Female Voice Recognition using Artificial Neural Networks and MATLAB Voicebox Toolbox

Stanley Glenn E. Brucal[1], Aaron Don M. Africa[1] and Elmer P. Dadios[2]

[1]*Department of Electronics and Communications Engineering, Gokongwei College of Engineering, De La Salle University Manila, 2401 Taft Avenue Manila, Philippines 1004.*

[2]*Department of Manufacturing Engineering and Management, Gokongwei College of Engineering, De La Salle University Manila, 2401 Taft Avenue Manila, Philippines 1004.*

sgebrucal@gmail.com

***Abstract*—Voice and speaker recognition performances are measured based on the accuracy, speed and robustness. These three key performance indicators are primarily dependent on voice feature extraction method and voice recognition algorithm used. This paper aims to discuss various researches in speech recognition that has yielded high accuracy rates of 95% and above. The extracted MFCCs from MATLAB Voicebox toolbox were used as inputs to the multilayer Artificial Neural Networks (ANN) for female voice recognition algorithm. This study explored the recognition performance of the neural networks using variable number of hidden neurons and layers, and determine the architecture that would provide the optimum performance in terms of high recognition rate. MATLAB simulation resulted to a training and testing recognition rate of 100.00% when using 3-hidden-layer neural network from speech samples of a single-speaker, and highest training recognition rate of 98.11% and testing recognition rate of 87.20% when using 4-hidden-layer neural network from speech samples of several speakers. When tested with homonyms, the best recognition rate was 75.00% from a 3-hidden-layer neural network trained from a single-speaker, and 81.91% from a 4-hidden-layer neural network trained from multiple speakers. The deviation in recognition rates were primarily attributed to the variations made in the number of input neurons, hidden layers, and neurons of the speech recognition neural network.**

***Index Terms*—Voice Feature; Mel Frequency Cepstral Coefficient; Artificial Neural Network; Voice Recognition; Speech Recognition.**

I. Introduction

An important application of audio processing is voice and speaker recognition used in voice-to-text searching, human-to-computer interaction, and autonomous robots. Various related researches have been published -- each differing on the algorithms applied. These include Dynamic Synapse-based Neural Networks [1] for classification of temporal patterns found in speech to perform speaker verification and speaker recognition at normal noise levels, combined Genetic Algorithm (GA) and Fisher's Linear Discriminant Ratio (FLDR) [2] for real-time spoken word recognition which aids potential users reduce large training sessions, Hidden Markov Model (HMM) [3] used in domestic application, and Probabilistic Neural Network [4] for speech recognition with shorter processing time. A comparative study [5] of the efficiency between a hybrid approach of Linear Predictive Coding (LPC) and artificial neural networks, and combination of Wavelet Packet Decomposition (WPD) and artificial neural networks (ANN) has been conducted for recognizing speaker independent spoken isolated words. From this study, WPD combined with ANN, resulted to a higher recognition accuracy than LPC combined with ANN. A method based on Fuzzy Neural Network (FNN) combined with 5-layer fuzzy logic [6] was proposed to improve the weaknesses of Particle Swarm Optimization - Forward Neural Network (PSO-FNN) and Back Propagation Forward Neural Network (BP-FNN). It was capable of recognizing and eliminating environment noises from the sample using firefly algorithm.

The high detection rates of speech recognition algorithms introduced in various studies have used Mel Frequency Cepstral Coefficient (MFCC) as voice feature representation of speech signal. Among these were Dynamic Time Warping [7] which uses both Linear Prediction Coefficients (LPC) and MFCC to achieve a detection rate of 90%; Voice Activity Detection (VAD) based on Radial Basis Function Neural Network (RBF-NN) and Continuous Wavelet Transform (CWT) [8], where the former was used to detect speech/non-speech signal, and the latter for identifying start- and endpoints of speech, that recorded its best performance of 95.72%; Hybrid Intelligent System based on Genetic Algorithms and Kohonen Self-Organizing Map (SOM) [9] for recognition of present phonemes in a word of a Spanish language, that recorded an accuracy rate of 92.3%; combined neural networks and fuzzy logic [10] for real-time voice request to the computer to guide a distributed robot, which attained its highest recognition rate of 98.7%; and machine learning technique, Optimum-Path Forest (OPF) [11] for voice-based robot interface with 98.94% accuracy rate and lower computational times than Support Vector Machines, Neural Networks and Bayesian classifier.

The necessity of achieving a high speech recognition rate was illustrated and implemented in the computerized system for Breast Self-Examination – Multimedia Training System (BSE-MTS) [12, 13]. The multimedia system that they created ANN) while 97.50% accuracy when using GA. There were 100 Hilgaynon test words in [12] from a training set of 79 words from English and Hiligaynon language, while 200 Hiligaynon test words in [13]. Highest speech recognition rate using ANN with distributed features reduction method was at 88%, while when using GA technique the average recognition rate was 97.50%. With these implementations as benchmark, this paper aims to identify optimum number of hidden layers in the neural network that would yield the highest recognition rate from a finite number of voice samples from four (4) female with different speech tonal quality. The results of this paper can be used to improve the performance of interactive audio-visual breast self-examination system, specifically on the domain of speech recognition. Similarly, the feasibility of implementing a multilayer feed-forward network ANN for voice and speech recognition can be presented based on the system's training and testing outcomes.

II. Review of Related Literature

MFCCs are coefficients used to represent characteristic of an audio signal computed from a short-term power spectrum using Discrete Cosine Transform (DCT) of a log-triangular weighting function of filter outputs, based on a nonlinear mel-scale of frequency. The scale is non-linear, because it has two types of filter spacing for each frequency range: linear for frequencies below 1kHz, and logarithmic for above 1kHz. Important characteristics of phonetic in speech can be found at frequencies below 1 kHz - similar with human hearing perceptions [14].

A. *Mel Frequency Ceptral Coefficients Computation*

Figure 1 illustrates the step-by-step process in extracting the voice feature using MFCC. This method applies for other voice recognition studies that uses MFCCs as input for voice feature vector, except for [15], that uses Inverse Discrete Fourier Transform (IDFT), instead of DCT to compress the feature vectors. The pre-processing stage of [3] and [15] differ in methods, but have the same output; that is, they both produce framed data. In [16], pre-emphasis was not discussed, except for its inclusion of filters with different coefficients and poles.

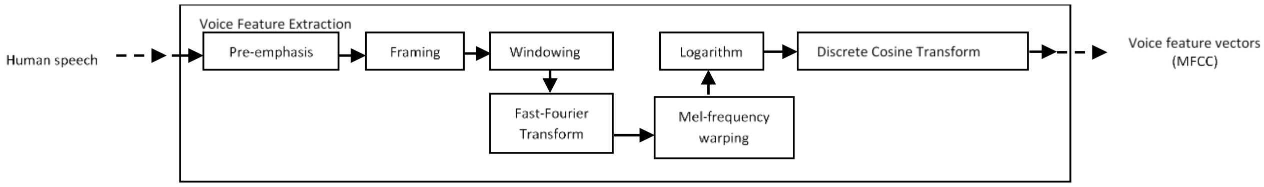


Figure 1: MFCC voice feature extraction process [16]

Framing breaks down the audio clip into smaller segments with a typical frame length value of 20-30ms [17]. A shorter frame length will not give enough sample points, while a longer frame length will provide too many samples that would be difficult to apply statistical treatment in computing for its characteristics. After breaking down the entire audio clip into smaller frames (now with defined number of samples), windowing follows which aims to prepare each frame for power spectrum calculation. In this process, frame step for each window has to defined, with 10ms as reference value [18]. The end-results would be overlapping frames, known as hamming windows with frame steps shorter than the frame length. Each hamming window will be subjected to Discrete Fourier Transform to derive its periodogram-based spectral value, taken from Equation (1) [18], where *Si(k)* is the DFT of the frame, *$S_i(n)$* is the speech frame *h(*n) is the hamming window analysis, and *K* is the DFT length.

$$S_i(k) = \sum_{n=1}^{N} S_i(n)h(n)e^{-j2\Pi kn/N} \quad 1 \leq k \leq K \tag{1}$$

Since mel-frequency works in the principle of human perception of human sound, that is formants are much perceived at lower frequencies, triangular bandpass filters (or hamming filters) are implemented. As discussed in [14] and [19], there will be a linear bandwidth for windows positioned below 1 kHz, and bandwidth increasing exponentially for windows positioned above 1 kHz. From the triangular windows will the Mel-spaced filterbank, ranging from 20-46 (26 is standard) [18], be derived from. The filterbank energies, can be computed by taking the product of each filterbank with the power spectrum derived from Equation (2).

$$P_i(k) = \frac{1}{N}\ |S_i(k)|^2 \tag{2}$$

To compute for the cepstral coefficients of each filterbank energies, Equation (3) is used [19]. *MF(r)* is the mel-frequency spectrum, while *R* refers to the number of mel-filter used for each triangular weighted function. The resulting values are known as the MFCC and will serve as feature vectors. Since the formants are identified at lower frequencies, only 12 to 13 (out of 26) numbers from each frame are used [9, 18, 19, 20].

$$MFCC = \frac{1}{R}\sum_{r=1}^{R} \log(MF[r])\cos[\frac{2\Pi}{R}(r+\frac{1}{2})m] \tag{3}$$

B. *Artificial Neural Network*

Neural network is based on the theory of how human brain works with its basic unit, neurons – the nerve impulse transmitters. Neurons functions by taking the summation of the inputs and their weights. Since these weights contain the information, there are different algorithms that can be used for its training. For neurons to learn, modeling schemes such as perceptron, adaptive linear, sigmoidal neuron, or Hebb neuron models may be used. Given a neural network of several input perceptrons, its summation is mathematically defined in Equation 4 [21], where y is the perceptron summation, *n* is the number of input signals, *x_j* is input signal, and *w_j* is the weight.

$$y = \sum_{j=1}^{n} w_j x_j \tag{4}$$

The architecture of neural networks always has the input and output layers. Input layer is the where the outside world is communicating with the network and where data is presented to train the network. Output layer is where the pattern is presented to the outside world. Between these two layers are hidden layers where the neurons are interconnected – known as hidden nodes or hidden neurons – thus transmits signals. The behavior of the output layer depends on its defined activation function such as linear activation, piecewise linear, tangent, hyberbolic, sigmoidal, or threshold function. Multilayer feed-forward networks come with several types of learning algorithms such as error backpropagation, backpropagation algorithm with momentum term, variable-metric, and Levenberg-Marquardt.

III. Methodology

The MFCC feature vectors will be computed using MATLAB software. Since there is no defined MFCC function in MATLAB, the existing pre-defined MATLAB functions used to compute for MFCC [22] is used in this study due to its small vector size. Other MATLAB functions in computing MFCC [23] can also be used if variable frame length, frame step, and numbers of MFCC coefficients are desired. A comparative study between Auditory and Voicebox toolboxes can be found in [24].

The *melcepst* function implements a mel-cepstrum front-end for a recognizer [25]. By default, it provides 12 mel-frequency cepstrum coefficients taken from 256 samples, or may also include the log energy of the 0th spectral coefficient, which is normally discarded, since it only represents the average value of the acoustic vectors [17, 26, 27].

A. *Speech Recognition using Artificial Neural Networks*

The neural network training and testing start with the capturing of audio clips from females uttering the pre-defined motion command words. Audacity® was used to record each .wav file with a duration of 1ms at a sampling rate of 11,025kHz. As a start-up, number of neurons will be set to a value that is between the number of the input and output elements [28], with one hidden layer until desired validation performance that can give an error rate of 0.001% as measured using Mean Squared Error (MSE). After training the network, testing was done out of the remaining speech samples from the speakers. These test data are composed of voice utterances of the pre-defined motion command words, homonyms of these words, and error words. Figure 2 illustrates the ANN training and testing process flow.

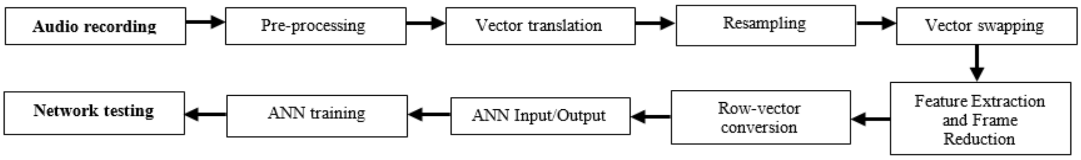
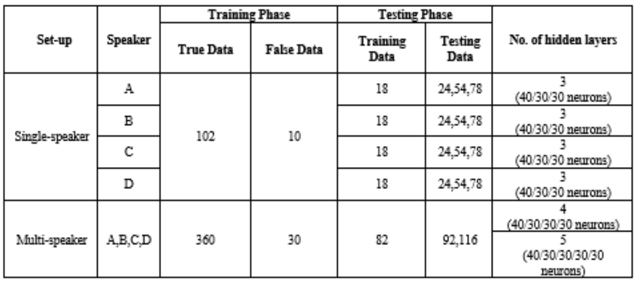


Figure 2: ANN training and testing process

The voice recognition system is basically composed of training and testing parts. For the training phase, there will be five (5) set-ups from four (4) female speakers of age 20-22. Each speaker is to feed the network with 17 utterances of command words to be recognized: “up’, “down”, “left” “right”, “center”, and “push” False words are utterances of words that are not in the pre-set command words and were recorded from other female speakers. The multiple speaker set-up is composed of audio clips from speakers A, B, C, and D; each with 15 utterances of the six words to be recognized. The characteristic of false words is the same as with the single-speaker, with no repeating words. Levenberg-Marquardt training will be used for the ANN. Table 1 shows the simulation scenarios for input neurons of 240 and 120; that is from 12 MFCC coefficients with 20 frames per coefficient. The frame reduction technique of 240 to 120 will be done by removing the last 10 frames of each MFCC; that is, only considering only the 1st 10 frames. This was done, since most of the extracted MFCC voice features has frame value of zero (0) from the 15th to 20th frame. There is no exact method of identifying the optimum number of hidden layers and neurons [29, 30, 31], hence defining the appropriate number is still a challenge.

Table 1
No. of Voice Samples Used for the Network Training



Set-up	Speaker	Training Phase		Testing Phase		No. of hidden layers
		True Data	False Data	Training Data	Testing Data	
Single-speaker	A	102	10	18	24,54,78	3 (40/30/30 neurons)
	B			18	24,54,78	3 (40/30/30 neurons)
	C			18	24,54,78	3 (40/30/30 neurons)
	D			18	24,54,78	3 (40/30/30 neurons)
Multi-speaker	A,B,C,D	360	30	82	92,116	4 (40/30/30/30 neurons)
						5 (40/30/30/30/30 neurons)

There are 3 cases to be performed for the single-speaker network testing. These are: (1) with the use of 42 audio samples from the 18 true inputs used during training, another set of 18 true inputs, and 6 false inputs; (2) with the use of 72 samples from 18 true inputs used during training, another set of 18 true inputs, and 36 false inputs; and lastly, (3) with the use of 96 audio samples from 18 train true inputs, another set of 30 true inputs, and 48 false inputs (homonyms). The testing of neural network trained from multiple speakers has two (2) cases to be performed: (1) with the use of 164 speech samples from 72 true inputs used in training from the four speakers, another set of 72 true inputs from the same speakers, and 20 false inputs, and lastly, (2) with the use of 188 speech samples with the use of 72 true inputs used during training from the four speakers, another set of 96 speech samples from the same speakers, and 20 false inputs (homonyms).

IV. Simulation Results

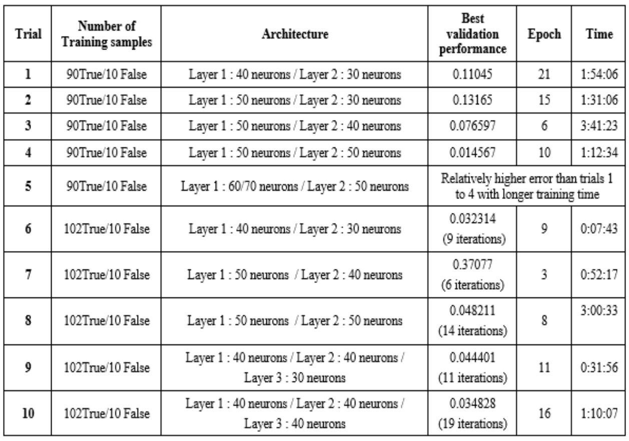
A. MATLAB Voicebox Toolbox

The Voicebox toolbox containing the *melcepst* routine function was used in the voice feature extraction. Considering the uniform vector size needed as inputs to the neural network, the length of the audio clip recordings for the six words taken from different speakers, were all limited to 1ms. The number of coefficients is set at 12, since the 0th coefficient will not be discarded. Using the *melcepst* function, each audio clip is fixed at matrix size of 1x 240, while the MFCC matrix size is at 20 x 12 (20 frames with 12 MFCCs), computed from a sampling frequency of 11,025kHz, frame samples of 330, and 15ms frame shift. Converting the MFCC matrix to a column vector will result to a size of 240 elements, relatively higher than that of [32] with only 160 real values.

B. Artificial Neural Networks

The *nntool* function with network type of feed-forward back propagation of MATLAB is used in determining the appropriate training algorithm, number of hidden layers and neurons. The resulting architecture for single-speaker ANN is three (3) hidden layers: 40 neurons and linear transfer function for the first layer, 30 neurons each for the second, and third layer with log-sigmoid transfer function, and the last layer as the output layer with log-sigmoid transfer function. For network using speech samples of multiple speakers, a four and five hidden layers feed-forward neural network will be used. Table 2 shows the performance of network architecture using other number of hidden layers at different numbers of hidden neurons. The number of hidden layers and transfer functions adopted in this study is the same as with [33], but differs in the number of neurons used.

Table 2
Performances of Other Network Architecture



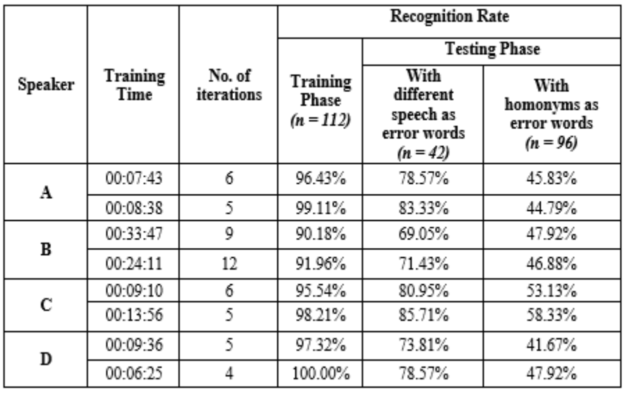
Trial	Number of Training samples	Architecture	Best validation performance	Epoch	Time
1	90True/10 False	Layer 1 : 40 neurons / Layer 2 : 30 neurons	0.11045	21	1:54:06
2	90True/10 False	Layer 1 : 50 neurons / Layer 2 : 30 neurons	0.13165	15	1:31:06
3	90True/10 False	Layer 1 : 50 neurons / Layer 2 : 40 neurons	0.076597	6	3:41:23
4	90True/10 False	Layer 1 : 50 neurons / Layer 2 : 50 neurons	0.014567	10	1:12:34
5	90True/10 False	Layer 1 : 60/70 neurons / Layer 2 : 50 neurons	Relatively higher error than trials 1 to 4 with longer training time		
6	102True/10 False	Layer 1 : 40 neurons / Layer 2 : 30 neurons	0.032314 (9 iterations)	9	0:07:43
7	102True/10 False	Layer 1 : 50 neurons / Layer 2 : 40 neurons	0.37077 (6 iterations)	3	0:52:17
8	102True/10 False	Layer 1 : 50 neurons / Layer 2 : 50 neurons	0.048211 (14 iterations)	8	3:00:33
9	102True/10 False	Layer 1 : 40 neurons / Layer 2 : 40 neurons / Layer 3 : 30 neurons	0.044401 (11 iterations)	11	0:31:56
10	102True/10 False	Layer 1 : 40 neurons / Layer 2 : 40 neurons / Layer 3 : 40 neurons	0.034828 (19 iterations)	16	1:10:07

a. Single-Speaker Training

The network is initially fed with 112 audio clips, of fixed length of 1ms, recorded from a single-speaker. These are combination of the six words to be recognized, each with 17 samples, and 10 samples of error words. During recording, the speaker is tasked pronounce each word several times with some pause for each utterance. Four speakers, with different voice characteristics (i.e.: timbre), were used in the same network architecture. Network training was run twice without changing the network parameters. The training resulted a highest recognition rate was 100% and least was 90.18%, while during testing, the highest was at 85.71% and lowest was 69.05%. To further examine the reliability of the network in recognizing words that were not originally part of the training and testing set, homonym words with additional correct words (66 correct words, 6 error words and 24 homonym words) were used as error words. This arrangement resulted to a poor recognition rate of only 58.33% as the highest.

From the several trainings done for each set of audio clips (from each speaker), it was in Speaker B that took a longer time of training - of approximately 34 minutes on the first training and 24 minutes on the second training - and has relatively lower recognition rates for both the training and testing at 91.96% and 71.43%, respectively on the second training. Retraining the network contributed to a significant improvement in the recognition ratio, with shorter training time for Speakers B and D (9 minutes faster for speaker B; 3 minutes faster for Speaker D) and lesser number of iterations for Speakers A, C, and D (by 1 iteration), but not in Speaker B (increase from 9 to 12 iterations). Comparing the recognition rates for both training and testing, it was from the speech samples of Speaker A and C that consistently yielded high recognition rate of 85.00% and above (99.11% during training and 85.71% during testing for Speaker A; 98.21% during training and 85.71% during testing for Speaker B). It is noticeable, however, that it was from the same speaker (A) that yielded the lowest recognition rate when tested with homonyms at 44.79% on the second training. Among the four speakers, it's Speaker B with a hoarse voice quality. The rest of the speakers have modal voice quality, each of different loudness and speaking speed levels. Table 3 provides a summary of the neural network recognition rates during training and testing.

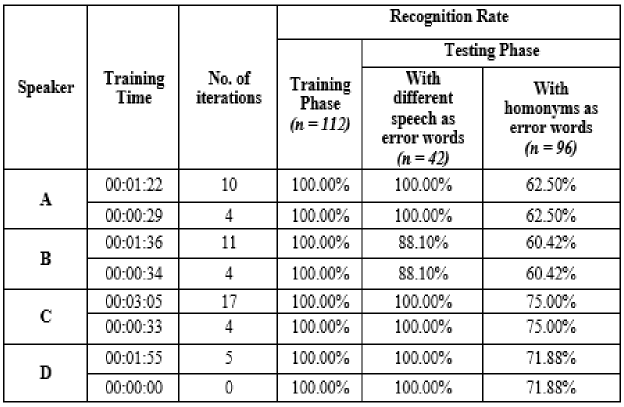
Table 3
Recognition Rates for Training and Testing of the Neural Network from Single Speaker Using 240 Input Neurons



Speaker	Training Time	No. of iterations	Recognition Rate		
			Training Phase *(n = 112)*	Testing Phase	
				With different speech as error words *(n = 42)*	With homonyms as error words *(n = 96)*
A	00:07:43	6	96.43%	78.57%	45.83%
	00:08:38	5	99.11%	83.33%	44.79%
B	00:33:47	9	90.18%	69.05%	47.92%
	00:24:11	12	91.96%	71.43%	46.88%
C	00:09:10	6	95.54%	80.95%	53.13%
	00:13:56	5	98.21%	85.71%	58.33%
D	00:09:36	5	97.32%	73.81%	41.67%
	00:06:25	4	100.00%	78.57%	47.92%

Since *melcepst* function by default has 12 MFCCs with 20 frames per coefficient, there are 240 input neurons for each speech sample. The size of hidden neurons affects the training and testing performance of neural network, thus number of frames reduction by 50% for each speech sample was investigated. This means that from 240 input neurons, there would be 120 input neurons to be fed to the neural network for training. It was done by removing the least significant frames of each MFCC – the last 10 frames. As provided in Table 4, there was a significant improvement in training time, number of iterations, and recognition rate when fewer inputs (120 instead of 240 neurons) was used. With same network architecture of 120 input neurons for single speaker, a 100% recognition rate was achieved during training in less than 3 minutes, as compared to the neural network with 240 input neurons, where it took more than 6 minutes to achieve a training recognition rate of 100%. The testing phase of neural network with 120 input neurons also yielded a very high recognition rate of 88% to 100%; notable 100% recognition rate from 3 out of 4 speakers. There was also a significant improvement in the network performance when tested with homonyms at a recognition rate ranging from 60.42% to 75.00%, as compared with neural network trained using 240 input neurons with recognition rate, when tested with homonyms, ranging from 41.67% to 58.33%. When the network was retrained, no significant improvement was recorded during testing. It is also interesting to know that best network performance was achieved using the *tansig* transfer functions for all the neural network layers, as compared with the *purelin* (input layer) and *logsig* (hidden and output layers) transfer functions used in network training with 240 input neurons.

Table 4
Recognition Rates for Training and Testing of the Neural Network from Single Speaker Using 120 Input Neurons

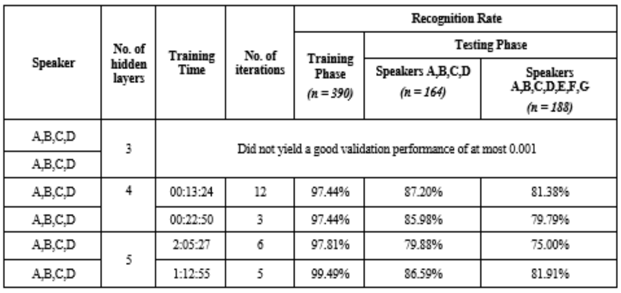


Speaker	Training Time	No. of iterations	Recognition Rate		
			Training Phase *(n = 112)*	Testing Phase	
				With different speech as error words *(n = 42)*	With homonyms as error words *(n = 96)*
A	00:01:22	10	100.00%	100.00%	62.50%
	00:00:29	4	100.00%	100.00%	62.50%
B	00:01:36	11	100.00%	88.10%	60.42%
	00:00:34	4	100.00%	88.10%	60.42%
C	00:03:05	17	100.00%	100.00%	75.00%
	00:00:33	4	100.00%	100.00%	75.00%
D	00:01:55	5	100.00%	100.00%	71.88%
	00:00:00	0	100.00%	100.00%	71.88%

b. *Multi-Speaker Training*

Using different architectures (i.e.: no. of hidden layers), audio recordings taken from Speakers A, B, C, and D, were fed to the neural network for training. 15 audio clips for each word from each speaker, and 30 error words (total of 360 true and 30 false inputs) were used as training data. Given the number of samples, the training time is longer as compared with the single-speaker. Two ways of network testing were performed: (1) using the voices of speakers used for the training, and (2) using voices of additional 3 speakers. From the results summarized in Table 5, the training resulted to a high recognition rate of 99.49% (on the second training) when implemented with five hidden layers than with only four hidden layers with recognition rate of 97.44%. The testing of the network performance was divided into two cases: (1) 12 samples of each word used during training, another set of 12 samples of each word for testing, and additional 20 error words (total of 72 train data, 72 test data, and 20 error words); and, (2) 12 samples of each word used during training, another set of 16 samples of each word for testing, and additional 20 error words. The 1st case has speech samples taken from Speakers A, B, C, and D – the same set of speakers that were used in the train data, while the 2nd case care combination of speech samples from Speakers A, B, C, and D, with the inclusion of test data from another set of speakers E, F, and G. The testing using the same set of speakers resulted likewise to a relatively higher recognition rate of 86.59% for a five-layer neural network as compared with 85.98% when using a four-layer neural network. Testing the network with speech samples from other speakers (Speaker E, F, and G) resulted to a relatively lower recognition rate of 81.91% for a five-layer neural network and 79.79% recognition rate when implemented in a four-layer neural network. As with the training and testing of a neural network using speech samples from a single-speaker, retraining the network resulted in a decrease in the training time and number of iterations.

Table 5
Recognition Rates for Trainig and Testing of the Neural Network from Multi-Speaker



Speaker	No. of hidden layers	Training Time	No. of iterations	Recognition Rate		
				Training Phase *(n = 390)*	Testing Phase	
					Speakers A,B,C,D *(n = 164)*	Speakers A,B,C,D,E,F,G *(n = 188)*
A,B,C,D	3	Did not yield a good validation performance of at most 0.001				
A,B,C,D						
A,B,C,D	4	00:13:24	12	97.44%	87.20%	81.38%
A,B,C,D		00:22:50	3	97.44%	85.98%	79.79%
A,B,C,D	5	2:05:27	6	97.81%	79.88%	75.00%
A,B,C,D		1:12:55	5	99.49%	86.59%	81.91%

Neural network performed better in terms of recognition rate from training that uses several speakers. However, the number of hidden layers (i.e. 40/30/30) used for single-speaker training did not yield a favorable result for multi-speaker training. Instead, the number of hidden layers were increased to 4 (i.e: 40/30/30/30 neurons) and 5 (i.e.: 40/30/30/30/30 neurons) with linear transfer function for the first layer and log-sigmoid transfer function for the succeeding up to the output layers - the same learning algorithm that was used with single-speaker network training. A network with higher number of hidden layers has better recognition rate during testing in audio clips taken from speakers not originally part of the training. Increasing the number of hidden layers from 4 to 5 yielded minimal improvement in the training recognition rate from 97.44% to 99.49% on during re-training. The improvement on the recognition rate, however, had its drawback in the training time, since from 22:50 minutes for a 4-hidden layer network, it went to 1:12 hours for a 5-hidden layer network. These durations were taken from the network performance on its 2nd round of training.

V. CONCLUSION

The principle of mel-frequency cepstral coefficients was based on the non-liner response of human hearing, where human ears perceive sound information at lower frequencies. With the smaller vector size of *melcepst* function, MFCCs were computed to extract voice feature using MATLAB VOICEBOX toolbox. The number of cepstral coefficients used was 12, removing the higher coefficients and the 0th coefficient, with 20 mel-frames per coefficient. The three-hidden layer neural network with 120 input neurons trained from a single-speaker voice samples performed better that with 240 input neurons at an excellent training and testing recognition rate of 100%, with significant improvement when tested with homonyms at a recognition rate of 75.00%. The multiple speakers network with 240 input neurons resulted to a high recognition rate of 97.44% during training and 87.20%

during testing when using four-hidden layer neural network, while failed to achieve an improvement in the training performance, when input neurons were decreased to 120. For single-speaker voice recognition, the optimum number of hidden layers is 3 at 120 input neurons, while 4 hidden layers at 240 input neurons when there are multiple-speakers

Area for improvement is finding the optimum MFCC size for word or phrase that will be fed to the neural network without compromising the information content and training time. Other pre-processing techniques such as, voice activity detection and noise elimination, can help improve the recognition rate. The effect of homonyms is another challenge that needs further studies. Since this study is focused on interactive systems for female users, a separate study to verify the performance rate of neural-network based speech recognition system for male speech, and the combination of both, can be explored.

REFERENCES

[1] George, S. et al, "Using dynamic synapse based neural networks with wavelet preprocessing for speech applications", *Proceedings of the International Joint Conference on Neural Networks*, 2003, pp. 666-669.

[2] Romo, Julio Cesar Martinez, et al, "Combining genetic algorithms and FLDR for real-time voice command recognition", *Seventh Mexican International Conference on Artificial Intelligence*, 2008. pp. 163-169.

[3] Tao, J., Jiang, C., "a domestic speech recognition based on hidden markov model", *Proceedings of IEEE Cloud Computing and Intelligent Systems (CCIS) 2011*, pp. 606-60.9

[4] Wisestu, U et al., "Indonesian speech recognition system using discriminant feature extraction – neural predictive coding (DFE-NPC) and probabilistic neural network", *IEEE International Conference on Computational Intelligence and Cybernetics*, 2012, pp. 158-162.

[5] Sunny, Sonia et al, "Feature extraction methods based on linear predictive coding and wavelet packet decomposition for recognizing spoken words in malayalam", *2012 International Conference on Advances in Computing and Communications*, pp.27-30.

[6] Hoseinkhani, Fatemeh et al, "Speech recognition by classifying speech signals based on the fire fly and fuzzy", *2012 International Conference on Advanced Computer Science Applications and Technologies*, pp. 187-191.

[7] Yang, Chenghui et al, "Based on artificial neural networks for voice recognition word segment", *IEEE 3rd Conference on Communication Software and Networks (ICCSN)*, 2011, pp. 394-396.

[8] Chin, Siew Wem et al, "Improved voice activity detection for speech recognition system", *2010 International Computer Symposium*, pp. 518-523.

[9] Florez-Choque, Omar and Cuadros-Vargas, Ernesto, "Improving human computer interaction through spoken natural language", *Proceeding of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)*, pp. 346-350.

[10] Sanchez, P. et al, "Hybrid neural-based guiding system for mobile robots", *Annual Meeting of the North American Fuzzy Information Processing Society*, 2008, pp. 1-6.

[11] Nakamura, R. et al, "fast robot voice interface through optimum-path forest", *IEEE 16th International Conference on Intelligent Engineering Systems*, 2012, pp.67-71.

[12] Billiones, R.K. et al., "Speech-controlled human-computer interface for audio-visual breast self-examination guidance system", *8th International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM)*, 2015.

[13] Billliones, R.K. et al., "Intelligent operating architecture for audio-visual breast self-examination multimedia training system", *TENCON 2015 – 2015 IEEE Region 10 Conference*, 2015.

[14] Muda, Lindasalwa, et. al.. "Voice recognition lgorithms using mel frequency cepstral coefficient (MFCC) and dynamic time wartping (DTW) techniques", *Journal of Computing*, Volume 2, Issue 3, March 2010, pp. 138-143.

[15] Abushariah, A., Gunawan, T., Abushariah, M., "english digits speech recognition system based on hidden markov models", *International Conference on Computer and Communication Engineering (ICCCE 2010)*, May 11-13, 2010, Kula Lumpur, Malaysia.

[16] Sajjan, S., Vijaya C., "Comparison of DTW and HMM for isolated word recognition", *Proceedings of the International Conference on Patter Recognition, Informatics and Medical Engineering*, March 21-23, 2012, pp.466-470.

[17] Jensen, J., Ellis, D., "Quantitative analysis of a common audio similarity measure", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 177, No. 4, May 2009, pp.693-703.

[18] Lyon, J. (2009-2012), Mel-frequency cepstral coefficient (MFCC) tutorial [Online]. Available: http://www.practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/#computing-the-mel-filterbank.

[19] Rabiner, L., Schafer, R., "Introduction to digital speech processing", *Foundations and Trends in Signal Process*ing, Vol. 1, Nos 1-2 (2007), pp.66-72.

[20] Ning, D., "Developing an isolated word recognition system in MATLAB" [Online], 2009 Math Digest, Mathworks, Available : http://www.mathworks.com/company/newsletters/articles/developing-an-isolated-word-recognition-system-in-matlab.html.

[21] Graupe, D., Principles of artificial neural networks 2nd Ed., 2007, Singapore, Scientific Publishing Co. Pte. Ltd.

[22] Osuna, R. "Cepstral analysis : computing for MFCC (ex9p2.m)" [Online], Toolbox Available : http://www.findthatzip-file.com/search-43664645-hZIP/winrar-winzip-download-l9.zip.htm.

[23] Wojcicki, K., HTK MFCC MATLAB [Online], Available : http://www.mathworks.com/matlabcentral/fileexchange/32849-htk-mfcc-matlab/content/mfcc/mfcc.m.

[24] Brucal, S., Dadios, E.,"Voice feature extraction using matlab auditory and voicebox toolbox", *6th International Conferene on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environement and Management (HNICEM)*, November 12-14, 2013, Philippines.

[25] Brookes, M., "VOICEBOX: Speech processing toolbox for MATLAB" [Online], Available : http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.

[26] Zheng, F., Zhang, G., Song, Z., "Comparison of different implementations of MFCC", *J.Comput.Sci. &Techno., Vol. 16 No. 6, Nov. 2001*, pp.582-589.

[27] Loong, J., et al, "Comparison of MFCC and cepstral coefficiencts as a feature set for PCG biometric systems", *World Academy of Science, Engineering and Technology*, 2010, pp.764-768.

[28] Mohammed, E., Sayed, M., Mosehly A., Abdelnaiem, A., " LPC and MFCC performance evaluation with artificial neural network for spoken language identification", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 6, No. 3, June 2013, p.5-66.

[29] Panchal, F., Panchal, M., "Review on methods of selecting humber of hidden nodes in artificial neural networks", *International Journal of Computer Science and Mobile Computing Vol. 3, Issue 11*, 2014, pp.455-463.

[30] Wagarachchi, N., Karunananda, A., "Mathematical modelling of hidden layer architecture in artificial neural networks", 2012 *International Conference on Information Security and Artificial Intelligence,* 2012, pp. 154-159.

[31] Stathakis, D., "How many hidden layers and nodes", *International Journal of Remote Sensing Vol. 30 No. 8,* 2009, pp. 2133-2147.

[32] Reda, A., Aoued, B., "Artificial neural network & mel-frequency cepstrum coefficients-based speaker recognition", *3rd International Conference : Science of Electronic, Technologies of Information and Telecommunication (SETIT 2005),* March 27-31, 2005, Tunisia.

[33] Karsoliya, S., "Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture", *International Journal of Engineering Trends and Technology*, Vol. 3, Issue 6, 2012 pp-714-717.