British Journal of Mathematics & Computer Science
4(17): 2521-2535, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

Automatic Check Digits Recognition for Arabic Using Multi-Scale Features, HMM and SVM Classifiers

Sameh M. Awaida[1*] and Sabri A. Mahmoud[2]

[1]Computer Engineering Department, Qassim University, Qassim 51452, Kingdom of Saudi Arabia.
[2]King Fahd University of Petroleum and Engineering, Dhahran 31261, Kingdom of Saudi Arabia.

Original Research Article

Received: 24 May 2014
Accepted: 18 June 2014
Published: 30 June 2014

Abstract

We propose in this work two Automatic Arabic (Indian) digits recognition systems using a real-life dataset of 3000 bank checks. The systems extracts features from training-set images of 7390 isolated digits (0-9). These features are multi-scale in which they capture narrow, intermediate, and large-scale qualities of the image. The gradient features correspond to the narrow scale, the structural features correspond to the intermediate scale, and the concavity features correspond to the large-scale. These features are employed by two different statistical classifiers; Hidden Markov Models (HMM) and Support Vector Machines (SVM). The two independent recognition systems utilize the proficient CENPARMI Arabic bank check database for training and testing. In order to select the optimal parameters for feature extraction and for the HMM classifier, the CENPARMI training dataset is divided into training and verification subsets. After adapting the two systems' parameters, they are tested on unobserved 3035 digit images. The average recognition rates for the HMM and SVM systems are 97.86% and 99.04%, respectively. The presented systems provides state-of-the-art recognition results on the CENPARMI database, as they reported a higher recognition rates when compared to twelve previously published systems, especially for the SVM system. After analyzing the classification errors, the authors conclude that some of these errors are inevitable as they are most probably attributed to errors in labeling the original database, distinct writing styles of certain digits, and genuine faults.

Keywords: Classifier design and evaluation, handwriting analysis, hidden Markov models, independent writer digit recognition, Arabic (Eastern Arabic) digits, support vector machines.

1 Introduction

Handwritten digits constitute an important part in handwritten documents. Recognition of these digits has many potential applications in today's world, e.g. courtesy amounts in bank checks,

**Corresponding author: s.awaida@qu.edu.sa;*

postal codes in mail letters, data entry applications, automatic exams correction, and other useful applications.

Although the Arabic language is written from right to left, its digits are written for left to right, similar to English, where the right-most digit is the least significant one and the left-most digit is the most significant one. For historical reasons, the set of Arabic digits are sometimes referred to as Indian digits. In this paper, we will refer to digits written in the Arabic language as 'Arabic digits'. Fig. 1 illustrates samples of handwritten Arabic and Latin digits '٠' (0) to '٩' (9) (from right to left). Digit '١' (1) is similar in Arabic and Latin. Arabic digit '٥' (5) is similar to Latin digit '0'. Digit '٩' (9) of Arabic and Latin are similar with lower stroke projecting to lower-right in Arabic and lower-left in Latin. There exist two styles of writing digit '4' in Latin and two styles in writing digit '٣' (3) in Arabic [1].

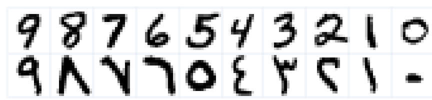

Fig. 1. Latin (Top) and Arabic (Bottom) Handwritten Digits ٠ (0) To ٩ (9)

For more than 50 years, the topic of automatic recognition of English handwritten has seen several proposed methods with high recognition rates [2-6]. Recently, researchers started extensively addressing the topic of Arabic text automatic recognition including Arabic digits [7-14]. However, researchers have rarely agreed on a common database to develop their recognition systems on. This is due to the lack of publicly available and acknowledged Arabic handwritten databases. The two most common databases in this area are the Institute of Communications Technology/Ecole Nationale d'Ingénieurs de Tunis (IFN/ENIT) database of handwritten Arabic words [15], and the CENPARMI Arabic check database developed by the Center for Pattern Recognition and Machine Intelligence [16]. However, the IFN/ENIT database only contains Latin digits and has no Arabic digits.

Sun et al. [17] used partial least squares (PLS) regression and feature fusion on the CENPARMI Arabic handwritten digits database [16]. They chose four types of features; Gabor transformation features, Legendre moment features, Pseudo-Zernike moment feature, and Zernike moment features. They applied their proposed non-iterative PLS algorithm and feature fusion method for choosing the best combination for optimal recognition results. Hu et al. [18] used multi-classifier combination on the same database. Their system used nine classifiers with different features and combined them to recognize the digits.

Mahmoud and Al-Khatib [19] used log-Gabor filter for feature extraction withfour different classifiers; i.e. K-Nearest Neighbor (KNN), Nearest Mean (NM), HMM, and SVM. They reported their results on the CENPARMI database.

Juan et al. [20] used multivariate Bernoulli mixture classifiers for the recognition of Arabic digits in the CENPARMI database. In [21], they also tried six different EM initialization techniques for their Bernoulli mixture classifier to improve their recognizer results. Gimenez et al. [22] used a similar approach by proposing a mixture of multi-class logistic regression model, inspired by Bernoulli mixture models. Finally, Sadri et al. [23] used Support Vector Machines (SVMs) and compared their results with a Multi Layer Perceptron (MLP) neural network classifier.

We present in this paper two successful recognition systems for offline handwritten Arabic isolated digits '٠' (0) to '٩' (9). These systems employ the Gradient, Structural, and Concavity (GSC) features [1]. The GSC features are multi-scale as they capture the narrow, intermediate, and large-scale qualities of the image. The gradient features detect the low-level gradient direction frequency and correspond to the narrow scale. The structural features compute several geometric characteristics such as the count of lines and corners at various directions and correspond to the intermediate scale. The concavity features correspond to the large-scale as they compute the count of large vertical and horizontal strokes, presence of holes, and direction of bays.

Both Hidden Markov Models (HMM) and SVM classifiers are implemented for the recognition task. The values of the HMM optimal parameters are estimated by dividing the CENPARMI training set into training and verification subsets, where HMM is trained with the training subset and tested with the verification subset. Once the optimal parameters are selected, HMM is trained using the CENPARMI training set. SVM parameters are fine-tuned using a 10% V-fold from the original training data to optimize its performance and accuracy. Then HMM and SVM is tested using the CENPARMI test dataset. The results of HMM and SVM recognition rates are compared to previously published work. The recognition rates of HMM and SVM proved to be superior to other published work as detailed in Section 5.

This paper is organized as follows. The database is described in Section 2; feature extraction is addressed in Section 3, where three types of features are used. Hidden Markov Models are addressed in Section 4. Support Vector Machines are summarized in Section 5. Training, recognition, and experimental results are addressed in Section 6. Finally, the conclusions are presented in Section 7.

2 Database

The database was developed by researchers from CENPARMI [16]. By scanning 7000 real world grey-level bank check images, they were able to produce a number of databases that can be used to advance research efforts in Arabic Intelligent Character Recognition (ICR) systems. 3000 of the scanned 7000 checks were used in building the databases. These databases include Arabic legal-amounts database, courtesy amounts database, Arabic sub-words database, and Arabic digits database.

Fig. **2** show a sample Arabic check from the CENPARMI database.

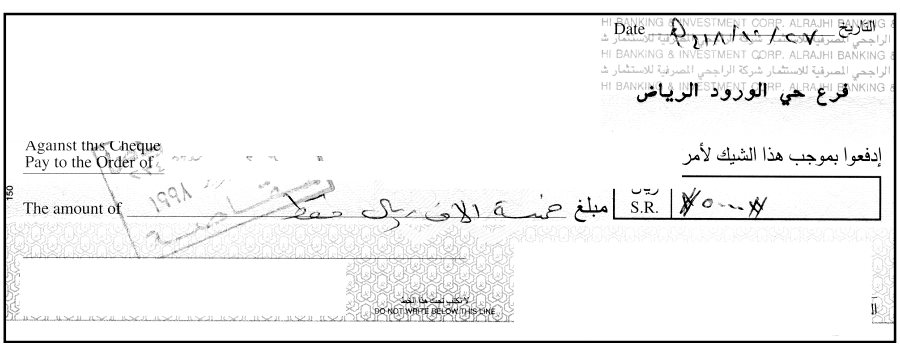

Fig. 2. Arabic check database image sample

The digits database was divided into two sets of touching and non-touching digits. If a digit contains at least one touching component to another neighboring digit, then it is located in the touching. Our ICR system is tested on the Arabic isolated non-touching digits database (10, 425 digits) to compare it to previously published results [17,18,20,21,23]. The database authors further divided the isolated digits into a training and a testing set, with the training set containing 70.89% of the database images and the testing set containing the remaining 29.11%.

A number of tagging errors are encountered in the isolated digits database. Fig. 3 shows some of those errors. It is clear that all of them, except for the last image, are due to segmentation errors and hence are not isolated digits. The last image was mistakenly tagged as '٩' (9). There are also many chopped images due to over-segmentation errors but these can be expected in handwritten databases.

Fig. 3. Tagging mistakes in the isolated digits database

3 Feature Extraction

The GSC features employs a multi-scale approach as they capture the narrow, intermediate, and large-scale qualities of the image. The gradient features detect the low-level gradient direction frequency and correspond to the narrow scale. The structural features compute several geometric characteristics such as the count of lines and corners at various directions and correspond to the intermediate scale. The concavity features correspond to the large-scale as they compute the count of large vertical and horizontal strokes, presence of holes, and direction of bays.

The Feature Extraction system first converts the input images into binary images by thresholding the gray levels using Otsu's method [24]. Next, we divide each image into n x m grids, where each row has uniform number of black pixels distributed over *n* rows, and each column has uniform number of black pixels distributed over *m* columns.

Fig. 4 shows different Arabic digits divided into 3by3, 4by4, 5by5, and 6by6 divisions, respectively. As can be seen from the figure, each horizontal section have same quantity of foreground pixels and each vertical section have same quantity of foreground pixels. Fig. 5 shows the extracted segments of Arabic digit '٨' (8). These segments are labeled Grid 1 to Grid 9.

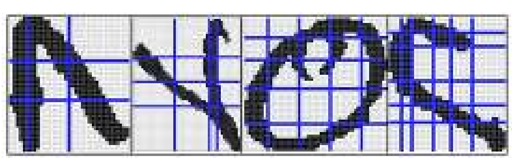

Fig. 4. Arabic digits divided into 4 different divisions

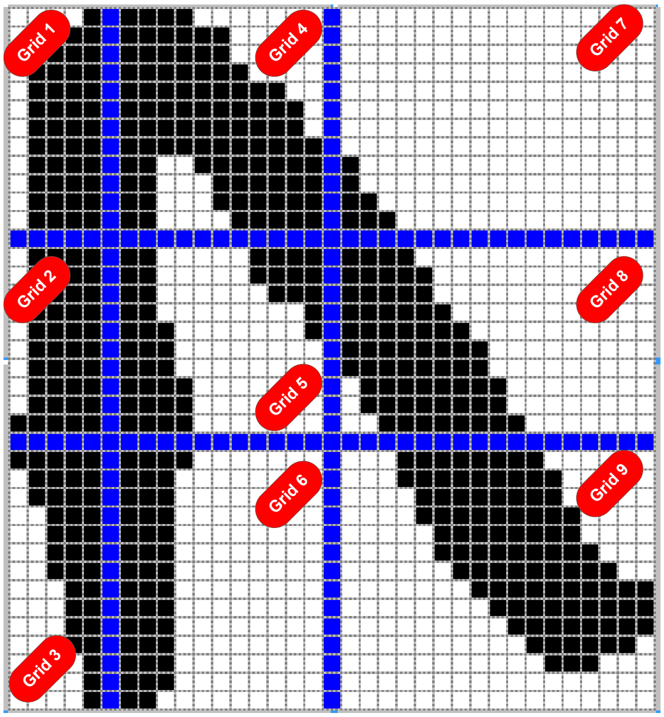

Fig. 5. Sample extracted segment for arabic digit ٨ (8)

The input image is split into multiple segments, and the multi-scale GSC features are extracted for each segment. The gradient features calculate the histogram for gradient direction for pixels in each grid, contributing twelve features per image segment. The structural features capture intermediate-strokes for each grid, providing twelve features per image segment. The concavity features consist of segment density, maximum strokes, and concavity attributes, with eight features per image segment. Then, all three types of features for each segment are joined to form one feature vector for each Arabic digit. The readers are referred to [1] for more details regarding the GSC feature extraction algorithm.

4 Classifiers

We used two classifiers (viz. SVM and HMM) for digits classification. Below we present brief details on each classifier and their configuration for the recognition task:

4.1 Support Vector Machines (SVM)

Vapnik and Cortes developed SVMs [25,26] as a statistical learning machine in the late 1990s. Within a short time, they became one of the most popular classification systems in data mining and pattern recognition applications, due to their high classification rates. Researchers successfully applied SVMs in many modern learning applications such as Optical Character Recognition (OCR), bioinformatics, document analysis, and image classification.

The following presents a brief description of the basic theory of SVM for a two-classification pattern recognition problem. Let $x \in R^d (i = 1,2, \ldots N)$ be a series of input vectors (set of samples), with corresponding labels $y_i \in \{+1, -1\} (i = 1,2, \ldots N)$. Here, $+1$ indicates the first class and -1 indicates the second class.

SVM seeks construct a binary classification system using the set of available input vectors by constructing a hyper-plane with the largest separation between the two classes' margin vectors. Thus, reducing the probability of misclassifying unknown test vectors. SVMs constructs this hyper-plane by its so-called kernel trick, or kernel function. The kernel function $K(x_i, x_j)$ maps the input vectors into a high- or infinite-dimensional feature space. Researchers have used several kernel functions in their applications. In this paper, we used one of the most popular kernel functions, the Radial Basis Function (RBF) kernel. After the mapping of the test vectors, SVM implements a decision function $f(x)$ to classify the future sample:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{N} y_i \alpha_i . K(x, x_i) + b\right)$$

solving a convex Quadratic Programming problem is used to obtain the coefficientsα_i.

4.2 Hidden Markov Models (HMM)

A Markov model is a finite state machine that either stays in its current state or jump to a new state at each time unit. A hidden Markov model presumes that a Markov model generates the unique feature vectors that represents a single digit. Hence, each move in time represent one part of the observed feature vector for each digit image. Each digit model λ has a probability $P(O, Q|\lambda)$ of generating the digit observation vector, O, through state sequence Q. This probability is calculated by multiplying the probabilities the transitions and the probabilities for the outputs:

$$P(O, Q|\lambda) = \pi_1 \times b_1(o_1) \times a_{12} \times b_2(o_2) \times a_{23} \times b_3(o_3) \cdots$$

where O = o_1, o_2, o_3, ..., is a sequence of digit observations; Q = q_1, q_2, q_3, …, is the state sequence; λ = (A,B,π); π_i, initial state transition; a_{ij} the transition probability from state i to state j; $b_i(o_m)$, the output probability at state i given observation m. Both i and j are 1, 2, ..,T; where T is the number of model states. The Baum-Welch algorithm estimates each model parameters in training phase.

However, the state series is hidden (and hence the name Hidden Markov Models). Therefore, the probability can be computed by summing all possible state sequences. In practice, this step consumes substantial time and space, and instead is replaced by the following approximation:

$$P(O|\lambda) = \max_{Q} \prod_{i=1}^{T} a_{q_{i-1} q_i} b_{q_i}(Oi)$$

Different model topologies can be used for classification and the left-to-right (Bakis) HMM topology – shown in Fig. 6 as a 6-state model – is the most common one for text recognition research. The left-to-right topology can withstand positional deviations for the Arabic digit image, and hence can be invariance to image rotation and skew. HMM models allow for different number of states for each digit model, however it is more common in research to use the same number of states for all digits, as was done in [27,28]. Abou-Moustafa et al. empirical experiments showed that having different number of states for each digit doesn't necessarily increase the performance of HMM-based classifier [29].

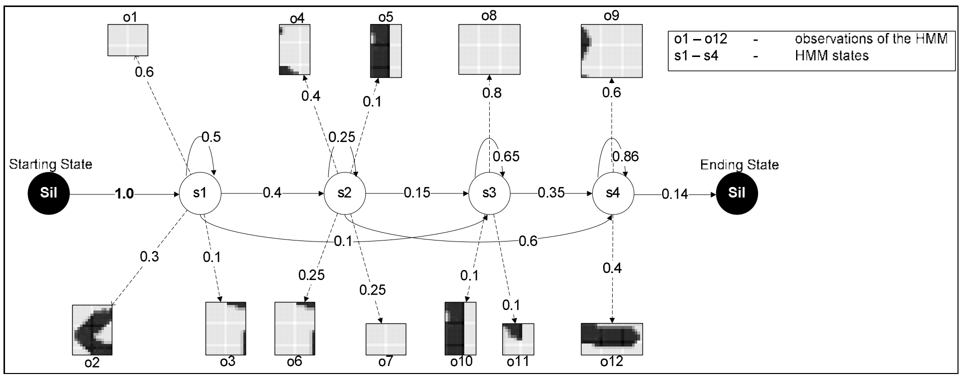

Fig. 6. Bakis model HMM with six states for digit ٤ (4)

5 Experiments and Analysis

Large number of experiments are conducted to assess the performance of the HMM and SVM classifiers. The original training set is divided into independent sub-training and validation sets. The verification setis used to optimize the HMM size of codebook and number of states. As for the SVM optimal parameters, we use a 10% V-fold on the original training data. These optimal parameters for both the HMM and SVM are expected to result in higher accuracy rate for the testing set without falling into over-fitting the classification models. The chosen optimal parameters for HMM and SVM are then used in constructing the classification models for further classification and analysis. The recognition rates of the presented techniques are compared with previously published recognition rates. The details of these experiments and analysis are presented next.

5.1 Hmm Classifier

The Hidden Markov Model Toolkit (HTK) [30] was used in the experimentation of digit recognition to assess the HMM classifier. Choosing the number of states and codebook size is usually done by experimentation [31]. The training samples given by CENPARMI are further divided into independent sub-training and verification sets as shown in Table 1. The sub-training set includes 70% of the available training data. We use verification data in selecting the optimal number of states and codebook size. The selected parameters are used in the HMM model which is trained using CENPARMI train data and tested using CENPARMI independent testing set. The verification data is also used to choose the optimal number of grid divisions for feature extraction.

Table 1. Distribution of training, sub-training, validation, and testing samples

Digit	# of training samples	# of sub-training samples	# of validation samples	# of testing samples
٠ (0)	3793	2655	1138	1574
١ (1)	782	547	235	304
٢ (2)	545	381	164	225
٣ (3)	362	253	109	144
٤ (4)	307	214	93	133
٥ (5)	649	454	195	263
٦ (6)	279	195	84	111
٧ (7)	233	163	70	109
٨ (8)	246	172	74	98
٩ (9)	194	135	59	74
Total	**7390**	**5169**	**2221**	**3035**

The sub-training and verification sets are also used to choose the optimal grid divisions for feature extraction. All grid size parameter estimation experiments are conducted using a 200 codebook size and different states ranging from 4to 8. Fig. 7 shows the optimal recognition rates on different grid divisions. The figure shows that the best recognition rate is achieved with 3 by 3 grid divisions at 97.39%.

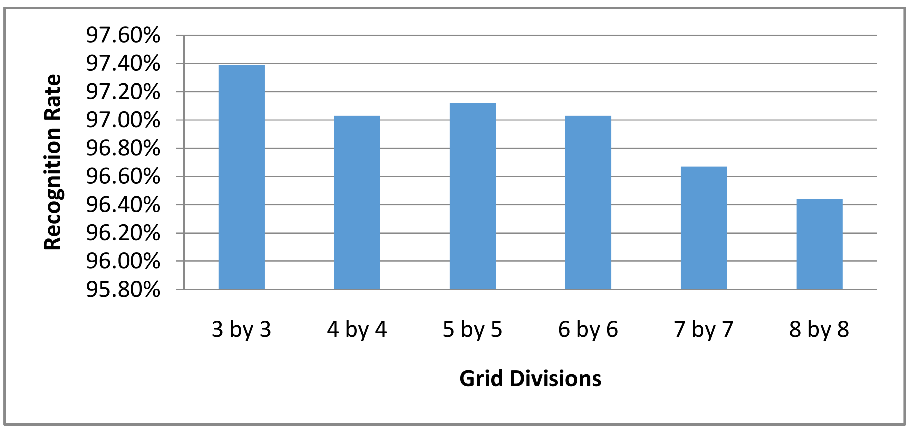

Fig. 7. Recognition rate at different divisions on verification data

After selecting the optimal grid divisions, another set of experiments are conducted to select the best codebook size. Bakis model topology is used which offers large flexibility in the modeling of duration and is very popular in the field of handwriting recognition [29,32]. Experiments are conducted using 3 by 3 grid divisions and a number of states ranging from four to eight states, different codebook sizes ranging from 100 to 1500 with steps of 100 are used. Fig. 8 shows the shows the recognition rates per codebook size with an optimal codebook size of 1100 and recognition rate of 98.15%.

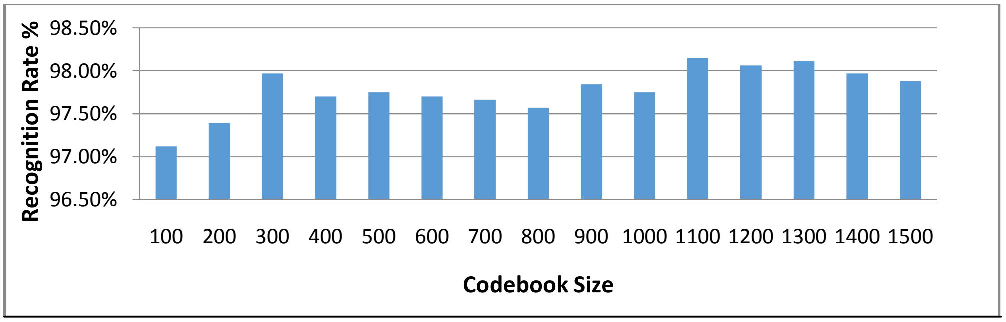

Fig. 8. Recognition rate per codebook size

Using grid divisions of 3 by 3 and codebook size of 1100, a number of experiments using four to eight states are executed.

Fig. 8 shows the recognition rates per number of states. Eight states are the maximum possible number of states for the proposed architecture. It also achieved the best recognition rate of 98.15%.

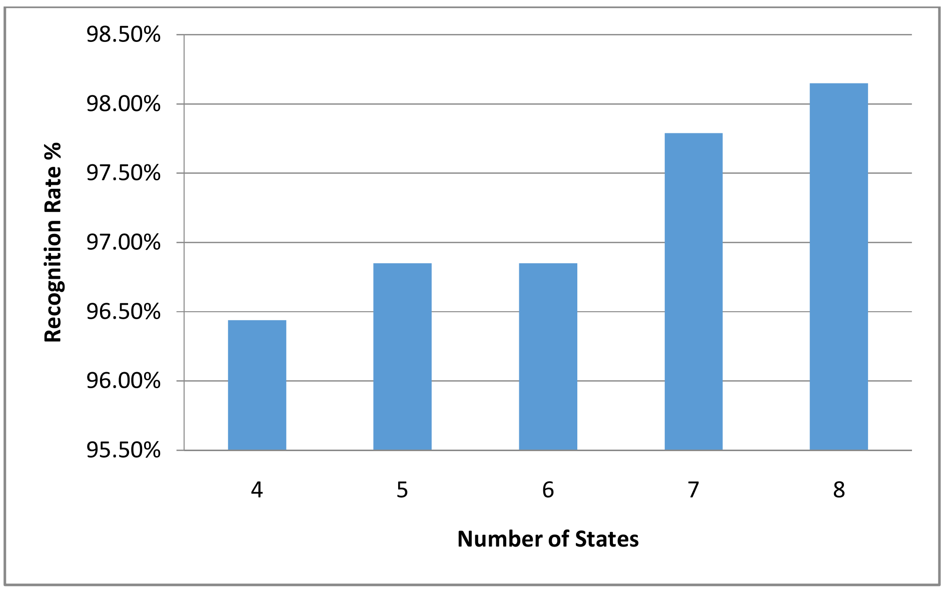

Fig. 8. Recognition rate per number of states.

An HMM with 8 states and a codebook size of 1100 are used. HMM is trained using CENPARMI training dataset (7390 samples) and tested using the test set (3035 samples). The confusion matrix for GSC features is shown in Table 2. The symbol %c represents the percentage of recognition rate, and %e the percentage of incorrectly labeled digits. The average recognition rate is 97.86%.

Table 2. Confusion matrix using GSC features with HMM classifier

	٠ (0)	**١ (1)**	**٢ (2)**	**٣ (3)**	**٤ (4)**	**٥ (5)**	**٦ (6)**	**٧ (7)**	**٨ (8)**	**٩ (9)**	**%c**	**%e**
٠ (0)	**1562**	6	0	0	0	5	1	0	0	0	99.24%	0.76%
١ (1)	1	**293**	2	1	0	0	2	2	1	2	96.38%	3.62%
٢ (2)	1	0	**216**	5	1	2	0	0	0	0	96.00%	4.00%
٣ (3)	0	1	5	**135**	1	1	0	0	1	0	93.75%	6.25%
٤ (4)	0	0	3	2	**127**	1	0	0	0	0	95.49%	4.51%
٥ (5)	1	0	0	0	0	**262**	0	0	0	0	99.62%	0.38%
٦ (6)	0	3	0	0	0	0	**108**	0	0	0	97.30%	2.70%
٧ (7)	0	0	0	6	0	1	1	**101**	0	0	92.66%	7.34%
٨ (8)	0	0	0	0	0	2	0	0	**96**	0	97.96%	2.04%
٩ (9)	0	1	0	0	0	0	3	0	0	**70**	94.59%	5.41%
											97.86%	2.14%

5.2 Svm Classifier

SVM parameters are fine-tuned using a 10% V-fold from the original training data to optimize its performance and accuracy. The estimated parameters are C = 7.937005 and γ = 0.046357. . The chosen optimal parameters for SVM are then used in constructing the classification models for further classification and analysis. Table 3 shows the recognition rates of digits ‘٠’ (0) to ‘٩’ (9) using GSC features and SVM classifier. The recognition accuracy for all digits is 99.04%.

Table 3. Confusion matrix using GSC features with SVM classifier

	٠ (0)	**١ (1)**	**٢ (2)**	**٣ (3)**	**٤ (4)**	**٥ (5)**	**٦ (6)**	**٧ (7)**	**٨ (8)**	**٩ (9)**	**%c**	**%e**
٠ (0)	**1571**	3	0	0	0	0	0	0	0	0	99.81%	0.19%
١ (1)	4	**299**	0	0	1	0	0	0	0	0	98.36%	1.64%
٢ (2)	2	0	**222**	0	0	1	0	0	0	0	98.67%	1.33%
٣ (3)	1	0	2	**141**	0	0	0	0	0	0	97.92%	2.08%
٤ (4)	1	0	5	0	**127**	0	0	0	0	0	95.49%	4.51%
٥ (5)	2	0	0	0	0	**260**	0	1	0	0	98.86%	1.14%
٦ (6)	0	1	0	0	0	0	**110**	0	0	0	99.10%	0.90%
٧ (7)	0	0	0	1	0	0	0	**108**	0	0	99.08%	0.92%
٨ (8)	1	0	0	0	0	0	0	0	**97**	0	98.98%	1.02%
٩ (9)	1	1	0	0	0	0	1	0	0	**71**	95.95%	4.05%
											99.04%	0.96%

Fig. 9 shows the recognition rate per digit for the HMM and SVM classifiers. SVM is superior to HMM in all digits, except for digit ‘٥’ (5). SVM dominance is most clear in digit ‘’ (7) where the difference of recognition rate is 6.42%.

5.3 Published Results

Sun et al. [17] achieved an optimal recognition rate of 95.97% by combining Gabor and Legendre features. However, it should be noted that instead of using the proposed training and testing sets by CENPARMI, they reported the use of the first 300 images of each class for training and the remaining 300 for testing. Thus, their total amount of training and testing samples are 3000. However, there are many digit classes that don’t have 300 or more samples for training or for testing as shown before in Table 1. Hu et al. [18] reported a 97.05% recognition rate. The numbers of used samples for training and testing sets were not reported. Juan et al. [20,21] used the same proposed training and testing distributions on their classifiers. In [20], they achieved an average recognition rate of 97.66% for their best recognizer. The average was computed from 50 runs of the standard experimental procedure. In [21] they achieved an average recognition rate of 97.82% for their best recognizer. Sadri et al. [23] reported a recognition rate of 94.14% using SVMs, compared with 91.25% obtained by MLP neural network classifier using the same features and test set. Mahmoud and Al-Khatib [19] reported recognition rates of 98.95%, 98.75%, 98.62%,97.21% and 94.43% achieved with SVM, 1-NN, 3-NN, HMM, and NM classifiers, respectively. Gimenez et al. [22] tested different parameters with a recognition rate of about 98% for their best configuration.

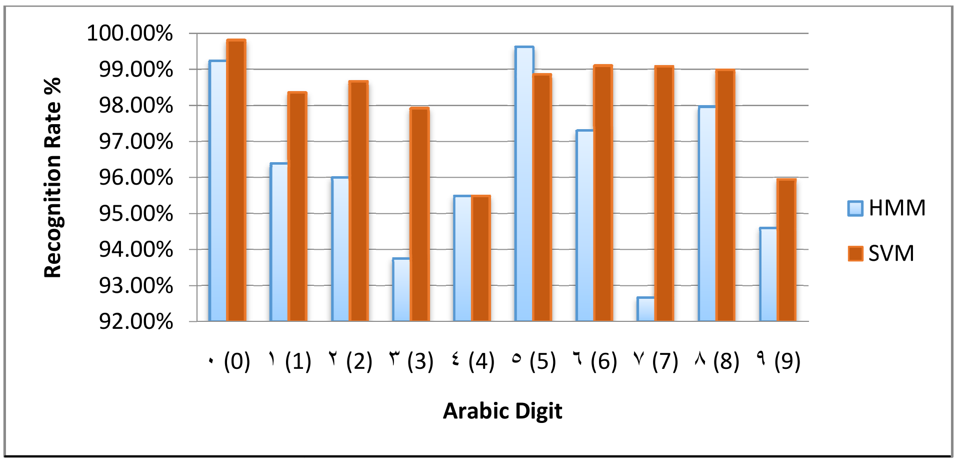

Fig. 9. Recognition rate per digit for SVM and HMM

Fig. 10 shows the recognition rates for our HMM and SVM classifiers with GSC features compared to other published results and sorted in descending order. It is clear from the figure that SVM has higher recognition rates than any other classifier.

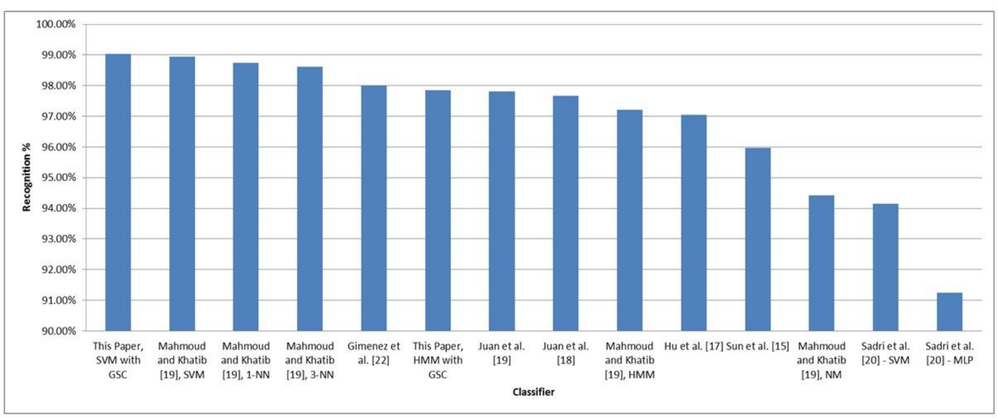

Fig. 10. Recognition rate for HMM and SVM classifiers compared to other classifiers

5.4 Analysis of Misclassified Samples

Our system misclassified 65 images from a total of 3035 images for the HMM classifier and 29 images for the SVM classifier. For the HMM classifier, five of these are due to writers writing digit '٣' (3) in a different style (with two upward strokes '٢') while the model was based on three upward strokes '٣'. Some of these errors are shown in Fig. 11, while other errors contain bad data or deformed digits strokes as shown in Fig. 12.

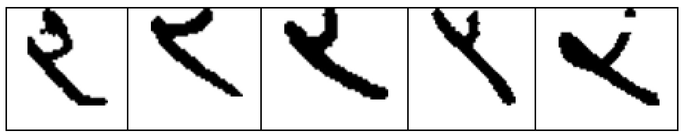

Fig. 11. Misclassified digit '٣' (3)

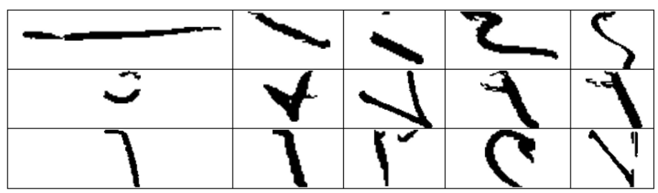

Fig. 12. Examples of misclassified digits due to bad data or deformed digits strokes

6 Conclusion

In this work, we present two system for handwritten Arabic digit recognition for multiple writers using a real-life dataset of 3000 bank checks. The systems employ multi-scale features that capture narrow, intermediate, and large-scale qualities of the image. The gradient features correspond to the narrow scale, the structural features correspond to the intermediate scale, and the concavity features correspond to the large-scale. These features are employed by two different statistical classifiers, HMM and SVM. The used database consists of 10,425 digits. The training and testing data sets, as constructed by CENPARMI, are used for the HMM and SVM classifiers.

The features in this work are multi-scale as they capture the narrow, intermediate, and large-scale qualities of the image. The gradient features detect the low-level gradient direction frequency and correspond to the narrow scale. The structural features compute several geometric characteristics such as the count of lines and corners at various directions and correspond to the intermediate scale. The concavity features correspond to the large-scale as they compute the count of large vertical and horizontal strokes, presence of holes, and direction of bays. A 3 by 3 grid size is used in extracting features.

Our recognition results are compared to other published work. The average accurate rates for all digits and for the HMM and SVM classifiers are 97.86% and 99.04%, respectively. It is shown that 65 digits out of 3035 are misclassified (2.14%) for the HMM classifier and 29 digits (0.96%) for the SVM classifier. Some of these errors are due to writers writing digit '٣' (3) in a different style (with two upward strokes '٢') while the model is based on three upward strokes '٣'. This can be addressed by having two models for digit '٣' (3). Other errors may be attributed to bad data or deformed digits' strokes. In general, the implemented features do not suffer from the well known digits combination problem (viz. ٥ (5) with ٠ (0), ٧ (7) with ٨ (8), ٩ (9) with ٦ (6), etc.).

The presented technique using robust features and both the HMM and SVM classifiers provides state-of-the-art recognition results on the CENPARMI database, as they reported a significantly higher recognition rates when compared to twelve previously published systems, especially for the SVM system. Future work for the researchers includes extending this work to complete bank-check document processing and recognition.

Acknowledgements

The first author would like to thank Qassim University for supporting this research and providing the computing facilities. The second author would like to thank King Fahd University of Petroleum and Minerals for supporting this research and providing the computing facilities.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Awaida S, Mahmoud S. A multiple feature/resolution scheme to Arabic (Indian) numerals recognition using hidden markov models. Signal Processing. 2009;89(6):1176–1184.

[2] Cheng-Lin L, Kazuki N, Hiroshi S, Hiromichi F. Handwritten digit recognition: investigation of normalization and feature extraction techniques. Pattern Recognit. 2004;37(2):265–279.

[3] Liu C, Nakashima K, Sako H, Fujisawa H. Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognit. 2003;36(10):2271–2285.

[4] Liu C, Koga M, Fujisawa H. Gabor feature extraction for character recognition: comparison with gradient feature. In Eighth International Conference on Document Analysis and Recognition. 2005;1:121–125.

[5] Teow L, Loe K. Robust vision-based features and classification schemes for off-line handwritten digit recognition. Pattern Recognit. 2002;35(11):2355–2364.

[6] Cheung K, Yeung D, Chin R. A bayesian framework for deformable pattern recognition with application to handwritten character recognition. IEEE Trans Pattern Anal Mach Intell. 1998;20(12):1382–1388.

[7] Al-Omari F, Al-Jarrah O. Handwritten Indian numerals recognition system using probabilistic neural networks. Adv Eng Informatics. 2004;18(1):9–16.

[8] Bouslama F. Structural and fuzzy techniques in the recognition of online arabic characters. Int J Pattern Recognit Artif Intell. 1999;13(7):1027–1040.

[9] Albert A, Ethem A, Akarun L. A selective attention-based method for visual pattern recognition with application to handwritten digit recognition and face recognition. IEEE Trans Pattern Anal Mach Intell. 2002;24:420–425.

[10] Hamid A, Haraty R. A neuro-heuristic approach for segmenting handwritten Arabic text. In ACS/IEEE International Conference on Computer Systems and Applications. 2001:110–113.

[11] Salourn S. Arabic hand-written text recognition. In 2001 ACS/IEEE International Conference on Computer Systems and Applications. 2001;106–109.

[12] Al-Ma'adeed S, Higgins C, Elliman D. Off-line recognition of handwritten Arabic words using multiple hidden Markov models. Knowledge-Based Syst. 2004;17(2–4):75–79.

[13] Al-Ma'adeed S, Higgins C, Elliman D. Recognition of off-line handwritten arabic words using hidden markov model approach. In Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02). 2002;3(3):30481.

[14] Touj S, Amara N, Amiri H. Arabic handwritten words recognition based on a planar hidden markov model. Int Arab J Inf Technol. 2005;2(4):318–325.

[15] Pechwitz M, Maergner V. HMM based approach for handwritten Arabic word recognition using the IFN/ENIT- database. In Seventh International Conference on Document Analysis and Recognition. 2003;890.

[16] Al-Ohali Y, Cheriet M, Suen C, Mohamedcheriet B. Databases for recognition of handwritten Arabic cheques. Pattern Recognit. 2003;36(1):111–121.

[17] Sun Q, Jin Z, Heng P, Xia D, Kong H. A novel feature fusion method based on partial least squares regression. In Third International Conference on Advances in Pattern Recognition. 2005;268–277.

[18] Hu Z, Yang J, Liu K, Sun J. Handwritten digit recognition based on multi-classifier combination. Jisuanji Xuebao/Chinese J Comput. 1999;22(4):369–374.

[19] Mahmoud S, Al-Khatib W. Recognition of Arabic (Indian) bank check digits using log-gabor filters. Appl Intell. 2010;35(3):445–456.

[20] Juan A, Vidal E. Bernoulli mixture models for binary images. In Proceedings of the 17th International Conference on Pattern Recognition. 2004;3(3):367–370.

[21] Juan A, García-Hernández J, Vidal E. EM initialisation for bernoulli mixture learning in proceedings of structural, syntactic, and statistical pattern recognition. Joint IAPR International Workshops Lisbon. 2004;635–643.

[22] Gimenez A, Andres-Ferrer J, Juan A, Serrano N. Discriminative bernoulli mixture models for handwritten digit recognition. In 2011 International Conference on Document Analysis and Recognition. 2011;558–562.

[23] Sadri J, Suen C, Bui T. Application of support vector machines for recognition of handwritten arabic/persian digits. In Second Conference on Machine Vision and Image Processing & Applications (MVIP 2003). 2003;1:300–307.

[24] Otsu N, A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern. 1979;9(1):66.

[25] Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3):273–297.

[26] Vapnik V. The nature of statistical learning theory. Springer; 1999.

[27] Bazzi I, Schwartz R, Makhoul J. An omnifont open-vocabulary OCR system for english and Arabic. IEEE Trans Pattern Anal Mach Intell. 1999;21(6):495–504.

[28] Bazzi I, LaPre C, Makhoul J, Raphael C, Schwartz R. Omnifont and unlimited-vocabulary OCR for English and Arabic. In Fourth International Conference on Document Analysis and Recognition. 1997;2(2):842–846.

[29] Abou-Moustafa K, Cheriet M, Suen C. On the structure of hidden Markov models. Pattern Recognit Lett. 2004;25(8):923–931.

[30] Young S, Evermann G, Gales M, Hain T, Kershaw D, Moore G, Odell J, Ollason D, Povey D, Valtchev V, Woodland P. The {HTK} Book, version 3.4. Cambridge, UK. Cambridge University Engineering Department. 2006;384.

[31] Günter S, Bunke H. Optimizing the number of states, training iterations and Gaussians in an HMM-based handwritten word recognizer. In Seventh International Conference on Document Analysis and Recognition. 2003;1:472–476.

[32] Fink G. Markov models for pattern recognition. Springer. 2007;248.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=583&id=6&aid=5129