Speaker Identification from Voice Signals Using Hybrid Neural Networks

Major Project II

SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by:

HARSHIT BHATT
2K19/ISY/19

Under the supervision of

DR. RITU AGARWAL
ASSISTANT PROFESSOR

DEPARTMENT OF INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi college of Engineering)
Bawana Road, Delhi-110042

OCTOBER, 2021

CANDIDATE'S DECLARATION

I, Harshit Bhatt, 2K19/ISY/19 hereby certify that the work which is presented in the M.Tech Thesis/Dissertation entitled "**Speaker Identification From Voice Signals Using Hybrid Neural Networks**" in fulfilment of the requirement for the award of the Degree of Master of Technology in Information Systems and submitted to the Department of Information Technology, Delhi Technological University, Delhi is an authentic record of my own, carried out during a period from August 2020 to June 2021, under the supervision of Dr. Ritu Agarwal.

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other Institute/University.

HBhatt

Place: Delhi

Date: October 25, 2021

Harshit Bhatt

2K19/ISY/19

CERTIFICATE

I hereby certify that the M. Tech Thesis/Dissertation titled "**Speaker Identification From Voice Signals Using Hybrid Neural Networks"** which is submitted by **Harshit Bhatt, Roll No. 2K19/ISY/19 Information Systems, Delhi Technological** University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: October 25, 2021

Dr. Ritu Agarwal
SUPERVISOR
ASSISTANT PROFESSOR
DEPARTMENT OF
INFORMATION TECHNOLOGY

ACKNOWLEDGEMENTS

I am very thankful to Dr. Ritu Agarwal (Assistant Professor, Department of Information Technology) and Prof. Kapil Sharma (Head of Department, Information Technology (M. Tech)) and all the faculty members of the Department of Information Technology at DTU. They all provided me with immense support and guidance for the project.

I would also like to express my gratitude to the University for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions.

I would also like to appreciate the support provided to us by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

ABSTRACT

Identifying the speaker in audio visual environment is a crucial task which is now surfacing in the research domain researchers nowadays are moving towards utilizing deep neural networks to match people with their respective voices the applications of deep learning are many-fold that include the ability to process huge volume of data robust training of algorithms feasibility of optimization and reduced computation time. Previous studies have explored recurrent and convolutional neural network incorporating GRUs, Bi-GRUs, LSTM, Bi-LSTM and many more[1]. This work proposes a hybrid mechanism which consist of an CNN and LSTM network fused using an early fusion method.

We accumulated a dataset of 1,330 voices by recording through a python script of length of 3 seconds in .wav format. The dataset consists of 14 categories and we used 80% for training and 20% for testing. We optimized and fine-tuned the neural networks and modified them to yield optimum results. For the early fusion approach, we used the concatenation operation that fuses neural networks prior to the training phase. The proposed method achieves 97.72% accuracy on our dataset and outperforms all existing baseline mechanisms like MLP, LSTM, CNN, and RNN. This research serves as a contribution to the ongoing research in speaker identification domain and paves way to future directions using deep learning.

CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

The task of speaker identification entails using a machine to determine a speaker's identity. To recognize a voice, people as well as machines must be familiar with it. Testing is the second step in the identifying process. Specifically, the process of scrutiny, which entails associating unrecognized vocalizations with coaching data and generating identification. The target speaker of a verify vocalization is referred to as the speaker of the check vocalization. Various speech parameterizations with victimization formant possibilities have recently piqued people's interest. The development of acoustic spectrum formant frequencies is extremely important. Formants, on the other hand, are extremely difficult to detect from a speech signal and are frequently missed[1]. As a result, formant-like alternatives will be employed instead of predicting resonant frequencies.

Depending on the application, the world of speaker recognition is divided into two parts. The first is identification, and the second is verification. Identification's purpose is to match the input voice sample to the available voice samples. We also look at the supplied voice sample to see who is speaking in speaker verification.

1.1 Motivation

The ability to utilize the speaker's speech to verify their identity is made possible through speaker identification. Management of services such as voice dialling/calling, telephone banking, information access services, voice mail, security management for guidance areas, and remote computer access and many others.

1.2 Problem Statement

Understanding the way to acknowledge advanced, unstructured, and high-dimensional voice/speech/audio information is one in all the best challenges of our time.

Traditional (GMMs) approach suffers from associate degree inherent assumption of one-dimensionality in speech signal dynamics. Such approaches are at risk of overfitting and have issues with generalization.

1.3 Objective

The goal of speaker identification is to use a machine to figure out a speaker's identity based on the sound of his or her vocal. No identity is claimed by the user.
In this work, we tend to project a thought for identity of a speaker by machine on the idea of his/her voice with the assistance of –

Deep Learning approach

- MLP(Multilayer Perceptron)
- CNN(Convolutional Neural Network)
- RNN(Recurrent Neural Network)
- LSTM(Long Short Term Memory)

1.4 Automatic Speech Recognition(ASR)

The spoken signal provides various levels of information to the listener. On the basic level, speech sends a message through words. On many levels, speech, on the other hand, offers information about the language being spoken and, as a result, the speaker's mood, gender, and identity. Unlike speech recognition, which seeks to recognise the words said in the speech, automatic speaker recognition systems aim to extract, classify, and recognise title speaker, identify knowledge within the voice signal[2].

Two additional basic activities fall under the umbrella of speaker recognition:

1.4.1 Speaker Identification

Speaker identification is the process of detecting which Group is speaking among a number of well-known voices or speakers. If the unknown person does not claim to be who they say they are, the system should categorize them as 1: N. Because the unknown voice is expected to return from a hard and fast group of more well-known speakers, the task is commonly referred to as closed-set identification.

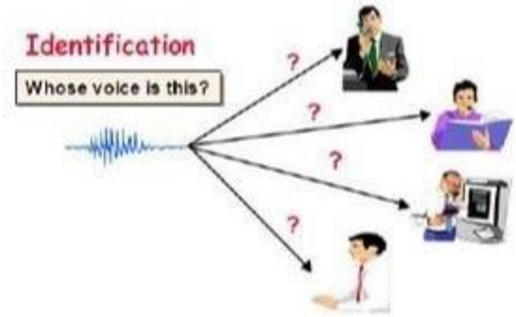


Figure 1.1 Speaker Identification (Figures courtesy of Douglas Reynolds, MIT LincolnLabs)

1.4.2 Speaker Verification

The process of determining whether or not an individual of particular organization that he or she claims to be (a yes/no conclusion) is known as speaker verification (also known as speaker authentication or detection). This work is commonly referred to as an open-set task because it is believed that imposters (those fraudulently claiming to be legitimate users) will be unfamiliar with the system. By adding a "none of the above" option to the closed-set identification job, the two tasks can be combined to form open-set identification.

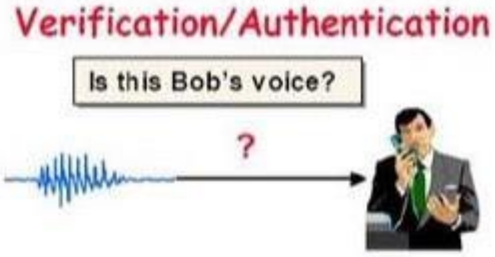


Figure 1.2 Speaker Verification (Figures courtesy of Douglas Reynolds, MIT Lincoln Labs)

1.5 Deep Learning Technique

1.5.1 Multilayer Perceptron

There are at least three layers of nodes in a Multilayer Perceptron (MLP): an associated input layer, a hidden layer, and an associated output layer. Aside from the input nodes, each node might be a nerve cell with a nonlinear activation pattern [3]. MLP employs backpropagation as a supervised learning technique during training. The multiple layers and non-linear activation distinguish MLP from a linear perceptron.

It will distinguish between information that is linearly divisible and knowledge that is non-linearly divisible.

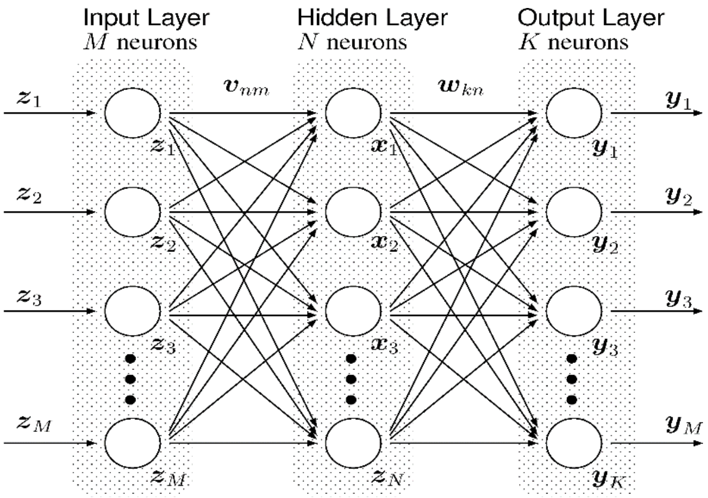


Figure 1.3 Typical Multilayer Perceptron Architecture (Figures courtesy MDPI)

1.5.2 Convolution Neural Network

Deep neural networks have made breakthroughs in a variety of pattern identification problems, including computer vision and voice recognition, in the last few of years[4]. A unique kind of neural network known as a convolutional neural network was one of the crucial aspects that resulted in these outcomes.

Convolutional neural networks (CNNs, or ConvNets) are a form of deep neural network that is used in deep learning to investigate visual representational processes and speech recognition. CNNs use a form of multilayer perceptron that is meant to require as little preprocessing as possible. Because to its shared-weights design and translation in variance features, they're also known as shift invariant or area invariant artificial neural networks (SIANN).

The architecture of the animal visual area is matched by the property pattern between neurons in convolutional networks, which was guided by biological processes. Individual tissue neurons in animals only respond to inputs in a small area of the visual field called the receptive field. Multiple neurons' receptive areas partially overlap, concealing the whole vision field.

Convolutional neural networks are one of the most significant breakthroughs in the area of machine vision. They excelled prior computer vision and achieved progressive outcomes. These neural networks have been shown to operate in a range of environments and applications in the real world, including:

- Face recognition, image categorization, object detection, segmentation.
- CNN-based vision systems are used in self-driving automobiles..
- Voice Recognition.
- And many more.

A CNN is comprised of an input layer and an output layer, as well as many hidden layers. Convolutional layers, RELU layers (activation function), pooling layers, fully connected levels, and social control layers are commonly found in the hidden layers of a CNN.

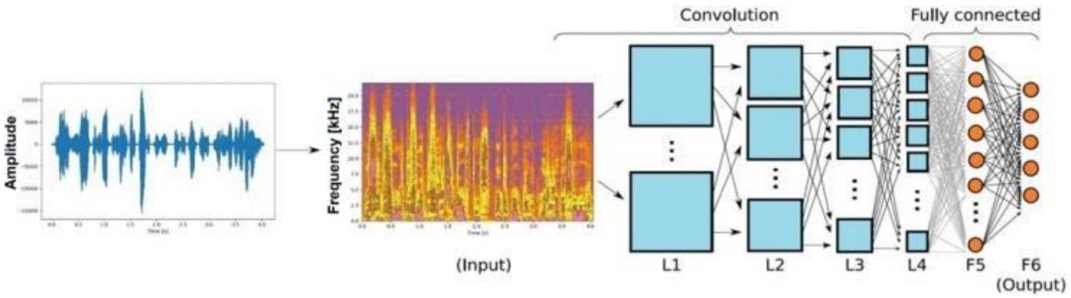


Figure 1.4 Architecture of the CNN (adapted from [16])

1.5.3 Recurrent Neural Network

Recurrent Neural Networks (RNN) are a resilient and robust kind of neural networks that are now among the most promising algorithms available since they are the only ones with internal memory.

RNNs can remember important details about the input they received because to their internal memory, allowing them to anticipate what will happen next with extreme precision.

This is why, when compared to other algorithms, they're the most preferred formula for serial knowledge like statistics, voice, writing, money info, audio, video, weather, and so on since they'll develop a much deeper grasp of a sequence and its context.

“When there is a series of data, the temporal dynamics that link the data are more significant than the spatial content of any individual frame.”

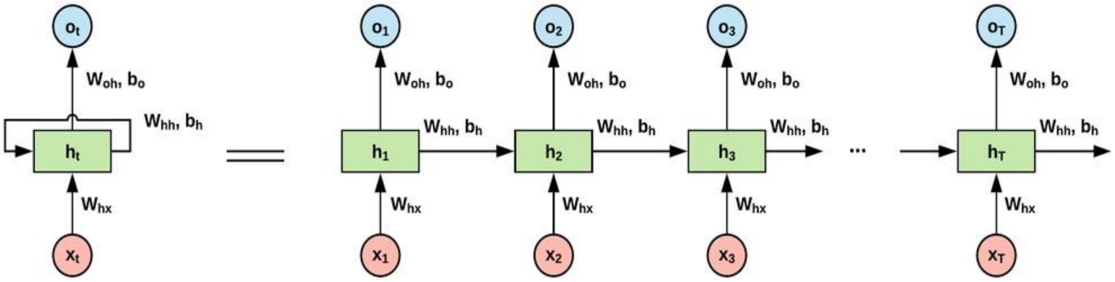


Figure 1.5 Sample RNN structure (Left) and its unfolded representation (Right)

1.5.4 Long Short-Term Memory

Long Short-Term Memory Networks - also known as "LSTMs" are a kind of RNN that may learn semi-permanent dependencies. They were introduced by Hochreiter & Schmidhuber [5]and many others developed and promoted them in the work that followed. They perform very effectively on a broad range of problems and are now in widespread usage[6].

LSTMs are specifically intended to prevent the disadvantage of semi-permanent dependence. It is not one of their struggles to be informed that memorizing material for extended periods of time is their natural habit[7].

A series of repeat modules in a neural network is the form of all continuous neural networks. This repetition module in typical RNNs may have a very simple structure, such as one tanh layer.

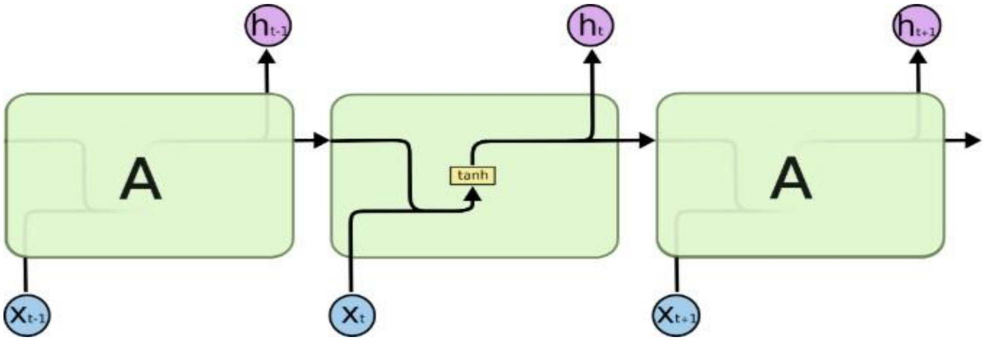


Figure 1.6 The repeating module in a standard RNN contains a single layer

Even though LSTMs have a chain-like structure, the repeating module has a completely distinct structure. Instead of just one neural network layer, there are four, all interacting in a very unique way.

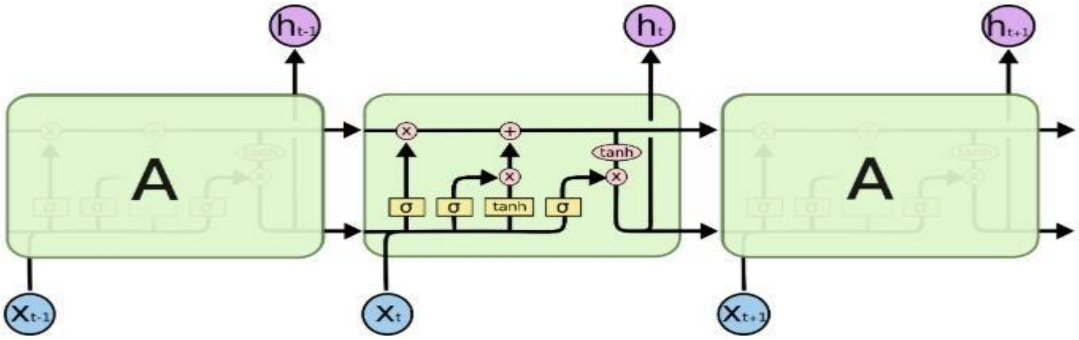


Figure 1.7 The repeating module in an LSTM contains four interacting layers

1.5.5 LSTM-CNN (Hybrid Model):

The idea to use this hybrid model comes from the high retentive power of LSTM RNN. Such Voice recognition task rarely utilize RNN networks, we exploit the memory advantage of LSTM as they are capable to store information for longer duration. LSTM selectively keep information that are useful to them and discard any useless features using forget gates. This automatic feature selection and robust memory form the basis of proposed model.

Convolutional neural networks are proved highly efficient in speech recognition task the layers of the network can easily select features from the input data and process them over convolution. In the propose model we combine the usefulness of LSTM and CNN by merging them using concatenation which is a method of early fusion.

1.6 Summary

In this Chapter we have encapsulated the introduction part of Deep learning techniques like Multilayer perceptron(MLP), Convolution Neural Network(CNNs), Recurrent Neural Network(RNN), Long short term Memory(LSTM) and the proposed the hybrid model.

The rest of the report is organized as:

Chapter 2 provides the introduction of existing solutions and the drawbacks of that.

Chapter 3 provides the proposed solution of Speaker Identification.

Chapter 4 presents implementation details and provides the results from the working model.

Chapter 5 finally concludes the project work.

CHAPTER 2

LITERATURE SURVEY

While considering a few Speaker Identification options, greater emphasis was placed on recently presented techniques because they address the majority of the shortcomings of previously proposed approaches in this domain.

2.1 Related work

Luckic et al.[8] showed how Deep learning, notably in the form of convolutional neural networks (CNNs), has resulted in significant advances in computer vision and related fields in recent years. This advancement can be attributed to a trend away from planning options and ensuant individual sub-systems and toward learning options and recognition systems that work from start to finish with practically unprocessed data. Handcrafted process chains, such as MFCC choices and GMM-based models, are still popular for speaker cluster. Simple spectrogram is given as input to a CNN and the best style of these networks is investigated for identification and clustering Moreover, Tried to elaborate on the subject of how to transfer an identification-trained network to a speaker cluster. Standard TIMIT dataset is used to obtain, results that are equivalent to the state of the art– without the need for handmade alternatives. An accuracy of 97.0% was achieved on the TIMIT dataset.

Trujillo-Romero and Caballero-Morales [9] performed speaker identification using neural networks on an FPGA. They presented the implementation of a speech recognition system in this work. A neural network was used as a recognizer element, and Linear Prediction Coding (LPC) was used as a vector input for the net. This four-layer neural network was built using a Digilent Nexys 2 board with an XILINX Spartan-3E Field Programmable Gate Array (FPGA). With ten users, the system was trained and tested. Each person recorded six samples of their voice with various intonations, resulting in 60 total samples. 50 percent of the samples were utilised for training, while the rest were used for testing. With this technique, a 98 percent identification rate was achieved.

Ravanelli and Bengio [10] performed speaker recognition from raw waveform with SincNET. This work offers SincNet, a completely new CNN design that encourages the primary convolutional Layer to find a large number of useful filters. SincNet works with parametrized sinc functions to create band-pass filters. In contrast to traditional CNNs, which learn all components of each filter, the projected approach just learns the low and high cutoff frequencies directly from knowledge. This provides a very compact and cost-effective way to create a custom filter bank that is individually optimised for the application. Our results on each recognition and speaker verification task reveal that the proposed approach converges faster and performs better on raw waveforms than a traditional CNN. With SincNet, we were able to get an 85 percent result on the TIMIT dataset.

Schmidt and Gish [11] performed Speaker identification via support vector classifiers. They introduced a novel way to speaker identification in this research. The approach, which is based on Vapnik's support vector work, is intriguing for various reasons. The support vector method is a discriminative strategy that models the borders between speakers' voices directly in some feature space rather than having to estimate speaker densities, which is a tough intermediary step. Most notably, support vector discriminant classifiers are unusual in that they separate training data while maintaining a low discriminating power, resulting in fewer test errors. As a result, viable classifiers may be built with far more parameters than training points. Furthermore, by determining boundaries on the predicted number of test errors, Vapnik's theory proposes which class of discriminating functions should be utilised given the amount of training data. When compared to other discriminant functions, support vector classifiers are faster to compute. Though the results are preliminary, using the Switchboard corpus, performance improvements above the BBN modified Gaussian Bayes decision system have been obtained, with an accuracy of 88 percent.

Zhang et al.[12] developed an End-to-End Neural Speaker Embedding System. Deep Speaker, a neural speaker embedding system that maps utterances to a hypersphere where cosine similarity is used to measure speaker similarity, is presented in this research. Deep Speaker's embeddings can be utilised for a variety of applications, such as speaker recognition, verification, and clustering. They use ResCNN and GRU architectures to extract acoustic features, then use a mean pool to generate utterance-level speaker embeddings and train with triplet loss based on cosine similarity. Deep Speaker appears to outperform a DNN-based i-vector baseline in three different datasets. On a text-independent dataset, Deep Speaker, for example, reduces the verification equal mistake rate by 50% and boosts identification accuracy by 60% (approximately). They also provide findings that imply that adopting a model trained in Mandarin can boost accuracy in recognising English speakers.

Parveen et al.[13] performed speaker recognition with recurrent neural networks. They describe the use of recurrent neural networks in an open set text-dependent speaker recognition problem in this study. They employ a feedforward net architecture based on Robinson et.al. Between the input and state nodes, as well as the output, we introduce a fully connected hidden layer. They demonstrate that this concealed Layer improves the efficiency of learning complex classification tasks. Backpropagation across time is used in the training. Each speaker has one output unit, with training targets that match to speaker identification. We get a real acceptance rate of 100% with a fake acceptance rate of 4% for 12 speakers (a mix of male and female). These percentages are 94 percent and 7% for 16 speakers, respectively.

2.2 Summary

This chapter compiled a list of related articles and research. With further breakthroughs in the field of deep learning, we are seeing a trend of deep learning in Speaker Identification, when earlier individuals relied on machine learning for speech recognition. Furthermore, these new deep learning approaches outperform current ones.

CHAPTER 3

METHODOLOGY

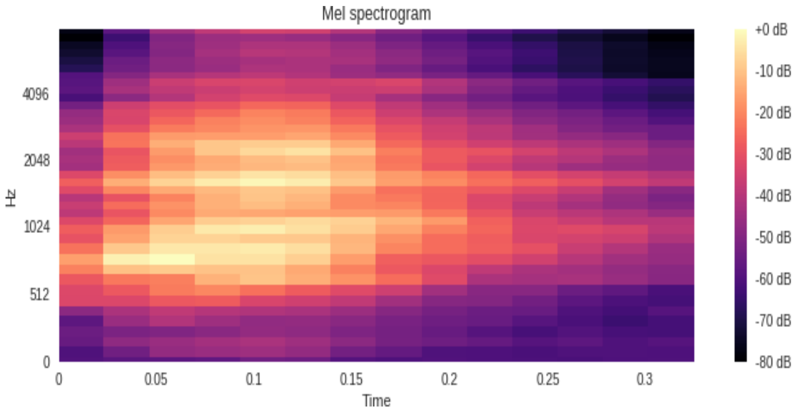
The proposed approach uses Hybrid of two models for the implementation of Speaker Identification from voice. The identification is done using Deep Learning models which are MLP, CNN, RNN, and LSTM at the end results are compared.

3.1 Dataset Description

In this research, we aim towards identification of a speaker by machine on the basis of his/her voice where no identity is claimed by user. The dataset consists of various voice recording with each voices record in different pitch, speed and with some noise. We are willing to train our machine to high levels and therefore get decent results with such a diverse dataset.

We analyze 500+ voice recordings from 13 classes and each class contains about 90 to 100 voice and the extension of voice is .wav, every category label is about with a speaker name. Feature extraction is finished by mfcc (Mel-frequency cepstral coefficients), melspectogram (mel-scaled spectrogram), chroma_stft (Short-Time Fourier Transform), chroma_cqt (Constant-Q transform), and chroma_cens (Chroma Energy Normalized). The neural network is trained by applying these options as input parameters. From every voice, extracting two hundred options by mfcc, melspectogram, chroma_stft, chroma_cqt, and chroma_cens which implies forty from every.

If we plot the the feature of one voice as image:



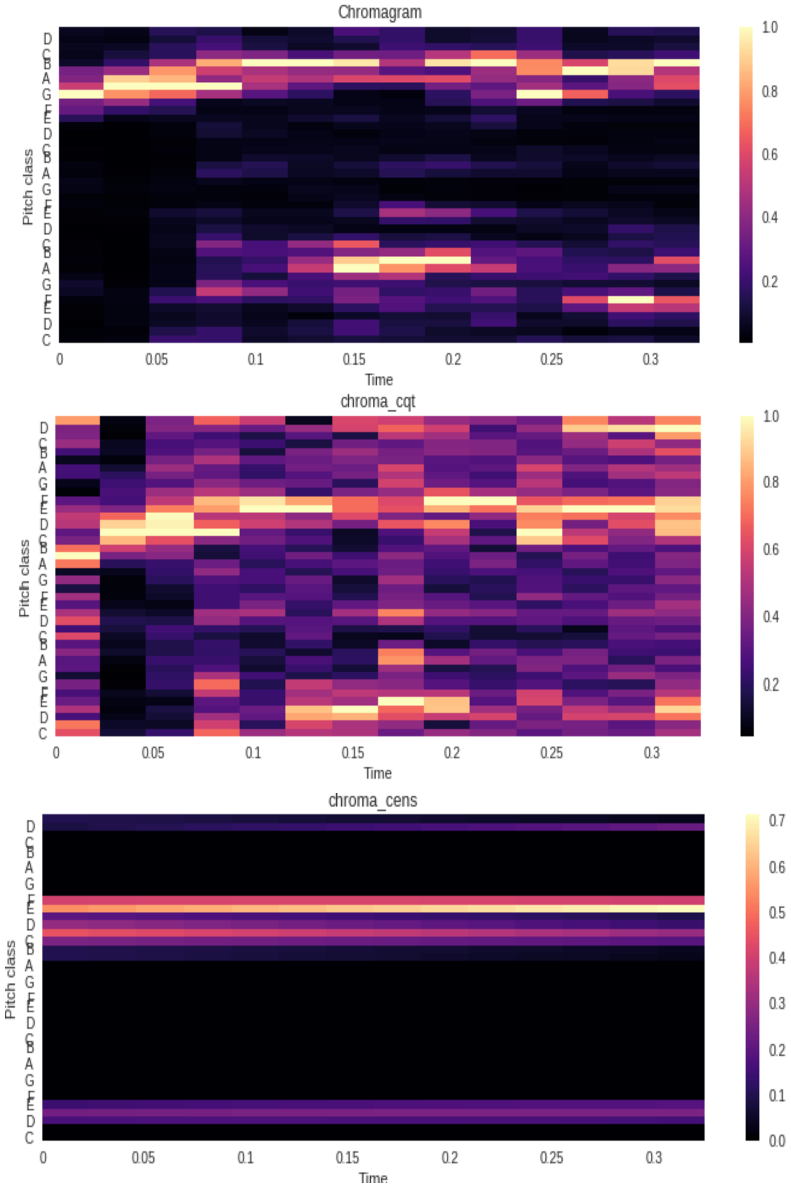


Figure 3.1 Sample of Features as in the form of image

3.2 Work Flow Diagram

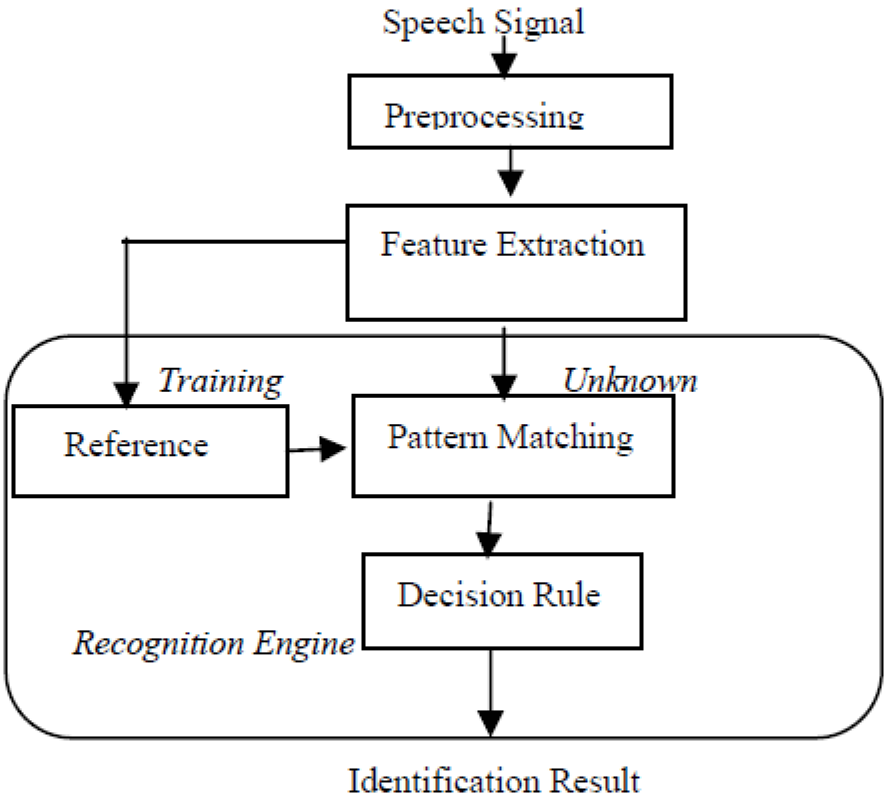


Figure 3.2 Work flow diagram

3.3 Implementation using Deep Learning

Multi Layer Perceptron, Convolutional Neural Network , Recurrent Neural Network, and Long Short Term Memory were employed in this implementation, each of which has an input and output layer, as well as a number of hidden layers.

There are four levels in Multi Layer Perceptron: one input layer, two hidden layers, and one output layer. We use the relu activation function in this and the softmax activation function in the output.

In Convolutional Neural Network the total number of Layer is five, two Layer is for convolution, two for dense Layer and one output layer which is fully connected. In this, we are using relu activation function and in output, we are using a softmax activation function.

In Recurrent Neural Network the total number of Layer is four, Two Layer is for RNN, one for dense Layer and one output layer which is fully connected[14]. In this, we are using tanh activation function and in output, we are using a softmax activation function.

In Long Short Term Memory the total number of Layer is four, Two Layer is for Long Short Term Memory, one for dense Layer and one output layer which is fully connected. In this, we are using tanh activation function and in output, we are using a softmax activation function. Every model is originally written with CUDA to run with GPU support. More details have been discuss in the next section of this chapter.

In Hybrid model Which is the mixture of Convolutional Neural Network and Long Short Term Memory , one side of CNN we have used 5 layer, Two are convolutional Layer, 2 are dense Layer and one Layer for flatten the output, On other side of LSTM we used 4 layers. Two layers of LSTM, One Dense Layer and one to flatten the output. Both the output one from CNN side and Other from LSTM were merged in the Hybrid Model which consist of 3 layers. Two dense Layer and one output Layer. In this we have used relu activation function and in output we used softmax activation function.

3.4 Proposed Architecture of Deep Learning models

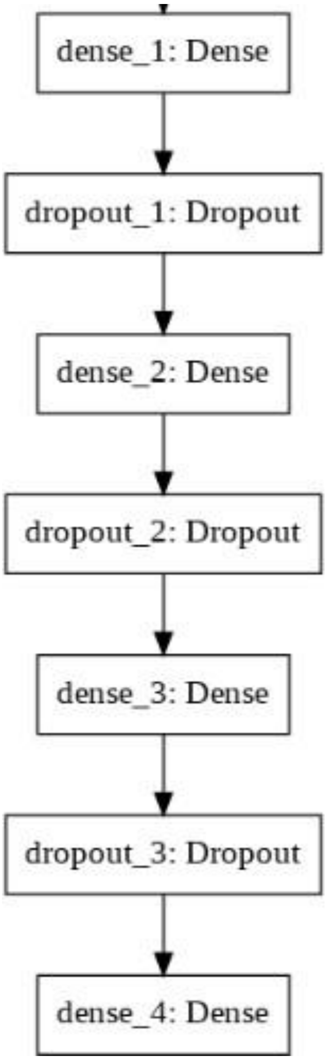


Figure 3.3 Architecture of MLP

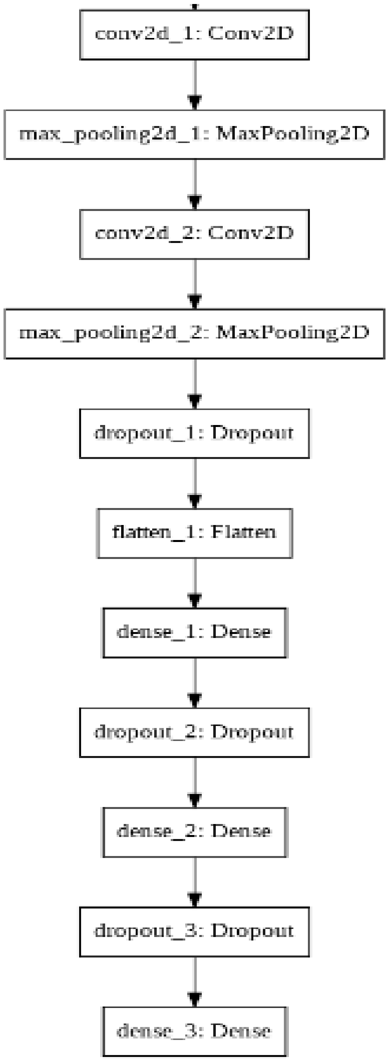


Figure 3.4 Architecture of CNN

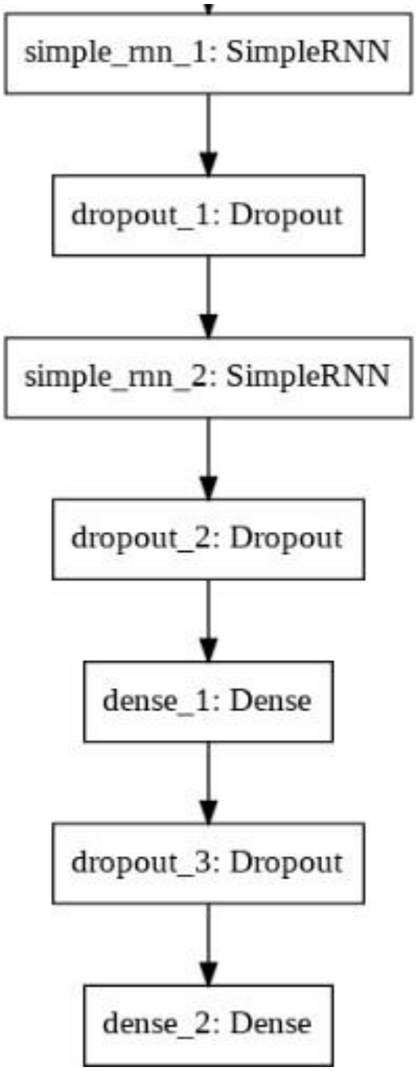


Figure 3.5 Architecture of RNN

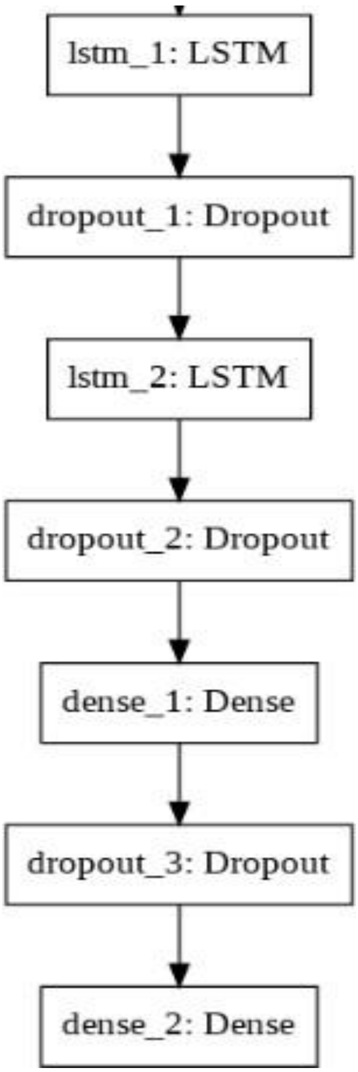


Figure 3.6 Architecture of LSTM

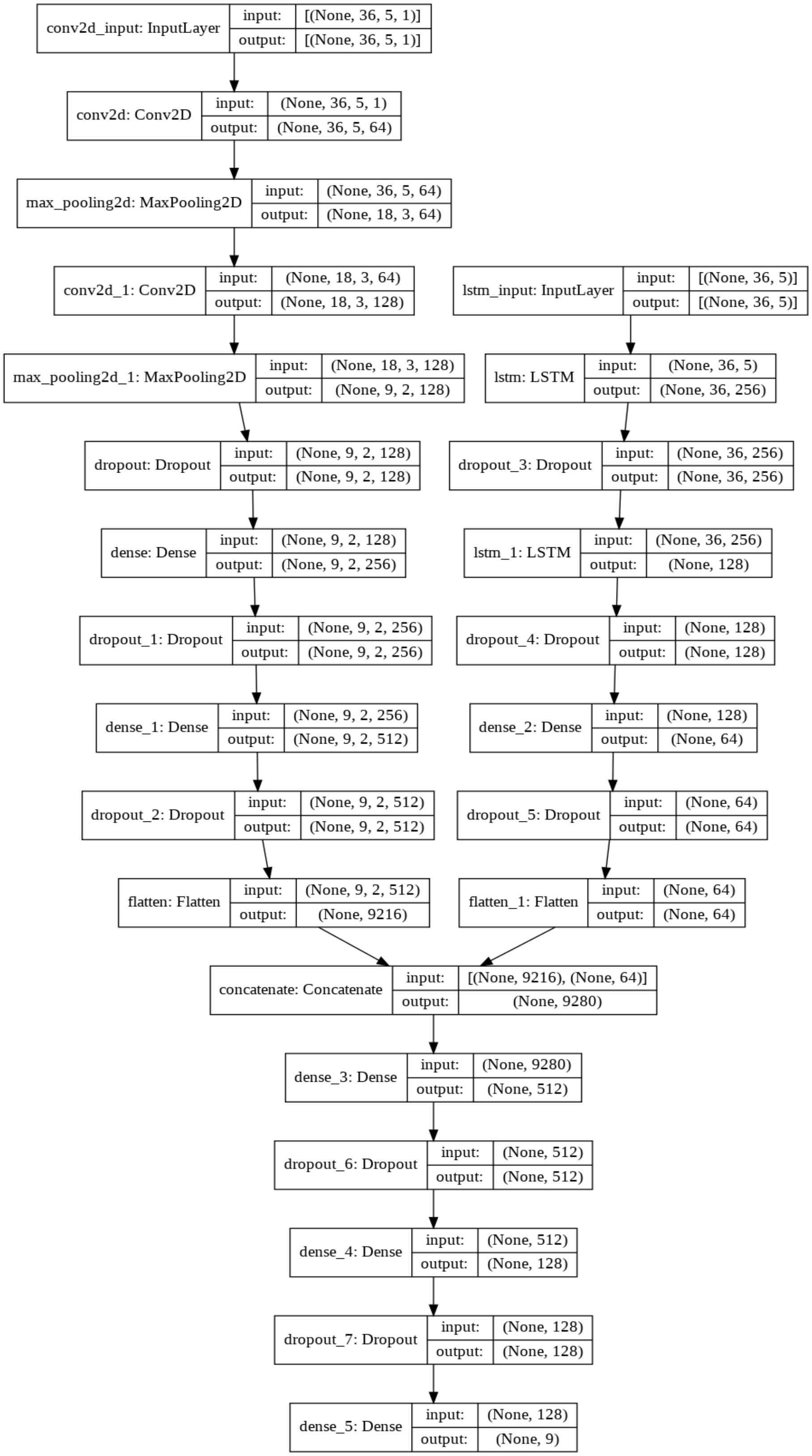


Figure 3.7 Architecture of Hybrid model

3.4.1 Convolutional Layers

Before passing the outcome to the next layer, convolutional layers conduct a convolution on the input. The convolution mimics the response of a single neuron to visual input. Each convolutional neuron is solely responsible for processing data for the receptive area to which it has been allocated. Although fully connected feedforward neural networks may be used to learn features and categorize data, this architecture cannot be utilized to classify pictures or sounds. Even with shallow (opposite of deep) architecture, a large number of neurons would be needed because to the very large input sizes associated with pictures, where each pixel is a significant variable. For a (small) picture of size 100 x 100, a fully linked layer, for example, has 10000 weights for each neuron in the second Layer. Convolution addresses this issue by reducing the amount of free parameters in the network, allowing for a deeper network with fewer parameters. For example, regardless of picture size, tiling 5x5 areas with the same shared weights needs just 25 learnable parameters. In multi-layer neural networks with many layers, backpropagation is employed to solve the issue of disappearing or exploding gradients[15].

3.4.2 Activation Function

We use an activation function to make our output non-linear. In the case of a Convolutional Neural Network, the output of the convolution will be transmitted through the activation function. It's conceivable that the ReLU activation function is involved..

ReLU function is given by:

$$\mathrm{F}(x) = \max(0, x)$$

3.4.3 Pooling Layer

After a convolution layer, it's common to add another pooling layer across CNN layers. The aim of pooling is to decrease the amount of parameters and calculations in the network by reducing dimensionality as much as feasible. This saves time in the gym while also avoiding overtraining.

The most frequent kind of pooling is max pooling, which takes the optimum output in each window. The dimensions of these windows must be set ahead of time. Another example is average pooling, in which the average value from each cluster of neurons in the preceding Layer is used.

3.4.4 Dropout Layer

Overfitting can be mitigated with the use of a dropout layer. Dropout at random disables a part of neurons throughout the training process, significantly decreasing dependence on the coaching set. A hyperparameter determines which percentage fractions of neurons you want to display and can be adjusted accordingly. Because not all neurons are active at the same time in this method, the network will be unable to store any information, and the inactive neurons are unable to learn something.

3.4.5 Fully connected Layer

In fully connected layers, every neuron in one layer is coupled to every neuron in the next layer. In theory, it works in the same way as a traditional multi-layer perceptron neural network.

3.4.6 Optimizers

During the training phase, we modify and alter the parameters (weights) of our model to minimise the loss function and produce the most accurate predictions possible.

The optimizer links the loss function and model parameters by modifying the model in response to the output of the loss function. Optimizers tamper with weights to mould and mould your model into the most exact form possible, in layman's words. The optimizer's loss function acts as a road map, showing if it is on the right or incorrect track.

3.4.7 Gradient Descent

Gradient descent is the most fundamental technique and the basis of how we train and optimize Intelligent Systems.

Gradient Descent finds the Minima, lowers variance, changes the Model's parameters, and ultimately brings the model to convergence.

3.4.8 Stochastic Gradient Descent

Instead of computing the gradients for all of given training samples on each gradient descent pass, it's occasionally more efficient to use only a subset of them each time. Stochastic gradient descent is a method that uses either batches of examples or random samples for each pass..

3.4.9 Momentum

Momentum is a popular strategy that is used in tandem with SGD. Rather of relying simply on the gradient of the current step to guide the search, momentum also takes into account the gradient of prior steps.

3.4.10 Adam

Adaptive moment estimate (Adam) is a technique for estimating current gradients based on previous gradients. Adam also uses momentum by integrating elements from previous gradients into the present one. This optimizer has gained a lot of traction and is now widely used in neural network training.

Momentum and RMSProp heuristics are combined in Adam or Adaptive Moment Optimization methods.

CHAPTER4

IMPLEMENTATION AND RESULTS

4.1 Implementation Details

4.1.1 System Requirement

The configuration of the system used for the experiment is as follows:

- Processor: Intel(R) Core(TM) i3-6006U CPU @ 2.00 GHz
- RAM: 8.00 GB
- GPU: 2.00 GB
- System Type: 64-bit
- Operating System: Windows 10

4.1.2 Software Requirement

- IDE : Anaconda (Jupyter, Spyder), Google Colaboratory
- Programming Language: Python
- Packages : Keras, Matplotlib, Pandas, NumPy, Scikit-learn, LibROSA

4.1.3 Experiment

The experiment was conducted on a system configured with Windows 10 operating system with Google Collaboratory. The source code is written in python language. About 1,330 voice recordings from 14 classes and each class contains about 90 to 100 voice are processed for the project.

For Feature extraction process we have used the Librosa Library, which is used for the feature extraction of the features like MFCC which is Mel-Frequenct cepstral coefficients, melspectogram which is mel-scaled spectrogram, chroma_stft it is a short time fourier transform, chroma_cqt it is constant_Q transform, and chroma_cens it is chroma energy Normalized. The proposed neural networks were trained by taking the input parameters applied of the above features. We have extracted 200 features from each sample voice by mfcc, melspectogram, chroma_stft, chroma_cens and chroma_cqt.

After feature extraction from the voice, we create a .csv file and save these feature. Datais split into test and train. Train data is having between 70 to 75 data point per class and Test data is having between 20 to 25 data point per class.

Implementation is done by Deep Learning model CNN, RNN and LSTM and the comparison is done on basis of accuracy achieved in both the models. Deep learning takes about 12 hours in which using 10 epoch and 125 steps per epoch to complete the training process on CPU while on GPU it takes about 1.5 hours and using 25 epochs and 125 steps per epoch.

4.2 Results

4.2.1 Multi-layer Perceptron

The proposed methodology uses a deep learning approach which is MLP (Multilayer Perceptron). Input data is passed from the input layer in which 256 neurons. The total number of Layer is four, one input layer two hidden layers and one output layer. In this, we are using relu activation function and in output, we are using a softmax activation function.

Adam optimizer with learning rate 0.001, and dropout 0.15

- **Model Accuracy**

  Accuracy: Test = 98.35%, Train = 86.67%,

- **Model Loss**

  Loss: Train = 0.8480, Test = 0.0321

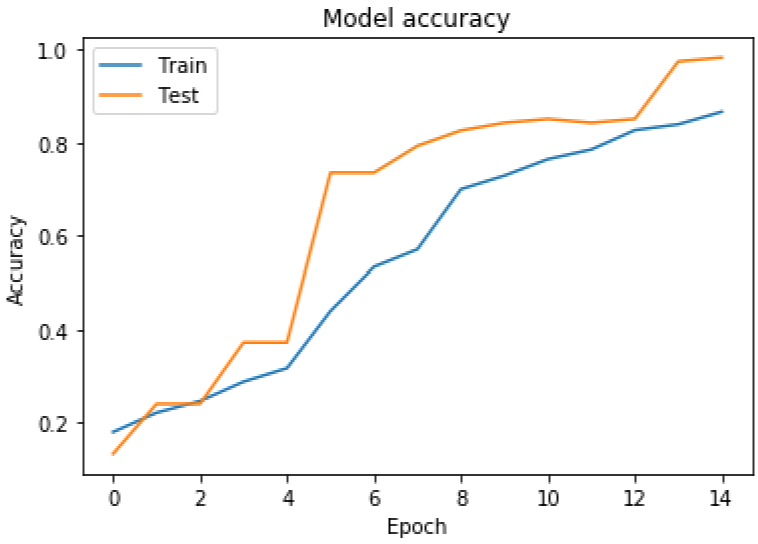


Figure 4.1 Model Accuracy of MLP

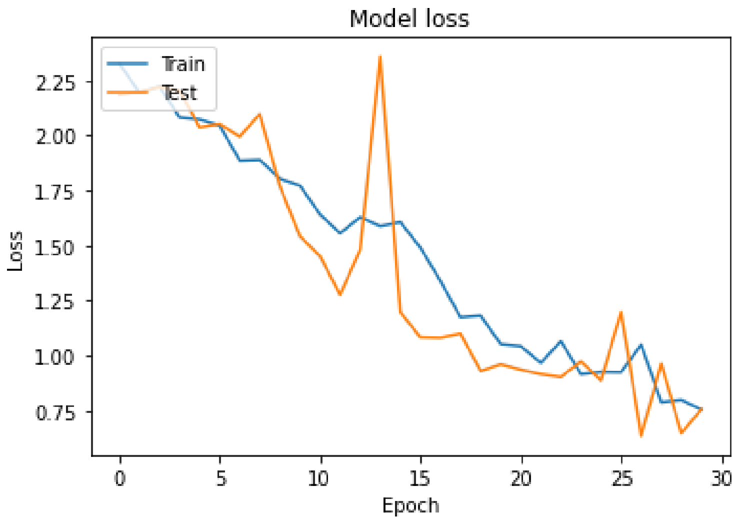


Figure 4.2 Model Loss of MLP

4.2.2 Convolution Neural Network

The proposed methodology uses a deep learning approach which is CNN (Convolution Neural Network). Input data is passed to input deep learning module (CNN). Convolution operation in proposed work uses 2 layers. The total number of kernels are 64 with thesize of 5*5 is used for the first convolution layer and last Layer having numbers of the kernel are 128 of the same size which is 5*5. The last Layer is fully connected followed by softmax layer in CNNs model.

Adam optimizer with learning rate 0.001, and dropout 0.15

- **Model Accuracy**

  Accuracy: Test = 99.17%, Train = 99.38%

- **Model Loss**

  Loss: Train = 0.0261, Test = 0.0248

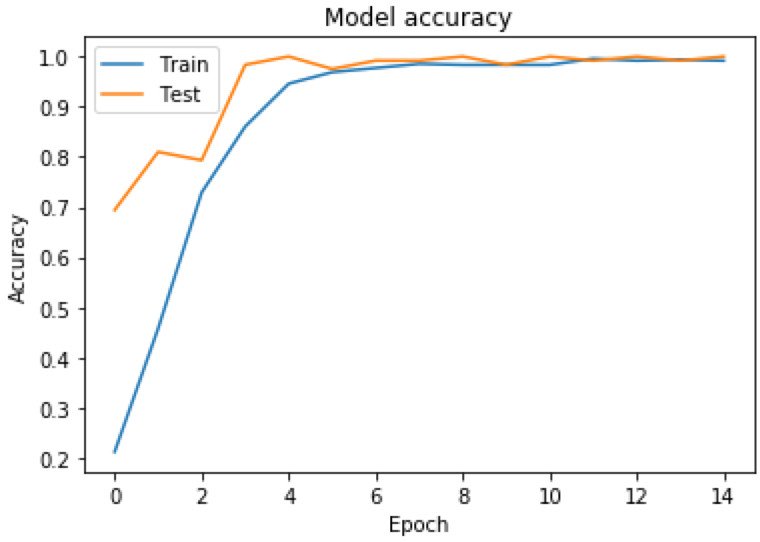


Figure 4.3 Model Accuracy of CNNs

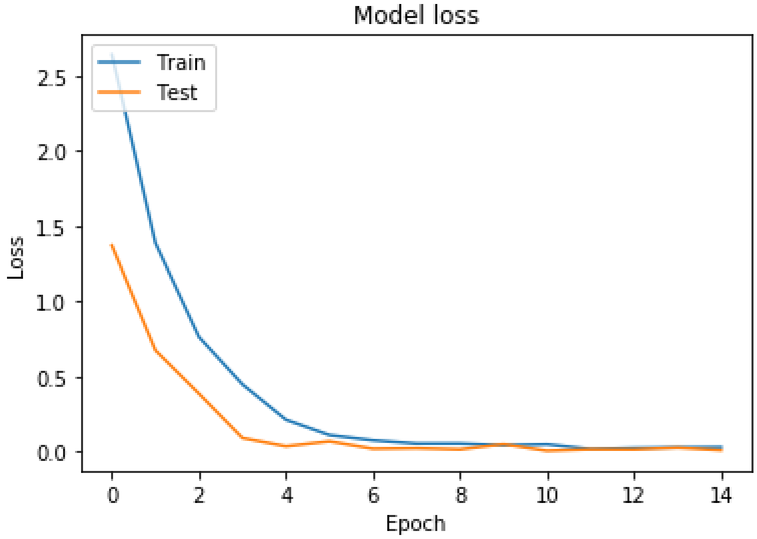


Figure 4.4 Model Loss of CNNs

4.2.3 RNN

The proposed methodology uses a deep learning approach which is RNN. Input data is from RNN layer with data size (36,5) The total number of Layer is four, two for RNN, one for dense Layer and last Layer is for output layer. In this, we are using relu activation function and in output, we are using a softmax activation function.

Adam optimizer with learning rate 0.001, and dropout 0.15

- **Model Accuracy**

  Accuracy: Test = 98.35%, Train = 96.04%

- **Model Loss**

  Loss: Train = 0.1229, Test = 0.0358

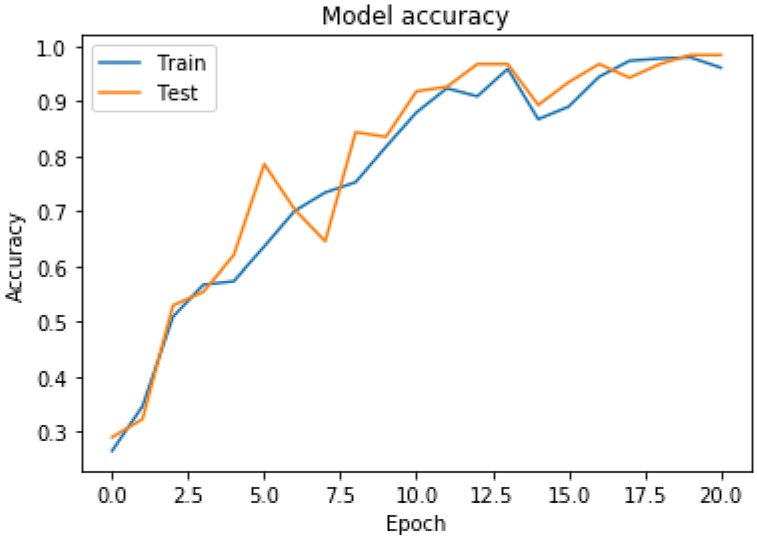


Figure 4.5 Model Accuracy of RNN

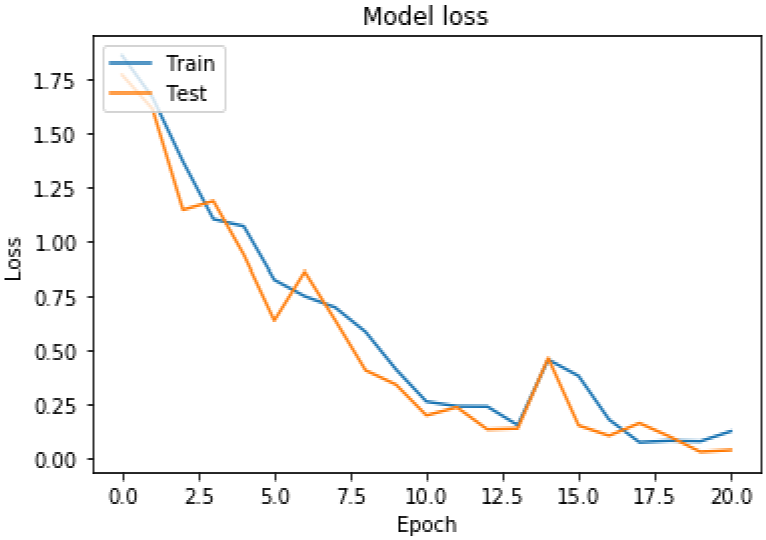


Figure 4.6 Model Loss of RNN

4.2.4 LSTM

The proposed methodology uses a deep learning approach which is LSTM. Input data is passed from LSTM layer with data size (36,5) The total number of Layer is four, two for lstm, one for dense Layer and last Layer is for output layer. In this, we are using tanh activation function and in output, we are using a softmax activation function.

Adam optimizer with learning rate 0.001, and dropout 0.15

- **Model Accuracy**

  Accuracy: Test = 99.67%, Train = 99.58%

- **Model Loss**

  Loss: Train = 0.0312, Test = 0.0091

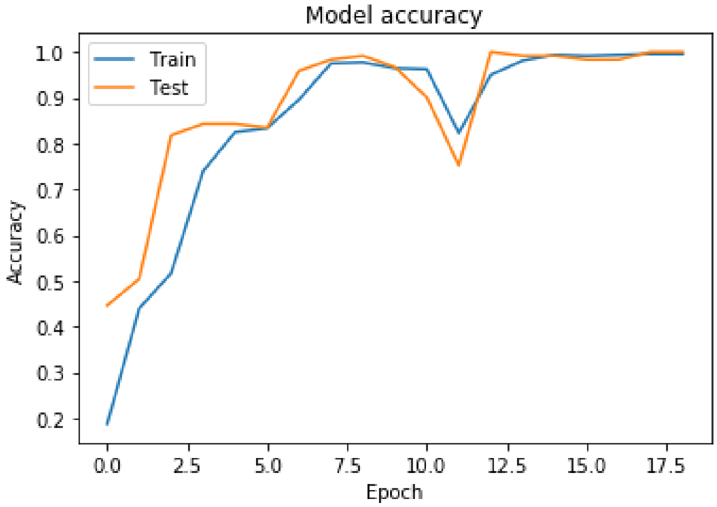


Figure 4.7 Model Accuracy of LSTM

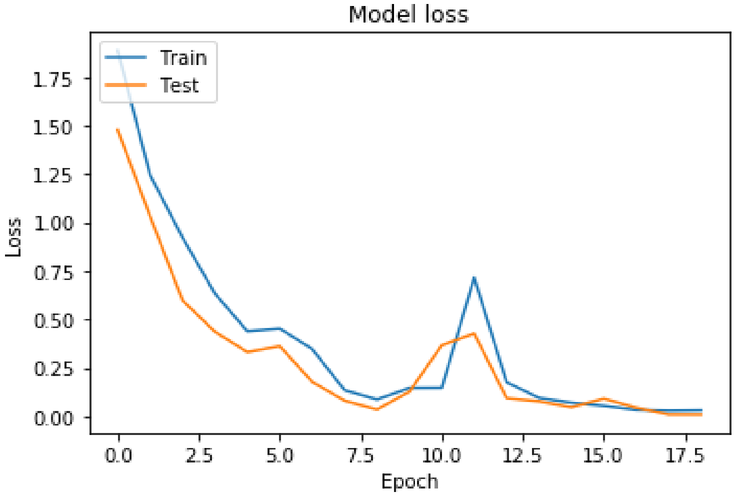


Figure 4.8 Model Loss of LSTM

4.2.5 GRU

The proposed methodology uses a deep learning approach which is GRU. Input data is passed from GRU layer with data size (36,5) The total number of Layer is four, two for gru, one for dense Layer and last Layer is for output layer. In this, we are using relu activation function and in output, we are using a softmax activation function.

Adam optimizer with learning rate 0.001, and dropout 0.15

- **Model Accuracy**

  Accuracy: Test = 97.52%, Train = 99.58%

- **Model Loss**

  Loss: Train = 0.0105, Test = 0.1984

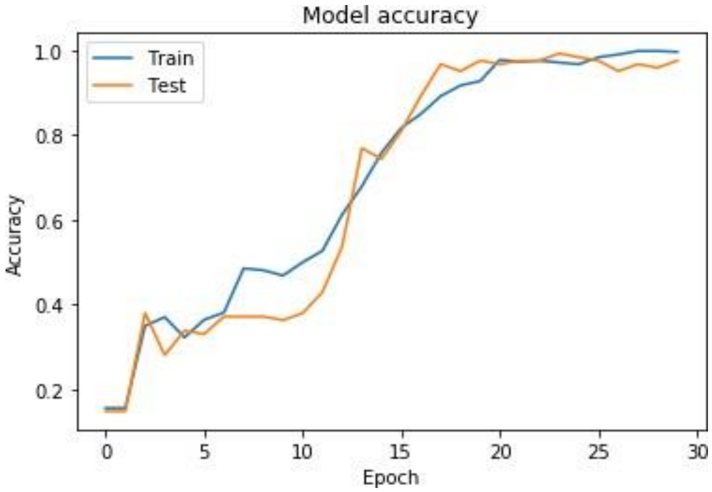


Figure 4.9 Model Accuracy of GRU

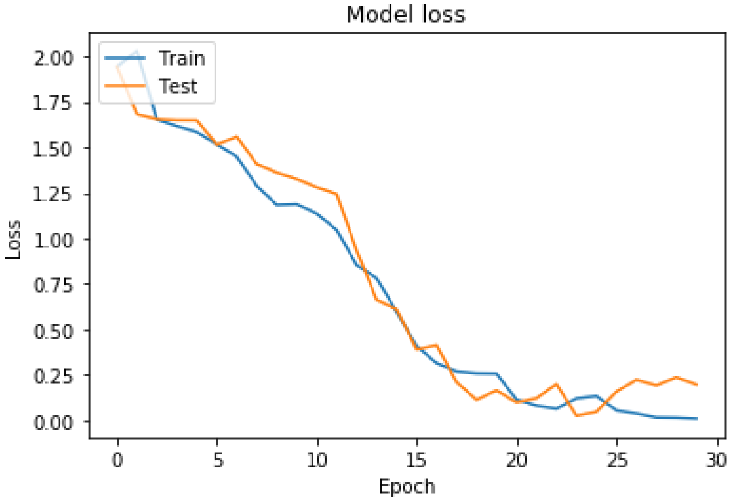


Figure 4.10 Model Loss of GRU

4.2.6 LSTM-CNN (Hybrid Model):

The proposed methodology uses a Hybrid deep learning approach in which CNN and LSTM is used to make a Hybrid Network. Input data is passed to input deep learning module (CNN) and the another module (LSTM). Convolution operation uses 2 layers. The total number of kernels are 64 with the size of 5*5 is used for the first convolution layer and last Layer having numbers of the kernel are 128 of the same size which is 5*5. And from another side Input data is passed from LSTM layer with data size (36,5) The total number of Layer is four, two for LSTM, one for dense Layer and last Layer is for output layer. After that Both models were concatenated. The last Layer is fully connected followed by softmax layer in CNNs model.

Adam optimizer with learning rate 0.001, and dropout 0.15

- **Model Accuracy**

  Accuracy: Test = 98.67%, Train = 98.58%

- **Model Loss**

  Loss: Train = 0.8480, Test = 0.0321

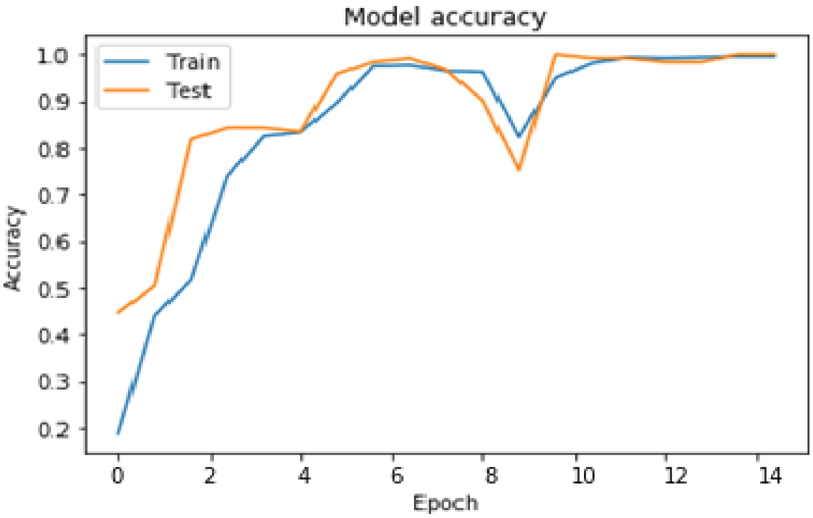


Figure 4.11 Model Accuracy of HYBRID

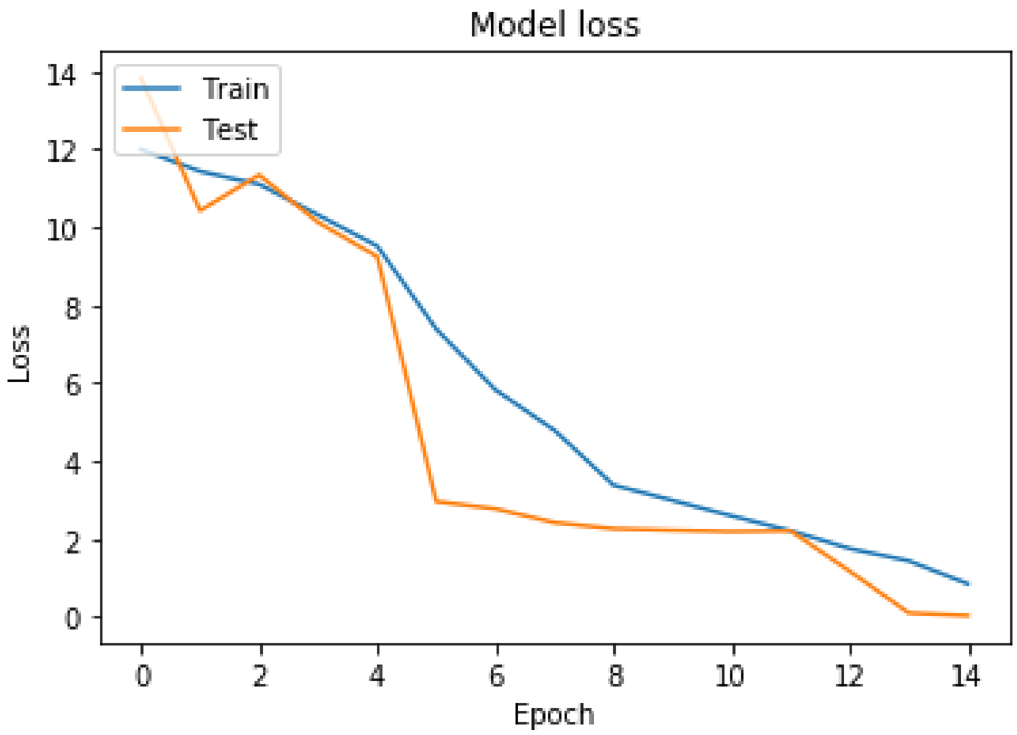


Figure 4.12 Model Loss of HYBRID

4.3 Graphs of Accuracy and Loss of different models

4.3.1 Combined accuracy value graph of different Deep Learning techniques

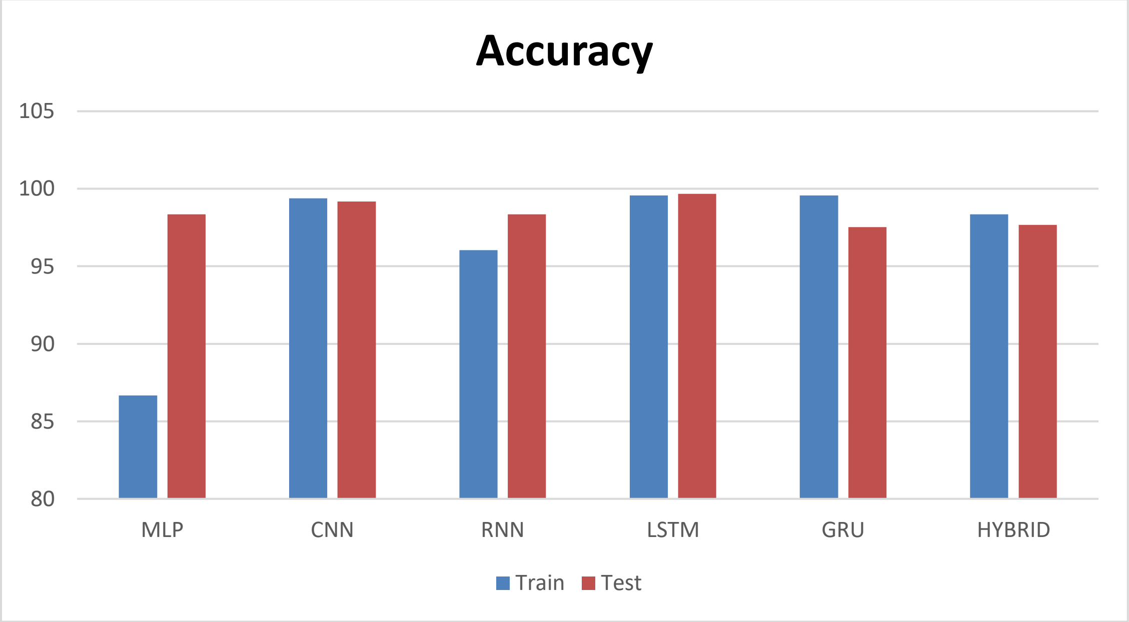


Figure 4.13 Accuracy graphs of different Deep Learning techniques

4.3.2 Combined loss graph of different Deep Learning techniques

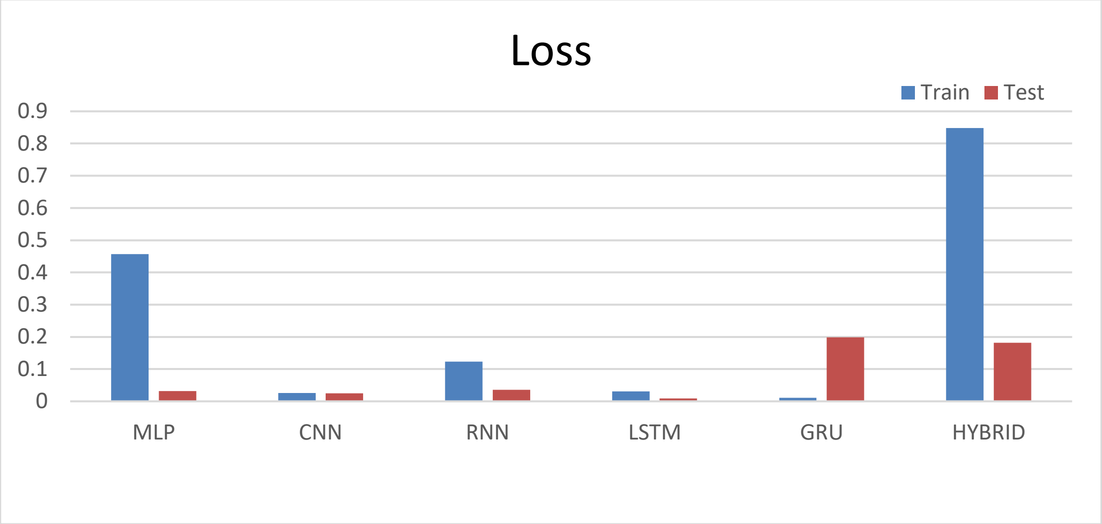


Figure 4.14 Loss graphs of different Deep Learning techniques

4.4 Summary

This chapter provides the implementation details of the proposed solution. In which different models are used for implementation and LSTM model gives better resultsamong all of them.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this project, we have presented an implementation of a speaker identification system on different Deep Learning models. This system used MLP, CNN, RNN, LSTM, and GRU in order to achieve recognition. The neural network was trained by using backpropagation as the learning algorithm.

The system is able to distinguish a number of 12 speakers in a satisfactory way. These were the speakers whose data samples we have collected for the system training. The speaker identification system was first trained and tested with the different samples that used for training it. This software works fine for the identification of speakers from a clusters of samples.

The achieved test accuracy from MLP, CNN, RNN, LSTM, GRU and HYBRID model was 98.35%, 99.17%, 98.35%, 99.67%, 97.52% and 98.67 respectively.

The future work include the feature like tagging the speaker from mix voice.

REFERENCES

[1] G. Assuncao, P. Menezes, and F. Perdigao, “Importance of speaker specific speech features for emotion recognition,” *Proc. 2019 5th Exp. Int. Conf. exp.at 2019*, pp. 266–267, 2019, doi: 10.1109/EXPAT.2019.8876534.

[2] R. Chakroun, L. Beltaïfa Zouari, M. Frikha, and A. Ben Hamida, “Improving text-independent speaker recognition with GMM,” *2nd Int. Conf. Adv. Technol. Signal Image Process. ATSIP 2016*, no. 1, pp. 693–696, 2016, doi: 10.1109/ATSIP.2016.7523169.

[3] F. Ur Rehman, C. Kumar, S. Kumar, A. Mehmood, and U. Zafar, “VQ based comparative analysis of MFCC and BFCC speaker recognition system,” *2017 Int. Conf. Inf. Commun. Technol. ICICT 2017*, vol. 2017-Decem, pp. 28–32, 2018, doi: 10.1109/ICICT.2017.8320160.

[4] Y. Zhan and X. Yuan, “Audio post-processing detection and identification based on audio features,” *Int. Conf. Wavelet Anal. Pattern Recognit.*, vol. 1, pp. 154–158, 2017, doi: 10.1109/ICWAPR.2017.8076681.

[5] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[6] R. Shashidhar, S. Patilkulkarni, and S. B. Puneeth, “Audio Visual Speech Recognition using Feed Forward Neural Network Architecture,” *2020 IEEE Int. Conf. Innov. Technol. INOCON 2020*, pp. 1–5, 2020, doi: 10.1109/INOCON50539.2020.9298429.

[7] S. Yang, X. Yu, and Y. Zhou, “LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example,” *Proc. - 2020 Int. Work. Electron. Commun. Artif. Intell. IWECAI 2020*, pp. 98–101, 2020, doi: 10.1109/IWECAI50956.2020.00027.

[8] Y. Lukic, C. Vogt, O. Durr, and T. Stadelmann, “Speaker identification and clustering using convolutional neural networks,” *IEEE Int. Work. Mach. Learn. Signal Process. MLSP*, vol. 2016-Novem, 2016, doi: 10.1109/MLSP.2016.7738816.

[9] F. Trujillo-Romero and S. O. Caballero-Morales, “Speaker identification using neural Networks on an FPGA,” *Proc. - 2012 9th Electron. Robot. Automot. Mech. Conf. CERMA 2012*, pp. 197–202, 2012, doi: 10.1109/CERMA.2012.39.

[10] M. Ravanelli and Y. Bengio, “Speaker Recognition from Raw Waveform with SincNet,” *2018 IEEE Spok. Lang. Technol. Work. SLT 2018 - Proc.*, pp. 1021–1028, 2019, doi: 10.1109/SLT.2018.8639585.

[11] M. Schmidt and H. Gish, “Speaker identification via support vector classifiers,” *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 1, pp. 105–108, 1996, doi: 10.1109/icassp.1996.540301.

[12] C. Li *et al.*, “Deep Speaker: an End-to-End Neural Speaker Embedding System,” 2017, [Online]. Available: http://arxiv.org/abs/1705.02304.

[13] S. Parveen, A. Qadeer, and P. Green, “Speaker recognition with recurrent neural networks,” *6th Int. Conf. Spok. Lang. Process. ICSLP 2000*, no. January 2000, 2000.

[14] D. Priyasad, T. Fernando, S. Denman, S. Sridharan, and C. Fookes, “Attention Driven Fusion for

Multi-Modal Emotion Recognition," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2020-May, pp. 3227–3231, 2020, doi: 10.1109/ICASSP40776.2020.9054441.

[15] T. Kikuchi and Y. Ozasa, "Watch, Listen Once, and Sync: Audio-Visual Synchronization with Multi-Modal Regression Cnn," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2018-April, pp. 3036–3040, 2018, doi: 10.1109/ICASSP.2018.8461853.

[16] R M Makwana, Deep Face Recognition Using Deep Convolutional Neural Network,

AIeHive.com, http://www.ais.uni-bonn.de/deep learning/images/Convolutional NN.jpg