Singer voice Recognition using MFCC,GMM and Neural network

S. Abinaya, S. Aruna, S. Dharmasamvarthini, R. Janani
final year UG students, ECE Department
Guided by: Mrs. Kalaivani,
Assistant professor, Department of ECE,
K. Ramakrishnan College of Engineering, Samayapuram

Abstract—**In order to meliorate the performance of the singer identification we put forward a system to separate singing voice from music. Our system consists of two key stages. In the first stage voice of the singers are trained, followed by the testing stage which compares the trained voice with the input voice. In this recognition we propose a novel deep neural network(DNN) bottleneck(BN) for learning speaker specific characteristics from Mel-Frequency Cepstral Coefficients(MFCC), an acoustic representation commonly used in voice recognition .To surmise the fidelity of frame Gaussian mixture models(GMM) for vocal and non-vocal frames to select the reliable vocal portion of musical pieces. Our experiments show that background removal approach improves the voice recognition accuracy significantly**.

***Index Terms*—Deep Neural Network(DNN), Gaussian Miture Model(GMM), BottleNeck(BN), singer identification, Mel-Freuency Cepstral Coefficient(MFCC).**

I. Introduction

Vocal information is important to people for seeking and retrieving music because the singing voice habitually draws more of listener's attention than other music information. As the singing voice is important the representation of its characteristics is useful for music information retrieval (MIR) .several studies in the field of singer identification pay attention to feature extraction directly from the songs. In general singer identification is more tedious than speaker identification by the fact that a singer's vocal characteristics in music are not largely modulated by metrics and melody but also mesh with background accompaniments. In order to identify the popular singers, their voice can be characterized by the concept of speech utterances. This problem was toilsome to solve because most singing voices are accompanied by the other musical instruments and feature vectors extracted from musical audio signals are esteemed by the sounds of accompanying instruments. it is therefore necessary to focus on the vocals in polyphonic sound mixtures. Speaker recognition is the identification of a person from characteristics of referred as voice biometrics. There is a difference between speaker recognition and speaker identification . These two terms are frequently confused, and "voice recognition" can be suits for both. In addition, there is a difference between the act of authentication (commonly referred to as speaker verification or speaker authentication) and identification. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on specific person's voices or it can be used to authenticate or verify the identity of a speaker as part of a security process.

II. SINGING VOICE REFINEMENT

A. Refinement of voice

An accurate voice identification from polyphonic music has long been a provocation, the inevitable errors in the obtained voice can procreate to the subsequent processes which may affect the performance significantly. A human can easily identify voice of the singer but it is still a difficult task for a computer to automatically recognize them. This is mainly because music in the real world is polyphonic.

Intensity and dynamic range of singing voice is greater than the speech. The pitch of normal dialect ranges from 80-400hz,while that of singing can be from 80-1000hz.High recognition accuracy and low requirement of the amount of training data are the desired but contrary goal for the design of recognition system the strategy can also used in identification to alleviate the need of acquiring singing voice from each target singer, it may still require tremendous effort to build such a system, because large amount of a cappella singing voices are not readily available as speech, especially for professional singers voices.

B. Phases of Singing Voice

Following speaker identification framework, we can build an singer voice identification system. The system operates in two phase: Training and Testing. During the training phase, a group of persons is represented by Gaussian mixture model. It is known that GMM provide good approximation of arbitrarily spaced densities of spectrum over a long period of time prior to Gaussian modeling speech waveforms are converted into frames by Mel-Frequency Cepstral Coefficients (MFCC)

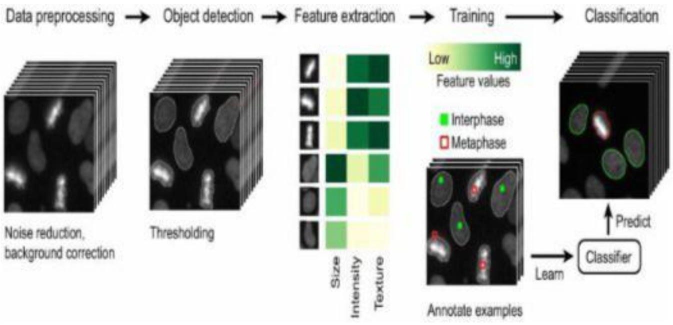
C. Process of Existing Method

MFCC carries less information on pitch than vocal tract configuration they should be able to absorb the discrepancy between singing and speech in the pitch variations. In the testing phase an unknown singing clip is converted into MFCC and then tested, then the system calculates and decides in favor of singer voice when the maximum likelihood condition is satisfied Mel-frequency cepstral coefficients, have been shown to largely satisfy the requirements, and dominate most speech-related work in existing literature. The steps involved in the extraction of MFCCs are as follows: the input speech signal first undergoes pre-emphasis of high frequency

Thus the values of all such frequency are obtained and the results are found out with the help of many feature extraction methods there are many feature extraction methods available which are as follows,

1. Mel-Frequency Cepstral Coefficients (MFCC)

2. Predictive Linear Prediction (PLP)



(a)

Fig 1: Process of existing method

Pre-emphasis:

The speech signal s(n) is sent to a high-pass filter:
s2(n) = s(n) - a*s(n-1)
where s2(n) is the output signal and the value of a is usually between 0.9 and 1.0.

Frame blocking:

The input speech signal is segmented into frames of 20~30 ms with optional overlap of 1/3~1/2 of the frame size. If the sample rate is 16 kHz and the frame size is 320 sample points, then the frame duration is 320/16000 = 0.02 sec = 20 ms. Additional, if the overlap is 160 points, then the frame rate is 16000/(320-160) = 100 frames per second.

Hamming windowing:

Each frame has to be multiplied with a hamming window in order to keep the continuity of the first and the last points in the frame (to be detailed in the next step). If the signal in a frame is denoted by s(n), n = 0,…N-1, then the signal after Hamming windowing is s(n)*w(n), where w(n) is the Hamming window defined by: **W (n, □).** Different values of corresponds to different curves for the Hamming windows shown next. In practice, the value of is set to 0.46. MATLAB also provides the command hamming for generating the curve of a Hamming window.

Fast Fourier Transform:

Spectral analysis shows that different timbers in speech signals corresponds to different energy distribution over frequencies. Therefore we usually perform FFT to obtain the magnitude frequency response of each frame. When we perform FFT on a frame, we assume that the signal within a frame is periodic, and continuous when wrapping around. If this is not the case, we can still perform FFT but the in continuity at the frame's first and last points is likely to introduce undesirable effects in the frequency response

III. PROCESS OF PROPOSED METHOD A. Preprocessing

MFCC feature extraction is done in the proposed method. In the training phase voice of 20 singers are taken and preprocessing is done for obtaining the value of feature vectors. MFCCs are as follows: the input speech signal first undergoes pre-emphasis of high frequency MFCC carry less information on pitch than vocal tract configuration they should be able to absorb the discrepancy between singing and speech in the pitch variations. In the testing phase an unknown singing clip is converted into MFCC and then tested, then the system calculates and decides in favors of singer voice when the maximum likelihood condition is satisfied

Preprocessing involves the above three steps of existing method such as

1. Pre-Emphasis

2. Framing

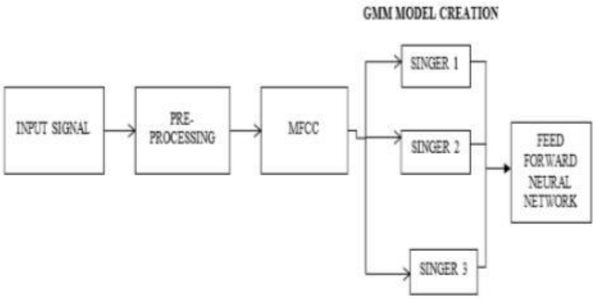
3. Windowing

B. Training phase

In the training phase the voice of the singers are trained and models are created by GMM. The training phase consists of preprocessing, feature extraction and model creation .In the feature extraction feature vectors are created.

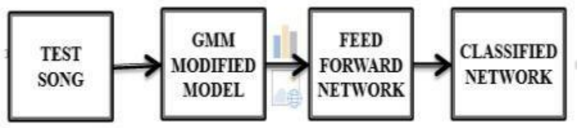
The block diagram of training phase is shown below,

Thus the singers voice are trained using neural network and features are obtained with the help of MFCC feature extraction. the output of the training phase are given as feature vectors in the form of matrix. For example if we take the matrix example of 5 input values with their corresponding matrix representation and its coefficients. GMM models are also created in the training phase of singer recognition.



(a)

In the testing phase test song is given as a input and checks whether it is in the database of the network. In this paper neural network is used for classification of voice signals



(b)

$$\begin{bmatrix} w_{11} & w_{21} & w_{31} & w_{41} & w_{51} & w_{61} & w_{71} & w_{81} & w_{91} \\ w_{12} & w_{22} & w_{32} & w_{42} & w_{52} & w_{62} & w_{72} & w_{82} & w_{92} \\ w_{13} & w_{23} & w_{33} & w_{43} & w_{53} & w_{63} & w_{73} & w_{83} & w_{93} \\ w_{14} & w_{24} & w_{34} & w_{44} & w_{54} & w_{64} & w_{74} & w_{84} & w_{94} \\ w_{15} & w_{25} & w_{35} & w_{45} & w_{55} & w_{65} & w_{75} & w_{85} & w_{95} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \\ i_8 \\ i_9 \end{bmatrix} = \begin{bmatrix} \textit{total input to } H_1 \\ \textit{total input to } H_2 \\ \textit{total input to } H_3 \\ \textit{total input to } H_4 \\ \textit{total input to } H_5 \end{bmatrix}$$

(c)

Fig 2: Block diagram of training phase (a) Block diagram of testing phase (b) Example of matrix representation of MFCC vector coefficients

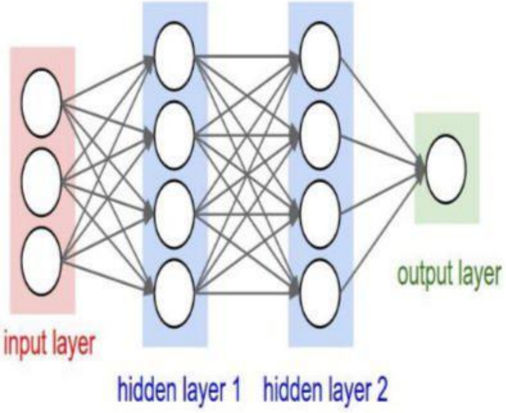
IV. NEURAL NETWORK IN VOICE RECOGNITION

In this Neural networks are the simplified models of the biological neuron systems. Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' .which contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output. An artificial Neural Network is defined as a data processing system consisting of a large number of interconnected processing elements or artificial neurons. There are three fundamentally different classes of neural networks. Those are.

1. Single layer feedforward Networks.

2. Multilayer feedforward Networks.

3. Recurrent Networks.

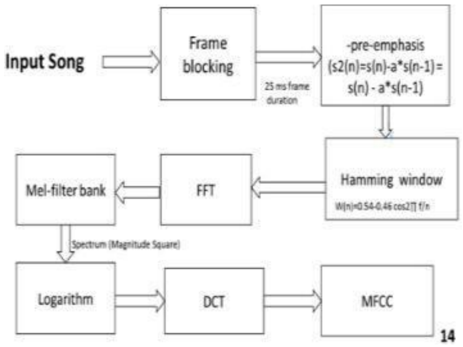


(a)

Fig 3: Layer diagram of neural network (a)

A. Block diagram of MFCC

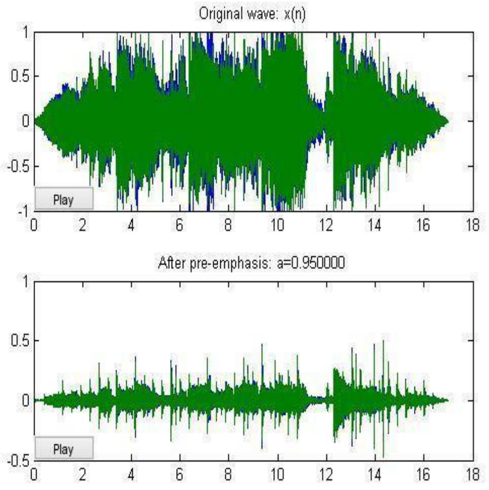
Deep-learning networks end in an output layer: a logistic, or soft max, classifier that assigns a likelihood to a particular outcome or label. We call that predictive, but it is predictive in a broad sense. Given raw data in the form of an image, a deep-learning network may decide, for example, that the input data is 90 percent likely to represent a person. The block of MFCC is given below for its training purpose .The process of MFCC consists of Fast Fourier Transform(FFT), Mel-Scale filtering , Taking log value and applying Discrete Cosine Transform(DCT) , which results in finding the cepstral values, finally the feature vectors are obtained.



V. EXPERIMENTAL RESULTS

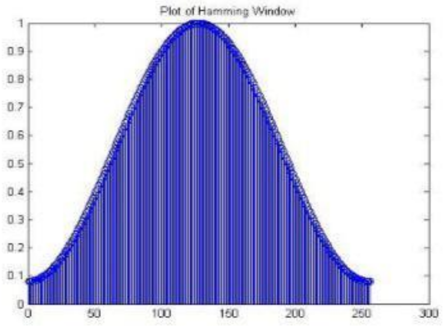
The results obtained in our project is shown below

The output obtained by applying the process like pre-emphasis, Hamming windowing , mel-filter bank output are explained and the windowing output in terms of time and frequency output are also obtained.

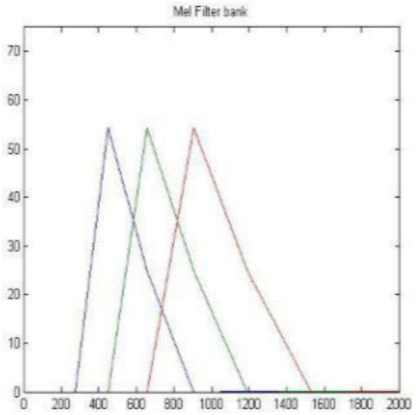


(a)

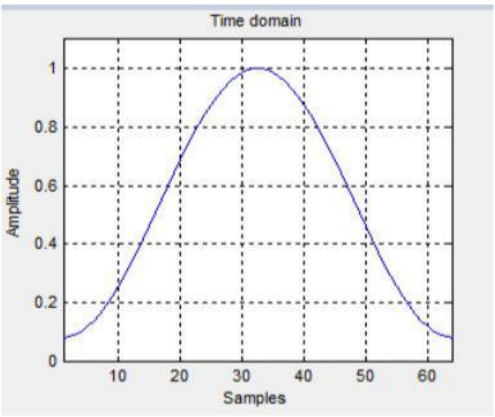
Thus the output of all such phases are resulted and finally the results of 20 singers are compared. This system also outperformed the DNN alignment approach by 20% relative at ERR and 30% relative at DCF min new. We have also analyzed decoupling of the sufficient statistics extraction by using separate GMM models for frame alignment, and for statistics normalization, and we have analyzed the use of BN and MFCC features (and their concatenation) in the two stages. We have also shown the effect of using full-covariance variants of the GMM models



(b)

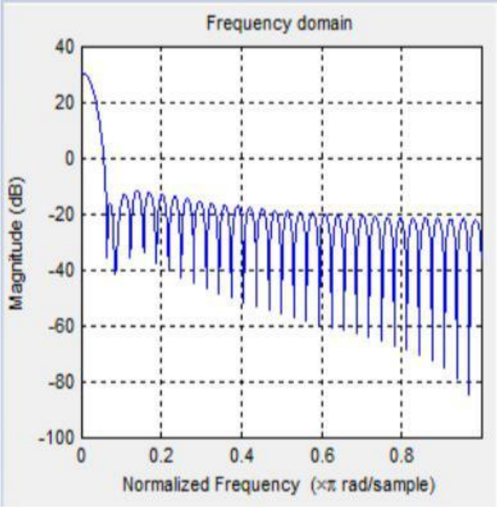


(c)



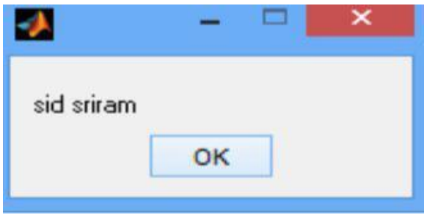
(d)

Window design analysis tool helps in obtaining the windowing output for the following outputs.



(e)

Fig 4: Original wave is represented as x (n) and the pre emphasis output is shown below with the value of a=0.95 (a), hamming window output is represented as (b), Mel-filer bank output is shown in (c), hamming window output in terms of time domain is shown in (d), hamming window output in terms of frequency domain is shown in (e).The singer name is finally displayed with the text box shown as (f)


sid sriram

(f)

VI. CONCLUSON

We have analyzed the i-vector based systems with Deep Neural Network (DNN) Bottleneck (BN) features together with the traditional MFCC features, and we have demonstrated substantial gain for NIST SRE 2010, telephone condition. Our best results, with BN trained on Fisher English and BN stacked with baseline MFCC, outperformed the baseline system relatively by 63% at EER and 70% at the DCF min new point. This system also outperformed the DNN alignment approach by 20% relative at ERR and 30% relative at DCF min new. We have also analyzed decoupling of the sufficient statistics extraction by using separate GMM models for frame alignment, and for statistics normalization, and we have analyzed the use of BN and MFCC features (and their concatenation) in the two stages. We have also shown the effect of using full covariance variants of the GMM models

REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath George Dahl, and Brian Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, November 2012.

[2] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, “A novel scheme for speaker recognition using a phonetically-aware deep neural network,” in ICASSP, 2014.

[3] Y. Lei, L. Ferrer, M. McLaren, and N. Scheffer, “Comparative study on the use of senone-based deep neural networks for speaker recognition,” Submitted to IEEE Trans. ASLP, 2014.

[4] Garcia-Romero D., Zhang X., McCree A., and Povey D., “Improving speaker recognition performance in the domain adaptation challenge using deep neural networks,” in SLT, 2014.

[5] Y. Song et al, “i-vector representation based on bottle neck feature for language identification,” in IEEE Electronics Letters, 2013.

[6] Pavel Matˇejka et al., “Neural network bottleneck features for language identification,” in IEEE Odyssey: The Speaker and Language Recognition Workshop, Joensu, Finland, 2014.

[7] Ignacio Lopez-Moreno, Javier Gonzalez-Dominguez, and Oldˇrich Plchot, “Automatic language identification using deep neural networks,” in ICASSP 2014, Florence, Italy, 2014.

[8] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” Audio, Speech, and Language Processing, IEEE Transactions on, vol. PP, no. 99, 2010.

[9] M. Diez, A. Varona, M. Penagarikano, L.J. Rodriguez-Fuentes, and G. Bordel, “On the use of phone log-likelihood ratios as features in spoken language recognition,” in SLT, 2012.

[10] Jeff Ma et al., “Improvements in language identification on the RATS noisy speech corpus,” in Interspeech 2013, Lyon, France, 2013.

[11] Najim Dehak Fred Richardson, Douglas A. Reynolds, “A uni- fied deep neural network for speaker and language recognition,” in Interspeech, 2015.

[12] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, “Deep feature for text-dependent speaker verifi- cation,,” Speech Communication, vol. 73, pp. 1–13, October 2015.

[13] Takanori Yamada, Longbiao Wang, and Atsuhiko Kai, “Improvement of distant-talking speaker identification using bottleneck features of DNN.,” in INTERSPEECH. 2013, ISCA.

[14] Yaman S., Pelecanos J., and Sarikaya R., “Bottleneck features for speaker recognition,” in Odyssey, 2012.

[15] Daniel Garcia-Romero and Alan McCree, “Insights into deep neural networks for speaker recognition,” in Interspeech, 2015.

[16] Yao Tian, Meng Cai, Liang He, and Jia Liu, “Investigation of bottleneck features and multilingual deep neural networks,” in Interspeech, 2015.

[17] Sandro Cumani, Olda Plchot, and Pietro Laface, “Comparison of hybrid dnn-gmm architectures for speaker recognition,” in ICASSP, Submited to ICASSP 2016.

[18] Mitchell McLaren, Martin Graciarena, and Yun Lei, “Advances in deep neural network approaches to speaker recognition,” in ICASSP, 2015.

[19] Kornel Laskowski and Jens Edlund, “A Snack implementation and Tcl/Tk interface to the fundamental frequency variation spectrum algorithm,” in Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10), Valletta, Malta, May 2010.

[20] David Talkin, “A robust algorithm for pitch tracking (RAPT),” in Speech Coding and Synthesis, W. B. Kleijn and K. Paliwal, Eds., New York, 1995, Elseviever.

[21] Martin Karafi´at, Franti´sek Gr´ezl, Karel Vesel´y, Mirko Hannemann, Igor Sz″oke, and Jan Cernock´y, “BUT 2014 Babel sys- ˇ tem: Analysis of adaptation in NN based systems,” in Interspeech 2014, 2014, pp. 3002–3006