Wavelet-Neural Networks Based Phonetic Recognition System of Arabic Alphabet letters

Moaz Abdulfattah Ahmad, Rasheed M. El Awady

Faculty of Engineering, Mansoura University, Egypt.

Summary

This paper is a new trial to recognize the Arabic letters (Arabic Alphabet letters) spoken by any speaker by using artificial neural networks through wavelet technique. This will be useful in converting the spoken words into written text and using of microphone instead of key board, also this can help disable people (handicapped) with limited movement to write any text by voice instead of their hands. A suggested recognition system is implemented to recognize Arabic Alphabet letters of independent speakers. This system is based on analyzing phonetic isolated Arabic alphabet letters. The voice signal is provided to wavelet toolbox to be analyzed. Daubechies (db4) mother function is used in the feature extraction process to produce letters wavelet coefficients. Wavelet coefficients corresponding to each alphabet are used to train multilayer perceptron neural networks to produce recognized binary codes corresponding to each letter. These binary codes (corresponding to alphabet letters) can be decoded to be used as displayed letters on monitors, or printed on paper or used as commands for moving a controlled mechanism. About 95% detection rate has been achieved over large dataset.

Key words:

Speech processing, Arabic speech recognition, wavelet Transform, ANN, speech technology.

1. Introduction

Speech recognition technology is used more and more for telephone applications like travel booking and information, financial account information, customer service call routing, and directory assistance. Using constrained grammar recognition, such applications can achieve remarkably high accuracy. Research and development in speech recognition technology has continued to grow as the cost for implementing such voice-activated systems has dropped and the usefulness and efficacy of these systems has improved. For example, recognition systems optimized for telephone applications can often supply information about the confidence of a particular recognition, and if the confidence is low, it can trigger the application to prompt callers to confirm or repeat their request. Furthermore, speech recognition has enabled the automation of certain applications that are not automatable using push-button interactive voice response (IVR) systems, like directory assistance and systems that allow callers to "dial" by speaking names listed in an electronic phone book [1].

Three primary speech technologies are used in voice processing applications: stored speech, text-to – speech and speech recognition. Stored speech involves the production of computer speech from an actual human voice that is stored in a computer's memory and used in any of several ways. Speech can also be synthesized from plain text in a process known as text-to – speech which also enables voice processing applications to read from textual database.

Speech recognition is the process of deriving either a textual transcription or some form of meaning from a spoken input. Speech analysis can be thought of as that part of voice processing that converts human speech to digital forms suitable for transmission or storage by computers. Speech synthesis functions are essentially the inverse of speech analysis – they reconvert speech data from a digital form to one that's similar to the original recording and suitable for playback. Speech analysis processes can also be referred to as a digital speech encoding (or simply coding) and speech synthesis can be referred to as Speech decoding [2].

This paper is organized as follows: section-II presents general constraints and data collections, section-III gives a brief discussion of previous work, section-IV discusses the proposed technique, section-V &VI, introduce the results and conclusions respectively, while, section-VII contains future work.

2. Constraints and Data Collection

Due to the limited time interval; a set of constraints have been placed on the system to make the algorithm more manageable. These constraints are: Acoustic Arabic alphabet letters (28 letters) are received by a sensitive wide band microphone and saved as samples of voice signals to be analyzed (2016 sound wave files). These letters are pronounced by (6) different persons, (3-males + 3-females) with 3 trials for each and by the 4 normal ways of uttering (vowels); Fatha, dhamma, kasra and sekoon (consonant).

Sound recorder and Wave_lab program are used to capture the sound signal while Matlab is used for the analysis and

Manuscript received December 5, 2010
Manuscript revised December 20, 2010

recognition process. Then data is stored in wave format and analyzed using Matlab software, Fig.1 shows an example of the voice signal.

Fig.1 An example of time amplitude voice signal

3. Previous Work

The first speech recognizer appeared in 1952 and consisted of a device for the recognition of single spoken digits [3]. Another early device was the IBM Shoebox, exhibited at the 1964 New York World's Fair.

In 1982 Kurzweil Applied Intelligence and Dragon Systems released speech recognition products. By 1985, Kurzweil's software, had a vocabulary of 1,000 words—if uttered one word at a time. Two years later, in 1987, its lexicon reached 20,000 words, entering the realm of human vocabularies, which range from 10,000 to 150,000 words. But recognition accuracy was only 10% in 1993. Two years later, the error rate crossed below 50%. Dragon Systems released "Naturally Speaking" in 1997 which recognized normal human speech. Progress mainly came from improved computer performance and larger source text databases. The Brown Corpus was the first major database available, containing several million words. In 2001 recognition accuracy reached its current plateau of 80%, no longer growing with data or computing power. In 2006, Google published a trillion-word corpus, while Carnegie Mellon University researchers found no significant increase in recognition accuracy [4].

4. The Proposed Technique

In this section the proposed algorithm will be explained then applied to the Egyptian license plate, the algorithm has four steps: (A) noise reduction and silence removal (B) feature extraction (C) principal component analysis (D) Artificial neural network (MLP).

4.1 Noise Reduction and Silence Removal

Speech enhancement aims to improve speech quality by using various algorithms. The objective of enhancement is improvement in intelligibility and/or overall perceptual quality of degraded speech signal using audio signal processing techniques.

Enhancing of speech degraded by noise, or noise reduction, is the most important field of speech enhancement, and used for many applications such as mobile phones, VoIP, teleconferencing systems , speech recognition, and hearing aids[5].

The algorithm of speech enhancement for noise reduction used is Minimum Mean-Square-Error Short-Time Spectral Amplitude Estimator (MMSE-STSA).see Fig.2.

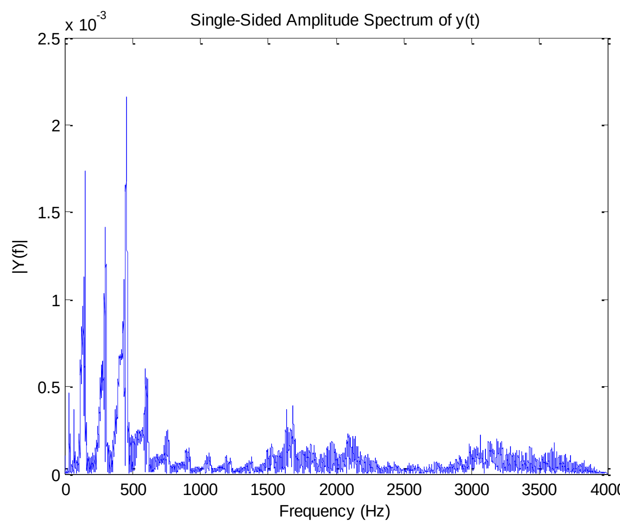


Fig.2 Single-sided amplitude spectrum of y(t)

4.2 Feature Extraction

Feature extraction is the transformation of the original data (using all variables) to a data Set with a reduced number of variables. In the problem of feature selection, the aim is to select those variables that contain the most discriminatory information. Alternatively, we may wish to limit the number of measurements we make, perhaps on grounds of cost, or we may want to remove redundant or irrelevant information to obtain a less complex classifier.

In feature extraction, all variables are used and the data are transformed (using a linear or nonlinear transformation) to a reduced dimension space. Thus, the aim is to replace the original variables by a smaller set of underlying variables. There are several reasons for performing feature extraction: (i) to reduce the bandwidth of the input data

(with the resulting improvements in speed and reductions in data requirements) ;(ii) to provide a relevant set of features for a classifier, resulting in improved performance, particularly from simple classifiers; (iii) to reduce redundancy;(v) to recover new meaningful underlying variables or features that the data may easily be viewed and relationships and structure in the data identified [6].
The discrete wavelet transform (DWT) presents a multiresolution analysis in the form of coefficient matrices which can be used in a manner similar to Fourier series coefficients. This DWT representation can be thought of as a form of "feature extraction" on the original sound.

4.3 Principal Component Analysis

PCA is a well-established technique for feature extraction and dimensionality reduction. It is based on the assumption that most information about classes is contained in the directions along which the variations are the largest. The most common derivation of PCA is in terms of a standardized linear projection which maximizes the variance in the projected space. For a given p-dimensional data set X, the m principal axes$T_1, T_2, \dots, T_m$, where $1 \leq m \leq p$, are orthonormal axes onto which the retained variance is maximum in the projected space. Generally, $T_1, T_2, \dots, T_m$, can be given by the m leading Eigen vectors of the sample covariance matrix
$S = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)(x_i - \mu)^T$, where $xi \in X, \mu$ the sample is mean and N is the number of samples, so that:

$$ST_i = \lambda_i T_i,\ i \in 1, \cdots, m \qquad (1)$$

Whereλ_i is the ^{i}th largest eigenvalue of S. The m principal components of a given observation vector $x \in X$ are given by:

$$y = [y_1, \cdots, y_m] = \left[T_1{}^T x, \dots, T_m{}^T x\right] = T^T x \qquad (2)$$

The m principal components of x are decorrelated in the projected space. In multiclass problems, the variations of data are determined on a global basis, that is, the principal axes are derived from a global covariance matrix:

$$\widehat{S} = \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N_j}(x_{ji} - \hat{\mu})(x_{ji} - \hat{\mu})^T \qquad (3)$$

Where $\hat{\mu}$ the global mean of all the samples, K is is the number of classes, N_j is the number of samples in class j, $N = \sum_{j=1}^{K} N_j$ and x_{ji}represents the ith observation from class j. The principal axes $T_1, T_2, \dots, T_m$ are therefore the m leading eigenvectors of $\widehat{S}$:

$$\widehat{S}T_i = \hat{\lambda}_i T_i, \quad i \in 1, \cdots, m \qquad (4)$$

Where $\hat{\lambda}_i$ is the ^{i}th largest Eigen value of $\widehat{S}$. An assumption made for feature extraction and dimensionality reduction by PCA is that most information of the observation vectors is contained in the subspace spanned by the first m principal axes, where m < p. Therefore, each original data vector can be represented by its principal component vector with dimensionality m [7].

4.4 Artificial Neural Network (MLP)

An ANN is an information processing system that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large Number of highly interconnected processing elements (Neurons) working with each other to solve specific Problems. Each processing element (neuron) is basically a summing element followed by an activation function. The output of each neuron (after applying the weight parameter associated with the connection) is fed as the input to all of the neurons in the next layer. The learning process is essentially an optimization process in which the parameters of the best set of connection coefficients (weighs) for solving a problem are found and includes the following basic steps:

- Present the neural network with a number of inputs (Vectors each representing a pattern)
- Check how closely the actual output generated for a Specific input matches the desired output.
- Change the neural network parameters (weights) to better approximate the outputs [8].

Multilayer perceptron (MLP) [9] is one of most commonly used neural network classification algorithms. The architecture used for the MLP during simulations with our speech dataset consisted of a three layer feed-forward neural network: one input, one hidden, and one output layers. hard limit transfer functions were used for each neuron in both the hidden and the output layers with slope value of 1.0. The learning algorithm used was stochastic gradient descent with mean squared error function. There were a total of 200 neurons in the input layer (200-feature input pattern), and 112 neurons in the output layer.

5. Result

Artificial Neural Networks used to recognize wavelet coefficients corresponding to Alphabet letters is created and trained by Incremental training style.

5.1 Creation of the network

In this style of work training data of each speaker are fed to the perceptron network separately. There are eight training matrices represent two trials of four speakers, each contain 112data files representing 28 Arabic alphabets letters with its 4 vowels. Testing data consists of four matrices represent the third trial of the previous four speakers. Validation data are 6 matrices representing 3 trials of 2 speakers. All the 18 matrices consist of 5000 rows represent the wavelet coefficients and 112 columns represent the letters vowels. The created network is named "network5000" consists of seven neurons with hardlim transfer function.

5.2 Training, Testing and validation data of the network

Matrices data for the neural network are given as shown in Fig. 3; it includes the training data (Mar50001, Mar50002, Aml50001, Aml50002, Moh50001, Moh50002, Moz50001 and Moz50002).

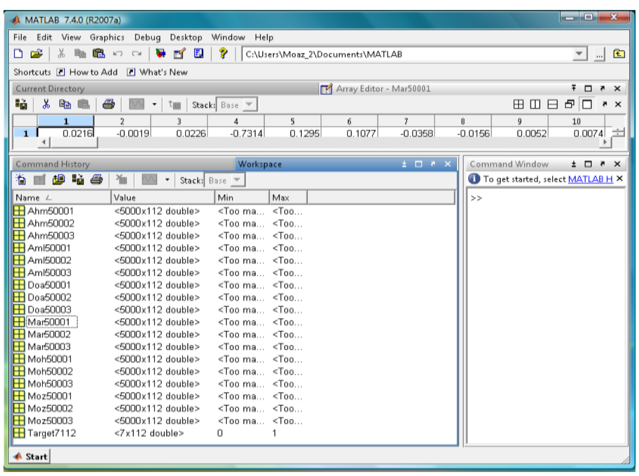


Fig. 3 Matrices for incremental training.

Testing data are given by Matrices (Mar50003, Aml50003, Moh50003 and moz50003), while validation data are given by matrices (Ahm50001, Ahm50002, Ahm50003, Doa50001, Doa50002 and Doa50003). The target data is given by matrix "Target7112". The output data of the network and error data are given as network5000_outputs and network5000_errors.

5. 3 Network Performance and Results

Network performance and results charts of eight incremental training data with separate validation and testing data are shown in Fig. 4 (a, b, c, d, e, f, g, h).

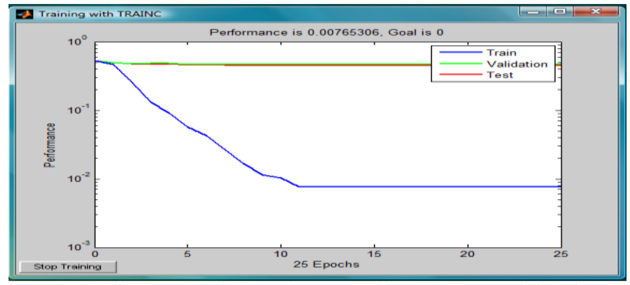


Fig. 4 (a) Performance of "network5000" with Mar50001 training data

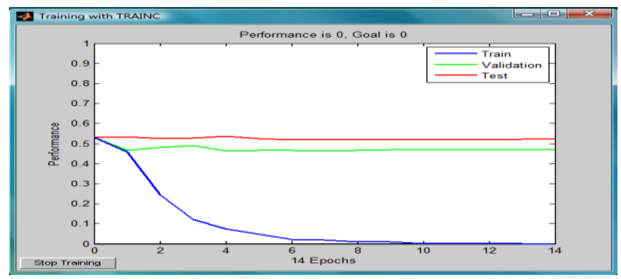


Fig. 4 (b) Performance of "network5000" with Mar50002 training data

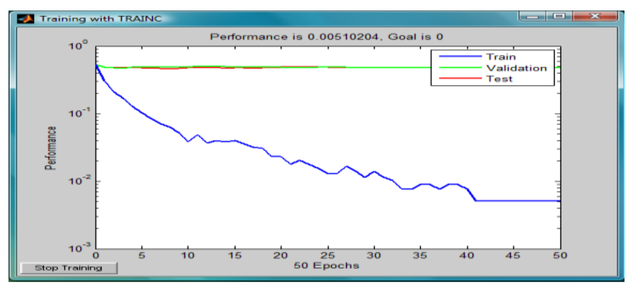


Fig. 4 (c) Performance of "network5000" with Aml50001 training data

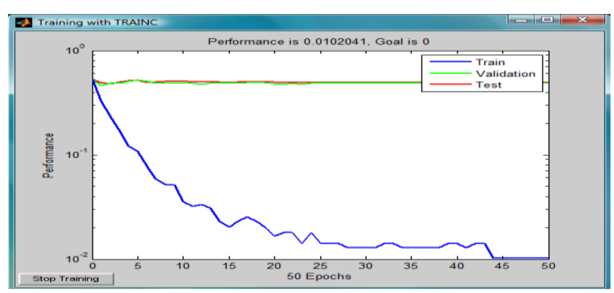


Fig. 4 (d) Performance of "network5000" with Aml50002 training data

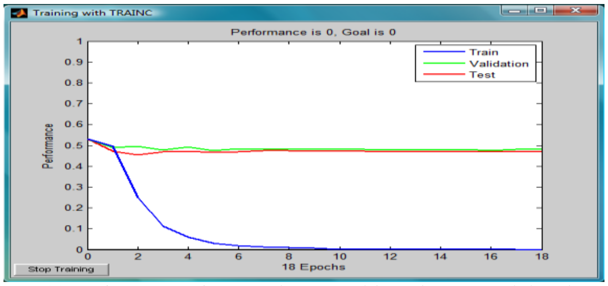


Fig. 4 (e) Performance of “network5000” with Moh50001 training data

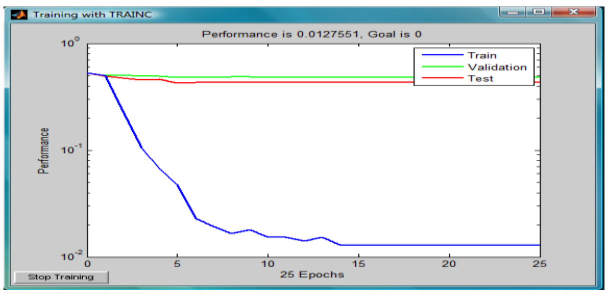


Fig. 4 (f) Performance of “network5000” with Moh50002 training data

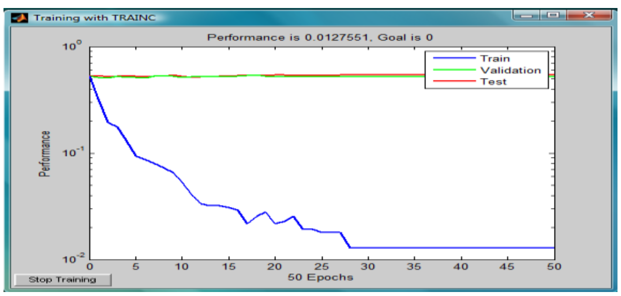


Fig. 4 (g) Performance of “network5000” with Moz50001 training data

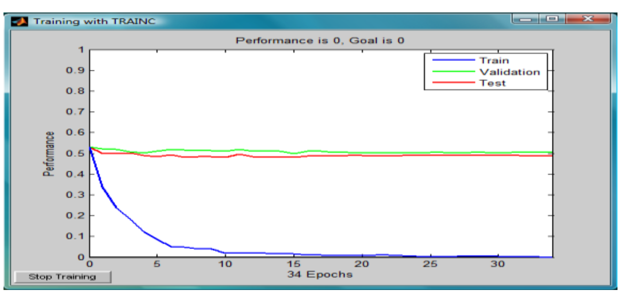


Fig. 4 (h) Performance of “network5000” with Moz50002 training data

Training information, testing and validation data are set to the network as shown in table 1. The output of the network (network5000_outputs) comparing to the target data determines the network output error (network5000_errors) which shows the network performance. Resulting errors (network performance) of each case of training are recorded also in table 1.

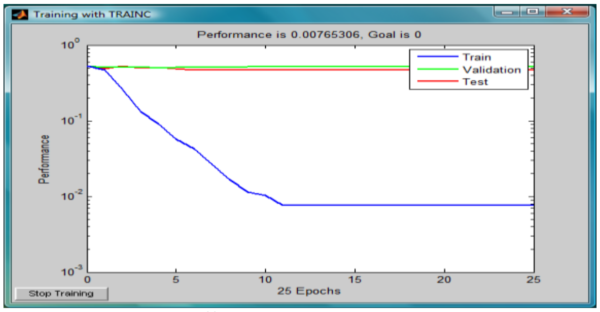
"Network5000" performance (Goal 0)

S/N	Training Data	Testing Data				Validation Data						Training performance
		Mar50003	Aml50003	Moh50003	Moz50003	Ahm50001	Ahm50002	Ahm50003	Doa50001	Doa50002	Doa50003	
1	Mar50001	0.4124	0.4345	0.4326	0.4732	0.42941	0.4421	0.5231	0.4241	0.5214	0.5016	0.0076531
2	Mar50002	0.5238	0.4389	0.4983	0.5089	0.4732	0.5089	0.5156	0.5421	0.4321	0.4956	0
3	Aml50001	0.4956	0.4826	0.4987	0.5125	0.4825	0.4984	0.4921	0.4824	0.4892	0.4912	0.0051020
4	Aml50002	0.4924	0.4523	0.4894	0.4892	0.4872	0.4687	0.4896	0.5089	0.4798	0.4945	0.0102041
5	Moh50001	0.47218	0.4821	0.4821	0.4598	0.4869	0.4856	0.51098	0.4742	0.4512	0.4825	0
6	Moh50002	0.50321	0.4987	0.4259	0.4689	0.5124	0.5052	0.4927	0.4897	0.4758	0.4975	0.0127551
7	Moz50001	0.5227	0.5078	0.5021	0.5047	0.5142	0.4987	0.51089	0.4982	0.4895	0.51074	0.0127551
8	Moz50002	0.5096	0.4802	0.4998	0.4785	0.4989	0.5128	0.51978	0.50859	0.5124	0.5189	0

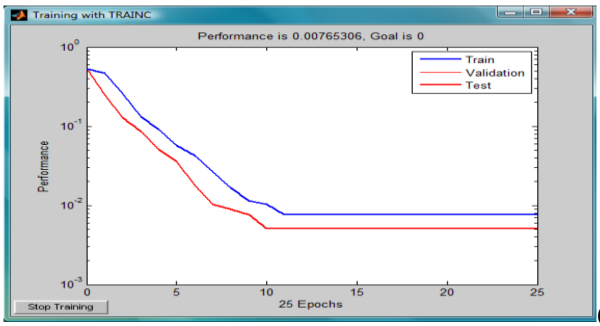
Table 1 "network5000" results for 8 incremental training data inputs

When the perceptron neural network is tested and validated with the same training input data, it gives better results compared with when it is different as shown in Fig.5.

From Fig.5 it is noticed that performance of neural network “network5000” have the same training performance in both cases (a) and (b), but testing and validation results are widely improved in case of using same validation and testing data as that of training data [Fig.5 (b)] than in case of different data [Fig.5 (a)]. In the second case (b), testing and validation network performance approaches to steady values (0.005) after about 10 epochs. These results are logical since the neural network is well recognizing the already trained data.



(a) With different testing and validation data



(b) With same testing and validation data

Fig.5 (a, b) Performance of perceptron network "network5000"

Results of "network5000" shown in Fig.5 are taken using data of Mar50001 matrix, similar better performance results are obtained when using the other seven training data matrices (Mar50002 up to Moz50002). Fig.6 (a, b, c, d, e, f, g, h) show the performance of "network5000" in incremental training with the eight self-testing and validation data matrices.

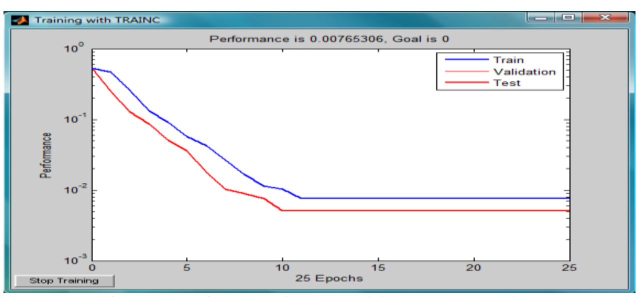


Fig.6 (a) performance of "network5000" with Mar50001 training data

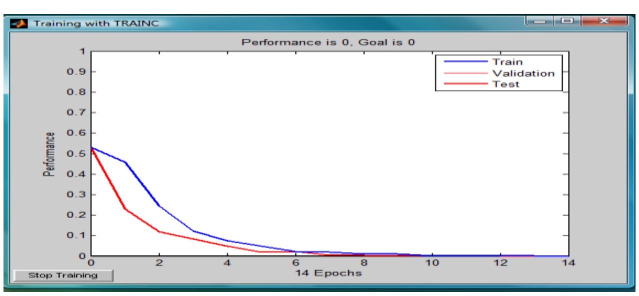


Fig.6 (b) performance of "network5000" with Mar50002 training data

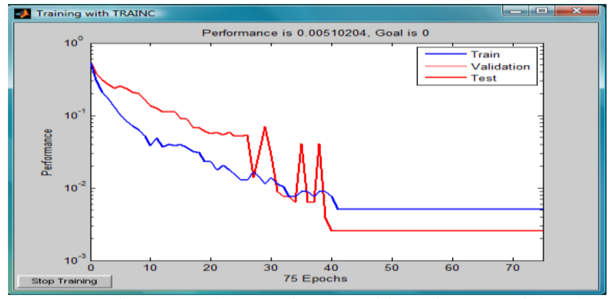


Fig.6 (c) performance of "network5000" with Aml50001 training data

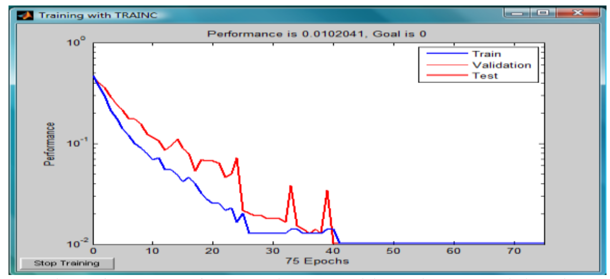


Fig.6 (d) performance of "network5000" with Aml50002 training data

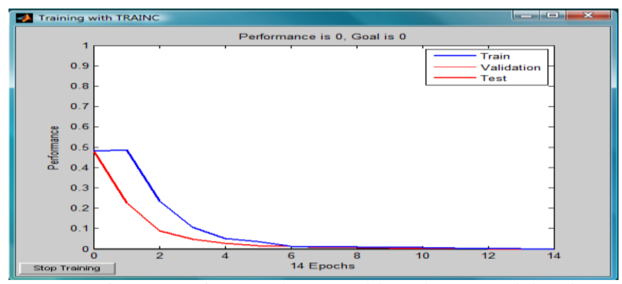


Fig.6 (e) performance of "network5000" with Moh50001 training data

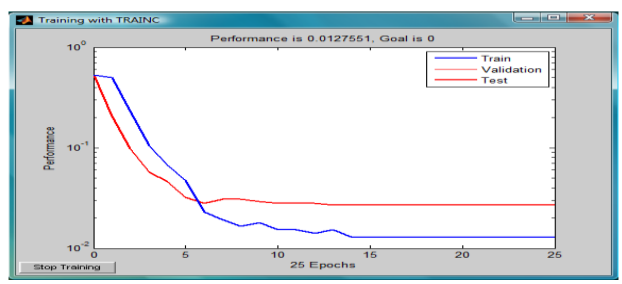


Fig.6 (f) performance of "network5000" with Moh50002 training data

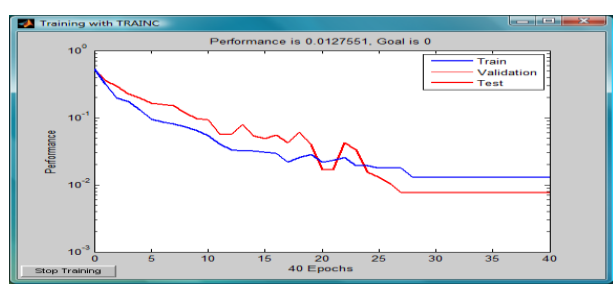


Fig.6 (g) performance of "network5000" with Moz50001 training data

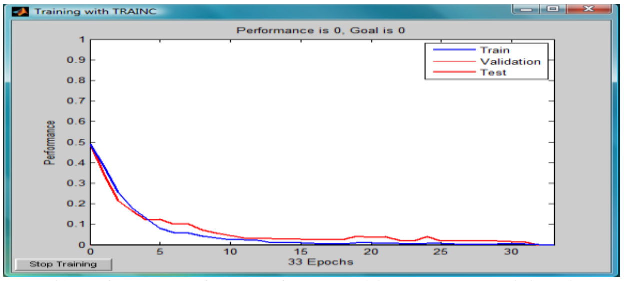


Fig.6 (h) performance of “network5000” with Moz50002 training data

Performance results of “network5000” in incremental training with the eight self-testing and validation data (Fig.6) are listed in table 2.

"Network5000" performance (Goal 0)

S / N	Training Data	Testing and validation data								No. of epochs	Training performance
		Mar 50001	Mar 50002	Aml 50001	Aml 50002	Moh 50001	Moh 50002	Moz 50001	Moz 50002		
1	Mar50001	0.0052								12	0.00765306
2	Mar50002		0							9	0
3	Aml50001			0.0017						42	0.00510204
4	Aml50002				0.0102					41	0.0102041
5	Moh50001					0				11	0
6	Moh50002						0.0275			14	0.0127551
7	Moz50001							0.0079		28	0.0127551
8	Moz50002								0	32	0

Table 2 "network5000" results for 8 incremental training data inputs with self-testing and validation

From table 2 it is noted that testing and validation errors reflect the training performance of the network. When the network training performance is perfect and approaches zero goal (as in case of training data number 2, 5 and 6 in the table) the validation and testing errors are zero too. The other values of testing and validation errors (case number 1, 3, 4, 6 and 7 in the table) are very small and near to the corresponding training performance values.

Training performance of the perceptron neural network “network5000” is a measure of the capability of the trained network to recognize the alphabet letters. Table 3 shows the network training performance (errors) and the corresponding errors of the network “network5000_errors” detected from Error Data in Network/Data Manger.

"Network5000" performance (Goal 0)

S / N	Training Data	Detected neurons Errors		corresponding alphabet	errors data			Training performance
		Location	No. of error Digits		sum	total digits	%	
1	Mar50001	83,84,102,106	1, 1, 2, 2	وِ ، هـِ ، قُ ، قْ ،	6	7x112=784	0.007653	0.00765306
2	Mar50002	0	0	**0**	0	0	0	0
3	Aml50001	104, 108	2, 2	وْ ، هـْ	4	784	0.005102	0.00510204
4	Aml50002	10,18,65,66,81,85	2,2,1,1,1,1	كَ ، قَ ، ظِ ، ظَ ، جِ ، تِ	8	784	0.010204	0.0102041
5	Moh50001	0	0	**0**	0	0	0	0
6	Moh50002	2, 109	5, 5	يَ ، أِ	10	784	0.0127551	0.01275511
7	Moz50001	7, 11, 60, 61, 70, 74	2, 2, 1, 1, 2, 2	غِ ، عِ ، طْ ، ضْ ، ثُ ، بُ	10	784	0.0127551	0.01275511
8	Moz50002	0	0	**0**	0	0	0	0

Table 3 Perceptron neural network "network5000" performance with respect to Arabic alphabet letters errors

From table 3 it is observed the following:
Only 20 letters from 112 vowel letters shows errors
Letters (ق، هـ، و، ظ) give more errors with respect to the speakers.
Speakers’ trial no. 2, 5 and 8 give no errors.
Network performance errors does not exceed than 1.3%
The average percentage error for females speakers (case 1, 2, 3, 4) is 0.0057147 is less than that of males (case 5, 6, 7, 8) which is 0.0063775.
The Arabic alphabet letters that give errors in testing and validation of the trained neural networks “network5000” are listed in table 4. It noticed that all the letters errors have not the same weight.

S/N	Arabic letter	order # of 112	Training data source	Repeated time	total digital data	Error value	Relative % error	Network % error
1	أِ	2	Moh50002	5	112	0.0446428	4.46428	0.558035
2	بُ	7	Moz50001	2	112	0.0178571	1.78571	0.2232137
3	تِ	10	Aml50002	2	112	0.0178571	1.78571	0.2232137
4	ثُ	11	Moz50001	2	112	0.0178571	1.78571	0.2232137
5	جِ	18	Aml50002	2	112	0.0178571	1.78571	0.2232137
6	ضْ	60	Moz50001	1	112	0.0089285	0.89285	0.1116071
7	طْ	64	Moz50001	1	112	0.0089285	0.89285	0.1116071
8	ظَ	65	Aml50002	1	112	0.0089285	0.89285	0.1116071
9	ظِ	66	Aml50002	1	112	0.0089285	0.89285	0.1116071
10	عِ	70	Moz50001	2	112	0.0178571	1.78571	0.2232137
11	غِ	74	Moz50001	2	112	0.0178571	1.78571	0.2232137
12	قَ	81	Aml50002	1	112	0.0089285	0.89285	0.1116071
13	قُ	83	Mar50001	1	112	0.0089285	0.89285	0.1116071
14	قْ	84	Mar50001	1	112	0.0089285	0.89285	0.1116071
15	كَ	85	Aml50002	1	112	0.0089285	0.89285	0.1116071
16	هِـ	102	Mar50001	2	112	0.0178571	1.78571	0.2232137
17	هْـ	104	Aml50001	2	112	0.0178571	1.78571	0.2232137
18	وِ	106	Mar50001	2	112	0.0178571	1.78571	0.2232137
19	وْ	108	Aml50001	2	112	0.0178571	1.78571	0.2232137
20	يَ	109	Moh50002	5	112	0.0446428	4.46428	0.558035

Table 4 Arabic Alphabet errors given by perceptron neural network "network5000"

It is noted from table 4 the following:

1. Letters (أِ ، يَ) give the highest network errors (0.558035 %).
2. Letters ((بُ ، تِ ، ثُ ، جِ ، عِ ، غِ ، هِـ ، هْـ ، وِ ، وْ give network error (0.2232137 %).
3. Letters (ضْ ، طْ ، ظَ ، ظِ ، قَ ، قُ ، قْ ، يَ) give network error (0.1116071 %).
4. The remaining 92 vowel letters give network error 0 %.
5. The average network error for the eight training data is 0.2120531 %.

6. Conclusion

A trial to build up a system to recognize phonetic Arabic alphabet letters spoken by any speaker is implemented. This model uses wavelet for letters analysis and artificial neural network for the recognition process. Spoken letters, with its normal 4 vowels, were captured from ordinary usable word as a step to recognize the whole word in future. Sound recorder and Wave lab program are used to capture the sound signal while Matlab is used for the analysis and recognition process. From this research work it can be concluded the following:

Recorded sound signals are digitally sampled with frequency 22.5 KHz and this rate gives good intelligibility and avoid aliasing but in the other hand shows more data size.

Using wavelet Daubechies family “dB4” with level 2 provide good results shown in the syntheses process.

Compressing the wavelet coefficient reduces the data size without affecting the analysis accuracy. Choice Perceptron neural network with hard limiter transfer function matches with the required binary format of target and network outputs.

Incremental training of the used neural network gives better results when same training data in testing and validation. Among 112 vowel Arabic letters only 20 letters showed errors with less than 1.3% while 92 letters showed errors-free. In recognition process, the used perceptron neural network “network5000” gives excellent (and not perfect) recognition results since the average percentage error is 0.2120531 %.

Future work

Arabic speech recognition will always remain as a challenging research area, and the trial of implementation of phonetic recognition systems on DSP chips will continue until it becomes a reality.

The future work research is to combine the recognized letters to form Arabic words and also to be in use with different accents of Arabic language speakers.

References

[1] WWW.advancedsourcecode.com at 5 Sep 2010.

[2] Anjali Kalra[1], Sarbjeet Singh[2], Sukhvinder Singh[3],'speech recognition', Sri Sai College of Engg. And Technology, Pathankot[1,2,3], IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[3] Davies , K.H., Biddulph, R. and Balashek, S. (1952) Automatic Speech Recognition of Spoken Digits, J. Acoust. Soc. Am. 24(6) pp.637 – 642.

[4] "The History of Automatic Speech Recognition Evaluations at NIST". National Institute of Standards and Technology. May, 2009.http://www.itl.nist.gov/iad/mig/publications/ASRhistory/index.html. Retrieved May, 2010.

[5] J. Benesty, S. Makino, J. Chen (ed). Speech Enhancement. pp.1-8. Springer, 2005.

[6] Andrew R. Webb, QinetiQ Ltd., Malvern, Statistical Pattern Recognition, John Wiley & Sons Ltd.2002.

[7] Sarfraz, Muhammad. "Computer-aided intelligent recognition techniques and applications", John Wiley & Sons Ltd, 2005.

[8] Sergios Theodorios and Konstantinos Koutroumbas, Pattern Recognition, Cambridge: Academic Press, 1999.

[9] Werbos,"Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences", PhD thesis, Harvard University, 1974.

Moaz Abdulfattah Ahmad Mohamad Abdullah, Date of Birth: 17/03/1952 , Zagazig, Egypt, Address: P.O. Box 817 Makkah, KSA, B.Sc.: Electrical and communications Engineering, Military Technical College, Cairo, Egypt. 1975, English Language Diploma, DLI, LAFB, San Antonio, Texas, USA.1981. M.Sc. Electrical and communications Engineering, Faculty of Engineering, Zagazig University, Egypt,1999. PhD (Studies), Electrical & Communications Engineering, Faculty of Engineering, Mansoura University, Egypt, 2004. Work Career: Engineer Officer in Egyptian Armed Forces (Air D.)1975-1993. Training Advisor, German Authority (GTZ), Mubarak /Kool project, Egypt, 1996-1997. Communications systems Supervisor, Seavin, (American Company), Egypt, 1994, 1999. Supervisor Engineer, Holly Haram, Makkah, KSA, 1999-2010.