Automatic Speaker Recognition Using Neural Networks

Submitted To

Dr. Joydeep Ghosh

Prepared By

Brian J. Love
Jennifer Vining
Xuening Sun

EE371D Intro. To Neural Networks
Electrical and Computer Engineering Department
The University of Texas at Austin

Spring 2004

CONTENTS

1.0 INTRODUCTION

Humans use voice recognition everyday to distinguish between speakers and genders. Other animals use voice recognition to differentiate among sounds sources. For example, penguin parents can tell which baby chick is theirs by the baby's distinctive call. Similarly, a blind person can accurately classify speakers based solely on their vocal characteristics.

In general, speaker recognition can be subdivided into *speaker identification* (Who is speaking?) and *speaker verification* (Is the speaker who we think he or she is?). In addition, speaker identification can be *closed-set* (The speaker is always one of a closed set used for training.) or *open-set* (Speakers from outside the training set may be examined.). Also, each variant may be implemented as *text-dependent* (The speaker must utter one of a closed set of words.) or *text-independent* (The speaker may utter any type of speech) [1].

In this paper we explore the ability of a multilayer perceptron (MLP) to perform both text-dependent and text-independent speaker identification. Our networks are trained on several sets of acoustical parameters extracted from samples obtained from a closed set of speakers uttering a set of known words. Our primary feature extraction tools are linear predictive coding (LPC), cepstrum analysis, and a mean pitch estimation made using the harmonic product spectrum algorithm.

All software for this project was created using Matlab, and neural network processing was carried out using the Netlab toolbox. All source code and data files for this project, other than the Netlab software, can be found at:

- http://webspace.utexas.edu/lovebj/EE371D_TermProjectCode/

The Netlab toolkit may be obtained at the following URL:

- http://www.ncrg.aston.ac.uk/netlab/down.php

2.0 FEATURES OF SPEECH

One might wonder what information is needed to classify between genders or to classify the speech of multiple speakers. In fact, speech contains a great deal of information that allows a listener to

determine both gender and speaker identity. In addition, speech can reveal much about the emotional state and age of the speaker. For example, an Israeli engineer created a signal processing lie detector system that out performs the traditional polygraph test.

2.1 PITCH

Pitch is the most distinctive difference between male and female speakers. A person's pitch originates in the vocal cords/folds, and the rate at which the vocal folds vibrate is the frequency of the pitch. So, when the vocal folds oscillate at 300 times per second, they are said to be producing a pitch of 300 Hz. When the air passing through the vocal folds vibrates at the frequency of the pitch, harmonics are also created. The harmonics occur at integer multiples of the pitch and decrease in amplitude at a rate of 12 dB per octave – the measure between each harmonic [].

The reason pitch differs between sexes is the size, mass, and tension of the laryngeal tract which includes the vocal folds and the glottis (the spaces between and behind the vocal folds). Just before puberty, the fundamental frequency, or pitch, of the human voice is about 250 Hz, and the vocal fold length is about 10.4 mm. After puberty the human body grows to its full adult size, changing the dimensions of the larynx area. The vocal fold length in males increases to about 15-25 mm while female's vocal fold length increases to about 13-15 mm. These increases in size correlate to decreased frequencies coming from the vocal folds. In males, the average pitch falls between 60 and 120 Hz, and the range of a female's pitch can be found between 120 and 200 Hz. Females have a higher pitch range than males because the size of their larynx is smaller. However, these are not the only differences between male and female speech patterns [].

2.2 FORMANT FREQUENCIES

When sound is emitted from the human mouth, it passes through two different systems before it takes its final form. The first system is the pitch generator, and the next system modulates the pitch harmonics created by the first system. Scientists call the first system the laryngeal tract and the second system the supralaryngeal/vocal tract. The supralaryngeal tract consists of structures such as the oral cavity, nasal cavity, velum, epiglottis, tongue, etc.

When air flows through the laryngeal tract, the air vibrates at the pitch frequency formed by the laryngeal tract as mentioned above. Then the air flows through the supralaryngeal tract, which

begins to reverberate at particular frequencies determined by the diameter and length of the cavities in the supralaryngeal tract. These reverberations are called “resonances” or “formant frequencies”. In speech, resonances are called formants. So, those harmonics of the pitch that are closest to the formant frequencies of the vocal tract will become amplified while the others are attenuated [].

2.2.1 Relation to Human Speech

In human speech, the formants change based on the position of the tongue, jaw, velum, and other structures in the vocal tract. This is how humans articulate. The relationship between all of the possible formants has to do with what vowel is being voiced. There are two principles at play here: (1) each formant has a corresponding bandwidth, and (2) each formant falls in a known spectral interval of the bandwidth. Because the structure of each human’s vocal tract is unique, the formants for each vowel will be unique. However, as principle (2) suggests, the formants for individual vowels will be similar among all humans because they must be recognizable as a particular sound such as /æ/ or /i/ [].

2.2.2 The Variable Filter Model

Since each human’s vocal tract creates different formants for each vowel, we can perceive the vocal tract as a variable filter. The input to the filter is the pitch and its harmonics coming from the vocal folds, while the output of the filter (perceived sound from the mouth) is the gain of harmonics falling in the formant frequencies. The goal of extracting the formant frequencies from speech signals is estimating this variable filter’s transfer function because one human’s filter function will be distinct from others.

3.0 COMPARISON OF FEATURES

Which is a better feature of speech: pitch or formants? Since we want a system that is robust, we must compare the features based on whether or not they are robust enough to comply with the following desirable feature characteristics:

1. Cannot be mimicked or consciously controlled by the speaker
2. Unaffected by health problems of the speaker
3. Independent of speaking environment

4. Distinguishable from noise caused by the recording process

Pitch is good for distinguishing between genders because female pitch is generally higher than male pitch. However, it is easily mimicked (consciously controlled by the speaker), which can throw the system off by giving a false pitch reading. Another problem with using pitch as a feature is that it can be affected by the health and emotional state of the speaker, thereby resulting in an inaccurate pitch reading. On the upside, pitch can be extracted in the presence of electrical noise. Since such noise is high frequency while pitch information is low frequency, we can apply a lowpass filter to suppress the noise, leaving the pitch intact.

Like pitch, formants can help distinguish between genders because a female's formants occur at higher frequencies than a man's. For example, the average first and second formants for the vowel /u/ in a woman fall at 370 Hz and 950 Hz respectively while a man's falls at 300 Hz and 870 Hz []. The production of nasal sounds, which occur when the nasal cavity is used in articulation, yields a distinctive formant structure among individuals. Therefore, the formants associated with nasal sounds can be used with automatic speaker verification systems because of their uniqueness between humans. Because of this characteristic, formants are more resistant to mimicry than pitch. Unfortunately, formants do change due to emotional and health state. Overall, formants are best for individual speaker and speech recognition [].

Used together, pitch and formants can give a good indication of who is speaking if the speaker is healthy and speaking normally. In our analysis, the subjects have not tried to disguise their voice or talk strangely. However, some of the voice samples were made on different computers with different sound cards, but as mentioned earlier, filtering should make these differences negligible. As will be described later, we chose to use both pitch and formant information for speaker identification.

4.0 DIGITAL SIGNAL PROCESSING TOOLS FOR SPEAKER RECOGNITION

Initially, we studied speech signals constrained to the range of 20 – 8,000 Hz, the band where most speech information resides []. However, without performing any type of processing on the speech signal, little information exists beyond the temporal characteristics. We quickly determined that a

raw temporal signal offers little data that may be useful to train a neural network, other than the case where a network is employed as a feature extractor for a successive network. At this point, we realized that we would need to implement some form of feature extraction to obtain useful information from recorded speech. In addition, we recognized that any recorded speech signals would require a certain degree of preprocessing to enhance the quality of the sample.

4.1 FEATURE EXTRACTION

As described in Section 2.0, we can generally characterize an individual's speech by his or her pitch and formant locations. In this section, we explore several methods for extracting estimates of these values from digital speech samples.

4.1.1 Discrete Fourier Transform

As an initial guess, using the Discrete Fourier Transform (DFT) as a feature extractor appears to be a good idea. Unlike the time domain representation of the digital sound sample itself, the frequency magnitude does contain information about the pitch and formants. However, the spectral magnitude also holds a great deal of other information besides the pitch and formant locations. Ideally, we would like to analyze only the spectral envelope of the frequency magnitude [2]. Thus, in order to avoid any unnecessary information, we considered three other techniques to extract pitch and formant information.

4.1.2 Linear Predictive Coding

As mentioned above, we are primarily interested in obtaining an estimate of the spectral envelope of the frequency magnitude spectrum in order to find formant locations and thus classify speakers based on their unique speech patterns. Linear predictive coding (LPC) offers a powerful, yet simple method to provide exactly this type of information. Basically, the LPC algorithm produces a vector of coefficients that represent a smooth spectral envelope of the DFT magnitude of a temporal input signal. These coefficients are found by modeling each temporal sample as a linear combination of the previous p samples as shown below:

$$x(n) = \sum_{k=1}^{p} a_k \, x(n-k) \; + \; e(n) = \hat{x}(n) \; + \; e(n)$$

In this formula from [3], $\hat{x}$ is the estimated value of the n^{th} sample, and e(n) is the difference between the estimate and the true value. The p coefficients, a_k, that minimize the total error between the signals $\hat{x}$ and x are known as the p^{th} order LPC coefficients for the signal x. Intuitively, we can see that the LPC coefficients represent a generalization of the signal x, and thus can serve as a useful feature to characterize an individual's voice. The 13^{th} order LPC coefficients for three utterances of the word "close" by a single male speaker are present in Figure 1. In this figure and in our analysis, we have removed the first LPC coefficient, a_1, which by convention is always 1. Thus, Figure 1 shows a_2 through a_{13}.

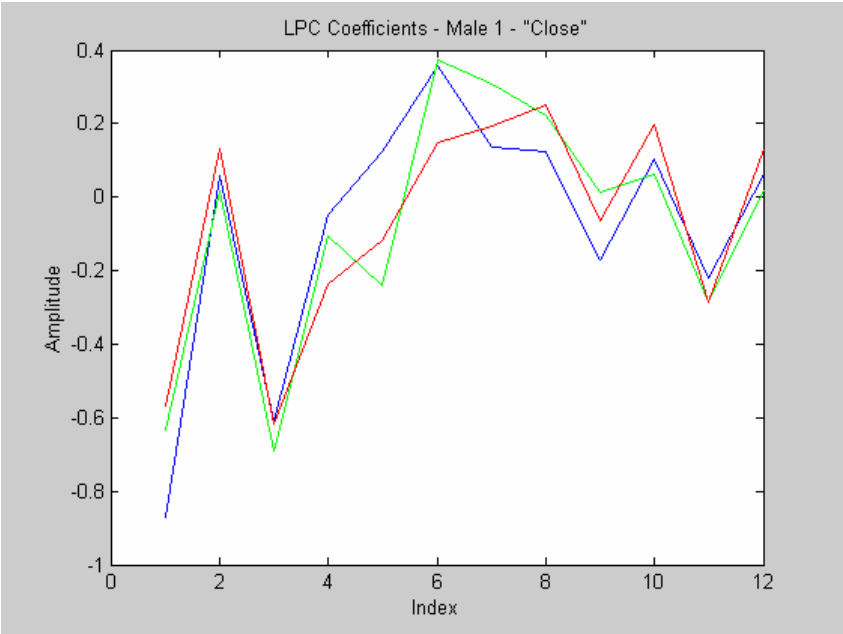

Figure 1. 12^{th} Order LPC Coefficients for 3 Utterances of the Word 'Close'

As we can see from Figure 1, the coefficient peaks tend to occur at the same locations. These are the formant locations. When used properly, the LPC coefficients emphasize the location of the formants in the frequency spectrum. They is also greatly influenced by the glottal shape and vocal cord duty cycles. With this in mind, we would expect the LPC coefficients to provide a good generalization of the speaker's unique vocal characteristics.

4.1.3 Cepstral Analysis

The cepstrum of a signal provides information similar to that of LPC coefficients in that both provide an estimated spectral envelope of the DFT magnitude. The idea of cepstral analysis was first introduced in 1963 in [4] for echo location purposes. The term "cepstrum" is a play on the word "spectrum." Similarly, the unit of the cepstrum is "quefrency."

Although defined in many different ways, the cepstrum of a signal is generally accepted to be the DFT of the log magnitude of the DFT of that signal*, or:

DFT (log (| DFT(signal) |))

No matter how it is defined, the cepstral process removes the decreasing linear trend of the spectrum that occurs with increasing frequency. In other words, it causes the spectrum to flatten out. This detrending effectively removes information about the speaker's glottal shape and vocal cord duty cycles. The loss of this information is desirable in many applications such as extracting phonetic information. The drawback to using the cepstrum for speaker identification is that it deletes most information about the speaker's identity other than the formant locations.

In addition to the direct computation defined above, the cepstrum can also be calculated as an n^{th} order vector calculated recursively from the LPC coefficients using the relationship (here, the a_{i-j} are the LPC coefficients):

$$c_i = a_i + \frac{1}{i}\sum_{j=1}^{i-1}(j)a_{i-j}c_j, i = 1, \ldots, n.$$

This method is also defined in a number of similar, yet different ways in the literature. The most common discrepancy is the definition of the initial cepstral coefficient c_1. In our processing we chose to calculate our cepstral coefficients in the recursive manner described above. We also followed the definition set forth in [5] where $c_1 = a_1$.

The 12^{th} order recursive cepstral coefficients for the three utterances of the word "close" used in Figure 1 are presented in Figure 2. As expected, we have lost the trend information present in

Figure 1. However, we have located the formant locations quite nicely. In our analysis we will show that the information lost to the cepstral coefficients does degrade their ability to differentiate between speakers.

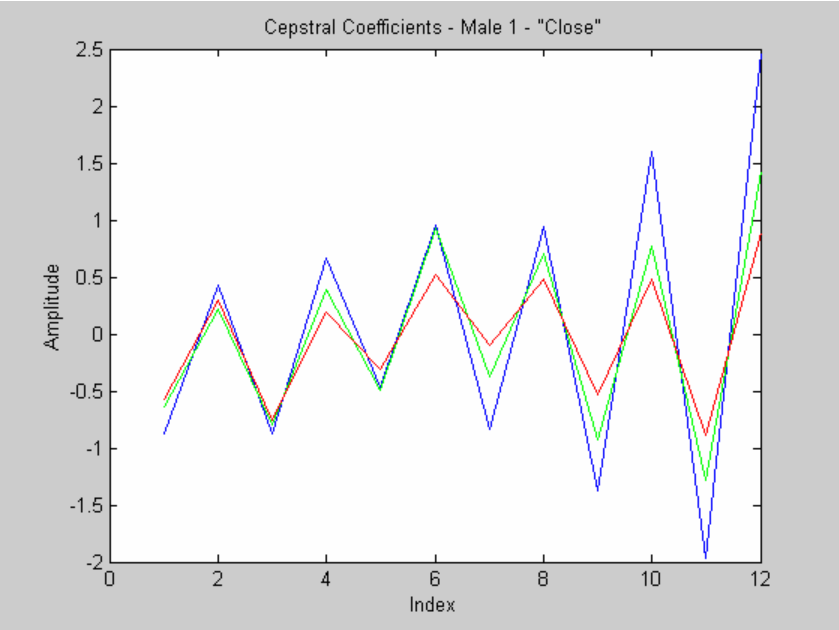

Figure 2. 12th Order Cepstral Coefficients for 3 Utterances of the Word "Close"

(* Other definitions of the cepstrum include replacing the last DFT with a discrete cosine transform, as in [6], and the inclusion of a melfrequency resampling as in [2]. Additionally, many texts mistakenly replace the last DFT with an inverse DFT. This however, is not the original definition.)

4.1.4 Pitch Detection with Harmonic Product Spectrum

In practical speaker identification systems, as the number of speakers grows it is often beneficial to first determine whether the speaker is most likely a male or female before or in addition to the use of with either LPC or cepstral coefficients. As mentioned in Section 2.1, the pitch of a speaker's voice should be a good indicator of the sex of the speaker.

In order to use this type of analysis, we needed some type of pitch detection algorithm. For this purpose, we chose to use the harmonic product spectrum (HPS) technique. HPS is based on the assumption that voice is essentially composed of a fundamental frequency (f_0), i.e. the pitch, and harmonics that occur at integer multiples of f_0. Thus, if we downsample the DFT of a voice signal

by 2, 3, 4 etc. and then multiply the original with its (zero-extended) downsampled versions, the product should be a single peak located at the pitch frequency [7]. This idea is displayed in Figure 3.

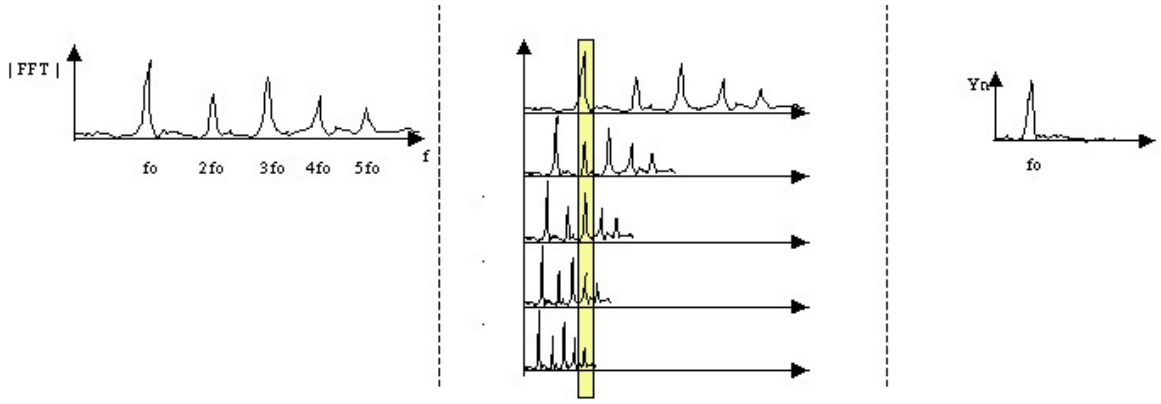

Figure 3. HPS Algorithm

Since voice can, in general, contained a variety of pitch elements, we used the HPS algorithm to create a new technique to generate a mean pitch value for a speech signal. We chose to use HPS to calculate the pitch of 30 ms frames of speech, each overlapping by 10 ms. Then, we considered the average pitch over all 30 ms frames in a single speech signal to be the single pitch value representing that signal.

In order to test this algorithm and to determine the accuracy of pitch for detecting the sex of the speaker, we collected 63 data samples of both male and female speakers saying the phrase “Hook ‘em Horns” and determined the mean pitch of each voice sample. The results are shown graphically in Figure 4, where red circles correspond to female pitch and blue crosses to male pitch. Clearly, male and female voices are almost linearly separable. In general, the linear discriminant is approximately 168 Hz [1].

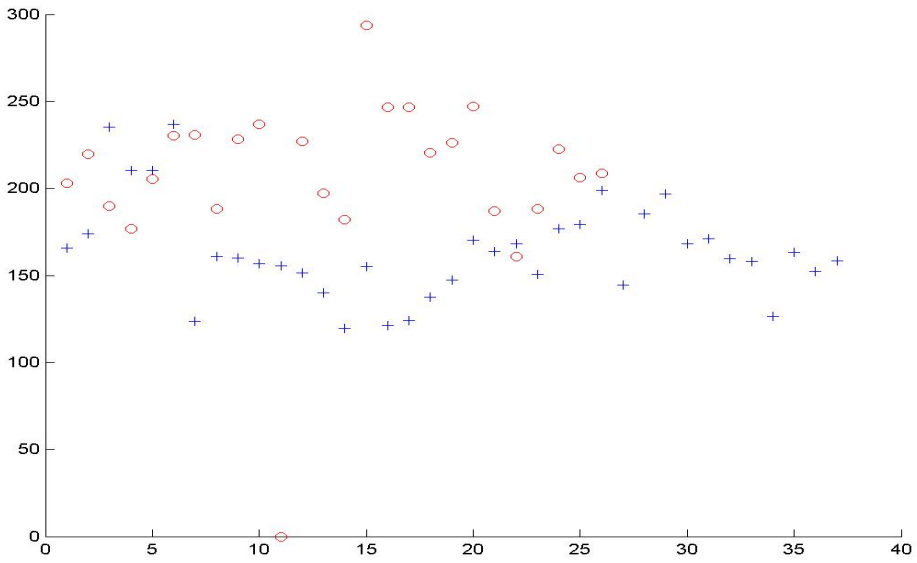

Figure 4. Extracted Mean Pitch of Male and Female Speakers

4.2 DATA PREPROCESSING

As with any real world digital application, any voice samples that we record will be corrupted by a finite amount of noise. In addition, since our subjects will not have a perfect reaction time, there will be durations of signal at the beginning and end of the recording where no speech will be present. In general, we would expect that both noise and the analysis of non-speech information would degrade the quality of our analysis and results. Thus, we implemented two modules of data preprocessing to be performed on our data before feature extraction.

4.2.2 Removal of Non-Speech Signal Durations

First, we wanted to remove all non-speech samples from the recorded, temporal signals. We implemented this using an energy detection algorithm developed in a heuristic manner from our data. Since none of our recordings contained speech in the first 100 ms of recording time, we analyzed this time frame and generated an estimate of the noise floor for the speaking environment. Then we analyzed each 20 ms frame and removed those frames with energy less than the noise floor. A recording of the word “close” truncated in this fashion is shown in the second plot of Figure 6.

4.2.3 Lowpass Filtering with Hamming Window

As previously mentioned, in digital signal processing electronic noise is generally assumed to be composed of mainly high frequencies. Thus, in order to improve the signal to noise ratio (SNR) of our samples, we chose to lowpass filter our signals after the removal of non-speech components. This filtering was performed with the use of Hamming window as shown in Figure 5. A Hamming filtered signal of the word “close” is presented in the third plot of Figure 6.

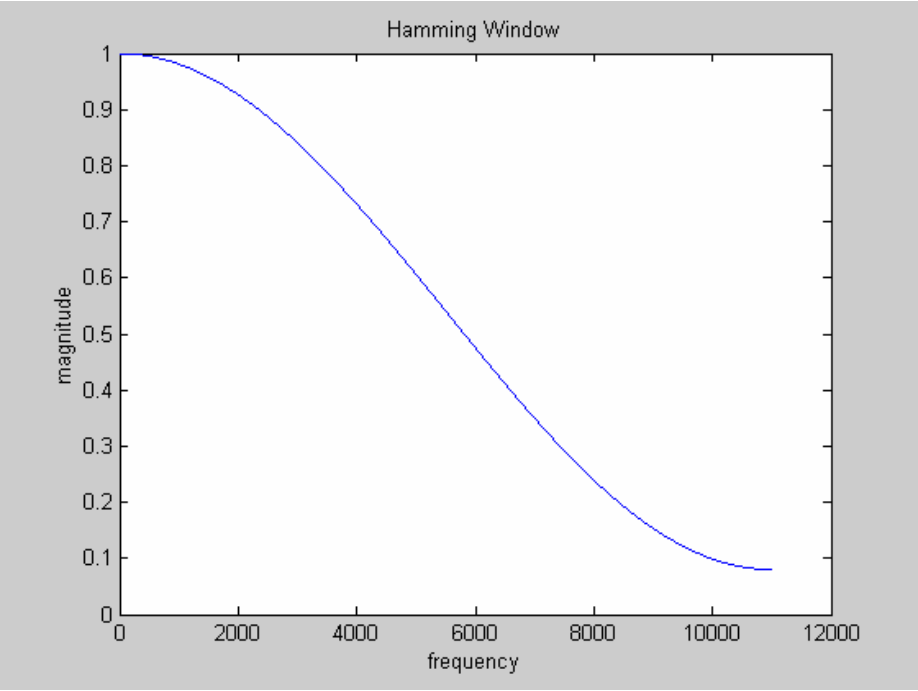

Figure 5. Hamming Window

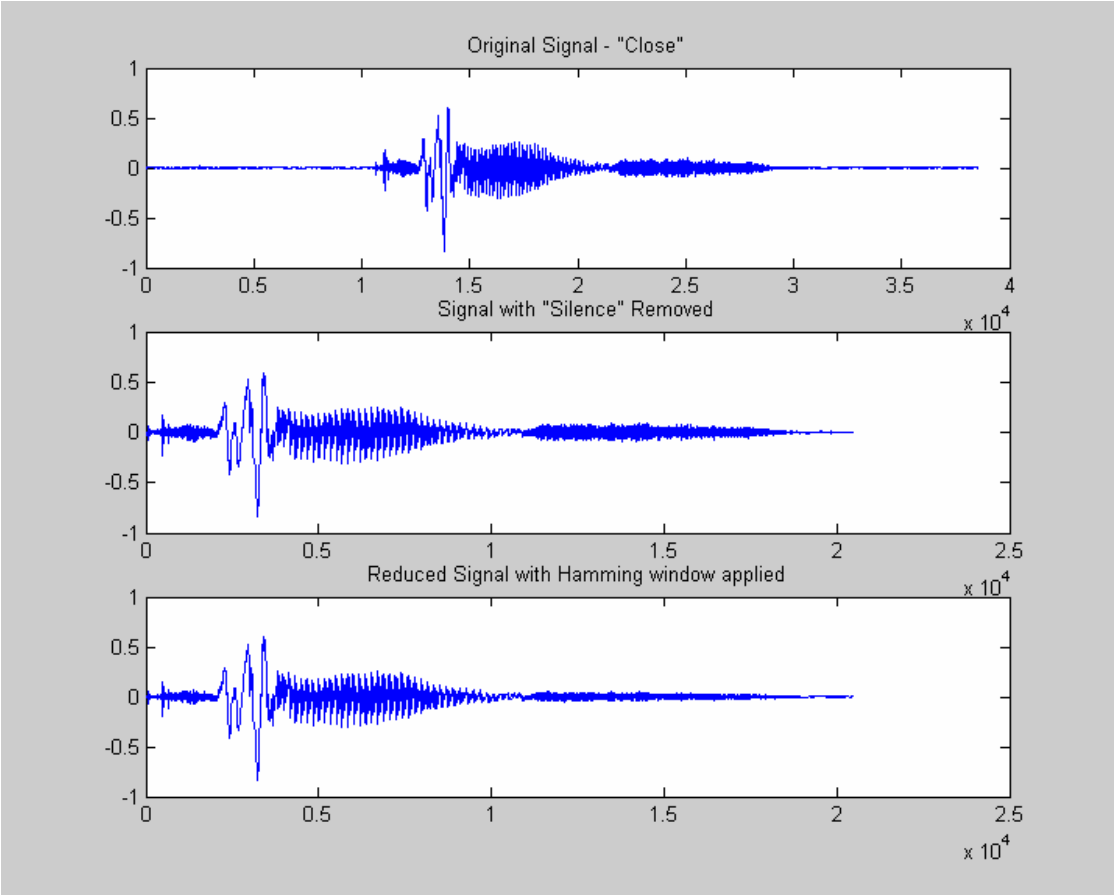

Figure 6. Preprocessing Stages Performed on the Word "Close"

5.0 SPEAKER RECOGNITION WITH ARTIFICIAL NEURAL NETWORK

With the signal processing tools in place to both preprocess and extract features from recorded sound signals, we were then ready to begin the neural network construction phase of our project. First, we needed to collect a large set of voice data.

5.1 TRAINING DATA

We chose six words for use as training data for our network, each chosen based on the vocal characteristics present when they were pronounced. These words, suggested in [3], include 'cash', 'goodbye', 'this', 'one', 'close', and 'man.' Each word is characterized by distinct formants, and this will allow our network to differentiate individual speakers with maximal accuracy.

We chose four speakers, Brian, Jennifer, Xuening, and Kelly, to individually record 10 samples of each word at different times. With data from two males and two females, our networks should be more robust and able to distinguish between speakers of both sexes with equal accuracy. In addition, by recording each sample at separate times, we made sure that each voice sample was pronounced independently of the previous samples. We hypothesized that by doing this, we could make our network more tolerant to the changes in a person's voice that occur over a short course of time. A total of 240 initial sound samples were taken using Microsoft™ Sound Recorder. The first six utterances of each word from every person (144 training samples) were used as training data. The other four samples (96 testing samples) were used for validation and test purposes in our networks.

Once these sound samples had been obtained and preprocessed, we wrote a Matlab program to extract the features we chose as inputs for our networks and set the target values for each set of inputs; they were then written into a data file. We wanted to test several different combinations of our data and features, so we created a total of 32 data sets, each a different combination of the speakers and features extracted from each sample. The target values were encoded using the One-of-N encoding convention. The values were arranged so that each speaker had the same number of training samples, six per speaker, and the rest were used as validation and test data. This way, the network would not be biased toward any speaker. Finally, we randomized the training data so that the network would not train on a long sequence of any one speaker's signals consecutively. This again allows the network to be trained more evenly among the speakers. Initially we used four different types of inputs, as shown in Table 1, with six speaker combinations for each, as presented in Table 2.

Table 1. Types of Inputs

A. 12 Cepstral Coefficients with Hamming Filter (12 inputs)
B. 12 Cepstral Coefficients (12 inputs)
C. 12 LPC Coefficients with Hamming Filter (12 inputs)
D. 12 LPC Coefficients (12 inputs)

Table 2. Types of data files

1. two male speakers : (2 outputs, 72 training data, 48 testing/validation data)
2. two female speakers : (2 outputs, 72 training, 48 testing/validation)

3. a male and a female speaker : (2 outputs, 72 training, 48 testing/validation)
4. two male speakers and one female speaker : (3 outputs, 108 training, 72 testing/validation)
5. one male speaker and two female speakers : (3 outputs, 108 training, 72 testing/validation)
6. two male speakers and two female speakers : (4 outputs, 144 training, 96 testing/validation)

After creating these initial data sets, we decided to include pitch information in an attempt to improve the accuracy of our network. However, since the magnitude of the mean pitch value differed so much from that of our coefficient inputs, we had to normalize this value so that it was on the same order of magnitude as the other inputs. We divided the mean pitch value by 550 Hz, which was the upper limit of pitches that we detected from our sound samples. With this information, we created two additional data sets, as presented Table 3, each with the same six speaker combinations.

Table 3. Added Input Types

1. 12 Cepstral Coefficients with normalized mean pitch value (13 inputs)
2. 12 LPC Coefficients with normalized mean pitch value (13 inputs)

We created a total of thirty-six data files, each used to train a separate network. This diversity should give us an accurate assessment of our algorithm's ability to perform under different situations.

5.2 NETWORK STRUCTURES

The network that we decided to implement was a feed forward Multi-Layer Perceptron (MLP) network. We then used Netlab, a large collection of neural network functions programmed in Matlab, to construct the MLP networks to characterize our many data sets.

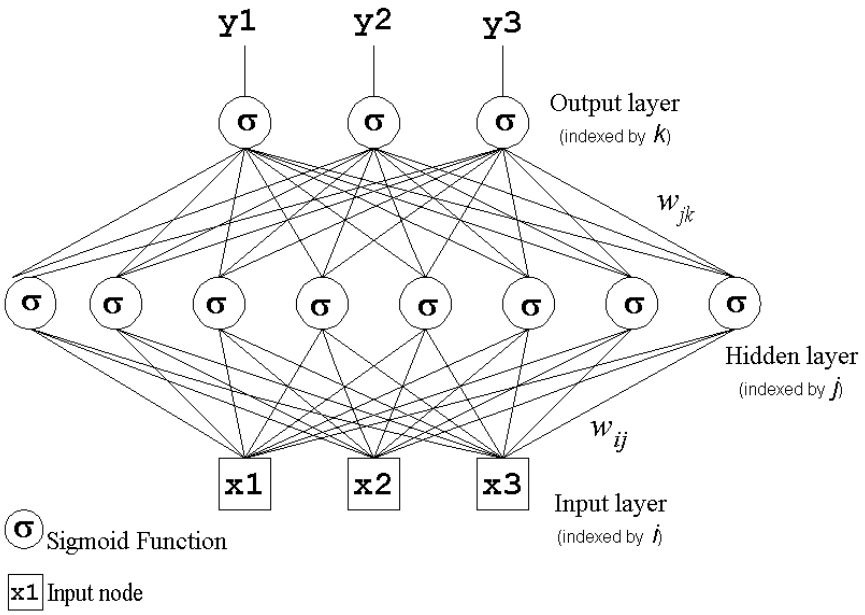

Figure 7. Multi-Layer Perceptron [8]

The MLP is an artificial neural network that can model non-linear functions by using non-linear sigmoid functions in its hidden layer. In fact, it has been proven that a MLP can model any arbitrary function using only three layers, given that it has enough inputs and hidden units [9]. This property makes the MLP a universal classifier/identifier and a perfect candidate for our speaker identification purposes.

Our MLP networks each contained only one hidden layer with hidden units ranging from 8 to 15, depending on the types of input data and the number of outputs. With each new training set, we initially trained with only 8 hidden units and adjusted depending on validation error values. The number of outputs of each network was the number of speakers that the network is designed to distinguish between. We trained networks to distinguish between 2, 3 and 4 different speakers. The number of epochs that the network trained on ranged from 100 to 200 epochs. Once again this was dependent on both the type and dimension of the input and output data used in each network.

In general, we found that using more hidden units did not always give the optimal result. We also learned that changing the number of hidden units had much more impact than changing the number of epochs. Generally, as the number of outputs (i.e. the number of speakers to identify) increased,

the network required more hidden units (12-15) as well as more training epochs. Cepstral coefficients also required more hidden units (12) to produce the best results.

In all, we created a total of 32 networks using the data files constructed during the data training stages. Four data files, containing Cepstral Coefficients with Hamming Filter, were not used because we discovered that these inputs gave unreasonably poor results.

5.3 TESTING DATA

Once our networks were properly trained, we tested our network with two types of data. We wanted to test our ability to perform both text-dependent and text-independent speaker recognition. Thus, we first used text-dependent testing data, and then compared these results with those obtained using text-independent data.

First, we tested the network using 4 utterances/word/speaker for each of the words used in training. Ideally, if a speaker utters a word used in network training, the network should be able to recognize that speaker with high probability. We also decided to test our networks with text-independent data, or data extracted from words not used for training. For this purpose, we chose four words, 'Austin', 'Ghosh', 'that', and 'ten,' each markedly different from the six words used in network training. Each word was recorded five times by three speakers, Brian, Jennifer, and Xuening. We then preprocessed these samples in the same manner as before and again created a data file. We decided to test these unknown words on our best performing networks, the one trained with LPC coefficients with mean pitch value network.

6.0 RESULTS

6.1 TEXT-DEPENDENT

Using text dependent testing data, we obtained very good results in our network. Figures 8 – 11, give several examles of the confusion matrices that we obtained from our networks.

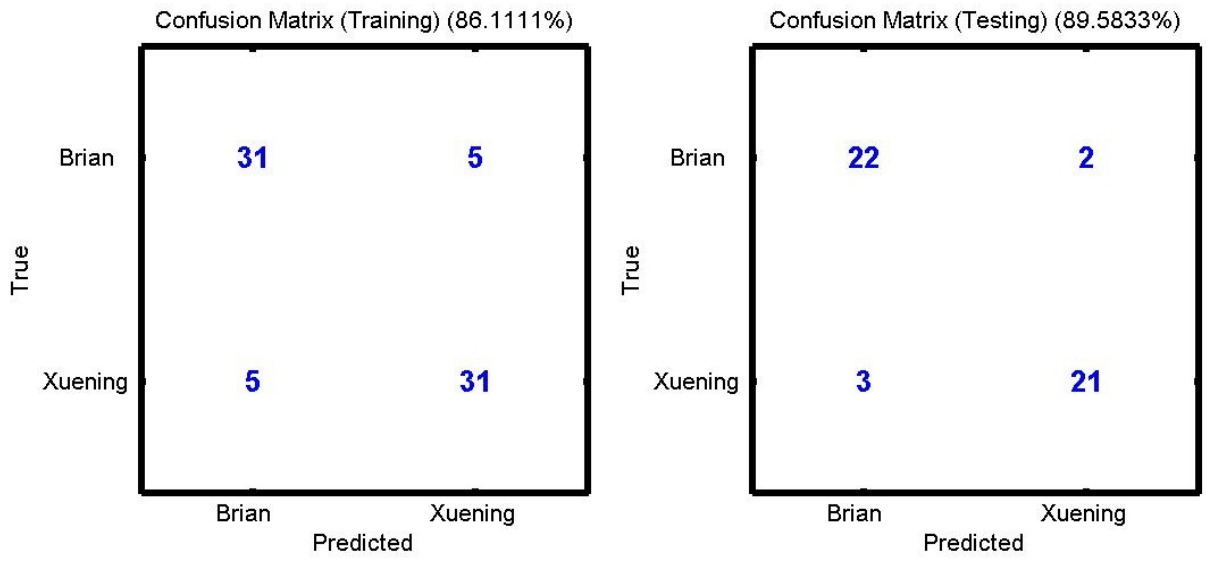

Figure 8. Confusion Matrices for Male-Male Speaker Recognition

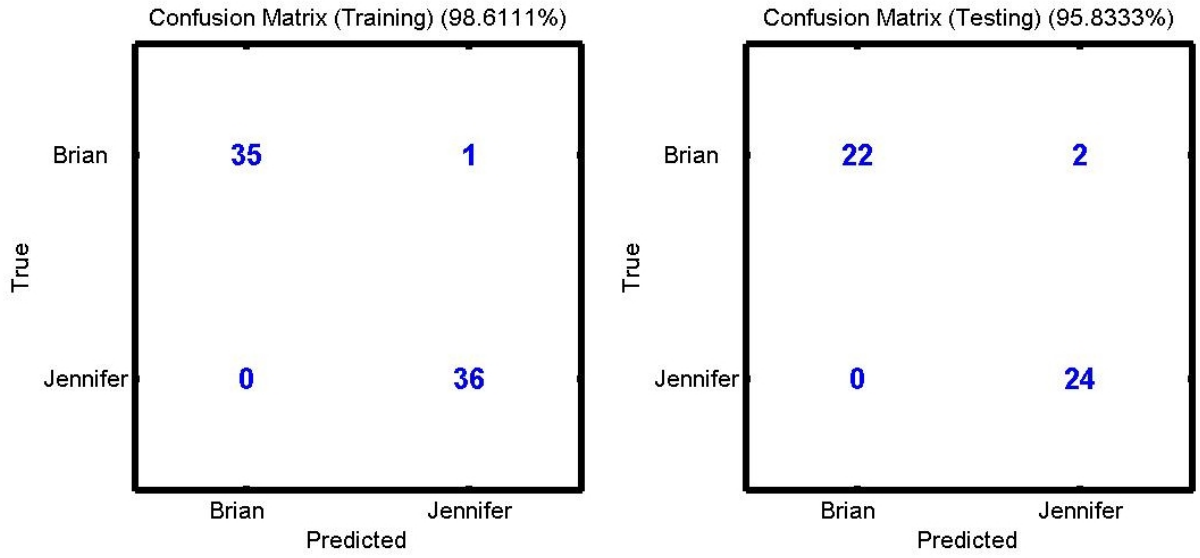

Figure 9. Confusion Matrices for Male-Female Speaker Recognition

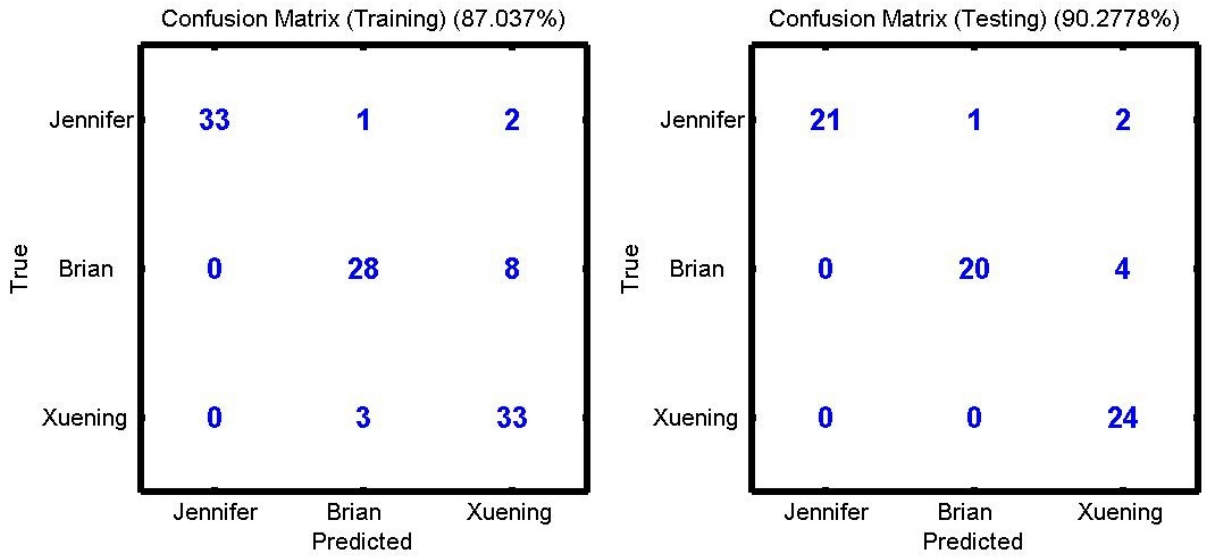

Figure 10. Confusion Matrices for Recognition among Three Speakers

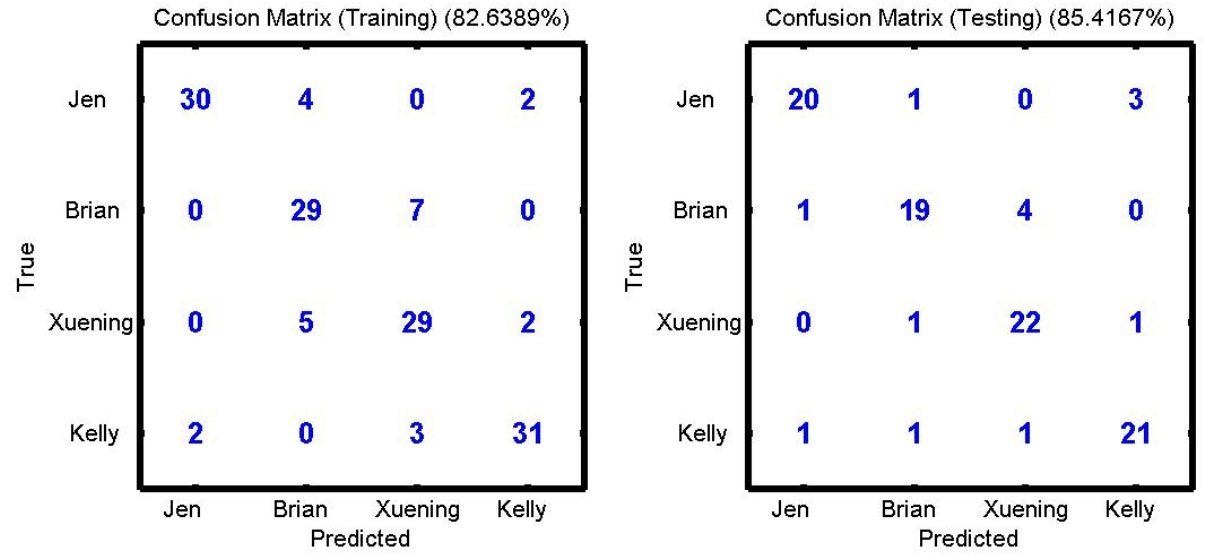

Figure 11. Confusion Matrices for Recognition among Four Speakers

The best results obtained was when distinguishing between Brain and Jennifer using LPC coefficients with mean pitch value as inputs into the MLP network. Overall, the trend of our networks when tested with text-dependent data seems to be that as the number of outputs of the network is increased, the percentage of accurate predictions decreases. Another trend that we noticed was that once we added the pitch information to the network, the average percentage of correct predictions improved dramatically, especially when classifying among a large number of

speakers. The percentage of correct predictions also seemed to have decayed much less as the number of speakers increased once the pitch information was added to the network.

Table 4. Average Percentage Correct Predictions Using Known Words (Text-Dependent)

(6 words, 10 utterances/word)

	BX	BJ	JK	BXJ	BJK	BXJK
Cepstral Coeffs with Hamming Filt.	NA	NA	76.39%	81.94%	NA	NA
Cepstral Coefficents	87.50%	86.81%	62.5%	76.85%	81.94%	48.96%
LPC Coeffs with Hamming Filt.	94.83%	92.19%	88.58%	68.52%	47.22%	48.99%
LPC Coefficients	91.65%	88.19%	89.58%	77.08%	81.94%	57.29%
Cepstral Coeffs with Pitch estimate	85.4%	93.75%	64.4%	65.28%	68.05%	56.25%
LPC Coeffs with Pitch estimate	91.67%	95.83%	89.6%	87.5%	87.5%	86.45%

B—Brian X—Xuening J—Jennifer K—Kelly

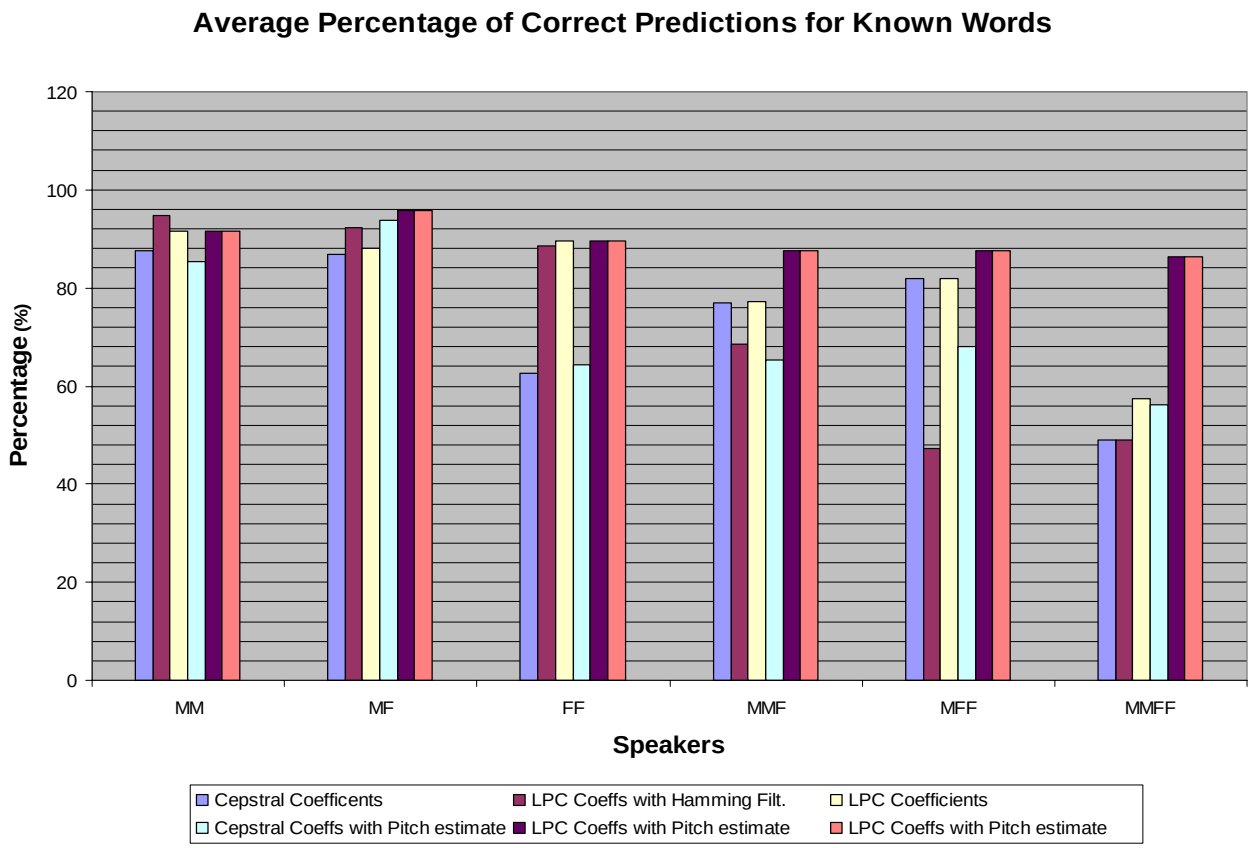

Figure 11. Text-Dependent Results

We then compiled a table of word statistics to show us how often each word was predicted correctly. We observed that words such as 'man' and 'goodbye' gave very good results. This is most likely due to the fact that the sounds of these words contains no unvoiced speech, so the LPC and Cepstral coefficients can model them very accurately. Words such as 'close' and 'cash' didn't perform so well due to their unvoiced components, the s- sound in 'close' and the -sh sound in 'cash'. However, despite these drawbacks, the average correct predictions for each person was still very good.

Table 5. Correct Predictions—Word Statistics

	Jennifer (%)	Brian (%)	Xuening (%)
6 Known Words, 4 Utterances/Word			

Cash	80	60	100
Goodbye	100	100	80
This	80	100	80
One	100	100	80
Man	100	100	100
Close	100	20	100
AVG:	93.33	80	90

6.2 Text-Independent

Upon acquiring excellent results for the text dependent speaker identification scenario, we felt that we naturally should explore the next step in the broad area of Automatic Speaker Recognition, which is text independent speaker identification. The results obtained for text independent testing data did not show the same degree of accuracy as that of the text dependent testing, which was somewhat expected. However, although the results we poor, they still showed some promising aspects. Words that were some what similar to the training data phonetically showed a lot better results than those who did not resemble the training data. This shows that our network can work for text independent speaker identification if was had a more extensive list of training vocabulary that can model nearly all formants in the English language. On a side note, surprisingly, Brian's voice, who performed the worst in text dependent testing, actually performed best for text independent testing. This could be attributed to recording technique and that his voice stayed fairly the same over time. If we had a laboratory setting to record our samples, the results for text independent speaker identification could possibly have increased more.

Table 6. Text Independent Speaker Recognition Results

	Jennifer (%)	**Brian (%)**	**Xuening (%)**
4 Unknown Words, 5 Utterances/Word			
Austin	60	40	0
Ghosh	0	40	40
that	60	100	0
ten	0	100	20

AVG:	30	70	15

7.0 CONCLUSION

After all the extensive data acquisition, network training, and network testing, we have drawn out several conclusions for our project. First of all, LPC coefficients performs much better than Cepstral coefficients in the area of speaker identification. Second, by adding in the mean pitch information of a person, the network is able to identify the speaker more accurately. Pitch information also makes the network more robust as more speakers are added into the system. Finally, our network obviously worked very well for text dependent speaker identification; however, it also showed very promising results for text independent speaker identification. If a more extensive list of training data containing a diverse group of formant information is obtained, then using our algorithm, our network can definitely identify the speakers independent what he or she says. We started our project merely wanting to identify the gender a speaker, but as we researched more into the topic, we found that our initial goal was very simple to achieve, so we decided to tackle the challenging problem of text dependent speaker identification. As we obtained fairly good results for a text dependent speaker recognition system, we extend even more and delve into the problem of text independent speaker recognition. Although our results were limited by the amount of training data we had, we still found the results to show potential. However, the problem of automatic speaker/voice recognition is very broad field with many problems yet to be solved. Further development on our project includes training the network to recognize unauthorized speakers; however, this is restricted again by the amount of training data that you obtain. We did not try to solve this problem because we simply could not collect enough data to characterize unauthorized speakers. A more advanced topic based upon our project would be to train the network to recognize and understand individual words and phrases. This can be done using Hidden Markov Models, but this is much too advanced for our group to research given the amount of time that we had.

REFERENCES

[1] S. Parveen, A. Qadeer, and P. Green, “Speaker Recognition with Recurrent Neural Networks,” Speech and Hearing Research Group, Dept. of CS, University of Sheffield.

[2] S. Furui, *Digital Speech Processing, Synthesis, and Recognition*, Marcel Dekker, Inc. New York, 2001.

[3] K. Kuah, M. Bodruzzaman, and S. Zein-Sabatto, “A Neural Network-Based Text Independent Voice Recognition System,” *Proc. of the 1994 IEEE Southeast Conference (Southeastcon ‘94)*, April 1994.

[4] B. Bogert, M. Healy, and J. Tukey, "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-autocovariance, CrossCepstrum, and Saphe Cracking," *Time Series Analysis*, John Wiley and Sons, New York, 1963, pgs 209-243.

[5] R. Klevans, R. Rodman, *Voice Recognition*, Artech House, Inc, 1997.

[6] J. Koolwaaij, “Speech Processing,” http://www.ispeak.nl/start.html?url=http://www.ispeak.nl/prfhtm/node12.html&n=1&ref=http://www.google.com/search (current 5 May 2004).

[7] G. Middelton, “Pitch Detection Algorithms,” http://cnx.rice.edu/content/m11714/latest/ (current 5 May 2004).

[8] “Mulitlayer Perceptrons,” http://homepages.gold.ac.uk/nikolaev/311multi.htm (current 5 May, 2004).

[9] Bishop, *Neural Networks for Pattern Recognition*,