SPEAKER RECOGNITION SYSTEM

Kulsoom Shahbaz
Dept. Of Computer Science
University Of Regina
kss128@uregina.ca

Kanisha Agarwal
Dept. Of Computer Science
University Of Regina
kag437@uregina.ca

Baljinder Sohi
Dept. Of Computer Science
University Of Regina
bkq705@uregina.ca

Simranjeet Randhawa
Dept. Of Computer Science
University Of Regina
ssr779@uregina.ca

Abstract. **Speaker recognition is an approach which is used to identify the characteristics of human voices or commonly known as voice biometrics. The main aim of this work is to implement and analyze the mechanism behind an efficient Speaker Recognition System. Speakers can be recognized by performing feature matching using vector quantization on the extracted features via MFCC. Speaker Recognition is carried out in both noisy and noise-less environment and the results are compared qualitatively and quantitatively. The outcomes prove that the system works effectively in a noise-free environment.**

Keywords - Speaker Recognition, Feature Extraction, Feature Matching, Training, Testing, MFCC, Vector Quantization, and LBG.

I. INTRODUCTION

Speaker recognition means who is the speaker while speech recognition means what is being said. The speaker recognition process based on a speech signal is treated as one of the most exciting technologies of human recognition. Speaker recognition can simplify the task of translating speech in systems that have been trained on specific voices or it can be used to authenticate or verify the identity of a speaker as part of a security process. Basically, voice recognition is a combination of Speaker recognition (a person who is speaking) and speech recognition (what's being said by the person). There is a difference between speaker verification (speaker authentication) and speaker identification. Also, speaker diarisation (identifying the voice of the same speaker) varies from speaker recognition. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on specific voices or it can be used to authenticate or verify the identity of a speaker as part of a security process.[1] The term Speaker recognition evolved approximately four decades ago in which the characteristics of speech acoustic is used to recognize the voices of different individuals.

Speaker Recognition holds a significant role in Speech Signal Processing with numerous applications; for example, security systems, telecommunication system, noise-reduction techniques, and voice-controlled devices. Microphones and technology including voice transmission over long distances using wired telephones or wireless telephones are used in the process of Speaker recognition. For accuracy, digitally recorded voice identification and analog recorded voice identification uses electronic measurements along with which critical listening skills are required. Being a well-known topic, many research projects have already been accomplished by famous researchers. In this work, our goal is to implement the speaker recognition algorithm using Python which was earlier carried out by renowned researchers.

Speaker recognition system is categorized into two parts:

1) *Text-dependent*
2) *Text-independent.*

Text-Dependent:
In this kind of system, the similarity in the text is compulsory for verification and enrollment. In addition, prompts used in the text-dependent system must be identical across all the speakers. Also, shared- secrets (PINs and Passwords) or knowledge-based information can be included for multi-factor authentication scenario.

Text-Independent:
Generally, the text-independent system is used for speaker identification because the requirement of speaker cooperation is comparatively very little in this case. Text can be different at the time of enrollment and test. In fact, user knowledge may not require during enrollment, in many forensic applications**.** As text-independent technologies do not compare what was said at enrollment and verification, verification applications tend to also employ speech recognition to determine what the user is saying at the point of authentication. In the text, independent systems both acoustics and speech analysis techniques are used.

Speaker recognition is been implemented using 2 main algorithms: MFCC (Mel-Frequency Cepstral Coefficients) and LBG (Linde, Buzo, and Gray). MFCC is used for feature extraction from voice and LBG is a standard vector quantization technique used for clustering of large datasets.

II. METHOD IMPLEMENTATION

The basic principle of speaker recognition is based on the extraction of features from the voice of the speakers. The model is trained after extracting the features and then testing is carried out in which feature matching

takes place. Fig 2 depicts the block diagram for the speaker recognition. The voice of *n* different speakers are taken in which the Mel-Frequency Cepstral Coefficients (MFCC) are extracted from each speaker and model is trained and tested with Vector Quantization technique (using the LBG algorithm) for feature matching.

Block Diagram:

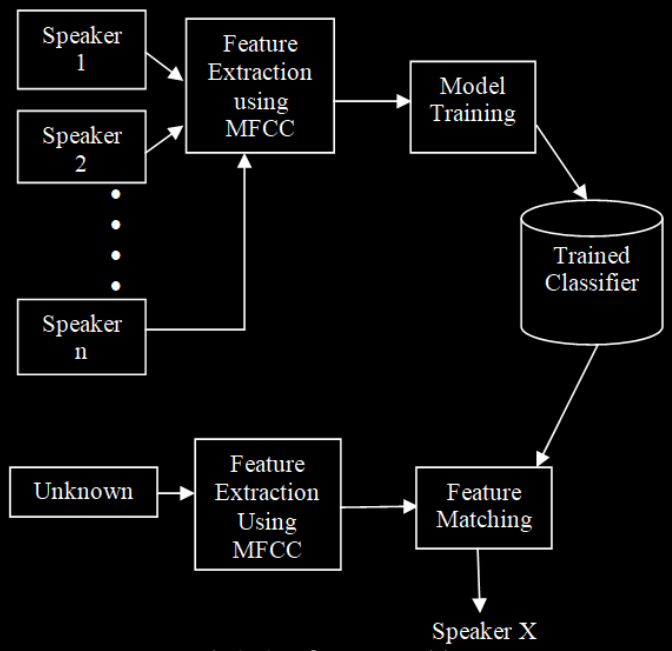

Fig 2: Speaker Recognition

In our work, initially voices of four different speakers (Kulsoom, Kanisha, Baljinder, Simranjeet) were recorded and features of those voices were extracted out using MFCC. This was followed by training which includes training the model and building a classifier on the basis of features extracted. The last step in this process is testing, in which the extracted feature of the unknown speaker is matched with trained data using classification algorithm LBG. The Python code is uploaded on GitHub.

A. Feature Extraction using MFCC:

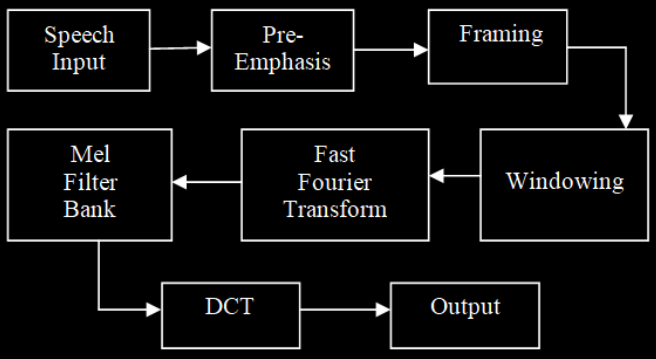

Fig 2.1: MFCC Block Diagram

Mel Frequency Cepstral Coefficients (MFCC) is used for extracting features for 4 different speakers i.e. (Kulsoom, Kanisha, Baljinder, Simranjeet). It is an effective way of extracting features because it is robust and quite effective in a noisy environment.
The block diagram showing the Feature extraction process is explained as below:

1. Pre-Emphasis:
The words spoken by a speaker at higher frequencies are very crucial in the process of speaker recognition. The pre-emphasis stage ensures that the high frequencies components in speakers' tone are identified and smooths the signal in order to eliminate the noises involved in the signal

2. Frame blocking:
The signal obtained after the pre-emphasis stage is divided into frames of small duration. Framing helps in short time spectral analysis.
The frame duration is chosen depending on how fast the frequencies in the signal are changing.

3. Hamming Windowing:
The continuity of the signal is maintained by introducing the concept of a Hamming window. After frame blocking is performed, the frames created are multiplied by the hamming window to maintain the continuity in the signal.
The main aim is to remove the discontinuity when the signal is divided into blocks. The discontinuity in the signal will produce spectral distortion. Hence to minimize spectral distortion and ensuring continuity hamming window is used.

4. Fast Fourier Transform:
Fast Fourier Transform is carried out on the result obtained after hamming window in order to get the frequency domain of the signal. The FFT is more efficient as compared to discrete Fourier transform. It takes less time to retrieve the frequency components in the signal.

5. Mel filter band:
Mel filter band is used to get smooth magnitude spectrum. The Mel filter band consist a series of filters which are used to identify the features in the voice. We have taken 16 triangular bandpass filter for extracting different features.

6. The discrete cosine transforms:
DCT is performed on the signal received after Mel filter band. DCT is different from DFT as it deals only with the real part. DCT was performed because it gives a real part and extracting features with real values helps to

make the comparison easy. Therefore, DCT was used at the last stage to get the output signal.

B. Feature Matching using LBG (Vector Quantization):

LBG a Vector Quantization technique is used for the purpose of feature matching. Vector quantization is a process to form clusters of similar data and thereby helping in feature matching. A cluster represents a region of similar items with identical features. The centre of the cluster is called codeword. The collection of codeword represents a codebook. Each speaker has their own codebook. For example, Kulsoom, Kanisha, Baljinder and Simranjeet have their own codebook. No two speakers have the same codebook. The codebook represents the uniqueness in the voice of any speaker.

The LBG algorithm [Linde, Buzo, and Gray], is used for clustering set codewords into a set of desired codebook. Flowchart for the LBG algorithm is depicted in the figure below.

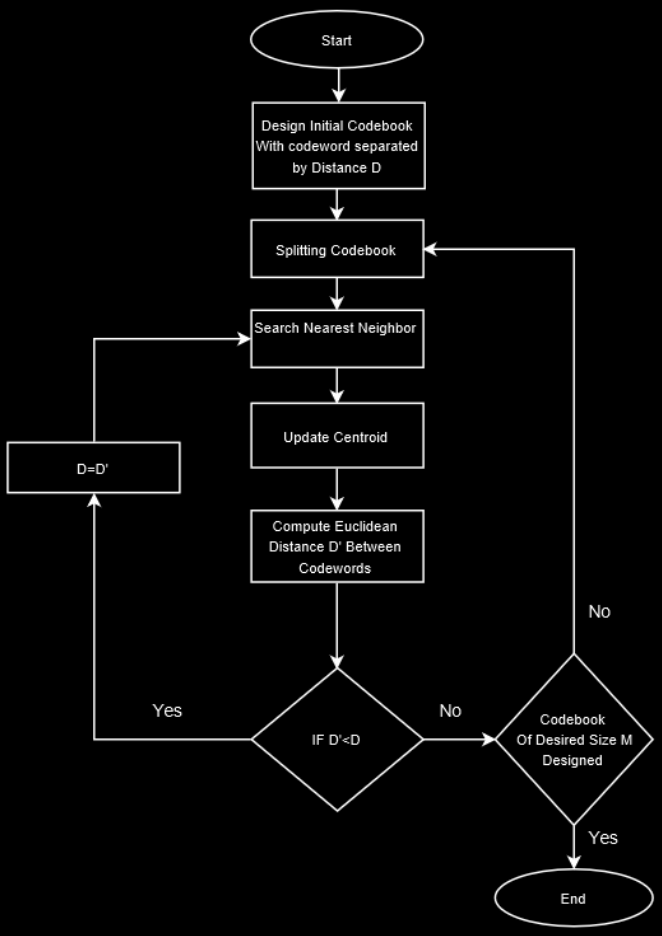

Fig 2.2: LBG Flowchart

The algorithm is implemented by the following recursive procedure:

1. The codebook is designed and initially considered as the centroid of the entire set of training data.

2. Size of the codebook is changed by splitting the codebook.

3. Nearest-Neighbor Search is performed for each set of training data. The codebook is associated with the closest codeword by using Euclidean Distance. The Euclidean distance is calculated from the codebook to the centroids and the least distance is taken into account and then the codebook is linked with that codebook. The Nearest-Neighbor Search helps to collect data and form clusters of the similar data type.

4. Centroid Update: After Nearest-Neighbor search, the centroid is updated and then step 5 is performed.

5. Repeat steps 3 and 4 until no new neighbours get added together in the cluster. If no changes are taking place and centroid is not getting updated anymore then step 6 is performed.

6. Repeat steps 2, 3 and 4 until a codebook of desired size is designed.

The LBG algorithm starts by taking an initial codebook and then performs splitting until a codebook of the desired size is obtained. The splitting is performed based on the similarity between codewords and the final codebook obtained consist of similar codewords. This is the main principle followed by the LBG algorithm.

III. TESTING AND TRAINING

A. Training datasets

Training datasets is an essential step in speaker recognition. The voices of different speakers are trained in order to retrieve essential features from each voice. For the purpose of training voices of 4 different speakers are taken speaking for a very short duration. Every voice has distinct features and training helps to extract those distinct features present in the voice of the speaker. After performing training, codebooks for all the speakers are generated. It is assured that voice trials are pronounced separately by the preceding trials that helps to make it more lenient to the variations in a person's voice that occur over a short time duration.

Training the model produces a classifier which helps in classification of the voices. Training helps to produce codebooks for different speakers. Every speaker has his or her own codebook.

B. Testing results

Testing is the last step to test that the speaker is recognized successfully or not. For the purpose of testing a speaker from the trained speakers is taken and feature matching is performed on the voice of the speaker. This helps us to recognize who is speaking. For example, Speaker 1 (Kulsoom) voice was trained in the training set. In training, the features of Kulsoom's voice was extracted. Now in the testing phase, Kulsoom's voice features are matched with the extracted features. If the features matches then the speaker has being identified as Kulsoom. The same procedure is followed for other speakers also.

In testing the unknown speaker is taken and by performing feature matching the speaker can be identified. Hence speaker recognition is useful in voice recognition and can also be used in various authentication purposes. The testing has being carried out for 4 different speakers in our research i.e. (Kulsoom, Kanisha, Baljinder, Simranjeet). Testing helps us to recognize or identify the unknown speaker from a lot of trained speakers.

IV. RESULT ANALYSIS

The speakers are trained and the following codebooks are generated for speakers using MFCC algorithm for feature extraction. The number of features is obtained when the voice is passed through different filters.

Plots Obtained Without Noise:

The experiments are conducted in a noise-free environment and the unique codebooks are generated for various speakers as shown below.

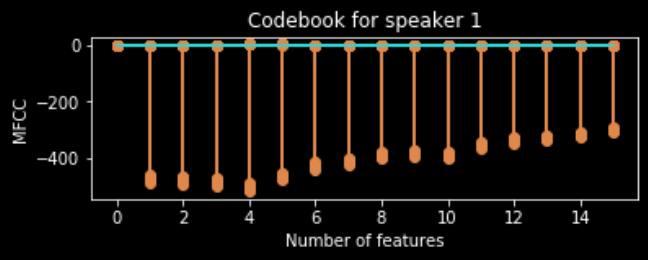

Fig 4.1: Codebook Of Kulsoom Without Noise

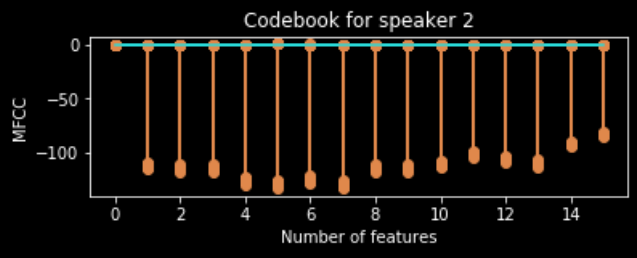

Fig 4.2: Codebook Of Kanisha Without Noise

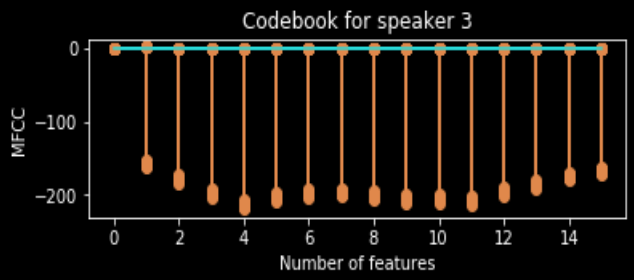

Fig 4.3: Codebook Of Baljinder Without Noise

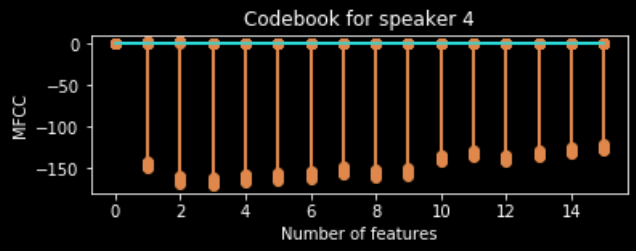

Fig 4.4: Codebook Of Simranjeet Without Noise

Plots Obtained With Noise:

The following observations are obtained in a noisy environment and again the unique codebooks are generated for various speakers as shown below.

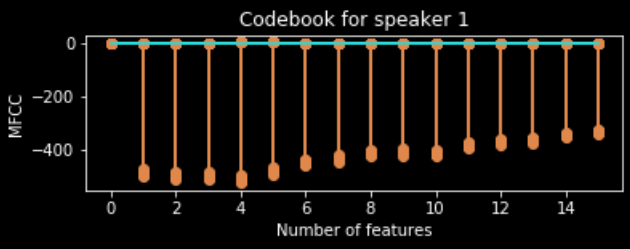

Fig 4.5: Codebook Of Kulsoom With Noise

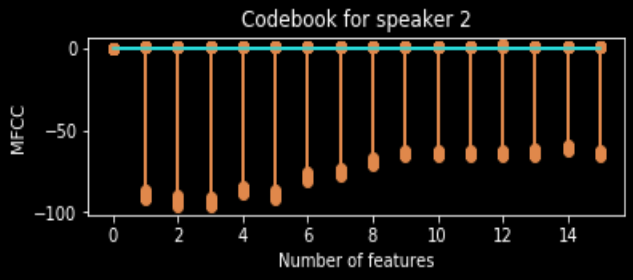

Fig 4.6: Codebook Of Kanisha With Noise

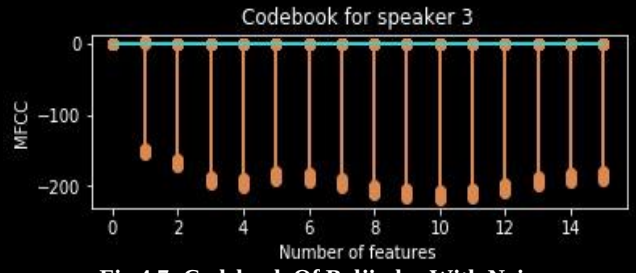

Fig 4.7: Codebook Of Baljinder With Noise

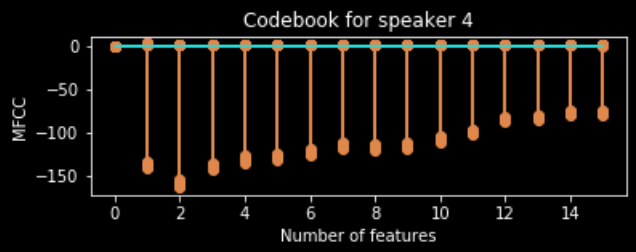

Fig 4.8: Codebook Of Simranjeet With Noise

A. Quantitative Analysis

The following table gives the identification results for each of the 4 speakers with MFCC and Vector Quantization with LBG algorithm for classification.

Serial number	True speakers	Recognized as:
1	S1(Kulsoom)	S1
2	S2(Kanisha)	S2
3	S3(Baljinder)	S3
4	S4(Simranjeet)	S4
		Accuracy = 100%

Table 1: Recognizing True Speakers (Without Noise)

Percentage (%) error involved in identifying speakers with no background noise is calculated as (Δ t / t) * 100,
Where Δt represents the difference between true speakers and identified speakers and t is the true speaker.
Total number of speakers considered = 4
Number of speakers correctly identified = 4
Number of speakers incorrectly identified = 4-4 = 0
% error involved in identifying speakers correctly
= ((4-4)/4)*100
= 0%

Serial number	True speakers	Recognized as:
1	S1(Kulsoom)	S1
2	S2(Kanisha)	S3
3	S3(Baljinder)	S3
4	S4(Simranjeet)	S4
		Accuracy = 75%

Table 2: Recognizing True Speakers (With Noise)

Percentage (%) error involved in identifying speakers with background noise is calculated as (Δ t / t) * 100,
Where Δt represents the difference between true speakers and identified speakers and t is the true speaker.
Total number of speakers considered = 4
Number of speakers correctly identified = 3
Number of speakers incorrectly identified = 4-3 = 1
% error involved in identifying speakers correctly
= ((4-3)/4)*100
= 25%

Thus, it can be inferred that the percentage error between the outcomes obtained in the noisy background and the non-noisy background is (25-0) = 25 %.

B. Qualitative Analysis

The qualitative comparison is carried out by overlaying the codebooks of different speakers obtained in a noisy and noiseless environment.

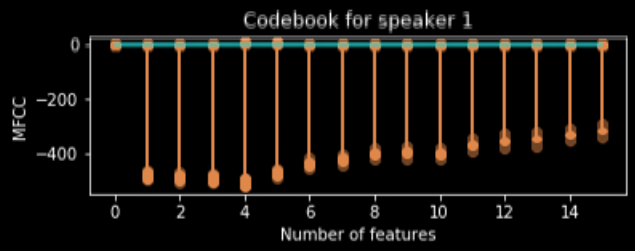

Fig 4.9: Overlaid Codebook Of Kulsoom

Fig 4.9 is obtained by overlaying Fig 4.1 and Fig 4.5 The results shows us that the introduction of noise does not alter the features in Kulsoom's voice. It is because the noise was introduced for a small-time frame for speaker 1.

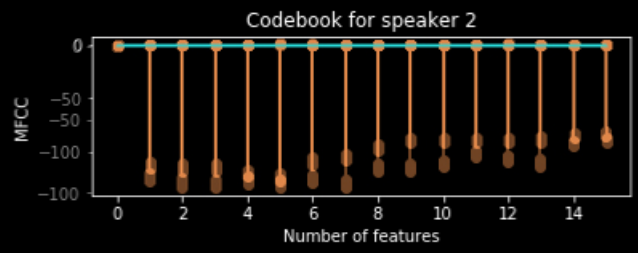

Fig 4.10: Overlaid Codebook Of Kanisha

Fig 4.10 is obtained by overlaying Fig 4.2 and Fig 4.6. For speaker 2 the noise was introduced for long time duration. Hence the overlaid figure shows that there is much variation involved. This signifies the fact that as background noise increases the extracted features from voice will be difficult to match.

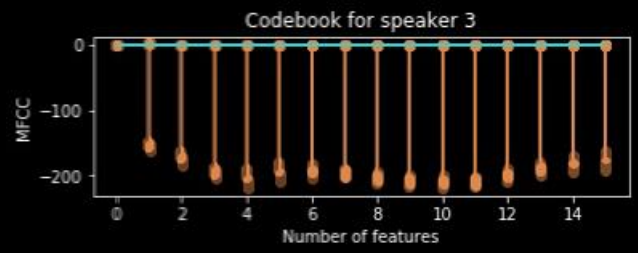

Fig 4.11: Overlaid Codebook Of Baljinder

Fig 4.11 is obtained by overlaying Fig 4.3 and Fig 4.7. For speaker 3 the noise was introduced for the short time duration. Hence there is less variation involved in the overlaid figure.

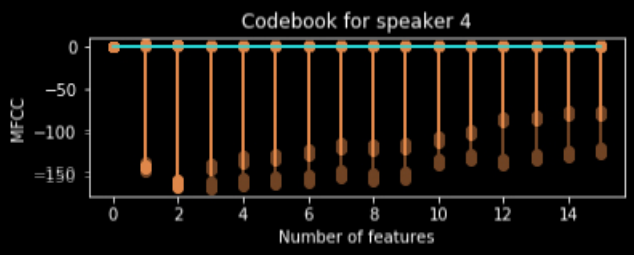

Fig 4.12: Overlaid Codebook Of Simranjeet

Fig 4.11 is obtained by overlaying Fig 4.4 and Fig 4.8. It shows much variation and does not match accurately because the noise was introduced for a long-time duration for Speaker 3. This clearly concludes that the accuracy of the system to recognize speakers will reduce when noise is added to the background.

V. WRAPPING UP

The speaker recognition determines the best features that are employed towards the generation of signal-dependent and text-independent method. Features are selected in such a way that, they can by itself or with a combination of other feature sets have the ability to distinguish the speakers reliably. Initially, individual features are considered and then a combination of these features with different classifiers is obtained. Thus, the experimental results reveal that the classification model helps in the selection of ideal and most reliable characteristics from the features that are extracted using MFCC for speaker recognition towards the identification of one's Codebook.

VI. CONCLUSION

Speaker Recognition is a process of recognizing persons from the recordings of their speech. Speaker recognition being performed is noisy as well as non-noisy environment gives percentage error of 25% and 0% respectively. It can be clearly inferred that speech recognition system works accurately in a noise-free environment whereas the accuracy decreases to some extent when the disturbance in the background increases.

It can be concluded that to create an efficient speaker recognition system, the unwanted disturbances in the form of noise must be eliminated completely. Noise reduction algorithms can be performed to improve accuracy, but incorrect use of noise reduction algorithm can have an adverse effect.

Speaker recognition systems have its applications in our daily lives and have huge benefits for people suffering from any kind of disabilities. However, changes in voice due to aging may affect system performance over time.

ACKNOWLEDGMENT

We would like to thank Professor Trevor Tomesh of computer audio for his help and support in making this work possible. Professor Trevor taught us about different signals, waveforms, filters, hardware and digital audio from which we implemented Speaker Recognition. He also assisted us in class and during his office hours. The knowledge and prior work of python guru Darko lukic were instrumental in developing the code.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films, and exchange anisotropy," in Magnetism, vol. III, G. T. Rado, and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in the press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

.