TEXT-INDEPENDENT SPEAKER VERIFICATION USING DEEP LEARNING TECHNIQUES

Master of Science (Computer Science)

Session (2017-2019)

Submitted By:

HAFSA HABIB 1725210008

Supervisor:
Dr. Huma Tauseef
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE
LAHORE COLLEGE FOR WOMEN UNIVERSITY, LAHORE

Text-Independent Speaker Verification using Deep Learning Techniques

Thesis

Submitted in Partial Fulfillment

of the Requirements for the
Degree of

Master of Science (Computer Science)

at the

Lahore College for Women University, Lahore

by

Hafsa Habib 1725210008

Dr. Huma Tauseef Supervisor Department of Computer Science	*Dr. Muhammad Abuzar Fahiem* Head of Department Department of Computer Science

CERTIFICATE

BY THE THESIS SUPERVISOR

I certify that the contents and form of thesis submitted by Hafsa Habib, Roll No. 1725210008 has been found satisfactory and according to the prescribed format. I recommend that it be processed for evaluation by the External Examiner for the award of degree in Master of Science (Computer Science).

Supervisor

Signature: _____

Name: Dr Huma Tauseef

Designation: Assistant Professor

DEDICATION

I dedicate this research work to my parents, supervisor and teachers and whole family, who stood for me, encouraged me to work better, gave me strength to pursue the goal. They encouraged me to give it my full focus and effort. I could not have done this without them.

ACKNOWLEDGEMENT

Firstly, I would like to thank ALLAH Almighty for blessing me with strength to achieve my research objectives. Then I like to thank our all respected supervisors for their support. Dr. Huma has been great supervisor and mentor. Her sage advice, insightful criticisms, and patient encouragement aided the writing of this document in innumerable ways. Her steadfast support of this Thesis work was greatly needed and deeply appreciated. We like to thank our classmates too who gave us encouragement.

ABSTRACT

In this era of digital advancement, researches are more focused on developing new techniques for better human-computer interaction. Speaker recognition is a course of biometric recognition that works by extracting patterns from selected human trait like voice or finger prints into a digital signature. Speaker or voice recognition is the most feasible biometric recognition technique for human and machine interaction because high attainability and availability. Lack of need for special hardware makes it very low in implementation cost. The text-independent scenario, speaker utterances are not limited to predefined phrases. It offers more convenience to the user and puts load on the system as it become more challenging to extract speaker specific information in speech signal. The phases of speaker verification protocol are training, enrollment of speakers and evaluation of unknown voice. In training, a universal background model is created to learn speaker representations. In enrollment phase, models for new speakers are generated using trained model. Lastly in evaluation phase, the speaker utterances are matched with the previously saved model of claimed speaker for verification. In this research, we solve text independent speaker verification task with binary operations based Siamese convolutional network. We have implemented a unique customized scoring scheme which utilizes Siamese' capability of shared layers to merge and apply convolutional learning. Experiments made on cross language audios of multi-lingual speakers emphasize the capability of our architecture to handle fraud and forgery. Moreover, our designed Siamese network, SpeakerNet, provided better results than most of the existing speaker verification approaches by decreasing the equal error rate to 2.6 %.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Importance of Speech

In this era of digital advancement, researches are more focused on developing new techniques for better human-computer interaction. Initially the urge of interaction with computer motivated the invention of input-output devices like keyboards, scanners, monitors, joysticks, touch screens and trackballs. However verbal communication with computers is still not possible with above mentioned devices. Which is the natural mean of communication for both humans and animals in the universe alike. The lack of communication with machines via speech leads the researchers towards inventing robust speech processing systems for convenient human computer interaction using speech. Speech is used in computation for various applications for example, speech recognition, language recognition, speech synthesis and speaker recognition.

1.2 Biometric Recognition

Speaker recognition is a course of biometric recognition. Biometric Pattern recognition techniques work by extracting patterns from selected human trait like voice or finger prints into a digital signature. This signature is later used to recognize or verify a person. The use of biometrics is increasing widely to secure access to high profile places, data, services and other procedures. They offer speed, convenience and efficiency in routine tasks and reduce chances of frauds and impersonation as biometric traits are difficult to imposter. Traditional methods of access control include passwords, Personal Identification Numbers and knowledge-based authentication questions. Pass phrases and PINs are easily forgettable and it is almost impossible for one to remember all of the passwords and PINs used across websites and databases. Which leads to

keeping them all stored in one place making it vulnerable to hackers. Even knowledge-based authentication questions can be answered just by knowing the person. Apart from the computer world, many countries are now issuing biometric identities instead of traditional documents because of more cases of forgery and fraud. Lately finger prints, palm detection and iris recognition techniques are widely used in high security areas.

1.3 Voice as Biometric

1.3.1 Comparison with other biometrics

There are some considerations before choosing any biometric authentication technique such as their robustness, exclusiveness, attainability and acceptability. Table 1.1 extends the comparison of above four considerations in authentication via multiple biometric traits.

Table 1-1: Comparison of biometric traits

	Iris	face	Finger prints	speech
robust	High	Medium	High	Medium
exclusive	High	Low	Medium	Medium
attainable	Low	High	Medium	High
acceptable	Medium	High	Medium	High

From table 1-1, it is evident that although its exclusiveness and robust performance is medium, speech is still highly attainable and acceptable biometric trait. Three goals of security measures include

1. maintain integrity of data,
2. data is only available to authenticated users, and
3. data is stored confidentially.

Biometric recognition or authentication systems can achieve second goal of security i.e., fulfillment of attainability and availability, by using speech as signature. Hence speaker or voice recognition is the most feasible biometric recognition technique for human and machine interaction because of the reasons stated below:

1. With the emergence of Voice over Internet Protocol (VoIP) technology and mobile phones, voice is the most reachable and easy to collect biometric trait.
2. Speech is primary source of human to human communication thus speech may work as primary source of interaction between human and computer.
3. It provides a relatively cheap and safer way of authentication as compared to passwords and pins.
4. A speaker's voice is very hard to imposter for biometric recognition purposes, considering a range of qualities affect the voice for example dialect, pitch and spectral, magnitude. The creation of voice from the vibration of vocal cords and the marks produced from other physical components are very distinct and unique for a person like fingerprints.

1.3.2 Advantages of Voice Biometrics

Impersonation attempts or using speech recordings to achieve deceitful authentication remain unsuccessful because of the unique characteristics of the biometric trait used for comparison. whereas voice forgery may sound like a definite equal to the humans, elaborate mathematical analysis of the digital template of voice tends to reveal immense variations. Likewise, audio recording used to authenticate also shows variation. To more thwart the employment of pre-recorded voiceprints, authentication employs a model of biometric authentication comparison called text freelance directed

speech. During this model, verification is performed against a phrase that's haphazardly generated, rather than employing a phrase predefined, like a fixed account password. A number of benefits are associated with voice or speaker authentication. Lack of need for special hardware makes it very low in implementation cost. A simple cellular or telephone call is all that is needed to perform authentication. On the contrary other biometric recognition methods like iris scan and fingerprints require special devices to complete the process. Speaker authentication is easily usable and its user acceptance rate is high. It is quite natural to talk. But putting up an eye to a scanner or sensor is not natural and causes inconvenience. It is also quite natural to identify humans on the basis of their voice. Our brain does it all the time when we answer phone calls. Most importantly, voice biometrics allow users to get access to certain places by remote authentication in low cost. It is much easier to call and authenticate with bank to perform a transaction than verifying fingerprints after going to the bank in person or use sophisticated device to scan fingerprints. Voice authentication systems are very fast to enroll a person with just few seconds of audio. From that audio, a digital signature of voice is generated. Authentication is performed by comparing the earlier digital signature with the new sample. Another benefit of using voice authentication is that the voice based digital signature has very small size. It can be around 1 kilobyte only. With all the benefits stated above, voice biometrics is still not the highest secure source to recognize a person. For this reason, it is more appropriate to use it with some other mean of authentication where security is very high. Speaker identification is better choice in forensic applications. Speaker verification is also widely introduced in deployable systems these days and appears as an emerging area.

1.4 Objective

Speaker verification needs large datasets and high computational resources to perfectly train the models. In spite of a lot of research progress in speaker recognition, the English and Mandarin based speakers dominate the datasets available for the task. Our objective is to improve speaker verification results for other languages i.e., Urdu as well while keeping the resource utilization low.

1.5 Thesis Structure

This thesis presents the proposed method to improve speaker verification by providing review of previous methods in terms of performances. Structure of the remaining thesis is as follows.

Chapter 2 covers review of existing features and methods adopted to perform speaker verification while also describing the overview of speech and signal processing. Important terms and definitions are also covered in chapter 2.

Chapter 3 comprises the datasets used, proposed preprocessing steps and architectural details of our proposed approach.

Chapter 4 starts with baseline systems and experimental protocol. Evaluation metrics used to analyze results are also covered. Our results are compared with baseline systems. Finally, other state of the art methods are also discussed.

Chapter 5 presents the conclusion of our research based on the analysis of the results

Chapter 2

2 Literature Survey

This chapter provides review of the existing speaker verification techniques and methodologies. Moreover, notations and nomenclature used the thesis is introduced here. This chapter begins with various speech and signal processing techniques followed by feature extraction methods. Lastly a brief review of all existing methodologies is presented.

2.1 Speech and Signal Processing

Speech is interpreted in the form of signal. This signal can be processed and analyzed with numerous techniques. Speech processing systems can utilize these techniques individually or in combination depending upon the need or problem. Four popular speech processing problem areas are Speech recognition, Speech synthesis, Language Recognition and Speaker Recognition as shown in figure 2-1.

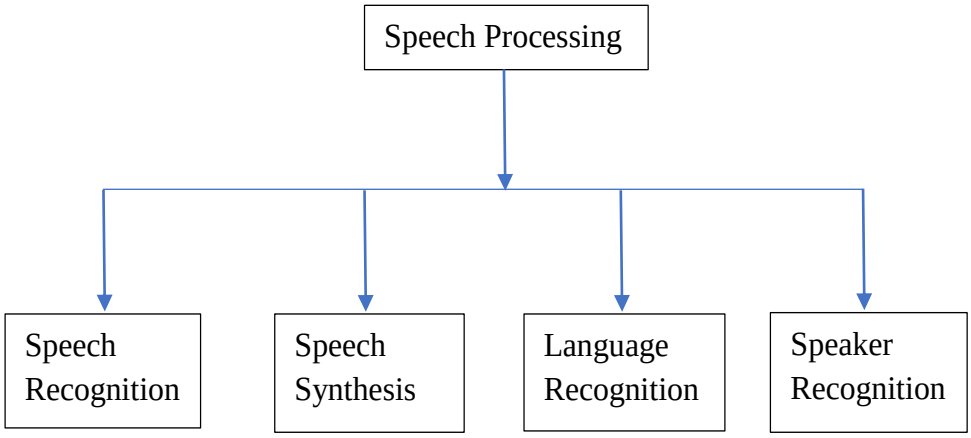

Figure 2-1: Problem areas in speech processing

2.1.1 Speech Recognition

Speech recognition is an area of signal processing which allows spoken words and sentences as input to the computer. This input may use to trigger some action for example performing specific task, or converting the words spoken into written form. This technology has offered an alternative way of input like typing or clicking. Speech recognition works by identifying spoken words by the computer. Then storing them in computer understandable format to be used to recognize the words in future. Speech recognition is used widely now a days in our daily technology usage. For example, mobile devices can now unlock and initiate any task on spoken instructions. Apart from its success in many areas, speech recognition still faces some limitations. It cannot translate all the spoken words with hundred percent accuracy as accents and pronunciation vary among same language speakers. Speech recognition can further be categorized into speaker-dependent and speaker-independent speech recognition. Speaker-dependent environment is relatively easy to deploy as the system will recognize the utterance of same person hence speaker related features do not affect the signal processing. While speaker-independent speech recognition is relatively complex task as the system has to take into account speaker variations as well as speech variations.

Speech Recognition protocol is divided into two phases, 1) training phase 2) recognition phase. In the training phase, features related to speech are extracted from speech signals and stored as a template of dictionary for later use. The second phase works in two steps, Firstly, features from the new speech signal are extraction. Secondly the features are match with stored template to classify the spoken words. The decision is made on distance-based measures. Training phase requires the speech signal to be transformed in any time-frequency domain. Popular time-frequency domain transformations are namely Short Time Fourier Transform (STFT), DCT

(Discrete Cosine Transform), Fourier Transform, LPC (Linear Predictive Coding) and Wavelet Transform (WT).

2.1.2 Speech Synthesis

Speech Synthesis works by the computer reading the written words out loud to the users also know as Text to Speech (TTS) systems. TTS systems follow a complex process which involves cleaning and separation of input into words and phrases, distributing the text according to phenomes with the help of built in dictionaries and finally a speech processor to match the phenomes with the sounds to produce speech. Performance of such systems is evaluated by calculating their similarity to the human pronounced speech and from the clarity of words. Speech synthesizers have been incorporated in many systems for many purposes. They can easily be heard at train stations and airports. TTS system helps visually impaired people in their use of mobile and other digital devices.

2.1.3 Language Recognition

Speech processing can also help in recognizing the language spoken by a person automatically. The need of language recognition increases with the advent of modern technologies such as, language to language translation and multi-lingual speech recognition. Before the era of machine learning, only humans could distinguish among different spoken languages. Language recognition systems try to fulfil the goal of mimicking this human capability. Spoken language recognition is relatively complex task in speech processing facing the main challenge of distinguishing the degree of change in numerous languages. Human language recognition process depends on the phonetic, phono-tactic, and prosodic features of the signal. Modern language recognition systems also use utilize these features to identify language.

2.1.4 Speaker Recognition

Speaker recognition is inverse of speech recognition. Although both methods use signal processing techniques alike. In speech recognition, the goal of processing is to extract linguistic information form speech signal to identify words while keeping out the person's information. On the contrary, speaker recognition focuses on the properties unique to the speaker, disregarding the word or language spoken.

2.1.4.1 Speaker Verification

Speaker recognition works by identifying the speaker based on voice features extracted from speech signal. Voice features are based on physical attributes such as vocal tracts, larynx, mouth, lips and nasal cavities that are used to speak. These are invariant for an individual but behavioral attribute such as the person mental ability to control the muscles or the current mental condition may affect it. Since behavioral attributes are subject to change continuously unlike physical attributes, an ideal speaker recognition system relies on the physical attributes only. However, it is observable that physical attributes like vocal tracts or larynx are difficult to measure. Speaker recognition can be categorized as 1) speaker identification and 2) speaker verification as shown in figure 2-2

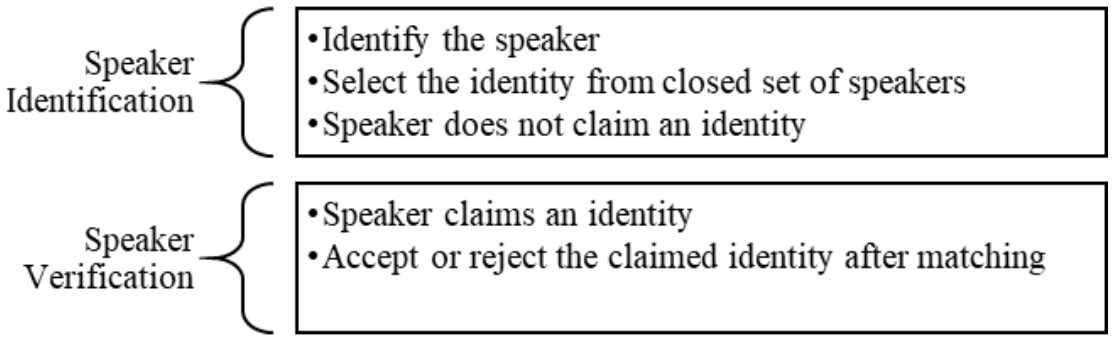

Figure 2-2: Speaker identification vs Speaker verification

In speaker identification, a voice sample from an unknown speaker is analyzed and matched with existing templates of known set of speakers. Then the unknown speaker is classified as the one whose model best matches with the input sample. While in speaker verification, speaker claims an identity and unknown utterance is matched with existing model of claimed speaker. If the matching score is above pre-calculated threshold, the speaker claim is verified. Speaker identification is better choice in forensic applications. Speaker verification is also widely introduced in deployable systems these days and appears as an emerging area. It helps in remote person authentication system deployed by [1]. A smart home security application is developed by [2] that takes instructions from verified speakers. Online or remote workplaces also uses speaker verification in their attendance systems [3].

2.1.5 Text-Independent Speaker Verification

Speaker verification can be further subcategorized into text-dependent and text-independent speaker verification. In text-dependent verification, speaker is enrolled by saying a specific utterance that is known to the system. Later on, the speaker has to say that same predefined utterance in order to get verified. This way phonetic variability in speech signals in compensated.

On the contrary, in text-independent scenario, speaker utterances are not limited to predefined phrases. It offers more convenience to the user and puts load on the system as it become more challenging to extract speaker specific information in speech signal. Hence text independent speaker verification offers a robust authentication system.

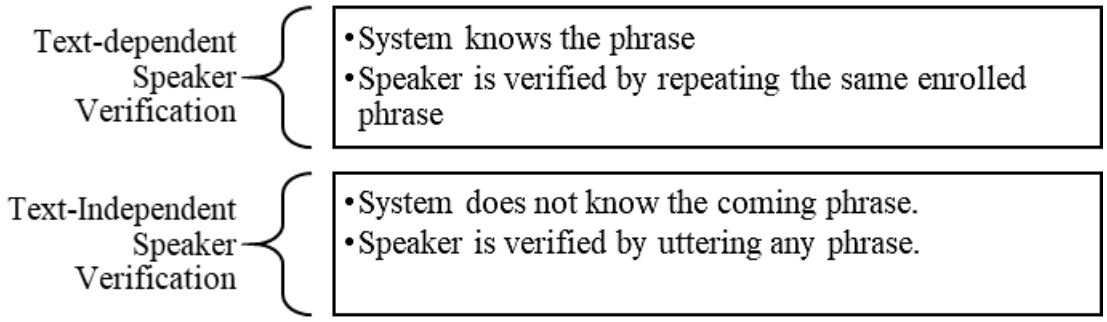

Figure 2-3: Text-dependent vs Text-independent speaker verification

With the advent of machine learning and deep learning and their rapid adaptation to new problems, researchers have focused their research on text-independent speaker verification.

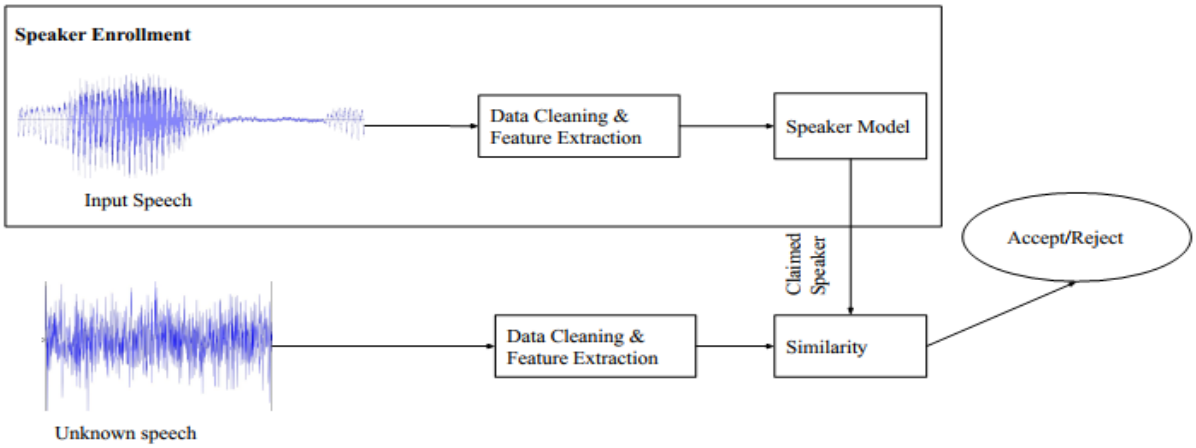

Figure 2-4: Process of Speaker verification

Generally, speaker verification works by first collecting speech samples to all the speakers to be enrolled. Speech data is then cleaned and features are extracted. A respective speaker model is generated on the basis of the unique features. This phase is called speaker enrollment. A voice sample is taken form unknown speaker and it is cleaned and features are extracted. The similarity between these features and claimed identity form the enrollment phase is calculated. The unknown speaker is accepted or rejected as claimed speaker on the basis of pre-defined similarity threshold. The whole process is depicted in figure 2-4.

2.2 Data Preprocessing

2.2.1 Format standardization

Audio data is converted to specific form at before applying further steps on it. There are several standard formats to store audio files. These include Compressed formats e.g., m4a, opus, flac and oog, wma and alac. Other non-compressing formats consist of Wave form audio file format (wav), mp3, pcm, acc.

2.2.2 Resampling

Audio signals are recorded at different sampling rate. Sample rate defines how fast samples are taken to convert sound wave to digital format. For example, 44.1 kHz sample rate reveals that analog signal is sampled 44,1000 times per second. Usually audios are recorded with sampling rate of 48kHz, 44.1 kHz, 16kHz and 8kHz. Higher sampling rate usually provides higher quality and contains more useful information but takes more storage. Since higher sampling rate provides lengthier signal, signal processing time also increases. To overcome the storage and time constraints, decimation or down sampling is applied on the audio signal. Down sampling or signal decimation works by reducing the samples time per second. Figure 2-5 shows the effect of down sampling on a cosine signal.

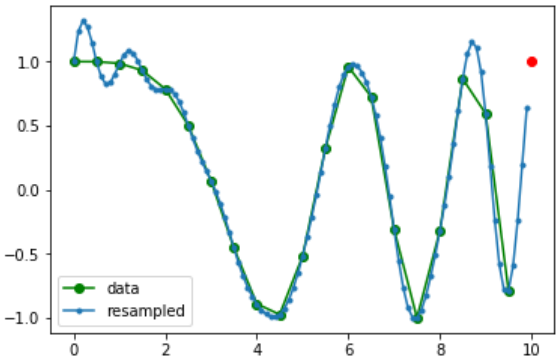

Figure 2-5: Effect of down sampling on a cosine signal

2.2.3 Voice activity Detection

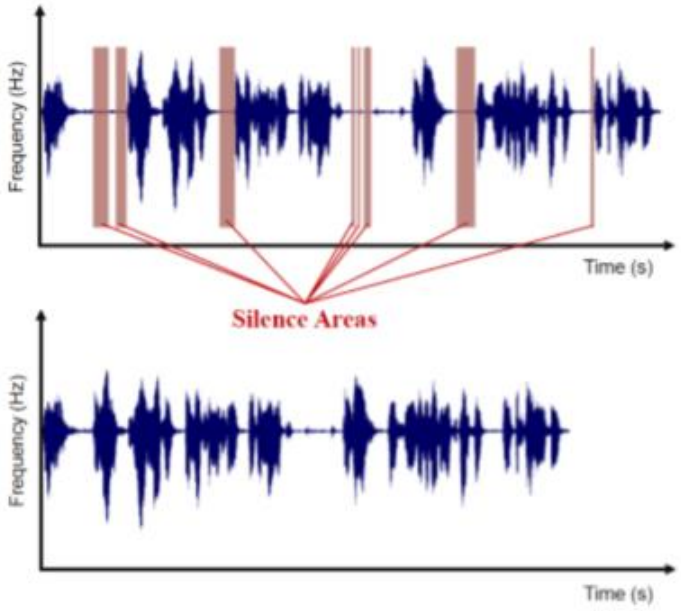

Figure 2-6: An example of applying VAD on a speech signal. Image from [4]

Voice Activity Detection (VAD) is a process to separate the voice segments of audio signal from non-voice segments. This is an important step to take to avoid the effect of non-voice segments on the decision of speaker verification systems and reduce the storage by discarding unnecessary information as in figure 2-6. Several techniques have been proposed over the course of time. Voice

Activity Detection is widely performed by training a model on labeled speech and non-speech parts of audio.

2.3 Audio Representation

Audio signal consist of bunch of raw numerical values. It is hard to extract distinguishing information from these numerical values. Therefore, it is always convenient to represent the signal in a more visual form. Ideally a chosen audio representation should

1. Show lowest within speaker changes in features and high inter-speaker changes.
2. Detect mimicry and imposters.
3. Have high rate of appearance in samples.
4. Be comparatively easy to calculate and use [5].

Some of the widely used visual forms include Mel Frequency Cepstral Coefficients (MFCC), Spectrograms and waveforms.

2.3.1.1 MFCC

Mel Frequency Cepstral Coefficients (MFCCs) are widely used representations in automatic speaker and speech recognition. Speech signal values are constantly changing. To extract the mfccs, first step is to divide signal into short frames by passing windows on them. Higher window size provides more sample values and signal varies too much. While too short windows do not have enough values to estimate reliable spectral. Therefore, medium sized window is e.g., 20 -40 ms is chosen. Next, signal is converted into frequency domain by applying discrete Fourier transform. Signal is warped with mel-frequency to highlight the non-uniformity in the frequency of speech signal. This aspect makes it closer to how humans distinguish sound. Log is applied on

the warped signal and discrete cosine transforms the signal to calculate desired MFCC features while discarding other. This process of representing speech in MFCC is also described in figure 2-7. MFCCs are often extended with delta and delta-delta coefficients.

2.3.1.2 LFCC

Linear Coefficient Cepstral Coefficient (LFCC) are computed similar to MFCC features. A different filter bank is applied that affect the FFT spectrum. Here linearly spaced 26, overlapping filters are applied which causes the equal weight to all frequencies due to its linearity. LOG and DCT is applied afterwards in the same way.

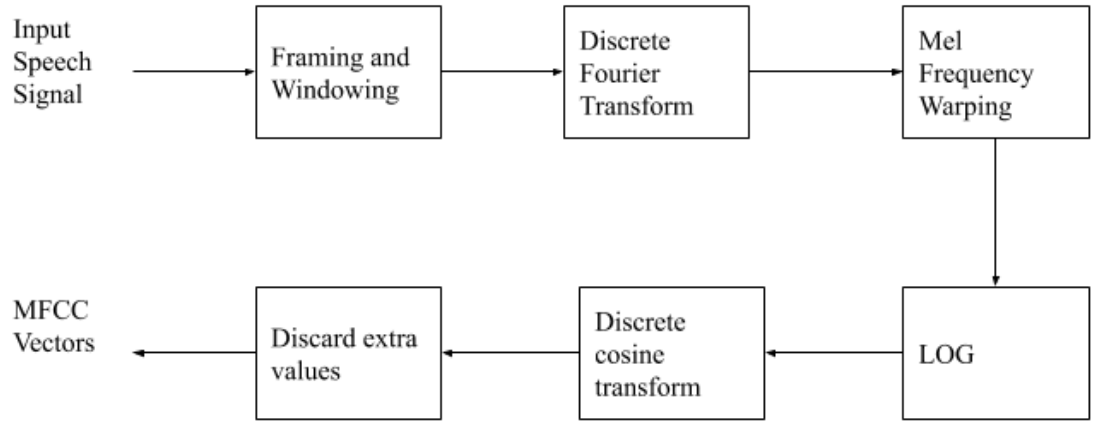

Figure 2-7: MFCC feature extraction steps

2.3.1.3 Spectrograms

A spectrogram is literally an image of the sound. It represents signal strength and loudness in spectrum over time in various frequencies. Spectrograms are also called voiceprints or voicegrams. Spectrograms have varying types for example, STFT (Short Time Fourier Transform) spectrograms and log- mel spectrograms. An example spectrogram of a voice signal is shown in figure 2-8. Spectrograms are proved to provide better results than MFCC and LPCC features. [6]

2.3.1.4 Waveform

Raw audio waveforms have been used in models based on deep learning for sound recognition and classification problems. They do not require any additional pre-processing steps. A sample of raw waveform of human voice is shown in figure 2-9.

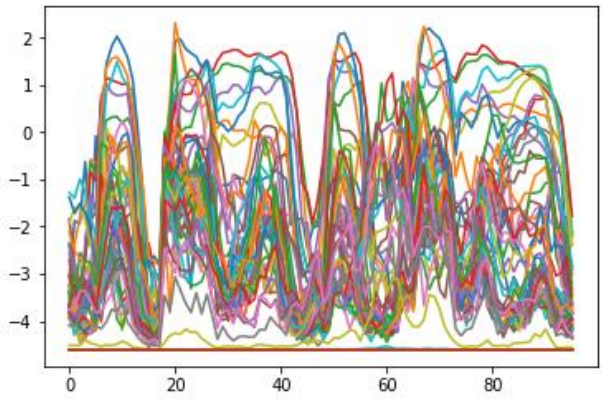

Figure 2-8: Spectrogram of speech signal of 1 second.

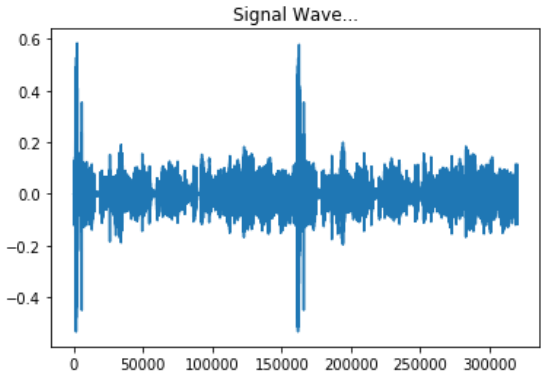

Figure 2-9: Waveform of a speech signal

2.4 Speaker Verification Protocol

A general Speaker verification protocol consists of three stages. 1) Background model training ,2) Speaker enrollment 3) Speaker evaluation. In training, a universal background model i.e., a deep neural network is trained on speaker identification task to make it adaptable to speaker related features. As figure 2-10 explains, labelled audio data is passed through various preprocessing steps described above i.e., resampling and voice activity detection. The preprocessed audio signal is then converted into specific representation i.e., mfcc or spectrograms. A deep neural network is trained for the speaker identification task with softmax activation. Enrollment phase of speaker

verification starts by first gathering the labelled audio data of speakers to be listed in verification system. These audios are then preprocessed by resampling and VAD is applied on them for better results. Audio data is converted in standard representation and passed to the DNN. The DNN trained in the last phase is reused in this phase by removing the last softmax layer from it. The output of second last layer is extracted to generate speaker models. Multiple audios of one speaker are passed through DNN and they are merged to generate generalized speaker models. At the time of evaluation, imposter speech audio is preprocessed and converted to standard representation. This imposter data is then passed to trained DNN to extract speaker model. This speaker model is matched with the model of claimed speaker. Acceptence or rejection decision is made based on the similarity score of two above-mentioned models. Sometimes enrollment and evaluation phases work side by side. Inference process is diagrammatically shown in figure 2-11.

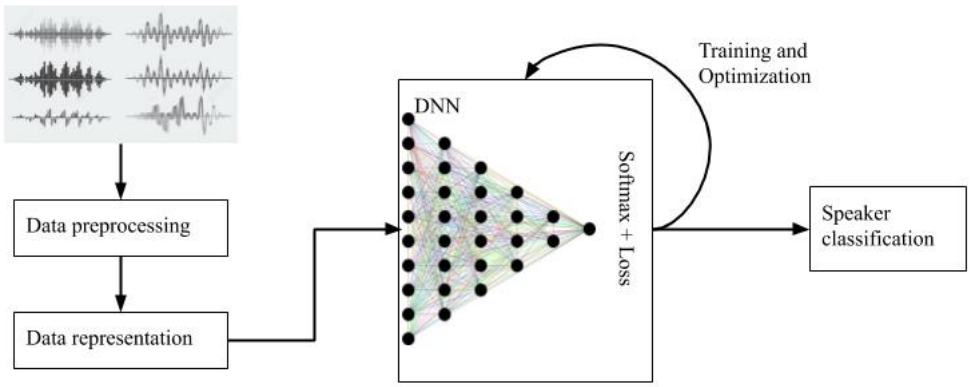

Figure 2-10 Training of speaker verification model

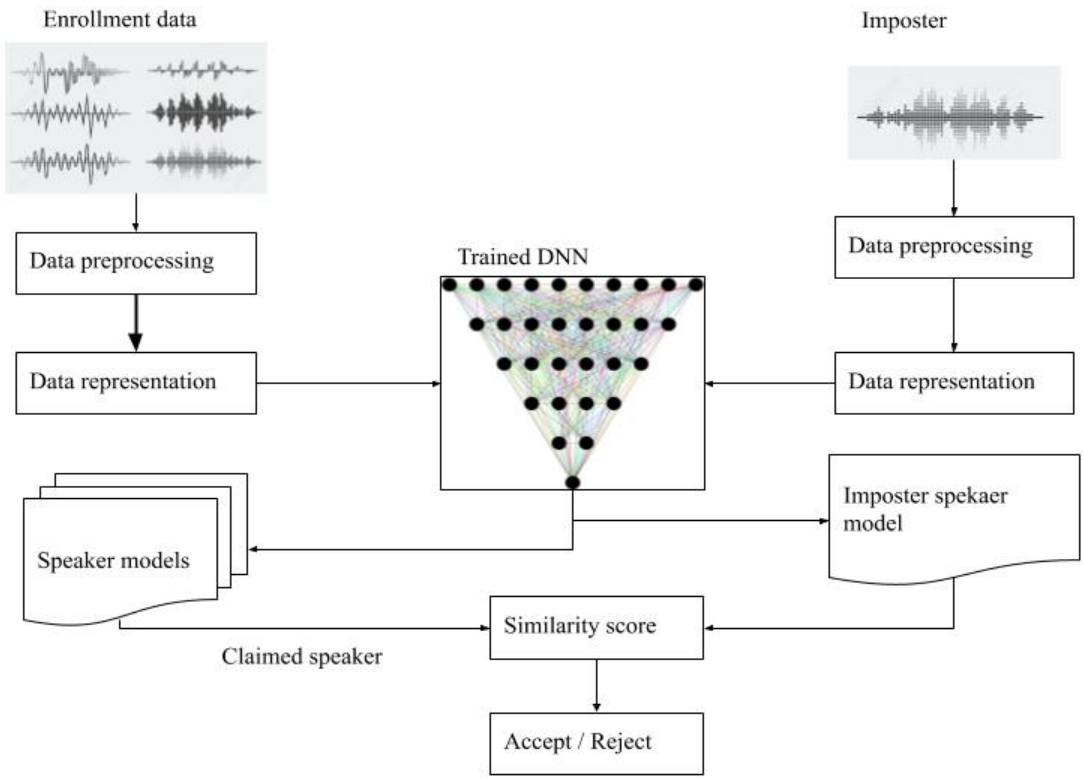

Figure 2-11 Inference from speaker verification model

2.5 Similarity measures

Similarity measures imply how much two objects are alike. For the verification purposes, several mathematical similarity finding functions can be used. Some widely used similarity measures are discussed below.

2.5.1.1 Log Likelihood ratio

Speaker verification must be able to tell if two samples of voice are coming from same speaker. There cannot be a categorical judgement about it. Hence LL ratio [7] was introduced to show the probabilistic score of the decision. LL ratio works as following if

X = recorded voice sample of the speaker

Y = new unknown sample of voice

H_o= X and Y are of same speakers

H_1 = X and Y are of different speakers

E = observed evidence

Then the LLR formula is as following

$$LLR = \frac{p(E|H_{o)}}{p(E|H_{1)}}$$

2.5.1.2 Cosine Distance

Cosine distance calculates the degree of angle between two points or objects. It is widely used in similarity checking of documents as well. It works well when orientation of the vector matters instead of their magnitude.

$$\vec{a}.\vec{b} = \|a\|.\|b\| \cos\theta$$

Cosine value of 1 shows the large extent of similarity between two points. Similarity is not present in case of cosine 0.

2.5.1.3 Euclidean Distance

Euclidean distance is most commonly used measure to calculate similarity or distance.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Euclidean distance calculates the distance between two points in a plane. Less Euclidean distance indicates more similarity.

2.5.1.4 L1 Distance

L1 distance is absolute sum of difference between two cartesian points in a grid like path.

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Here, n is the number of points or variables in vector x and y. Less L1 distance indicates more similarity. L1 distance is also called Manhattan Distance.

2.5.1.5 Siamese Network

Siamese networks are special purpose networks that share two identical networks. These networks have same shape, parameters and configuration. Parameters are updated simultaneously in both networks during training. This framework has been successfully applied in verification problems like face verification and signature verification [8, 9]. These subnetworks are merged by a loss function to compute similarity scores of the features calculated by each network. Contrastive loss is the most widely used loss function in Siamese networks [9]. Contrastive loss is defined below

$$(1 - Y)\frac{1}{2}\,(D_w)^2 + (Y)\frac{1}{2}\{\max(0, m - D_w)\}^2$$

Here Y is 0 for similar input and 1 in case of different class. D_w is Euclidean distance of predicted and target values. Unlike traditional approaches, binary labels are not assigned immediately to the outputs rather Siamese networks work in a fashion that brings similar impostor and original inputs together and push the dissimilar pairs far away from each other. If we see each branch of Siamese

network as a function to map inputs into a space, this loss function has the property to map the different embeddings far from each other into the spaces while keeping the same embeddings near to each other. Both networks are joined with a merge layer. In order to decide if two audios belong to the same class, one needs to determine threshold value for the merged layer.

2.6 Transfer Learning

Transfer learning is also a machine learning technique that refers to the reuse of a model trained on one task to perform another related task. The goal here is to leverage the knowledge gathered in source task for the better results in target task. Deep learning models require very large amount of data and take more time and resources to train. Transfer learning on the other hand give better initial performance, fast convergence and high-speed training.

2.7 Traditional Speaker recognition methods

Different approaches have been used in research for text independent speaker verification give their features. The goal of the model used is to bring together all utterances of same speaker together while keeping unknown utterances far. Generative and discriminative algorithms are used for this purpose. Generative models explicitly use all data samples to build models for each class. Then map new samples with all models and look for the similarity or higher probability. While discriminative models learn the boundary between multiple classes through a function. They do not require training samples after learning the boundary. Both of them have their perks. Algorithms, covering both aspects, used in speaker verification are described below.

2.7.1 K-nearest neighbor

k-nearest neighbor is the simplistic one in discriminative learning algorithms. It just requires one distance parameter to train. A kNN is provided with labelled training data in the form of labelled vector. An unknown sample is identified as one or two groups depending upon the distance from the unknow sample [10].Even with its straight forward identification approach, It is costly and unproductive due to following basis [11]. Firstly it has to store all training data to make predictions. Moreover, all computational process is performed when unknown samples arrive to get predicted. If the unknown sample falls in two groups, it does not effectively take decision to map the sample in one group. The effect of these drawbacks is reduced by using kNN with other classifiers[12].

2.7.2 Hidden Markov model

In sound processing techniques, order of spoken words and their content are always important equally. Hidden Markov Models (HMM) deal with time series of observed data, hence proved as powerful statistical approach in text dependent speaker verification. [13, 14] .The verification process works by building HMM for each speaker. Distance of testing sequence is measured with all speakers and the HMM providing highest probability is chosen to be predicted. HMM also faces the shortfalls accompanied by generative models namely 1) more computational power and 2) large data and memory.

2.7.3 Gaussian Mixture Model

In text-independent speaker recognition, Gaussian Mixture Model [15] has been very popular approach for years. GMM is a generative approach to find probabilistic model of a speaker through voice. Mean vector and covariance matrix from all data are combined to form GMM. Each speaker

has its own GMM parameters for speaker identification. GMM models are trained by estimating likelihood. A GMM model makes its prediction by returning the speaker whose utterances maximize the posterior probability. Give an utterance T and parameters y_1 , y_2 to y_n , the posterior probability can be defined as below :

$$P(T|y) = argmax_{1<k<n} P(T|y_k) = \frac{P(T|y_k)P(y_k)}{P(T)}$$

Regardless of high performance, GMM still face some drawbacks [16]. GMM parameters are very large in number that requires more computational power to train. The results are not reliable in case of small training data. Like other generative models, GMM yields poorly on unseen data. Above mentioned two problems are overcome by constructing speaker dependent GMM. Main idea is to train GMM on all speaker identities at once to make a Universal Background Model(UBM) . [15]

The posterior probability of GMM based UBM is also calculated using support vector machine kernels as proposed by [17]. [1] has deployed a GMM based speaker verification system that works over telephone line. MFCC features are extracted after VAD, UBM is applied on the GMM supervectors. Loglikelihood ratio is used as scoring measure. But The system fails to perform good in case of gender and age variations.

2.7.4 GMM-UBM based i-vectors

Extracting UBM form GMM for all speakers is called speaker adaptation. Calculated mean vectors for an individual speaker is called GMM supervector. The GMM supervector can also be calculated for an utterance in the same way. These supervectors contain large data. Joint factor

Analysis decomposes the supervector in a low dimensional space called total variability space or i-vectors. I-vectors show poor performance if the new enrolled speaker utterances are short. As these are unsupervised models, it is difficult for them to learn supervised speaker discriminative features hence they are not suitable for verification process. This drawback is overcome by using any supervised model like Support Vector Machine (SVM) with GMM-UBM [17] or Probabilistic Linear Discriminant Analysis (PLDA)-based i-vectors model [18]. These i-vectors are used in SVM by [19] . It showed that SVM divides all English speakers in a linear fashion. The results improved when applied cosine kernel with SVM.

2.7.5 Naïve Bayes

Naïve Bayes supervised algorithms are based on Bayes theorem based on independence assumption that all features are independently related to the label. This model is easy to build as no parameter selection or estimation is needed to perform. This property makes it useful for large amount of data. In spite of all its simplicity and assumptions, it surprisingly performs well and outperforms other advanced machine learning algorithms. Authors of [4] have used Naïve Bayes as classifier for speaker verification specifically designed for mobile devices. VAD and audio normalization affects the results in positive manner. 20 LPCC features are extracted to perform verification. There are multiple downfalls associated with this technique. The independence condition works well in certain problems but if data is not completely balanced, it often causes bad results. As Naïve Bayes requires continuous features to be binned, a lot of important information can be lost from some samples even in efficient binning methods.

2.8 Deep learning based methods

Adequately sufficient statistics are gathered by using deep learning methods instead of traditional GMM-UBM framework. [20] uses raw waveforms to extract embeddings for speaker verification from CNN-LSTM models. Binary vectors are extracted from the embeddings and passed to fully connected layers. The results were reduced by applying same technique on spectrograms. [21] has proposed a method called x-vectors replacing i-vectors. X-vectors are extracted by training a deep neural network on speaker discrimination task then extracting features from the trained DNN. X-vectors perform better than i-vectors in case of short utterances. [22] has proposed a model for domain mismatched and language mismatched speaker verification. Speaker embeddings are extracted using DNN proposed by [21] . PLDA is applied on the embeddings for scoring. [23] improved the x-vector performance by introducing Adversarial networks as language classifier on top of x-vector extraction DNN adapted by the method proposed by [24]. [24] proposed adversarial training for language independency in sentiment analysis. Adversarial network is trained on x-vectors and resulting embeddings are called Adversarial Discriminative Domain Adaptation (ADDA) embeddings. It shows promising performance with PLDA log likelihood scheme. But this method takes a lot of training time and resources. As two networks have to be trained separately.

Deep Neural Network is applied on MFCC, LFCC and LPCC features extracted form voice by [25]. A new scoring method is also proposed namely bet Bernoulli. This method requires a lot of calculation to extract these three kinds of features. [26] uses CNN as feature extractor which was previously proposed by [27] and has been in use for image classification and speech-related applications successfully. The CNN is trained for speaker identification then put in Siamese[9]

settings. Two same shared CNNs are implemented for verification purpose. L2 distance is applied on the outputs to accept or reject.

A 3d convolutional neural network on top of MFCC like MECC features is proposed by [28]. Although it provided optimistic results but more research is needed in this regard as results are still low as compared to other approaches because of applying CNN on the selected features.

Siamese neural network as proposed in [9] and implemented in many research efforts [8, 29-32]. Transfer learning has been proved to be an efficient way to tackle the classification problems. We also investigated the use of transfer learning of CNN and Siamese networks in speaker verification. [33, 34] extracts CNN embeddings and average them out to create speaker models. But CNN requires very large data to be trained effectively that is why transfer learning provides better results. Transfer learning is a technique in machine learning that trains the network on one task and then uses the trained model to do another task that is relatively similar to the previous one. Transfer learning has shown good results in image classification [35], speech recognition [36, 37] , ASR and verification [38, 39], medical image classification [40] and face verification [41]. [42] has used transfer learning for speaker verification purpose by using an inceptionResNet v1 to extract speaker embeddings. [43] proposed a non- probabilistic scoring technique by applying binary operations on i-vector of each pair of utterances. Binary operations include element wise dot product and sum. The binary vector is passed to a deep neural network based classifier by adding few dense layers. They showed great performance on NIST SRE corpus. The concise comparison of above-mentioned approaches is presented in table 2-1.

Table 2-1: Comparison of methods in speaker verification

Paper	**Technique**	**Pre processing**	**Dataset**
[10]	K-nearest neighbor	Spectral features	15300
[13, 14]	Hidden Markov model with loglikelihood ratio	Spectral features	-
[15]	Guassian Mixture Model, Universal Background Model	MFCC feature extraction	NIST SRE (1999)
[17]	GMM-UBM with support vector machine	16 MFCC with Delta feature	NIST SRE (2005)
[18]	i-vectors with PLDA analysis		NIST (2008)
[19]	i-vectors with cosine kernel SVM	19 MFCC, delta and delta features	NIST SRE (2004) and Fisher English
[4]	Naïve Bayes	VAD, Audio normalization and LPCC features	Self-captured and TIMIT
[20]	CNN-LSTM to extract embeddings, and binary vector based scroing	Raw waveforms	Voxceleb
[21]	x-vectors	20 MFCC features after VAD,	US English telephone speech
[23]	x-vectors from Adversarial Discriminative Domain Adaptation with PLDA log likelihood	23 MFCC after VAD	SRE04-08, Mixer6and Switchboard for English, AISHELL-ASR0009 for Chinese corpus
[25]	DNN with Bet Bernoulli scoring	MFCC and LPCC feature extraction	58k utterances (Madarin and tagolog)

Paper	Technique	Pre processing	Dataset
[26]	CNN and L2 distance	40 log-energy of filter banks per hamming window alongside their first and second order derivatives are generated to form 3 × 40 × 100 input feature map.	No. of speakers: 1251 Utterances: 145,124
[28]	3D CNN	VAD and MECC features	Multi Modal
[43]	i-vector SVM with binary operations kernel	60 MFCC features after VAD	NIST SRE (05)

Chapter 3

3 Materials and Methods

This Chapter explains the datasets used and proposed pre-processing and method for speaker verification.

3.1 Datasets

3.1.1 Voxceleb2

Voxceleb2[44] comprised of a collection of over 1 million utterances for 6,112 celebrities on YouTube. The dataset is 61% gender balanced with male population. The speakers stretch on a wide scale of different ethnicities, accents, professions and ages. Audios present in the dataset are degenerated with background chatter, laughter and varying room acoustics to diversify them. Approximate length of utterances is 4 – 20 seconds and there are on average 185 utterances per person. All speakers are talking in English in their native accents.

3.1.2 Self collected Dataset

We have collected our own dataset for this research of native Urdu Speakers. Their audio files are recorded across multiple devices and multiple environments I.e., libraries, outdoors and classrooms. Data recording techniques very from Whatsapp audios to standard voice recording applications. Dataset is recorded in Urdu, Arabic and English languages. Female population consists of 60 % of data making is fairly gender balanced. Data is divided into training and test sets. Speakers in training set have average recorded audio of 15 minutes. However, test set speakers have approximate length of 5 minutes. Recorded audios span over speakers of different ages

similar to Voxceleb2. The audios are not cleaned from background noise and clutter to add generalization. We have not labelled our dataset on the basis of gender hence it is more robust towards identifying both male and female speaker equally. Table 3-1 shows the description of our collected dataset briefly.

Table 3-1: Description of our self collected dataset

Total no. of speakers	40
No of female	60%
No of male speakers	40%
Languages	Urdu, English and Arabic
Age	Kids, teenagers, adults, elder

3.2 Proposed Approach

The proposed approach for speaker verification is discussed below.

3.2.1 Preprocessing

Data is preprocessed to convert it to a standard format before passing to the model figure 3-1. All audio files are converted into .wav format. In the next step each audio frame is classified as voice or not voice also known as Voice Activity Detection (VAD) [45]. Only voice parts of audio are passed to the next step. Voice data is resampled to 16 kHz and normalized between -1 to 1. Resampled data is divided into frames each of 0.96 seconds and decomposed with a short-time

Fourier transform by applying 25 ms windows with 10 millisecond step size. The resulting spectrogram is integrated into 64 mel-spaced frequency bins, and the magnitude of each bin is log transformed after adding a small offset to division by zero and zero logs. This gives log-mel spectrogram patches of 96 × 64 bins forming the input to the Neural Network.

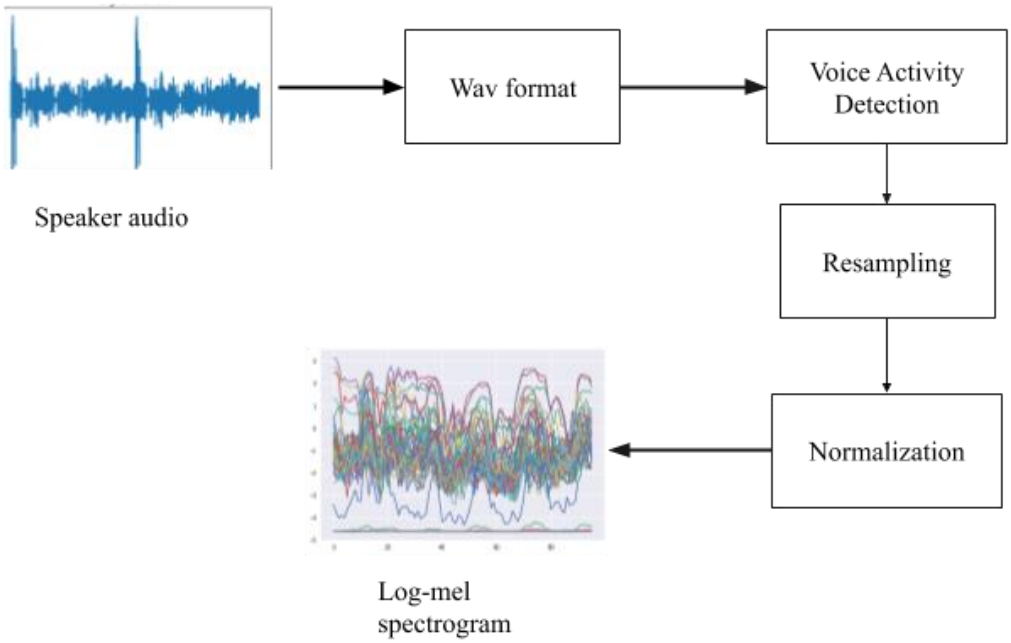

Figure 3-1 Proposed preprocessing steps

3.2.2 Speaker model generation

3.2.2.1 Deep Neural Networks

Artificial neural network (ANN) is a very common algorithm in machine learning. They are called neural as the try to mimic human brain in identifying things. An artificial neural network consists of input, output and optional hidden layers. Each layer of neural network is a group of neurons. Neurons in each layer are connected to other layers and these connections have weights. Neural networks are trained by exposing labeled data to it in an iterative process called backpropagation.

Backpropagation works by going backwards after each iteration to update the weights of the neural network. The extent to which weights can be updated is called learning rate. Neural network learns in bigger steps if the learning rate is higher while convergence is slow in small learning rate(lr).

Complex neural networks are called deep neural networks (DNN). They have more hidden layers then standard neural networks. As deep neural network has large structure, its training process is more challenging than that of standard ANN. Several kinds of DNN have been proposed in the past. Namely Convolutional Neural Networks, Long Short-Term Memory networks (LSTM) and recurrent neural networks (RNN).

3.2.2.2 Convolutional Neural Networks

CNNs are specially designed DNNs that work best in capturing spatial and temporal information from the data i.e., images. It reduces the effort to find best hand-crafted features from the images. The convolutional layers in CNN make it distinguishable from other neural networks. Convolutional layer comprises of filters whose parameters are learned in training process. Each filter creates feature maps from input by convolving on them. Convolving is a process of sliding the filter window, also called kernel, across the width and height of the input data at every spatial position. Higher filter size extract higher level of information from data while small filters extract tiny details. Output of convolutional layer is stacked activation or feature maps of all filters. Activation function is applied on the outputs to decide which values to keep. Typically, ReLU is applied in convolutional layers. ReLU works by replacing all negative values to 0 and positive values remain same.

$$R(x) = \max(0, x)$$

Input at each layer is down sampled before going to next layer. This way they find more generalized and abstract features of input data. Parameters and weight dimensions are reduced by using pooling layers in the network, an nxn pooling layer returns a single winning value for all nxn blocks. The output of convolutional layer is passed to fully connected layer after flattening it. The fully connecting layer has all neurons connected to each other as the name suggests. It helps in transforming the input into specific classes. One or more Fully connected layers can be used in a CNN. Last layer is called output layer. Sigmoid or softmax activations are used in last layer. Sigmoid activation works best in binary classification tasks. It places all values form [-1 to 1]

$$\varphi(z) = \frac{1}{1 + e^{-z}}$$

A softmax activation function is more generalized form of sigmoid for multiclass classification. Large datasets are passed to the network in batches which may results in overfitting according to the specific batch. This effect is neutralized by using batch normalization layers in between convolutional layers. Batch normalization layers normalizes each batch of input by transforming the data into zero mean and unit variance distributions. This layer does not work in testing or inference. After each pass of data, difference between predicted and original output is calculated by a loss function. Famous loss functions include Mean square error, cross entropy error and cosine loss. Optimization techniques are adapted to reduce the loss and update the weights and parameters of the network accordingly. Gradient Descent, Adagard and Adam are most widely used optimization functions. Adaptive Moment Estimation (Adam) calculates the learning rate at each iteration therefore it outperforms other optimization functions.

3.2.2.3 Proposed Speaker embedding model

Our proposed approach uses a CNN trained for speaker classification to extract speaker embeddings for the verification presented 3-2. It reconfigures the original Visual Geometry Group (VGG) neural network [46] by changing its final layer and adding batch normalization to get audio embeddings. The fifth convolutional block is also removed thus reducing the total number of parameters from 144M weights and 20B multiplies to 62M weights and 2.4B multiplies. The final network architecture, illustrated in figure 1, consists of four convolutional blocks having convolution and max pooling layers. Followed by a fully connected block having two dense layers and an embedding layer. More details about this network can be found in [47]

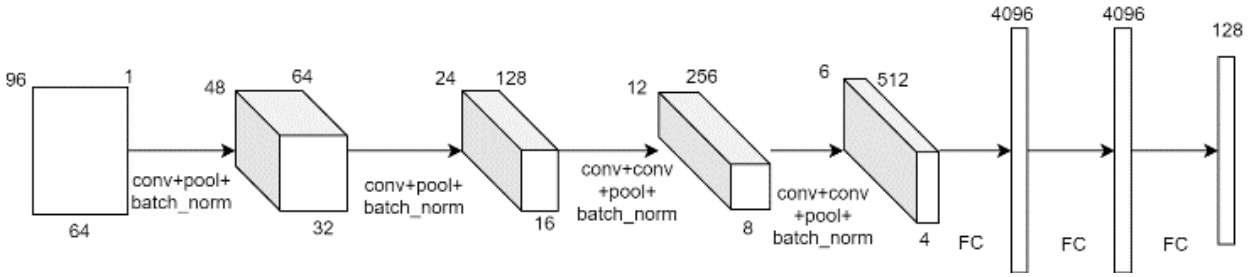

Figure 3-2 Architecture of CNN to extract embeddings

More details about filter size, filter height and width, strides and padding are explained in table 3-2. In CNN, stride indicates the distance between subsequent samples achieved by applying convolutional and max pooling filters. ReLU activation is applied in all layers, activation decides whether a neuron should fire or not. Padding indicates the type of pixels applied on the boundary of the input so that filters cover whole input. Same padding is applied to input in all layers so that output has same length as original input.

Table 3-2 Layer wise configuration details of embedding CNN

Blocks	**Size** **(Number of filters x height x width)**	**Parameters**
Input	64x96x1	strides= 1 , padding='same’
Convolution 1	64x3x3	strides= 1 ,padding='same’
Pooling 1	2x2	strides= 1, padding='same’
Batch_normalization 1	-	-
Convolution 2	128x3x3	strides= 1, padding='same’
Pooling 2	2x2	strides= 2, padding='same’
Batch_normalization 2	-	-
Convolution 3_1	256x3x3	strides= 1, padding='same’
Convolution 3_2	256x3x3	strides= 1, padding='same’
Pooling 3	2x2	strides= 2, padding='same’
Batch_normalization 3	-	-
Convolution 4_1	512x3x3	strides= 1, padding='same’
Convolution 4_2	512x3x3	strides= 1, padding='same’
Pooling 4	2x2	strides= 2, padding='same’
Batch_normalization 4	-	-
Fully connected 1_1	4096	-
Fully connected 1_2	4096	-
Fully connected 2	128	-

First convolutional layer filters the 96 x 64 input Mel spectrogram with 64 kernel windows of size 3 x 3. The second convolutional layer accepts normalized and pooled output from first layer and apply 128 kernel windows of size 3 x 3. Third convolutional block has two consecutive convolutional layers followed by pooling batch normalization layer having 256 kernel windows of size 3 x 3 each. Forth convolutional block is connected to the third one with 2 convolutional layers of 512, 3x3 kernel windows each followed by pooling and batch normalization layers. This brings about the neural network learning more high-level abstract features and less low-level specific features. The output of last convolutional layer is flattened and pass to the first fully connected layer having 4096 nodes. The second fully connected layer also has 4096 nodes. Finally, last fully connected layer has 128 neurons. This stipulates that each audio is converted into embedding of size 128 each. We initialized the weights of model according to the one presented in [47]. The model is trained using Adam optimizer for 10 epochs. Training starts with learning rate of 0.003 with hyper parameter epsilon 1e-8. All these values are shown in Table 3-3.

Table 3-3 Adam optimizer parameters value

Parameter	Value
Learning rate	0.003
Weight decay	0.0
Epsilon	Le-4

The framework is implemented in Keras[1] library using Tensorflow[2] backend. The model was trained using GPU on cloud Google Colaboratory[3].

3.2.3 SpeakerNet : Distance Learning model

The trained convolutional model is used to generate embeddings for each speaker audio. The median is calculated for embeddings of 10 seconds for each speaker. These mediated embeddings are passed to the Siamese network. Architecture for Siamese is presented in Table.3-4.

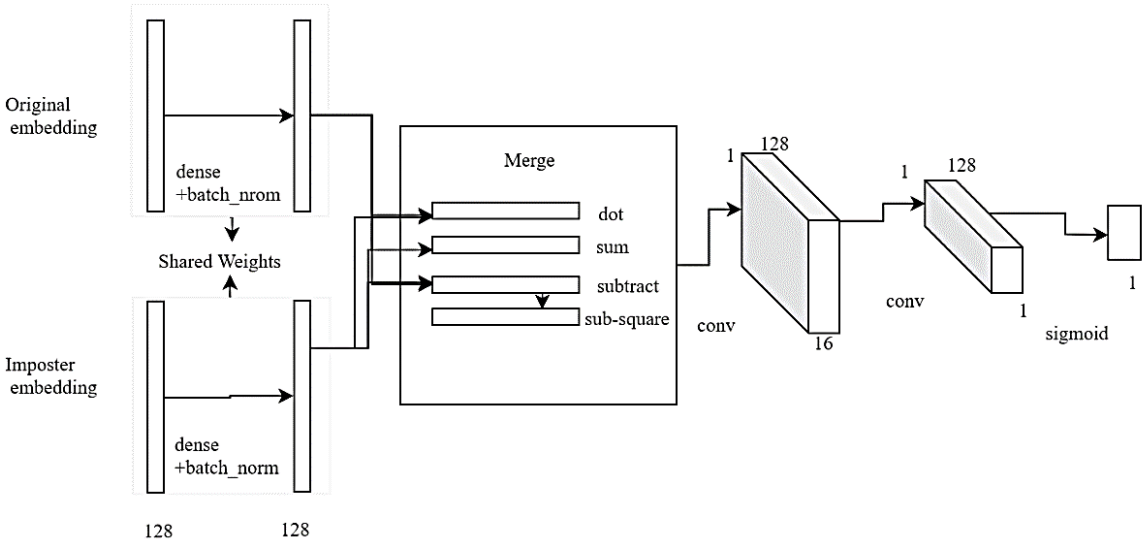

Figure 3-3 Architecture of proposed SpeakerNet

[1] http://keras.io/

[2] https://www.tensorflow.org/

[3] https://colab.research.google.com/

Each branch of Siamese has fully connected layer of 128 nodes followed by dropout and batch normalization. The outputs of these branches are merged by multiplying, adding, subtracting and taking the square is applied on stacked output of last layer with 16 kernels of size 4 x 1 each of difference. Output of above-mentioned operations is stacked on top of each other. Convolution

The output is again reshaped into 1 x 16 x 1. Second convolution is applied on reshaped input with 1 kernel of size 1 x 16. This gives a vector of size 128. L1 kernel regularizer and ReLU activation is used in all of above layers. Lastly a fully connected layer with sigmoid activation is applied and result 0 is deduced as same person while 1 as fake. Figure 3-4 visualizes the SpeakerNet.

Table 3-4 Layer wise configuration details of proposed SpeakerNet

Layer	**Size**
Fully connected 1	128
Batch Normalization 1	-
Multiply	128
Plus	128
Subtract	128
Square of subtract	128
Convolution 1	16 x 4 x 1
Convolution 2	1 x 1 x 16
Fully Connected 1	1

Chapter 4

4 Results and Discussion

This chapter discusses our baseline models used in experiments followed by the process of experimentation. The results are presented in various forms to analyze. In order to evaluate our speaker verification algorithm, we have used a benchmark dataset Voxceleb2 [44] and a self-collected dataset of multiple languages by native Urdu speakers.

4.1 Baseline Systems

4.1.1 L1- distance Siamese

We have implemented a simple Siamese on top of our embeddings as baseline model. This Siamese network uses L1 distance as similarity measure. L1 distance Siamese takes genuine and imposter pairs and calculate their L1 distances. A threshold is applied on the distances to accept or reject them. The threshold is applied in the last sigmoid layer of the model. Contrastive loss is used to minimize loss of this network.

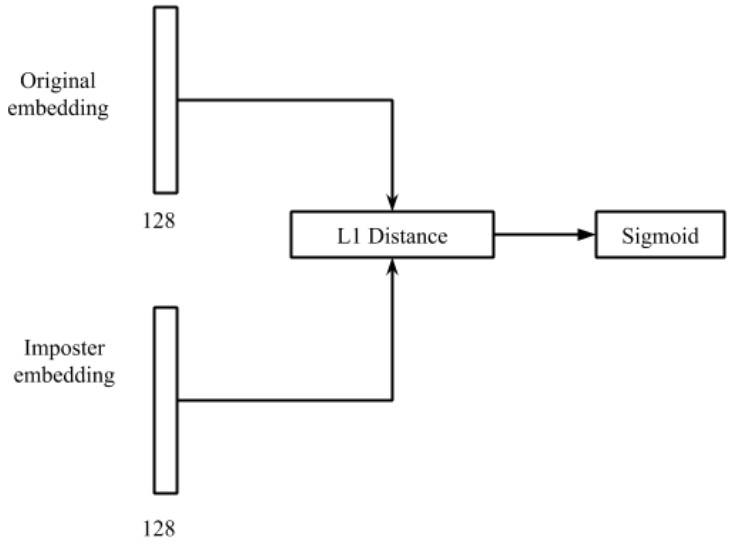

Figure 4-1: Baseline model consisting Siamese with L1 distance layer

4.1.2 Cosine Distance Siamese

Second baseline model also utilizes same pre-processing and embeddings steps. It calculates cosine difference instead of L1 distance and activate the results with sigmoid activation. Contrastive loss is used to train the Siamese.

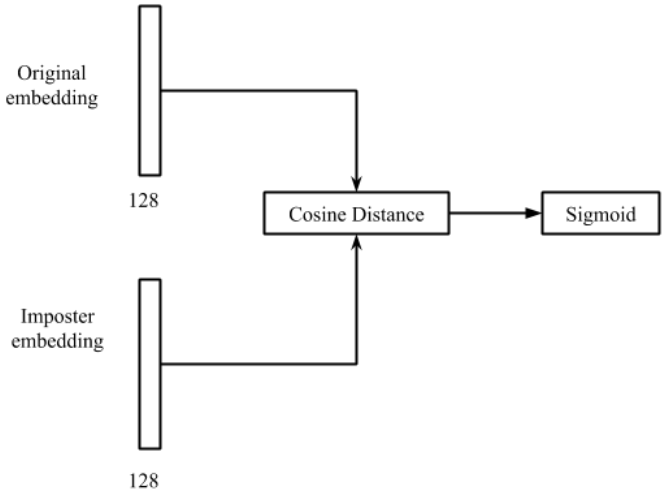

Figure 4-2: Baseline model consisting Siamese with cosine distance layer

4.1.3 Binary vectors

Another method used as baseline is the variation of SpeakerNet proposed. SpeakerNet apply convolution on top of binary vectors. Here we only calculate b-vector from our original and imposter embeddings by concatenating their sum, difference, dot products and square of difference. This long b-vector is passed through sigmoid layer to get output scores. No convolution is involved here.

4.2 Experimental Protocol

As our model is designed for text-independent speaker verification, we divided the above two datasets for training and evaluation of the model. We randomly selected 20 speakers from voxceleb2 and 20 from our self-collected dataset. We generate equal number of pairs for positive and negative samples for each batch. For the evaluation phase, 5 speakers from voxceleb2 and 20 from our own datasets are chosen. Both training and evaluation data is gender balanced. Total duration of samples in both phases is described in table 4-1.

Table 4-1: Training and evaluation data

	Training	**Evaluation**
No. of speakers	40	25
Total duration of samples (seconds)	30,000	7,500
Duration of samples per speaker (Seconds)	900	300

Each audio is divided into 10 sec length and mel-spectrogram is calculated for each second then passed to CNN. The extracted embeddings from CNN are then averaged. The process is repeated on each audio again after adding gaussian noise to make it more robust. Contrastive loss margin is set to 1 in training.

4.3 Evaluation Metrices

The performance of experiments is evaluated and compared using accuracy, area under the Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) and Equal Error Rate (EER). Accuracy is the ratio of true predicted values to the total number of input samples as shown in equation.

$$Accuracy = \frac{TN+TN}{TN+FN+TP+FP} \qquad (1)$$

ROC curve is a graph that shows overall performance of the model on all thresholds. The graph shows two measures, True Positive Rate (TPR) and True Negative Rate (TNR). While higher AUC depicts the model's superior performance for distinguishing original and imposter speakers. EER is an algorithmic approach that measures error margin of a biometric system by utilizing TPR and TNR.

4.4 Results

The comparison of accuracies, AUC and EER of our proposed SpeakerNet with the baseline models are given in Table 4-2.

Table 4-2 Comparison of proposed model with baseline models

Model	**Method**	**Accuracy (%)**	**AUC (%)**	**EER**
Baseline	Embedding CNN + L1 distance Siamese	85	88	0.158
Baseline	Embedding CNN + cosine distance Siamese	85	97	0.073
Proposed	Embedding CNN + binary vectors	92.9	98	0.061
Proposed	**Embedding CNN + SpeakerNet**	**93.08**	**98.5**	**0.026**

The Accuracy is lowest when we use L1 distance or cosine distance. But table also shows that area under the curve and EER for cosine distance is better than L1 distance. The reason for that is cosine distance maps the values closer to original labels i.e., 0 or 1. This behavior can be witnessed in their ROC curve as well. We analyzed the scores from L1 distance and cosine distance (see figure 4-3 and 4-4). It is evident that L1 distance does not draw the similar pairs together and different pairs far from each other. Instead they are closed enough hence it fails to find a proper threshold that hold for all values. While cosine distance marks the pairs closer to their labels and an appropriate threshold can improve results.

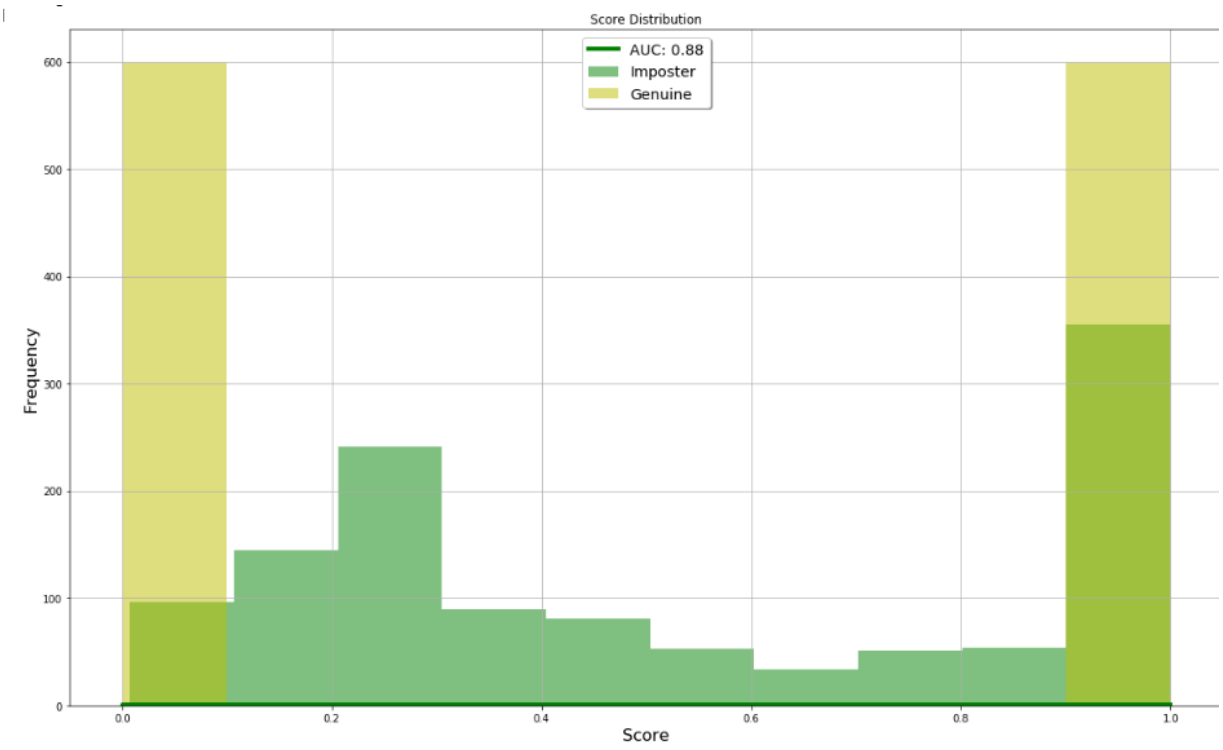

Figure 4-3: Score distribution for L1 Siamese

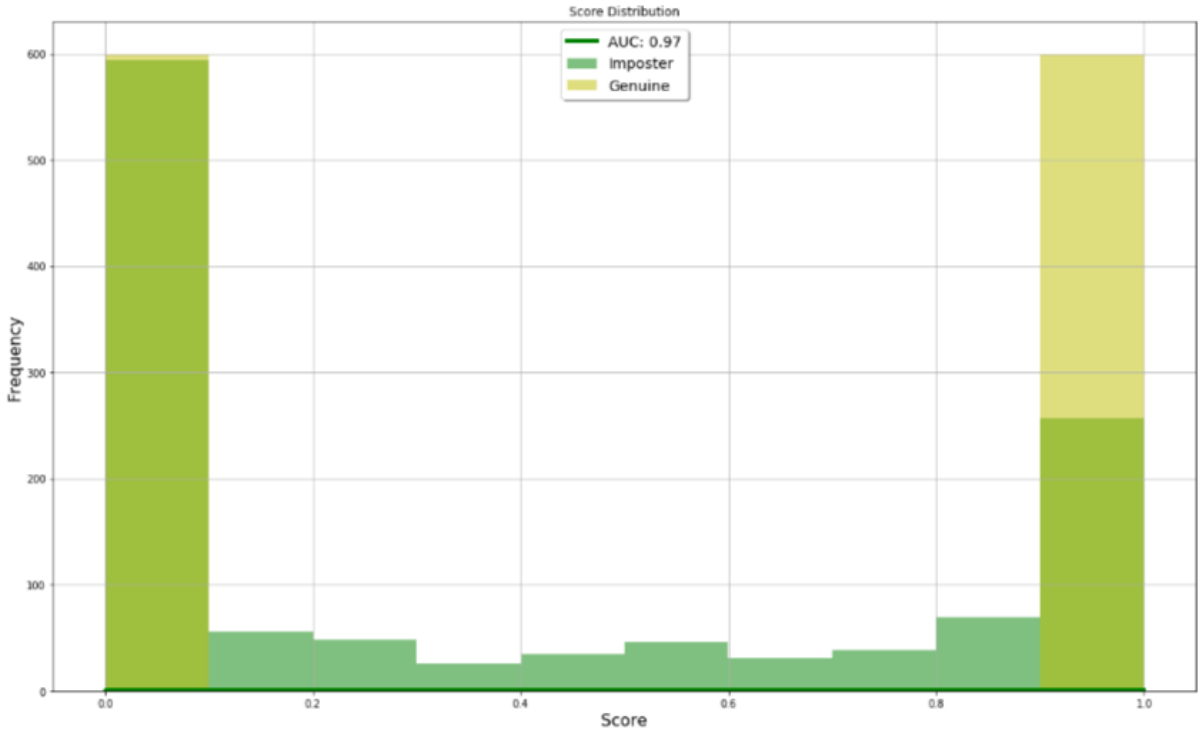

Figure 4-4 Score distribution for cosine Siamese

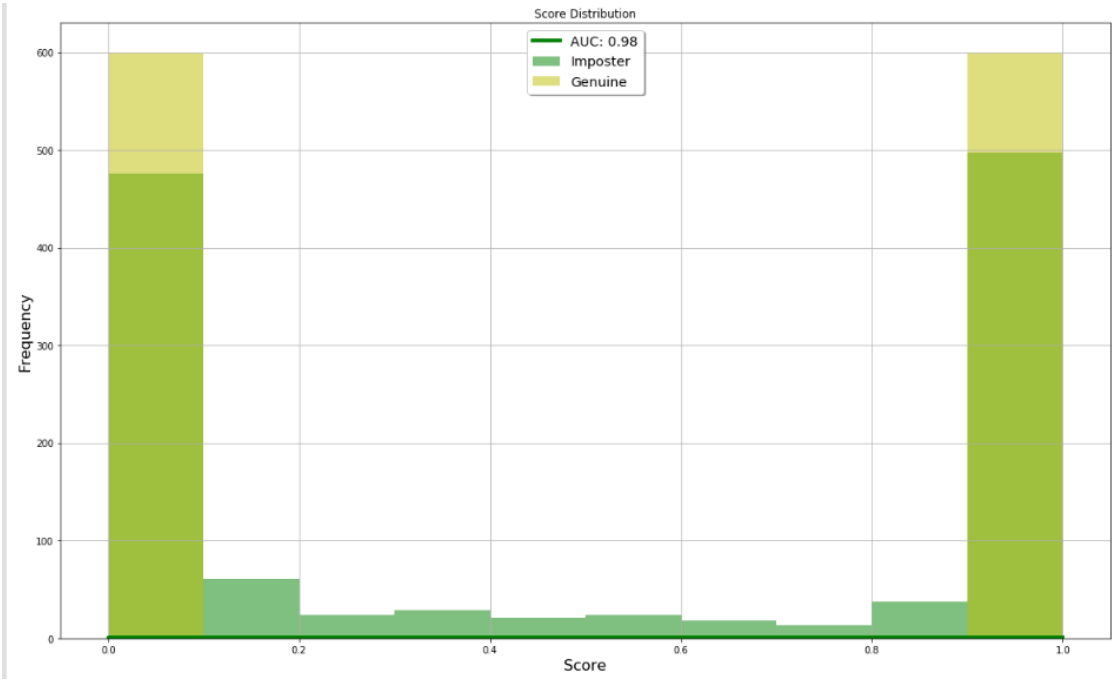

Figure 4-5: Score distribution resulting from b-vectors

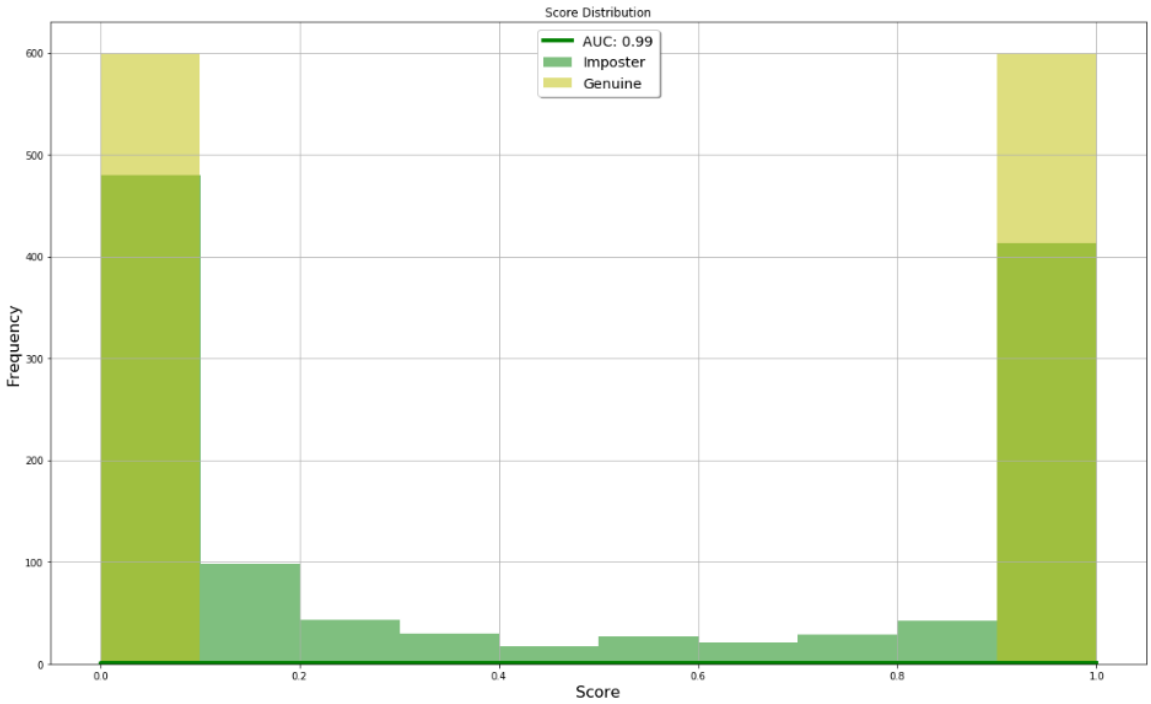

Figure 4-6 score distribution resulting from SpeakerNet

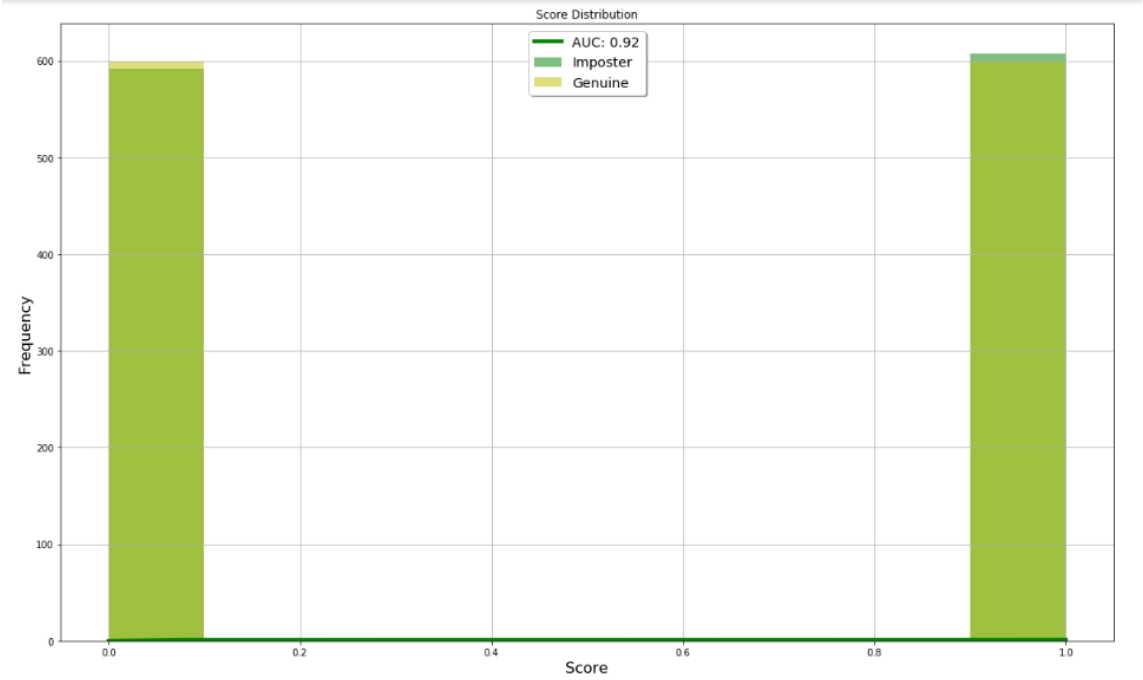

Figure 4-7: Score distribution of SpeakerNet at 0.4 threshold

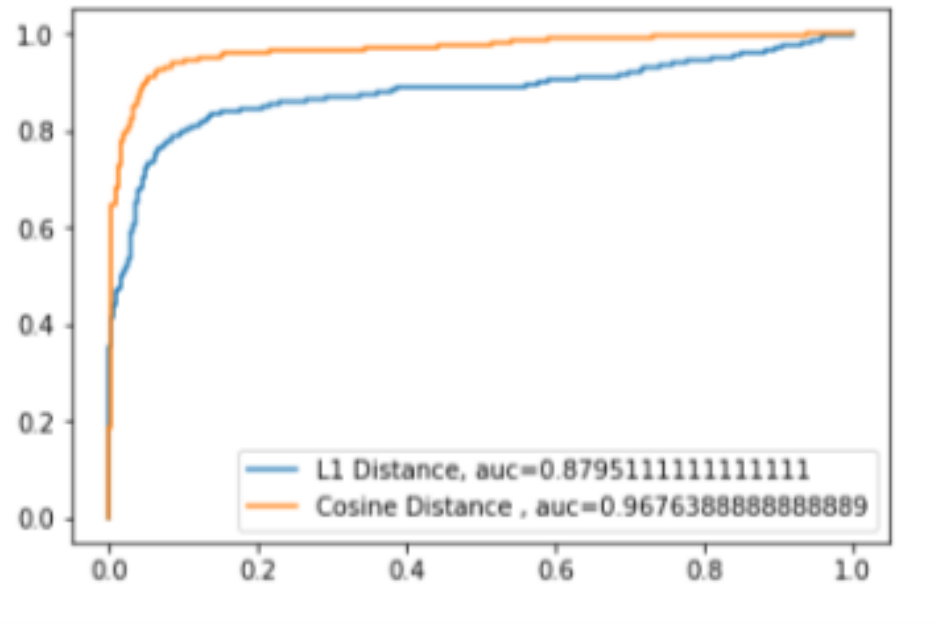

Figure 4-8: ROC curve for L1 Siamese and Cosine Siamese

Cosine distance appeared to be the best in first two baselines. As cosine distance can be interpreted as dot product of two vectors it is also called binary operation. We added more binary operations with cosine distance. Our aim was to detect a binary operation A(a! , a!) which can not only be applied to any two vectors of dimensions, a!=[a!!,…..,a!!] and a!=[a!^,…..,a!!], but is also served to do mapping between a non-empty set E and a function F, where F has output for all pair of elements in E individually and uniquely links with every pair of elements in E, E: $F \times F \rightarrow F$. (25) The feature representation based on binary operations doesn't need results from weak learners or any sub systems, whereas other methods including ensemble-based models use combination of similarity measures or discriminant scores from sub-systems. Binary operations-based feature representation acts as a package of information which a complex classifier can take and use what's helpful from that pack of information. In SpeakerNet three kinds of basic binary operations are used; a!⊕a!, a!⊗a! and a!⊖a! which are element-wise addition, multiplication and subtraction

respectively. These binary functions can also be used together to form vectors of higher dimensions. For example, the resultant vectors from addition and subtraction are concatenated. It is proved that the function that links vectors a! and a! to their binary vectors (in short b-vector) is injective (one to one) and surjective (onto). It is also worth mentioning that all of the above binary operations are commutative i.e they are independent of the order. A binary vector is generated by taking dot, sum, difference and square of difference is calculated instead of a single distance calculation. This approach improved results significantly. It has improved accuracy from 85% to 92.9%, Area under the curve by 1% and significantly improved EER by 1.2%. It is depicted in figure 4-9 below.

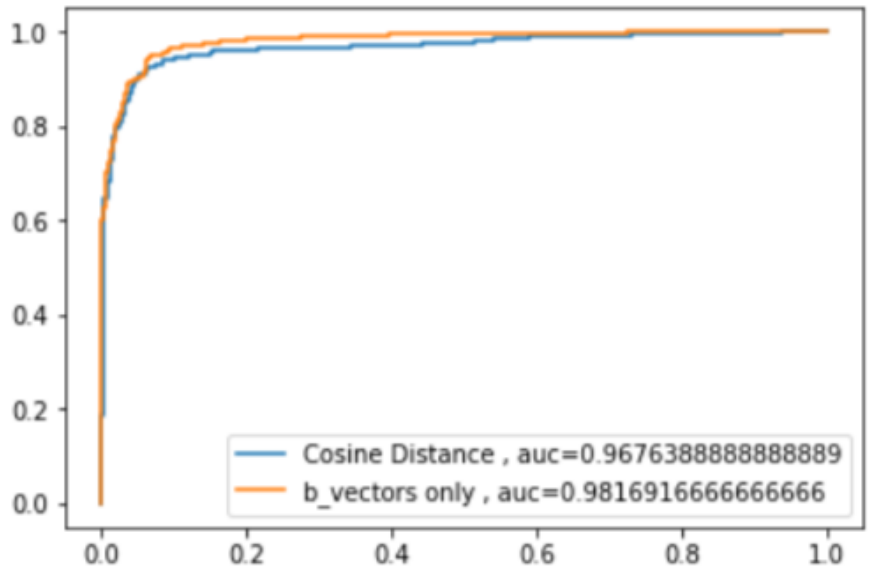

Figure 4-9 : ROC curve for Cosine Siamese and b-vectors

Above results proved the effectiveness of b-vectors. We further analyzed the effect of stacking them on each other forming a multi-dimensional shape. Convolution is applied on the stack of these binary vectors. Surprisingly it gave better results than using just concatenated b-vectors. One

reason for that is stacked b-vectors contain some patterns in them and convolutional layers tend to be the best to find image like patterns. ROC curve for just using concatenated binary vectors and proposed SpeakerNet is depicted in figure 4-10 below. Analysis of the score distribution for b-vectors and proposed SpeakerNet also shows properly divided scores (figure 4-5 and 4-6). After experimenting with thresholds, SpeakerNet showed a significant decrease in EER on 0.4 threshold in figure 4-7.

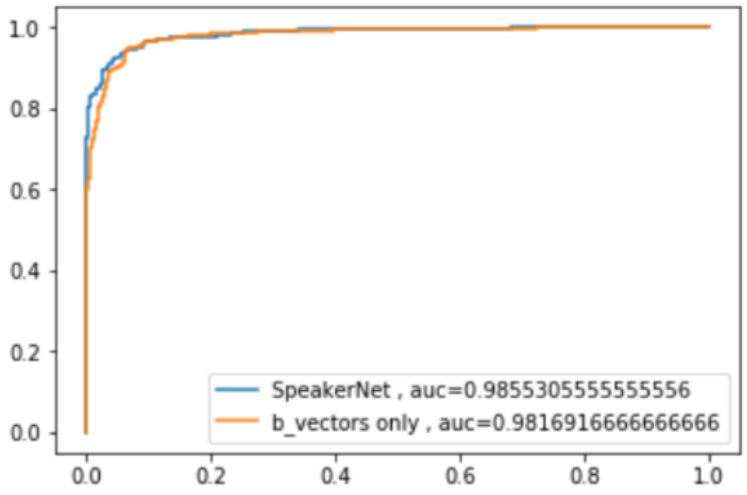

Figure 4-10: ROC for b-vectors and SpeakerNet

Finally, the ROC graph of all models to show overall results is presented in figure.

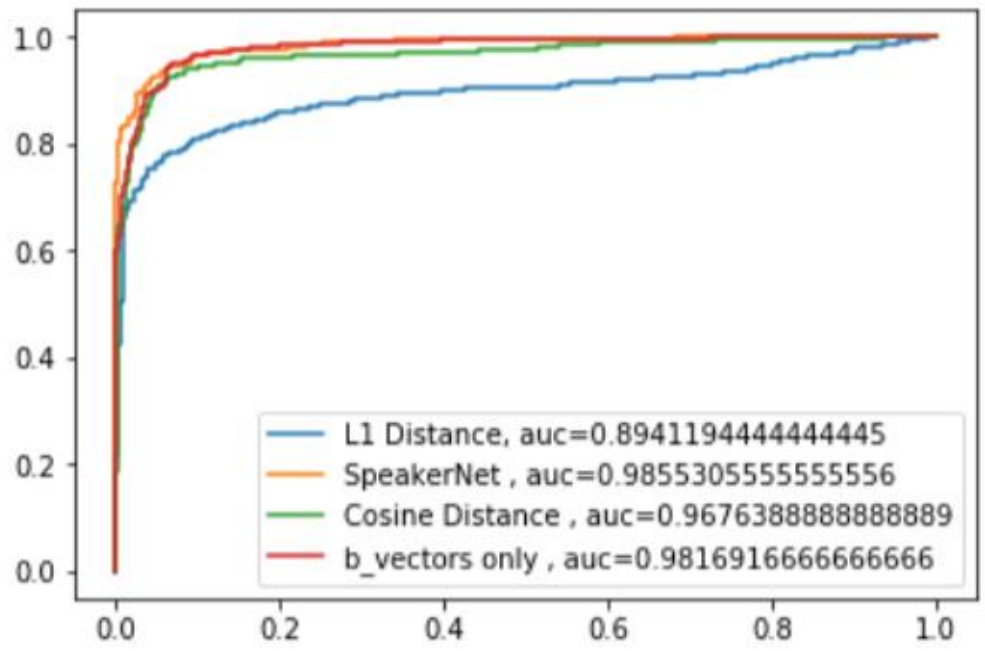

Figure 4-11: Combined ROC for baselines and SpeakerNet

4.5 SpeakerNet vs other methods

Most of the state-of-the-art methods use CNN for embedding extraction then apply scoring or similarity measures on them. While CNN embeddings provide better results than i-vector and d-vector approaches. The results outshine because of improper selection of similarity metrics. Cosine similarity is widely used in verification tasks and provide better results. But in text independent speaker verification scenario a single threshold does not work quite well. Keeping the speaker specific thresholds to solve this issue affects the automation of the process and demands more memory consumption. Our model is compared with the performance of other state of the art models. Table 4-3 shows their comparison in terms of Accuracy, Area under the curve and EER. It is evident that our model performed better than all of the methods in terms of accuracy, EER and AUC. Research has unveiled that the use of embedding networks in Siamese settings improves

verification as compared to use of a single threshold in previous practice .Embedding networks in Siamese work well for one-shot learning problems [48] but treating speaker verification problem as one-shot learning does not yield promising results due to lack of generic speaker model. Therefore, we have proposed a customized scoring scheme which utilizes Siamese' capability of applying distance measures with the convolutional learning. Another difference with above mentioned state of the art methods is that all the other systems were trained only on English or Chinese speakers but our system is multi lingual. Our system is trained on Urdu, Arabic and English utterances from the speakers of all ages.

Table 4-3 : Comparison of our method with sate of the art methods

Paper	**Method**	**Scoring / similarity measure**	**Train/test #utt**	**Input length (seconds)**	**Acc (%)**	**AUC (%)**	**EER**
[34]	ResNet with triplet loss	-	140,664/ 4,175	20	91.4	-	2.17
[34]	ResNet with LGM loss (alpha =1)	mahalanobis distance	140,664/ 4,175	20	90.26	-	2.37
[28]	3D CNN	cosine	-	20	-	87	22
[49]	LSTM	L1 distance					22
[26]	CNN	Euclidean Siamese	13k / 6k	-	-	-	20.5
[50]	Prosodic-Enhanced Siamese Convolutional Neural Network	Euclidean distance	2148/300	-	90	-	16
[25]	DNN	Bet Bernoulli	58k	-	-	-	9.34
[43]	i-vector SVM with binary operations kernel	Binary operation	170k	-	-	-	9.33
[21]	x-vector extracted from CNN	PLDA	-	-	-	-	8.23

Paper	**Method**	**Scoring / similarity measure**	**Train/test #utt**	**Input length (seconds)**	**Acc (%)**	**AUC (%)**	**EER**
[23]	ADDA embedding + PLDA adaptation	PLDA	12k / 7k	-	-	-	7.5
[20]	CNN-LSTM to extract embeddings, and binary vector based scroing	B-vectors passed to dense layers	19k / 600				7.4
[19]	SVM LDA and WCCN	Normalized cosine kernel					5.76
[51]	ResCNN, softmax (pre-train) + triplet	cosine	223k / 3k	3.6-4.5	91	-	3.14
[52]	d-vector + LDA	LDA	10k / 7k	30	-	-	3.02
[4]	Naïve Bayes	Likelihood ratio	1500	-	87	-	-
Baseline Model	CNN+ L1 Distance Siamese	L1 distance	30k / 7k	10	85	88	15.8
Baseline	Embedding CNN + cosine distance Siamese	Cosine distance	30k / 7k	10	85	97	7.3
Proposed Model	Embedding CNN + binary vectors	Binary operation	30k / 7k	10	92.9	98	6.3
Proposed Model	**CNN + SpeakerNet**	**proposed**	**30k / 7k**	**10**	**93.08**	**98.5**	**2.6**

Chapter 5

5 Conclusion

In this thesis, we have proposed a unique similarity learning method based on binary operations in Siamese like network for text-independent speaker verification. Which uses speech, language and age independent feature learning. This method does not rely on traditional feature engineering contrary its predecessors, alternatively it automatically learns the features form raw audio signals. Experiments has been made on cross language audios of multi-lingual speakers that highlights how well our proposed model detects the fraudulence of different speeches of different speakers with diverse background and scripts. Moreover, the proposed SpeakerNet has achieved improved results, by reducing EER to 2.6 % which is encouraging for further research in this direction.

The future work will focus on the development of a boosted network model trained on larger multi lingual dataset. Additionally, different modifications for verification task can also be experimented in future. More importantly, our proposed similarity learning scheme can be used in other verification problems.

6 References

[1] D. Chakrabarty, S. M. Prasanna, and R. K. Das, "Development and evaluation of online text-independent speaker verification system for remote person authentication," *International Journal of Speech Technology*, vol. 16, pp. 75-88, 2013.

[2] K. A. Lee, A. Larcher, H. Thai, B. Ma, and H. Li, "Joint application of speech and speaker recognition for automation and security in smart home," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[3] S. Dey, S. Barman, R. K. Bhukya, R. K. Das, B. C. Haris, S. R. M. Prasanna, *et al.*, "Speech biometric based attendance system," in *2014 twentieth national conference on communications (NCC)*, 2014, pp. 1-6.

[4] F. Thullier, B. Bouchard, and B.-A. Menelas, "A Text-Independent Speaker Authentication System for Mobile Devices," *Cryptography*, vol. 1, p. 16, 2017.

[5] L. A. Ramig and R. L. Ringel, "Effects of physiological aging on selected acoustic characteristics of voice," *Journal of Speech, Language, and Hearing Research*, vol. 26, pp. 22-30, 1983.

[6] S. Chachada and C. C. J. Kuo, "Environmental sound recognition: A survey," *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.

[7] J. Neyman and E. S. Pearson, "IX. On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 231, pp. 289-337, 1933.

[8] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," presented at the NIPS, 1994.

[9] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005, pp. 539-546.

[10] A. L. Higgins, L. G. Bahler, and J. E. Porter, "Voice identification using nearest-neighbor distance measure," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1993, pp. 375-378.

[11] J. Kacur, R. Vargic, and P. Mulinka, "Speaker identification by K-nearest neighbors: Application of PCA and LDA prior to KNN," in *18th International Conference on Systems, Signals and Image Processing*, 2011, pp. 1-4.

[12] H. Zeinali, H. Sameti, and B. Babaali, "A fast Speaker Identification method using nearest neighbor distance," in *IEEE 11th International Conference on Signal Processing*, 2012, pp. 2159-2162.

[13] A. Poritz, "Linear predictive hidden Markov models and the speech signal," in *ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1982, pp. 1291-1294.

[14] N. Z. Tisby, "On the application of mixture AR hidden Markov models to text independent speaker recognition," *IEEE Transactions on Signal Processing*, vol. 39, pp. 563-570, 1991.

[15] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, pp. 19-41, 2000.

[16] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *IEEE circuits and systems magazine*, vol. 11, pp. 23-61, 2011.

[17] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE signal processing letters,* vol. 13, pp. 308-311, 2006.

[18] Z. Lei, J. Luo, and Y. Yang, "A Simple Way to Extract I-vector from Normalized Statastics," presented at the Chinese Conference on Biometric Recognition, Cham, 2014.

[19] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Tenth Annual conference of the international speech communication association*, 2009.

[20] J.-W. Jung, H.-S. Heo, I.-H. Yang, H.-J. Shim, and H.-J. Yu, "Avoiding speaker overfitting in end-to-end dnns using raw waveform for text-independent speaker verification," presented at the extraction, 2018.

[21] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *IEEE Spoken Language Technology Workshop (SLT)*, 2016 pp. 165-170.

[22] J. Alam, G. Bhattacharya, and P. Kenny, "Speaker verification in mismatched conditions with frustratingly easy domain adaptation," in *Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 176-180.

[23] W. Xia, J. Huang, and J. H. L. Hansen, "Cross-lingual Text-independent Speaker Verification Using Unsupervised Adversarial Discriminative Domain Adaptation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5816-5820.

[24] X. Chen, Y. Sun, B. Athiwaratkun, C. Cardie, and K. Weinberger, "Adversarial deep averaging networks for cross-lingual sentiment classification," *Transactions of the Association for Computational Linguistics,* vol. 6, pp. 557-570, 2018.

[25] M. J. Alam, P. Kenny, G. Bhattacharya, and M. Kockmann, "Speaker Verification Under Adverse Conditions Using i-Vector Adaptation and Neural Networks," in *INTERSPEECH*, 2017, pp. 3732-3736.

[26] H. Salehghaffari, "Speaker Verification using Convolutional Neural Networks," in *arXiv preprint arXiv:1803.05427*, ed, 2018.

[27] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *arXiv preprint arXiv:1405.3531*, ed, 2014.

[28] A. Torfi, J. Dawson, and N. M. Nasrabadi, "Text-independent speaker verification using 3d convolutional neural networks," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2018, pp. 1-6.

[29] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, 2015.

[30] Y. Qi, Y.-Z. Song, H. Zhang, and J. Liu, "Sketch-based image retrieval via siamese convolutional neural network," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2460-2464.

[31] X. Sun, A. Torfi, and N. Nasrabadi, "Deep siamese convolutional neural networks for identical twins and look-alike identification," *Deep Learning in Biometrics*, p. 65, 2018.

[32] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *European conference on computer vision*, 2016, pp. 791-808.

[33] A. Torfi and R. A. Shirvani, "Attention-based guided structured sparsity of deep neural networks," in *arXiv preprint arXiv:1802.09902*, ed, 2018.

[34] X. Shi, M. Zhu, and X. Du, "End-to-End Residual CNN with L-GM Loss Speaker Verification System," in *arXiv preprint arXiv:1805.00645*, ed, 2018.

[35] L. Zhang, J. Yang, and D. Zhang, "Domain class consistency based transfer learning for image classification across domains," *Information Sciences*, vol. 418, pp. 242-257, 2017/12/01/ 2017.

[36] J. S. García-Salinas, L. Villaseñor-Pineda, C. A. Reyes-García, and A. A. Torres-García, "Transfer learning in imagined speech EEG-based BCIs," *Biomedical Signal Processing and Control*, vol. 50, pp. 151-157, 2019/04/01/ 2019.

[37] Z. Huang, S. M. Siniscalchi, and C.-H. Lee, "A unified approach to transfer learning of deep neural networks with applications to speaker adaptation in automatic speech recognition," *Neurocomputing*, vol. 218, pp. 448-459, 2016/12/19/ 2016.

[38] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," presented at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.

[39] Q. Hong, L. Li, J. Zhang, L. Wan, and H. Guo, "Transfer learning for PLDA-based speaker verification," *Speech Communication*, vol. 92, pp. 90-99, 2017/09/01 2017.

[40] H. Hermessi, O. Mourali, and E. Zagrouba, "Deep feature learning for soft tissue sarcoma classification in MR images via transfer learning," *Expert Systems with Applications,* vol. 120, pp. 116-127, 2019/04/15/ 2019.

[41] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A Practical Transfer Learning Algorithm for Face Verification," in *IEEE International Conference on Computer Vision*, 2013, pp. 3208-3215.

[42] C. Zhang, S. Ranjan, and J. Hansen, "An Analysis of Transfer Learning for Domain Mismatched Text-independent Speaker Verification," in *Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 181-186.

[43] H.-S. Lee, Y. Tso, Y.-F. Chang, H.-M. Wang, and S.-K. Jeng, "Speaker verification using kernel-based binary classifiers with binary operation derived features," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1660-1664.

[44] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *arXiv preprint arXiv:1706.08612*, ed, 2017.

[45] J. Ramirez, J. M. Górriz, and J. C. Segura, "Voice activity detection. fundamentals and speech recognition system robustness," in *Robust speech recognition and understanding*, ed: IntechOpen, 2007.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *arXiv preprint arXiv:1409.1556*, ed, 2014.

[47] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, *et al.*, "CNN architectures for large-scale audio classification," in *ICASSP*, 2017, pp. 131-135.

[48] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," in *Advances in neural information processing systems*, 2016, pp. 3630-3638.

[49] A. Mobiny, "Text-Independent Speaker Verification Using Long Short-Term Memory Networks," in *arXiv preprint arXiv:1805.00604*, ed, 2018.

[50] S. Soleymani, A. Dabouei, S. M. Iranmanesh, H. Kazemi, J. Dawson, and N. M. Nasrabadi, "Prosodic-Enhanced Siamese Convolutional Neural Networks for Cross-Device Text-Independent Speaker Verification," in *9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2018, pp. 1-7.

[51] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, *et al.*, "Deep speaker: an end-to-end neural speaker embedding system," in *arXiv preprint arXiv:1705.02304*, ed, 2017.

[52] D. Wang, L. Li, Z. Tang, and T. F. Zheng, "Deep speaker verification: Do we need end to end?," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 177-181.