Journal of Intelligent Learning Systems and Applications, 2012, 4, 41-52
http://dx.doi.org/10.4236/jilsa.2012.41004 Published Online February 2012 (http://www.SciRP.org/journal/jilsa)

Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks

Manar Maraqa[1], Farid Al-Zboun[2], Mufleh Dhyabat[3], Raed Abu Zitar[4]

[1]Department of Management Information Systems, Al-Isra' Private University, Amman, Jordan; [2]College of Information Technology, Ajloun National University, Ajloun, Jordan; [3]College of Information Technology, AL al-Bayt University, Mafraq, Jordan; [4]School of Engineering and Technology, New York Institute of Technology, Amman, Jordan.
Email: manar@maraqa.net, rzitar@nyit.edu.jo

Received February 20th, 2011; revised March 14th, 2011; accepted March 24th, 2011

ABSTRACT

The objective of this research is to introduce the use of different types of neural networks in human hand gesture recognition for static images as well as for dynamic gestures. This work focuses on the ability of neural networks to assist in Arabic Sign Language (ArSL) hand gesture recognition. We have presented the use of feedforward neural networks and recurrent neural networks along with its different architectures; partially and fully recurrent networks. Then we have tested our proposed system; the results of the experiment have showed that the suggested system with the fully recurrent architecture has had a performance with an accuracy rate 95% for static gesture recognition.

Keywords: Arabic Sign Language; Feedforward Neural Networks; Recurrent Neural Networks; Gesture Recognition

1. Introduction

Deaf people use sign language as an alternative to spoken language A sign language provides signs for whole words as well as signs of letters for words that don't have a corresponding sign in that sign language. Hearing people have difficulty to learn sign languages, also it is difficult for deaf people to learn oral languages. Sign language recognition systems can facilitate the communication between those two communities. Also for the hearing people, such systems can help in interacting with machines in a more natural way.

This research aims to reduce the gap of communications between deaf and hearing people by focusing on the recognition of letters, which are the bases of a sign language. Hand gestures recognition systems can be classified into two categories: static and dynamic. A static gesture is a particular hand movements represented by a single image, while a dynamic gesture is a moving gesture represented by a sequence of images. In principle, dynamic gesture recognition is static gesture recognition with frame segmentation functions. These functions decide on the start and end of a single static gesture image frames. This work focuses on the ArSL (Arabic Sign Language) which uses one hand and 28 gestures (see **Figure 1**) to communicate the 28 letters of the Arabic alphabet.

Research on hand gesture usually falls into three categories. The first one is glove-based analysis, which employs sensors attached to the glove. Look-up table software is usually provided with the glove to be used for hand gesture recognition [1].

The second category involves analysis of drawing gestures; this category is merely a character recognition problem. The third category, vision-based analysis, is based on the use of video cameras to capture the movement of the singer' hand that is sometimes aided by making the signer wear a glove that has painted areas indicating the positions of the fingers and the wrist then use those measurements in the recognition process.

As mentioned before, deaf people use their hands in particular movements of palm and fingers to express their thoughts and feelings. Therefore palm orientations of hands' images are estimated to perform gesture recognition. This will be adopted in our work since video-based ArSL recognition methods are still major areas of research and plenty of investigates can be carried in this area. K. Assaleh and M. Al-Rousan [2] proposed the use of polynomial networks as a classification engine for automatic Arabic Sign Language (ArSL) recognition system. They built the system and measured its performance using real ArSL data collected from deaf people. Images were collected from deaf participants performing Arabic sign gestures using a colored glove. Thirty features were extracted from the segmented color regions which represent the fingertips and their relative positions and orientations with respect to the wrist and to each

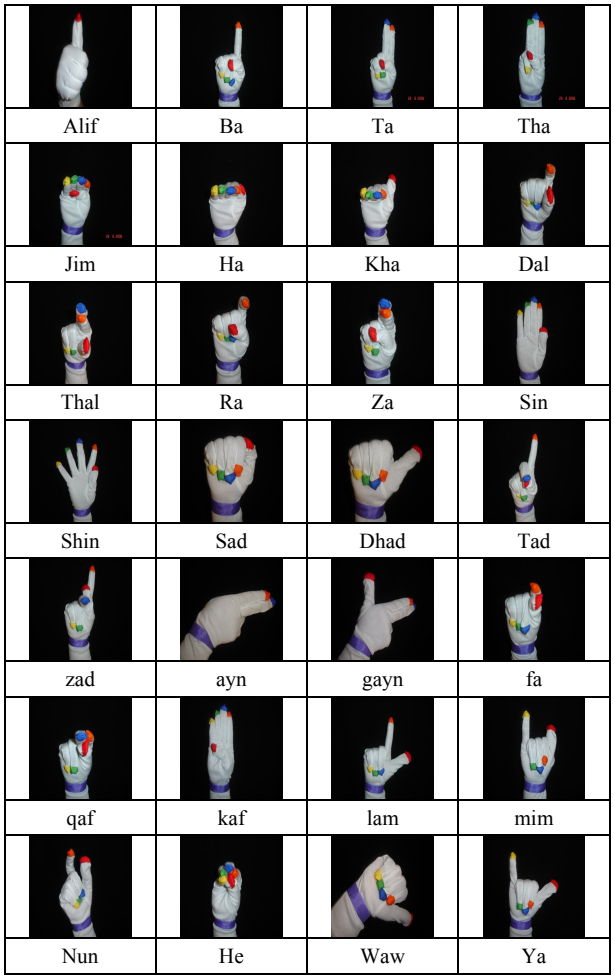

Figure 1. Arabic sign language gestures.

other. They compared the performance of their system with previously published work using ANFIS-based classification on the same data set and the same corresponding feature set. The polynomial-based system has produced superior recognition results to those obtained by the ANFIS-based system for both training and test data. The recognition of the proposed system has shown an excellent performance with a low error rate of 6.59% corresponding to a recognition rate of 93.41%.

Two cameras with 3-D techniques were used [3] in several applications. It worth mentioning that Gesture recognition was also applied on different languages other than English and Arabic [4]. Techniques such as fuzzy decision trees, transition-movement models, saptio-temporal, or common sense context were applied with remarkable success [5-8].

Y. Yuan, Y. Liu and K. Barner [9] designed a set of command-like gestures for users with limited range and function in their digits and wrist. Trajectory and angle features are extracted from these gestures and passed to a recurrent neural network for recognition. Experiments are performed to test the feasibility of gesture recognition system and determine the effect of network topology on the gesture recognition rate. The proposal approach that they adopted achieved an overall correct rate of approximately 94.5%.

R. Salomon and J. Weissmann [10] this paper is about a data glove as a powerful input device for virtual reality and multi media applications. The data glove system allows the user to present the system with a rich set of intuitive commands. This paper applied algorithms to fine tune pre-learned radial basis function networks. In experiments they held, they used the CyberGlove, which measures the angles of 18 joints of the hand: two for each finger, one each for the angles between neighboring fingers, as well as one each for thumb rotation, palm arch, wrist pitch, and wrist yaw. They choose a set of 20 static hand gestures. The experiments were performed with neural network that has three standard three-layered backpropagation with different structures to exploit a (tentative) correlation between gestures and finger combinations.

F. Florez, J. Garcia, and A. Hernandez [11] in this paper a structure capable of characterizing hand posture and its movement was presented. Topology of a self-organizing neural network determines posture, whereas its adaptation dynamics throughout time determines gesture. To validate this method, they have trained the system with 12 gestures, some of which are very similar, and have obtained high success rates.

M. Mohandes and S. Buraiky [12] introduced a system to recognize isolated signs from the Arabic sign language using the Support Vector Machine algorithm. An instrumented glove was used to get raw measurements sent by the glove in the form of a sequence of frames with values that represent the position of the hand, the roll of the hand, the flexes of the thumb, the index finger, the forefinger and the ring finger. Results obtained in this research were promising.

K. Murakami and H. Taguchi [13] developed a posture recognition system using neural networks which could recognize a finger alphabet of 42 symbols of Japanese sign language. Then developed a gesture recognition system where each gesture specifies a word. To deal with dynamic processes they used a recurrent neural network described a gesture recognition method which can recognize continuous gesture. The recognition rate was found to be 94%.

O. Al-Jarrah, A. Shatnawi, and A.H. Halawani [14] presented two neural network systems for the recognition of the sign language alphabets. The first is based on the feedforward neural network architecture, while the second is based on probabilistic neural networks. The user was not required to wear any electromechanical device or any marker to interact with the system. The system takes

images of gestures and converts them to a set of features suitable to be fed to the neural network structure for recognition. The feature extraction involved the orientation and the position of hand within the image. It is shown that an accuracy of 94.4% is achieved using the first system and 91.3% using the second.

C programming language and MATLAB software package were used in this research. MATLAB was used for image processing while the selected neural networks (feedforward and recurrent networks) were designed and implemented using the C programming language.

2. Methodology

The methodology of this research includes four stages (see **Figure 2**) which can be summarized as follows: 1) data collection and image acquisition, 2) image processing, 3) feature extraction and finally 4) gesture recognition.

Stage one—image acquisition—is done using a digital camera and a colored glove, hand gestures are performed by participants. The stage of image processing for static and dynamic system is done based on the color system HSI (Hue, Saturation, and Intensity) and then segmentation process will take place to divide the image into six layers representing the five fingertips and wrist.

Next the feature extraction stage comes. This is implemented according to the color layers expanded from the segmented color regions. Thirty features are extracted, same features used by K. Assaleh and M. Al-Rousan [2], and grouped in one vector that represents a single gesture. In dynamic systems, however, same features will be used but with different preprocessing. Our proposed system tries to find the most effective way to decide on the beginning and the end of a gesture. Finally, stage four is concerned with using feedforward Neural Networks with back propagation and Recurrent Neural Networks for the gesture recognition purpose.

3. Image Processing

This stage includes image acquisition and segmentation.

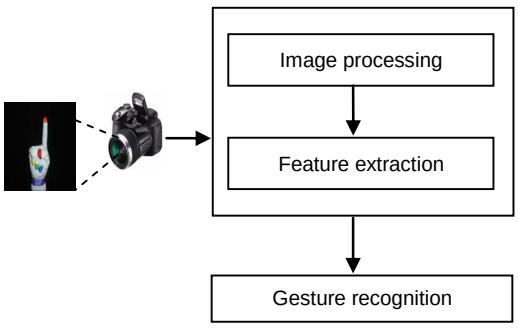

Figure 2. System stages.

Since colored images are being used in this research, the basic fundamentals of images, colors and color models will be introduced.

3.1. Image Basics

An image consists of a two-dimensional array of numbers. The color or gray shade displayed for a given picture element (pixel) depends on the number stored in the array for that pixel. The simplest type of image data is black and white. It is a binary image since each pixel is either 0 or 1. The next, more complex type of image data is gray scale; these images are black, white, and shades of gray. Each pixel here takes on a value between zero and the number of gray scales that the scanner can record [15]. The most complex type of image is color images. They have three channels corresponding to the colors red, green, and blue. Thus, each pixel has three values associated with it. Any color processing device uses red, green, and blue filters to produce those values [15].

3.2. Color Fundamentals

Color is the brain's reaction to a specific visual stimulus (*i.e.*, an object reflecting light of a certain wavelength) [16]. The ability of the human eye to distinguish colors is based upon the varying sensitivity of different cells in the retina to light of different wavelengths [17].

The human retina has three types of color photoreceptor cells, two of them, rods and cones receive light and transform it into image-forming signals which are transmitted through the optic nerve to the brain [17]. The third and more recently discovered category of photoreceptor cells is probably not involved in image-forming vision.

Rods are photoreceptor cells in the retina of the eye that are effective at extremely low light levels unlike the cone cells that function at bright light. Since rods are more light-sensitive, they are responsible for night vision whereas cones are less sensitive to light than the rod cells but allow the perception of color [17].

The signals from these color sensitive cells (cones), together with those from the rods (sensitive to intensity), combine in the brain to give "sensations" of different colors.

3.3. Color Spaces

Since color is subjective, there was a need to attribute numbers to help in the process of describing colors, either between people or between machines or programs. A color space is a method by which we can specify, create and visualize color [18].

There are many different ways of representing color; the most common way in computer graphics is to use a triplet of intensity values. Any unique combination of the

three values yields a distinct color. The three dimensional space which describes the distribution of physical colors is called a color space [18].

Color vectors in each of these spaces differ from one another; this difference means that two colors in one space being separated by one distance value would be separated by a different distance value in another space. A color represented in one space can be changed to another representation by performing some transformation.

The most common color model is the *RGB*; it is from this space that all other color spaces are derived. However the two color spaces that will be defined and used in this research are both *RGB* and *HSI*.

3.3.1. *RGB*

The *RGB* color model is commonly used in computer graphics and image processing applications. It is based on the physical (*i.e.*, wavelength) representation of the three primary colors; red, green and blue. An *RGB* color image can be represented as an M × N × 3 array of color pixels, where each color pixel is a triplet values corresponding to the red, green and blue component and is usually scaled by 255 for an 8-bit representation [19].

The *RGB* color space is usually shown graphically as an *RGB* color cube, as shown in **Figure 3**. The vertices of the cube are the primary (red, green and blue) and secondary (cyan, magenta and yellow) and any *RGB* color will be defined by its values on the three axes [20]. We can define any color simply by giving its red, green, and blue values, or coordinates, within the color cube [15].

However, it has the drawback that the luminance information is split across all three axes and it doesn't closely model the psychological understanding of color [19].

3.3.2. *HSI*

HSI color space was developed to be more intuitive in manipulating color and was designed to approximate the way human perceive and interpret color. Humans usually describe a color object by its hue, saturation and brightness. Hue is an attribute that describes a pure color as described by wavelength; saturation gives a measure of the degree to which a pure color is diluted by white light *(i.e.* the colorfulness of an area relative to its brightness), while intensity describes the color sensation (the overall brightness or the amount of light) [19]. **Figure 4** illus-

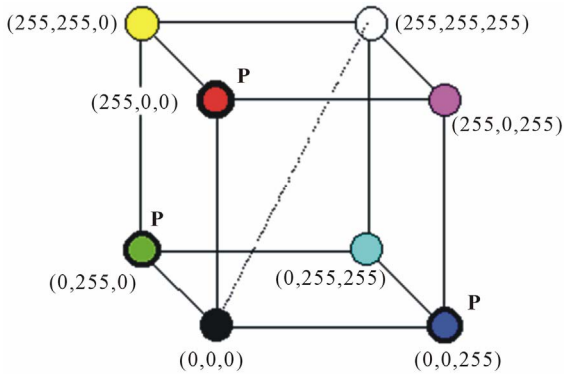

Figure 3. The *RGB* color space.

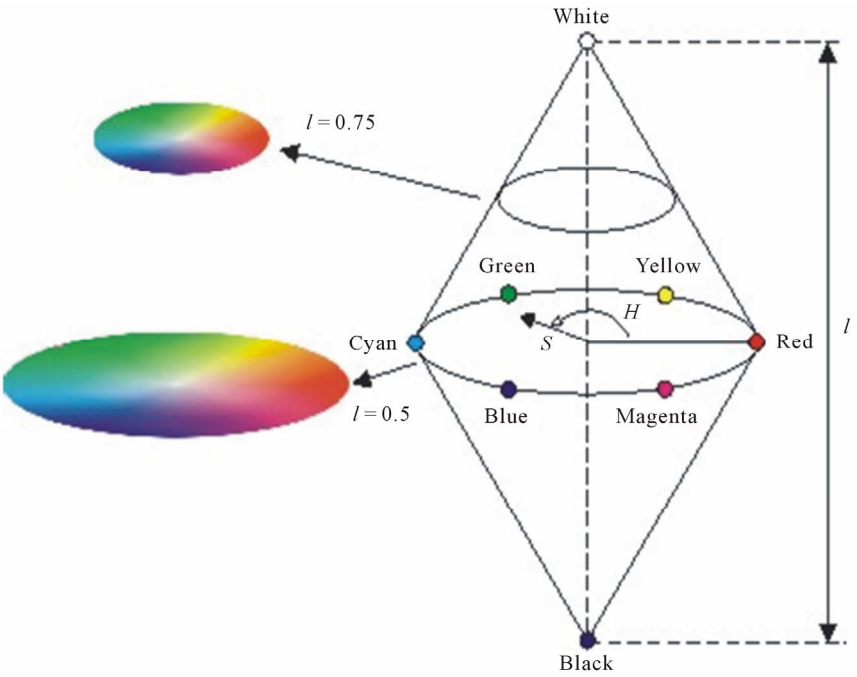

Figure 4. (Gonzalez [19]): The *HSI* color model based on circular color planes. The circles are perpendicular to the vertical intensity axis.

trates a common representation of this space. The cone shape has one central axis representing intensity. Along this axis are all the gray values, with black at the pointed end of the cone and white at its base. The greater the distance along this line from the pointed end, or origin, the brighter or higher the intensity. Since HSI color space decouples the intensity component from the color-carrying information (hue and saturation) in a color image; HSI is considered to be an ideal tool for developing image processing algorithms based on color description and will be adopted in our work [19].

3.3.3. Converting Colors from *RGB* to *HSI*

The transformation required to convert the *RGB* color system to *HSI* is as follows:

The *H* component of each *RGB* pixel is obtained using the equation:

$$H = \begin{cases} \theta \text{ if } B \le G \\ 360 - \theta \text{ if } B > G \end{cases}, \quad (1)$$

with

$$\theta = \arccos\left[\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}}\right]. \quad (2)$$

The saturation component is also obtained by using

$$S = 1 - \frac{3}{R+G+B}\left[\min(R,G,B)\right]. \quad (3)$$

Finally, the intensity component is given by:

$$I = \frac{R+G+B}{3}. \quad (4)$$

The above calculations are performed assuming that the *RGB* values have been normalized to the range [0,1], and that angle θ is measured with respect to the red axis of the HSI space.

3.4. Image Acquisition

Images were captured using a colored digital camera, each one of them was resized to 256 × 256 pixels, and image processing was performed using Matlab 6.5.

The image processing toolbox in Matlab represents color images as *RGB* (red, green, blue) values directly in an *RGB* image, which is then transformed to an *HSI* color space. The color segmentation is performed by grouping the pixels which are similar in some region property such as the *H* (Hue) value that defines the color. According to this, different color regions are determined by selecting the appropriate H value for each color.

The resulting value is six image layers representing the color areas of the five fingertips and the wrist as shown in **Figure 5**. To determine the best point that represents a color there was a need to use a clustering method which is The Fuzzy c-means.

4. Features Extraction

Image acquisition was done using a digital camera and a colored glove. 900 colored images have been used to represent the 30 different hand gestures and have been used as a training set; another 900 images have been taken and used as a test set.

Feature extraction stage is then implemented according to the color layers expanded from the segmented color regions, more specifically the colors centers that where determined using the FCM clustering method.

Based on the region centers; thirty features are extracted from each image, the same features which were adopted by K. Assaleh and M. Al-Rousan [2], and grouped in one vector that represents a single gesture.

The thirty features are based on the segmented color regions which are taken from the fingertips and their relative positions and orientations with respect to the wrist and to each other [2]. These features include the vectors from the center of each color region to the center of all other regions, and the angles between each of these vectors and the horizontal axis as shown in **Figure 6**.

To eliminate the effect of the distance separating the camera from the colored glove, the feature vector was normalized by dividing each entry of it by the maximum value of the whole vector [2].

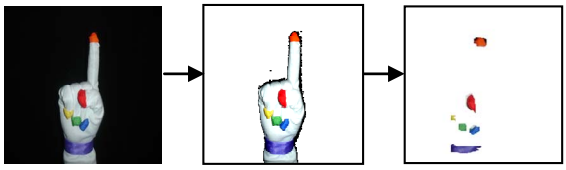

Figure 5. Color segmentation.

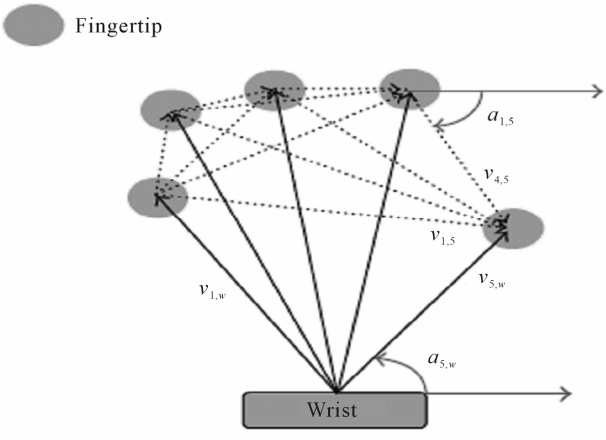

Figure 6. (Assaleh and Al-Rousan [2]): vectors-length and angles.

5. Neural Network Architectures

Two main types of neural networks architectures were used; Feedforward networks and recurrent networks.

5.1. Feed-Forward Neural Networks

Feedforward neural networks are the most popular and most widely used models in many applications. In this architecture each neuron in one layer has directed connections to the neurons of the subsequent layer; there are no links between neurons in the same layer neither with any of the previous layers [21]. The data flow in a strictly forward behavior. Since single-layer neural network has limited capabilities regarding pattern recognition; a multilayer feedforward neural network will be used.

Multilayer Feedforward Neural Networks

A multilayer feedforward network has a layered structure. The input layer where its neurons serve only as distribution points, one or more hidden layers of computation neurons, and the output layer [22].

Units in each layer receive their input from units from a layer directly below and send their output to units in a layer directly above the unit which means that the values only move from input to hidden to output layers; no values are fed back to earlier layers. Multilayer neural networks have proven their ability to solve many difficult problems such as pattern recognition as well as the ability of to extract more meaningful features from the input patterns through the use of hidden layers [22].

The multilayer network architecture chosen for this research is a three layers neural network, *i.e.* a network that has one hidden layer. Since each gesture is represented by a vector containing thirty features; the input layer has been chosen to have 30 input units. There is no rule for determining the number of nodes that the hidden layer should have; many simulations lead us to decide on its number; fifteen units. The output layer is 30 units since we have 30 gestures in the Arabic sign language, each output unit will represent one of the gestures.

Each unit in the input layer has been fully connected to every other unit in the second layer—the hidden layer. Also, every unit in the hidden layer is connected to every other neuron in the output layer in a feedforward behavior.

Figure 7 shows a model of the feedforward fully connected multilayer neural network that has been designed and tested for this research.

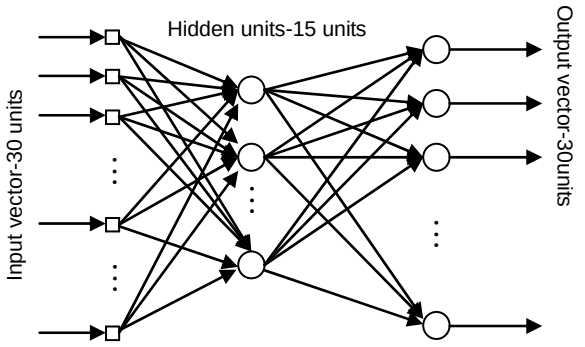

Figure 7. A three layers feedforward network.

5.2. Recurrent Neural Networks

Recurrent neural networks have been an interesting and important part of neural network research during the 1990's. They are designed to learn sequential or time varying patterns and have been applied to a wide variety of problems that involve many interesting human behaviors [23]. Applications for such networks vary from simple problems (temporal version of XOR) to discovering syntactic/semantic features for words [24].

Dynamic or recurrent neural networks differ from static neural networks since they are constructed to include feedback, or recurrent connections between the network layers and within the layer itself [25]. This feedback implies that the network has local memory characteristics that is able to store activity patterns and present those patterns to the network more than once, allowing the layer with feedback connections to use its own past activation in its preceding behavior [26].

So at any given time, the network output is calculated by propagating the input pattern forward through the network and the recurrent activations are propagated back to the extra layer—called the context layer—which copies the activation pattern from the output layer on the last instant. Different architectures can be created by adding recurrent connections at different points in the basic feedforward architecture [25]. Two of the most popular recurrent neural networks were introduced by Elman and Jordan are presented next followed by a combination of both; fully recurrent network.

5.2.1. Elman Neural Networks

In 1990, Jeff Elman introduced a recurrent neural network which uses context units. The Elman network is a two-layer network in which the hidden layer is recurrent; each node in the hidden layer is connected through a recurrent link with all the nodes in the hidden layer as well as the typical feedforward connections from each input node [27]. Therefore, at each time step the outputs of the hidden units are calculated, their values are used to compute the output of the network and are all stored as "extra inputs" (called context unit) to be used when the next time the network is operated.

The recurrent contexts provide a weighted sum of the previous values of the hidden units as input to the hidden units and thus provide the network with information

about previous activation values [28].

As shown in **Figure 8**, the activations are copied from hidden layer to context layer so that each context node is merely a copy of the previous activation of its corresponding hidden node [29]. This process allows the network to memorize information of its internal state for better patterns detection.

Same number of units as in previous feedforward neural network was used; 30 input units followed by 15 hidden units, and finally 30 output units.

5.2.2. Jordan Neural Networks

Jordan's network is similar to Elman neural network in that the hidden layer is recurrent; each context node (the copied nodes) has recurrent links to all of the other context nodes and it also has a feedback from the output layer to the hidden one, as shown in **Figure 9**. The activity of the output nodes is recurrently copied back into the context nodes, the recurrent connections between the context nodes appear to help the network to stabilize the network's and provide the network with a memory of its previous state [29].

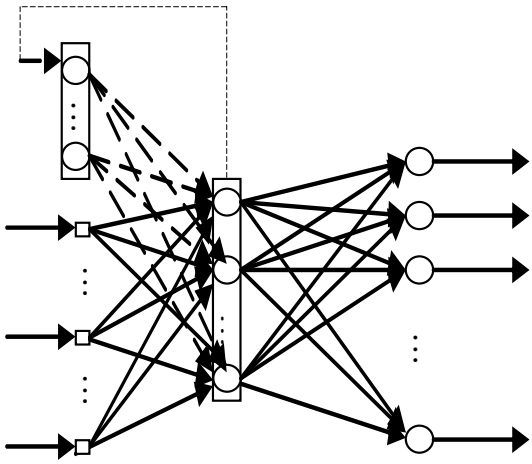

Figure 8. Elman neural network architecture demonstrated using the context layer.

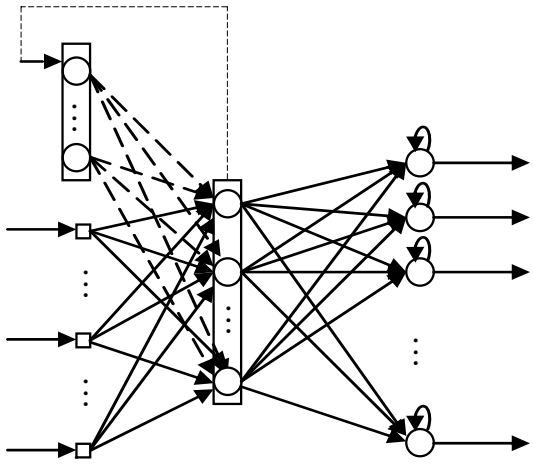

Figure 9. Jordan neural network architecture.

5.2.3. Fully Recurrent Neural Networks

In this type of networks both hidden and output layers are recurrent; moreover, a recurrent link exists to connect the output layer with the input layer (see **Figure 10**).

This architecture is expected to have more complex and time consuming computations since it has more complicated structure; but the existence of those extra recurrent links is supposed to allow the network to reach a better convergence to the optimal weights that leads the network to more accurate gesture recognition.

5.3. Training Algorithms

Neural networks learn to perform tasks by being presented with example input patterns and adjusting the weights on the connections between the nodes of the network. Many possible learning algorithms can be used to calculate the necessary weight changes [29].

Standard back-propagation was used to train the feedforward neural networks. Regarding the recurrent neural networks; similar algorithm was used. Simple Recurrent Networks (SRN) was used to train the Elman neural network while Backpropagation through Time (BPTT) was used to train both the Jordan and the fully recurrent network.

SRN method was developed to mimic the backpropagation but with a slight modification, Elman was the first to use it to train his recurrent network [29].

Backpropagation Through Time (BPTT) algorithm is another way to modify backpropagation to be able to work with time sequences on recurrent networks such as Jordan and fully recurrent neural networks. This method is based on converting the network from a feedback system to purely feedforward system by folding the network

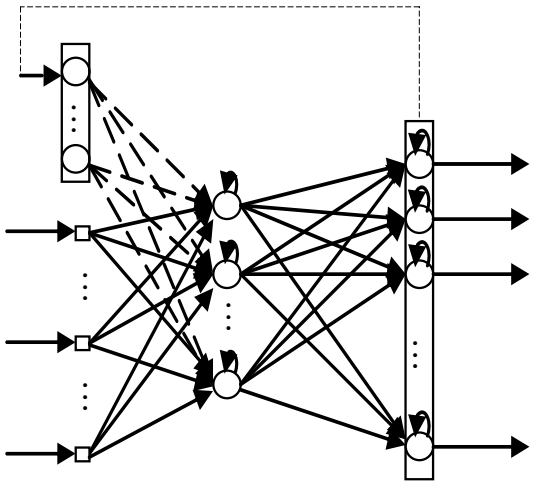

Figure 10. Fully recurrent neural network.

over time [30]. The network can then be trained if it is one large feedforward network with the modified weights being treated as shared weight [31].

5.4. Gesture Recognition Using Neural Networks for Static Gesture Recognition

5.4.1. Gesture Recognition Using Feedforward Networks

Figure 11 shows the way in which the sum of squared error declines until it reaches the target value. The training process was slow comparing to the recurrent neural networks. The recognition results where also poor using this type of architecture. One class for instance—Waw class—had zero recognition rate which was totally unacceptable for such a system. **Table 1** shows the recognition rate for each class.

5.4.2. Gesture Recognition Using Elman Neural Networks

A very recognizable improvement has been made when the system was implemented using recurrent neural network; specifically speaking, using Elman architecture. The accuracy rate has risen to 89.66%. But there are still some drawbacks in certain classes and can be noticed by the large error rate for that class and this has to do with the similarities among some hand gestures. **Figure 12** and **Table 2** both shows the Elman neural network recognition performance.

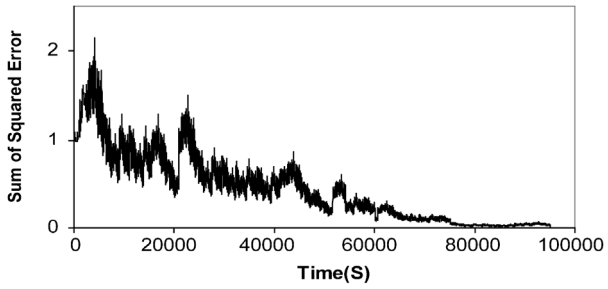

Figure 11. The sum of squared error during the feedforward training process.

Table 1. Recognition results using feedforward neural network.

Class	Error rate	Class	Error rate	Class	Error rate
alif	0/30	za	0/30	qaf	0/30
ba	3/30	sin	29/30	kaf	0/30
ta	0/30	shin	3/30	lam	0/30
tha	0/30	sad	28/30	mim	0/30
jim	14/30	dhad	17/30	nun	0/30
ha	0/30	tad	0/30	he	0/30
kha	16/30	zad	0/30	waw	30/30
dal	21/30	ayn	0/30	la	0/30
thal	0/30	gayn	25/30	ya	0/30
ra	0/30	fa	0/30	t	3/30

Total = 186/900 = 20.67%

Accuracy rate = 79.33%

5.4.3. Gesture Recognition Using Jordan Neural Networks

The results came a bit close to the previous one-Elman neural network- and with nearly the same number of epochs. **Figure 13** and **Table 3** both show the Jordan neural network recognition performance.

5.4.4. Gesture Recognition Using Fully Recurrent Neural Networks

The final architecture of recurrent neural network is the most promising type of the three previous architectures. That is because of the feedback links that lead the net-

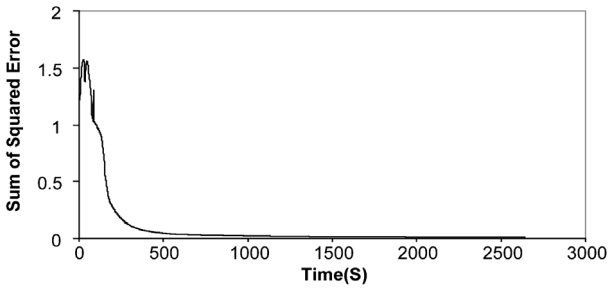

Figure 12. The sum of squared error during the training of Elman neural network.

Table 2. Recognition results using Elman neural network.

Class	Error rate	Class	Error rate	Class	Error rate
alif	0/30	za	0/30	qaf	0/30
ba	28/30	sin	0/30	kaf	0/30
ta	0/30	shin	2/30	lam	0/30
tha	0/30	sad	0/30	mim	0/30
jim	0/30	dhad	15/30	nun	0/30
ha	0/30	tad	19/30	he	0/30
kha	4/30	zad	0/30	waw	3/30
dal	0/30	ayn	0/30	la	0/30
thal	0/30	gayn	17/30	ya	0/30
ra	0/30	fa	0/30	t	5/30

Total = 93/900 = 10.33%

Accuracy rate = 89.66%

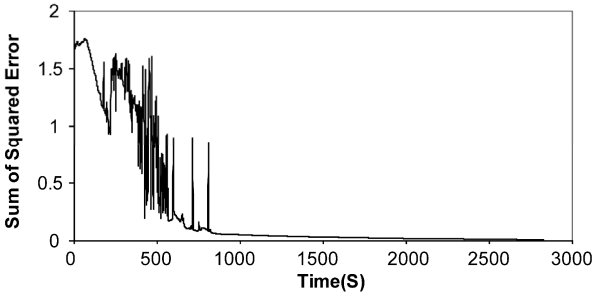

Figure 13. The sum of squared error during the training process of Jordan neural network.

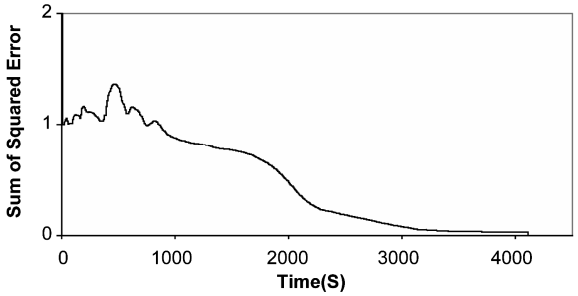

Figure 14. The sum of squared error during the training process of the fully recurrent neural network.

Table 3. Recognition results using Jordan neural network.

Class	Error rate	Class	Error rate	Class	Error rate
alif	0/30	za	0/30	qaf	0/30
ba	14/30	sin	6/30	kaf	3/30
ta	4/30	shin	3/30	lam	0/30
tha	6/30	sad	0/30	mim	2/30
jim	0/30	dhad	0/30	nun	0/30
ha	0/30	tad	19/30	he	6/30
kha	22/30	zad	0/30	waw	8/30
dal	25/30	ayn	0/30	la	0/30
thal	0/30	gayn	0/30	ya	0/30
ra	0/30	fa	0/30	t	21/30
Total = 139/900 = 15.44%					
Accuracy rate = 84.56%					

Table 4. Recognition results using fully recurrent neural network.

Class	Error rate	Class	Error rate	Class	Error rate
alif	0/30	za	0/30	qaf	0/30
ba	4/30	sin	0/30	kaf	0/30
ta	0/30	shin	3/30	lam	0/30
tha	0/30	sad	0/30	mim	0/30
jim	0/30	dhad	0/30	nun	0/30
ha	0/30	tad	0/30	he	0/30
kha	11/30	zad	0/30	waw	0/30
dal	0/30	ayn	0/30	la	0/30
thal	0/30	gayn	22/30	ya	0/30
ra	0/30	fa	0/30	t	4/30
Total = 44/900 = 4.89%					
Accuracy rate = 95.11%					

work to stability, convergence and the best accuracy rate among the explored types. **Figure 14** shows the reduction in the sum of squared error values whereas **Table 4** shows the recognition rate for each class.

6. Dynamic Gesture Recognition

The previous results suggest that the best architecture that can help in detecting hand gestures is the fully recurrent neural networks. Based on that, the dynamic process will be implemented using the later architecture. It will be used for recognizing gestures of frames extracted from a video stream of a moving hand performing sign language.

For the research purposes we will introduce a dynamic experiment that was held for video containing three gestures.

Frames were extracted at a rate of 15 frames per second. The idea behind identifying the dynamic gestures is simple; we will train the neural network using the static images as a training set; recall that the training database contains 900 images. Then the extracted images will be presented to the neural network to perform the recognition process, and in order to separate the characters there was a need to use a statistical measurement. It was noticed that the output of each neuron in the output layer of the i-th frame is close in value or related to the output of the same neuron of the (i + 1)-th frame only if they both belong to the same gesture. According to that the correlation value was computed for output values of the neural network for every two successive frames [31]. A correlation coefficient is a number between −1 and 1 which measures the degree to which two data sets are linearly related. Whenever the correlation value gets low comparing to the values around; it means that we have a gesture change. In particular, a correlation value of zero means that there is no correlation at all and hence a change in hand gesture. **Figure 15** shows the correlation values of the extracted frames. Since the number of the extracted frames is large; frame numbers in the image sample in **Figure 17** do not correspond to their actual

order appearance in the original video. Figures that show the correlation values between frames will reflect the actual frames order.

Choosing a zero threshold value can help in sensing the changes to decide on the beginning of a new character. **Figure 15** shows the suggested threshold value.

6.1. Words Recognition

To show how this procedure works for complete words; an experiment is conducted for the word Shams (شمس).

6.1.1. Testing Shams (شمس)

The word Shams consists of three gestures; Shin, Mim and Sin (**Figure 16**). A group of 120 frames were extracted from a 8 second length video (**Figure 17**). The correlation test in **Figure 18** was able to detect the two transitions that occur on frames 30 and 74.

7. Conclusions and Future Work

A static and dynamic hand gesture recognition system was presented in this research using a set of features that take into account the fingertips and the wrist along with orientation between them [32].

Two major network architectures have been presented; feedforward and recurrent neural networks along with a database consisting of 900 images formed by 2 persons performing 30 repetitions of each gesture has been used to train the neural networks.

The images have been captured by a color camera and digitized into 256 × 256 pixel images which have been converted into HSI system. Color segmentation was implemented using Matlab 6.5 and the hand gesture recog-

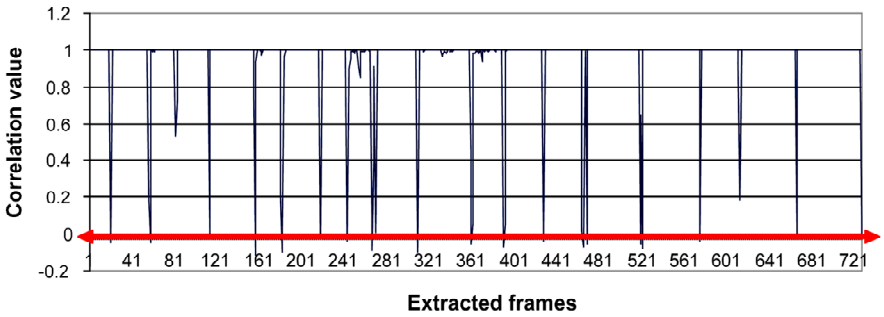

Figure 15. The threshold value that separates one gesture from another.

Figure 16. Classes from left: Shin, Sin and Mim.

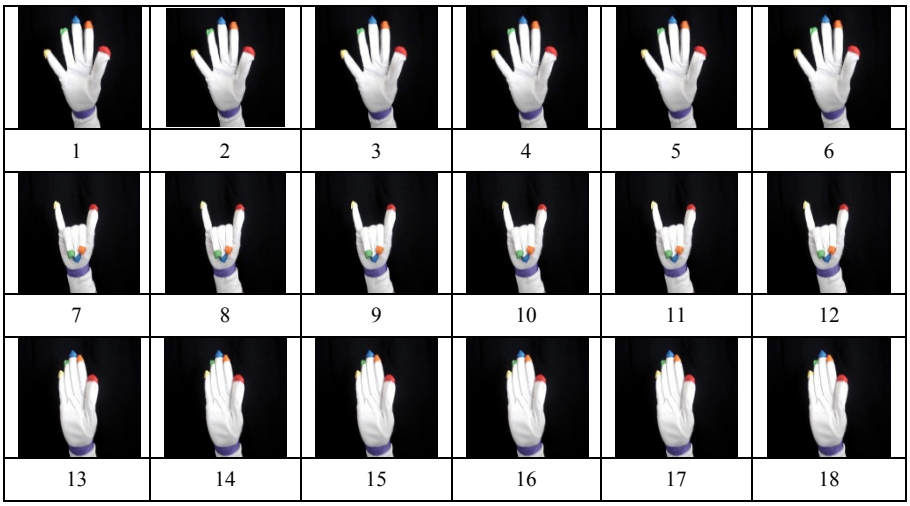

Figure 17. A sample of the extracted frames that contain transitions.

Figure 18. Correlation values which shows that transition has occurred on frames 30 and 74.

nition system was implemented in C.

The system has been tested using a test set of 900 images; represent all of the 30 gestures. The recognition rate for the static images under the described conditions has showed a significant improvement in the recognition rate that has reached up to 95.11% for the fully recurrent network; it performed better than the Elman's and Jordan's neural networks.

As for the dynamic system, using a simple statistical measure helped in determining the transition between different hand gestures and was useful in recognizing whole words in Arabic Sign Language (ArSL).

Some of the mistakes of the recognition process are due to false feature extraction and others are due to the similarities of some gestures; the results show an improvement in the generalisability of the system when using a fully recurrent network comparing with using other types of neural networks.

One of my major goals was speed and the avoidance of special hardware. This was achieved since the application was written in C. MATLAB is considered to be slower but allowed us to work faster and easier with image processing issues.

This research has also shown that it is feasible to utilize a data glove and neural network classifier to assist in human computer interaction applications. Regarding the future work; research will be devoted to the following topics:

- Using both hands may be a considerable improvement;
- The recognition of gesture sequences imposes the problem of detecting and eliminating unwanted intermediate gestures that might unintentionally be formed during the transition.

REFERENCES

[1] J. Eisenstein, S. Ghandeharizadeh, L. Golubchik, C. Shahabi, D. Yan and R. Zimmermann, "Device Independence and Extensibility in Gesture Recognition," *Proceedings of IEEE Virtual Reality*, Los Angeles, 22-26 March 2003, pp. 207-214. doi:10.1109/VR.2003.1191141

[2] K. Assaleh and M. Al-Rousan, "A New Method for Arabic Sign Language Recognition," *EURASIP Journal on Applied Signal Processing*, Hindawi Publishing Corporation, New York, 2005, pp. 2136-2145.

[3] K. Abe, H. Saito and S. Ozawa, "Virtual 3-D Interface System via Hand Motion Recognition from Two Cameras," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 32, No. 4, 2002, pp. 536-540. doi:10.1109/TSMCA.2002.804821

[4] J.-S. Kim, W. Jang and Z. Bien, "A Dynamic Gesture Recognition System for the Korean Sign Language (KSL)," *SMC-B*, Vol. 26, No. 2, 1996, pp. 354-359.

[5] G. Fang, W. Gao and D. Zhao, "Large Vocabulary Sign Language Recognition Based on Fuzzy Decision Trees," *SMC-A*, Vol. 34, No. 3, 2004, pp. 305-314.

[6] G. Fang, W. Gao and D. Zhao, "Large-Vocabulary Continuous Sign Language Recognition Based on Transition-Movement Models," *SMC-A*, Vol. 37, No. 1, 2007, pp. 1-9.

[7] T. Shanableh, K. Assaleh and M. Al-Rousan, "Spatio-Temporal Feature-Extraction Techniques for Isolated Ges ture Recognition in Arabic Sign Language," *SMC-B*, Vol. 37, No. 3, 2007, pp. 641-650.

[8] I. Infantino, R. Rizzo and S. Gaglio, "A Framework for Sign Language Sentence Recognition by Commonsense Context," *SMC-C*, Vol. 37, No. 5, 2007, pp. 1034-1039.

[9] Y. Yuan, Y. Liu and K. Barner, "Tactile Gesture Recognition for People with Disabilities," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, Kyoto, 18-23 March 2005, pp. 461-464.

[10] R. Salomon and J. Weissmann, "Evolutionary Tuning of Neural Networks for Gesture Recognition, Evolutionary Computation," *Proceedings of the* 2000 *Congress on Evolutionary Computation*, Vol. 2, La Jolla, 16-19 July 2000, pp. 1528-1534.

[11] F. Florez, J. Garcia and A. Hernandez, "Hand Gesture Recognition Following the Dynamics of a Topology-Preserving Network," *Proceedings of IEEE 5th International Conference on Automatic Face and Gesture Recognition*, Washington DC, 19-20 May 2002, pp. 303-308.

[12] M. Mohandes and S. Buraiky, "Automation of the Arabic

Sign Language Recognition using the PowerGlove," *AIML Journal*, Vol. 7, No. 1, 2007, pp. 41-46.

[13] K. Murakami and H. Taguchi, "Gesture Recognition Using Recurrent Neural Networks," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Gaithersburg, 15-17 March 1991, pp. 237-242.

[14] O. Al-Jarrah, A. Shatnawi and A. H. Halawani, "Recognition of Gestures in Arabic Sign Language using Neural Networks," *Proceeding of Artificial Intelligence and Soft Computing*, Palma De Mallorca, 28-30 August 2006.

[15] W. K. Pratt, "Digital Image Processing, PIKS Scientific Inside," 4th Edition, A John Wiley & Sons, Inc., Hoboken, 2007.

[16] A. Ford and A. Roberts, "Color Space Conversions," 1998. http://www.poynton.com/PDFs/coloureq.pdf

[17] A. Balaguer, "Analytical Methods for the Study of Color in Digital Images," 2006. http://www.tesisenxarxa.net/TESIS_UIB/AVAILABLE/TDX-1027106- 4116//tapb1de1.pdf

[18] S. Wesolkowski, "Color Image Edge Detection and Segmentation: A Comparison of the Vector Angle and the Euclidean Distance Color Similarity Measures," 1999. http://etd.uwaterloo.ca/etd/swesolko1999.pdf

[19] R. Gonzalez, R. Woods and S. Eddins, "Digital Image Processing Using MATLAB," Pearson Prentice Hall, Upper Saddle River, 2004.

[20] C. Y. Wen and C. M. Chou, "Color Image Models and Its Applications to Document Examination," *Forensic Science Journal*, Vol. 3, No. 1, 2004, pp. 23-32.

[21] J. Weissmann and R. Salomon, "Gesture Recognition for Virtual Reality Applications Using Data Gloves and Neural Networks," *Proceedings of IEEE International Joint Conference on Neural Networks*, Washington DC, 10-16 July 1999.

[22] S. Hykin, "Neural Networks: A Comprehensive Foundation," 2nd Edition, Prentice Hall, Upper Saddle River, 1998.

[23] L. C. Jain, "Recurrent Neural Networks," CRC Press, Boca Raton, 2001.

[24] J. L. Elman, "Finding Structure in Time," *Cognitive Science*, Vol. 14, No. 2, 1990, pp. 179-211. doi:10.1207/s15516709cog1402_1

[25] S. Ismail and A. bin Ahmad, "Recurrent Neural Network with Backpropagation through Time Algorithm for Arabic Recognition," *Proceeding* 18*th European Simulation Multiconference*, Magdeburg, 13-16 June 2004.

[26] N. Sinha, M. Gupta and D. Rao, "Dynamic Neural Networks: An Overview," *Proceedings of IEEE International Conference on Industrial Technology*, Vol. 1, Goa, 19-22 January 2000, pp. 491-496.

[27] S. Zappacosta, G. Baldassarre and S. Nolfi, "Elman Neural Networks and Time Integration for Object Recognition," Technical Report, Istituto di Scienze e Tecnologie della Cognizione, Roma, 2006.

[28] T. Chen and V. So, "A Comparative Study of Recurrent Neural Network Architectures on Learning Temporal Sequences," *IEEE International Conference on Digital Object Identifier*, Vol. 4, 1996, pp. 1945-1950.

[29] P. Vamplew, "Recognition of Sign Language Using Neural Networks," PhD Thesis, University of Tasmania, Newnham, 1996.

[30] P. Werbos, "Backpropagation through Time: What It Does and How To Do It," *Proceedings of the IEEE*, Vol. 78, No. 10, 1990, pp. 1550-1560. doi:10.1109/5.58337

[31] S. Ismail and A. bin Ahmad, "Recurrent Neural Network with Backpropagation through Time Algorithm for Arabic Recognition," *Proceedings* 18*th European Simulation Multiconference*, Magdeburg, 13-16 June 2004.

[32] M. Maraqa and R. Abu-Zaiter, "Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks," *Proceeding of the First International Conference on Applications of Digital Information and Web Technologies*, 2008, pp. 478-481.

