Applied Computing and Informatics (2017) 13, 19–26

Saudi Computer Society, King Saud University
Applied Computing and Informatics
(http://computer.org.sa)
www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

Novel feature extraction technique for the recognition of handwritten digits

Abdelhak Boukharouba [a,*], **Abdelhak Bennia** [b]

[a] *Faculté des Sciences et de la technologie, Département d'Electronique et de Télécommunications, Université 8 Mai 1945 Guelma, BP 401, Guelma 24000, Algeria*
[b] *Faculté des Sciences de la Technologie, Département d'Electronique, Université Constantine 1, Algeria*

Received 15 December 2014; revised 1 May 2015; accepted 17 May 2015
Available online 27 May 2015

KEYWORDS

Feature extraction;
Feature selection;
Digit recognition;
Support vector machine

Abstract This paper presents an efficient handwritten digit recognition system based on support vector machines (SVM). A novel feature set based on transition information in the vertical and horizontal directions of a digit image combined with the famous Freeman chain code is proposed. The main advantage of this feature extraction algorithm is that it does not require any normalization of digits. These features are very simple to implement compared to other methods. We evaluated our scheme on 80,000 handwritten samples of Persian numerals and we have achieved very promising results.

1. Introduction

The recognition of handwritten script is a difficult task due to the different handwriting qualities and styles that are subject to inter-writer and intra-writer variations. Many recognition systems in many applications have been proposed in recent years where higher recognition accuracy is always desired. Typically, the recognition systems are adapted to specific applications to achieve better performance. They can be divided into three main steps: preprocessing step, feature extraction and selection step, and classification and verification step. Handwritten digit recognition problem can be seen as a subtask of the optical character recognition (OCR) problem. Unconstrained handwritten digit recognition has been applied to recognize amounts written on checks for banks or zip codes on envelopes for postal services, etc.

This paper focuses on feature extraction and classification. The performance of a classifier can rely as much on the quality of the features as on the classifier itself. A good set of features should represent characteristics that are particular for one class and be as invariant as possible to changes within this class [1]. Commonly used features in character recognition are: invariant moments [2], projections [3], zoning feature [4], Fourier descriptors [5], and contour direction histogram [6]. A feature set made to feed a classifier can be a mixture of such features.

While handwritten Latin digits recognition has been extensively investigated [7–10] through various techniques, little

* Corresponding author. Tel.: +213 777 08 32 32; fax: +213 37 20 72 68.
E-mail address: boukharouba_abdelhak@hotmail.com (A. Boukharouba).
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.aci.2015.05.001
2210-8327

work has been done for Arabic/Farsi digit recognition. Direction histograms using segmented characters from words in the CEDAR database [11] and transition information from the background to the foreground pixels in the vertical and horizontal directions of a character image [12] have been investigated. Later, feature extraction techniques generating local and global features were proposed [13] wherein local features were obtained from sub-images of the character including foreground pixel density information and directional information. The global features measured the fraction of the character appearing below the word baseline and the characters' width/height ratio. Furthermore, gradient features have been proposed for handwritten character recognition [14,15] where Awaidah and Mahmoud combined them with structural and concavity features for the recognition of Arabic (Indian) numerals using hidden Markov models (HMM) [16].

A probabilistic neural network (PNN) approach for the recognition of the handwritten Indian numerals [17] based on the center of gravity and a set of vectors to the boundary points of the digit has been presented however Montazer et al. [18] proposed a holistic approach using neuro-fuzzy inference engine to recognize the Farsi numeral characters. Finally, Impedovo et al. introduced a genetic algorithm based clustering approach using zoning features [19] whereas an adaptive zoning techniques for handwritten digit recognition are presented [20,21] where the features are extracted according to an optimal zoning distribution. The experimental tests show the effectiveness of the latter with respect to traditional approaches for zoning design.

The literature details many high accuracy recognition systems for handwritten Farsi digit database [22] used in our research. While an efficient feature set based on modified contour chain code has been proposed in [23], two types of feature sets based on modified chain-code direction frequencies in the contour pixels of input image and modified transition features have been presented [24]. A support vector machine (SVM) is proposed as classifier to recognize Persian isolated digits. Besides, combinational methods for improving the recognition rate of multi-class classifiers using confusion matrix and Genetic Algorithms have been proposed [25,26]. From the literature survey of the existing feature extraction techniques for character/digit recognition, most of them need digit normalization and consequently cannot preserve the shape of the input image for feature extraction step, which could react negatively to the recognition phase. For that reason, the main contribution of this work focuses on novel feature extraction approach where the digit image does not require any normalization.

This paper is broken down into five sections. Section 2 provides details on the feature extraction and selection techniques. Section 3 deals with the classifiers used for the recognition purpose, experimental results are discussed in Section 4, and finally Section 5 presents some conclusions and perspectives.

2. Feature extraction and selection

In this section, we describe two feature extraction techniques that are investigated in this work. The first is the chain code histogram (CCH) [27], which is developed to simply describe statistically the boundary of each digit's image. To eliminate the effect of contour direction distortion caused by digit image normalization, we compute the feature vector without normalizing digit image. Finally, we normalize feature vector by making its module equal to one. The second feature extraction technique builds on the white–black transition information in the vertical and horizontal directions of a digit image. Each transition feature is characterized by the area defined by the corresponding transition and normalized by dividing it by the whole digit's area.

The first feature set is the chain code histogram (CCH), which is a statistical measure for the directionality of the contour of a digit. In this work, we measure the slope between two successive points, which would give the angle made by the line joining them and the x-axis. Then the set of possible slopes is (0°, 45°, 90°, 135°), which are identical to the directions (180°, 225°, 270° and 315°). Thus, the directions between two successive pixels are encoded as 0, 1, 2 and 3 direction codes, respectively as shown in Fig. 1. Consequently, one of four direction codes is assigned to each two connected pixels.

The Canny algorithm is more robust to noise and more likely to detect true weak edges but some digit samples are with bad qualities so that after applying canny operator some redundant pixels are remained, which affect the contour following, these pixels should be removed. The redundant pixel removal process is based on the condition that every contour pixel can only have two 8-connected pixels around it. If there is a pixel which has more than two 8-connected pixels around it, the redundant pixels must be detected and removed. An example of redundant pixel removal process is shown in Fig. 2 where pixel 2 is a redundant one and must be removed. After removing redundant pixels, the contour becomes of unit thickness and consequently, we find that only 4 possible segments exist in the contour structure, which are considered as basic segments consisting of two pixels. Consequently, the feature extraction process is independent of the tracing direction.

To extract features from digit contour, we employ a histogram of the 4-chain code directions. The 4-bin histogram of chain code directions is computed where each bin represents the frequency of the one direction.

First, the smallest bounding box enclosing the digit contour is computed. If any of the number of rows or columns of the bounding box is not a multiple of 3, one or two rows/columns of zeros should be added. If only one row of zeros is to be added then it will be added on the right side of the image. If two rows of zeros are to be added then each one must be on one side and similarly for columns. Consequently, the image can be divided into 3 × 3 zones of equal area. The tracing process starts from the end point on the contour element, searching for the next nonzero pixel in any direction until end point is

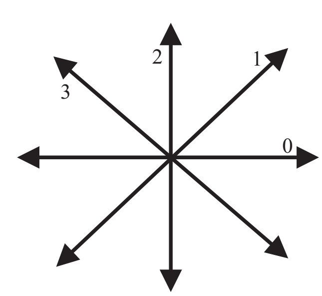

Fig. 1 Diagram of the 4-chain code directions.

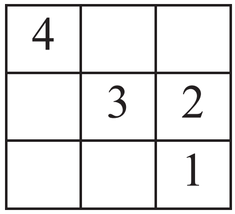

Fig. 2 The redundant pixel removal principle: pixel 2 is a redundant pixel and must be removed.

reached. The remaining contour elements are traced in the same way.

Fig. 3 shows an example of chain code histogram of the contour points of the middle-left and top-right zones of the digit ٣.

Afterward, the contour feature vector composed of 9×4 (36) components is normalized as follow.

$$f_i = \frac{h_i}{\|h_i\|}, \quad i = 1, 2, \ldots, 36 \tag{1}$$

where h_i is the ith bin of the tangent histogram of the whole digit, f_i is the ith component of the feature vector f, which is between 0 and 1, and $\| \ \|$ is the 2-norm. Thus, $\|f\| = 1$. Consequently, these features are a translation and scale invariant contour descriptors and they are independent of the way a contour is traced.

The second set, which is used as complementary description of digits, is based on the white–black transition information in the vertical and horizontal directions of a digit image. This technique is an extension of that presented in [28] where the transition description is based on the length of the transitions. However, this technique is based on the areas between these transitions and the bounding box of the digit as region descriptors.

Each transition feature is calculated as the ratio between the area defined by each transition type and the whole digit's area. The area is computed as the number of pixels between the left/top boundary of the bounding box of the digit image and the digit's edge defined by the transition location in the horizontal/vertical direction.

$$fh_k = \frac{ha_k}{w \cdot h}, \; fv_k = \frac{va_k}{w \cdot h} \text{ for } k = 1, 2 \text{ and } fh_3 = \frac{\sum_{k \geqslant 3} ha_k}{w \cdot h},$$
$$fv_3 = \frac{\sum_{k \geqslant 3} va_k}{w \cdot h}.$$

where ha_k and va_k are the areas of the regions defined by the edge of the kth transition in the horizontal and vertical direction, respectively. w and h are the width and the height of the digit image.

In the horizontal direction, the bounding box of each digit is divided horizontally in two equal parts. For each part, the three transition features are calculated in the horizontal direction. In the vertical direction, the three transition features are evaluated on the whole digit.

Fig. 4 shows an example of the different areas made by the white–black transitions of the digit ٨. In the horizontal direction we have only tow transition types of each part: the area formed by the first transition is at the left side of each part, the area formed by the second transition is at the right side of each part, and as we have not a third transition, the corresponding area is equal to zero.

In the vertical direction we have only one transition: the corresponding area is at the bottom side of the digit; and as we have only one transition, the second and third area is equal to zero.

Finally, relative area feature is calculated with respect to pre-fixed area ($w_0 \cdot h_0$): $ratio = w \cdot h / w_0 \cdot h_0$ where w and h are the width and the height of the digit image respectively. In this work the pre-fixed area $w_0 \cdot h_0$ is equal to 2000.

As a result, we obtain a feature vector of 46 components per digit.

Feature selection aims to reduce the dimensionality of the feature space for classification by selecting the most informative features. The most informative features are the ones that best separate the different classes. Feature selection has been used previously for many applications, yielding higher speed and reduced computational cost for the classification process.

One way to evaluate the pertinence of the extracted features is to calculate the discriminative power of each feature. In order to select the most discriminative features, we adopt the Fisher criteria, which compute directly the discriminative power D_k of each feature k as follow [29]:

$$D_k = \frac{1}{\overline{\sigma_k^2}}, \text{ where } \overline{\sigma_k^2} = \sum_{i=1}^{C} p(w_i)\sigma_{ik}^2 \tag{2}$$

where C is the number of classes, $p(w_i)$ is a prior probability of class w_i, and σ_{ik} is the variance of feature k according to the class i. These parameters are estimated from the training dataset.

The features must be ordered in the descending order in terms of their discriminative powers in order to select the most

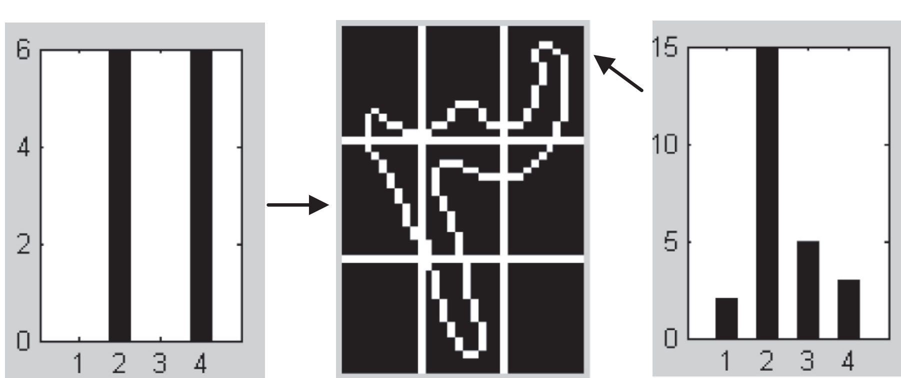

Fig. 3 Chain code histogram of the contour points of the middle-left and top-right zones of the digit ٣.

Fig. 4 Example of the different areas made by the white–black transitions of the digit ٨.

pertinent features. There is no theoretical criterion to calculate the threshold that separates the pertinent features from the redundant ones. The threshold is then determined with the help of many experiments by undertaking trainings on the selected features. Finally, we retain the threshold that provides the best performance in terms of recognition rate.

3. Support vector machines

SVM is a classifier derived from statistical learning theory first presented by Boser et al. [30]. SVMs were introduced in [31] as learning machines with capacity control for regression and binary classification problems. It has also been proved to be very successful in many other applications such as handwritten digit recognition, image classification, face detection, object detection, and text classification. In the case of classification, SVM try to find an optimal hyperplane that correctly classifies data points by separating the points of two classes as much as possible. For the linearly separable case, the support vector algorithm simply looks for the separating hyperplane with largest margin. This can be formulated as follows: suppose that all the training data satisfy the following constraints:

$x_i^T w + b \geqslant +1$ for $y_i = +1$ and $x_i^T w + b \leqslant -1$ for $y_i = -1$. Then the hyperplane $(x^T w + b)$ separates the data if and only if:

$$y_i(x_i^T w + b) \geqslant 1, \quad \forall i. \tag{3}$$

The optimal separating hyperplane is a margin classifier whose output is given by:

$$f(x) = \mathrm{sign}(x^T w + b) \tag{4}$$

where x is the input pattern, w is the weights vector, and b is the bias. The bias and the weights are computed by maximizing the margin $1/\|w\|$ subject to the constraint that the N training patterns are well classified and outside the margin:

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 \\ &\text{s.t. } y_i(x_i^T w + b) \geqslant 1, \quad i = 1, \ldots, N. \end{aligned} \tag{5}$$

with $y_i \in \{-1, 1\}$ representing the label of the training pattern xi. The solution corresponds to the saddle point of the primal Lagrangian:

$$Lp = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(x_i^T w + b) - 1] \tag{6}$$

where the α_i are the Lagrange multipliers. This problem leads to the maximization of the dual Lagrangian with respect to α_i:

$$\begin{aligned} &L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j}^{N} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \\ &\text{s.t. } \alpha_i \geqslant 0, \quad i = 1, \ldots, N, \\ &\sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{7}$$

This is a standard quadratic problem, where a global maximum α_i can always be found and w can be recovered as:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{8}$$

Many of α_i are zero, which implies that w is a linear combination of a small number of data. The set of elements x_i with non-zero α_i are called *support vectors*.

Then, the resulting separating rule is:

$$f(x) = \mathrm{sign}\left(\sum_{support \ \ vectors} y_i \alpha_i (x_i^T x) + b \right) \tag{9}$$

The SVs are the training patterns that lie on the margin boundaries. An advantage of this algorithm is its sparsity since only a small subset of the training examples are used to compute the output of the classifier. Fig. 5 represents a binary classification problem where filled circles and squares are the training data while hollow circles and triangles are the testing data.

In case of such separating hyperplane does not exist, we introduce a set of slack variables ξ_i to allow points inside the margin during the training.

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \\ &\text{s.t. } y_i(x_i^T w + b) \geqslant 1 - \xi_i, \quad i = 1, \ldots, N. \end{aligned} \tag{10}$$

where penalty parameter C is used to tune the trade-off between the classification errors and the maximization of the margin. The formulation (6) is often called soft margin

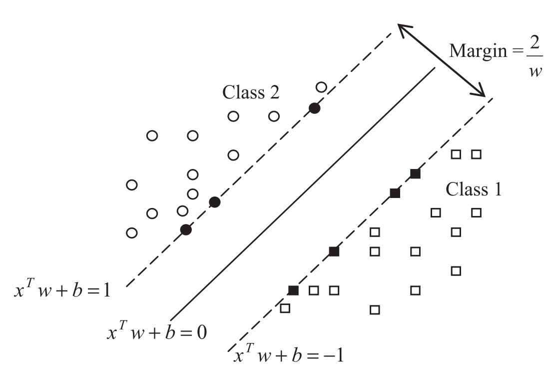

Fig. 5 Linear separating hyperplanes for the separable case: filled circles and squares are the support vectors.

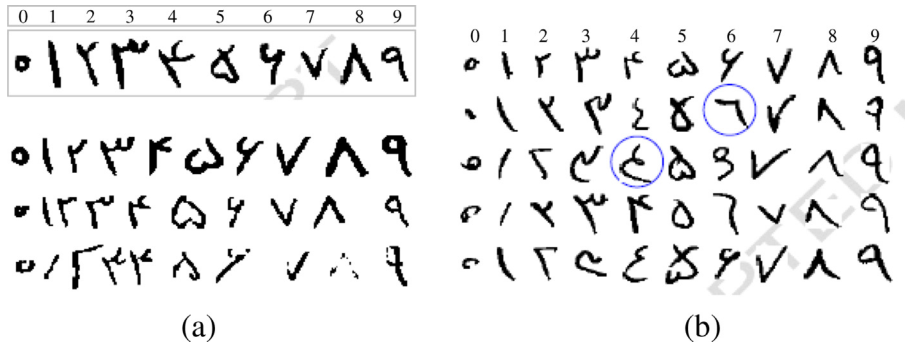

Fig. 6 Digit samples of handwritten Farsi digits [22], (a) different qualities, (b) different styles.

SVM. This new formulation leads to the same dual problem but with box constraints on the Lagrange multipliers:

$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, N, \tag{11}$$

The tuning of the hyperparameter C is a delicate task. A common method is to perform a grid search, i.e. to test many values of C and estimate for each the generalization error.

This approach is valid whenever the set of points of the two classes are linearly separable. Nevertheless in real data this is usually not the case. In order to work with non-linear decision boundaries the key idea is to transform x_i to a higher dimension space using a transformation function, so that in this new space the samples can be linearly separable. SVM solve these problems using kernels. One only has to calculate the inner products of the vectors in the feature space via the kernel function $K(\cdot, \cdot)$. This is the *kernel trick* that allows the construction of a decision function that is nonlinear in the input space but equivalent to a linear decision function in the feature space:

$$f(x) = \text{sign}\left(\sum_{support\ \ vectors} y_i\alpha_i K(x_i, x) + b\right) \tag{12}$$

where $K(x_i, x)$ stands for the kernel function. Typical kernel functions are:

RBF (Gaussian) kernel: $K(x_i, x) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right)$.

Sigmoid kernel $K(x_i, x) = \tanh(\gamma(x^T x_i) + c)$.

Polynomial kernel $K(x_i, x) = (\gamma\, x^T x_i + c)^d$.

There are two common methods to solve a multi-class problem with binary classifiers such as SVMs: one-against all (or one-vs-rest) and one-against-one. In the one-against-all scheme, a classifier is built for each class and assigned to the separation of this class from the others. For the one against-one method, a classifier is built for every pair of classes to separate the classes two by two. Another approach to the recognition of n different digits is to use a single n-class SVM instead of n binary SVM subclassifiers with the one-against-all method, thus solving a single constrained optimization problem.

4. Experiments and results

In this section we present the experimental results to illustrate the benefits of the chain code histogram combined with the transition features in digit recognition field. We evaluate our method on a large handwritten dataset of Farsi digits, named "Hoda" [22]. This dataset consists of ten digit classes from 0 to 9 (٠١٢٣۴۵۶٧٨٩). For experimental results, 80,000 handwritten samples are considered; 6000 samples per class for training and 2000 samples per class for testing. Fig. 6 shows some digit samples extracted from the used database with different styles and qualities.

Firstly, Canny operator is used for digit contour extraction then the bonding box of each digit contour is divided into nine equal zones. Within each zone the redundant pixels is removed and the contribution of the 4-chain codes are counted in the corresponding histograms. The transition features are calculated as shown above: in the horizontal direction, the bounding box of each digit is divided horizontally in two equal parts. For each part, the transition features are calculated in the horizontal direction. In the vertical direction, the transition features are evaluated on the whole digit. Finally, relative area is calculated. The used feature vector is equal to 10.*f*, where *f* is the vector of the 46 features.

Secondly, the proposed architecture of SVM based classifier is composed of ten one-against-all SVMs. The classifier structures were empirically set as follows. The RBF kernel is used where the variance parameter σ is equal to 1 ($\sigma = 1$) and the hyperparameter C is equal to 100 ($C = 100$).

Table 1 shows the details of misrecognition and recognition accuracy of each digit where the recognition accuracy of 98.48% is obtained on the whole test set (20,000 samples). It may be noted that out of 305 misrecognized samples, 183 (60%) samples belong to {٢, ٣, ۴} group and 122 (40%) samples belong to the remaining digits. Thus, the major misrecognized digits are among 2, 3 and 4 digits. To improve the recognition accuracy, we use a scheme similar to one introduced in [24]. We utilize a classifier composed of seven one-against-all SVMs where {٠, ١} and {٢, ٣, ۴} represent two separate classes as shown, in Fig. 7. After that, we use one against-all SVM to separate the combined {٠,١} class. For the recognition of digits ٢, ٣, ۴, we also use twice one-against-all SVM.

Fig. 7 illustrates the recognition system where arrow weights represent the number of the misrecognized digits of each class. Out of a total of 20,000 digits in the testing set, there are 291 digits that are not successfully recognized. Consequently, we obtain 98.55%, which showed slight improvement when compared with the first scheme. Note that we have got an accuracy of 100% on the training samples (60,000 samples) and also an accuracy of 100% when the training is made on the whole dataset (80,000 samples).

Table 1 Misrecognition and recognition accuracy of the ten digits.

Number (Farsi)	Number of misclassified digits	Recognition accuracy (%) (test)
0	14	99.30
1	7	99.65
2	42	97.90
3	109	94.55
4	32	98.40
5	15	99.25
6	30	98.50
7	20	99.00
8	4	99.80
9	32	98.40

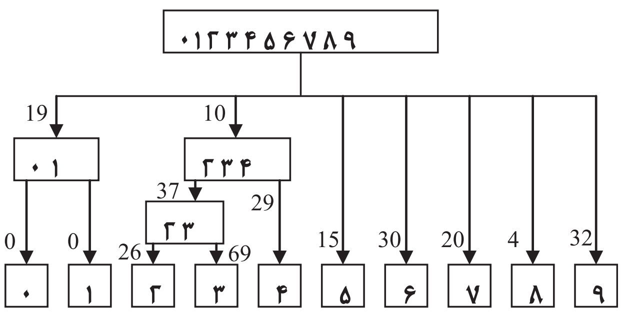

Fig. 7 Recognition scheme: arrow weights represent the number of the misrecognized digits.

The performances of most of the works available for Persian numerals are presented in [23–26] where detailed comparisons with recent published works are discussed. Table 2 shows a comparison with the most excellent existing works, that are to the best of our knowledge the only works in the literature that deal with Farsi digits composing Hoda dataset.

From Table 2 it is clear that the highest recognition rate is 99.02% [24] when 196 features are used for training. In our work, we reached an interesting recognition rate of 98.55% with only 46 features.

The second experiment investigates the performance of SVM classifier based on the reduced feature sets.

In order to yield higher speeds and reduced computational cost for the classification process, we choose to reduce the number of features involved in the training and testing stages of SVM.

Table 2 Comparisons to other systems in the literature.

Algorithms	Number of features	Dataset size		Accuracy (%)	
		Train	Test	Train	Test
[23]	196	60,000	20,000	99.99	98.71
[23]	196	80,000	–	99.37	–
[24]	196	60,000	20,000	99.99	99.02
[25]	106	60,000	10,000	–	98.89
[26]	106	40,000	20,000	–	97.12
Proposed algorithm	46	60,000	20,000	100	98.55
	46	80,000	–	100	–

Table 3 Discriminative powers of 46 used features.

Transition features	Chain code histogram features		
0.1553	0.1241	0.2023	0.1667
0.2092	0.1926	0.1504	0.1824
0.2671	0.1769	0.1732	0.1679
0.3647	0.1887	0.1704	0.3490
0.8217	0.2135	0.1553	0.2145
0.2760	0.1451	0.1562	0.2129
0.3711	0.2191	0.1251	0.3142
0.3879	0.3249	0.1668	0.1790
0.1391	0.2594	0.3810	0.1755
0.2024	0.1415	0.1901	0.2491
	0.2287	0.1318	0.2347
	0.2178	0.1849	0.2302

Table 3 illustrates the individual discriminative power of each feature. The first column represents the discriminative powers of the features extracted from transitions and the relative area respecting the order described in the transition feature procedure. The discriminative powers of CCH features are presented in a 3×3 grid with 4 values in each cell matching the corresponding image region. The threshold value of discriminative powers used to select the pertinent primitives must be determined by practical tests. First, the primitives have been ranked in descending order of their discriminative powers. The threshold is then determined by undertaking trainings on the selected features. We retain the threshold that provides the best performance in terms of recognition rate.

Table 4 shows the best results obtained for different numbers of selected features using the SVM classifier with the same parameters used above ($\sigma = 1$ and $C = 100$).

The first column represents the value of threshold of the discriminative power, second column represents the number of selected features, and the last column represents the recognition rate achieved on the test data set.

The highest recognition rate achieved here is 98.46% using 40 features and 98.44% using only 36 features, which are slightly less than the rate obtained with the 46 features. It is then sufficient to retain only the 36 primitives whose discriminative powers are the most pertinent.

Moreover, the discriminative powers help us to analyze the pertinence of the different features. From Table 3, among the 10 less pertinent features there are only two from the transition features and the remaining are from the CCH features. We can also notice that the most pertinent feature is the area made by the second transition of the low part in the horizontal direction with discriminative power of 0.8217, which prove the pertinence of the transition features.

The best performance/complexity is obtained with an SVM classifier ($\sigma = 1$ and $C = 100$) using 36 primitives, which is more efficient than the system trained on 106 primitives [25].

Table 4 Effect of the size of the feature vector on the recognition rate.

Threshold of discriminative power	Number of selected features	Recognition rate (%)
0.1500	40	98.46
0.1600	36	98.44

Moreover, it is important to note that the architecture of a classifier, by reducing its size, requires less storage capacity for its parameters.

We can augment the feature space using some structural features to efficiently remove some confusion and to achieve best results but the main purpose of this work is to demonstrate the efficiency of these new features.

The most noticeable improvement is that a 100% recognition rate is achieved in the training phases for all digits (60,000 and 80,000 samples), which is better than all of those presented in the literature. We believe that our results are very competitive and quite promising since we used only 36 simple features.

As we have achieved a rate of 100% on training datasets, then we can achieve a higher accuracy rate if we train SVM classifiers on appropriate support vectors, because the generalization property of an SVM does not depend on all the training data, but only support vectors.

Finally, we can see from the results that the achieved accuracy is due to the discriminatory power of features and the regression capabilities of SVM classifiers.

5. Conclusion

This paper presents a system for the recognition of the handwritten Persian numerals that could be used for automatic reading of numerical amounts of checks. The main contribution of this work focuses on feature extraction where a novel feature set based on transition information in the vertical and horizontal directions of a digit image combined with the well-known chain code histogram (CCH) is discussed and compared with others in the literature. The classification system is based on SVM, which is considered one of the most powerful classification techniques and is now widely used in many pattern recognition applications. The results of our experiments show that feature selection procedure reduces the dimensionality of the feature space without affecting the performance of the classifier where the system can maintain high performance with less computational complexity comparing to the systems in the literature.

From experimental results, it is evident that our system resulted good performance. We noted that most of misclassified samples were from classes of ۲, ۳, and ۴, which are similar in shapes where their recognition is sometimes difficult even for human being.

Among the most important advantages of this feature extraction algorithm; it does not require any normalization of digits where the most of the published works need digit normalization, which degrade the image quality. These features are also very simple to implement compared to other methods.

It is obvious that to improve the performance of proposed system further, we need to investigate more on sources of errors. Potential features other than the presented ones may exist. In future, we plan to use some structural features like concavity analysis which may remove some of confusions among similar classes.

Concerning the SVM classifiers, the tuning of the hyperparameter C is a delicate task. We are sure that the RBF kernel parameters (σ and C) used in our experiments are not the best choices and are not implemented optimally because we have tried only a few experiments to choose them. Thus, we can improve the performance of the system by testing many values of these parameters and estimate for each the generalization error.

Finally, because we have achieved a rate of 100% on training datasets then as perspective we will try to train automatically SVM classifiers on the support vector set that represent the more delicate examples instead of the whole training set to achieve higher recognition accuracy.

References

[1] F. Lauer, C.Y. Suen, G. Bloch, A trainable feature extractor for handwritten digit recognition, Pattern Recogn. 40 (2007) 1816–1824.

[2] C.H. The, R.T. Chin, On image analysis by the methods of moments, IEEE Trans. Pattern Anal. Mach. Intell. 10 (1988) 496–513.

[3] A.L. Koerich, Unconstrained handwritten character recognition using different classification strategies, in: Proceedings of International Workshop on Artificial Neural Networks in Pattern Recognition, 2003.

[4] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, Automatic recognition of handwritten numerical strings: a recognition and verification strategy, IEEE Trans. Pattern Recogn. Mach. Intell. 24 (11) (2002) 1438–1454.

[5] H. Kauppinen, T. Seppanen, M. Pietikamen, An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 207–210.

[6] K.M. Mohiuddin, J. Mao, A comprehensive study of different classifiers for handprinted character recognition, Pattern Recogn. (1994) 437–448.

[7] X.X. Niu, C.Y. Suen, A novel hybrid CNN–SVM classifier for recognizing handwritten digits, Pattern Recogn. 45 (2012) 1318–1325.

[8] Z. Man, K. Lee, D. Wang, Z. Cao, S. Khoo, An optimal weight learning machine for handwritten digit image recognition, Signal Process. 93 (2013) 1624–1638.

[9] E. Mohebi, A. Bagirov, A convolutional recursive modified self organizing map for handwritten digits recognition, Neural Netw. 60 (2014) 104–118.

[10] M. Hanmandlu, O.V. Ramana Murthy, Fuzzy model based recognition of handwritten numerals, Pattern Recogn. 40 (2007) 1840–1854.

[11] H. Yamada, Y. Nakano, Cursive handwritten word recognition using multiple segmentation determined by contour analysis, IECE Trans. Inform. Syst. E79-D (1996) 464–470.

[12] P.D. Gader, M. Mohamed, I.H. Chiang, Handwritten word recognition with character and inter-character neural networks, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 27 (1997) 158–164.

[13] F. Camastra, A. Vinciarelli, Combining neural gas and learning vector quantization for cursive character recognition, Neurocomputing 51 (2003) 147–159.

[14] C.L. Liu, Normalization-cooperated gradient feature extraction for handwritten character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (8) (2007).

[15] C.L. Liu, C.Y. Suen, A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters, Pattern Recogn. 42 (2009) 3287–3295.

[16] S.M. Awaidah, S.A. Mahmoud, A multiple feature/resolution scheme to Arabic (Indian) numerals recognition using hidden Markov models, Signal Process. 89 (2009) 1176–1184.

[17] F.A. Al-Omari, O. Al-Jarrah, Handwritten Indian numerals recognition system using probabilistic neural networks, Adv Eng. Inform. 18 (2004) 9–16.

[18] G.A. Montazer, H.Q. Saremi, V. Khatibi, A neuro-fuzzy inference engine for Farsi numeral characters recognition, Expert Syst. Appl. 37 (9) (2010) 6327–6337.

[19] S. Impedovo, F.M. Mangini, G. Pirlo, A genetic algorithm based clustering approach for improving off-line handwritten digit classification, in: International Conference on Information Science, Signal Processing and their Applications (ISSPA), 2012.

[20] S. Impedovo, F.M. Mangini, G. Pirlo, A new adaptive zoning technique for handwritten digit recognition, in: ICIAP, 2013.

[21] F.M. Mangini, G. Pirlo, Adaptive zoning design by supervised learning using multi-objective optimization, Int. J. Comput. Intell. Appl. 13 (01) (2014).

[22] H. Khosravi, E. Kabir, Introducing a very large dataset of handwritten Farsi digits and a study on the variety of handwriting styles, Pattern Recogn. Lett. 28 (10) (2007) 1133–1141.

[23] A. Alaei, U. Pal, P. Nagabhushan, Using modified contour features and SVM based classifier for the recognition of Persian/Arabic handwritten numerals, in: 7th Int. Conf. on Advances in Pattern Recognition, 2009 'a', pp. 391–394.

[24] A. Alaei, P. Nagabhushan, U. Pal, Fine classification of unconstrained handwritten Persian/Arabic numerals by removing confusion amongst similar classes, in: 10th Int. Conf. on Document Analysis and Recognition, 2009 'b'.

[25] H. Parvin, H. Alizadeh, B. Minaei-Bidgoli, M. Analoui, A scalable method for improving the performance of classifiers in multiclass applications by pairwise classifiers and GA, in: 4th Int. Conf. on Networked Computing and Advanced Information Management, 2008.

[26] H. Parvin, H. Alizadeh, M. Moshki, B. Minaei-Bidgoli, N. Mozayani, Divide & conquer classification and optimization by genetic algorithm, In: 3rd Int. Conf. on Convergence and Hybrid Information Technology, N., 2008.

[27] J. Iivarinen, A. Visa, Shape recognition of irregular objects, in: David P. Casasent (Ed.), Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling, SPIE 2904, 1996, pp. 25–32.

[28] A. Boukharouba, A. Bennia, Recognition of handwritten Arabic literal amounts using a hybrid approach, Cogn. Comput. 3 (2) (2011) 382–393.

[29] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990.

[30] B.E., Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: 5th Annual Workshop on Computational Learning Theory, 1992, pp. 144–152.

[31] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.