Speech Emotion Recognition

IS460

Final Report

Authors

Tan Zuyi Joey
Bodine Salomi Marie-Louise Stubbe
Shambhavi Goenka
Darryl Soh Soon Yong
Chung Zhi Huai
Elston Eng Shi Yang

Group number: 2

Date handed in: November 7th 2023

Contents

1 Abstract

This research project delves into the intricate domain of Speech Emotion Detection, aiming to unravel emotional cues from audio data. It explores various modeling techniques, encompassing deep learning, classical machine learning, and feature selection approaches, to construct robust emotion recognition systems.

Among the models developed, a baseline model using Support Vector Machines (SVM) serves as a comparison standard. SVM, a classical machine learning method, was implemented with a focus on distinctly classifying data points in a multidimensional space. Extensive experimentation was performed, including GridSearch with multiple kernel options. The SVM model was trained on the speech dataset, forming a benchmark for the subsequent models.

To address the challenge of high-dimensional feature space, the feasibility of Principal Component Analysis (PCA) for feature reduction was explored. PCA was applied to the standardized data, yielding a reduced set of principal components while maintaining a desired explained variance. However, subsequent evaluations revealed a slight decrease in model performance, prompting the decision to prioritize a more comprehensive feature set over processing time.

2 Introduction

2.1 Problem Statement

Emotions are pivotal in communication, especially for humans, where subtle tonal variations convey a spectrum of emotions, from joy to anger. Decoding these emotional cues is crucial for compelling interactions and holds promise in domains like healthcare, customer service, and entertainment. Machine Learning has emerged as a powerful tool for unraveling these emotional signals in speech and audio data. This project delves into Emotional Speech Detection, also known as Speech Emotion Recognition (SER).

SER is the attempt to recognize human emotion and affective states from speech. Typically a classification problem, SER capitalizes on the fact that voice reflects underlying feelings through tone and pitch. Through this, emotion is classified into a few predefined categories. Notably, this is also the phenomenon that animals (e.g., dogs and horses) utilize to understand human emotions. This type of speech recognition is gaining popularity, and the demand for it is increasing. Researchers work by treating audio signals as time-series data or using spectrograms to generate numeric and image representations of audio. Due to this transformation, there is likely some feature loss.

For instance, SER can be used in call centers to classify calls according to emotions. The staff in call centers can utilize it as a performance parameter for conversational analysis. This allows them to identify unsatisfied or satisfied callers, helping companies improve their services. The challenge lies in accurately and automatically classifying emotions from speech and audio data. Given the complex and subjective nature of human emotions, emotion detection poses a significant challenge. Our goal is to construct a robust system that can categorize speech samples into various emotional states like happiness, sadness, anger, fear, and more. The primary objective is to develop a highly accurate machine-learning model capable of discerning these emotional nuances.

So, how does speech emotion recognition work? Converting audio signals into numeric or vector format is not as straightforward as with images. The transformation method determines how much information is retained when one abandons the "audio" format. Emotion expressed through audio includes many subtleties, and if these are lost in translation, it is hard for the classifier to learn the nuances between different emotions. Typical methods include:

- Transforming audio into images using Mel spectrograms. Audio is visualized into signals based on frequency components and is plotted as an audio wave, then fed to a CNN as an image classifier. This is captured using Mel Frequency Cepstral Coefficients.

- Speech-to-text conversion is more complex since words need to be mapped to text. Long Term Short Term Memory (LSTM) and transformer models have propelled research in this field to incredible success, as subtitles or audio transcripts are available on almost every video streaming service.

2.2 Motivation

The motivation for this project lies in the following:

- **Enhancing Human-Machine Interaction**: As we move towards a future where human-machine interaction becomes more common, the ability to understand and respond to human emotions is critical. Emotional speech and audio classification can significantly improve chatbots, virtual assistants, and customer service systems, making them more empathetic and user-friendly.

- **Facilitating Mental Health Support**: Emotion detection can have a profound impact on mental health support systems. By analyzing audio data, we can potentially identify individuals who may be experiencing emotional distress or depression, allowing for early intervention and support.

- **Optimising Content Recommendations**: In the realm of entertainment and content consumption, understanding the emotional state of users can lead to more personalized recommendations. It can help platforms suggest movies, music, or content that aligns with the user's current emotional state.
- **Enhancing Market Research and Customer Feedback**: Businesses can benefit from emotion detection in market research and customer feedback analysis. By analyzing customer service calls, for example, companies can gauge customer satisfaction and identify areas for improvement.

We should care about emotional speech detection because it bridges the gap between human emotions and technology. In an increasingly digital world, interactions are often mediated by machines. The ability of AI systems to understand and respond to our emotions can make technology more humane and user-centric.

Furthermore, this problem is inherently interesting due to its interdisciplinary nature. It combines elements of linguistics, psychology, signal processing, and machine learning. Solving it not only advances the field of artificial intelligence but also offers insights into the intricate relationships between linguistics, psychology, signal processing, and machine learning. It sheds light on the complexity of human emotions, which are a fundamental aspect of the human experience.

3 Literature Review

As previously mentioned, Speech Emotion Recognition (SER) is a system whereby emotion can be recognized from audio samples. (Raval, 2023) In particular, Speech emotion recognition is a sub-branch of Automatic Emotion Recognition (AER), which is the process of identifying human emotion from signals such as speech, text, and facial expressions. As a result, it shares several similarities to problems in the domain of Natural Language Processing, such as sentiment analysis.

Like sentiment analysis, SER is a classification problem, where an input (audio) is to be classified into one of several predefined emotions. (Raval, 2023) As such, a typical pipeline for SER is similar to that of sentiment analysis or any classification problem; first feature extraction is performed, relevant features are selected, and finally an appropriate classifier is employed.

As for feature extraction, audio can either be treated as image data in the form of spectrograms or as time series data with the value of amplitude at each timestep. For the latter, some of the most commonly used features are (de Lope & Graña, 2023):

- Zero Crossing Rate: The rate at which a signal changes sign. It provides information about the number of times the signal crosses the horizontal access of amplitude.

- Energy: Energy is defined as the size of the amplitude over a given time period. Its calculation involves squaring the amplitude values of the sound wave in the time frame. (de Lope & Graña, 2023)

- Entropy of Energy: The entropy of energy of a sound wave measures the disorder in the distribution of the sound wave (signal), making it useful to assess the variability of energy within the sound wave.(Burnwal, 2020)

- Spectral Centroid: The spectral centroid indicates where the center of mass of an audio wave is located, thereby giving information about the location of 'brightness' or elevated sound in the sound wave.(Burnwal, 2020)

- Spectral Bandwidth: Spectral bandwidth measures the range of frequencies where most of the energy of the sound wave is located.(Burnwal, 2020)

- Spectral Spread: The spectral spread provides a measure for the width of the sound wave with respect to the frequency range.(Burnwal, 2020)(de Lope & Graña, 2023)

- Spectral contrast: Spectral contrast is a measure that helps to identify the difference in the maximum and minimum energies of the sound wave. More generally, it is a measure of variation of "loudness" of sound. (Burnwal, 2020)

- Spectral Entropy: Whereas the entropy of energy provides a measure of the variability of energy within the sound wave, the spectral entropy provides a measure for noisiness or complexity of the amplitude. (Burnwal, 2020)

- Spectral Flux: The spectral flux is used to measure abrupt changes in an audio signal.(Burnwal, 2020) (de Lope & Graña, 2023)

- Spectral Rolloff: The spectral roll off allows one to find the cut-off point where the higher-pitched sound starts to become less prominent. As such, it is a way to measure how sharp or dull a sound is using its high-pitched parts. (Burnwal, 2020)

- MFCCs: Mel Frequency Cepstral Coefficients are one of the most commonly used features in speech emotion recognition, as they act as "audio fingerprints", helping computers understand sound. They are, as the name suggests, a set of coefficients that capture the most important features of the sound wave and how it varies over time. (Burnwal, 2020) (de Lope & Graña, 2023)

- Chroma STFT: The Chroma STFT measures the chroma value of sound, which encapsulates the information of the twelve different pitch classes in sound. (Das, Ghosh, Pal, Dutta, & Chakrabarty, 2020)

- Chroma Vector: Chroma vectors are numeric vectors that describe the harmonic content of sound. (Burnwal, 2020)
- Chroma Deviation: Chroma deviation provides a measure of how much the sound deviates from an expected or standard chroma vector. (Burnwal, 2020)

These features are typically extracted manually from sound waves through specific feature extraction measures. In addition, global statistics are applied to such features, like the maximum, minimum, or range, to capture patterns across the full interval of measurement (time). (de Lope & Graña, 2023)

For example, Gao et al (2017) use a variety of global and local features in their work "Speech emotion recognition using local and global features". They extract various prosodic and spectral features, including, pitch, LSP, ZCR, intensity and MFCC, from two audio databases, among which is the RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset, to which they apply global statistics. These global statistics are fed into an SVM classifier with a linear kernel, achieving an average accuracy of 79.4% on the cross-validation set of RAVDESS. (Gao, Li, Wang, & Zhu, 2017)

The wide range of features employed in SER and their effect on the classification task is further explored by Ramakrishnan et al. Like Gao et al, they also employ global statistics and apply it to a wide variety of features and also feed it into a SVM. They find the best results with the MFCC and Pitch features, yielding an F1 value for the classifier between 0.60 and 0.84 for different emotions. (Ramakrishnan & El Emary, 2011)

The use of SVM in both papers is illustrative of the popularity of the classifier in the research domain of SER. Part of that can be explained by the fact that global features require actually the use of a classification algorithm like SVM. (Ramakrishnan & El Emary, 2011)

Apart from classical machine learning methods, the use of deep learning is also becoming more popular in SER.(de Lope & Graña, 2023) One of the reasons deep learning is gaining popularity is because of the automatic feature extraction, which reduces the chances of information loss. The features previously mentioned require manual extraction and applying global statistics are extra steps of processing in which information loss could occur.(Raval, 2023)

Retaining information from the data is particularly important in SER. Emotions are incredibly subtle and so a classifier can only pick up on subtleties between different emotions if enough information is kept in the data.

For example, Badsah et al (2017), treat the audio as image data and feed spectrograms into a Convolutional Neural Network. They allow the network to extract the most noticeable discriminative features in the three convolutional layers. The output is then fed into 3 fully connected layers, followed by a softmax classification layer. The overall method has an accuracy of 84.3%, exceeding some of the accuracies achieved by manually extracted features and SVM classifiers. (Badshah, Ahmad, Rahim, & Baik, 2017)

Other than for the automatic feature extraction, deep learning is also becoming more popular because it seems to yield very high performance compared to some classical machine learning algorithms. For example, de Pinto et al still use manual feature extraction but employ a deep learning method. They extract the MFCC of the audio in the RAVDESS data and feed it into a convolutional neural network. Performance is very promising, with an average F1 score of 0.91 on the test set. They use two convolutional layers, two dropout layers, max pooling and one fully connected layer. (de Pinto, Polignano, Lops, & Semeraro, 2020)

Considering the aforementioned research, it would be interesting to further investigate the role deep learning can play in creating a well-performing classifier. In addition, investigation into what features contribute most to accurate predictions is another topic that asks for more attention.

4 Data

4.1 Source

The dataset utilized for this study is the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), which was obtained from Kaggle, providing a valuable resource for research in Speech Emotion Recognition (SER) (LivingStone, 2018) (LivingStone & Russo, 2018). RAVDESS features a diverse set of 24 professional actors, equally split between genders (12 male and 12 female). These actors were tasked with vocalizing two lexically matched statements, "Kids are talking by the door" and "Dogs are sitting by the door," in both spoken and singing formats. These statements were delivered at two different intensity levels: normal and strong. Importantly, actors conveyed these statements while expressing eight distinct emotions: calm, happy, sad, angry, fearful, surprised, disgusted, and neutral.

The RAVDESS dataset comprises a total of 1440 unique audio files, each representing a specific combination of actor, statement, intensity, and emotion. This large dataset was constructed from 60 trials for each actor. Consequently, the dataset's structure can be summarized as 24 actors multiplied by 60 trials, resulting in the comprehensive collection of 1440 audio files.

This dataset's richness and diversity, encompassing various emotions, vocalization formats, and intensity levels, make it a valuable resource for the development and evaluation of SER models. Researchers can leverage this dataset to investigate the nuances of emotional expression in speech and assess the performance of different feature extraction and classification techniques in recognizing emotions from audio data.

The data set' files were split into speech and song, as to create two separate data sets. First, the focus of this project was on the speech data set. In later models, the song data set was added as well in an attempt to boost performance.

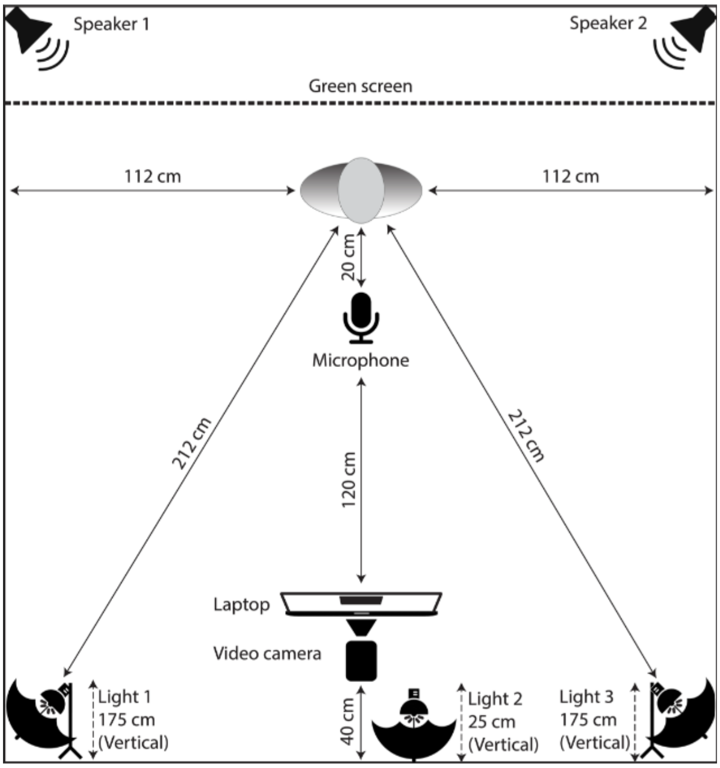


Figure 1: Set Up of voice recordings in RAVDESS dataset

Figure 1 shows the setup that was used to make the voice recordings. The link to the dataset is: `https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio`

Attributes	Details
Modality	01 = Full -audio Video 02 = Video- only 03 = Audio - only
Emotion	01 = Neutral 02 = Calm 03 = Happy 04 = Sad 05 = Angry 06 = Fearful 07 = Disgust 08 = Surprised
Emotional Intensity	01 = Normal 02 = Strong
Statement	01 = ”Kids are talking by the door” 02 = Dogs are sitting by the door”
Actor Index	01 to 24 Odd-numbered actors are male Even-numbered actors are female

Table 1: Attributes of the data and key identifiers

Table 1 shows the attributes in the given data set, labeling each record file with the key identifiers. In our project, only the Audio files (modality =03) were used.

4.2 Data Processing

Following the downloading of the data set, the data was processed and exploratory data analysis was performed.

4.3 Exploratory Data Analysis

4.3.1 Data Distribution

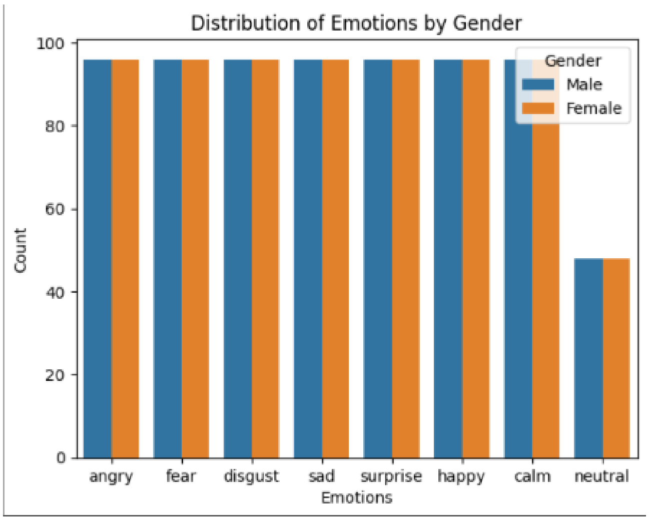


Figure 2: Data distribution per gender

Figure 2 shows the data distribution per gender. As shown, the number of data files per gender is equal for each emotion, since an equal number of male and female actors were used to record the data. Furthermore, one can see that the number of data records for the emotion neutral is half than that of the other emotions. This can be explained by the fact that neutral is used as a baseline emotion and so it was recorded at only one intensity.

4.3.2 Waveforms and Spectrograms

Following the inspection of the data distribution, the data was handled to produce amplitude waveforms and spectrograms for each respective emotion. As aforementioned, producing waveforms treats the audio data as time series, whereas producing spectrograms converts the audio data to image data.

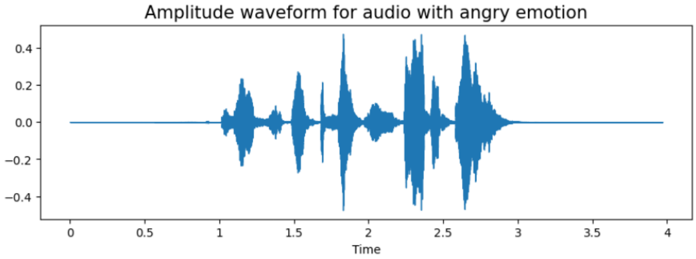


Figure 3: Amplitude waveform over time for angry emotion

Figure 3 is an example of the waveforms generated. It shows the amplitude waveform of the emotion ”angry” over time.

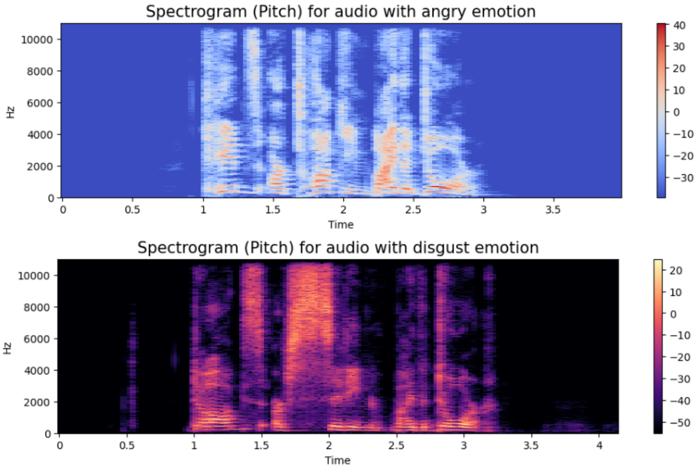


Figure 4: Spectrograms for anger and disgust for an arbitrary actor

Figure 4 shows the spectrogram for the angry and disgust emotion for an arbitrary actor. Upon analysis of the spectrograms it was noticeable that for anger and fear the spectrograms had less dark blue/black tones and less bright yellow and red tones than the the spectrograms of the other emotions. This contrast is exemplified by the spectrograms for anger and disgust in figure 4. Bright red and yellow colors indicate high amplitude or more energy in the sound, whereas dark blue and black colors indicate low amplitude or low energy. As such, for anger and fear, there is less pronounced high and low energy compared to the other emotions.

5 Methodology

5.1 Data Augmentation

Following the initial exploratory data analysis, and prior to the feature extraction and selection, data augmentation was performed. Data augmentation is a technique to increase the size of the data set by generating new data samples through slight modification of the original data set. (Awan, 2022) In doing so, one aims to prevent over-fitting and improve model performance. This is particularly important when neural networks are to be used because they require large amounts of data to yield good generalization capabilities. (Awan, 2022) Furthermore, particularly for sound classification, it is considered standard practice to augment the data.

Some of the most common augmentation techniques for sound data include noise injection, stretching, and pitching. (Monigatti, 2023) Noise injection is the addition of static data to the sound. Stretching literally stretches the sound wave, thereby increasing time duration and causing a "slow motion" sound effect. Pitching involves accentuating the high-pitch notes or the peaks and valleys on a sound wave.

To create the augmented data set, all three augmentation techniques were applied separately to the original data set. As such, one was left with the original data set, the data set with noise, the data set where sound was stretched, and the data set on which pitching was applied. To each of these four data sets, feature extraction was applied. Afterward, the features from all four data sets were pooled. In the end, this generated the final data set to train and test the model.

5.2 Data Preparation and Preprocessing

5.2.1 Data Loading

Data was loaded into a Pandas DataFrame from a CSV file, providing an initial view of the dataset's structure. The DataFrame's first few rows were examined using the `head()` method to ensure correct loading and to understand the dataset's format.

5.2.2 Feature Extraction

Features were extracted from the raw data, representing various aspects relevant to the sound classification task. The dataset included both spectral and temporal features, crucial for effective machine learning models in audio analysis.

5.2.3 Target Variable Encoding

The target variable, representing the class labels, was encoded using one-hot encoding. This step transformed the categorical labels into a binary matrix representation, essential for multiclass classification problems.

5.2.4 Data Splitting

The dataset was split into training and testing sets using the `train_test_split` function from Scikit-learn, ensuring a random and unbiased division of data. This split facilitated the evaluation of the model's performance on unseen data.

5.2.5 Feature Scaling and Normalization

Feature scaling was performed using Scikit-learn's `StandardScaler` to standardize the features. Additionally, `Normalizer` was applied for L2 normalization, ensuring that the scale of each feature was appropriate for the modeling process.

5.3 Model Architecture and Compilation

5.3.1 Sequential Model Construction

A Sequential model was constructed using Keras, starting with a 1D convolutional layer, followed by activation, dropout, and flattening layers. The model architecture was designed to effectively capture the temporal dynamics of the audio data.

5.3.2 Regularization Techniques

L1 and L2 regularizations were employed in the convolutional and dense layers to mitigate overfitting, ensuring that the model generalizes well to new data.

5.3.3 Activation Functions and Pooling

The 'relu' activation function was utilized for its efficiency in non-linear transformations. Global Average Pooling was chosen for its ability to reduce model parameters and computational cost. The output layer employed a softmax activation to output probability distributions for the multiclass classification.

5.3.4 Model Compilation

The model was compiled with the Adam optimizer, known for its effectiveness in handling sparse gradients. The loss function was set to 'categorical_crossentropy', suitable for multiclass classification tasks. Accuracy was chosen as the metric for evaluating model performance.

5.4 Model Training

The model was trained over 300 epochs with a batch size of 350. Validation data was used during training to monitor the model's performance. The `ReduceLROnPlateau` callback was implemented to adjust the learning rate based on the loss metric, enhancing the training process.

5.5 Model Evaluation and Results

5.5.1 Training and Testing Accuracy and Loss

Plots of training and testing accuracy and loss were generated to visualize the model's learning trajectory across epochs. This graphical representation provided insights into the model's learning efficiency and convergence behavior.

5.5.2 Prediction on Test Data

The trained model was used to make predictions on the test data. The predictions were then inversely transformed to retrieve the original class labels, facilitating an understandable comparison between predicted and actual values.

5.5.3 Confusion Matrix and Classification Report

A confusion matrix and classification report were generated.

6 Feature Extraction and Selection

6.1 Feature Extraction

6.1.1 Features Extracted

As aforementioned in the literature review, there are several features that can be extracted from sound data. For this particular project, the following features were extracted using the librosa library:

- Zero Crossing Rate
- Chroma STFT
- MFCC
- Root Mean Square Error
- MelSpectograms
- Spectral centroid
- Spectral Bandwidth
- Spectral Contrast
- Spectral Rolloff
- Chroma Deviation

Features were extracted using 0.5 second frames. Following extraction, the number of features was 1104.

6.1.2 Standardization

Feature extraction transformed the sound data to a numeric format. However, the numeric range of the respective features differed widely. As a result, following feature extraction, the data was standardized through Z score normalization.

6.2 Train Test Split

Following all necessary data processing steps, separated into a train and test set. 80% Of the data was used for training and 20% for testing. A random state of 10 was used to split the data.

7 Modelling

Following the data processing and feature extraction, different models were developed.

Because of the rise of popularity of deep learning models, in particular Convolutional Neural Networks (CNNs), it was decided to use such techniques as well in this project. In particular, several CNN's were developed as well as a multilayer perceptron (MLP).

However, to provide a comparison standard and not neglect the power of classical machine learning methods, a baseline model was developed using Support Vector Machines (SVM).

7.1 Baseline model

7.1.1 SVM

The SVM was implemented using Sklearn's SVC with a default value for C=1.0. The main goal of SVM is to distinctly classify the data points by finding a hyperplane in the N-dimensional space, where N refers to the number of features. The objective function is to minimise the difference in distance between the hyperplane and the data points that should be linearly separable.

Furthermore, a Gridsearch was employed on the best kernel with a four time cross validation on the training data set. The candidate HyperParameters selected were 'rbf', 'linear', 'poly' and 'sigmoid'. Finally, the GridSearch returned 'linear'as the optimal hyperparameter for the kernel.

The SVM was trained on the speech data set.

7.1.2 Feature Selection: Exploring PCA

Following the implementation of the SVM, which took a significant amount of time with 1104 features, the possibility of using PCA in order to reduce the number of features and thus possibly boost performance was explored.

Using a 0.98 variance threshold, PCA was performed on the standardized data.

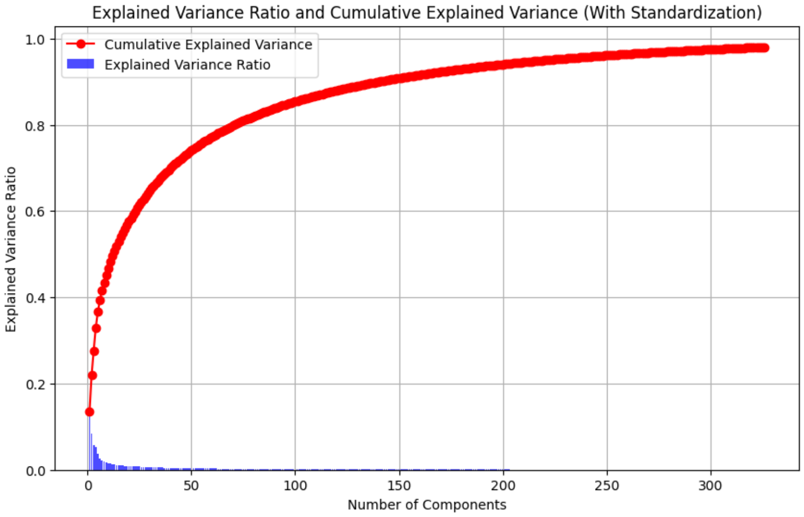


Figure 5: Explained Variance Ration and Cumulative Explained Variance

As shown in 5, following PCA, the total number of principal components to satisfy the desired explained variance was 326. Although 326 features in absolute terms is still a lot, it it is significantly less than the previous 1104 features.

To test the applicability and usefulness of PCA in this project, another SVM model was built using the 326 principal components generated by PCA. The same procedure to build the SVM model with the original data was performed in order build the SVM model with the principal components. As such, SVM was implemented using SKlearn's SVC with a default value of C=1.0 and a gridsearch was performed to find the best kernel with a four time cross validation on the training data.

Notably, the processing time of SVM with the reduced number of components was significantly shorter. However, upon comparison of SVM model with the original data and the principal components, there was a decrease across almost all evaluation metrics by 0.2. As such, since the goal of the project was to build a model as robust possible, it was chosen to prefer a longer processing time with better performance than a shorter processing time with a slightly worse performance. Therefore, the idea to use PCA for data reduction was rejected and the principal components acquired from PCA were not used in any further models.

7.2 Deep Learning Models

7.2.1 MLP

The MLP was implemented using SKlearn's MLPClassifier. For the number of hidden layers, SKlearn's default of 100 layers was used. Furthermore, ReLu was used as an activation function, and Adam as a solver. A GridSearch was performed on the learning rate, again with a four-time cross-validation on the training data set. The candidate Hyper Parmaters selected were 'constant', 'invscaling' and 'adaptive'. It was found that an adaptive learning rate would yield the best results. The MLP was trained on the speech data set.

7.2.2 CNN

Using Convolutional Neural Networks 9 different models were developed. They were developed progressively, meaning subsequent models were changed in structure or parameters were tuned all in an attempt to boost performance and find the best one.

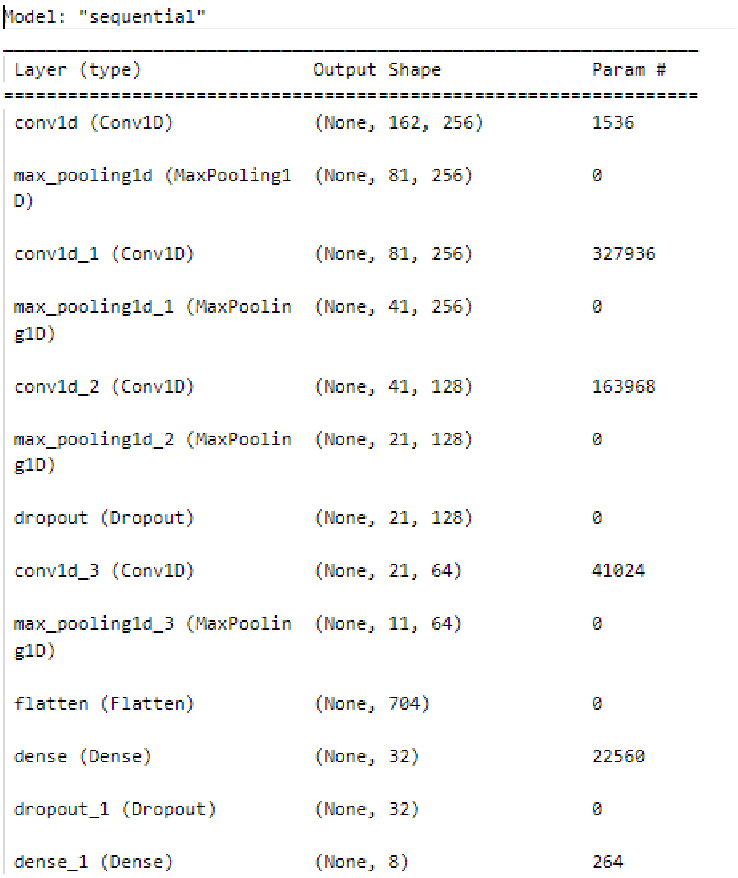
Some of the earlier models were trained on the speech data set, whereas some of the later models were trained on both the speech and the song data set in an attempt to boost performance.

CNN ID	Data sets used	Number of features used	Number of 1D Convolutional Layers	Frame time
1	Speech	5	4	0.5
2	Speech	10	4	0.5
3	Speech	10	1	0.5
4	Speech	1	1	0.5
5	Speech	10	1	0.5
6	Speech	10	1	0.5
7	Speech	10	1	0.5
8	Speech + Song	10	1	0.5
9	Speech + Song	10	1	0.2

Table 2: Overview of CNN models developed

The overview of all models developed is given in figure 2. The frame time refers to the time for the frames used when extracting the features. As outlined earlier in feature extraction, first, 0.5 second frames were used. However, to explore the effect of shorter frames, in the last model a smaller frame time was used.

CNN 1



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv1d (Conv1D)             (None, 162, 256)          1536      
                                                                 
 max_pooling1d (MaxPooling1  (None, 81, 256)           0         
 D)                                                              
                                                                 
 conv1d_1 (Conv1D)           (None, 81, 256)           327936    
                                                                 
 max_pooling1d_1 (MaxPoolin  (None, 41, 256)           0         
 g1D)                                                            
                                                                 
 conv1d_2 (Conv1D)           (None, 41, 128)           163968    
                                                                 
 max_pooling1d_2 (MaxPoolin  (None, 21, 128)           0         
 g1D)                                                            
                                                                 
 dropout (Dropout)           (None, 21, 128)           0         
                                                                 
 conv1d_3 (Conv1D)           (None, 21, 64)            41024     
                                                                 
 max_pooling1d_3 (MaxPoolin  (None, 11, 64)            0         
 g1D)                                                            
                                                                 
 flatten (Flatten)           (None, 704)               0         
                                                                 
 dense (Dense)               (None, 32)                22560     
                                                                 
 dropout_1 (Dropout)         (None, 32)                0         
                                                                 
 dense_1 (Dense)             (None, 8)                 264       
```

Figure 6: CNN model 1 architecture

The first model, inspired by (Burnwal, 2020), follows the architecture depicted in Figure 6. The model comprises the following layers and configurations:

- **Convolutional Layers**: Four convolutional layers were employed, each with the following specifications:
  - Layer 1: 256 filters, a kernel size of 5, and a stride length of 1.
  - Layer 2: 256 filters, a kernel size of 5, and a stride length of 1.
  - Layer 3: 128 filters, a kernel size of 5, and a stride length of 1.
  - Layer 4: 64 filters, a kernel size of 5, and a stride length of 1.
  - Activation function: ReLU.
- **MaxPooling Layers**: Following each convolutional layer, a MaxPooling layer was applied.

- **Dropout Layers**: A dropout layer with a probability of p=0.3 was utilized after the third convolutional layer.

- **Fully Connected Layers**: After the convolutional layers, a fully connected layer with 32 units was added, followed by another dropout layer with a dropout probability of p=0.2. The activation function was ReLu for the fully connected layer.

- **Output Layer**: The final fully connected layer consisted of 8 units with softmax activation, suitable for the multiclass classification problem at hand.

The model was compiled with Adam as optimizer with a learning rate of 0.001, categorical cross-entropy as loss function and accuracy as metric to monitor the learning. It was trained with a batch size of 64, for 50 epochs. Learning rate was reduced when the loss function did not improve.

As shown in figure 2, for this model only five features were used, as inspired by (Burnwal, 2020). The features were: Zero Crossing Rate, Chroma STFT, MFCC, Root Mean Square value and MelSpectrogram.

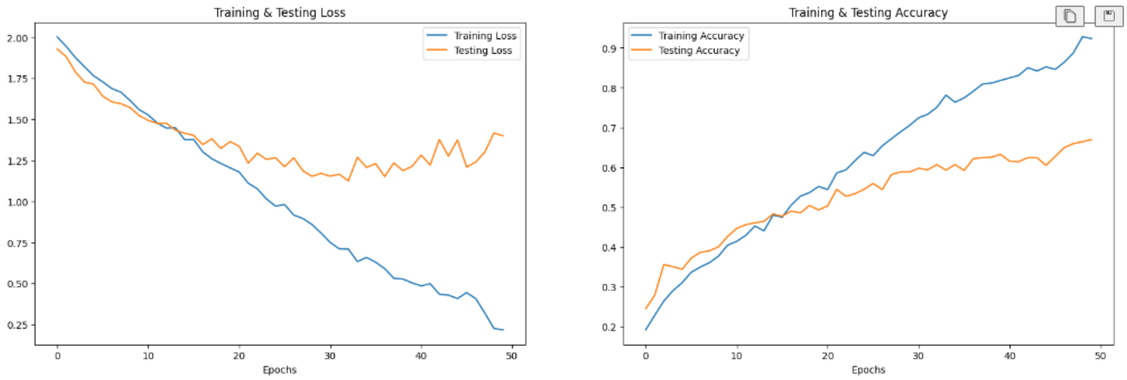


Figure 7: CNN model 1 Loss and Accuracy curves of the training and test set while training the model

Figure 7 shows the Loss and Accuracy curves on the training and test set as the CNN1 model trains. It is possible to see that there is some significant overfitting that occurs at about 25 epochs, as the loss curve of the test starts to exceed the training loss curve, and the accuracy curve of the training set overtakes that of the test set.

CNN 2

The second model develops onto the first model. The same architecture and configuration of layers as in CNN1 are used. However now instead of five features, all 10 features as outlined in section 6.1 are used.

In addition, all convolutional layers are followed by a dropout layer with probability p=0.3, instead of only the third convolutional layer as in CNN1. This in an attempt to reduce the overfitting seen as seen in figure 7. In addition, batch size was increased to 512 and it was trained for 100 epochs instead of 50.

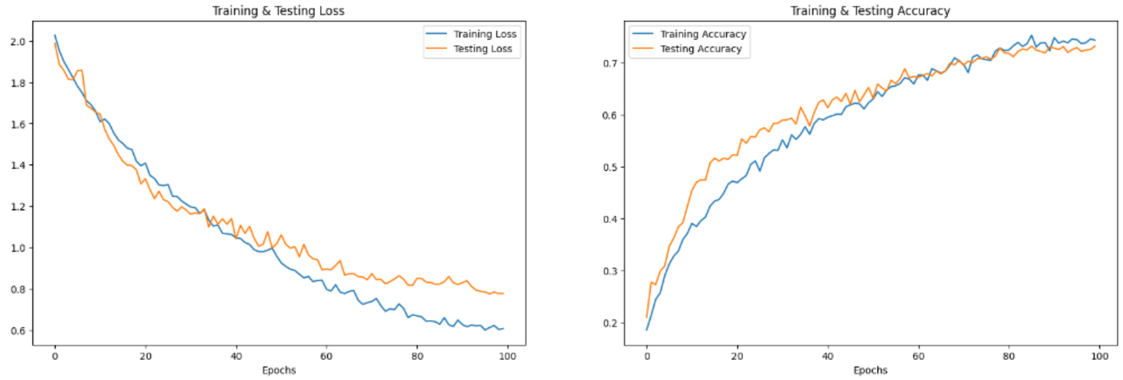


Figure 8: CNN model 2 Loss and Accuracy curves for on the training and test set while training the model

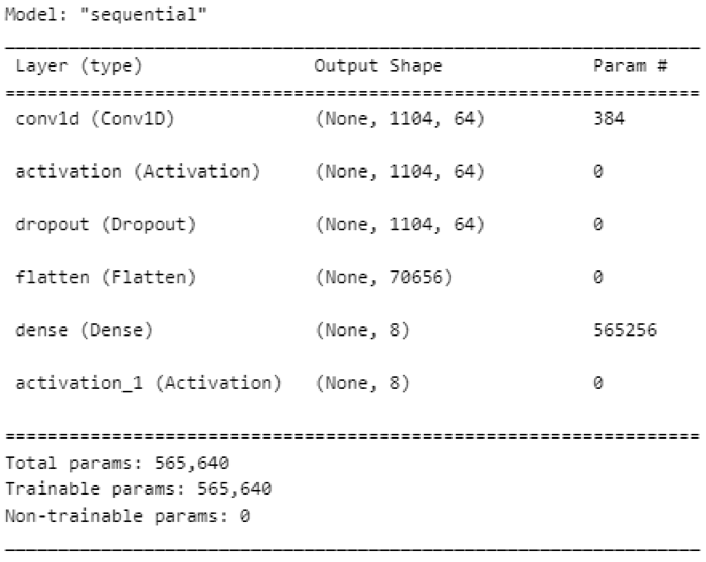
Figure 8 shows the Loss and Accuracy curves on the training and test set as the CNN2 model trains. It is possible to see that with the addition of the extra features as well as the extra dropout layers, overfitting has significantly decreased. It is well known that dropouts can help overfitting since neurons cannot rely on input, since it might be dropped. As a result, bias is decreased. (Ndiritu, 2021)

CNN 3 Because the first two models showed somewhat unsatisfactory results, it was decided to change up the architecture of the model. Specifically, inspired by the great results that de Pinto et al achieved on their model with a single convolutional layer (F1 of 0.91), the third model built onto their architecture.

Taking inspiration from his model, an adjusted model was developed and the architecture is shown in 9.

Specifically, the model has the following configuration:

- **Convolutional Layers**: One convolutional layer was used, with the following specifications.
  - 64 filters, a kernel size of 5, and a stride length of 1.
  - L1 regularization on the kernel with a value of 0.001
  - L2 regularization on the kernel with a value of 0.001
  - Activation function: ReLU.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 1104, 64)          384

 activation (Activation)     (None, 1104, 64)          0

 dropout (Dropout)           (None, 1104, 64)          0

 flatten (Flatten)           (None, 70656)             0

 dense (Dense)               (None, 8)                 565256

 activation_1 (Activation)   (None, 8)                 0

=================================================================
Total params: 565,640
Trainable params: 565,640
Non-trainable params: 0
_________________________________________________________________
```

Figure 9: CNN model 3 architecture

- **Dropout Layers**: A dropout layer with a probability of p=0.3 was utilized after the convolutional layer.
- **Output Layer**: The final fully layer was a fully connected layer consisting of 8 units with softmax activation, suitable for the multiclass classification problem at hand.

All features were used. Kernel regularisation was implemented to manage and reduce over fitting.

The model was compiled with the same solver, loss function and used the same evaluation metric for training as model 1 and 2. It was trained with a batch size of 512 for 100 epochs. Learning rate was reduced when the loss function did not improve.

While trying a new model, figure 10 shows the difficulty the model has while learning. Not only does the model suffer from overfitting as shown by the training accuracy curve that is significantly higher than that of the testing set, the loss curve of the testing set does not decrease at all. In fact, it almost immediately increases, which shows the poor generalization capability of the model.

CNN 4

The fourth model is a variation of model 3. Since CNN 3 had such difficulty learning, one tried to use (de Pinto et al., 2020) methodology, which was to only include MFCC is used as an input feature. In addition, no regularization is used in the

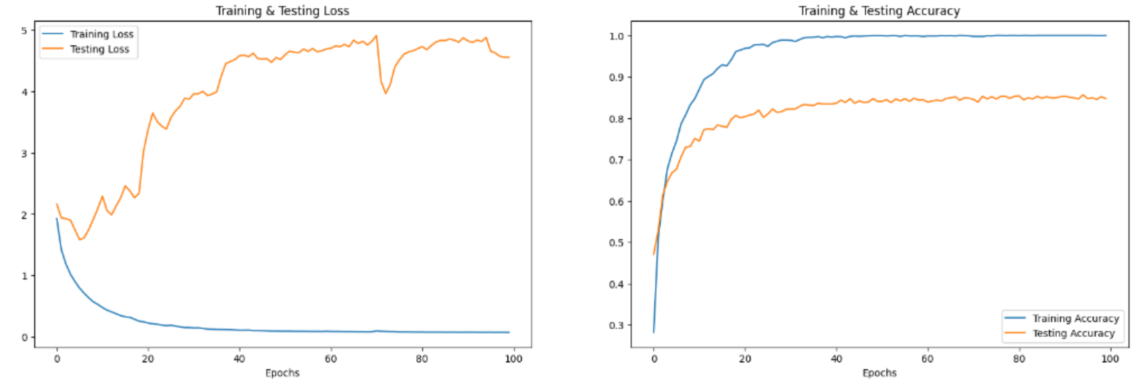


Figure 10: CNN model 3 Loss and Accuracy curves for on the training and test set while training the model

convolutional layer, to see if it would help stabilise the loss function for performance on the test set.

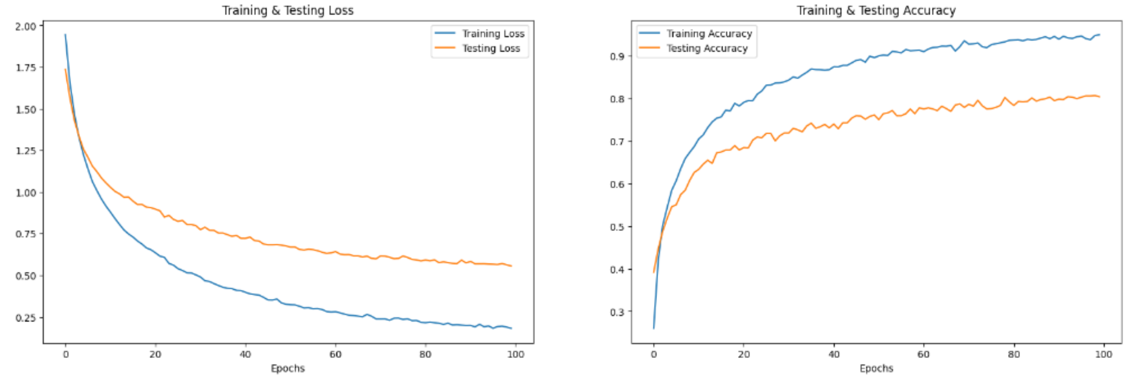


Figure 11: CNN model 4 Loss and Accuracy curves for on the training and test set while training the model

The resulting loss and acccuracy curves are shown in 11. Clearly, the model learns better with just MFCC as input feature, instead of all 10. Nevertheless, there is still some overfitting. In addition, upon preliminary results, evaluation metrics actually decreased, which could be because of the reduction of features used.

CNN 5

The fifth model builds on the fourth. The architecture is the same, except that kernel regularization has been added to the final dense layer. The values for regularisation are 0.001 and 0.001 for L1 and L2 regularisation respectively, to see if it would improve the overfitting. Furthermore, the training batch size has been decreased 500 and training epochs have been increased to 300. In addition, all features are used again, since the use of just one feature in CNN4 did not help performance across the evaluation metrics.

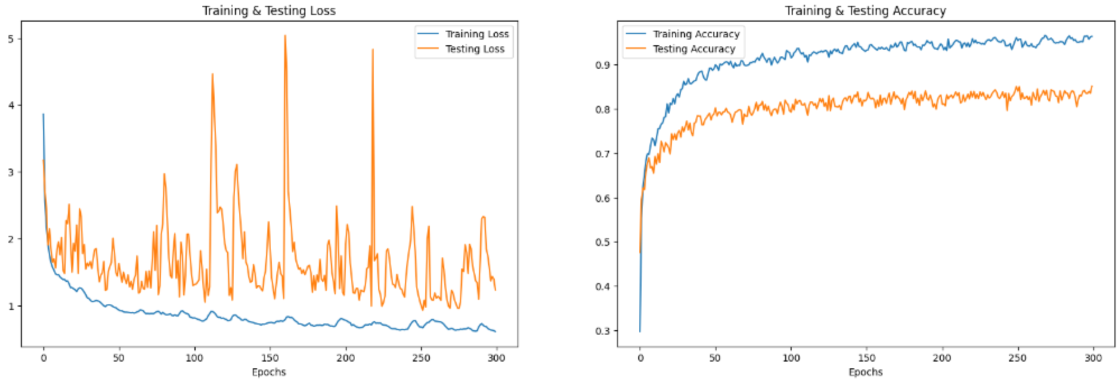


Figure 12: CNN model 5 Loss and Accuracy curves for on the training and test set while training the model

The resulting loss and accuracy curves are shown in 12. Although the accuracy curves are a bit closer together, one can see that the testing loss curve fluctuates extremely, an indication of instablity on the test set.

CNN 6

The sixth model builds on the fifth model. Regularisation on the kernel has been added to the convolution layer again, with values of 0.0001 and 0.0001 for L1 and L2 regularisation respectively. This in attempt to further reduce over fitting and see what effect it would have on the stability of the testing loss curve. In addition, the L1 regularisation in the dense value has been reduced from 0.001 to 0.0005.

Batchsize was reduced from 500 to 350, whilst still training on 300 epochs.

The resulting loss and accuracy curves are shown in 13. One can see that the loss curve of the test set is less volatile than for CNN5, which is an improvement.

CNN 7

CNN seven builds on the sixth model, adding in BatchNormalization layers after the flattening operation and after the last fully connected layer before activation. Batch normalization was added because in model five and six the loss curve fluctuated significantly for the testing data set, as shown in figure 12 and figure 13. Bath Normalization can help stabilize the training and learning of the model by scaling and re-centering the inputs. (Brownlee, 2019)

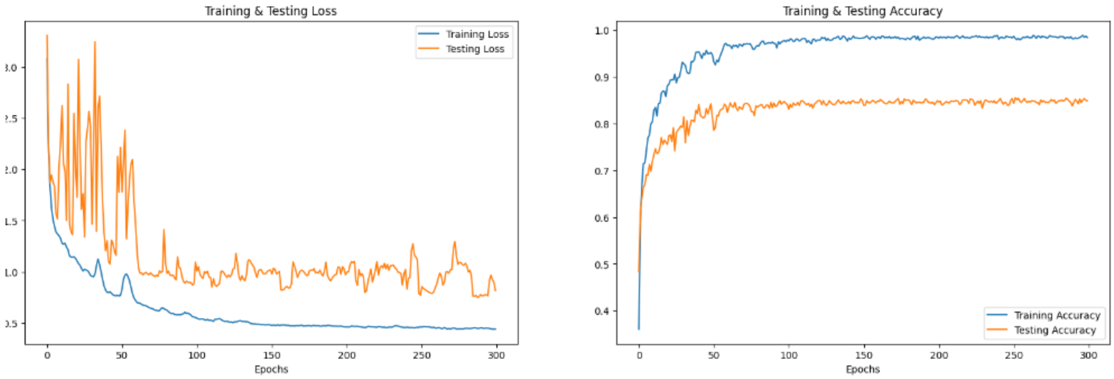


Figure 13: CNN model 6 Loss and Accuracy curves for on the training and test set while training the model

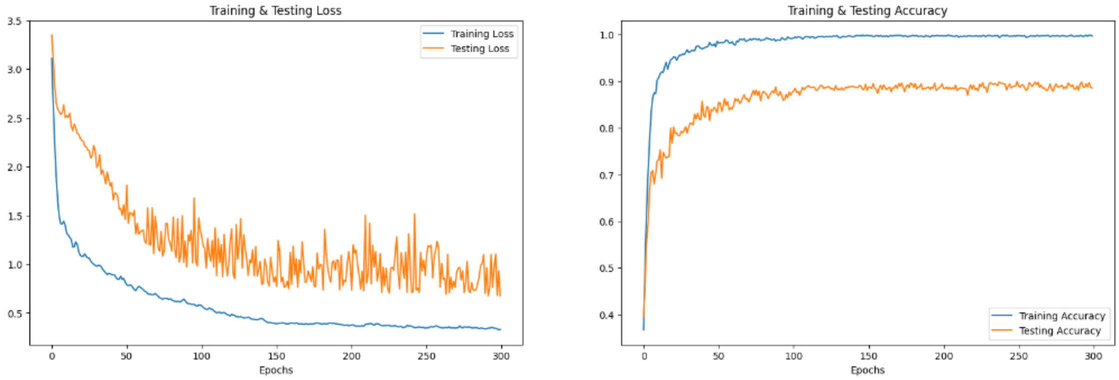


Figure 14: CNN model 7 Loss and Accuracy curves for on the training and test set while training the model

As shown in figure 14, there is a significant improvement in stability of the loss curve of the testing set, which confirms the validity of the decision to use batch normalization.

CNN 8

Model 8 uses the exact same architecture as model 7. However, for model 1-7, only the speech data set was used, whereas in model 8 both speech and songs were used. The benefit of neural networks is that their performance actually improves as more and more data is added, unlike some classical machine learning models. As such, to see if the performance of model 7 would improve, extra data was added.

As shown in figure 15, there is an even further improvement in the stability of the loss curve of the testing set, it is almost completely smooth. Furthermore, over-

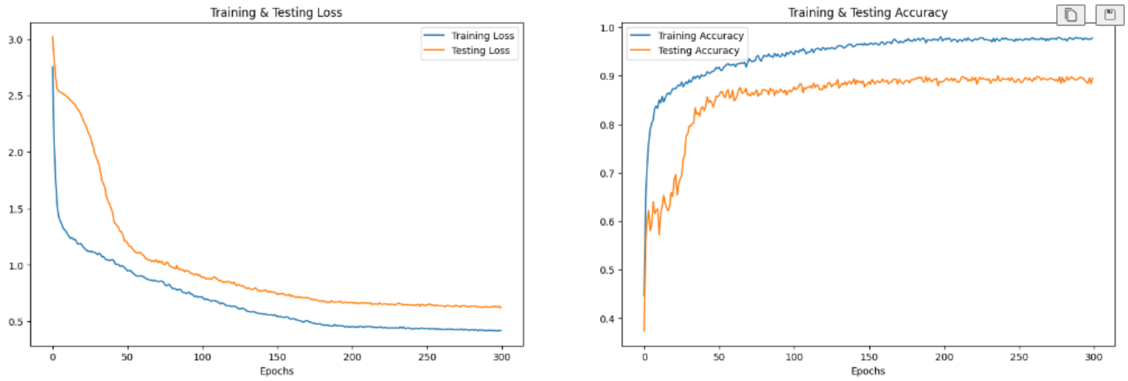


Figure 15: CNN model 8 Loss and Accuracy curves for on the training and test set while training the model

fitting is reduced as the accuracy curves of the training and test set are closer together.

CNN 9

The last model, model nine, uses the architecture of model 8. However, whereas for all other models 0.5 second frames were used to extract the data, for model nine this was reduced to 0.2 second frames. After extracting the features using 0.5 second frames, the total number of resulting features was 2700. As illustrated, more information was extracted using a smaller frame. As explained before, Neural Networks can benefit from more data, which was why the decision was made to use a smaller frame length to extract the features for the last model. In addition, using a smaller timeframe leads to less information loss, thereby also increasing the chance of better performance.

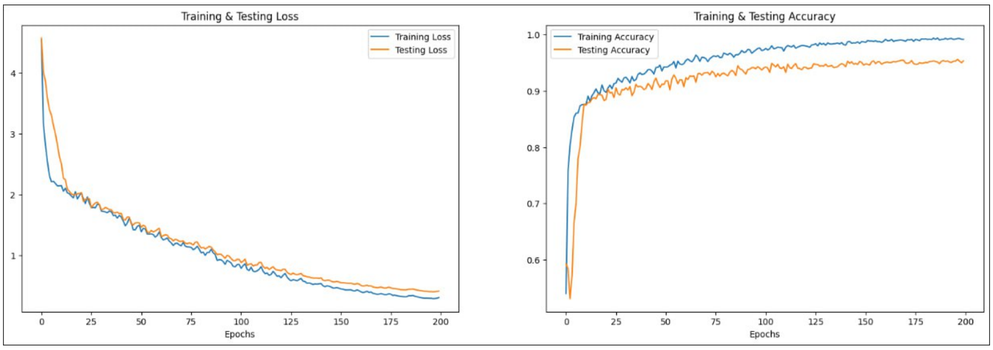


Figure 16: CNN model 9 Loss and Accuracy curves for on the training and test set while training the model

The loss and accuracy curves for the final model are shown in 16. It is possible to see there is almost no overfitting anymore as a development from CNN8, as the loss curves of the train and test almost completely overlap. It indicates the significance of reducing information loss in SER, perhaps through using smaller timeframes in feature extraction, as it significantly improved the generalization capabilities of the model.

7.3 Evaluation Metrics

For the evaluation metrics, it was decided that accuracy, precision, recall and F1 were to be used. In doing so, one would be able to assess the models in a holistic manner. Furthermore, due to the fact that the data set is rather balanced, no metrics such as accuracy had to be excluded.

8 Results

Given below are the results of the models on the test set, followed by respective discussions.

8.1 Baseline Models

8.1.1 SVM

Model	Accuracy	Precision	Recall	F1
SVM	0.80	0.80	0.80	0.80

Table 3: Weighted results for SVM

Emotion	Precision	Recall	F1
angry	0.80	0.81	0.81
calm	0.80	0.88	0.84
disgust	0.79	0.82	0.81
fear	0.81	0.85	0.83
happy	0.78	0.76	0.77
neutral	**0.77**	**0.58**	**0.66**
sad	**0.77**	0.72	0.75
surprise	0.86	0.86	0.86

Table 4: Results per emotion for SVM

Table 3, 4 and figure 17 summarize the results for the SVM model. The red values in table 4 show the lowest value for each metric. There are several observations that can be noted. First of all, as shown in table 4, the hardest emotion to classify is neutral, since it has the lowest values for precision, recall and F1 compared to all other emotions.

There are several reasons as to why this could be the case. First of all, as shown in figure 2, the number of records for neutral is only half that of all other emotions, since it functions as the baseline emotion and it is spoken at only one ”intensity”. As a result, perhaps due to there being less instances of neutral in the training set, the classifier has a harder time classifying this emotion.

Second of all, in figure 17 one can see that most often, neutral is classified as calm. What the difference is between neutral and calm is hard to say. As such, it could also be the inherent ambiguity that defines ’neutral’ that makes the classifier have a hard time classifying the emotion.

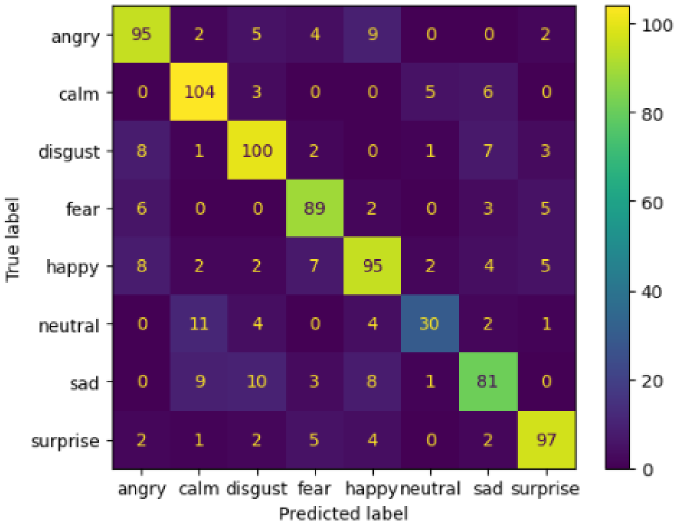


Figure 17: Confusion Matrix for SVM

The other emotion that is harder to classify is sad, given it has the same low precision value as neutral. Most misclassifications of sad are for emotions such as disgust, calm and happy, respectively. Given the the fact that emotions like disgust calm and happy are quite different from each other as well as sad, it can suggest that there is no clear explainable reason as to why the classifier has difficulty classifying sad. Perhaps this is chance, or the quality of the sound for sadness.

Using F1 score as a holistic measure of general classification abilities, then surprise can be seen as the easiest emotion to classify, as it has the highest F1 score with a value of 0.86.

Overall, the values given in table 3 appear to be valid, given that the same values are seen in the literature. As mentioned, Gao et al, (Gao et al., 2017) when using an SVM with various prosodic features, achieved an accuracy of 79.4%. This is close to the accuracy of 0.80% given in table 3.

8.2 Deep Learning Models

8.2.1 MLP

Model	Accuracy	Precision	Recall	F1
MLP	0.89	0.89	0.89	0.89

Table 5: Weighted results for MLP

Emotion	Precision	Recall	F1
angry	0.91	0.83	0.87
calm	0.87	0.94	0.90
disgust	**0.86**	0.91	0.88
fear	0.89	0.93	0.91
happy	0.92	0.85	0.88
neutral	0.87	**0.79**	**0.83**
sad	0.88	0.88	0.88
surprise	0.91	0.94	0.93

Table 6: Results per emotion for MLP

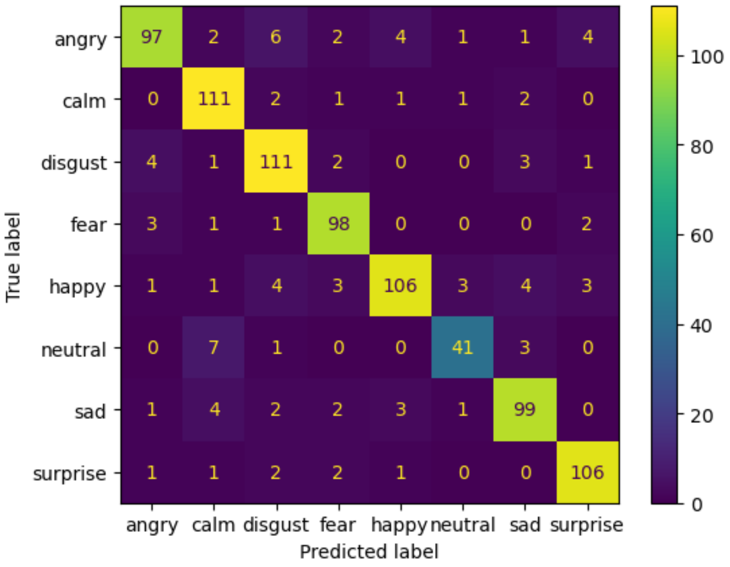


Figure 18: True vs Predicted Label Matrix

Table 5, 6 and figure 18 show the results for the mlp model. The red values in table 6 show the lowest value for each metric.

The performance of the MLP model was evaluated using a test size of 0.2 (20% testing data, 80% training data) with a random state of 10 to ensure consistency and reproducibility. This indicates the methodology used for splitting the data into training and testing sets.

After thorough testing, the model achieved an average accuracy of 0.8839 (89%) across multiple cross-validation folds. This shows the accuracy of the model in classifying emotions from audio data.

The model's effectiveness in classifying emotions was further assessed using average class accuracy, providing insights into its ability to distinguish between different emotional states. This suggests that the model's performance was evaluated for each emotion class individually, not just overall accuracy.

To visualize the model's classification results, a confusion matrix diagram was generated, as shown in figure 18, offering a comprehensive view of the model's performance on the test data. This visual representation helps understand where the model's predictions match the true labels and where they differ.

Overall, as shown in table 6, lowest values for most metrics occur for neutral, similar as to what was observed for SVM. Particularly recall is significantly lower than for neutral than for any of the other emotions. The recall for neutral is 0.79, whereas the next lowest value is 0.04 points higher at 0.83 for anger. Although disgust has in absolute terms the lowest value for precision at 0.86, neutral's value is just 0.01 point higher at 0.87. Hence, one can still say that for MLP neutral is the emotion that is the hardest to classify, a similar observation that was also noted for SVM.

Furthermore, when looking at figure 18, one can see that most misclassifications of neutral are for calm, another observation that was also noted for SVM. Similar reasoning as to why this may be the case for SVM applies to the MLP model, including the ambiguity that defines the neutral emotion.

Lastly, using table 6, the highest F1 value is for surprise, which, again, was also seen for SVM. The value is rather high, at 0.93, indicating that the classifier does very well at classifying surprise. Given that this was also seen for SVM, a possible explanation could be because surprise is a well-defined emotion in that it is easy to differentiate from the other emotions.

8.2.2 CNNs

Table 7 shows the aggregated results for the different CNN models, where red values indicate the lowest values and bold values the highest. There are several observations that can be made.

First of all, the worst performing models are CNN 1 and CNN 2 with for all metrics

Model	Accuracy	Precision	Recall	F1
CNN 1	**0.67**	**0.68**	**0.67**	**0.67**
CNN 2	0.73	0.73	0.73	0.73
CNN 3	0.85	0.85	0.85	0.85
CNN 4	0.80	0.80	0.80	0.80
CNN 5	0.85	0.85	0.85	0.85
CNN 6	0.85	0.85	0.85	0.85
CNN 7	0.89	0.89	0.89	0.89
CNN 8	0.90	0.90	0.90	0.89
CNN 9	**0.95**	**0.95**	**0.95**	**0.95**

Table 7: Weighted results for all CNN models

values being lower than 0.8. Moving from model 2 to model 3, the performance metrics improve with more than 0.1. This difference is explained by the switching of architecture between model 1-2 and model 3. In model 1-2 an architecture inspired by (Badshah et al., 2017) was used, which consisted of four convolutional layers. In model 3, a much simpler architecture inspired by (de Pinto et al., 2020) was used.

Considering the intricacies of sound, it was thought that a complex model was need to classify sound. Hence why first an architecture with four al layers was used for CNN 1-2. However, as performance shows in table 7, a much simpler architecture was needed to properly classify sound.

From model 3 onwards, there is an increasing trend across the evaluation metrics as adjustments on the models are made. It is noticeable however that the exception of this is model 4, in which values for the evaluation metrics are lower than for model 3. The difference between model 3 and model 4 is that in model 4 only one feature was used as opposed to all features, taking inspiration from (de Pinto et al., 2020). As explained in the methodology the loss and accuracy curves improved moving from model 3 to model 4. However, it is now possible to see that this does not have to indicate that the performance of the model increases. One explanatory reason for the worse performance of model 4 compared to model 3 is that with fewer features, less information is available to the model. As such, it has less information to use to classify emotions, thereby worsening performance.

The most significant increase in performance is from model 8 to model 9, with all metrics increasing by at least 0.06. Between these two models, the only thing that changed was that smaller timeframes were used in the feature extraction. Instead of using 0.5-second timeframes, in model 9, 0.2-second timeframes were used. Using smaller timeframes means that more information from the sound is retained, thus leasing to smaller information loss. As already explained in the literature review, the reduction of information loss is essential in achieving good classification performance due to the subtleties and intricacies of emotions through sound. Clearly, this is also seen in this project as a reduction in the timeframes led to a significant increase in performance of the classification model.

Best Model

Overall the best model is CNN 9, with the highest values for all evaluation metrics. Further results for model 9 are shown in table 8 and figure 19. In table 8, the red values indicate the lowest values for the evaluation metrics.

Emotion	Precision	Recall	F1
angry	0.97	0.97	0.97
calm	0.95	0.97	0.96
disgust	0.97	0.97	0.97
fear	0.95	0.95	0.95
happy	0.97	**0.92**	0.95
neutral	0.97	0.96	0.96
sad	**0.92**	0.94	**0.93**
surprise	0.93	0.99	0.95

Table 8: Results per emotion for CNN 9

Both table 8 and figure 19 show how well the CNN9 does at classifying the emotions. The lowest values for the metrics are still above 0.9, which is a significant performance. As such there is not really one emotion that the classifier really struggles with to classify. In particular, what is is interesting to note is that the classifier appears to have no trouble at all at classifying the neutral emotion. For both MLP and SVM this was the hardest emotion to classify. In effect, looking at the red values in table 8, happy and sad appear to be more difficult to classify since they have the lower values for precision, recall and F1.

To further investigate the classification ability of the different CNN model for the neutral emotion, table 9 was produced.

In table 9, the red values indicate the lowest values whereas the bold values indicate the highest values. One can see that as the models progress, so does the models' ability to classify as neutral. Nevertheless, it is possible to see that in particular, CNN9 has an incredible ability to classify neutral, as is performance metrics for neutral are significantly higher than all the other models.

8.3 Comparative Analysis of the Models

Table 10 shows the performance across SVM, MLP and the best CNN (CNN9).

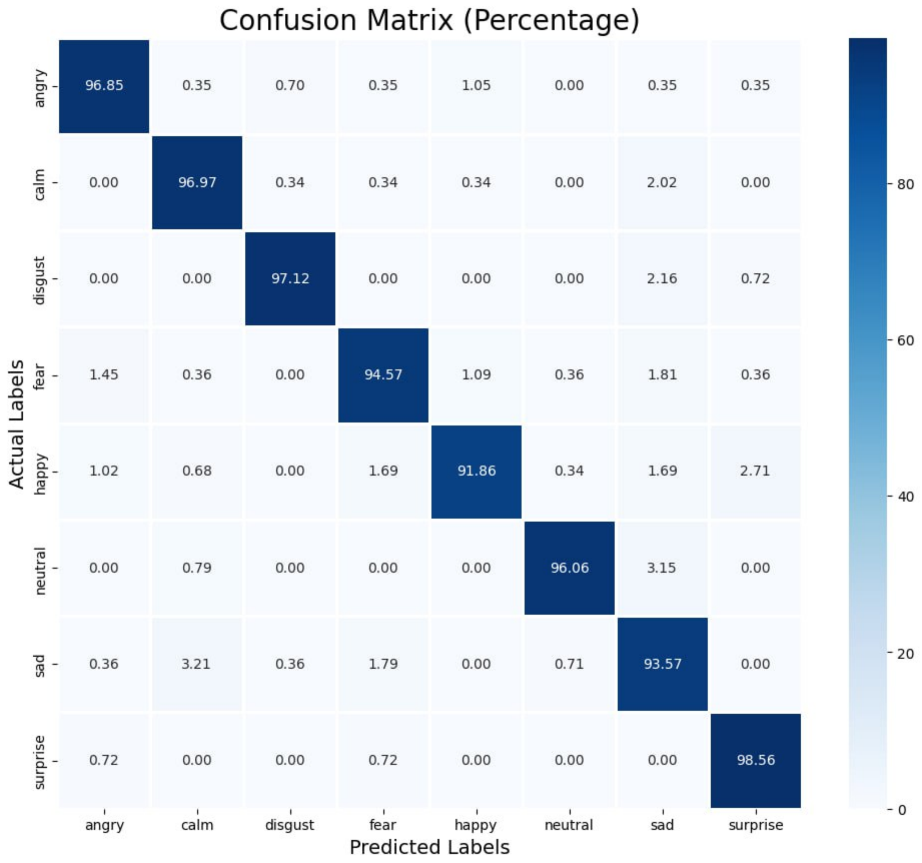


Figure 19: True vs Predicted Label Matrix

CNN ID	Precision	Recall	F1
1	**0.52**	**0.52**	**0.52**
2	0.66	0.57	0.61
3	0.79	0.85	0.82
4	0.77	0.82	0.79
5	0.83	0.75	0.79
6	0.73	0.85	0.79
7	0.79	0.88	0.83
8	0.87	0.92	0.90
9	**0.97**	**0.96**	**0.96**

Table 9: Results for Neutral emotion for different CNN models

Overall one can see that CNN-9 is the best performing model and SVM the worst.

Model	Accuracy	Precision	Recall	F1
SVM	0.80	0.80	0.80	0.80
MLP	0.89	0.89	0.89	0.89
CNN 9	0.95	0.95	0.95	0.95

Table 10: Comparison of SVM, MLP and the best CNN model (CNN9)

In general the deep learning models perform better than the baseline SVM model.

Nevertheless, even the worst-performing model, SVM, performs rather well with all evaluation metrics still above 0.8.

Furthermore, the best performing model, CNN9, was built on the architecture by (de Pinto et al., 2020). In the paper by (de Pinto et al., 2020), the authors managed to get a score of 0.91. CNN9 obtained an F1 score of 0.95, which indicates that our model outperformed the original model. The most significant differences are that in CNN9 more features are used and a smaller timeframe is used. As such, this is an important observation for the research space for SER in that more features and smaller timesteps can lead to significantly better results. Additionally, this proves that we were able to achieve our goal of building a more robust machine learning model.

9 Conclusion and Evaluation

9.1 Conclusion

For this project, the goal was create a highly accurate machine learning model to categorize speech samples into various emotional states, such as happiness, sadness, anger, fear, and more, with the goal of constructing a robust system for emotional state classification.

Ten different features were extracted from waveforms of the sound data, which were fed to different models. Specifically, a baseline SVM model was built, as well as an MLP and various CNN's in order to find the most robust and best performing one.

Overall one found that the best performing model was a CNN built on the architecture proposed by (de Pinto et al., 2020), using all 10 features and a smaller time step of 0.2 second frames. In doing so, the final model reached an accuracy of 0.95, precision of 0.95, recall of 0.95 and F1 of 0.95. As such, the model outperformed the original model by (de Pinto et al., 2020) who got an F1 score of 0.91. As such, one can conclude that the goal to build a robust classifier is attained.

Through the process, it was learned that using more features and a smaller timestep can significantly boost model performance. This is a valuable takeaway for further research in the domain of Speech Emotion Recognition.

9.2 Evaluation

9.2.1 Challenges and Limitations

There are several limitations to this research.

First of all, the use of PCA for data reduction and potentially improved model performance was neglected after it resulted in worse performance for SVM. However, the decline in performance observed for SVM, and might have not been the case for the other models, such as the MLP or the CNN's. As such, for a fairer comparison, it would have been better to test all models with the reduced number of PCA components found in order to truely assess whether it would be valuable or applicable to the project.

Furthermore, considering that the MLP had quite a good performance with limited parameter tuning, one could have further explored the model's potential. That is, it would have been interesting to apply the features extracted with the smaller timeframe as used in the CNN-9 on the MLP.

9.3 Future Plans

Moving forward, there are several suggestions for further research.

First of all, more feature selection methods can be applied to really pin down what features are most important in classifying emotion from speech.

In addition, it would be interesting to take the route of using automatic feature extraction from image data, converted from sound data, as (Badshah et al., 2017) did. That is, the mel spectrograms can be fed into a CNN and the model can be used to extract the most distinctive features necessary to classify emotion.

Lastly, it would be interesting to try different deep learning algorithms such as RNN's or state-of-the-art transformers to see if performance can be boosted even further. .

10 References and Citations

References

Awan, A. A. (2022, Nov). *A complete guide to data augmentation.* DataCamp. Retrieved from `https://www.datacamp.com/tutorial/complete-guide-data-augmentation`

Badshah, A. M., Ahmad, J., Rahim, N., & Baik, S. W. (2017). Speech emotion recognition from spectrograms with deep convolutional neural network. *2017 International Conference on Platform Technology and Service (PlatCon).* doi: 10.1109/platcon.2017.7883728

Brownlee, J. (2019, Dec). *A gentle introduction to batch normalization for deep neural networks.* Retrieved from `https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/`

Burnwal, S. (2020, May). *Speech emotion recognition using ravdess.* Kaggle. Retrieved from `https://www.kaggle.com/code/shivamburnwal/speech-emotion-recognition`

Das, J. K., Ghosh, A., Pal, A. K., Dutta, S., & Chakrabarty, A. (2020). Urban sound classification using convolutional neural network and long short term memory based on multiple features. *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS).* doi: 10.1109/icds50568.2020.9268723

de Lope, J., & Graña, M. (2023, Apr). An ongoing review of speech emotion recognition. *Neurocomputing*, *528*, 1–11. doi: 10.1016/j.neucom.2023.01.002

de Pinto, M. G., Polignano, M., Lops, P., & Semeraro, G. (2020, May). Emotions understanding model from spoken language using deep neural networks and mel-frequency cepstral coefficients. *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS).* doi: 10.1109/eais48028.2020.9122698

Gao, Y., Li, B., Wang, N., & Zhu, T. (2017, Nov). Speech emotion recognition using local and global features. *Brain Informatics*, 3–13. doi: 10.1007/978-3-319-70772-3_1

LivingStone, S. R. (2018, October). *Ravdess emotional speech audio.* Retrieved from `https://www.kaggle.com/datasets/uwrfkaggler/ravdess-emotional-speech-audio`

LivingStone, S. R., & Russo, F. A. (2018, May 16). The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. , *13*(5), 1–35. Retrieved from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5955500/` doi: 10.1371/journal.pone.0196391

Monigatti, L. (2023, May). *Data augmentation techniques for audio* *data in python.* Towards Data Science. Retrieved from https://towardsdatascience.com/data-augmentation-techniques-for-audio-data-in-python-15505483c63c

Ndiritu, F. (2021, Dec). *Dropout regularization to handle overfitting in deep learning models.* Retrieved from https://www.section.io/engineering-education/dropout-regularization-to-handle-overfitting-in-deep-learning-models/#:~:text=Dropout%20regularization%20will%20ensure%20the,learn%20redundant%20details%20of%20inputs.

Ramakrishnan, S., & El Emary, I. M. (2011, Sep). Speech emotion recognition approaches in human computer interaction. *Telecommunication Systems*, *52*(3), 1467–1478. doi: 10.1007/s11235-011-9624-z

Raval, P. (2023, October 12). *Speech emotion recognition project using machine learning.* Retrieved from https://www.projectpro.io/article/speech-emotion-recognition-project-using-machine-learning/573