

# Supervised Speaker Identification

Adam O'Donovan and Balaji V Srinivasan

[AdamOD@gmail.com](mailto:AdamOD@gmail.com) [er.balajivasan@gmail.com](mailto:er.balajivasan@gmail.com)

## Abstract:

In this paper we present a comparison of modern acoustic features and quantify their effectiveness at classification of human speakers. Features that were evaluated were several sets of Neuromimetic Cortical model features [4], linear predictive coefficients (LPC), Mel Frequency Cepstral Coefficients (MFCC), and Line spectrum pair parameters. Each of these feature sets were independently evaluated based on their ability to classify between two speakers using 1) Support Vector Machine (SVM), 2) Probabilistic neural networks (PNN), and 3)-nearest neighbor algorithm. Classification error of each feature and each classifier are compared and the results are discussed

## 1. Introduction:

Humans have an amazing ability to distinguish between audio signals coming from multiple simultaneous speakers. Additionally, humans are very good at classifying other human sources in these environments. This is in part due to the ability of humans to directionally isolate incoming sounds [1] and to perform higher level classification and processing of the directionally isolated sound [6]. Recently, spherical microphone arrays have gained attention due to the fact that they can isolate sounds directionally in a similar way to that of the human auditory system. The goal of this project is to explore the ability of modern machine learning algorithms to provide a means of mimicking the second, higher level processing of the brain, and classify the output of a directional sensor such as a spherical microphone array [3,4]. The applications of a system that can mimic this human ability are vast including speaker tracking and identification, teleconferencing and surveillance to name a few.

We hypothesize that the K-NN algorithm will perform fairly poorly due to the high probability of including insignificant features and noisy data. Additionally we believe that neural nets and SVM will perform significantly better than K-NN. In this project we have compiled a large set of features and several modern classification algorithms to evaluate. The main objectives were to identify the set of features which would bring out the distinction between different speakers and to study how different classifiers would classify these features. Four separate sets of features and their combinations with each other were studied. The performances of these features were tested with Nearest Neighbor Classifier, Support Vector Machine and Neural Networks.

This report is organized as follows. In section 2, we discuss the different classifiers that were used to classify and compare these features. In section 3, we discuss about the different features that were studied and the reasons why those features were considered. In section 4 we provide some results that were obtained for the different features with the different classifiers.

## 2. Methods

To test the various feature extraction techniques we recorded a small dataset consisting of two male speakers speaking continuously. Each of the recordings were recorded at 44100 samples per second with 16 bits of resolution. For each classification method 5 fold cross validation was used to evaluate the classification error. Here we briefly discuss the classifiers we investigated along with the feature sets investigated.

### 2.1 Nearest Neighbor

The k-nearest neighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The features which perform well with the k-NN are those which are located physically close to each other in the feature space. The Euclidean distance was used as the distance metric in our experiments.

It must be noted that the k-NN algorithm works well when the features of different classes are well separated in the feature space. If a particular set of features is classified poorly by a k-NN classifier, it goes to suggest that the features need some kind of transformation to another space for separability. In this project, we have considered 1-NN, 3-NN and 5-NN classifiers

## **2.2 SVM**

Support vector machines (SVMs) [10] are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers. Support vector machines map input vectors to a higher dimensional space where a maximal separating hyperplane is constructed. The mapping function is called a kernel.

The classifier stores the details of the separating hyper-plane and also the feature vectors which lie close to the separating plane. Such vectors are termed as Support Vectors. The understanding is that those feature vectors which lie far from the separating plane can be classified comfortably. Classifying those which lie close to the plane is important for the performance of the classifier because these feature vectors define the boundary of the class.

The support vector machine can be used with different kernel functions. In this project we have studied the performance of the classification for a linear and radial basis function kernels.

## **2.3 Probabilistic Neural Networks**

Probabilistic Neural nets are a form of radial basis neural networks that classify data based on a probabilistic framework [7]. In fact, they are quite similar to the K-Nearest neighbor algorithm except they incorporate a weighting function in the computation of the class vote and a method for optimizing these weighting functions. The network used consists of 3 layers.

- 1) The input layer which has one neuron for each of the feature vector elements and a bias.
- 2) Hidden layer- In this layer there is one neuron for each class in the problem. The input to these neurons is a vector of distances between the input value and the values used to train the network. The activation function used was the Gaussian.
- 3) Pattern layer/ summation layer – The outputs of the hidden layers are fed into a competitive layer which weights the output of the hidden layer and decides which class to choose based on the input that has highest value or probability. The output of the network is the class that is most probable for a given input.

These networks have advantages over perceptron networks in that they can implement arbitrary decision surfaces and do not suffer from convergence issues[7]. The implementations used for these experiments was that as implemented in Matlab in the Neural Networks Toolbox

## **3. Features from Speech Signal**

A given speech signal has a lot of features associated with it. However, the challenge is to analyze those features that can distinguish between speakers. In order to study such features, we have considered the different ways of interpreting the speech signal. Firstly we studied how a speech is analyzed by the human brain and what features might be used by humans to

distinguish between speakers. Then we discuss features that have been used to synthetically produce human voices and use these features and their variants to study their utility in speaker recognition.

### 3.1 Cortical Neuromimetic Model:

The Cortical Neuromimetic model proposed initially in [5] and further developed in [4] attempts to process the audio signal in a way that mimics mammalian physiological auditory processes. Because of the ease of which humans can classify multiple speakers speaking simultaneously [6] a model that filters the signal in a way that mimics the human auditory system is a natural place to start. This model is particularly attractive because it has been shown to preserve and represent those features which humans can perceive most clearly. Also, the model has been shown to provide a natural method for separating the timbre and pitch of an acoustic signal [4].

1) The first stage of processing is designed to mimic the transformation that is performed in the cochlear stage of the auditory system. In this stage an acoustic spectrogram is generated by first windowing the audio stream into short sequences and then applying a bank of band-pass filters designed to mimic the response of the cochlea to the windowed audio. An example of the resulting spectrogram for the phrase "linearly constrained" is shown in figure 1.

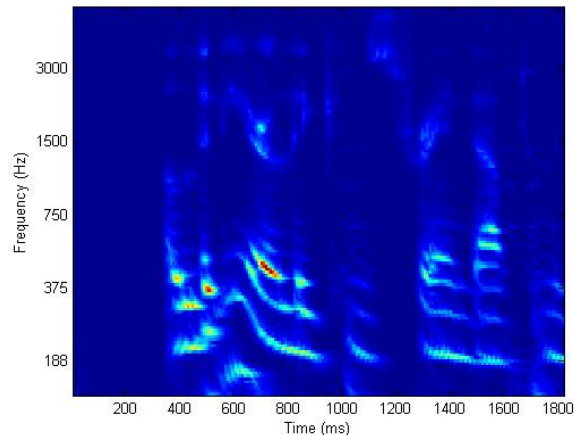


Figure 1. Auditory spectrogram for the phrase "linearly Constrained".

2) The second stage of processing mimics the late stage processing that occurs in the higher central auditory stage and primary auditory cortex. This stage attempts to analyze the spectral and temporal modulation of the auditory spectrogram [6]. This stage can be thought of as treating the auditory spectrogram as an image and performing convolution with 2 dimensional filters designed to respond to varying modulation rates and scales in the spectrogram. The filters are classified by two values, a rate in Hz which responds to the sloping progression of spectral features through time, and a scale in cycles per octave indicating how many oscillations the filter makes along the vertical scale in the auditory spectrogram. This effectively isolates features that are varying with different dynamics in the spectrogram. For instance, in figure 1 at the first onset of the speech signal we see that the spectral content seems to be traveling upward with a positive slope on a fairly small scale. We would expect a filter with positive rate and a scale around that of the feature ripples to respond to a feature like this. Figure 2 shows the cortical representation of the above auditory spectrogram for a rate of 8Hz and a scale of 2 cycles per octave.

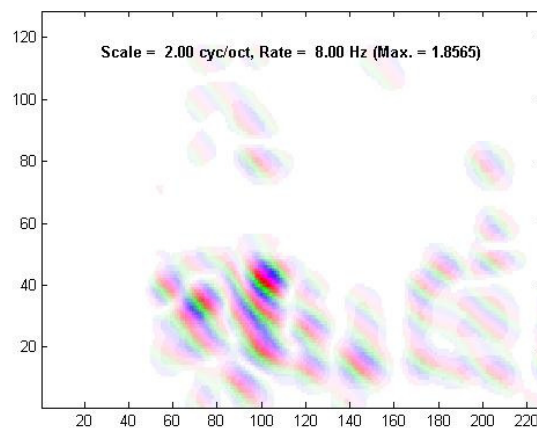
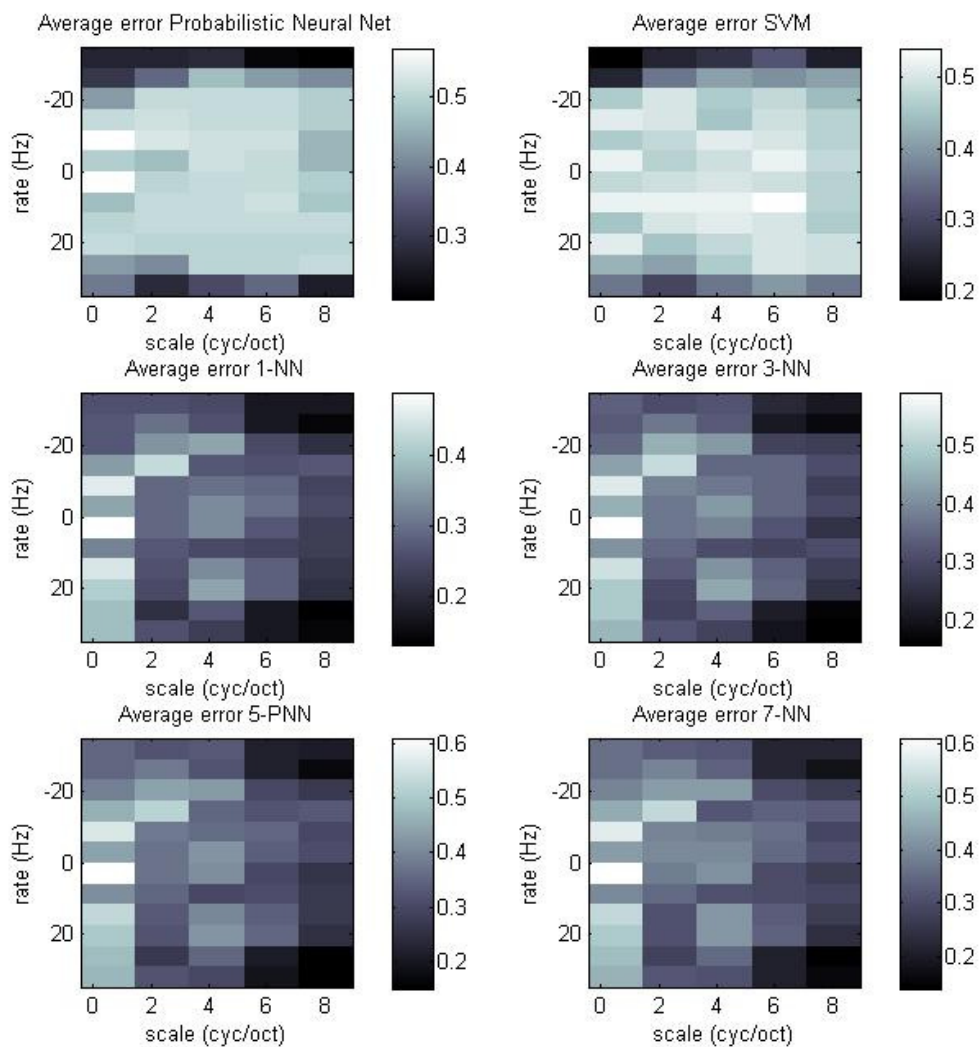


Figure 2. Cortical response with a rate of 2 cyc/oct and a scale of 8Hz. for the phrase "Linearly constrained"

To generate a feature vector using this method we first segmented the input speech signals from each speaker into 1 second bins. Bins that correspond to silence were filtered out by estimating the average power in the signal for a given bin and comparing it to the average power in the total input data. Each of the 1 second samples was then processed to extract the cortical response using the NSLtoolbox for Matlab [4]. Because we have no a priori knowledge of what rates and scales will best classify speakers we computed the response for all combinations of scales = {0.5 1 2 4 8} and rates = {-32,-16, -8, -4, -2, -1, 1, 2, 4, 8, 16, 32}. Additionally because each cortical response evolves with time within the window we performed a time averaging over the window to generate a single feature element for each frequency in the auditory spectrogram and each rate and scale. In our experiments this generated a single 128 point feature vector for each 1 second audio sequence. These features were then fed through the algorithms to evaluate the best performer. Figure 3 shows the error rate calculated using 5 fold cross validation for each of the algorithms investigated.



**Figure 3. Error rate for classification using ,PNN, SVM, and K-NN algorithms. Each pixel in the plots represents the error rate for a given rate and scale cortical feature vector.**

It is evident in Figure 3 that the features that seem to classify between speakers most effectively are those concentrated at the extremes. That is, large and small rates seem to classify better than mid range rates. In all classification algorithm rates as rates become more and more sloped the accuracy increases. In the case of neural networks the largest scale outperforms other values where as for PNN and SVM classification performance is more even across scale. The rate and scale that performed best was then used in the overall analysis in section 4.

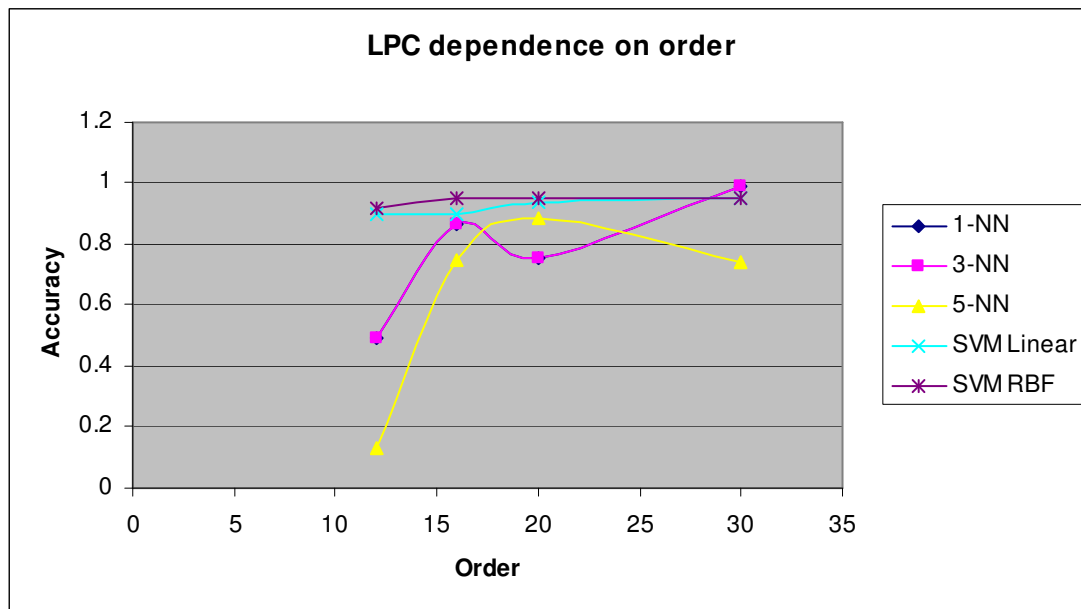
### 3.2 Linear Predictive Vocoder

Speech is produced by a cooperation of lungs, glottis (with vocal cords) and articulation tract (mouth and nose cavity) [13]. For the production of voiced sounds, the lungs press air through the epiglottis, the vocal cords vibrate, they interrupt the air stream and produce a quasi-periodic pressure wave. The human speech production can be illustrated by a simple model. In this model, the lungs can be replaced by a DC source, the vocal cords by an impulse generator and the articulation tract by a linear filter system. A noise generator produces the unvoiced excitation. In practice, all sounds have a mixed excitation, which means that the excitation consists of voiced and unvoiced portions. The filter, representing the articulation tract, is a simple recursive digital filter; its resonance behavior (frequency response) is defined by a set of filter coefficients. The coefficients that are used in Vocoder are the Linear Predictive Coefficients. For a speech signal  $x(n)$ , LPC is given by,

$$x(n) = 1 + \sum_{i=1}^k a_i x(n-i)$$

where  $a_i$  are the linear predictive coefficients and  $k$  is the order.

Since Linear Predictive Coefficients (LPCs) are used to synthesize speech signal, it can be inferred that LPCs can be used to model the speaker characteristics of speakers. As seen, LPCs are the coefficients that express the signal at a particular instant as a linear function of past values. The LPCs can be modeled for different orders. Figure 4. shows how the LPCs of different orders classify the signal with different classifiers.



**Figure 4: LPC dependence on Order**

Based on the results here, an LPC order of 30 was used to compare the performances. Higher orders involve a lot more computation and hence were not considered. The accuracy measured is for a test speech signal of length 10 seconds after training for two speech signals from two different speakers, each of length 60 seconds.

**3.3 Line Spectral Pair:**

If LPC coefficients are directly quantized, some of the poles located just inside the unit circle before quantization may shift outside the unit circle after quantization, causing instability. One way by which this problem is overcome in vocoder is to convert the LPCs to Line Spectrum Pair (LSP) parameters [14] which are more amenable to quantization. Since the LSP is a transformation of the LPCs, it would be a good experiment to use these features and study their effectiveness in classification. The LSP coefficients are defined by first defining polynomials  $P(z)$  and  $Q(z)$ ,

$$P(z) = 1 + (a_1 - a_{10})z^{-1} + (a_2 - a_9)z^{-2} + \dots + z^{-n}$$

$$Q(z) = 1 + (a_1 + a_{10})z^{-1} + (a_2 + a_9)z^{-2} + \dots + z^{-n}$$

where  $a_k$  are the LPC coefficients. Now, rearranging these two polynomials,

$$P(z) = (1 - z^{-1}) \prod_{k=2,4,\dots} (1 - 2 \cos w_k z^{-1} + z^{-2})$$

$$Q(z) = (1 + z^{-1}) \prod_{k=1,3,\dots} (1 - 2 \cos w_k z^{-1} + z^{-2})$$

where  $w_k$ 's are the LSP coefficients.

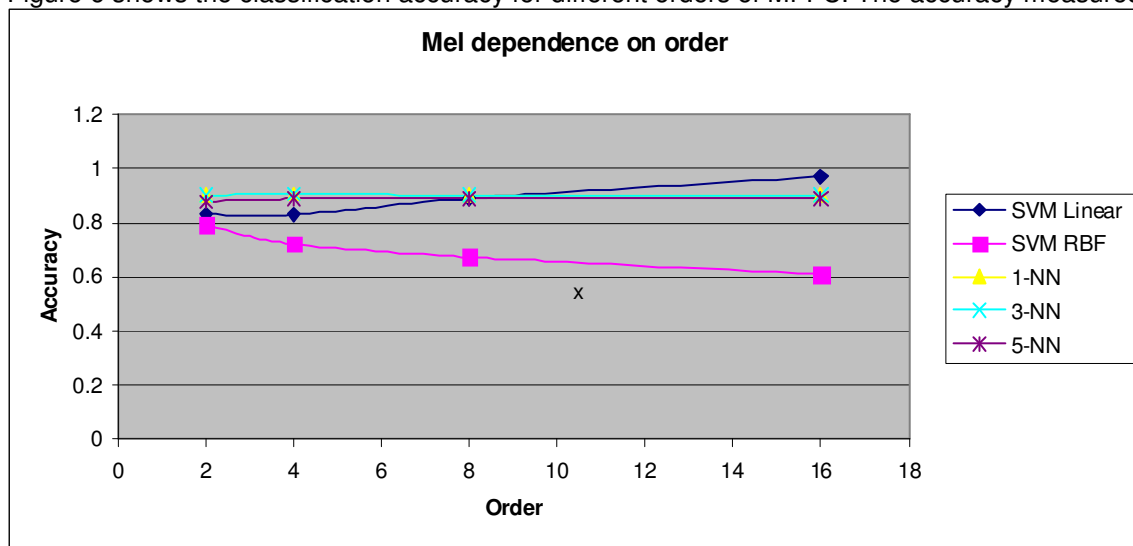


**Figure 5 : LSP dependence on Order**

Similar to the LPCs, LSPs can be obtained for a speech signal for different orders. Figure 5 shows how the LSPs of different orders classify the signal with different classifiers. Based on the results here, LSP of order 12 was chosen for the experiments in this project. This was because there was no appreciable improvement in increasing the order. The accuracy measured is for a test speech signal of length 10 seconds after training for two speech signals from two different speakers, each of length 60 seconds.

### 3.4 Melspectrum

Mel Frequency Cepstral Coefficients (MFCCs) are coefficients that represent audio. They are derived from a type of cepstral representation of the audio clip (a "spectrum-of-a-spectrum"). The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are positioned logarithmically (on the mel scale) which approximates the human auditory system's response more closely than the linearly-spaced frequency bands obtained directly from the FFT or DCT. MFCCs are mostly used for audio compression. However, since MFCC are known to capture audio features, the MFCC were also used for the experiments. Figure 6 shows the classification accuracy for different orders of MFCC. The accuracy measured



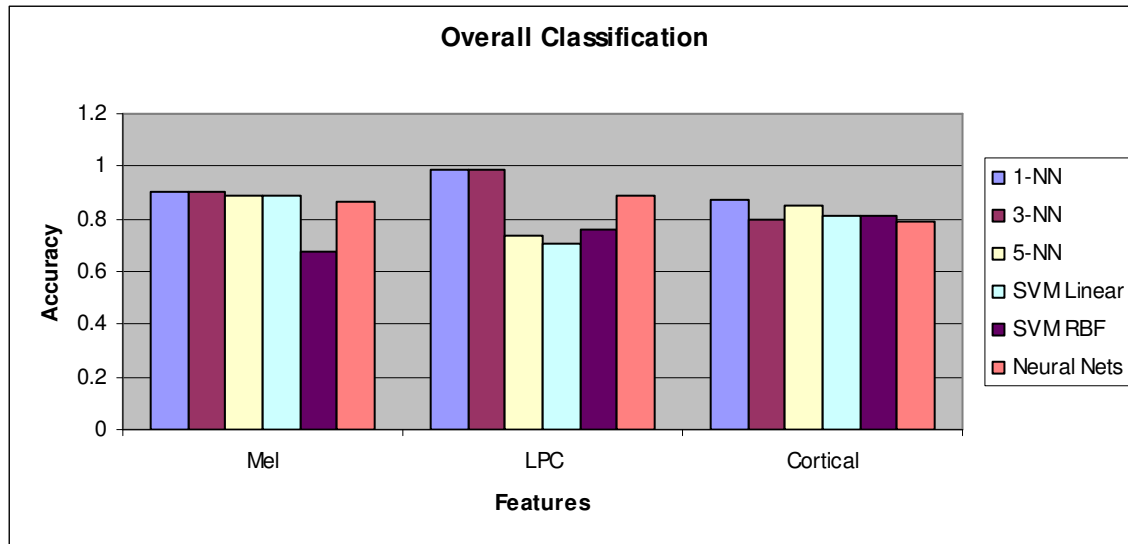
**Figure 6: Mel dependence on Order**

is for a test speech signal of length 10 seconds after training for two speech signals from two different speakers, each of length 60 seconds. It can be seen that the a linear-kernel SVM has the best performance at an order of 16. However at this order the accuracy of RBF-kernel SVM drops to 60%. So for the sake of comparison, order 8 was used where the accuracies of Nearest Neighbor and SVM were almost identical.

### 4. Results:

When the feature sets were tested with the various classifiers, it was found that, against our hypothesis, the Nearest Neighbor outperformed the SVM and Neural Nets. Such an observation can possibly be attributed to the fact that that the signals treated here are audio signals sampled at a high rate and the features were extracted from small windows of this signal. Hence the number of training samples available for each speaker is very high and hence Nearest Neighbor performed better than other classifiers.

Figure 7 summarizes the overall classification result for each of the features analyzed in this project. It can be seen here that with all three features, the Nearest Neighbor outperforms SVM and Probabilistic Neural Nets.



**Figure 7**

Comparing between the performances of SVM and PNN, it can be seen that the Neural Nets outperforms the SVM classifier with the LPC while SVM marginally outperforms the Neural Nets with the cortical features. With the Mel coefficients, both Neural Nets and SVM performed pretty similar.

### 5. Future Work:

When considering the problem of face recognition, it is possible to reduce dimensionality problems and redundancy issues by using a PCA space or a LDA space [16]. It would be interesting to see how such a projection works with speech recognition problems. As a part of the future work, we would like to see how such a projection occurs. The mel cepstral features discussed here are known to work better with Gaussian Mixture Models [15]. We would analysis the performance of such a classification model also.

By considering how exactly human beings perform speaker recognition, it is easy to see that the Human Neural Network does not use a single feature or single method to classify a speaker. It analyzes different features of a speaker to find his identity. On extending the same analogy, it would be intuitive to use a combination of different methods and different features and use a voting to make a final decision. The different methods used here and suggested in previous paragraph can be combined or cascaded in order to build a good ensemble classifier.

Once such a classifier is available, it will be interesting to observe the performance of the classifier in a multiple-speaker environment to observe the classification accuracy in an overlapping mixed speech.

### 6. Conclusion:

In this project we have considered different features that can classify a speaker based on a speech signal. The performance of different features across different classifier was also studied and discussed.

### Reference:

- [1] Cherry, E. C. (1953) Some experiments on the recognition of speech, with one and with two ears. *Journal of Acoustical Society of America* 25(5), 975--979.
- [2] Zhiyun Li and Ramani Duraiswami. Flexible and optimal design of spherical microphone arrays for beamforming. *IEEE Transactions on Speech and Audio Processing*, 15:702 – 714, 200.
- [3] J. Meyer and G. Elko, "A highly scalable spherical microphone array based on an orthonormal decomposition of the soundfield," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP'02)*, May 2002, vol. 2, pp. 1781–1784.
- [4] N. Zotkin, T. Chi, S. A. Shamma, R. Duraiswami Neuromimetic sound representation for percept detection and manipulation", *D. EURASIP journal on Applied Signal Processing*, vol. 2005(9), 2005, pp. 1350-1364.
- [5] Chi, T., Ru P., and Shamma S., "Multiresolution spectrotemporal analysis of complex sounds", submitted to *Speech Communication*, 2003.
- [6] S. Bregman (1991). "Auditory scene analysis: The perceptual organization of sound", MIT Press, Cambridge, MA
- [7] Specht, D. Probabilistic Neural Networks for Classification, Mapping or Associative Memory. *IEEE Neural Networks*, 1988. San Diego, CA
- [8] Reynolds, D.A.; Rose, R.C., "Robust text-independent speaker identification using Gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on* , vol.3, no.1, pp.72-83, Jan 1995
- [9] Moreno, Pedro J. / Ho, Purdy P. (2003): "A new SVM approach to speaker identification and verification using probabilistic distance kernels", In *EUROSPEECH-2003*, 2965-2968.
- [10] C. Burges. "A tutorial on support vector machines for pattern recognition" *Data Mining and Knowledge Discovery*, 2:121-167, 1998.
- [11] Vergin, R.; O'Shaughnessy, D.; Farhat, A., "Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition," *Speech and Audio Processing, IEEE Transactions on* , vol.7, no.5, pp.525-532, Sep 1999
- [12] Philippe Thevenaz and Heinz Hugli, Usefulness of the LPC-residue in text-independent speaker verification, *Speech Communication Volume 17, Issues 1-2* , August 1995, Pages 145-157.
- [13] <http://en.wikipedia.org/wiki/Vocoder>
- [14] Soong, F.; Juang, B., "Line spectrum pair (LSP) and speech data compression," *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '84.* , vol.9, no., pp. 37-40, Mar 1984
- [15] Douglas A. Reynolds, Speaker identification and verification using Gaussian mixture speaker models, *Speech Communication Volume 17, Issues 1-2* , August 1995, Pages 91-108.
- [16] Martinez, A.M.; Kak, A.C., "PCA versus LDA," *Transactions on Pattern Analysis and Machine Intelligence* , vol.23, no.2, pp.228-233, Feb 2001