EURASIP Journal on Applied Signal Processing 2005:13, 2136–2145

Recognition of Arabic Sign Language Alphabet Using Polynomial Classifiers

Khaled Assaleh
Electrical Engineering Department, American University of Sharjah, P.O. Box 26666, Sharjah, UAE
Email: kassaleh@ausharjah.edu

M. Al-Rousan
Computer Engineering Department, Jordan University of Science and Technology, Irbid, Jordan
Email: malrousan@ausharjah.edu

Received 29 December 2003; Revised 31 August 2004

Building an accurate automatic sign language recognition system is of great importance in facilitating efficient communication with deaf people. In this paper, we propose the use of polynomial classifiers as a classification engine for the recognition of Arabic sign language (ArSL) alphabet. Polynomial classifiers have several advantages over other classifiers in that they do not require iterative training, and that they are highly computationally scalable with the number of classes. Based on polynomial classifiers, we have built an ArSL system and measured its performance using real ArSL data collected from deaf people. We show that the proposed system provides superior recognition results when compared with previously published results using ANFIS-based classification on the same dataset and feature extraction methodology. The comparison is shown in terms of the number of misclassified test patterns. The reduction in the rate of misclassified patterns was very significant. In particular, we have achieved a 36% reduction of misclassifications on the training data and 57% on the test data.

Keywords and phrases: Arabic sign language, hand gestures, feature extraction, adaptive neuro-fuzzy inference systems, polynomial classifiers.

1. INTRODUCTION

Signing has always been part of human communications. The use of gestures is not tied to ethnicity, age, or gender. Infants use gestures as a primary means of communication until their speech muscles are mature enough to articulate meaningful speech. For millennia, deaf people have created and used signs among themselves. These signs were the only form of communication available for many deaf people. Within the variety of cultures of deaf people all over the world, signing evolved to form complete and sophisticated languages. These languages have been learned and elaborated by succeeding generations of deaf children.

Normally, there is no problem when two deaf persons communicate using their common sign language. The real difficulties arise when a deaf person wants to communicate with a nondeaf person. Usually both will get frustrated in a very short time. For this reason, there have been several attempts to design smart devices that can work as interpreters between the deaf people and others. These devices are categorized as human-computer-interaction (HCI) systems. Existing HCI devices for hand gesture recognition fall into two categories: glove-based and vision-based systems. The glove-based system relies on electromechanical devices that are used for data collection about the gestures [1, 2, 3, 4, 5]. Here the person must wear some sort of wired gloves that are interfaced with many sensors. Then based on the readings of the sensors, the gesture of the hand can be recognized by a computer interfaced with the sensors. Because glove-based systems force the user to carry a load of cables and sensors, they are not completely natural the way an HCI should be. The second category of HCI systems has overcome this problem. Vision-based systems basically suggest using a set of video cameras, image processing, and artificial intelligence to recognize and interpret hand gestures [1]. These techniques are utilized to design visual-based hand gesture systems that increase the naturalness of human-computer interaction. The main attraction of such systems is that the user is not plagued with heavy wired gloves and has more freedom and flexibility. This is accomplished by using specially designed gloves with visual markers that help in determining hand posters, as presented in [6, 7, 8]. A good review about vision-based systems can be found in [9].

Once the data has been obtained from the user, the recognition system, whether it is glove-based or vision-based, must use this data for processing to identify the gesture.

Several approaches have been used for hand gestures recognition including fuzzy logic, neural networks, neuro-fuzzy, and hidden Markov model. Lee et al. have used fuzzy logic and fuzzy min-max neural networks techniques for Korean sign language recognition [10]. They were able to achieve a recognition rate of 80.1% using gloved-based system. Recognition based on fuzzy logic suffers from the problem of a large number of rules needed to cover all features of the gestures. Therefore, such systems give poor recognition rate when used for large systems with high number of rules. Neural networks, HMM [11, 12], and adaptive neuro-fuzzy inference systems (ANFIS) [13, 14] were also widely used in recognition systems.

Recently, finite state machine (FSM) has been used in several works as an approach for gesture recognition [7, 8, 15]. Davis and Shah [8] proposed a method to recognize human-hand gestures using a model-based approach. A finite state machine is used to model four qualitatively distinct phases of a generic gesture: static start position, for at least three video frames; smooth motion of the hand and fingers until the end of the gesture; static end position, for at least three video frames; smooth motion of the hand back to the start position. Gestures are represented as a sequence of vectors and are then matched to the stored gesture vector models using table lookup based on vector displacements. The system has very limited gesture vocabularies and uses marked gloves as in [7]. Many other systems used FSM approach for gesture recognition such as [15]. However, the FSM approach is very limited and is really a posture recognition system rather than a gesture recognition system. According to [15] FSM has, in some of the experiments, gone prematurely into the wrong state, and in such situations, it is difficult to get it back into a correct state.

Even though Arabic is spoken in a wide spread geographical and demographical part of the world, the recognition of ArSL has received little attention from researchers. Gestures used in ArSL are depicted in Figure 1. In this paper, we introduce an automatic recognition system for Arabic sign language using the polynomial classifier. Efficient classification methods using polynomial classifiers have been introduced by Campbell and Assaleh (see [16, 17, 18]) in the fields of speech and speaker recognition. It has been shown that the polynomial technique can provide several advantages over other methods (e.g., neural network, hidden Markov models, etc.). These advantages include computational and storage requirements and recognition performance. More details about polynomial recognition technique are given in Section 5. In this work we have built, tested, and evaluated an ArSL recognition system using the same set of data used in [6, 19]. The recognition performance of the polynomial-based system is compared with that of the ANFIS-based system. We have found that our polynomial-based system largely outperforms the ANFIS-based system.

This paper is organized as follows. Section 2 describes the concept of ANFIS systems. Section 3 describes our database and shows how segmentation and feature extraction are performed. Since we will be comparing our results to those obtained by ANFIS-based systems, in Section 4 we briefly describe the ANFIS model as used in ArSL [6, 19]. The theory and implementation of polynomial classifiers are discussed in Section 5. Section 6 discusses the results obtained from the polynomial-based system and compares them with the ANFIS-based system where the superiority of the former is demonstrated. Finally, we conclude in Section 7.

2. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM

Adjusting the parameters of fuzzy inference system (FIS) proves to be a tedious and difficult task. The use of ANFIS can lead to a more accurate and sophisticated system. ANFIS [14] is a supervised learning algorithm, which equips FIS with the ability to learn and adapt. It optimizes the parameters of a given fuzzy inference system by applying a learning procedure using a set of input-output pairs, the training data. ANFIS is considered to be an adaptive network which is very similar to neural networks [20]. Adaptive networks have no synaptic weights, instead they have adaptive and nonadaptive nodes. It must be said that an adaptive network can be easily transformed to a neural network architecture with classical feedforward topology. ANFIS is an adaptive network that works like adaptive network simulator of the Takagi-Sugeno fuzzy [20] controllers. This adaptive network has a predefined adaptive network topology as shown in Figure 2. The specific use of ANFIS for ArSL alphabet recognition is detailed in Section 4.

The ANFIS architecture shown in Figure 2 is a simple architecture that consists of five layers with two inputs x and y and one output z. The rule base for such a system contains two fuzzy if-then rules of the Takagi and Sugeno type.

(i) Rule 1: if x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$.
(ii) Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$.

A and B are the linguistic labels (called quantifiers).

The node functions in the same layer are of the same function family as described below: for the first layer, the output of node i is given as

$$O_{1,i} = \mu_{A_i}(x) = \frac{1}{1 + ((x - c_i)/a_i)^{2b_i}}. \quad (1)$$

The output of this layer specifies the degree to which the given input satisfies the quantifier. This degree can be specified by any appropriate parameterized membership function. The membership function used in (1) is the generalized bell function [20] which is characterized by the parameter set $\{a_i, b_i, c_i\}$. Tuning the values of these parameters will vary the membership function and in turn changes the behavior of the FIS. The parameters in layer 1 of the ANFIS model are known as the premise parameters [20].

The output function, $O_{1,i}$ is input into the second layer. A node in the second layer multiplies all the incoming signals and sends the product out. The output of each node represents the firing strength of the rules introduced in layer 1 and is given as

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y). \quad (2)$$

Figure 1: Gestures of Arabic sign language (ArSL).

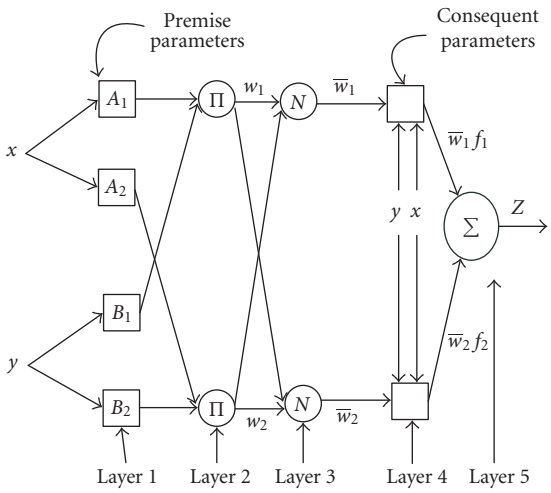


Figure 2: ANFIS model.

In the third layer, the normalized firing strength is calculated by each node. Every node (i) will calculate the ratio of the ith rule firing strength to the sum of all rules' firing strengths as shown below:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2}. \tag{3}$$

The node function in layer 4 is given as

$$O_{4,i} = \overline{w}_i f_i, \tag{4}$$

where f_i is calculated based on the parameter set $\{p_i, q_i, r_i\}$ and is given by

$$f_i = p_i x + q_i y + r_i. \tag{5}$$

Similar to the first layer, this is an adaptive layer where the output is influenced by the parameter set. Parameters in this layer are referred to as consequent parameters.

Finally, layer 5 consists of only one node that computes the overall output as the summation of all incoming signals:

$$O_{5,1} = \sum \overline{w}_i f_i. \tag{6}$$

For the model described in Figure 2, and using (4) and (5) in (6), the overall output is given by

$$O_{5,1} = \frac{w_1(p_1 x + q_1 y + r_1) + w_2(p_2 x + q_2 y + r_{21})}{w_1 + w_2}. \tag{7}$$

As mentioned above, there are premise parameters and consequent parameters for the ANFIS model. The number of these parameters determines the size and complexity of the ANFIS network for a given problem. The ANFIS network must be trained to learn about the data and its nature. During the learning process the premise and consequent parameters are tuned until the desired output of the FIS is reached.

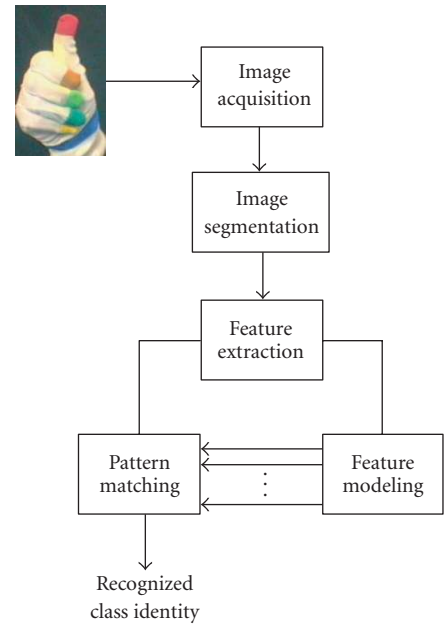


Figure 3: Stages of the recognition system.

3. ArSL DATABASE COLLECTION AND FEATURE EXTRACTION

In this section we briefly describe and discuss the database and feature extraction of the ArSL recognition system introduced in [6]. We do so because our proposed system shares the same exact processes up to the classification step where we introduce our polynomial-based classification. The system is comprised of several stages as shown in Figure 3. These stages are image acquisition, image processing, feature extraction, and finally, gesture recognition. In the image acquisition stage, the images were collected from thirty deaf participants. The data was collected from a center for deaf people rehabilitation in Jordan. Each participant had to wear the colored gloves and perform Arabic sign gestures in his/her way. In some cases, participants have provided more than one gesture for the same letter. The number of samples and gestures collected from the involved participants is shown in Table 1. It should be noted that there are 30 letters (classes) in Arabic sign language that can be represented in 42 gestures. The total number of samples collected for training and testing taken from a total of 42 gestures (corresponding to 30 classes) is 2323 samples partitioned into 1625 for training and 698 for testing. In Table 1, one can notice that the number of the collected samples is not the same for all classes due to two reasons. First, some letters have more than one gesture representation, and second, because the data was collected over a few months and not all participants were available all the time. For example, one of the multiple gesture representations can be seen in Figure 1 for the alphabet "thal."

The gloves worn by the participants were marked with six different colors at different six regions as shown in Figure 4a. Each acquired image is fed to the image processing stage in which color representation and image segmentation are performed for the gesture. By now, the color of each pixel in the

Table 1: Number of patterns per letter for training and testing data.

Class	Number of training samples	Number of test samples	Number of gestures
Alif أ	33	14	1
Ba ب	58	21	1
Ta ت	51	21	1
Tha ث	48	19	1
Jim ج	38	18	1
Ha ح	42	20	1
Kha خ	69	26	2
Dal د	112	32	3
Thal ذ	77	35	3
Ra ر	71	22	2
Za ز	66	24	2
Sin س	36	17	1
Shin ش	37	21	1
Sad ص	54	19	1
Dhad ض	49	16	1
Tad ط	68	27	2
Zad ظ	74	29	2
Ayn ع	39	18	1
Gayn غ	82	36	1
Fa ف	74	27	2
Qaf ق	37	21	1
Kaf ك	41	34	1
Lam ل	68	19	1
Mim م	38	19	1
Nun ن	51	23	2
He ه	36	21	1
Waw و	59	22	2
La لا	42	22	1
Ya ي	39	33	1
T ة	36	22	1
Total	1625	698	42

image is represented by three values for red, green, and blue (RGB). For more efficient color representation, RGB values are transformed to hue-saturation-intensity (HSI) representation. In the image segmentation stage, the color information is used for segmenting the image into six regions representing the five fingertips and the wrist. Also the centroid for each region is identified in this stage as illustrated in Figure 4b.

In the feature extraction stage, thirty features are extracted from the segmented color regions. These features are taken from the fingertips and their relative positions and orientations with respect to the wrist and to each other as shown in Figure 5. These features include the vectors from the center of each region to the center of all other regions, and the angles between each of these vectors and the horizontal axis. More specifically, there are five vectors (length and angle) between the centers of fingertip regions and the wrist region $(v_{i,w}, a_{i,w})$ where $i = 1, 2, \ldots, 5$; and another ten vectors between the centers of the fingertip regions of each pair of fingers $(v_{i,j}, a_{i,j})$ where $i = 1, 2, \ldots, 5$, $j = 1, 2, \ldots, 5$, and $i \neq j$.

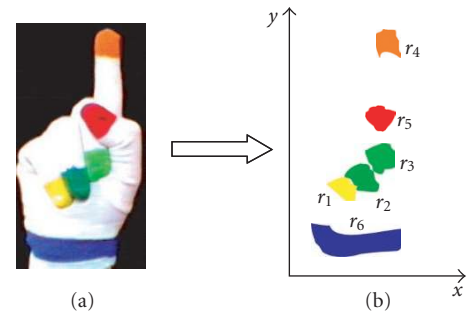


Figure 4: (a) colored glove and (b) output of image segmentation.

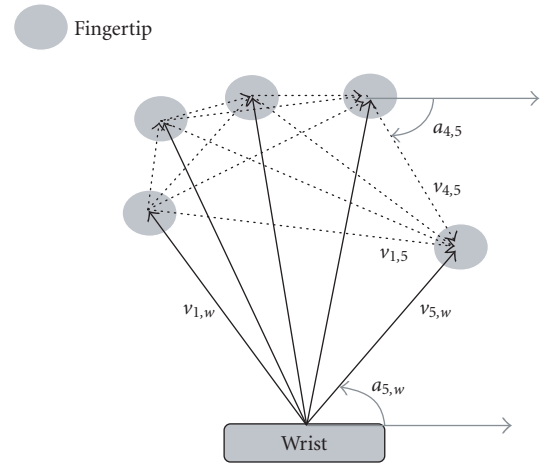


Figure 5: Vectors (lengths and angles) representing the feature set.

Since the length and the angle of each of the 15 vectors are used, thirty features are extracted from a given gesture as shown in Table 2. Therefore, a feature vector $\mathbf{x}$ is constructed as $\mathbf{x} = [v_{1,w}, a_{1,w}, \ldots, v_{5,w}, a_{5,w}, v_{1,2}, a_{1,2}, \ldots, v_{4,5}, a_{4,5}]$.

It is worth mentioning that the lengths of all vectors are normalized so that the calculated values are not sensitive to the distance between the camera and the person communicating with the system. The normalization is done per feature vector where the measured vector lengths are divided by the maximum value among the 15 vector lengths.

4. ANFIS-BASED ArSL RECOGNITION

The last stage of the ArSL recognition system introduced in [6] is the classification stage. In this stage they constructed 30 ANFIS units representing the 30 ArSL finger-spelling gestures. Each ANFIS unit is dedicated to one gesture. As shown in Figure 6, each ANFIS unit has 30 inputs corresponding to elements in the set of features that has been extracted in the previous stage. Like the model described in Figure 2, the ANFIS model consists of five layers. For the adaptive input layer, layer 1, Gaussian membership functions of the form

$$\mu(x) = e^{-((x-c)/\sigma)^2} \tag{8}$$

are used, where c and σ are tunable parameters which form the premise parameters in the first layer. Building the rules

Table 2: Calculated features: vectors and angles between the six regions.

Feature		Region centers considered
Vector	Angle	
$v_{1,w}$	$a_{1,w}$	1st fingertip (little finger) and wrist
$v_{2,w}$	$a_{2,w}$	2nd fingertip and wrist
$v_{3,w}$	$a_{3,w}$	3rd fingertip and wrist
$v_{4,w}$	$a_{4,w}$	4th fingertip and wrist
$v_{5,w}$	$a_{5,w}$	5th fingertip and wrist
$v_{1,2}$	$a_{1,2}$	1st and 2nd fingertips
$v_{1,3}$	$a_{1,3}$	1st and 3rd fingertips
$v_{1,4}$	$a_{1,4}$	1st and 4th fingertips
$v_{1,5}$	$a_{1,5}$	1st and 5th fingertips
$v_{2,3}$	$a_{2,3}$	2nd and 3rd fingertips
$v_{2,4}$	$a_{2,4}$	2nd and 4th fingertips
$v_{2,5}$	$a_{2,5}$	2nd and 5th fingertips
$v_{3,4}$	$a_{3,4}$	3rd and 4th fingertips
$v_{3,5}$	$a_{3,5}$	3rd and 5th fingertips
$v_{4,5}$	$a_{4,5}$	4th and 5th fingertips

for each gesture is done based on the use of subtractive clustering algorithm and least-squares estimator techniques [6, 21].

5. POLYNOMIAL CLASSIFIERS

The problem that we are considering here is a closed set identification problem which involves finding the best matching class given a list of classes (and their models obtained in the training phase) and feature vectors from an unknown class.

In general, the training data for each class consists of a set of feature vectors extracted from multiple observations corresponding to that class. Depending on the nature of the recognition problem, an observation could be represented by a single feature vector or by a sequence of feature vectors corresponding to the temporal or spatial evolution of that observation. In our case, each observation is represented by a single feature vector representing a hand gesture. For each class, i, we have a set of N_i training observations represented by a sequence of N_i feature vectors $[\mathbf{x}_{i,1}\ \mathbf{x}_{i,2}\ \cdots\ \mathbf{x}_{i,N_i}]^t$.

Identification requires the decision between multiple hypotheses, H_i. Given an observation feature vector $\mathbf{x}$, the Bayes decision rule [22] for this problem reduces to

$$i^{\text{opt}} = \arg\max_i p(H_i|\mathbf{x}), \tag{9}$$

with the assumption that $p(\mathbf{x})$ is the same for all observation feature vectors.

A common method for solving (9) is to approximate an ideal output on a set of training data with a network. That is, if $\{f_i(\mathbf{x})\}$ are discriminant functions [23], then we train $f_i(\mathbf{x})$ to an ideal output of 1 on all in-class observation feature vectors and 0 on all out-of-class observation feature vectors. If f_i is optimized for mean-squared error over all possible functions such that

$$f_i^{\text{opt}} = \arg\min_{f_i} E_{\mathbf{x},H}\left\{|f_i(x) - y_i(\mathbf{x},H)|^2\right\}, \tag{10}$$

then the solution entails that $f_i^{\text{opt}} = p(H_i|\mathbf{x})$, see [22]. In (10), $E_{\mathbf{x},H}$ is the expectation operator over the joint distribution of $\mathbf{x}$ and all hypotheses, and $y_i(\mathbf{x},H)$ is the ideal output for H_i. Thus, the least-squares optimization problem gives the functions necessary for the hypothesis test in (9). If the discriminant function in (10) is allowed to vary only over a given class (in our case polynomials with a limited degree), then the optimization problem of (10) gives an *approximation* of the a posteriori probabilities [23]. Using the resulting polynomial approximation in (9) thus gives an approximation to the ideal Bayes rule.

The basic embodiment of a Kth-order polynomial classifier consists of several parts. In the training phase, the elements of each training feature vector, $\mathbf{x} = [x_1, x_2, \ldots, x_M]$, are combined with multipliers to form a set of basis functions, $\mathbf{p}(\mathbf{x})$. The elements of $\mathbf{p}(\mathbf{x})$ are the monomials of the form

$$\prod_{j=1}^{M} x_j^{k_j}, \tag{11}$$

where k_j is a positive integer, and $0 \le \sum_{j=1}^{M} k_j \le K$. The sequence of feature vectors $[\mathbf{x}_{i,1}\mathbf{x}_{i,2}\cdots\mathbf{x}_{i,N_i}]^T$ representing class i is expanded into

$$\mathbf{M}_i = \left[\mathbf{p}(\mathbf{x}_{i,1}) \quad \mathbf{p}(\mathbf{x}_{i,2}) \quad \cdots \quad \mathbf{p}(\mathbf{x}_{i,N_i})\right]^t. \tag{12}$$

Expanding all the training feature vectors results in a global matrix for all C classes obtained by concatenating all the individual $\mathbf{M}_i$ matrices such that

$$\mathbf{M} = \left[\mathbf{M}_1 \ \ \mathbf{M}_2 \ \ \cdots \ \ \mathbf{M}_C\right]^t. \tag{13}$$

Once the training feature vectors are expanded into their polynomial basis terms, the polynomial classifier is trained to approximate an ideal output using mean-squared error as the objective criterion.

The training problem reduces to finding an optimum set of weights, $\mathbf{w}_i$, that minimizes the distance between the ideal outputs and a linear combination of the polynomial expansion of the training data such that

$$\mathbf{w}_i^{\text{opt}} = \arg\min_{\mathbf{w}_i} \|\mathbf{M}\mathbf{w}_i - \mathbf{o}_i\|_2, \tag{14}$$

where $\mathbf{o}_i$ represents the ideal output comprised of the column vector whose entries are N_i ones in the rows where the ith class's data is located in $\mathbf{M}$, and zeros otherwise.

The weights (models) $\mathbf{w}_i^{\text{opt}}$ can be obtained explicitly (noniteratively) by applying the normal equations method [24]:

$$\mathbf{M}^t\mathbf{M}\mathbf{w}_i^{\text{opt}} = \mathbf{M}^t\mathbf{o}_i. \tag{15}$$

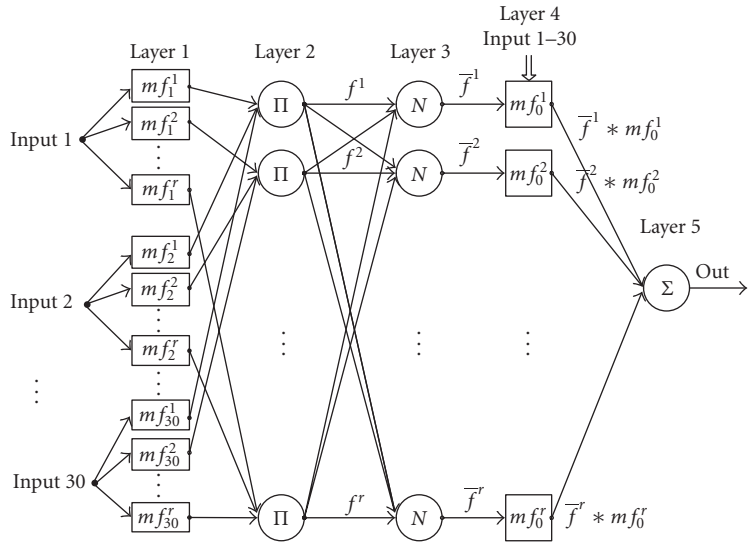


Figure 6: One-gesture ANFIS unit.

Define **1** to be the vector of all ones. We rearrange (15) to

$$\sum_{j=1}^{C} \mathbf{M}_j^t \mathbf{M}_j \mathbf{w}_i^{\text{opt}}{}_i = \mathbf{M}_i^t \mathbf{1}. \tag{16}$$

If we define $\mathbf{R}_j = \mathbf{M}_j^t \mathbf{M}_j$, $\mathbf{R} = \sum_{j=1}^{C} \mathbf{R}_j$, and $\mathbf{m} = \mathbf{M}_i^t \mathbf{1}$, then (10) yields an explicit solution for $\mathbf{w}_i^{\text{opt}}$ expressed as

$$\mathbf{w}_i^{\text{opt}} = \mathbf{R}^{-1}\mathbf{m}. \tag{17}$$

This suggests that the straightforward method of finding $\mathbf{w}_i^{\text{opt}}$ is by inverting the $\mathbf{R}$ matrix which represents the main bulk of the computational complexity of the training process. However, in [16, 17, 18] Campbell and Assaleh discuss the computational aspects of solving for $\mathbf{w}_i^{\text{opt}}$ and they present a fast method for training polynomial classifiers by exploiting the redundancy in the $\mathbf{R}_j$ matrices. They also discuss in detail the computational and storage advantages of their training method.

In the recognition stage when an unknown feature vector, $\mathbf{x}$, is presented to all C models, the vector is expanded into its polynomial terms $\mathbf{p}(\mathbf{x})$ (similar to what was done in the training phase) and a set of C scores $\{s_i\}$ are computed. The class c to which the vector $\mathbf{x}$ belongs is the index of the maximum score such that

$$c = \arg\max_i s_i, \tag{18}$$

where

$$s_i = \mathbf{w}_i^{\text{opt}} \mathbf{p}(\mathbf{x}). \tag{19}$$

The Kth-order polynomial expansion of an M-dimensional vector $\mathbf{x}$ generates an $O_{M,K}$-dimensional vector $\mathbf{p}(\mathbf{x})$.

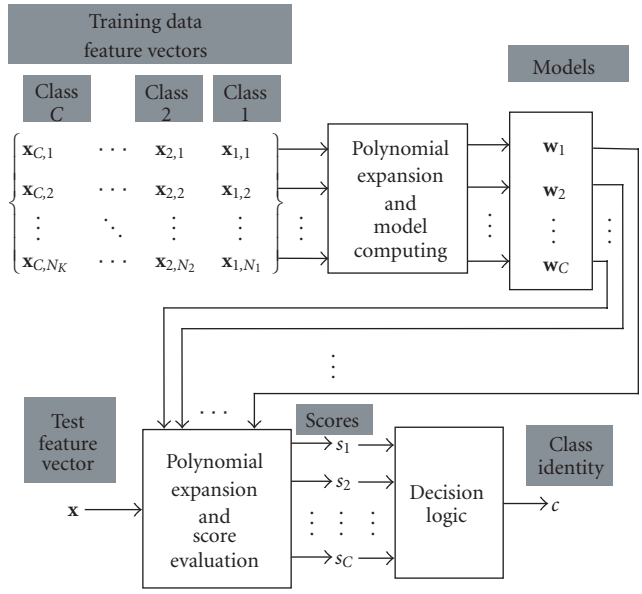


Figure 7: Block diagram of training and testing polynomial classifiers.

$O_{M,K}$ is a function of both M and K and can be expressed as $O_{M,K} = 1 + KM + \sum_{l=2}^{K} C(M,l)$, where $C(M,l) = \binom{M}{l}$ is the number of distinct subsets of l elements that can be made out of a set of M elements. This suggests that for a relatively high value of M, one is restricted to low-order polynomial expansions such as 2nd or 3rd order. In our case for $M = 30$, we found that a 2nd-order polynomial expansion is sufficient. Higher-order expansions were not found to provide any further performance improvements.

The block diagram for the training and identification via polynomial classifiers is depicted in Figure 7.

Table 3: Error rates of the system.

Class	Error rate for training data	Error rate for test data	Number of gestures
Alif أ	0/33	0/14	1
Ba ب	8/58	5/21	1
Ta ت	0/51	2/21	1
Tha ث	0/48	0/19	1
Jim ج	1/38	1/18	1
Ha ح	1/42	0/20	1
Kha خ	0/69	1/26	2
Dal د	2/112	5/32	3
Thal ذ	0/77	0/35	3
Ra ر	7/71	9/22	2
Za ز	2/66	1/24	2
Sin س	0/36	0/17	1
Shin ش	0/37	0/21	1
Sad ص	2/54	5/19	1
Dhad ض	0/49	1/16	1
Tad ط	0/68	0/27	2
Zad ظ	0/74	1/29	2
Ayn ع	0/39	0/18	1
Gayn غ	0/82	0/36	1
Fa ف	0/74	1/27	2
Qaf ق	0/37	1/21	1
Kaf ك	0/41	0/34	1
Lam ل	0/68	5/19	1
Mim م	0/38	3/19	1
Nun ن	1/51	2/23	2
He ه	0/36	0/21	1
Waw و	1/59	0/22	2
La لا	0/42	1/22	1
Ya ي	0/39	0/33	1
T ة	1/36	2/22	1
Total	26/1625 = 1.6%	46/698 = 6.59%	42

6. RESULTS AND DISCUSSION

We have applied the training method of the polynomial classifier as described above by creating one 2nd-order polynomial classifier per class, resulting in a total of 42 networks. The feature vectors for the training data set are expanded into their polynomial terms, and the corresponding class labels are assigned accordingly before they were processed through the training algorithm outlined in (12) through (17). Consequently, each class i is represented by the identification model $\mathbf{w}_i^{\text{opt}}$. Therefore, alphabets with multiple gestures were represented by multiple models.

After creating all the identification models, we have conducted two experiments to evaluate the performance of our polynomial-based system. The first experiment was for evaluating the training data itself, and the second was for evaluating the test data set. In the first experiment, the performance of the system is found to be superior as is usually expected when the same training data is used as test data. The system has resulted in 26 misclassifications out of 1625 patterns. This corresponds to 1.6% error rate, or to a recognition rate of 98.4%. The detailed per-class misclassifications are shown in Table 3.

However, the appropriate indicative way of measuring the performance of a recognition system is to present it with a data set different from what it was trained with. This is exactly what we have done in the second experiment when we used a test data which has not been used in the training process. This test data set is comprised of 698 samples distributed among classes as shown in Table 1. Our recognition system has shown an excellent performance with a low error rate of 6.59% corresponding to a recognition rate of 93.41% as indicated in Table 3.

The results of our polynomial-based recognition system are considered superior over previously published results in the field of ArSL [6, 13, 19]. A direct and fair comparison can be done with our previous papers [6, 19] in which we have used exactly the same data sets and features for training and testing using ANFIS-based classification as described in Section 3. Both systems are found to perform very well on the training data. Nevertheless, the polynomial-based system still performs better than the ANFIS-based system as it results in 26 misclassifications compared to 41 in the ANFIS-based system. This corresponds to a 36% reduction in the misclassifications and hence in the error rate.

More importantly, the polynomial-based recognition provides a major reduction in the number of misclassified patterns when compared with the ANFIS-based system in the case of the test data set. In this case, the number of misclassifications is reduced from 108 to 46 which corresponds to a very significant reduction of 57%. These results are shown in Table 4.

The misclassification errors are attributed to the similarity among the gestures that some users provide for different letters. For example, Table 3 shows that a few letters such as the ba, ra, and dal have higher error rates. A close examination of their images explains this phenomenon as shown in Figure 8.

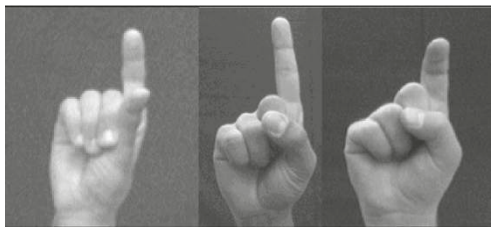
It is worth mentioning that the above results are obtained using all the collected samples from all the gestures. However, in [6] some of the multiple gesture data was excluded to improve the performance of the systems. This implies that users are restricted to using specific sign gestures that they might not be comfortable with. In spite of this restriction, the ANFIS performance was still significantly below the obtained performance using polynomial-based recognition.

7. CONCLUSION

In this paper we have successfully applied polynomial classifiers to the problem of Arabic sign language. We have also compared the performance of our system to previously published work using ANFIS-based classification. We have used the same actual data collected from deaf people, and the

Table 4: Comparison between the polynomial-based and ANFIS-based systems.

	ANFIS-based	Polynomial-based	Reduction
Misclassifications using the training data	41	26	36.6%
Misclassifications using the test data	108	46	57.4%



(a) (b) (c)

Figure 8: Depiction of the similarity of the gestures of different alphabets: (a) dal, (b) ba, (c) ra.

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 corresponding percent reduction of misclassified patterns was very significant. Specifically, it was 36% when the systems were evaluated on the training data and 57% when the systems were evaluated on the test data. It should be noted that there is a lot of room for further performance improvement considering different feature sets. Moreover, additional improvements can be obtained by compensating for prior probabilities in the polynomial classifier training considering that the distribution of the training data is not uniform.

REFERENCES

[1] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: a review," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 677–695, 1997.

[2] S. S. Fels and G. E. Hinton, "Glove-talk: a neural network interface between a data-glove and a speech synthesizer," *IEEE Trans. Neural Networks*, vol. 4, no. 1, pp. 2–8, 1993.

[3] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," *IEEE Comput. Graph. Appl.*, vol. 14, no. 1, pp. 30–39, 1994.

[4] D. L. Quam, "Gesture recognition with a dataglove," in *Proc. IEEE National Aerospace and Electronics Conference (NAECON '90)*, vol. 2, pp. 755–760, Dayton, Ohio, USA, May 1990.

[5] J. Eisenstein, S. Ghandeharizadeh, L. Huang, C. Shahabi, G. Shanbhag, and R. Zimmermann, "Analysis of clustering techniques to detect hand signs," in *Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing (ISIMP '01)*, pp. 259–262, Hong Kong, China, May 2001.

[6] M. AL-Rousan and M. Hussain, "Automatic recognition of Arabic sign language finger spelling," *International Journal of Computers and Their Applications*, vol. 8, no. 2, pp. 80–88, 2001, Special Issue on Fuzzy Systems.

[7] J. Davis and M. Shah, "Gesture recognition," Tech. Rep. CS-TR-93-11, Department of Computer Science, University of Central Florida, Orlando, Fla, USA, 1993.

[8] J. Davis and M. Shah, "Visual gesture recognition," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 141, no. 2, pp. 101–106, 1994.

[9] Y. Wu and T. S. Huang, "Vision-based gesture recognition: a review," in *Proc. 3rd International Gesture Workshop (GW '99)*, pp. 103–115, Gif-sur-Yvette, France, March 1999.

[10] C.-S. Lee, Z. Bien, G.-T. Park, W. Jang, J.-S. Kim, and S.-K. Kim, "Real-time recognition system of Korean sign language based on elementary components," in *Proc. 6th IEEE International Conference on Fuzzy Systems (ICFS '97)*, vol. 3, pp. 1463–1468, Barcelona, Spain, July 1997.

[11] T. Starner and A. Pentland, "Visual recognition of American sign language using hidden Markov models," in *Proc. International Workshop on Automatic Face and Gesture Recognition (AFGR '95)*, pp. 189–194, Zurich, Switzerland, June 1995.

[12] T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 12, pp. 1371–1375, 1998.

[13] O. Al-Jarrah and A. Halawani, "Recognition of gestures in Arabic sign language using neuro-fuzzy systems," *Artificial Intelligence*, vol. 133, no. 1-2, pp. 117–138, 2001.

[14] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, 1993.

[15] P. Hong, M. Turk, and T. S. Huang, "Constructing finite state machines for fast gesture recognition," in *Proc. 15th International Conference on Pattern Recognition (ICPR '00)*, vol. 3, pp. 691–694, Barcelona, Spain, September 2000.

[16] W. M. Campbell, K. T. Assaleh, and C. C. Broun, "Speaker recognition with polynomial classifiers," *IEEE Trans. Speech Audio Processing*, vol. 10, no. 4, pp. 205–212, 2002.

[17] K. T. Assaleh and W. M. Campbell, "Speaker identification using a polynomial-based classifier," in *Proc. 5th International Symposium on Signal Processing and Its Applications (ISSPA '99)*, vol. 1, pp. 115–118, Brisbane, Queensland, Australia, August 1999.

[18] W. M. Campbell, K. T. Assaleh, and C. C. Brown, "Low-complexity small-vocabulary speech recognition for portable devices," in *Proc. 5th International Symposium on Signal Processing and Its Applications (ISSPA '99)*, vol. 2, pp. 619–622, Brisbane, Queensland, Australia, August 1999.

[19] M. A. Hussain, "Automatic recognition of sign language gestures," M.S. thesis, Jordan University of Science and Technology, Irbid, Jordan, 1999.

[20] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1997.

[21] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & Fuzzy Systems*, vol. 2, no. 3, pp. 267–278, 1994.

[22] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, NY, USA, 1990.

[23] J. Schurmann, *Pattern Classification*, John Wiley & Sons, New York, NY, USA, 1996.

[24] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins, Baltimore, Md, USA, 1989.

Khaled Assaleh received his Ph.D. degree in electrical engineering from Rutgers, The State University of New Jersey in 1993; the M.S. degree in electronic engineering from Monmouth University, NJ, in 1990; and the B.S. degree in electrical engineering from the University of Jordan, Amman, in 1988. Dr. Assaleh has been an Assistant Professor of electrical engineering at the American University of Sharjah (AUS), UAE, since September 2002. Prior to joining AUS, from March 1997, till July 2002 he was with Conexant Systems, Inc. (formerly Rockwell Semiconductor Systems), Newport Beach, California. Before joining Conexant Systems, Inc., Dr. Assaleh was a Senior Staff Engineer at Motorola, Inc., Phoenix, from November 1994 till March 1997 where he was a member of the Speech and Signal Processing Lab. From October 1993 till November 1994, Dr. Assaleh was a Research Professor at the Center for Computer Aid and Industrial Productivity (CAIP), Rutgers, The State University of New Jersey. Dr. Assaleh holds 10 issued patents and has published over 35 papers in fields related to signal processing and its applications. He is a Senior Member of the IEEE. His research interests include biometrics, speech processing, and biosignal processing.

M. Al-Rousan received his Ph.D. degree from Brigham Young University, USA, in 1996. He is an Associate Professor of Computer Engineering, Jordan University of Science and Technology. Currently, he is on sabbatical leave at the American University of Sharjah, UAE. His search interests include wireless networking, system protocols, intelligent systems, and Internet computing.