National College of Ireland

Classification of Human Age Group by Implementing Deep Learning Models on Audio Data

MSc Research Project
MSc in Data Analytics

Srijan Pandey
X18127312

School of Computing
National College of Ireland

Supervisor: Dr. Rashmi Gupta

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Srijan Kumar Pandey

Student ID: X18127312

Programme: MSc in Data Analytics **Year:** 2019/2020

Module: Research Project

Supervisor: ……

Submission Due Date: ……

Project Title: Classification of Human Age Group by Implementing Deep Learning Models on Audio Data

Word Count: …10451…… **Page Count**……24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Srijan Kumar Pandey

Date: 28th September 2020……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Classification of Human Age Group by Implementing Deep Learning Models on Audio Data

Srijan Kumar Pandey
X18127312

Abstract

Audio data classification is one of the most challenging fields in data science because of the complexity in the pre-processing of the data. This motivates the researchers to perform various techniques to reduce the complexity of data and improve performance. This research is aimed to predict the age group of an individual based on his/her voice data. This technique would be beneficial for the organization that wants to focus on its target age group and pitch the right product to the right section of the society. This project aimed at reducing the complexity of the audio data by first separating noise and dead air from the audio to generate clean data using signal enveloping. After that key audio features were directly extracted using the Mel Frequency Cepstral Coefficient rather than taking the Discrete Cosine Transform of log of filter bank. The reason for choosing MFCC was because it retains a large amount of information from the audio as it carries with itself time, frequency, and energy domain in each frame. The coefficients were scaled and then converted into an array of audio features. The labels were generated with the corresponding CSV file. The techniques applied had a temporal approach that was directly used on the audio samples. Speech Accent Archive dataset was used and the model was trained using a Fully Connected Convolutional Neural Network and Time Distributed Long Short Term Memory Recurrent Neural Network. This research also compares the performance of both the model on the same dataset through the accuracy obtained by both of them. CNN gave an accuracy of 62.45% on the test set whereas the LSTM-RNN model outperformed CNN and gave an accuracy of 66.07% on the same dataset.

1 Introduction

1.1 Background and Scope

Audio classification is a process of analyzing audio to gather information which can help to identify demographics, linguistic background, and extensive research opportunities on the audio source. Audio classification is generally termed as sound classification as well and it is currently one of the most researched topics in the AI industry. This is the core of other AI technologies that are going through extensive research right now e.g. Automatic Sound Recognition (ASR), Google's or Apple's virtual assistants, and text to speech technology for emotional and quality analysis and in the medical field. (Gustavo Noffs et al., 2020) It is worth mentioning that this technology has gained a lot of benefits in the development of business strategies, improves communication and future interaction using audio

sentiment analysis. Audio in general contains so much information and that information can be used to surface intelligence for building effective cost containment and service strategies. (Xiang Li et al., 2020) It has huge importance in multimedia indexing and its retrieval. Audio classification provides a categorical analysis of audio data which can later be used for advanced functionalities and valuable intelligence from the audio source. It also helps us to discover information relating to strategy product, process, operational issues, and people's performance. The sound classification has also proved a significant development in areas like smart home security systems and predictive maintenance. There are 4 types of audio classification and they are:

- Acoustic data classification: This is a classification performed on recorded audio signals. It has major applications in building audio multimedia.
- Environmental sound classification: This is a classification of sounds that are present in our environment. It has major applications in security systems and predictive maintenance.
- Music classification: This method is used to classify different music, genres, and instruments. It has its application in recommender systems (Jie Jiang, 2020), and discovering trends. (Snehlata Barde, 2020)
- Natural language utterance classification: This is the classification of audio based on the language, accent, language features, and human speech classification. Major applications are virtual machines, text to speech analysis. The research performed in this project falls under the Natural Language Utterance Classification.

1.2 Motivation and Industry Application

Human beings can differentiate between two sources of sound very easily. Their intelligence helps them to identify the sound and feel the emotion however machines don't have such kind of intelligence. Although human beings can identify most of the sounds it becomes difficult to identify some sounds if they are low or very weak, from human ears. Human ears have some limitations as well. Sounds which have very high frequency, human ears won't be able to differentiate then because above a certain frequency all the sounds appear the same but if we can train machines, they would be able to differentiate any kind of sound based on how they are trained. Similarly, there are few other areas where only an expert ear can tell the difference but a normal person can't e.g. a doctor can detect heart issues listening to heartbeat but a common man cannot. An automobile mechanic can detect problems in an engine by the sound of it but a normal person might not. If computers are trained it would be easier for a common person to perform such a task as well. The motivation for this research is that due to the growth in the e-commerce industry service plays a very key role in company growth. More and more companies are trying to pitch the right product to the customers and our project objective would be very beneficial for such organizations. This project would try to predict the age group of a person based on his voice data. If we can develop such a system it would become very easy for companies to pitch the right product to the right person as different age groups have different requirements and they would save a lot of money in advertisements.

1.3 Problems in audio classification:

Although audio classification is a widely researched topic in the AI world there are few challenges while we work with audio data. Some of them are discussed below:

Audio data is very different from other types of data and it requires a lot of pre-processing compared to any other type of data. Pre-processing audio data is a very challenging and complex process. Dealing with the noise in data is a major challenge. Sometimes we need to perform source separation so that we can use a model. This research project is using samples created from MFCC's to classify audio and we'll see in this research MFCC's are also image like but in general, it is very different from image data. Positioning is very important which is not important in image data. Data annotation also adds a lot of computational costs. Some other challenges of audio data are that it requires a lot of storage space. This also makes it difficult to obtain a huge amount of audio data. (Yonas Woldemariam, 2020) In audio classification, the quality of audio also plays a very key role.

Interpreting the data in the time domain is challenging and getting high precision is also difficult. Last but not least it is very difficult to find audio data for research that is available to all in the public domain. The reason being audio data is considered to be private data as it has a human identifiable feature. There are also a lot of copyright issues with audio data as well which makes the availability of data difficult to work.

For the sake of domain knowledge, it is very important to understand a few terms in audio. It would be very beneficial while we would be carrying out our research. Some of them are: -

Sound Wave: A sound is produced when an object vibrates. These vibrations are determined by the oscillation of air molecules. Waves are caused by changes in the air pressure which means when High pressure is alternated with low pressure. The waves can be represented using waveforms.

Waveform: Waveform is a graphical representation of waves between amplitude and time. It can also be explained as a point that oscillates with different amplitude with different points in time. In this research, the waveform is represented in the form of a sine function. It can be represented as:
$Y(t) = A \sin(2\pi ft + \phi)$ Where A is Amplitude, f is Frequency, t is for time and ϕ stands for the phase of the waveform. This is a simple mathematical representation of our waveform. The graphical representation of male and female voices is represented below in figure 1 and 2.

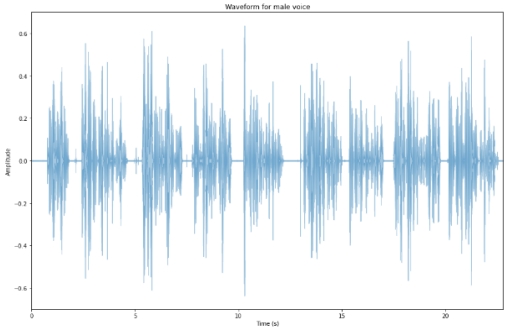


Figure 1: waveform for male voice

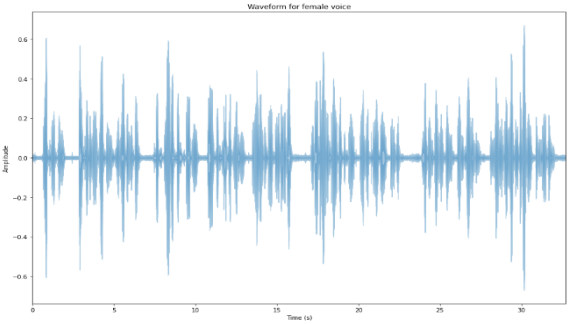


Figure 2: waveform for female voice

There are few important elements of a sound wave. Some of them are:

Period: Period can be defined as the time taken to finish one cycle. This gives us an idea of the start of a wave and then the occurrence of the next wave or interval before we see the next peak again. Its standard unit of measurement is in seconds. It is very strictly co-related with frequency.

Frequency: Frequency can be defined as the inverse of period. Which means higher the period lowers the frequency and vice versa.

Amplitude: Amplitude can be defined as the distance of a point from equilibrium. It gives the maximum vertical displacement of a point within a wave.

Pitch: Pitch can be defined as the quality of a sound produced due to the rate of the vibrations which are producing the sound. In the practical world, it generally helps to identify how high or low our tone is there in the voice. Pitch is connected to the frequency which means higher frequency will have a higher pitch.

Loudness: Loudness can be defined as a way to calculate the energy of a sound. It is represented in the form of Decibels(dB) which is a logarithmic scale on which the power of sound energy is quantified. Loudness is connected to the Amplitude which means higher amplitude will have higher loudness.

Fourier transform: Fourier transform can be defined as a process of decomposing a periodic sound into a sum of sine waves which all vibrate at different frequencies. This is used to describe a very complex sound. Mathematically it is represented as $S = A_1 \sin(2\pi f_1 t + \varphi_1) + A_2 \sin(2\pi f_2 t + \varphi_2)$ Where S represents sound waves, A represents Amplitude, f represents Frequency, t represents Time and ϕ represents the phase. In a Fourier transform, we are particularly interested in Amplitude. The reason for it is because amplitude tells us how much a specific frequency contributes to a complex sound. The higher the amplitude more the frequency is contributing to the complex sound which has to be decomposed.

Power Spectrum: Once we perform a Fourier transform, we get a power spectrum. Spectrum gives us magnitude as a function of frequency. When we do Fourier transform, we move from the time domain to the frequency domain. So, in a spectrum, we lose temporal information. Power Spectrum is shown below for a male and female voice in figure 3 and 4.

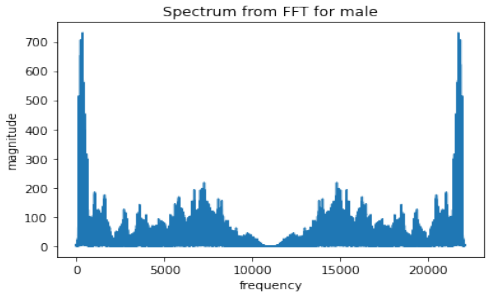


Figure 3: spectrum for male voice

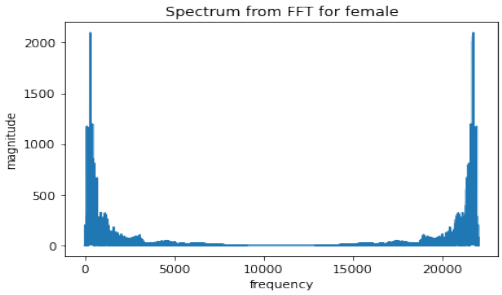


Figure 4: spectrum for female voice

Short-Time Fourier Transform: STFT computes several FFT at different intervals. By doing so it preserves information about time and the way sound moves with respect to time. The different interval where we perform STFT is called Frame Size. So, Frame size consists of a number of samples that we fix later and is then termed as Fixed Frame Size. STFT gives us a Spectrogram which is a representation giving us information about Magnitude as a function of Frequency and Time. Spectrograms are fundamental for performing deep learning operations on audio data. The images of a spectrogram are shown below in figure 5 and 6:

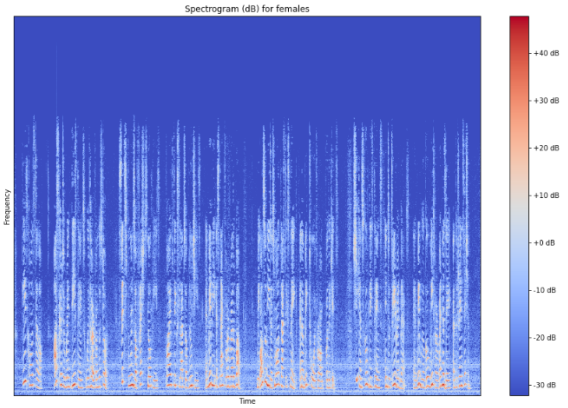


Figure 5: spectrogram for female voice

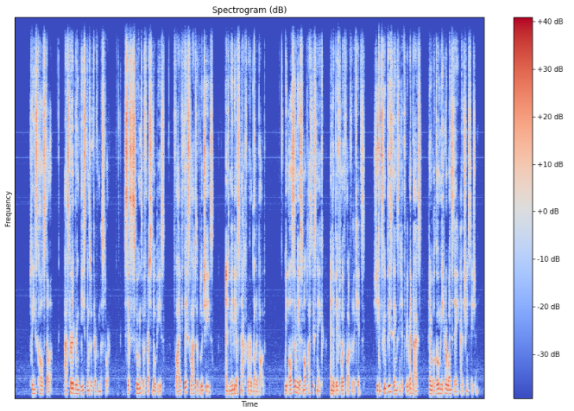


Figure 6: spectrogram for male voice

Mel Frequency Cepstral Coefficients (MFCCs): MFCCs is another important feature which is very fundamental for audio classification in deep learning. It is a frequency domain feature used to capture a timbral and textural aspect of a sound which means if we have the same frequency and pitch of audio there would be a difference in the timbral or quality of sound and MFCCs are capable of capturing that information. To perform MFCC we perform a Fourier Transform and move from the time domain to the frequency domain. One of the major benefits of MFCCs over spectrogram is that they approximate the human auditory system very well which means they try to model the way we perceive frequency. The result of extracting MFCCs is a bunch of coefficients. The coefficients are

calculated at each frame so we have an idea of how MFCCs are evolving over time. Various applications of MFCCs are in

- Speech recognition
- Audio Classification
- Music Genre classification
- Musical Instrument classification

MFCC images for male and female audios are shown in figure 7 and 8:

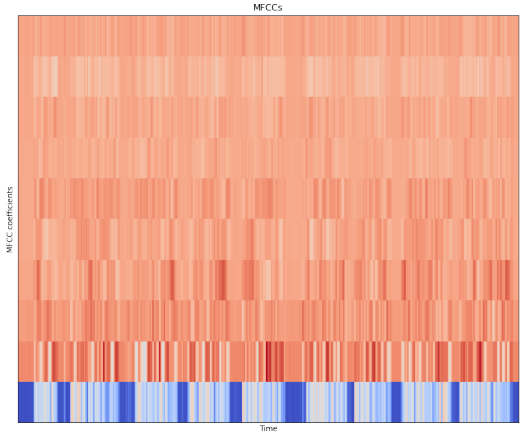


Figure 7: mfcc for male voice

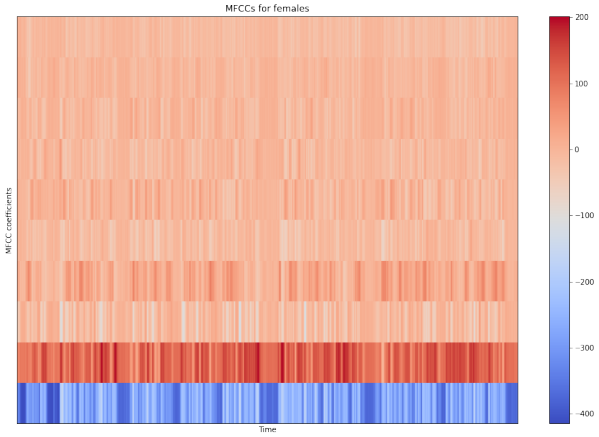


Figure 8: mfcc for female voice

Along with the above-mentioned terms, this research would use a signal envelope technique to clean the audio and remove the dead air. This helps us in getting pure audio without void spaces. The experiment was first carried out using a fully connected convolutional neural network which is one of the best techniques for image classification in deep learning. Since MFCC's are image type so it was expected to give a good result with it. The CNN used had (3*3) convolutional layer, (2*2) pooling layer, and a fully connected layer. The other model used was Long Short Term Memory neural network which is a type of recurrent neural network and it was used with time distributed layer. So basically after this detailed intuition about the methods used we come to our research question which is

2 Research Question

How effectively can machine learning models identify age groups based on the voice sample of an individual?

2.1 Report structure

The next segment will discuss the recent work which has been done in various fields directly connected to the scope of our topic. It will be followed by the methodology used which will discuss the pre-processing and modeling in detail followed by the design specification, result, and conclusion of our finding. In the last of the report, we will discuss the future work which we propose or were not able to carry out due to the time constraint in this report and references would be there at the end.

3 Related Work

3.1 Machine Learning techniques used for Audio Classifications

In a research performed by (Wu, 2019) the author tried to detect multiple techniques which can help in the detection of an audio event and performed his analysis in audio with the very complex

surrounding. The author used an ensemble learning approach where he performed feature selection using a decision tree classifier and SVM was used for classification. Modeling was performed using a decision tree classifier along with AdaBoost. The performance of the model was evaluated by comparing it with the experimental data. The theory came up with an excellent result where the accuracy of pure voice was 98.01% and the music was 98.33%.
Interesting research work performed on baby cry to classify the current situation of them as if they are hungry, ill, deafness or brain damage, etc. This research was carried out to detect early-stage problems by analyzing the cries of a baby as they cannot talk or share their problems like a grown-up. The research was focused on comparing the performance from a pre-trained ResNet50, SVM and then ensemble technique using a combination of ResNet50 and SVM (Lillian Le, 2019). The experiment showed that the ensemble technique deployed by combining the ResNet50 and SVM gave the highest accuracy compared to traditional CNN, SVM, and ResNet50. CNN gave an accuracy of 87.03%, TL through ResNet50 gave 90.80% accuracy, SVM gave 90.10% accuracy and the combined model gave 91.10% accuracy. This research talks about imbalanced classes however no measures taken were discussed in the paper. The dataset size was also low however no data augmentation technique was followed. The research used a spectrogram as a feature extraction technique which might be not as useful as MFCC so it could even try MFCC to see if the accuracy increases.

Deep learning techniques are very widely used when it comes to the classification of multi-channel audio. (Abigail Copiaco, 2019) along with his team experimented on domestic multi-channel audio where the team performed a comparison between existing pre-trained Neural Network models along with SVM. Spectro-temporal features were extracted from the audio which is generally done by using an autoregressive model over time-related envelops of a signal. Features that were extracted from the audio were termed as activations and it came from the important layers of the neural network. These activations were later used to train the model. A spectrogram represents frequency as a function of time. The activations obtained from trained neural network components helped to make the features of the components of the spectrogram as a strong learner. This also enhanced the performance of the model and the research gave an accuracy of 97% when it was used along with SVM.

When it comes to detection of violent content in a video an ample amount of work has been done in the field of content classification over a video but it happens sometimes due to environmental conditions it becomes difficult to process video data so in that case audio classification becomes important to detect the violent content e.g. in the case of surveillance. For this purpose, (Sercan Sarman, 2018) experimented with audio data to classify the video content as violent or not violent. They used ensemble learning techniques to do so. ZCR (Zero Crossing Rate) was used as a feature extraction technique and the random forest was used as a classifier. As per the author, the dataset was highly imbalanced so they tried to do under-sampling and then used the Bagging technique along with SVM for all the features extracted using ZCR and MFCCs. The results gave the highest efficiency as compared to other works done on the same dataset where bagging with ZCR gave an accuracy of 68.8% and Random forest with ZCR gave 66% accuracy. The research gave a low accuracy and the reason could be that neglecting the video content and focusing on the audio content of the data. A multimodal approach might give better accuracy. The research also does not mention any denoising technique which was used to remove the noise from data.

When we listen to music our body and mind respond differently. These movements due to music are termed musical gestures. (Paul Best, 2018) researched on the classification of these gestures using audio extracts. Audio extracts were labeled and recognition and tracking of motion were performed using machine learning tools. The team applied both of them to predict a hypothetical gesture. Body movements were analyzed using a model created with a combination of the Hidden Markov Model and the Gaussian Mixture Model. The experiments were carried out in 4 sets of descriptors which were a single descriptor, a combination of two descriptors, a combination of 3 descriptors and Genetic algorithms. The descriptors used were MFCC from 2 to 12, Energy, loudness, ACI, frequency, periodicity. The experiment gave maximum accuracy when all the descriptors were used and it was 61%. The experiment does not give us good accuracy and it was high on computational complexity.

Another research performed by (Rong, 2016) the research was carried out using 4 layers of audio data which are frames, clips, shots, and high-level semantic units of audio. 3 types of feature extraction techniques were used which were MFCCs (Mel Frequency Cepstral Coefficients), STE (Short Time Energy), and ZCR (Zero Crossing Rate). The model was created using SVM along with Gaussian Kernel. The research gave a higher accuracy for audio on which the classification was performed. The research was performed on 2 different datasets and they gave an accuracy of 87.6% and 86.3% respectively. The summary is shown in table 1.

Table 1: Audio classification using machine learning techniques

Author	Sample	Title	Source	Methods	Findings
Dan Wu (2019)	Dataset collected from TV, radio, live reports, music, street, transport, clocks, etc around 10000 samples 10 hr long recordings.	An audio classification approach based on machine learning	International conference on ICITBS	Feature selection with Decision tree and modeled with SVM and Decision tree with AdaBoost.	accuracy of pure voice was 98.01% and the music was 98.33%.
Feng Rong (2016)	General sounds and audio scenes from multiple origins	Audio classification method based on ML	International conference on ICITBS	Feature extraction of ZCR, MFCC. Modeling with SVM + Gaussian Kernel	87.6% and 86.3% accuracy resp. on both data.
Abigail Copiaco (2019)	Sound Interfacing through the Swarm (SINS) database	Scalogram NN activation with ML	International Symposium on Signal Processing and Information Technology	Features were extracted with spectrogram and modeling was done using pre-trained Neural Network models along with SVM	accuracy of 97%
Sercan Serman (2018)	Technicolor Violent Scene Detection dataset	Audio Based Violent Scene Classification Using Ensemble Learning	6th International Symposium on Digital Forensic and Security (ISDFS)	bagging along with SVM and ZCR & Random forest with ZCR	68.8 and 66% respectively.
Paul Best (2018)	Alireza Farhang cello music piece compilation	Musical Gesture Recognition Using Machine Learning and Audio Descriptors	International Conference on Content-Based Multimedia Indexing (CBMI)	combination of the Hidden Markov Model and Gaussian Mixture Model along with multiple descriptors	61% accuracy when all the descriptors were used
Lilian Le (2019)	Baby Chillanto Database obtained from the National Institute of Astrophysics and Optical Electronics, CONACYT, Mexico	Using Transfer Learning, SVM, and Ensemble Classification to classify Baby Cries based on their Spectrogram Images	16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)	pre-trained ResNet50, SVM and then ensemble technique using a combination of ResNet50 and SVM	ResNet50 gave 90.80% accuracy, SVM gave 90.10% accuracy and the combined model gave 91.10% accuracy

3.2 Deep Learning techniques used in Audio Classifications

Acoustic scene clustering is a technique that has been introduced recently where we try to group the audio which has similar characteristics together even if we don't have any information about the audio. (Yanxiong Li, 2020) came up with research where the features were extracted from the audio using CNN. The features extracted are a deep embedding whereas a hierarchy of clusters was built in a process to perform cluster analysis. A clustering algorithm was used and it was integrated with CNN using a unified loss function. Deep embedding gave a 64.6% score.

In a research paper by (Shahin Amiriparian, 2019) the author performed research where he suggested the implementation of deep learning in analyzing the eating habits of people by classifying the sounds of eating. This research was very challenging however the team came up with some good results by using Mel Spectrogram as feature extractor, pre-trained CNN's, and training the model using pre-trained AlexNet and VGG16 models. Activations of 3 fully connected layers were used which are fc6, fc7, and fc8. They were used as feature vectors. fc6 gave the best classification result. The research came up with an average recall value of 79.9%.

Another research done over audio samples was performed by (Naranchimeg Bold, 2019) where the team introduced a multimodal approach towards the classification of bird species using their sound and visuals. The research aims at studying the features of both audio and video and then using a kernel-based fusion to classify the species. Feature extraction was done using activation values of the inner layer of CNN and then features were combined using Multiple Kernel Learning. This approach gave an accuracy of of78.15% which is the better result when compared from all the conventional techniques of single modality learning, simple kernel, and conventional fusion technique.

The cochlea is a small spiral-shaped bone that is present in our ear and helps in auditory transduction. This is one of a very important bone and damage to it causes loss of hearing as well. A research performed by (Enea Ceolini, 2019) collected data from a spiking silicon cochlea which was used as an event-driven sensor installed at the front end. Deep learning was used to study these waves. 3 architectures were practiced on a 1-sec frame. These were CNN, RNN with CNN at the front end, and Multilayer Perceptron. RNN with CNN at the front end gave the highest accuracy of 81%. The research used spike features instead of traditional log filter banks or MFCC's and it states that the result was close to the traditional feature extraction technique. Using MFCC or log filter banks might have improved the performance of the model.

When it comes to classifying the instrumental audio research by (Justin Hall, 2019) used deep learning method to classify audio. The research used a log spectrogram to convert audio data into its corresponding image. This was also used as a feature extraction technique. The audio images were classified using a Convolutional neural network which is a part of deep learning techniques. The tests gave a result of 73.7% where the author claimed that it was a very limited dataset. The research doesn't talk about any data augmentation technique being applied and also no denoising technique was used. Downsampling of audio and normalization techniques also works well in this scenario and implementation of these techniques might have yielded better results.

Music segmentation is a technique where we divide a piece of audio into multiple segments. These segments could be chords, rhythms, pitches which sounds the same, etc. A research performed by (McCallum, 2019) used CNN for music segmentation. The experiment performed unsupervised training of CNN using deep feature embeddings. Only the temporal dimensions of the audio feature were considered however using this technique gave the state of the art result in unsupervised deep learning of musical segmentation. The experiment perfectly sampled the data and the boundaries were detected. Two datasets were used for the test and it gave a 66% score for recall.

Audio classification also has a great application in the medical industry. It has been proved that it can detect multiple health conditions related to the heart and lungs by analyzing the audio. Similar research was carried out for the detection of Parkinson's disease by analyzing the audio signals. Parkinson's generally affects the movement in a person due to nervous breakdown. Research by (Marek Wodzinski, 2019) the data consisted of audios of people suffering from Parkinson's disease. Their voice was recorded with sustained phonation. The team calculated the spectrum from the audio samples where they focused on the frequency and amplitude of the signals. Data augmentation was performed in the time domain. This helped them to prevent overfitting. The images obtained from the spectrum were trained using a transfer learning technique where ResNet architecture was used to perform the classification. The test gave a result of 91.7% and was very helpful in detecting Parkinson's disease. This result is important because it showed that if we only consider the frequency

content of audio, we can still achieve a very good result. The author compared his result with state-of-the-art results in this field. Summary is explained in table 2.

Table 2: Audio classification using deep learning techniques

Author	Sample	Title	Source	Methods	Findings
Yanxiong Li (2020)	LITIS-Rouen and DCASE-2017	Acoustic Scene Clustering Using Joint Optimization of Deep Embedding Learning and Clustering Iteration	IEEE Transactions on Multimedia	Embedded deep learning with clustering	64.6% accuracy
Marek Wodzinski (2019)	Parkinson's dataset from PC-GITA database	DL Approach to Parkinson's Disease Detection Using Voice Recordings and CNN for Image Classification	Engineering in medicine and biology society	Transfer learning using ResNet	91.7% accuracy
McCallum (2019)	BeatlesTUT and SALAMI dataset	Unsupervised Learning of Deep Features for Music Segmentation	International Conference on Acoustics, Speech and Signal Processing (ICASSP)	CNN with deep feature embeddings	66% recall
Justin Hall (2019)	IRMAS dataset	An Efficient Visual-Based Method for Classifying Instrumental Audio using Deep Learning	South East Conference	Feature extraction with log spectrogram along with CNN	73.7% accuracy
Enea Ceolini (2019)	A subset of an audio set which has audio samples from YouTube videos	Audio classification systems using deep neural networks and an event-driven auditory sensor	IEEE sensors	CNN, RNN with CNN and MLP	CNN with RNN gave maximum accuracy of 81.
Naranchimeg Bold (2019)	CUB-200-2011 dataset along with data collected by the researcher	Bird Species Classification with Audio-Visual Data using CNN and Multiple Kernel Learning	International Conference on Cyberworlds (CW)	CNN with Multiple kernel learning	78.15% accuracy
Shahin Amiriparian (2019)	Manually calculated through surveying people	Audio-based Eating Analysis and Tracking Utilising Deep Spectrum Features	E-Health and Bioengineering Conference (EHB)	Mel Spectrogram as feature extractor, pre-trained CNN's and training the model using pre-trained AlexNet and VGG16 models	79.9% recall

3.3 Audio Signal Processing

Digital Signal Processing has shown a wide application in the present world e.g. Cell phones and hearing aids and a lot of research has been carried out in this filed. One of the key challenges in this field is to model the raw audio. Research by (Ahmad Moussa, 2020) used GAN architecture to model raw audio when given an audio input of a kind. The characteristics obtained from GAN were then applied to input signals so basically it went for audio to audio transition. The research also proposed a neural network architecture using causal convolution which means applying a filter at a timestamp t and discarding all other signals past time t. The evaluation of audio was done using a new structure of discriminators.

Another research in the field of Digital Signal Processing worked on developing an automated system that can identify the species of a bird (Chandu B, 2020). The team used neural networks and audio

signal processing as a base to experiment. The pre-processing was done using pre-emphasis on audio tracks, framing of each section of a waveform, cleaning, and reconstruction of audio waves. Each of these steps gave a spectrogram image which was later used as an input to convolutional layers to collect more key features. CNN was used to model the task and it gave 97% accuracy in predicting the species of a bird based on the audio available for that bird.

4 Dataset Description:

The dataset for this research was taken from Kaggle. [1]The dataset consists of a CSV file named speakers_all.csv, a reading-passage.txt file which contains a sentence which is "Please call Stella. Ask her to bring these things with her from the store: Six spoons of fresh snow peas, five thick slabs of blue cheese, and maybe a snack for her brother Bob. We also need a small plastic snake and a big toy frog for the kids. She can scoop these things into three red bags, and we will go meet her Wednesday at the train station." Every talker is speaking the same sentence in his/her accent in the English language and 2138 audio files corresponding to the speakers. There are 2140 speech samples in CSV files from 177 countries and 214 native languages.

5 Research Methodology

This research project will follow a methodology called CRISP-DM (shown in figure 9 below). The reason being we are working towards a business problem and trying to find a solution to that business problem through machine learning. We want to develop an algorithm that can help the business strategy team or marketing team to pitch their product to the right people based on their voice data. This project will also help in enhancing our speech analytics software so our first step would be to understand our business needs and then understand what kind of data do we have. Based on that we'll make our strategy of how to prepare the data, Model it, and then finally evaluate it. Within this process, we need to go back and forth based on our evaluation result to implement changes based on outputs and then finally deploy the best product in the market. A diagram of the process can be shown as

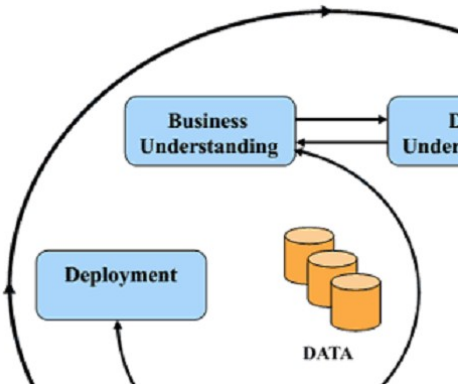


Fig 9: CRISP-DM workflow

[1] https://www.kaggle.com/rtatman/speech-accent-archive?select=recordings.

5.1 Data Pre-processing

The research was performed in google Colab where it was connected to my google drive account. The entire dataset was uploaded on google drive for research. All the required libraries were installed and then the dataset was uploaded in colab using pandas.

5.2 Removing Null values

There were 3 issues with the dataset we had and they are:

- Three columns didn't have any elements in it and were completely blank.
- A true missing file means we do not have its corresponding recording with us.
- There were 2 records were csv shows we have a recording but actually, we do not.

With the first view of the dataset, it appears that the last 3 columns have all null values so I decided to delete the last 3 columns. If we check the last column which says file_missing? We have a Boolean value of True and False there. True means that we don't have a corresponding audio file for that record and false means that we have a corresponding audio file for it. I checked the distribution of true and false and then decided to remove all the rows which have a value True. After performing this basic step when the number of records was checked it came to be 2140. Now the next step involved checking the length of audio files and we have 2138 audio files. So, using a for loop the missing files were removed and a new dataset was created where the number of rows in the csv file is the same its equivalent records in recordings.

5.3 Conversion from mp3 to wav format

All the 2138 recording of our dataset was in .mp3 format which is a compressed format of audio and it also prone to loss in audio quality while processing sound and in lower energy levels so it was converted into .wav format which is very suitable for audio classification as it is an uncompressed form of sound and it doesn't lose its quality during the pre-processing. All the converted files were stored in a separate folder from where we can use .wav files directly.

5.4 Feature Extraction on audio files

The next step which was involved in this research was extracting the length of these audio files because that would be very useful to us for future steps in our project. The length (in a sec) was calculated for each audio in our dataset by dividing the length of the signal by its sampling rate. The results were stored in a new column in our csv file and each row contains the length of its corresponding audio. Similarly, the length of our cleaned audio samples after applying signal enveloping was also created and saved into a new csv file as another column.

5.5 Exploratory data analysis

The dataset available was meant to predict the ethnicity of an individual based on its audio samples however our exploration showed that we had over 177 countries and we had over 214 different languages so to reduce the computational complexity and for some business use I decided to look over the distribution of age in the dataset and try to predict the age group of an individual based on its audio. While going over the distribution of ages it shows that the dataset is spread between minimum age of 0(which is impossible but the data had 4 such rows) and a max-age of 97 years. So, the column age was converted into 4 categories of Alpha, Gen Z, Baby Boomers, and millennials. The alpha contained age group less than 5 years and then Gen Z was the one between the age of 5 and 25 years. Baby Boomers were people between 25 to 41 years of age and people above 41 were termed as Millennials. All these 4 generations were one hot encoded into categorical variables under a new

column name age_group. Now an attempt was made to check the class distribution of all these 4 generations and the result of it was:

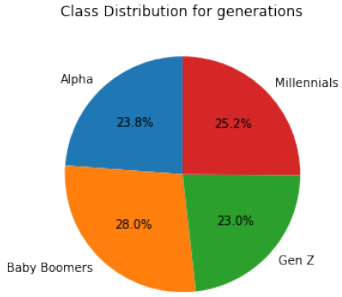
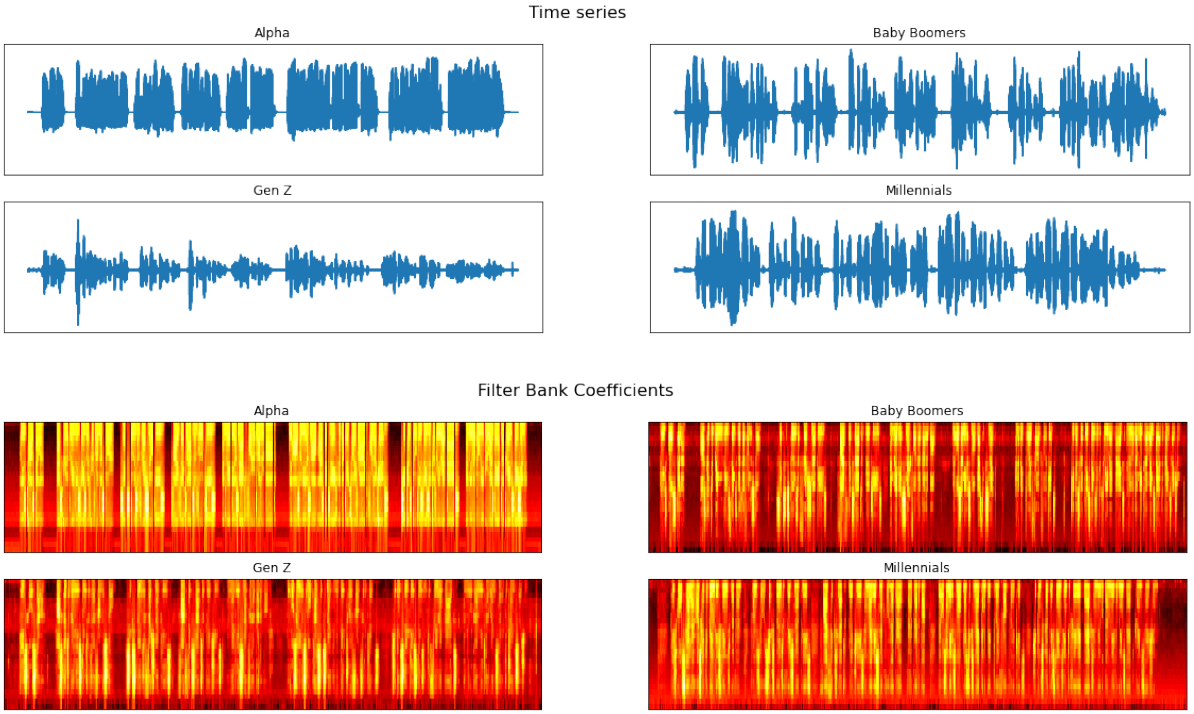


Fig 10: Class distribution of labels

As we can see in figure 10 that all the four generations are evenly distributed and there is no class imbalance.

5.6 Visualization of waveforms of different generations

The next part of our data exploration involves visualizing the waveforms and Fourier transform of features generated from audio data of all 4 generations and they are represented in figure 11:



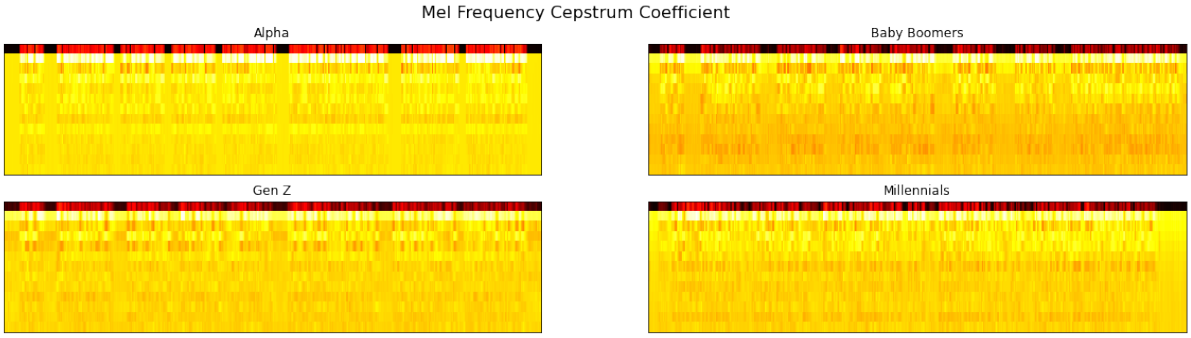


Fig 11: Images of a waveform, filter bank, and mfcc from one audio of each label

5.7 Calculating Signal Envelope

After checking the class distribution and visualizing the waveforms of all the four generations we can see a lot of dead spaces in the waveform. These dead spaces are of no use to us because it won't help us in predicting anything so it's important to get rid of all the dead spaces. For that, we will use Signal enveloping. Signal enveloping is a technique where we check the estimation of magnitude to see if it falling(as shown in figure 12). This means that if the audio is falling too low then it is considered as the audio getting died out and it's not important and this will be determined using a threshold value. To do that the author created a mask that will be supported by NumPy Boolean indexing. So, the mask will have true, false values which can be used to reduce or remove empty portions of the data. A series was created and a signal was passed through it. The signal will be negative most of the time so we will take an absolute value of that. In addition to moving all the absolute value, we will use a rolling window over our data. The reason why it is done is that we don't just want to check over the value of our data, we want to use a rolling window that goes over the entire audio and then takes the mean of the values to make sure the entire signal is completely dropped out. The other reason is that the signal crosses the x-axis very frequently and lots of value will get removed and our data can be completely ruined. The window is equal to a tenth of our collection rate. That means I have taken window size to be a tenth of a second and it will be passed like a min period that is the minimum number of values that we need in our window to create a calculation. Min period will also avoid the number of NAN values because the window is generally positioned at the center starting from the first value and it will move towards left and won't see anything in that array and then return a bunch of null value. As we move the window across the signal, we will take the mean of it. The result would be a collection of all the values that are in that window as we pass along the signal, we will check that if that mean is greater than the threshold value, we will apply the envelope to the signal else we will just drop it. Mathematically it can be explained as a function of space and time. The threshold value taken for this research was 0.005.

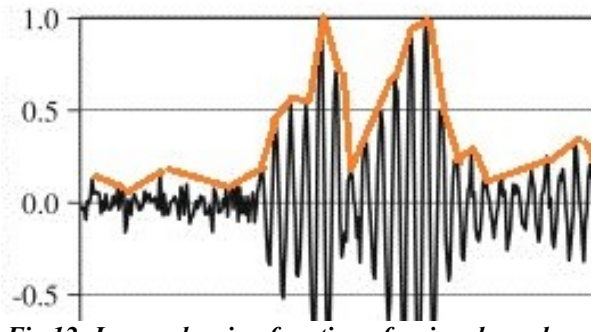


Fig 12: Image showing function of a signal envelope

After performing masking, we will plot the waveform, filter energies, and MFCC's of our signals and see the difference.

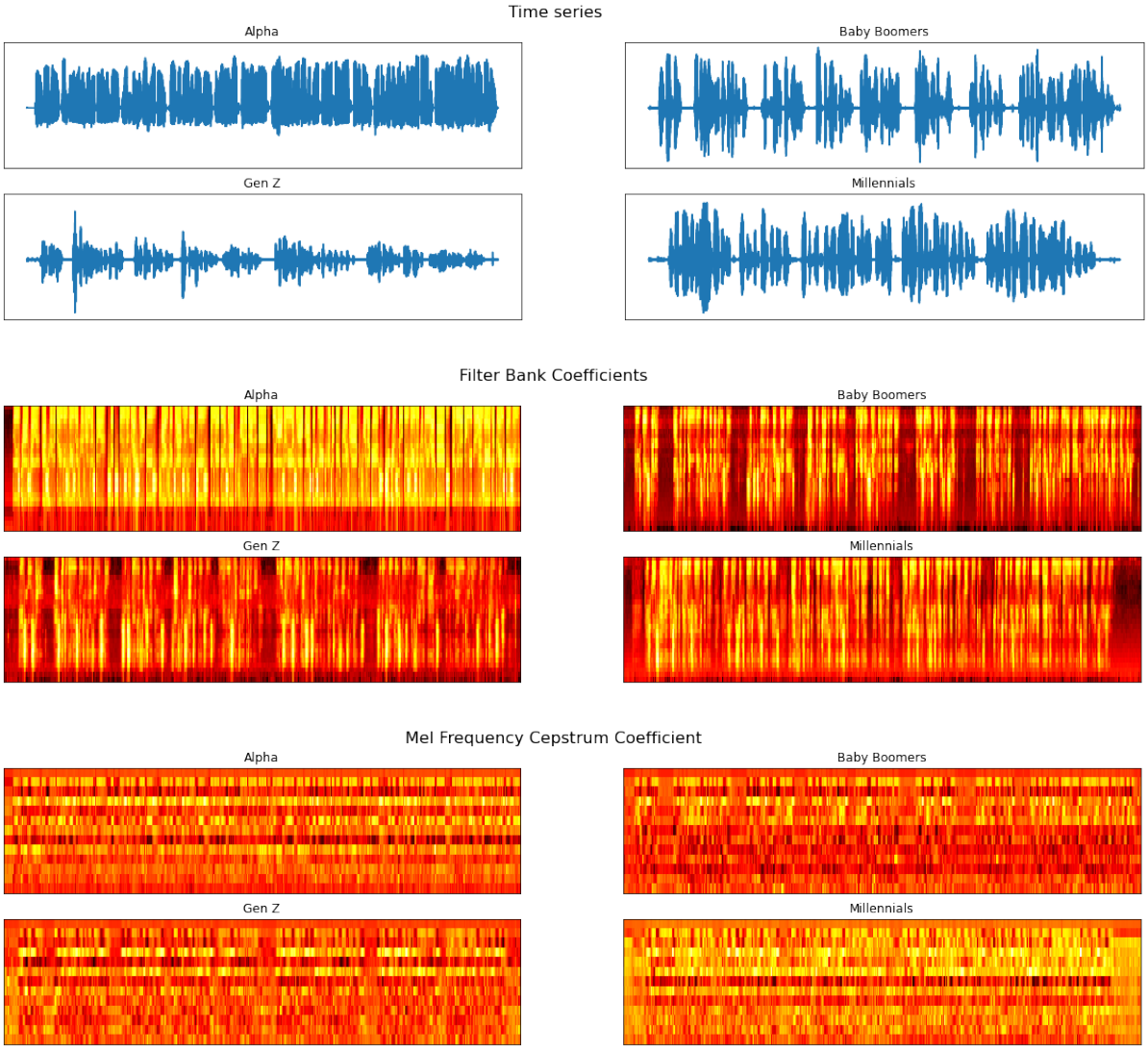


Fig 13: Images of a waveform, filter bank, and mfcc from one audio of each label after removing dead air

We can work with a threshold value and experiment with it to obtain a better result for it. The process done above is also called as noise floor detection.

5.8 Downsampling and putting the mask over our audio

The next step is downsampling the clean audio after removing the noise and then applying the same mask over the down-sampled audio. Those samples would be used in the future for our modeling. We are downsampling the audio to a sampling rate of 16000 from 44100 because we don't have much information available on the higher frequency range. We will write the downsampled audio, after Boolean indexing it with the envelope we created, to our clean directory for future use.

5.9 Preparation for our model

In the next step for our pre-processing, we created our X matrix and y matrix with all of our clean audio samples along with their corresponding classes. Audio is different from other types of data because it is being sampled so frequently that we have to create an arbitrary length of time that we want to build our sample with. This research has used a second of audio for each sample. We can choose a lower fraction of a second as well to capture more variation but we choose a second to reduce the computational complexity. The next step which was done is randomly sampling along the length of our audio files and then we took a one-second chunk of that audio. Now in our case, it is

sampled in a way where if we move through our audio file and keep chunking it bit by bit we will move forward with every second. We will also create a probability distribution because we need to convert the values of our classes so that when we add them it becomes 1. This is done by dividing the class distribution by the sum of the class distribution.

5.10 Random sampling from audio data

Random sampling from our audio data was done for the next step. In this step, we will choose our classes based on the probability distribution that we already created. We will also generate a random class that follows the probability distribution we created before. We will take a random generation and our probability is going to be equal to our probability distribution.

5.11 Creating an empty list of X and y

The reason for doing this is because it is easier to pre-process audio in form of a list and then we will later convert the list into a NumPy array.

5.12 Normalising the audio

we will create a min and a max value and to determine the scaling we can either use a min-max scaler and figure out how we can normalize our values but this research used a different technique where min was initialized with the highest value a float can take on and max was initialized with the lowest value a float can take on to make sure that these get updated. Another reason is that this project will use neural networks and the input of the neural networks has to be normalized between 0 and 1 so we should know what is the min and the max values.

There is another level of randomness within the actual file that we pick so actually we will pick our file randomly based on the class allocated by us. We check the label and then we check to see if that equals the class that we just picked and take the index of the class which will be the filename. So basically, it will produce a file belonging to the class. Now the file we have generated, we will go ahead and read that file. We will read the file from our clean directory. This step is followed by reading the age_group where we will index at the filename for the specific age group. After this, we will create a random index where we start to randomly pick value based on the length of our audio and will sample it at its index, take everything for a second after that index and that will be my X distribution which will pair with the label we just created which is age group. The range will be from 0 to the length of the file and then subtract one sec of the file at the end to make sure we don't reach to end of the file and run out of data point and we're just going to stop there. The sample was generated by taking an index from the random index we picked and then a second after that. This will give our first X sample and then we will compute our MFCC and save it in our X sample. We will pass the sample, rate, the number of MFC Coefficient is taken as 13, 26 filter bank energies, 512 FFT. This will also help us in finding our dynamic min and max value. We will enter our entire matrix and then grab a minimum and a maximum value from the matrix. Once it is done, we compared the minimum and maximum value from our existing min and max values. If we have a new minimum and maximum value, we will update that value. Once the values are updated, we will start appending the samples to our actual list which we created in beginning. Similarly, we will do the same thing by appending our age group to y list.

5.13 Converting X and y matrix into an array from a list

X and y lists need to be converted into an array because the neural network takes input which is in form of an array. Now, this was done using NumPy.

5.14 Normalising the matrix of feature

The X feature was normalized by rescaling it between 0 to 1. It was performed by subtracting the minimum value from X and then dividing it with the difference between the maximum and minimum value.

5.15 Reshaping X and y feature for giving it as an input to the convolutional and recurrent model.

Reshaping of our X array is important because we will be using a convolutional model and a recurrent neural network model. The shape of array X would be 4 dimensional to use it in a Convolutional network. Its shape looks like (num_features, time, num_of_mfcc, channel) whereas for Recurrent neural net we need a 3-Dimensional array whose shape looks like (num_features, time. num_of_mfcc). We remove the channels dimension while reshaping X for RNN.

5.16 One hot encoding of y array

Y matrix consists of our class and since we have 4 classes with which we're dealing which are categorical to perform our audio classification. The reason why we do this is that when we do the cost function for these neural networks, we'll have categorical cross-entropy and 4 classes between 0-3. Hot encoding them converts into a matrix.

All these techniques gave me around 115916 features with 99-time dimensions and 13 coefficients which we will use in our model.

5.17 Splitting the dataset into training and test set.

The final step in our pre-processing is splitting our X and y features into training and test set. It was performed using train_test_split from sklearn. The data was split in an 80:20 ratio where 80% of data was given to the training set and rest 20% to our test set.

6 Modelling using Convolutional Neural net

A convolutional neural network is a very good deep learning modeling technique that is used on image data. CNN's perform better than multilayer perceptron with fewer parameters than the dense layer. The data on which we're working is audio data. So, the question which arises here is can CNN's be used on audio data. And the answer is yes because we went through an intensive pre-processing technique through which we created our X and y feature matrix using Mel Frequency Cepstral Coefficients. It is also like an image however positioning of the features is more important than it is in the image. A Fully Connected Convolutional neural network was used and a structure was created whose purpose was to classify our age groups based on the audio samples that we have for the corresponding age group.

6.1 Architecture of our CNN

The convolutional neural network which we used has a 2D kernel and convolutional filter. Our convolutional filter used size of (3*3) throughout the experiment. Our input shape was (99, 13, 1) where 99 shows the number of temporal frames, 13 is the number of Mel coefficients and 1 is the number of channels for our input. ReLU was used as an activation function. Pool size was taken as (2*2) during the entire test to downsample our MFCC's. 4 convolution layers were used along with 2 dense layers and the final output layer. Softmax was used as the activation function because we had 4 classes to classify. The optimizer used was Adam, loss function was calculated by using categorical cross-entropy because we had categorical output, and performance was evaluated using accuracy. The model summary is shown in the table below.

6.2 Modelling using Time Distributed-LSTM

RNN which also stands for recurrent neural networks is another modeling technique that is used in this project. In RNN the ordering of data is very important. RNN's are very good at classifying sequential data. In RNN's the next step depends on the output produced by the previous step. Audio data resonates very well with RNN because audio data has a temporal component in it. The parameters we have created are in series of points created at 1-sec interval of time. E.g. our waveform is like a univariate time series which means we have only one measure taken at each interval. So the shape of the data with a waveform expressed as a time series is (sample_rate*time, 1). 1 shows a single dimension of the waveform which is amplitude. This project has created its parameters using MFCC's which is a multivariate time series because we're taking more measurements for each interval and the relative dimension is given by the overall number of intervals. The total parameters are calculated by dividing the sample rate by hop length. The second dimension is the number of MFCC's which is 13, The issue with simple RNN is that it doesn't have a long-term memory which means it cannot use the information which occurred in distant past and is also unable to learn pattern if it has long dependencies. That's why we used LSTM (Long Short Term Memory) which has a memory cell that enables us to use a longer-term of patterns. An LSTM cell contains a simple RNN cell with a tanh dense layer, a second state vector also called long-term memory which stores longer-term patterns followed by forgetting gates, input gates, and output gates which acts as filters which decide things to forget, input and output. The shape of our data is (num of samples, time, mfcc's) that is (n, 99, 13). Our input shape for the model was (time, mfcc's) which was (99, 13).

6.3 Architecture of LSTM model

The architecture we used 3 LSTM layers followed by 4-time distributed layers and our final output layer. The input shape is a 2D input where we're leaving the channels from the input we used for the convolutional layer. ReLU is used as an activation function for the time distributed layer and softmax as the activation function for the output layer because we're doing multiclass classification. The loss function was binary cross-entropy and the metrics used were accuracy. The summary of the model is shown in figure 14 below.

```
_________________________________________________________________
Layer (type)                 Output Shape
=================================================================
conv2d_4 (Conv2D)            (None, 99, 13,
_________________________________________________________________
activation_6 (Activation)    (None, 99, 13,
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 97, 11,
_________________________________________________________________
activation_7 (Activation)    (None, 97, 11,
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 48, 5,
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 48, 5,
_________________________________________________________________
activation_8 (Activation)    (None, 48, 5,
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 46, 3,
_________________________________________________________________
activation_9 (Activation)    (None, 46, 3,
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 23, 1,
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape
=================================================================
lstm (LSTM)                  (None, 99, 256)
_________________________________________________________________
lstm_1 (LSTM)                (None, 99, 128)
_________________________________________________________________
lstm_2 (LSTM)                (None, 99, 128)
_________________________________________________________________
dropout (Dropout)            (None, 99, 128)
_________________________________________________________________
time_distributed (TimeDistri (None, 99, 64)
_________________________________________________________________
time_distributed_1 (TimeDist (None, 99, 32)
_________________________________________________________________
time_distributed_2 (TimeDist (None, 99, 16)
_________________________________________________________________
```

```
_________________________________________________
activation_11 (Activation)   (None, 128)
_________________________________________________
dropout_2 (Dropout)          (None, 128)
_________________________________________________
dense_10 (Dense)             (None, 4)
=================================================
```

Fig 14: Summary of CNN and RNN-LSTM model showing an architecture

Early Stopping: Early stopping was used in both the models to avoid the model from overfitting. It was called while fitting the model. We monitored the validation loss with the patience of 5 because we used almost 50 epochs while restoring the best weights. This helps in the termination of the epochs when there is no reduction in the validation loss after a certain patience level.

ReduceLRonPlateau: Also called as reduce learning rate on plateau proved to be very useful when our metrics stopped showing any improvement. This function keeps on reducing the learning rate of our model if it shows no improvement after a certain patience level and then finally executes. It also monitored validation loss with the patience of 2, factor of 0.2, and min learning rate of 1e-6.

7 Design Specification

The design of our workflow is presented in figure 15 below

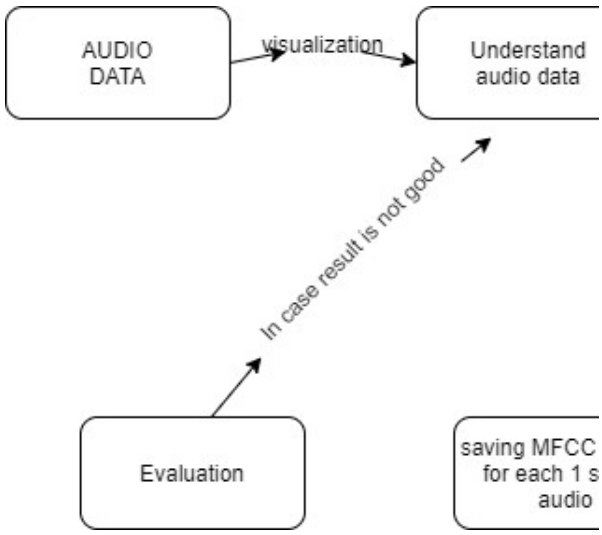


Fig 15: design specification diagram

8 Implementation

This research, in the beginning, was very difficult to get implemented due to the computational complexity so a lot of hyperparameter tuning needs to be done so that the program could be easily executed. The techniques being used are very easy to implement on an industrial level. The reason why it becomes easy to implement these models is because of the extensive pre-processing which has been done on the data. The size of our cleaned dataset was less than that of the original data.

9 Evaluation

Evaluation technique is generally referred to as a comparison of actual value to our predicted value. It helps us to verify the performance of our model on real-life data where the model is not trained on. This project used two models to experiment and they were Fully connected deep convolutional neural

network and a Time distributed Long Short-Term Memory Recurrent neural network. This research used evaluation methods which are accuracy, loss, precision, recall, and F1 score.

Accuracy: Accuracy in simple terms can be defined as how free our model is from the errors. Mathematically it is defined as a ratio of correctly predicted values to the total observation we consider.

$$\text{Accuracy} = \frac{TP+}{TP+FP+}$$

.

Loss: Loss is a value that is calculated by our loss function which is categorical cross-entropy in our case. The function of loss is to minimize the error. Our model tries to calculate the perfect weight using the data we enter to train the model. These weights keep changing so that the next prediction can have a reduced error than the previous one. The loss of a model should decrease when the model starts learning from our data.

Precision: Precision can be defined as TP/(TP + FP). It is another way to evaluate a model.

Recall: Recall value can be defined mathematically as TP/(TP+FN). This also evaluated model performance.

F1 Score: This is the final metric used represented as 2*TP/(2*TP + FN+ FP).

Table 3: Evaluation result table

Model/ Metric	Accuracy	Loss	Precision	Recall	F1
FC-CNN	62.45%	0.7572	62.45%	62.45%	62.45%
TD-LSTM	66.07%	0.3271	66.07%	66.07%	66.07%

The graphs of our test results are shown below:

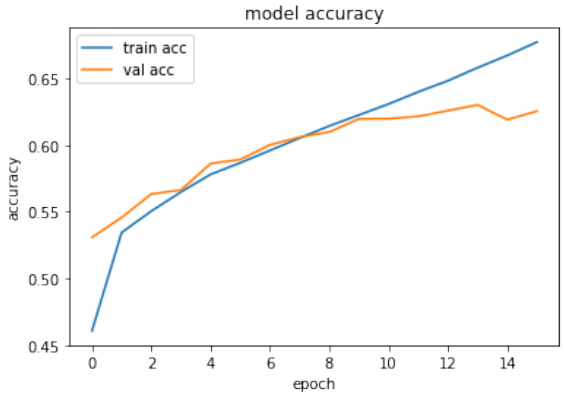


Fig 16: Model accuracy for FC-CNN

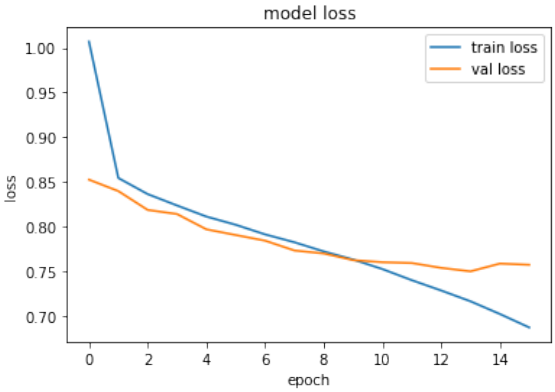


Fig 17: Model loss for FC-CNN

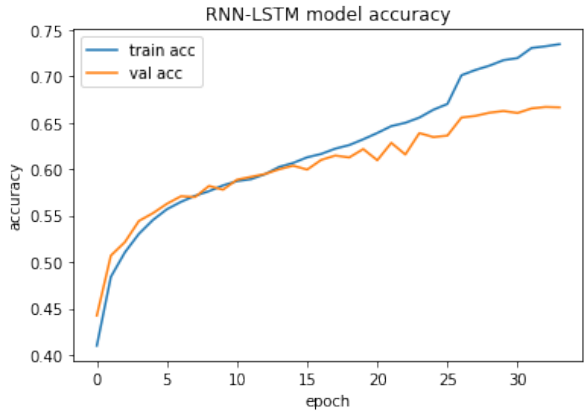


Fig 18: Model accuracy for RNN-LSTM

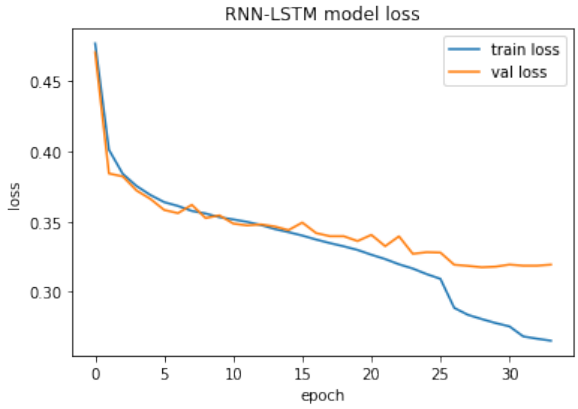


Fig 19: Model loss for RNN-LSTM

10 Conclusion and Future Work

This research was performed on a dataset called speech accent archive which is available on Kaggle. This research went through an extensive audio pre-processing technique to generate our feature X and y from our audio data. The dataset was annotated to make sure the model learns effectively. This research project used the Mel Frequency coefficient as a feature extraction technique which is a Discrete Cosine Transform of Log Mel powers. This research used a signal enveloping technique to denoise the data and pre-process it in a way where we can get rid of dead spaces. The audio data was also downsampled to 16000hz because there was not much information available at higher frequencies. The frame size being selected was 1 second for the length of the audio and then the classification was performed from the mfcc features extracted from those frames. This research used a Fully Connected deep convolutional neural network and Long Short-Term Memory Recurrent neural networks with time distributed layers. Applying mfcc features generated 115916 samples from our audio data which was then divided into a ratio of 80:20 where 80% of the data was given to the training set and 20% to the test set. The performance was evaluated using accuracy and loss. The first set of experiments were performed using a Fully connected convolutional neural network. Multiple series of tests were conducted by changing the number of epochs, increasing convolutional filters, hyperparameter tuning the optimizers and finally, early callbacks and ReduceLRonPlateau was used and the final experiment gave a validation accuracy of 62.55% with a loss of 0.7572. Since audio data has time dimension attached to it so a need was felt to test the data over a Long short term recurrent neural network with time distributed layers. As mentioned in the result Long Short-Term Memory Recurrent Neural Net gave validation accuracy of 64.51%. So, as we can see RNN-LSTM with time distributed layers outperformed Fully connected CNN as is proved to be better in classifying the age groups based on the corresponding audio data.

10.1 Future work

There are a few things which this research would like to carry out in the future which might help in increasing the accuracy of our model. The dataset used for the research contained only 2138 audio samples which are very less for our model to learn features from those audios. So, increasing the data size will help in enhancing the performance of our model. Audio data as we know needs a lot of pre-processing and due to time complexity, the research was not able to touch on all the pre-processing techniques which might have improved the performance. E.g. data augmentation techniques like time stretching and time-shifting would help a lot in generating more synthetic data so this would help in improving the performance of our model. Another thing which could be done is that reducing our frame size from one second of the length of audio to one-tenth of a second. this research tried to use that but unfortunately was not able to proceed with it because of hardware challenges and time complexity in computing a large number of features and then using it to create our model. This project

would also get benefit in using transfer learning as a classifier or a feature extractor. This has been a topic in which the author of this project would like to research because there were multiple challenges that the author faced and was not able to come up with a solution within the time frame of this project. Few of them were the input size which is different for the samples which we have created and it's different on which a pre-trained model is trained. Similarly, the pre-trained model is trained over 3 channels and our feature had single channels. The audio data was downsampled from the original sampling rate we had but further downsampling will help in extracting more key features from audio and will improve the performance of our model. If this research had more time these are the key areas where we would like to work and see the performance.

11 References

Abigail Copiaco, 2019. Scalogram Neural Network Activations with Machine Learning for Domestic Multi-channel Audio Classification. *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT).*

Ahmad Moussa, 2020. Audio Translation with Conditional Generative Adversarial Networks. *International Conference on Artificial Intelligence in Information and Communication (ICAIIC).*

Chandu B, 2020. Automated Bird Species Identification using Audio Signal Processing and Neural Networks. *International Conference on Artificial Intelligence and Signal Processing (AISP).*

Enea Ceolini, 2019. Audio classification systems using deep neural networks and an event-driven auditory sensor. *IEEE SENSORS.*

Justin Hall, 2019. An Efficient Visual-Based Method for Classifying Instrumental Audio using Deep Learning. *Southeast Conference.*

Lillian Le, 2019. Using Transfer Learning, SVM, and Ensemble Classification to Classify Baby Cries Based on Their Spectrogram Images. *IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW).*

Marek Wodzinski, 2019. Deep Learning Approach to Parkinson's Disease Detection Using Voice Recordings and Convolutional Neural Network Dedicated to Image Classification. *41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).*

McCallum, M. C., 2019. Unsupervised Learning of Deep Features for Music Segmentation. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

Naranchimeg Bold, 2019. Bird Species Classification with Audio-Visual Data using CNN and Multiple Kernel Learning. *International Conference on Cyberworlds (CW).*

Paul Best, 2018. Musical Gesture Recognition Using Machine Learning and Audio Descriptors. *International Conference on Content-Based Multimedia Indexing (CBMI).*

Rong, F., 2016. Audio Classification Method Based on Machine Learning. *International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS).*

Sercan Sarman, 2018. Audio based violent scene classification using ensemble learning. *6th International Symposium on Digital Forensic and Security (ISDFS).*

Shahin Amiriparian, 2019. Audio-based Eating Analysis and Tracking Utilising Deep Spectrum Features. *E-Health and Bioengineering Conference (EHB).*

Wu, D., 2019. An Audio Classification Approach Based on Machine Learning. *International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS).*

Yanxiong Li, 2020. Acoustic Scene Clustering Using Joint Optimization of Deep Embedding Learning and Clustering Iteration. *IEEE transactions on Multimedia.*

Gustavo Noffs, Frederique M C Boonstra, Thushara Perera, Scott C Kolbe, 2020. Acoustic Speech Analytics Are Predictive of Cerebellar Dysfunction in Multiple Sclerosis, National center for biotechnology information doi: 10.1007/s12311-020-01151-5

Xiang Li, Xiaodong Jia, Qibo Yang & Jay Lee, 2020. Quality analysis in metal additive manufacturing with deep learning. Journal of Intelligent Manufacturing.

Snehlata Barde, Veena Kaimal, 2020. Speech recognition technique for identification of raga, Application to Neural Engineering, Robotics, and STEM.

Jie Jiang, Harry Haoxiang Wang, 2020, Application intelligent search and recommendation system based on speech recognition technology, International Journal of Speech Technology.

Yonas Woldemariam, 2020, Transfer Learning for Less-Resourced Semitic Languages Speech Recognition: the Case of Amharic, European Language Resources association