualization

We noted that our dataset contains approximately 60.3% and 39.7% female and male audio files respectively. Emotions were distributed as follows: 15.1% angry, 1.7% calm, 15.1% disgust, 15.1% fear, 15.1% happy, 12.9% neutral, 15.1% sad, and 9.8% surprise. We dropped the intensity column due to the disparity between the number of the known intensity data and the unknown ones, and used the audio signal pitch as a replacement. The initial size for the model was 10913. The distribution adopted for splitting our dataset into training, validation, and testing is a 75%,15%, and 10% split. This was done to guarantee enough data points were provided to the model for sufficient training and as well ensure correct model validation. Provided that we had a size of 10,913 for our dataset, we feared that the model would overfit the dataset if trained on a large set of datapoints. To improve generalizability of the model, we performed a random sample of approximatly 32% resulting in a datasize of 3470, an amount we deemed

satisfactory for training and testing the model. To gain more insights, we visualized our the distributions using each column and their unique values, See 1, 2 and 3.

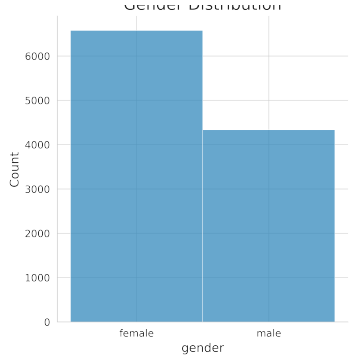

Figure 1: Gender Distribution

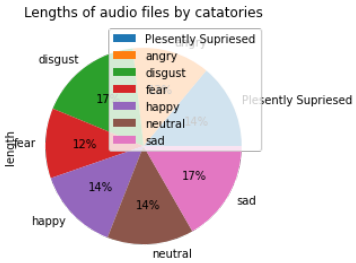

Figure 2: Emotion Distribution

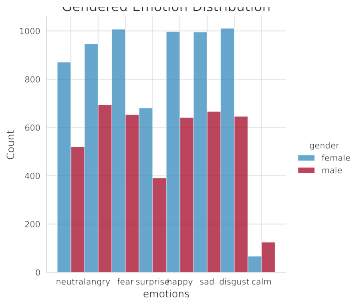

Figure 3: Gendered Emotion Distribution

2 demonstrates the average length of the audio files in each emotion category. There is a somewhat even distribution of the length for each audio file. This assures us that no emotion will not greatly influence our model and dominate the dataset.

3.3 Data Pre-Processing

A python library called Librosa was used to analyze the audio files in the dataset and extract the essential features. The audio files were loaded in a floating point time series with a sample rate of 44100Hz, a duration of 2.5, and a Mono Channel. We extracted the Chroma, Mel, and MFCC features from each audio sample to train our CNN model. The Chroma feature evaluates the pitches of the audio file over a set of timestamps. This feature extraction presented us with 12 feature columns. The Mel feature denoted as the logarithmic transformation of an audio file's frequency. This feature extraction provided 128 feature columns. The last feature extracted from our audio dataset was the MFCC(Mel Frequency Cepstral Cofficients) feature. This feature takes the Mel-Spectorgram and selects a compressed representation of the frequency bands. This feature extracted 40 feature columns from the dataset. Graphical representations of these features can be seen in the Fast Fourier Transforms 4, Filter Bank Coefficients5, Mel Frequency Cepstrum Coefficients 6 and Spectragram 7 diagrams.

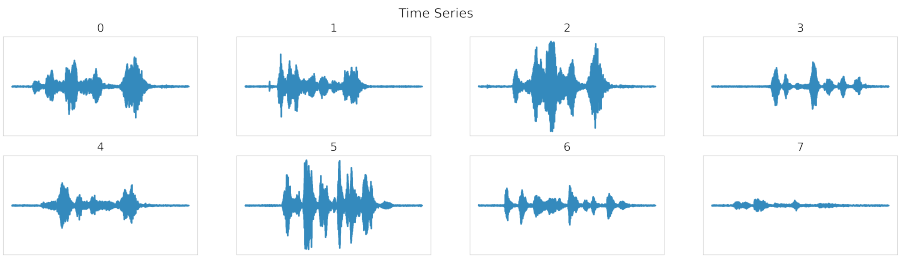

Figure 4: Time Series

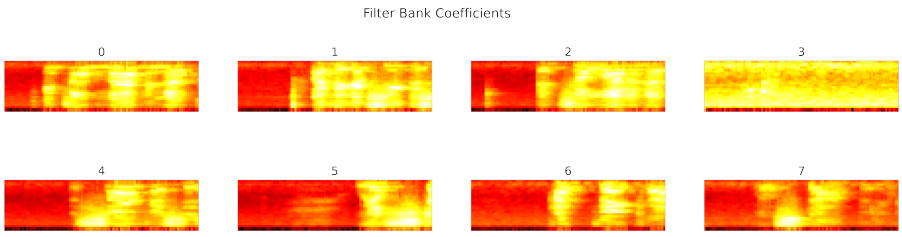

Figure 5: Filter Bank Coefficients

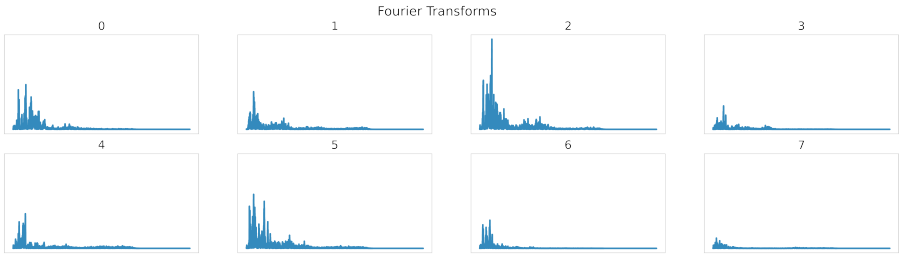

Figure 6: Audio Fast Fourier Transforms

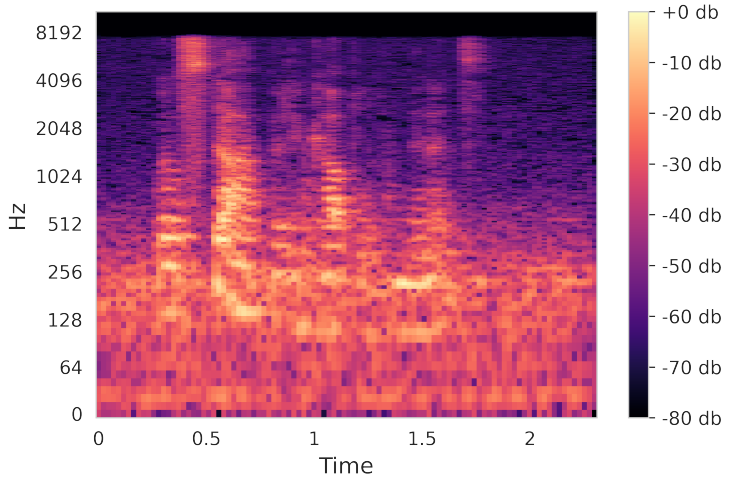

Figure 7: Audio Spectrogram

We augmented the features and true values in order to improve accuracy and boost data diversity. The operations conducted on the respective features and true values are one-hot encoding, Normalization and dimension expansion. The data flow of these processing techniques is described and depicted in 8.

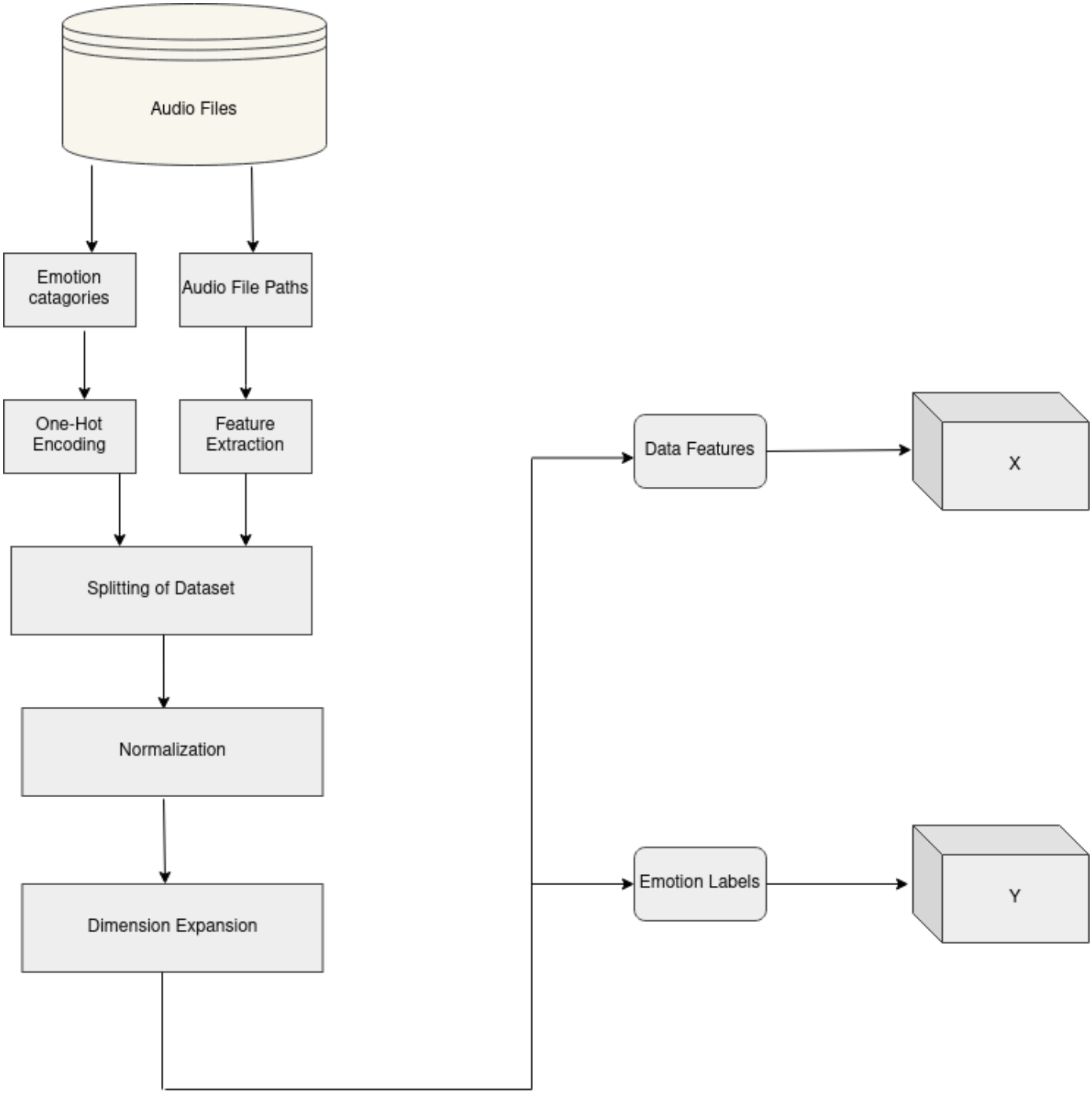

Figure 8: Project Architecture

3.4 Baseline Model Methodology and Implementation

For our baseline model we incorporated a 13 layer Conventional Neural Network. The first 3 layers are a 1-D convolutional layers that take in the size of our input that is a 180 x 1 which correlates to our 180 data features. This is followed by another two 1-D convolutional layers with reduced dimensionality to our previous. This is so our feature dimensions are scaled down to better encompass the data. The activation function used

of these layers is the RELU function as well as the sigmoid function at the output layer. The full architecture of our baseline model is presented below in figure []. Using this model presented us with the highest accuracy as seen in Section 4.

Initially the model was to be trained on the different set of features to examine which of the features presented us with the most optimal prediction rule. Though this was our aim initially, after training three different models on the three separate features , the results returned from this experiment were quite unsatisfactory. The validation accuracy on the Chroma, Mel and MFCC were approximately 15%. With such validation accuracy , it isn't logical to proceed to test the models on the testing data. The respective accuracy graphs for these models are presented in figure[]. To Solve this issue we incorporated the technique of stacking these features together into one data frame in the hopes of producing promising results. This presented us with a feature dataset containing 180 feature columns. This solution enhanced our accuracy greatly and is further shown in Section 4.

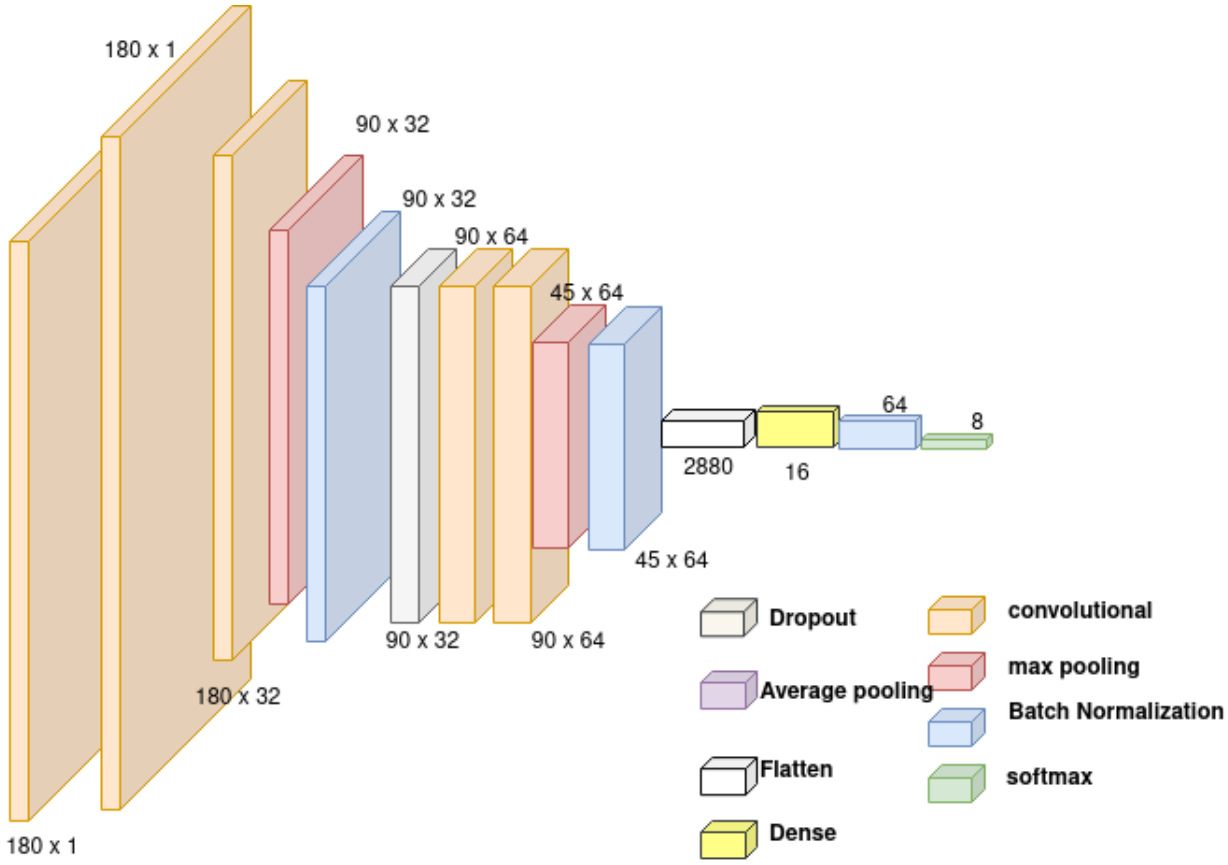

Figure 9: Model Architecure

3.5 Platform

All exploration, techniques, and experiments in this project were conducted in the Jupyter Notebook using the Python programming language. We chose Google Colab for the coding component of our project due to a lack of processing power on our personal

PCs, as deep learning requires advanced tools. To accomplish this, we set up a shared Google Drive ('https://drive.google.com/file/d/1eTOk6YWwOmE54VKOvd8xxWNzc0-1YzDF/view?usp=sharing') with all of the required files, including the notebook. The Google Colab notebook requires Google Drive authentication, and the rest of the notebook will only work if the shared folder is in the same main folder as our Google Drives.

4 Results

A 75% training and 15% validation set were fed into the model to get obtain the results above. In our final classification accuracy, through our model training on the augmented dataset, we obtained results of 86.6% and 85.2% accuracy for the training and validation sets respectively. The detailed results of the baseline training are presented in 10: Model Graph, 11: Classification Report and 12: Confusion Matrix

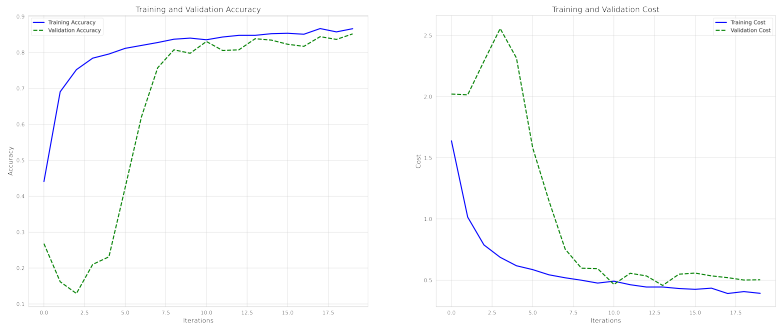

Figure 10: Baseline Model Graph

	precision	recall	f1-score	support
0	0.93	0.81	0.86	78
1	0.22	0.80	0.35	15
2	0.99	0.99	0.99	67
3	0.78	0.75	0.76	87
4	0.90	0.82	0.86	74
5	0.98	0.92	0.95	64
6	0.97	0.76	0.85	75
7	0.98	1.00	0.99	59
accuracy			0.85	519
macro avg	0.84	0.86	0.83	519
weighted avg	0.90	0.85	0.87	519

Figure 11: Baseline Classification Report

Our experimental results indicate that our model performed reasonably well; nevertheless, we can use optimization approaches to determine the optimum hyperparameters for our model, as detailed in the following Section 4.1.

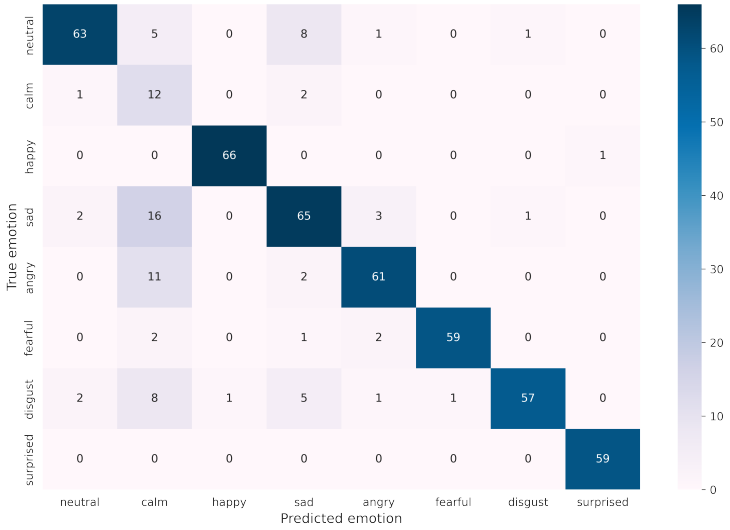

Figure 12: Baseline Confusion Matrix

4.1 Optimization Techniques

CNNs grow exponentially as the training parameters are increased. This is because the deep model has a huge number of hyperparameters to tweak. As a result, we fitted an additional model to optimize the selected hyperparameters to get the optimal result. To assist us in searching the best model, we used the Random Search method from the keras tuner Library, which assesses numerous different factors on a model to discover the best combination. This procedure selects from a set of hyperparameters with boundaries, number of trials, and epochs per trial. In our Notebook, this was implemented using validation accuracy as our target aim to optimise the model using the following bounded hyper parameters per layer to tune:

Layer	Hyperparameter	Default	Range
Conv_Filters_1	Number of Filters	64	{64, 128, 256}
Conv_Activation_1	Activation Function	relu	{relu, sigmoid}
Dropout_Rate_1	Dropout Rate	0.25	(0.0, 0.5) step: 0.05
Conv_Filters_2	Number of Filters	16	{16, 32, 64}
Conv_Activation_2	Activation Function	relu	{relu, sigmoid}
Dropout_Rate_1	Dropout Rate	0.25	(0.0, 0.5) step: 0.05
Adam_Optim_LR	Optimizer	0.001	$(1e^{-4}, 1e^{-2})$

Figure 13: Hyperparameters

The hyperparameters tuner accept a seed value as well as the values listed in the table below.

Argument	Value
Max trials	20
Iterations per trial	10
Number of Iterations	20

Figure 14: Tuner Arguments Table

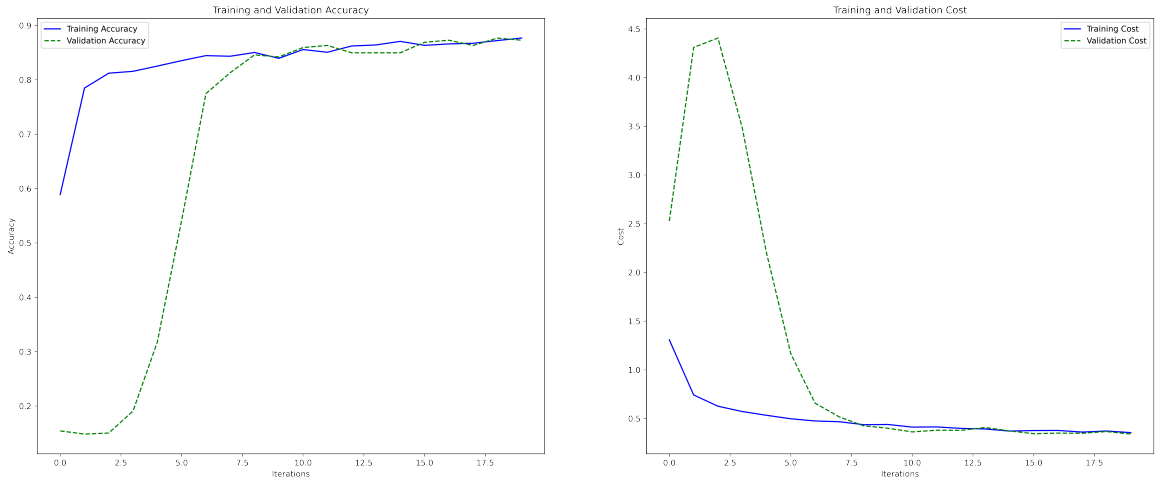

Figure 15: HyperTuned Prediction Graph

The ideal set of hyperparameters is determined by doing a number of trials with different values for the bounded hyperparameters, following which the keras tuner selects the best values by selecting the combination with the best performance on the validation set. Because of the nature of the optimiser, we also had to identify the ideal iterations with the highest accuracy, otherwise the model degrades rather than improves. This is accomplished by running the model with the tuned parameters and selecting the best number of iterations. The accuracy of the tuned set of parameters was 89.8 percent, compared to 86.6 percent for the baseline model.

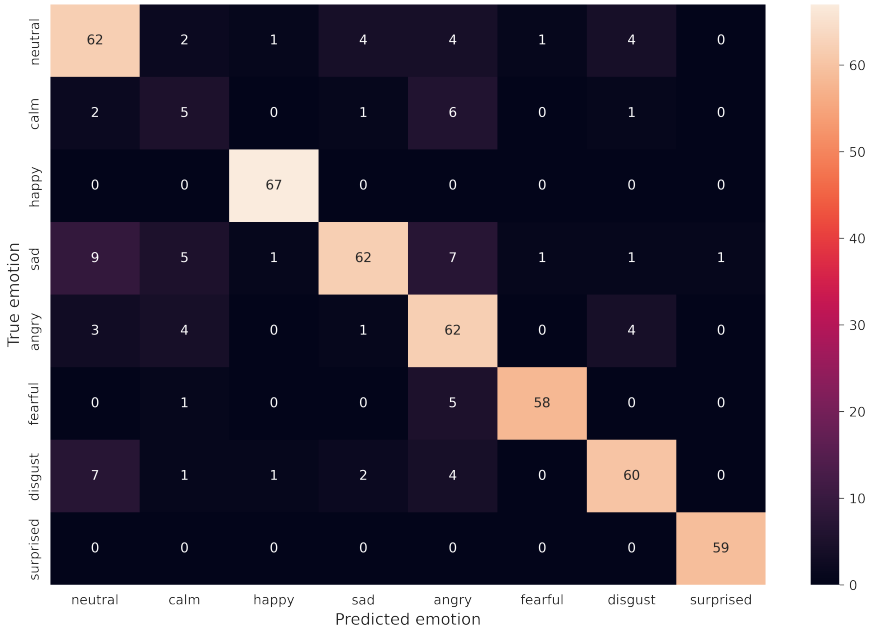

Figure 16: Hypertuned Correlation Matrix

	precision	recall	f1-score	support
0	0.75	0.79	0.77	78
1	0.28	0.33	0.30	15
2	0.96	1.00	0.98	67
3	0.89	0.71	0.79	87
4	0.70	0.84	0.77	74
5	0.97	0.91	0.94	64
6	0.86	0.80	0.83	75
7	0.98	1.00	0.99	59
accuracy			0.84	519
macro avg	0.80	0.80	0.80	519
weighted avg	0.85	0.84	0.84	519

Figure 17: HyperTuned Model Classification Report

Layer	Hyperparameter	Before Tuning	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
Conv_Filters_1	Number of Filters	32	256	128	128	128	128
Conv_Activation_1	Activation Function	relu	sigmoid	relu	relu	sigmoid	sigmoid
Dropout_Rate_1	Dropout Rate	0.5	0.1	0.25	0.05	0.05	0.05
Conv_Activation_2	Activation Function	relu	sigmoid	relu	sigmoid	sigmoid	relu
Adam_Optim_LR	Optimizer	0.01	0.000578	0.000803	0.00102	0.000307	0.000528
Score	Score	0.866	0.886	0.888	0.896	0.898	0.898

Figure 18: Baseline vs HyperParameter

5 Conclusion

In this project we were able analysed audio files to get more information about our dataset, extract various features from the audio files (10914) using MFCC,Chrome and Mel to proceeded to train a Convolutional Neural Networks (CNN) with those extracted features. Furthermore, we optimised our hyper parameters using keras random search and got the the best fit for our model.

References

[Baram 2021] Tal Baram. *Classifying emotions using audio recordings and python*, Mar 2021.

[Dinoanasta] Dinoanasta. *Fashion*$_M NIST_C NN/Fashion_M NIST.ipynbatmasterdinoanasta/FASH$

[kdn] *Audio data analysis using Deep Learning with python (part 1)*.

[Livingstone 2019] Steven R. Livingstone. *Ravdess emotional speech audio*, Jan 2019.

[Lok 2019] Eu Jin Lok. *Toronto emotional speech set (tess)*, Aug 2019.

[Nandi 2021] Papia Nandi. *CNNs for Audio Classification*, Dec 2021.

[Pykes 2022] Kurtis Pykes. *Improving the accuracy of your neural network*, Feb 2022.