Handwritten Arabic Words Recognition using Multi Layer Perceptron and Zernik Moments

I. El-Fegh[1] , Zakaria Suliman Zubi[2], Ali .A.Elrowayati[3] ,Faraj A. El-Mouadib[4]

idrise@ee.edu.ly, zszubi@yahoo.com , arwyate@yahoo.com, elmouadib@yahoo.com

[1] Department of Electrical Engineering, Faculty of Engineering, Al-Fateh University, Tripoli, Libya,

[2]Computer Science Department, Faculty of Science, Al-Tahadi University, Sirt, Libya,

[3] Department of Electrical Engineering, Faculty of Engineering, 7th October University, Misurata, Libya,

[4] Computer Science Department, Faculty of Information Technology, Garyounis University, Benghazi, Libya

Abstract:- *This paper presents the application of Multi Layer Perceptron (MLP) Artificial Neural Network to classification of handwritten Arabic words. Zernik Moments are used as a feature vector for each word. An efficient way to select the most suitable order of Zernik moments is also presented. The MLP is trained in a supervised fashion using the Back Propagation learning algorithm. Having being trained, the MLP is tested on different set of handwritten Arabic words that has never been seen by the MLP. Several experiments are performed to select the best MLP structure. Experimental results have shown that with the presented structure and the order of the Zernik Moments more than 87% of correct recognition was obtained.*

Keywords:- *Zernik Moments, Multi Layer Perceptron, Artifical Neural Networks, Handwritten Arabic Recognition,*

1. Introduction

Automated recognition of text has been an active subject of research since the early days of computers. A 1972 survey cites nearly 130 works on the subject [1].

Despite the age of the subject, it remains one of the most challenging and Exciting areas of research in the are of machine vision. Recently, it has grown into a mature discipline, producing a huge body of research work. Handwriting Recognition is defined as the task of text expressed in graphical format into its symbolic representation. This is useful in digital copies of handwritten text as well as other automated processes such as check handling, and mail sorting.

Arabic language is the language of the holy Quran and is spoken by more than 200 million people, in over 22 different countries. Arabic handwriting recognition is faced by several changelings and can be summarized as follows:

1. Arabic word is cursive.
2. Arabic contains dots and other small marks that can change the meaning of a word, and need to be taken into account by any recognition system.
3. The same letter at the beginning and end of a word can have a completely different form.
4. Some letters have three forms, namely the beginning, the middle and the end.
5. The handwritten words may be of any shapes, sizes, orientations and styles. Since different persons write different ways and even handwriting of a person may be different in different situations

To overcome the above mentioned problems, there has bee a tremendous amount of research in the recognition of Arabic words. This research has taken different avenues including dynamic programming, hidden Markov modeling, neural networks, nearest neighbor classifiers, expert system and combination techniques, more references can be found in [2].

The performance of any recognition system is usually governed by the features used to represent the image. Features of feature vector is mapping of the image into 1-D vector. Feature used must be representative and short. If the dimension of the features vector is short then it will not represent the image properly and if it is long we might run into what is know as the *curse of dimensionality*[3]. Another important aspect of features vector is that it must be invariant to rotation, scaling and shifts[4]

Among the many features used for pattern recognition which are known to be rotation, scale and shift invariant are moments. Moments of an image can be thought of as the decomposition of the image into a series of numbers as invariant descriptors of the image shape that describe the distribution of the image function.

The original work on moment invariants was done by Hu [4]. He derived equations for seven quantities invariant to rotation, translation and scaling known as Hu moments. Hu moments were succesfuly used in the recognition of Latin characters, but concluded more invariants were needed to achieve acceptable performance.

Flusser *et.al* [5], derived a new set of moment invariants that were invariant under linear transformations. Therefore handle word linear, but not non-linear distortion such as rotation.

Zernike [5], introduced a set of complex polynomials that formed a complete orthogonal set over the interior of a unit circle. Teague [6] then introduced the idea of projecting the image function onto orthogonal polynomials. Since Hu moment are not orthogonal, there is redundancy in the information they capture.

Khotanzad and Hong [7], successfully applied the use Zernike Moments as a feature vector to image recognition. They describe the recognition of both Latin characters and lake outlines. This paper also details their method of compensating for translation and scaling of the image as well as rotation.

El-Dabi *et. al* [8], presents a recognition system for typewritten Arabic text. This system uses invariant moments to recognize the Arabic characters in a word. Pixel-wide columns of the image are accumulated until the moments of this image portion are found to be similar to an example character. The image portion is then allocated to that character, and the process starts again on the next image portion. In this way the characters are recognized and segmented simultaneously. Although, this method report good recognition results in typed Arabic text, it will not succeed in the handwritten text since letters can not be separated.

II. Proposed Method

In this work, we propose the use of Zernik moments as a feature vector representing the image of the Arabic word. The Zernik moments vector will constitute the input to a Multi Layer Perceptron (MLP) Artificial Neural Network (ANN). The ANN will be trained using the back propagation training algorithm. The training is performed in a supervised fashion by minimizing the error between the obtained result(obtained by the ANN) and the desired results(result of the known output).

The data set will be split into two sets. Training sets and a test set. The ANN will be trained using the training set while the test set is kept completely hidden. When the ANN is trained, it is then presented with the test set. We used 40 data sets of different size, rotation and shifts as shown in table 1. We use 20 training data sets for training and 20 sets for testing. For each word Zernik moments are extracted and stored in a file. The 20 training and 20 test sets are randomly selected from the total set. The data was collected by asking 40 students to write a set of data each in his own hand writing

Table 1. Sample data set with different Arabic words of different size, rotations and shifts**.**

.

2.1 Zernike Moments

Zernike moments are used in pattern recognition applications as invariant descriptors of the image shape. They have been proven to be superior to

moment functions such as geometric moments in terms of their feature representation capabilities and robustness in the presence of image quantization error and noise [10]. They provide a compact way of describing an object's overall shape using a small set of values.

Zernike moments are defined over a set of complex polynomials which forms a complete orthogonal set over the unit disc $(x^2 + y^2 \le 1)$. First introduced by Zernike (1934) [11], this set of polynomials ZP can be denoted by:

$$ZP = \{V_{nm}(x,y), x^2 + y^2 \le 1\} \qquad (1)$$

The form of the Zernike polynomial basis of order *n* and repetition *m* is [12]:

$$V_{nm}(x,y) = R_{nm}(r) e^{\hat{j} m\theta} \qquad (2)$$

Where:

$$r = \sqrt{x^2 + y^2}, \quad \hat{j} = \sqrt{-1} \quad \text{and} \quad \theta = \arctan\left(\frac{y}{x}\right)$$

$R_{nm}(r)$ is the orthogonal radial polynomial of order n and repetition m and is defined by:

$$R_{nm}(r) = \sum_{s=0}^{\frac{n-m}{2}} (-1)^s \frac{(n-s)!}{s!\left(\frac{n+m}{2}-s\right)!\left(\frac{n-m}{2}-s\right)!} r^{n-2s} \qquad (3)$$

Where *n* is a positive integer, and $m \le n$ and *n-m* is even. The radial polynomial, and hence, Zernike moments are defined in terms of polar coordinates (r,θ) with r ≤ 1, their computation requires a linear transformation of the image coordinates *(i,j), i,j* = 0,1,2,….,N-1 to a suitable domain $(x,y) \in R^2$ inside a unit circle. The most commonly used one is to transform the entire image inside the unit circle. Based on this transformation, the Zernike moment of order n and repetition m can be calculated as follows [13]:

$$Z_{nm} = \lambda(n,N) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R_{nm}(r_{ij}) e^{-\hat{j} m\theta_{ij}} f(i,j) \qquad (4)$$

Where the transformation parameters are given by:

$$r_{ij} = \sqrt{x_i^2 + y_j^2}, \quad \theta_{ij} = \arctan\left(\frac{y_j}{x_i}\right),$$

$$x_i = c_1 i + c_2, \quad y_j = c_1 j + c_2, \qquad (5)$$

$$\lambda(n,N) = \frac{2(n+1)}{\pi(N-1)^2} \qquad (6)$$

$$c_1 = \frac{\sqrt{2}}{N-1} \quad \textit{and} \quad c_2 = -\frac{1}{\sqrt{2}}$$

The geometric moment of order *n+m* of an image is defined by:

$$M_{n,m} = \sum_x \sum_y x^n y^m f(x,y) \qquad (7)$$

Scale invariance is achieving by bringing M $_{0,0}$ (which is equal to the number of foreground pixels) to a If we have all moments of an image f(x,y), up to order n max. Then it is possible to compute a discrete image function f`(x,y), whose Zernike moments are exactly the same as that of $f$(x,y) up to order nmax. As nmax approaches infinity, $f$`(x,y) will become f (x,y)[12]

$$f'(x,y) = \sum_{n=0}^{n_{\max}} \sum_m Z_{nm} V_{nm}(r,\theta) \qquad (8)$$

Equation (8), involves the multiplication of two complex numbers. The modulus of the result is taken to be the reconstructed pixel intensity value.

2.2MLP Classifier Design

MLP consists of one input layer, one or more hidden layers and one output layer. Number of neurons in the input layer is equal to the length of the features vector. Number of neurons in the hidden layers is decided experimentally by starting with small number and increasing the number of neurons in the hidden layer each time while watching the error value. The number of neurons is fixed at the number that gives the lowest square error for a fixed number of training epochs. The number of neurons in the output layer is equal to the number of classes the date is to be classified to. In our case 10 classes are required. The MLP approach helps in distributing the functionality among different interacting components in each layer. Each component is responsible for a small amount of the whole system functionality [14]. The output of the Multilayer Perceptron (MLP) network is:[14]

$$g(x_1,...,x_p) = \sum_{i=1}^{M} \omega_i . g(\sum_{j=1}^{p} w_{ij}\, x_j - \theta_i) \qquad (9)$$

where g(.) is the network output, [x_1, x_2,.. x_p] is the input vector having p inputs, M denotes the Number of hidden neurons, w_{ij} represents the hidden layer connection weights, θ is the threshold value associated with hidden neurons, and ω represents the output layer connection weights which in effect serves as coefficients to the linear output function.

There are two possible representation for the output layer. The first one encodes the output as a vector having a dimension equal to the number of the classes, where all of the outputs are set to zero except for the desired class. In the second encoding scheme, a binary representation for the index of the desired class; this will encode the same number of classes as the previous one but with fewer neurons. In this work, we have selected the second representation

2.3 Back Propagation Training Algorithm

After setting a few parameters, such as the number of samples per class, and the required number of epochs, back propagation training is started; several network topologies were implemented. The following provides a brief summary of the results.

Activation Function: the best results were obtained using the sigmoid activation function. The hyperbolic tangent gave unexpectedly poor performance.

Output Encoding: the best results were obtained using the scheme with a length equal to the number of classes. The binary encoding scheme gave very poor results. This might be explained as follows: In the first scheme, every output neuron can be viewed as a single perceptron dedicated to recognize a single class, while in the second scheme, the overlap between classes makes it harder to classify each example correctly.

Normalized Inputs: normalized inputs resulted in slightly better performance than "raw" inputs.

Cross Validation: cross validation training unexpectedly gave poor results compared to regular back propagation training.

Number of Hidden Layers: from the tests performed a single hidden layer with a sufficient number of neurons performed as well as or better than any number of hidden layers.

Order of Zernik moments: to select the best value for m and n in the Zernik moment, we had to perform a reconstruction test. We have reconstructed the original image from its Zerik moment using inverse Zernik. Results are shown in table 2.

Table (2): Accuracy of reconstructed Arabic word image" ".

Order of moment n	Repetition of order m	Accuracy of reconstructed image (%)
18	4	89.645
18	6	66.321
20	6	79.87
20	3	89.881
21	5	91.8
21	7	88.675
22	9	89.432
22	10	82.345
23	5	90.564
23	7	90.0145
24	7	87.098
24	9	87.009
25	7	95.76
25	9	91.8
25	12	91.43
26	7	89.764
26	9	83.456
27	7	90.0023
27	9	92.543
28	7	92.65
28	9	91.800
Average accuracy		88%

This testing was performed on several word and the most accurate order was obtained with n=25 and m=7. The accuracy is measured by the difference of square error between the reconstructed image and the original image.

III.Experimental Results

The Zernik moments were extracted for every word in the set. The setup of the MLP is fixed number of input and output neurons. The number of hidden layers and the number of neurons in each hidden layer was decided experimentally. Different values of the learning rate were used. If the learning rate is high then the recognition is low and if the learning rate is small then the learning is slow. It was found the most suitable value for learning rate is 0.6.

The training set is presented to the input of the ANN and the output is compared with the expected output and the weights are adjusted according to the back propagation algorithm. Each time the complete set is finished is considered one training epoch or cycle. The training is repeated for 1000 epochs for each set.

After being trained, the ANN is tested using a test data set. The test set was never seen by the ANN during training. The 20 test set were presented to the input of the ANN and the out put is compared with the expected out put. If the obtained output is same as the expected out put , the recognition is considered success other wise is a fail. The number of successful recognition is listed in table 3.

Table 4. summary of the obtained results

Experiments	**1**	**2**	**3**	**4**
Neurons in First hidden Layer	56	112	33	84
Neurons in Second hidden Layer	28	0	33	0
Recognition rate	77%	78%	70.5	87%

IV. Conclusions

The accuracy of reconstructing the image for Arabic word at various order of moments (n) and repetition of order (m) was calculated and found to be maximum at order of moment (n) = 25 with repetition of order (m) = 7. The accuracy of reconstruction was found to be 95.76 %. Therefore, an order of moment (n) = 25 with repetition of order (m) = 7 could be proposed as the best combination for reconstruction of Arabic word. The performance of this method is 87.5 % accuracy on that sets.

The remaining 12% of test examples that were misclassified can be explained by more local variations in the words. Whilst this Zernike moment's method is invariant to global rotation, translation and scaling, it is not invariant to variation in parts of this word. There is also the potential for letter and word skew causing change in the moments. A lot of the variations in handwritten words are of this more local nature, which is hard to deal with when looking at the Arabic word image as a whole.

Reference

[1] L. Harmon. automatic recognition of print and script. Proc. IEE, 60:1165– 1176, 1972.

[2] A.L. Koerich, R. Sabourin and C.Y. Suen (2003), "Large vocabulary off-line handwriting recognition: A survey", Pattern Analysis Applications, vol. 6, 97-121.

[3] M.K. Hu. Visual pattern recognition by moment invariants. IRE Tras. Inform. Theory IT, 8(2):179–187, February 1962.

[4] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. Pattern Recogntion, 26(1):167–174, January 1993.

[5] F. Zernike. Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode (diffraction theory of the cut procedure and its improved form, the phase contrast method). Physics, 1:689–704,1934.

[6] M. Teague. Image analysis via the general theory of moments. Journal of the Optical Society of America, 70(8):920–930, 1980.

[7] A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. IEEE Trans. Pattern Anal. Mach. Intell., 12(5):489–497, 1990.

[8] Sherif Sami El-Dabi, Refat Ramsis, and Aladin Kuwait. Arabic character recognition system: a statistical approach for recognizing cursive typewritten text. Pattern Recognition, 23(5):485–495, 1990.

[9] Peter Burrow. Arabic Handwriting Recognition, Master of Science School of Informatics University of Edinburgh, 2004.

[10] Chee-Way Chong, P. Raveendran, R. Mukundan, A comparative analysis of algorithms for fast computation of Zernike moments. Pattern Recognition Journal volume 36, (2003) 731-742.

[11] Samer M. Abdallah, Object Recognition via Invariance, Phd Thesis, The University of Sydney, 2000.

[12] A study of Zernike moments and their use in Devanagari handwritten character recognition, Proceedings of the International Conference on Cognition and Recognition.

[13] Chee-Way Chong, P. Raveendran, R. Mukundan, Translation invariants of Zernike moments. Pattern Recognition Journal volume 36, (2003) 1765-1773.

[14] Simon Haykin, Neural Networks A Comprehensive Foundation, Second Edition, Prentice Hall Publishing. 1997.

ISSN: 1790-5109 ISBN: 978-960-474-067-3