A Proposed Hybrid Technique for Recognizing Arabic Characters

S.F. Bahgat, S.Ghomiemy
Computer Eng. Dept.
College of Computers and Information Technology
Taif University
Taif, Saudi Arabia

S. Aljahdali, M. Alotaibi
Computer Science Dept.
College of Computers and Information Technology
Taif University
Taif, Saudi Arabia

***Abstract*— Optical character recognition systems improve human-machine interaction and are urgently required for many governmental and commercial departments. A considerable progress in the recognition techniques of Latin and Chinese characters has been achieved. By contrast, Arabic Optical Character Recognition (AOCR) is still lagging although the interest and research in this area is becoming more intensive than before. This is because the Arabic is a cursive language, written from right to left, each character has two to four different forms according to its position in the word, and most characters are associated with complementary parts above, below, or inside the character. The process of Arabic character recognition passes through several stages; the most serious and error-prone of which are segmentation, and feature extraction & classification. This research focuses on the feature extraction and classification stage, being as important as the segmentation stage. Features can be classified into two categories; Local features, which are usually geometric, and Global features, which are either topological or statistical. Four approaches related to the statistical category are to be investigated, namely: Moment Invariants, Gray Level Co-occurrence Matrix, Run Length Matrix, and Statistical Properties of Intensity Histogram. The paper aims at fusing the features of these methods to get the most representative feature vector that maximizes the recognition rate.**

Keywords- Optical Character Recognition; Feature Extraction; Dimensionality Reduction; Principal Component Analysis; Feature Fusion.

I. INTRODUCTION

OCR is the process of converting a raster image representation of a document into a format that a computer can process. Thus, it may involve many sub-disciplines of computer science including image processing, pattern recognition, artificial intelligence, and database systems. Despite intensive investigation, the ultimate goal of developing an optical character recognition (OCR) system with the same reading capabilities as humans still remains unachieved and more so in the case of Arabic language. Most commercially available OCR products are for typed English text because English text characters do not have all the extra complexities associated with Arabic letters.

Arabic is a popular script. It is estimated that there are more than one billion Arabic script users in the world. If OCR systems are available for Arabic characters, they will have a great commercial value. However, due to the cursive nature of Arabic script, the development of Arabic OCR systems involves many technical problems, especially in the segmentation and feature extraction & classification stages. Most characters have dot(s), zigzag(s), madda, etc, associated with the character and this can be above, below, or inside the character. Many characters have a similar shape, the position or number of secondary strokes and dots makes the only difference. Although many researchers are investigating solutions to solve the problems, little progress has been made.

Feature extraction is one of the important basic steps of pattern recognition. Features should contain information required to distinguish between classes, be insensitive to irrelevant variability in the input, and also be limited in number to permit efficient computation of discriminant functions and to limit the amount of training data required. In fact, this step involves measuring those features of the input character that are relevant to classification. After feature extraction, the character is represented by the set of extracted features.

Features can be classified into two categories: Local features, which are usually geometric (e.g. concave/convex parts, number of endpoints, branches, joints, etc), and Global features, which are either topological (connectivity, projection profiles, number of holes, etc) or statistical.

The objective of this paper is to examine the performance of four of these global statistical features; namely: Moments Invariants (MIs), Gray Level Co-occurrence Matrix (GLCM), Run Length Matrix (RLM), and Statistical Properties of Intensity Histogram (SFIH), and to study the effect of fusing two or more of these features on the recognition rate.

The rest of the paper is organized as follows. Section II summarizes the related work. Section III introduces the proposed approach. Results and discussion are presented in Section IV. The paper is terminated by concluding remarks and proposals for future work.

II. RELATED WORK

The features extraction stage, playing the main role in the recognition process, controls the accuracy of recognition by the information passed from this stage to the classifier (recognizer). These information can be structural features such as loops, branch-points, endpoints, and dots; or statistical which includes, but is not limited to, pixel densities, histograms of chain code directions, moments, and Fourier descriptors. Because of the importance of this stage many approaches and techniques have been proposed.

In [1], two methods for script identification based on texture analysis have been implemented: Gabor filters and GLCMs. In tests conducted on exactly the same sets of data, the Gabor filters proved to be far more accurate than the GLCMs, producing results which are over 95% accurate.

[2] presented a new technique for feature extraction based on hybrid spectral-statistical measures (SSMs) of texture. They studied its effectiveness compared with multiple-channel (Gabor) filters and GLCM, which are well-known techniques yielding a high performance in writer identification in Roman handwriting. Texture features were extracted for wide range of frequency and orientation because of the nature of the spread of Arabic handwriting compared with Roman handwriting. The most discriminant features were selected with a model for feature selection using hybrid support vector machine-genetic algorithm techniques. Experiments were performed using Arabic handwriting samples from 20 different people and very promising results of 90.0% correct identification were achieved.

In [3], a novel feature extraction approach of handwritten Arabic letters is proposed. Pre-segmented letters were first partitioned into main body and secondary components. Then moment features were extracted from the whole letter as well as from the main body and the secondary components. Using multi-objective genetic algorithm, efficient feature subsets were selected. Finally, various feature subsets were evaluated according to their classification error using an SVM classifier. The proposed approach improved the classification error in all cases studied. For example, the improvements of 20-feature subsets of normalized central moments and Zernike moments were 15 and 10%, respectively. This approach can be combined with other feature extraction techniques to achieve high recognition accuracy.

In [4], a new set of run-length texture features that significantly improve image classification accuracy over traditional run-length features were extracted. By directly using part or all of the run-length matrix as a feature vector, much of the texture information is preserved. This approach is made possible by the utilization of the multilevel dominant eigenvector estimation method, which reduces the computation complexity of KLT by several orders of magnitude. Combined with the Bhattacharyya measure [5], they form an efficient feature selection algorithm. The advantage of this approach is demonstrated experimentally by the classification of two independent texture data sets. Experimentally, they observed that most texture information is stored in the first few columns of the RLM, especially in the first column. This observation justifies development of a new, fast, parallel RLM computation scheme. Comparisons of this new approach with the co-occurrence and wavelet features demonstrate that the RLMs possess as much discriminatory information as these successful conventional texture features and that a good method of extracting such information is key to the success of the classification.

In [6], Zernike and Legendre Moments for Arabic letter recognition have been investigated. Experiments demonstrated both methods' effectiveness in extracting and preserving Arabic letter characteristics. ZM is used due to its ability to compute the complex orthogonal moments precisely. The system has achieved satisfactory performance when compared with other OCR systems. The translational and scaling invariant, on the other hand, had struggled in LM to detect rotational invariant forms in the experiments. The objective for maximising the correct matching and retrieval from the Arabic database while minimising the false positive rate has been achieved.

[7] explores a design-based method to fuse Gabor filter features and co-occurrence probability features for improved texture recognition. The fused feature set utilizes both the Gabor filter's capability of accurately capturing lower frequency texture information and the co-occurrence probability's capability in texture information relevant to higher frequency components. Fisher linear discriminant analysis indicates that the fused features have much higher feature space separation than the pure features. Overall, the fused features are a definite improvement over non-fused features and are advocated in texture analysis applications.

III. Proposed Approach

Substantial research efforts have been devoted during last years to AOCR and many approaches have been developed (structural, geometric, statistics, stochastic…). However, certain problems remain open and deserve more attention in order to achieve results equivalent to those obtained for other scripts such as Latin. Besides, other methods must be explored and various sources of information have also to be used [8].

The process of isolated Arabic optical character recognition comprises three main stages: Preprocessing, Feature extraction, and Classification. The structure of the proposed approach is shown in Figure 1.

The training dataset includes the 28 (100 x 100) *jpg* images of the isolated Arabic characters shown below:

خ	ح	ج	ث	ت	ب	أ
ق	ف	غ	ع	ظ	ط	ض
ص	ش	س	ز	ر	ذ	د
ي	و	هـ	ن	م	ل	ك

The test datasets include:

1) 3 datasets composed of the clean set corrupted by salt and pepper noise of intensity 1 %, 3 %, and 5 % respectively.

2) 3 datasets composed of the clean set corrupted by impulse noise of intensity 1 %, 3 %, and 5 % respectively.

3) 3 datasets composed of the clean set corrupted by Gaussian noise of intensity 1 %, 3%, and 5% respectively.

Figure 2 displays the letter “ ش “ as an example of the 10 datasets.

The procedure proceeds as follows:

1) In the preprocessing phase, the noise removal is carried out using median filter, and the binarization is done with histogram thresholding.

2) In the feature extraction phase, four feature sets are calculated using MIs, GLCM, RLM, and SFIH, respectively. These initial feature vectors are used for evaluating the maximum possible recognition rate for the corrupted datasets, using each set of features. The relations used for calculating the features of the different techniques are discussed below.

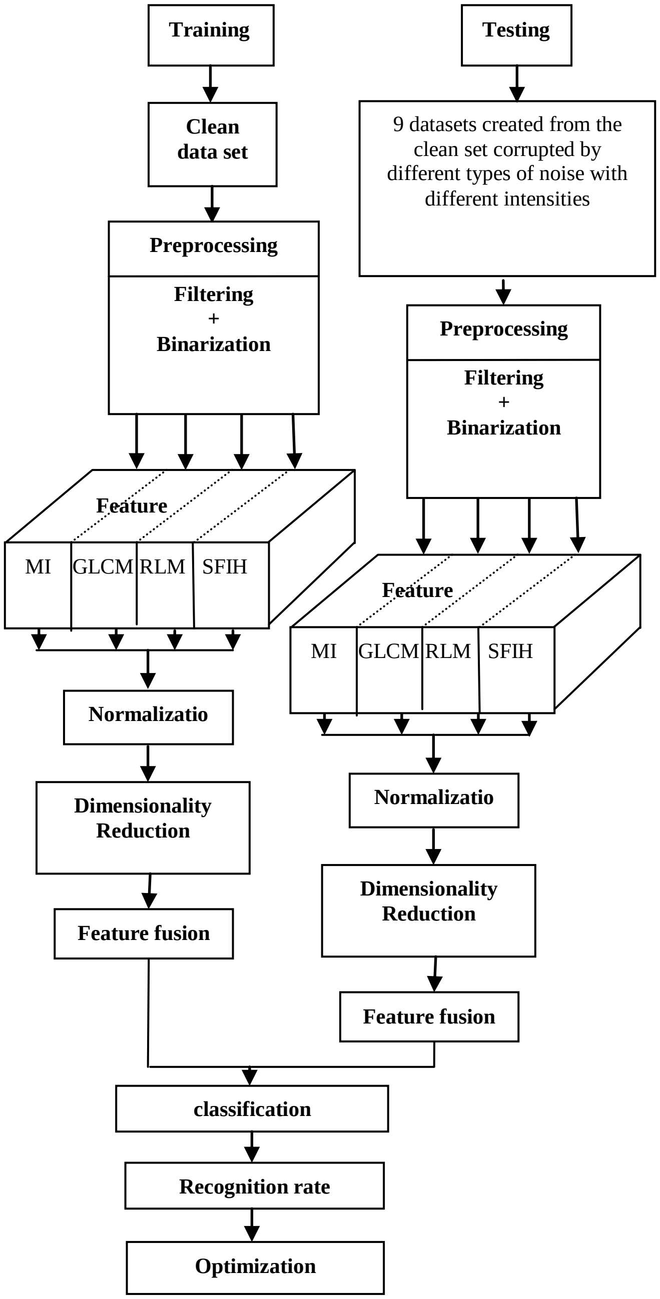


Figure 1. Proposed approach flow diagram

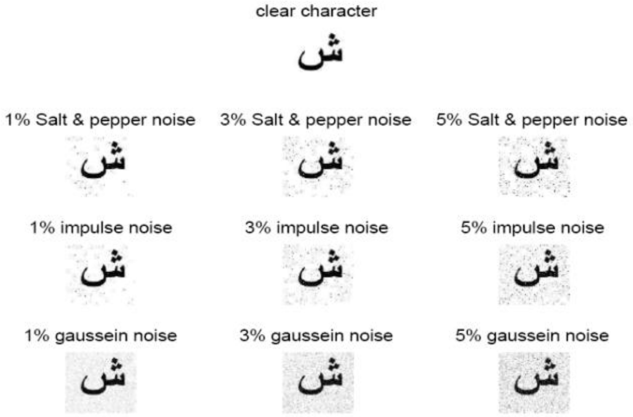


Figure 2. Sample Images of different datasets

A. MIs Features:

The regular moment of a shape in an M by N binary image is defined as:

$$u_{pq} = \sum_{j=0}^{N-1}\sum_{i=0}^{M-1} i^p j^q f(i,j) \qquad (1)$$

where *f(i,j)* is the intensity o

f the pixel (either 0 or 1) at the coordinate (i,*j*) and *p+q* is said to be the order of the moment. The coordinates of the centroid are determined using the relations:

$$i^{'} = \frac{u_{10}}{u_{00}} \quad and \quad j^{'} = \frac{u_{01}}{u_{00}} \qquad (2)$$

Relative moments are then calculated using the equation for central moments defined as:

$$u_{pq} = \sum_{j=0}^{N-1}\sum_{i=0}^{M-1} (i-i^{'})^p (j-j^{'})^q f(i,j) \qquad (3)$$

A set of seven rotational invariant moment functions which form a suitable shape representation were derived by Hu [9, 10, 11]. These equations, used throughout this work are shown in Appendix A_i .

B. GLCM Features

The GLCM is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image [13]. The GLCM is used for a series of "second order" texture calculations. GLCM texture considers the relationship between *groups of two (usually neighboring) pixels* in the original image. at a time, called the reference and the neighbour pixel. The neighbour pixel is chosen to be the one to the east (right) of each reference pixel. This can also be expressed as a (1,0) relation: 1 pixel in the x direction, 0 pixels in the y direction. Each pixel within the window becomes the reference pixel in turn, starting in the upper left corner and proceeding to the lower right. Pixels along the right edge have no right hand neighbour, so they are not used for this count.

To create a GLCM, use the graycomatrix function. The graycomatrix function creates a gray-level co-occurrence matrix (GLCM) by calculating how often a pixel with the intensity (gray-level) value *i* occurs in a specific spatial relationship to a pixel with the value *j*. By default, the spatial relationship is defined as the pixel of interest and the pixel to its immediate right (horizontally adjacent), but you can specify other spatial relationships between the two pixels. Each element (*i*,*j*) in the resultant GLCM is simply the sum of the number of times that the pixel with value *i* occurred in the specified spatial relationship to a pixel with value *j* in the input image.

Because the processing required to calculate a GLCM for the full dynamic range of an image is prohibitive, graycomatrix scales the input image. By default, graycomatrix uses scaling to reduce the number of intensity values in grayscale image from 256 to eight. The number of gray levels determines the size of the GLCM. To control the number of gray levels in the GLCM and the scaling of intensity values, using the NumLevels and the Gray Limits parameters of the graycomatrix function. The GLCM can reveal certain properties about the spatial distribution of the gray levels in the texture image. For example, if most of the entries in the GLCM are concentrated along the diagonal, the texture is coarse with respect to the specified offset. You can also derive several statistical measures from the GLCM. The set of features extracted from the GLCM matrix [14] is shown in Appendix A_{ii}.

C. RLM Features

Run-length statistics capture the coarseness of a texture in specified directions. A run is defined as a string of consecutive pixels which have the same gray level intensity along a specific linear orientation. Fine textures tend to contain more short runs with similar gray level intensities, while coarse textures have more long runs with significantly different gray level intensities [15].

A run-length matrix *P* is defined as follows: each element *P(i, j)* represents the number of runs with pixels of gray level intensity equal to *i* and length of run equal to *j* along a specific orientation. The size of the matrix *P* is *n by k*, where *n* is the maximum gray level in the image and *k* is equal to the possible maximum run length in the corresponding image. An orientation is defined using a displacement vector *d(x, y)*, where *x* and *y* are the displacements for the *x-axis* and *y-axis*, respectively. The typical orientations are 0°, *45*°, *90*°, and *135*°, and calculating the run-length encoding for each direction will produce four run-length matrices.

Once the run-length matrices are calculated along each direction, several texture descriptors are calculated to capture the texture properties and differentiate among different textures [15]. The set of RLM features is shown in Appendix A_{iii}.

D. SFIH Features

A frequently used approach for texture analysis is based on statistical properties of intensity histogram. One such measures is based on statistical moments. The expression for the n^{th} order moments about the mean is given by:

$$\mu_n = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \qquad (4)$$

Where z_i is a random variable indicating intensity, $p(z_i)$ is the histogram of the intensity levels in the image, L is the number of possible intensity levels and

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \qquad (5)$$

is the mean (average) intensity. The set of features following this approach is shown in Appendix A_{iv}.

Feature selection helps to reduce the feature space which improves the prediction accuracy and minimizes the computation time. This is achieved by removing irrelevant, redundant and noisy features, i.e., it selects the subset of features that can achieve the best performance in terms of accuracy and computation time. It performs the Dimensionality reduction. Principal Components Analysis (PCA) is a very popular technique for dimensionality reduction. Given a set of data on *n* dimensions, PCA aims to find a linear subspace of dimension *d* lower than *n* such that the data points lie mainly on this linear subspace. Such a reduced subspace attempts to maintain most of the variability of the data. Applying PCA for dimensionality reduction, we get the minimum number of features giving the maximum possible recognition rate obtained earlier using the full feature vector), for each procedure. Analyzing the effect of feature fusion by fusing the features of each two of the four procedures, and evaluating the resultant recognition rate.

Classification is the main decision stage of the OCR system in general. In this stage the features extracted from the primitive is compared to those of the model set. As the classification is generally implemented according to the criterion of minimizing the Euclidian distance between feature vectors, it is necessary to normalize the fused features. The normalization should comply with a rule that each feature component should be treated equally for its contribution to the distance. The rationale usually given for this rule is that it prevents certain features from dominating distance calculations merely because they have large numerical values. A linear stretch method can be used to normalize each feature component over the entire data set to be between zero and one. A feature selection procedure can be used after the feature vectors are fused. A weighting method called feature contrast, is employed to perform an unsupervised feature selection.

Denote the i^{th} n-D fused feature vector as $F_i = \{f_{i,1}, f_{i,2}, \cdots, f_{i,n}\}$. The feature contrast of the j^{th} component of the feature vector is defined as:

$$\xi_j = \frac{\max_i(f_{i,j}) - \text{mean}_i(f_{i,j})}{\max_i(f_{i,j}) + \text{mean}_i(f_{i,j})} \qquad (6)$$

Then each feature component is weighted by its feature contrast divided by the maximum feature contrast of all feature components, that is,

$$F_i^* = \frac{1}{\max_j(\xi_j)} \{\xi_1 f_{i,1}, \{\xi_2 f_{i,2}, \ldots., \{\xi_n f_{i,n}\} \qquad (7)$$

A common strategy of feature fusion is first to combine various features and then perform feature selection to choose an optimal feature subset according to the feature data set itself, such as by principal component analysis (PCA).

As we are interested mainly in feature extraction, no great emphasis is paid for the classier. We will implement only the basic classifier; namely: Nearest-Neighbor Classifier, based on the Euclidean distance between a test sample and the specified training samples. Let xi be an input sample with p features (xi1,xi2,…,xip) , n be the total number of input samples (i=1,2,…,n) and p the total number of features (j= 1,2,…,*p*) . The Euclidean distance between sample $\mathbf{x}_i$ and $\mathbf{x}_l$ (l =1,2,…,*n*) is defined as:

$$d(x_i,x_l) = \sqrt{((x_{i1}-x_{l1})^2 + (x_{i2}-x_{l2})^2 + \cdots + (x_{ip}-x_{lp})^2)} \qquad (8)$$

IV. Results & Discussion

In the training phase, four sets of features are calculated for the clean dataset using the four methods under consideration (MIs, GLCM, RLM, and SFIH). The PCA algorithm is also applied in each case, and the corresponding feature vectors are stored for further processing.

In the testing phase, the same approach is followed for the data in the nine corrupted datasets. Using the full feature vectors, the recognition rate is determined for each method, and is labeled as the maximum possible recognition rate that can be achieved in this situation. As the feature vectors are sorted in a descending order as a result of applying PCA, we searched for the minimum number of features giving the maximum possible recognition rate for each method. According to Table 1, the maximum recognition rate was achieved using MIs, followed by GLCM, and RLM. The SFIH gave the least recognition rate. Figure 3, clarifies these results. The minimum number of features satisfying maximum recognition rate was found to be 2, 3, 4, and 1 for IMs, GLCM, RLM, and SFIH, respectively.

TABLE 1. Maximum possible recognition rate for the corrupted datasets,

Noise Type	MIs	GLCM	RLM	SFIH
Salt & Pepper	99.107	96.429	95.536	87.5
Gaussian Noise	98.214	95.536	91.071	80.357
Impulse Noise	99.107	96.429	91.071	87.5
Average	98.813	96.13	92.559	85.119

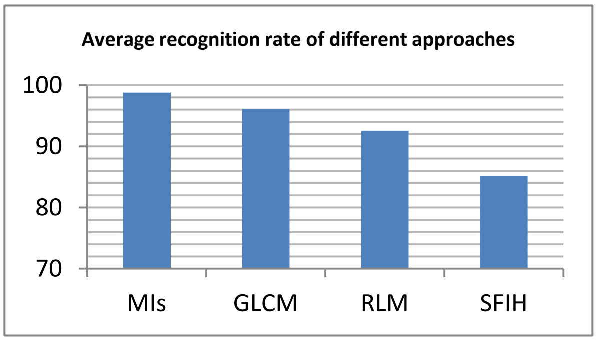


Figure 3. Average recognition rate of different approaches

The effect of the number of features of MIs on the recognition rate is shown in Table 2 for the different types of noise. On the average, the effect of the number of features of MIs on the recognition rate is shown in Figure 4.

TABLE 2. The relation between the number of features and the obtained recognition rate for MIs

Noise Type	2 Features	1 Feature
Salt & Pepper	99.107	73.214
Gaussian Noise	98.214	67.857
Impulse Noise	99.107	75.893
Average	98.8093333	72.321333

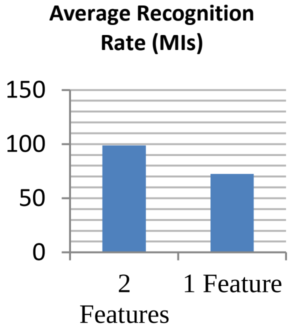


Figure 4. Average recognition rates of MIs as a function of the number of features

The effect of the number of features of GLCM on the recognition rate is shown in Table 3 for the different types of noise. On the average, the effect of the number of features of MIs on the recognition rate is shown in Figure 5.

TABLE 3. The relation between the number of features and the obtained recognition rate for GLCM

Noise Type	3 Features	2 Features	1 Feature
Salt & Pepper	96.429	96.429	91.071
Gaussian Noise	95.538	94.643	85.714
Impulse Noise	96.429	96.429	85.714
Average	96.132	95.833667	87.499667

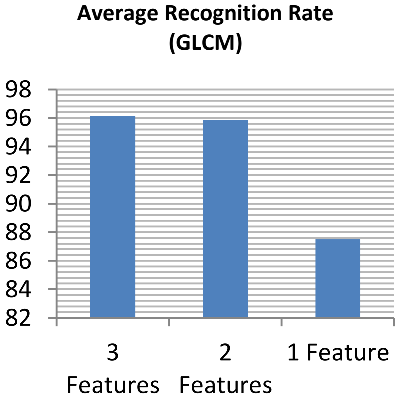


Figure 5. Average recognition rate of GLCM as a function of the number of features

The effect of the number of features of RLM on the recognition rate is shown in Table 4 for the different types of noise. On the average, the effect of the number of features of MIs on the recognition rate is shown in Figure 6.

TABLE 4. The relation between the number of features and the obtained recognition rate for RLM

Noise Type	4 Features	3 Features	2 Features	1 Feature
Salt & Pepper	95.536	91.071	75	75
Gaussian Noise	91.071	85.712	69.643	65.179
Impulse Noise	91.071	88.393	77.679	73.214
Average	92.55933	88.392	74.107333	71.131

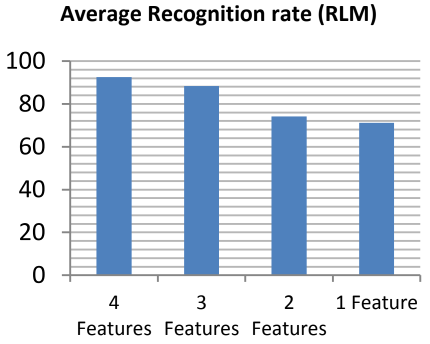


Figure 6. Average recognition rate of RLM as a function of the number of features

The effect of the number of features of SFIH on the recognition rate is shown in Table 5 for the different types of noise. On the average, the effect of the number of features of MIs on the recognition rate is shown in Figure 7.

TABLE 5. The relation between the number of features and the obtained recognition rate for SFIH

Noise Type	2 Features	1 Feature
Salt & Pepper Noise	87.5	87.5
Gaussian Noise	80.357	80.357
Impulse Noise	88.393	87.5
Average	85.4166667	85.119

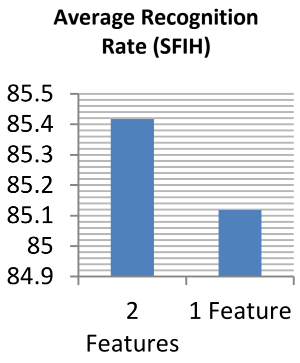


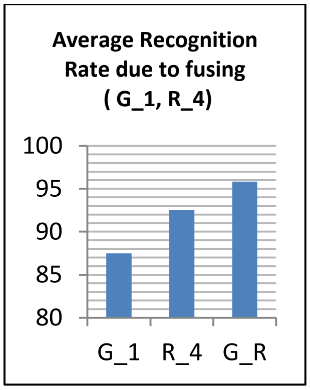
Figure 7. Average recognition rate of SFIH as a function of the number of features

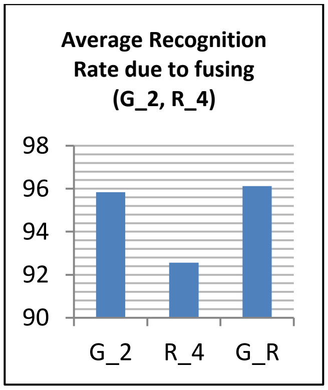
As the main objective is to emphasize the effect of hybridization (feature fusion) on the enhancement of recognition rate, Tables 6, 7, and 8 illustrate the resultant recognition rate due to fusing features GLCM with RLM , MIs with GLCM, and GLCM with SFIH.

TABLE 6. The GLCM features with the RLM features

Recognition rate	Gaussian	impulse noise	salt & pepper	Average
GLCM with one feature (G_1)	85.714	85.714	91.071	87.5
RLM with 4 features (R_4)	91.0714	91.0714	95.535	92.559
GLCM and RLM (G_R)	95.5357	94.6428	97.321	95.833
GLCM with 2 features (G_2)	94.642	96.428	96.428	95.833
RLM with 4 features (R_4)	91.071	91.071	95.535	92.559
GLCM and RLM (G_R)	95.535	95.535	97.321	96.130
GLCM with one feature (G_1)	85.714	85.714	91.071	87.5
RLM with 3 features (R_3)	85.714	88.392	91.071	88.392
GLCM and RLM (G_R)	95.535	94.642	97.321	95.833

According to Figure 8, fusing 2 features of GLCM with 4 features of RLM, gives the best recognition rate (96.13 %).





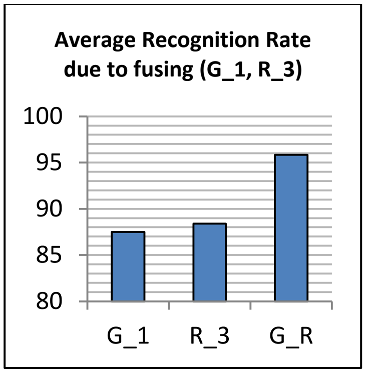


Figure 8. Average recognition rate of Fusing GLCM features and RLM features

On the other hand, fusing two MIs features with two GLCM features leads to the best recognition rate (99.4%), as shown in Figure 9.

TABLE 7. The moment features with the GLCM features

Recognition rate	Gaussian	impulse noise	salt & pepper	Average
moments with 2 features (M_2)	98.214	99.107	99.107	98.809
GLCM with 2 features (G_2)	94.642	96.428	96.428	95.833
moments and GLCM (M_G)	98.214	100	100	99.404
moments with one feature (M_1)	67.857	75.892	73.214	72.321
GLCM with one feature (G_1)	85.714	85.714	91.071	87.5
moments and GLCM (M_G)	88.392	89.285	94.642	90.773
moments with 2 features (M_2)	98.214	99.107	99.107	98.809
GLCM with 1 feature (G_1)	85.714	85.714	91.071	87.5
moments and GLCM (M_G)	88.392	89.285	94.642	90.773
moments with 1 feature (M_1)	67.857	75.892	73.214	72.321
GLCM with 2 features (G_2)	94.642	96.428	96.428	95.833
moments and GLCM (M_G)	98.214	100	100	99.404

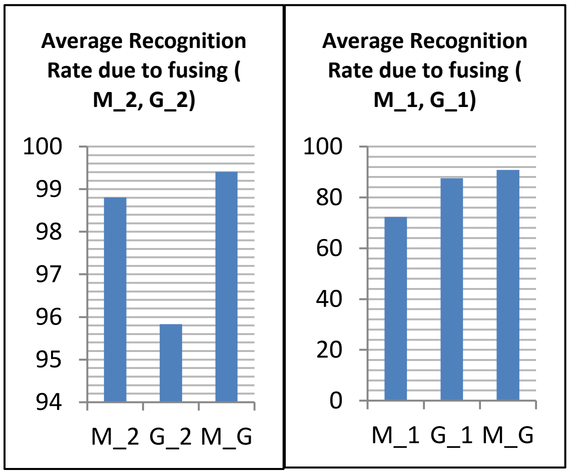


Figure 9. Average recognition rate of Fusing IMs and GLCM features

TABLE 8. GLCM features with the Statistical features

Recognition rate	Gaussian	impulse noise	salt & pepper	Average
statistical with 2 features (S_2)	80.357	88.392	87.5	85.416
GLCM with 2 features (G_2)	94.642	96.42	9	95.833
statistical and GLCM (S_G)	95.535	97.321	97.321	96.726
statistical with one feature (S_1)	80.357	87.5	87.5	85.119
GLCM with one feature (G_1)	85.714	85.714	91.071	89.285
statistical and GLCM (S_G)	85.7142	86.607	91.964	89.880

However, fusing features of GLCM with features of SFIH, gives very small enhancement in the recognition rate as shown in Figure 10.

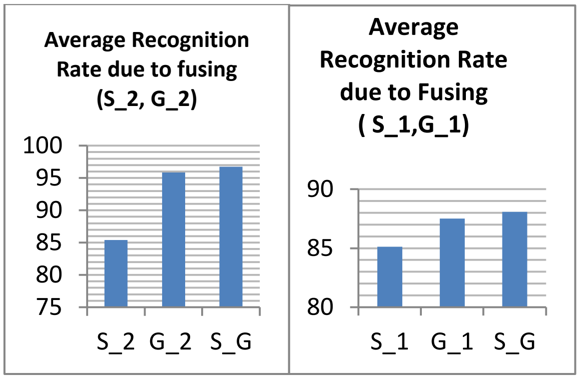


Figure 10. Average recognition rate of Fusing GLCM features and SFIH features

V. Conclusions & Future Work

This paper investigates the performance of approaches for the statistical feature extraction techniques, namely; Moment Invariants, Gray Level Co-occurrence Matrix, Run Length Matrix, and Statistical Features of Intensity Histogram, and proposes a hybrid technique fusing features from the four methods for enhancing the Arabic characters recognition rate. Three types of noise, with different intensity levels were used for estimating the gained enhancement, namely; salt & pepper, impulse, and Gaussian noise, with intensity levels of 1%, 3%, and 5% for each type of noise. It was found that the fusion of the moment features with those of GLCM leads to about 100% recognition rate for all noise intensity levels used. Further investigation is needed for fusing more than two types of features and using higher intensity levels to generalize the obtained results.

References

[1] G.S.Peake and T.N.Tan, “Script and Language Identification from Document Images”, Proceeding ACCV '98 Proceedings of the Third Asian Conference on Computer Vision-Volume II, Springer-Verlag London, UK ©1997, ISBN:3-540-63931-4.

[2] Al-Dmour, Ayman; Zitar, Raed Abu, “Arabic writer identification based on hybrid spectral-statistical measures”, http://en.zl50.com/1201106198616921.html, Volume 19, Number 4, December 2007, pp. 307.

[3] Gheith Abandah and Nasser Anssari, “Novel Moment Features Extraction for Recognizing Handwritten Arabic Letters”, Journal of Computer Science 5 (3): 226-232, 2009, ISSN 1549-3636.

[4] Xiaoou Tang, “Texture information in run-length matrices”, IEEE Transactions on Image Processing, Vol.7, Issue 11, pp 1602-1609, 1998.

[5] Frank J. Aherne; Neil A. Thacker; Peter I Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data", Kybernetika, Vol. 34 (1998), No. 4, [363]—368.

[6] H. Aboaisha1, Zhijie Xu1, I. El-Feghi, "An Investigation On Efficient Feature Extraction Approaches For Arabic Letter Recognition", Proceedings of The Queen's Diamond Jubilee Computing and Engineering Annual Researchers' Conference 2012: CEARC'12. University of Huddersfield, pp. 80-85. ISBN 978-1-86218-106-9.

[7] David A. Clausi, Huawu Deng, "Design-based texture feature fusion using Gabor filters and co-occurrence probabilities", IEEE Transactions on Image Processing, 2005,Vol. 14, issue 7.pages 925—936.

[8] Øivind Due Trier, Anil K. Jain, Torfinn Taxt, "Feature extraction methods for character recognition-A survey" ELSEVIER, Pattern Recognition, Volume 29, Issue 4, April 1996, Pages 641–662.

[9] Qing Chen, "Evaluation of OCR Algorithms for Images with Different Spatial Resolutions and Noises", A Master of Applied Science thesis submitted to the School of Graduate Studies and Research, Ottawa-Carleton Institute for Electrical Engineering, School of Information Technology and Engineering, Faculty of Engineering, University of Ottawa, Canada, 2003.

[10] R.Muralidharan1, C.Chandrasekar, "Object Recognition using SVM-KNN based on Geometric Moment Invariant", International Journal of Computer Trends and Technology- July to Aug Issue 2011, ISSN: 2231-2803, Page 215-220.

[11] R. J. Ramteke, "Invariant Moments Based Feature Extraction for Handwritten Devanagari Vowels Recognition", International Journal of Computer Applications (0975 - 8887), Volume 1 – No. 18, 2010.

[12] Fritz Albregtsen, "Statistical Texture Measures Computed from Gray Level Co-occurrence Matrices, Image Processing Laboratory, Department of Informatics, University of Oslo, November 5, 2008.

[13] Zidouri, A., "PCA-Based Arabic Character Feature Extraction" IEEE, 9th International Symposium on Signal Processing and its Applications, Vol S1-3; pp: 652-655;King Fahd University of Petroleum & Minerals, (2007) .http://www.kfupm.edu.sa.

[14] http://www.fp.ucalgary.ca/mhallbey/GLCM_as_probability.htm

[15] Dong-Hui Xu, Arati S. Kurani, Jacob D. Furst, Daniela S. Raicu, "Run-Length Encoding For Volumetric Texture",

[16] X. Tang, Texture information in run-length matrices, IEEE Transactions on Image Processing, 7(11), 1998, 1602-1609.

APPENDICES

APPENDIX A$_{I}$ (MOMENT FEATURES)

$$M_1 = (u_{20} + u_{02})$$

$$M_2 = (u_{20} - u_{02})^2 + 4u_{11}^2$$

$$M_3 = (u_{30} - 3u_{21})^2 + (3u_{21} - u_{30})^2$$

$$M_4 = (u_{30} + u_{12})^2 + (u_{21} + u_{03})^2$$

$$M_5 = (u_{30} - 3u_{12})(u_{30} + u_{12})((u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2) + (3u_{12} - u_{30})(u_{21} + u_{03})(3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2)$$

$$M_6 = (u_{20} - u_{02})((u_{30} + u_{12})^2 - (u_{21} + u_{03})^2) + 4u_{11}(u_{30} + 3u_{12})(u_{21} + u_{03})$$

$$M_7 = (3u_{12} - u_{30})(u_{30} + u_{12})((u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2) - (u_{30} - 3u_{12})(u_{21} + u_{03})(3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2)$$

APPENDIX A$_{II}$ (GLCM FEATURES)

$$Contrast = \sum_{i,j=0}^{N-1} P_{i,j}\,(i-j)^2 \quad ,$$

$$Homogenity = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$$

$$Dissimilarity = \sum_{i,j=0}^{N-1} P_{i,j}\,|i-j|,$$

$$Similarity = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+|i-j|}$$

$$Homogenity = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$$

$$Angular\ Second\ Moment(ASM) = \sum_{i,j=0}^{N-1} {P_{i,j}}^2$$

$$GLCM\ Mean: \mu_i = \sum_{i,j=0}^{N-1} i(P_{i,j}), \qquad \mu_j = \sum_{i,j=0}^{N-1} j(P_{i,j})$$

$$GLCM\ Variance: \ {\sigma_i}^2 = \sum_{i,j=0}^{N-1} P_{i,j}\,(i-\mu_i)^2,$$

$${\sigma_j}^2 = \sum_{i,j=0}^{N-1} P_{i,j}\,(j-\mu_j)^2$$

$$Standard\ Deviation: \ \sigma_i = \sqrt{{\sigma_i}^2}, \ \ \sigma_j = \sqrt{{\sigma_j}^2}$$

$$GLCM\ Correlation: \ \sum_{i,j=0}^{N-1} P_{i,j}[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{{\sigma_i}^2}\sqrt{{\sigma_i}^2}}]$$

APPENDIX$_{III}$ (RUN LENGTH FEATURES)

$$SRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)}{j^2}$$

$$LRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i,j) * j^2$$

$$HGRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i,j) * i^2$$

$$LGRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)}{i^2}$$

$$SRLGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)}{i^2 * j^2}$$

$$SRHGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j) * i^2}{j^2}$$

$$LRLGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j) * j^2}{i^2}$$

$$LRHGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N} p(i,j) * i^2 * j^2$$

APPENDIX A$_{IV}$ (STATISTICAL FEATURES)

$$\mu = \bar{X} = \frac{1}{N}\sum_{i} X_i$$

$$\sigma^2 = \frac{1}{N-1}\left(\sum(X_i - \bar{X})\right)^2)$$

$$\sigma = \sqrt{\tfrac{1}{N-1}(\sum(X_i - \bar{X})^2)}$$

$$R = 1 - \frac{1}{(1+\sigma^2)}$$

$$\mu_3 = \frac{\sum(X-\mu)^3}{N\sigma^4}$$

$$\mu_4 = \frac{\sum(X-\mu)^4}{N\sigma^4}\text{-}3$$

$$U = \sum_{i=0}^{L-1} p^2(z_i)$$

$$e = -\sum_{i=0}^{L-1} p(z_i)\,log_2\,p(z_i)$$

$$RLNU = \frac{1}{n_r}\sum_{i=1}^{M}(\sum_{j=1}^{N} p(i,j))^2$$

$$RPC = \frac{n_r}{P(i,j) * j}$$