A Novel Framework for Farsi and Latin Script Identification and Farsi Handwritten Digit Recognition

Alireza Behrad, Malike Khoddami and Mehdi Salehpour

Abstract— **Optical character recognition is an important task for converting handwritten and printed documents to digital format. In multilingual systems, a necessary process before OCR algorithm is script identification. In this paper novel methods for the script language identification and the recognition of Farsi handwritten digits are proposed. Our method for script identification is based on curvature scale space features. The proposed features are rotation and scale invariant and can be used to identify scripts with different fonts. We assumed that the bilingual scripts may have Farsi and English words and characters together; therefore the algorithm is designed to be able to recognize scripts in the connected components level. The output of the recognition is then generalized to word, line and page levels. We used cluster based weighted support vector machine for the classification and recognition of Farsi handwritten digits that is reasonably robust against rotation and scaling. The algorithm extracts the required features using principle component analysis (PCA) and linear discrimination analysis (LDA) algorithms. The extracted features are then classified using a new classification algorithm called cluster based weighted SVM (CBWSVM). The experimental results showed the promise of the algorithms.**

Index Terms— **Curvature scale space, Script identification, Optical character recognition, CBWSVM, Clustering, Handwritten digit recognition, PCA, PCA-LDA .**

I. INTRODUCTION

TODAY, high volume of available printed and paper documents are converted to digital image documents by use of scanners or cameras. Storage, retrieval and efficient management of these image documents are very important in many applications, such as office automation and digital libraries. It is sometimes necessary to retrieve required information from such digital documents. Therefore optical character recognition systems have been developed which are techniques for converting text images into computer readable/editable texts. In multilingual documents, script identification is a main stage before applying OCR algorithms. Among different applications of OCR, the recognition of handwritten digits has many usages including the recognition of digits in post office and bank applications [1]. Different algorithms and features are used for the identification of digits and characters. In [2],[3] the combination of classifiers are used for recognition of handwritten Farsi digits. In [4] normalized Gabor filter bank with zero DC response is used for indexing and the retrieval of character images. However the algorithm is sensitive to selection of Gabor filter parameters including bandwidth and orientation.

In [5] one-dimension geometric moments with Bayes classifier is used for the recognition of Arabic characters; however the method is only tested for the recognition of printed documents and the efficiency of the algorithm has not been demonstrated for the recognition of the handwritten characters.

In [6] the recognition of Persian digits is performed by overlaying different normalized images of one class and counting the number of black pixels in different pixel coordinates. The resultant image representing the number of black pixels in different pixel coordinates is used for the classification of test images. However the method is sensitive to the location of digits in the area and outliers. In addition, the rotation of the digits gives rise to the generation of different patterns than patterns of database and hence erroneous recognition. Neuro-fuzzy algorithm is another method for the recognition of handwritten characters proposed by [7]. However the method needs the intervention of human experts in some parts of the algorithm.

Script identification is a main stage before applying OCR algorithms in multilingual documents. Different methods have been presented for script identification which can be divided into two well-known groups including algorithms based on global features and algorithms based on local features. The first group is based on global features of the text such as textures. For example, in [8] co-occurrence matrix and wavelet analysis are used to extract texture features for differentiation between eight languages. In [9] and [10] recognition of scripts is performed in word-level using texture analysis through Gabor filter. The second group of script identification algorithms is based on local analysis which relies on local features and mostly acts on word and connected components levels. Features in these methods are further divided into four categories including 1- Structural or geometric features, 2- Morphological features, 3- Statistical features and 4- Spatial spread features. Structural features include features such as physical size of

Alireza Behrad is with the Faculty of Engineering, Electrical Engineering Department of Shahed University Tehran, Iran. (phone: (+98 - 21) 51212036; e-mail: behrad@shahed.ac.ir).

Malike Khoddami is Msc. Student of Faculty of Engineering, Shahed University Tehran, Iran (e-mail: khoddami@shahed.ac.ir).

Mehdi Salehpour is Msc. Student of Faculty of Engineering, Shahed University Tehran, Iran (e-mail: salehpour@shahed.ac.ir).

DOI: 10.2298/JAC1001017B

the blocks, black pixels density, width, height, aspect ratio, area and so forth [11], [12]. Morphological features are general description of text, according to script and connected components, regardless of its physical structure. The morphological features such as number of holes and upward concavities are related to inherent shape features of each script [12], [13]. Statistical features are simple mathematical approaches dealing with distribution of pixels in the character image. This set of structural features are mostly based on connected components inside the specified image blocks; such as mean and variance of the width, height, ratio and the area of connected components.
Statistical features are extracted at higher levels such as words, lines and text blocks [12], [13]. Spatial spread features provide information about scattering of ascenders and descenders, regional pixels concentration and characters density [10], [11]. In Latin based languages that have similar texture characteristics, identification is performed using word and character shape coding. Word shape coding convert the word images into shape codes using shape characteristics [14], [15].

In this paper we propose new algorithms for Farsi and Latin script identification and Farsi handwritten digit recognition. We assumed that the bilingual scripts may have Farsi and English words, characters and digits together; therefore we used new local features based on curvature scale space (CSS) representation to identify scripts in different levels [16]. The proposed algorithms can operates in different levels such as connected component, word, line and page levels.

We used cluster based weighted support vector machine for the classification and recognition of Farsi handwritten digits that is reasonably robust against rotation and scaling [17]. We used two different methods for feature extraction as well as two different methods for classification and the results are compared with each other and those of other methods. Experimental results showed the effectiveness of the proposed algorithms.

The structure of this paper is as follows: in the next section we review the fundamentals of curvature scale space representation, and the proposed algorithm for script identification. In section 3 the proposed algorithm for recognition of Farsi handwritten digits are described. Experimental results are given in section 4 and conclusions appear in section 5.

II. SCRIPT IDENTIFICATION IN FARSI AND LATIN BILINGUAL DOCUMENTS

A. Curvature scale space representation of planar curves

Planar curves Γ in an image can be represented as parametric vector equation as follow:

$$\Gamma = r(u) = (x(u), y(u)) \tag{1}$$

where u is an arbitrary parameter. The curvature of the parametric curve is expressed using the following equation [18]:

$$k(u) = \frac{\dot{x}(u)\ddot{y}(u) - \dot{y}(u)\ddot{x}(u)}{(\dot{x}(u)^2 + \dot{y}(u)^2)^{3/2}} \tag{2}$$

To make the curvature values robust against the curve noises, the evolved version of the curve is used to calculate the curvature. The evolved version of a curve is calculated by convolving the curve with a 1D Gaussian kernel of width σ as follows:

$$Y(u,\sigma) = y(u) \otimes g(u,\sigma) \tag{3}$$

$$X(u,\sigma) = x(u) \otimes g(u,\sigma) \tag{4}$$

$$\Gamma_\sigma = R(u,\sigma) = (X(u,\sigma), Y(u,\sigma)) \tag{5}$$

where $\otimes$ represents 1D convolution. The curvature of the evolved curve is calculated using the following equation:

$$K(u,\sigma) = \frac{X_u(u,\sigma)Y_{uu}(u,\sigma) - X_{uu}(u,\sigma)Y_u(u,\sigma)}{(X_u(u,\sigma)^2 + Y_u(u,\sigma)^2)^{3/2}}) \tag{6}$$

In the above equation the first and second derivatives of X and Y are estimated using the first and second derivative of Gaussian function.

The function defined by the equation, $K(u,\sigma) = 0$, is called the CSS image of a planar curve Γ [18].

In [18], [19] different properties of CSS image are discussed. It is shown that renormalized CSS image is more robust against curve noise. In renormalized CSS image the evolved curve Γ_σ is reparametrized by its normalized arc length parameter ω as follows:

$$\omega = \phi_\sigma(u) = \frac{\int_0^u |R_v(v,\sigma)| dv}{\int_0^1 |R_v(v,\sigma)| dv} \tag{7}$$

$$\hat{X}(\omega,\sigma) = X(\phi_\sigma^{-1}(\omega),\sigma) \tag{8}$$

$$\hat{Y}(\omega,\sigma) = Y(\phi_\sigma^{-1}(\omega),\sigma) \tag{9}$$

Fig. 1 shows the renormalized CSS image for letter H at different scales. The renormalized CSS representation is robust with respect to scale, noise and change in orientation. The rotation of the curve usually causes a circular shift on its representation. Due to arc length normalization, scaling does not change the representation, and noise may create some small contours in the CSS image, but the main contours and therefore the corresponding maximums remain unaffected.
CSS image retains the local properties of the shape. Every contour of the CSS image corresponds to a concavity or a convexity of the shape. A local deformation of the shape mainly causes a change in the corresponding contour of the CSS image.

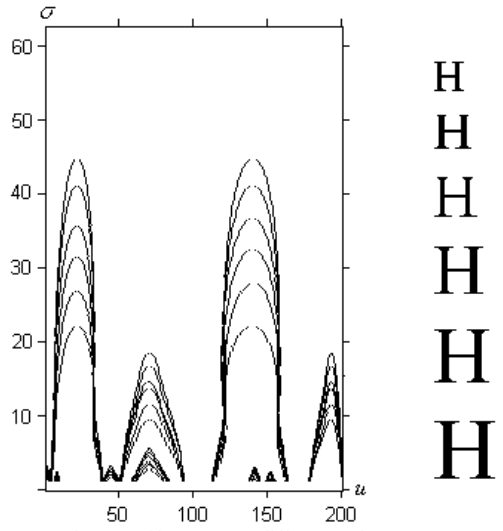

Fig. 1. CSS images of letter "H" at different scales

B. The proposed algorithm

We assumed that the bilingual scripts may have Farsi and English words and characters together; therefore we proposed an algorithm to recognize scripts in the connected components level. The output of recognition is then generalized to word identification. For the scripts that a complete line or page contains characters of one language, the algorithm can be extended to identify scripts in line or page levels by checking the number of recognized Farsi and Latin word in the line or the number of the recognized Farsi and Latin lines in the page. Therefore the proposed algorithm can be used in connected component, word, line and page levels. The proposed algorithm has the following steps:

- Apply required preprocessing to extract lines, words and connected components.
- Represent each connected component as planar curve and extract required features.
- Classify the features and identify script language in connected component level
- Recognize script language in higher levels

- *Preprocessing*

The first stage of the proposed algorithm is the identification based on connected components. To identify connected components we first smooth the input image and apply threshold to convert it to binary image. We used Otsu method [20] to calculate threshold. We then extract connected components and remove small components such as dots and commas which are common in both Farsi and Latin language. To obtain the threshold for removing small connected components, we calculate the number of pixels for largest component in the text and divide it by 13 to calculate the required threshold. The number was obtained experimentally. To recognize script in higher levels we need to extract word and line in the text. To extract lines in the input text, the image profile in horizontal directions are calculated. When the lines are extracted the vertical profile of each line is used to extract words.

- *Feature extraction*

To obtain proper features for the classification of connected components, we first extract the outer contour of connected components and represent them as planar curve $\Gamma = r(u) = (x(u), y(u))$. To extract outer contours, we fill holes in the connected components and find pixels in the boundary of connected components. To normalize the arc length, the extracted curves are resampled and represented by 200 equally distant samples.

We used renormalized CSS representation to extract features from curves. As it is shown in Fig. 1, CSS image contains several contours with different local maximums. We used local maximums of CSS contours to extract proper features for script identification. CSS contours with local maximum of less amplitude are related to noise or small ripples of the curve. In order to avoid noisy feature we remove CSS contours with maximum amplitude value of less than 0.2 of the largest maximum value in CSS image.

Each local maximum in CSS image is specified with amplitude value σ and the renormalized curve index ω. Although the general shape of CSS image doesn't change with curve scale and rotation, the rotation and change in the starting point of a curve causes a circular shift in ω and the CSS representation. The change in curve scale also scales the amplitude of CSS representation. To cope with this problem we used the difference of ω and the ratio of σ values for two successive local maximums as features. We used 5 largest local maximums of CSS image to extract 10 CSS based features including 5 ω differences and 5 σ ratios. It is important to note that considering the circular shift property of CSS image, the fifth local maximum is compared to the first local maximum.

Removing CSS curve with less amplitude, may result in CSS images with less than five local maximums, in this case we fill the empty elements with a constant number larger than $\omega_{\max}$ and $\sigma_{\max}$.

For CSS image with only one local maximum, it is also impossible to calculate the difference and ratio. In this situation, we find the ω values for $\sigma = 0.5\sigma_{\max}$ and the difference of the obtained ω values is used as feature. The σ ratio for this case is assumed to be one.

Curvature scale space representation produce no local maximum for sphere like connected components such as "o" letter in Latin and "ه" letter in Farsi, in this case the feature vectors are similar for both letters. To handle this problem we added five other shape based and structural features to our feature vector as follows:

- The ratio of vertical profile maximum after filling holes to the height of connected component (VPMXHR).
- The ratio of vertical profile mean after filling its holes to the height of connected component (VPMNHR).
- Height to width ratio of connected component (HWR).
- The number of holes in connected component (NH).
- The number of local maximums extracted from CSS image (NLM).

Table I shows the summary of the proposed feature vector for the recognition of connected component language.

TABLE I
THE SUMMARY OF THE PROPOSED FEATURE VECTOR FOR SCRIPT IDENTIFICATION

Feature Title	ω Differences	σ Ratios	VPMXHR	VPMNHR	HWR	NH	NLM
Number of Elements	5	5	1	1	1	1	1

- *Classification*

We used improved KNN[1] classifier for the language identification in connected component level. In KNN classifier the distance between input feature vector and feature vectors in database are calculated. Then K nearest feature vectors are selected from the database. Finally the output class is determined using a voting algorithm. The calculation of distance is the most important part of the KNN algorithm and affects the accuracy of algorithm. Our experimental results showed that the Euclidian distance is not a proper distance measurement method for the proposed feature vector. This is mainly because of difference in the type and the number range of feature elements in the feature vector. To handle this problem we used a new algorithm to calculate distance for two feature vector as follows:

- Build a vector with the same size of the proposed feature vector which is called distance vector.
- Calculate the difference between the first five elements of two feature vectors. The first five elements in the distance vector are set to zero if the calculated difference is less than 6, otherwise they are set to one. The value of 6 has been obtained experimentally.
- Calculate the difference between the second five elements of two feature vectors. The second five elements in the distance vector are set to zero if the calculated difference is less than 0.5, otherwise they are set to one. The value of 0.5 has been obtained experimentally.
- The remaining elements of the distance vector are filled with squared difference of remaining elements in two vectors.
- The distance between two feature vectors is defined as the sum of distance vector elements.

C. Script identification at higher levels

In addition to script identifications in connected components level, it is possible to recognize script language in higher levels like word, line or page levels where the identification in word level is more important. In script identification in line and page levels it is assumed whole words in line and page belong to one language. The proposed algorithm for script identification in higher levels has the following steps:

1. The lines and words in the input image are extracted using image profile in horizontal and vertical directions.
2. The number of connected components forming the word is calculated. If the majority of components belong to Latin language, the word is Latin; otherwise it is a Farsi word. If the number of Farsi and Latin components is the same, the word will be recognized as Farsi word experimentally.
3. The words of each line are extracted.
4. If the majority of words in a line belong to Farsi language, the line is considered as Farsi, otherwise it is Latin.
5. Similarly according to the number of Farsi and Latin lines in a page, identification in page level is carried out.

III. RECOGNITION OF FARSI HANDWRITTEN DIGITS

A. Block scheme of the proposed algorithm

Fig. 2 shows the block scheme of the proposed algorithm for training and test stages. The algorithm starts by applying required preprocessing to input image. In preprocessing stage the input image is first converted to binary image and using morphology based algorithms noisy points are removed. We then extract the bounding box and center of mass for the handwritten digit and normalize it to a constant size of 40 × 40. Feature extraction is the next stage of the proposed algorithm. We used two different algorithms for feature extraction including PCA and PCA-LDA algorithms. The extracted features are then used for training and testing the classifier algorithm. We utilized cluster based weighted (fuzzy) SVM for the classification and recognition of handwritten Farsi digits.

B. Feature extraction

We used two different methods for feature extraction including PCA and PCA-LDA algorithms which are explained in the following sections.

- *Feature extraction using PCA method*

In this method, all the images in the database are represented as one dimensional vectors X_i after applying required preprocessing. Then covariance matrix (C) is calculated for all training images in database of handwritten Farsi digits using the following equations [21]:

$$\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N} \tag{10}$$

$$\hat{X}_i = X_i - \overline{X} \tag{11}$$

$$W = [\hat{X}_1, \hat{X}_2, ..., \hat{X}_N] \tag{12}$$

$$C = \frac{1}{N}\sum_{i=1}^{N} \hat{X}_i \hat{X}_i^T = \frac{1}{N} WW^T \tag{13}$$

where $\overline{X}$ and C are mean vectors and covariance matrix for digit images in the database and N is the total number of database images.

[1] K nearest neighbor

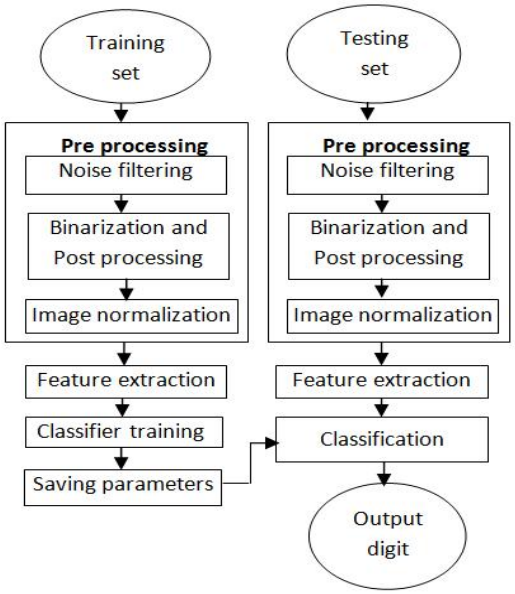

Fig. 2. Block scheme for the proposed digit recognition algorithm

Then the principle component analysis (PCA) is applied to the covariance matrix C and M largest principle components are used for feature extraction as follows:

$$Y_i = (X_i - \overline{X})^T D \tag{14}$$

where Y_i are the extracted features and D is the matrix of M principle vectors.

- *Feature extraction using PCA-LDA method*

The feature extraction using PCA-LDA algorithm is similar to PCA method. In this method, the covariance matrix is calculated like PCA algorithm; However M_2 largest principle components are used for feature extraction where M_2>M. We then reduce the size of features using the LDA algorithm. The algorithm for size reduction and the extraction of features using the LDA algorithm has the following steps:

- The average of vectors for each digit is calculated using the output of PCA algorithm as follow:

$$\overline{Y}^k = \frac{\sum_{i=1}^{N_k} Y_i^k}{N_k} \qquad \text{k=1,2,…,10} \tag{15}$$

where N_k is the number of images for digit k in the database, Y_i^k are the output of PCA algorithm for class k and $\overline{Y}^k$ are the mean vector value for class (digit) k.

- The scattering matrix for each class (digit) is calculated using the following equations:

$$\hat{Y}_i^k = Y_i^k - \overline{Y}^k \qquad \text{k=1,2,…,10} \tag{16}$$

$$S^k = \sum_{i=1}^{N_k} (\hat{Y}_i^k \cdot \hat{Y}_i^{kT}) \qquad \text{k=1,2, ...,10} \tag{17}$$

$$S_W = \sum_{k=1}^{10} S^k \tag{18}$$

where S^k are scattering matrix for each class.

- The intra class scattering matrix S_B and matrix A are calculated:

$$S_B = \frac{N}{10} \sum_{k=1}^{10} (\overline{Y}^k - \overline{X})(\overline{Y}^k - \overline{X})^T \tag{19}$$

$$A = S_w^{-1} S_B \tag{20}$$

- Eigen value and eigen vectors for matrix A are calculated and M largest eigen values are used to map Y_i vectors to obtain feature vectors [22].

C. Classification algorithm

SVM algorithm classifies the input data by mapping the input samples to higher dimensions in which the separation between classes is performed more efficiently according to cover's theorem [3].

Binary SVM classifier is a classifier with two classes ω_1, ω_2 The SVM classifier in this case creates a hyper plane with the following equation to separate these two classes [22]:

$$G(\mathbf{x}) = \omega^T \mathbf{x} + \omega_0 \tag{21}$$

The classification for the input vector x_i is carried out using the following equation:

$$\begin{cases} \omega^T \mathbf{x}_i + \omega_0 > 0 \Rightarrow \mathbf{x}_i \in \omega_1 \\ \omega^T \mathbf{x}_i + \omega_0 < 0 \Rightarrow \mathbf{x}_i \in \omega_2 \end{cases} \tag{22}$$

In this article, we utilized weighted or fuzzy SVM algorithm based on K-means clustering. The advantage of this algorithm over SVM classifier is that in the fuzzy mode, the input data is not exclusively assigned to a specified class and its membership function defines its dependency to different classes.

SVM is a binary classifier. In order to use it for the classification of several classes there are two approaches, one against all and one against one [22]

- *One against all*

In this method, the K binary SVM classifiers are used for classification where K is the number of classes and the separator line for each class separate it from all other classes. Fig. 3 illustrates one against all method. For classification in this method, each sample is given to all SVM classifiers and the winner class is selected.

- *One against one*

In this method, H different binary SVM classifiers are used and each SVM separates class i from class j where i and j are different class indexes [22]. Fig. 4 shows one against one method.

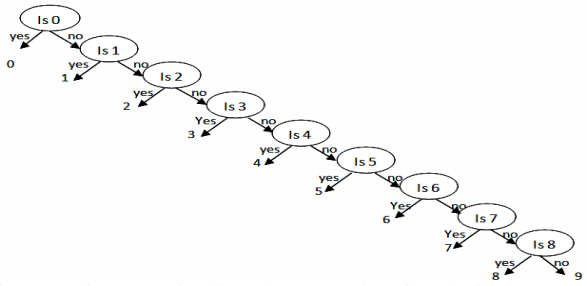

Fig 3. Ten-class SVM classifier using one against all method

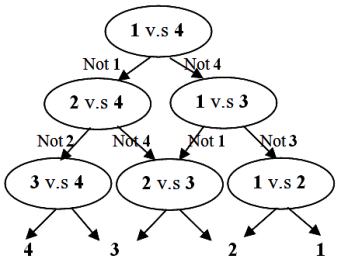

Fig. 4. 4-class SVM classifier using one against one method

- *Training Process*

Block diagram for training process is given in Fig. 5. For training purpose we first cluster training data to N clusters using k-means clustering algorithm. Then the statistical properties of each cluster including mean vector and covariance matrix are calculated and hyper plane weights.

Then SVM training process is carried out in each cluster individually by using the data of different classes in that cluster. We used weighted SVM algorithm with quadratic kernel function and one against one method for classification in each cluster. Then the average distance between the data of one class in each cluster and the calculated hyper planes of that class is calculated. The output of training process includes parameters such as mean and covariance of feature vectors in each cluster, the average distance of different classes in the clusters to different hyper planes and the weights of hyper planes. These parameters are used in the test process.

- *Classification*

Classification or test process also uses one against one structure. Fig. 6 shows the block scheme for the classification algorithm. For classification, the cluster weights for input data are calculated first. Cluster weights are Mahalanobis distance between input data and N different clusters which are calculated using the following equation:

$$w_n = \frac{1}{\sqrt{2\pi}\left|\Sigma\right|^{\frac{1}{2}}} \exp(-(\mathbf{x}_t - \overline{\mathbf{x}}_n)^T \Sigma_n^{-1} (\mathbf{x}_t - \overline{\mathbf{x}}_n)) \qquad (23)$$

where $\sum_n$ and $\overline{\mathbf{x}}_n$ are covariance matrix and mean vector for nth cluster respectively and w_n is the weight for cluster n.

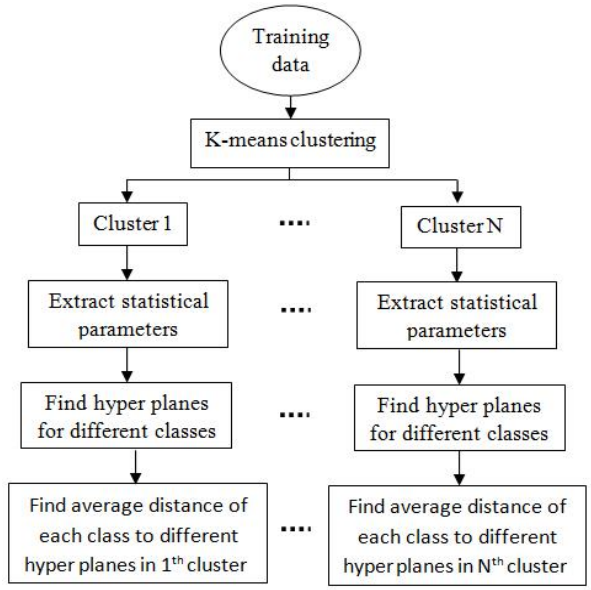

Fig. 5. Block diagram of training process using cluster based weighted SVM for the proposed digit recognition algorithm

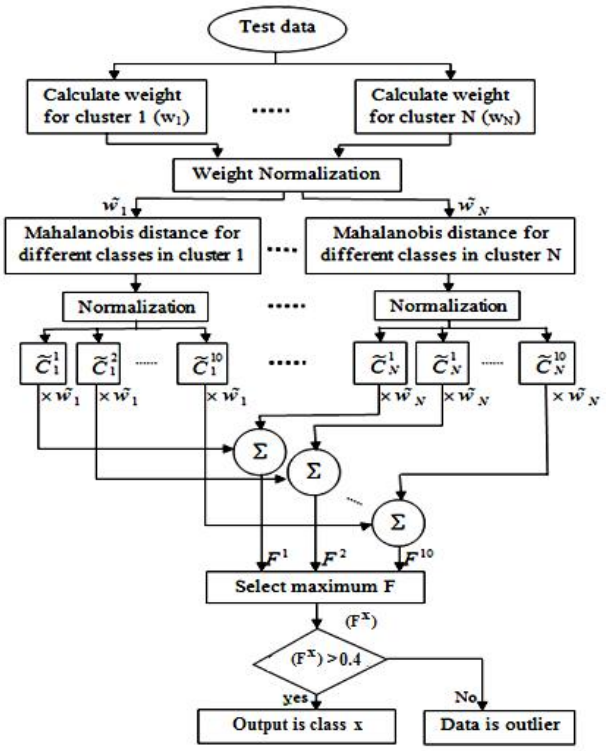

Fig. 6. Block diagram of test process for the proposed digit recognition algorithm.

We then normalize w_n weights to a value between 0 and 1 as follow:

$$\tilde{w}_n = \frac{w_n}{\sum_{k=1}^{N} w_k} \qquad (24)$$

where $\tilde{w}_n$ is the normalized cluster weight for cluster n. We then calculate the Mahalanobis distance between input data and 10 different classes in each cluster. To calculate Mahalanobis distance between input data and class m in a cluster, the Mahalanobis distance between input data and all hyper planes related to class m are calculated in that cluster.

Then sum of all calculated distance for class m is considered as Mahalanobis distance between input data and class m in the cluster n. The calculated Mahalanobis distances between input image and 10 different classes in a cluster is further normalized to a value between 0 and 1 and represented by $\tilde{C}_n^m$ where n and m are cluster and class indexes respectively.

We then calculate membership function of input image for each class using the following equation:

$$F^m = \sum_{n=1}^{N} \tilde{w}_n \tilde{C}_n^m \qquad m=1, 2, \ldots, 10 \qquad (25)$$

where F^m is the membership function of input data for class m. Among 10 resultant membership functions, the maximum value is selected and compared to a threshold value obtained experimentally. If it is less than the threshold value, it is known as outlier data. Otherwise the related class is known as the winner class.

IV. EXPERIMENTAL RESULTS

A. Results of script identification

We implemented the proposed algorithm for script identification using a MATLAB program and tested with the provided data set. Our data set contains 99 pages including 62 Farsi and 37 Latin pages. This data set contains documents with resolution of 300dpi that are randomly provided from Internet. Each page has an average of 28 lines. For both Farsi and Latin scripts, three fonts in different sizes and three different types of pens including natural, bold and italic pen are used.

In Latin scripts each letter is a connected component while because of the cursive nature of Farsi scripts, connected components include letters, words and sub-words. Additionally letters may appear differently at the start, end and middle of the word in Farsi scripts. Therefore we need more scripts for the classification of Farsi scripts in database.

Table II shows the results of script identification in connected component level. Table III and IV show the identification results in word and line levels respectively. We obtained the recognition rate of 100% for script identification in page level.

TABLE II
THE RESULTS OF SCRIPT IDENTIFICATION IN CONNECTED COMPONENT LEVEL

	Farsi	Latin	accuracy
Farsi	16129	73	99.55%
Latin	307	23698	98.72%

TABLE III
THE RESULTS OF SCRIPT IDENTIFICATION RESULTS IN WORD LEVEL

	Farsi	Latin	accuracy
Farsi	6046	12	99.8%
Latin	24	4285	99.44%

TABLE IV
IDENTIFICATION RESULTS IN LINE LEVEL

	Farsi	Latin	accuracy
Farsi	441	1	97.52%
Latin	0	457	100%

To show the efficiency of the proposed algorithm, the results of the proposed method has been compared to the results of the method presented in [9]. This method uses a 16 channel Gabor filter in different frequencies and channels applied to fixed size 128×128 input image. The input image is formed by repeating the connected component image. In each channel, two features are extracted from the image which results in a 32 dimensional feature vector. We tested the mentioned method using our database. Table V shows the results of script identification using this method in connected component level on our dataset. Comparing the results of table II with table V illustrates that the results of the proposed method is more satisfactory.

TABLE V
IDENTIFICATIONS RESULTS IN CONNECTED COMPONENT LEVEL FOR METHOD PRESENTED IN [9]

	Farsi	Latin	accuracy
Farsi	5872	186	96.93%
Latin	1004	3305	76.71%

- *Testing algorithm with fonts not presented in training dataset*

To show the efficiency of proposed features for script identification, we tested the proposed algorithm on documents with different fonts that were not presented in the training set. The test documents contain 5 Farsi documents with 5 different fonts, and 5 Latin documents with 5 different fonts. The results of this test in connected component and word levels presented in tables VI and VII respectively. The results of tables VI and VII show good accuracy especially in word level.

TABLE VI
IDENTIFICATION RESULTS IN CONNECTED COMPONENT LEVEL WITH FONTS NOT PRESENTED IN TRAINING DATASET

	Farsi	Latin	accuracy
Farsi	3710	426	89.7%
Latin	226	7925	97.23%

TABLE VII
IDENTIFICATION RESULTS IN WORD LEVEL WITH FONTS NOT PRESENTED IN TRAINING DATASET

	Farsi	Latin	accuracy
Farsi	1797	20	98.9%
Latin	17	1557	98.92%

- *Error analysis*

We analyzed the source of errors for the proposed algorithm. The main error originated from some symbols that are common in two languages such as bracket, parenthesis, comma and so forth. This error mostly occurs for test images which have fonts with different sizes in a page. This makes some of small symbols not to be removed in preprocessing step and results in some errors. Another source of error that occurs mostly in Latin scripts is related

to two adjacent connected components touched to each other. This contact causes the change in features extracted from CSS image as well as other features such as holes number and results in erroneous recognition. Unlike to Latin scripts, most of errors in Farsi scripts are related to the disconnection of connected components in some of Farsi fonts.

Most errors in line level belong to short lines that have few words.

B. Results of digit recognition

The proposed algorithm for digit recognition, implemented using a MATLAB program and tested by Farsi handwritten database [2], [3]. The database includes 8600 samples for ten Farsi handwritten digits, 860 samples for each digit. Fig. 7 shows samples of Farsi handwritten digits in the database.

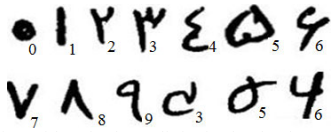

Fig. 7. Samples of Farsi handwritten digits in the database

We used 660 samples of each digit for training and remaining 200 samples for test. We repeated the test and train process several times and calculate the average of results as final results.

To compare the results of proposed algorithm with those of other algorithms, we implemented two methods presented in references [2] and [3] as well. In [2] the combination three Perceptron neural networks with different number of neurons in middle layer are used for digit recognition. The paper use PCA features for recognition. Table VIII shows the results of this algorithm for different number of PCA features and different number of neurons in middle layer. The maximum recognition rate for this algorithm is 80.1%.

The algorithm in reference [3] utilizes 10 two-class MLP classifiers for digit recognition. Table IX shows the results of this algorithm for different number of PCA features and different number of neurons in middle layer. The maximum recognition rate for this algorithm is 95.4%.

Table X illustrates the results of proposed method for different number of features. As it is shown in this table, the result of the proposed algorithm is better than two previous methods.

TABLE VIII
DIGIT RECOGNITION RATES FOR DIFFERENT NUMBER OF PCA FEATURES AND DIFFERENT NUMBER OF NEURONS IN MIDDLE LAYERS FOR ALGORITHM OF REFERENCE [2].

PCA Features	Number of neurons in middle layer of MLP			
	400	600	700	800
10	60	73.1	75.2	72.5
20	55.4	74.7	80.1	77.1
30	65.2	77.2	77	78.2

TABLE IX
DIGIT RECOGNITION RATES FOR DIFFERENT NUMBER OF PCA FEATURES AND DIFFERENT NUMBER OF NEURONS IN MIDDLE LAYERS FOR ALGORITHM OF REFERENCE [3].

PCA Features	Number of neurons in middle layer of MLP					
	10	15	20	25	30	35
10	91.3	92.5	93.2	93.5	93.9	94
20	92.2	93.4	94.4	95	95.4	95.3
30	92.2	93.4	93.3	93.6	94.5	94.1

TABLE X
DIGIT RECOGNITION RATES FOR DIFFERENT NUMBER OF FEATURES USING THE PROPOSED ALGORITHM

Feature Extraction	Number of features				
	10	20	30	40	50
PCA	91.3	93.4	95.1	94.6	94.2
PCA-LDA	93.1	94.2	96.5	95.3	93.7

Some parts of errors of the proposed algorithms are related to images taken with low resolution.

To show the efficiency of proposed algorithm in the presence of digit rotation, we tested the proposed algorithm using images with rotation. For this purpose we rotated 30 percent of test images with different angles. Table XI shows the results of proposed algorithm and the algorithm of reference [3] for test data with different rotations which are applied randomly clockwise or counter clockwise to test data.

As it is shown in table XI, the proposed algorithm is more robust to digit rotation and PCA-LDA feature extraction algorithm exhibits more stability.

TABLE XI
DIGIT RECOGNITION RATE FOR TEST DATA WITH DIFFERENT ROTATION.

Algorithm	Rotation angle		
	15	30	45
Algorithm of reference [3]	88.7	85.2	84.6
Proposed algorithm with PCA features	94.2	93.3	92.6
Proposed algorithm with PCA-LDA features	95.6	94.8	93.2

V. CONCLUSION

In this paper we introduced new algorithms for the identification of Farsi and Latin documents as well as Farsi handwritten digits recognition. Our language identification algorithm is based on features extracted from curvature scale space representation of connected components in the scripts. The proposed algorithm is able to identify script language in four levels including connected component, word, line and page levels. The proposed features and algorithm are robust against font and scale change. For Farsi handwritten digits recognition, we used PCA and PCA-LDA algorithm for feature extraction which PCA-LDA algorithm is more robust against image rotation. We used a new method called cluster based weighted SVM for classification. We tested the proposed algorithms with different data and compared the results with those of other methods. The experimental results showed the efficiency of the proposed algorithms.

REFERENCES

[1] A. Webb. *Statistical pattern recognition*, 2nd ED., John Wiley & Sons, 2003.

[2] H. N. Karizi, R. Ebrahimpour and E. kabir, "Combination of classifiers for recognition of Farsi handwritten digits," *in Proc. 3rd Iranian Conference on Machine Vision and Image (MVIP2004)*, pp. 115-119, 2004. [Published in Farsi]

[3] M. Nahvi, M. Rafeei, R. Ebrahimpour and E. Kabir, "Combination of two-class classifiers for the recognition of Farsi handwritten digits," *in Proc 16th Iranian Conference on Electrical Engineering ICEE2008*, pp. 203-207, 2008. [Published in Farsi]

[4] B. S. Manjunath, and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837-842, August 1996.

[5] H. Al-Usefi and S. Udpa, "Recognition of Arabic characters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 853-857, August 1992.

[6] M. Khosravi , and E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on their varieties," *Pattern Recognition. Letter*, vol. 28, no. 10, pp. 1133–1141, February 2007.

[7] P. Bahri, "Farsi handwritten character recognition using Neuro-fuzzy algorithms", MSc. Thesis, Iran University of Science & Technology, Tehran, 1998.

[8] Andrew Busch, Wageeh W. Boles and Sridha Sridharan, "Texture for Script Identification", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp.1720-1732, Nov. 2005.

[9] Huanfeng Ma and David Doermann, "Word Level Script Identification for Scanned Document Images", *in proc. of Int. Conf. on Document Recognition and Retrieva(SPIE)l*, pp.178-191, 2004.

[10] D Dhanya, A G Ramakrishnan and Peeta Basa Pati, "Script identification in printed bilingual documents", *Sadhana* vol. 27, Part 1, pp. 73–82, February 2002.

[11] Santanu Chaudhury and Rabindra Sheth, "Trainable Script Identification Strategies for Indian Languages", *in Proc. of the ICDAR Fifth Int. Conf. on Document Analysis and Recognition (ICDAR'99)*, pp.657 - 660, 1999.

[12] I. Moalla, A. M. Alimi and A. Benhamadou, "Extraction of Arabic Words from Multilingual Documents", *in Proc. Of Artificial Intelligence and Soft Computing Conference (ASC2004)*, 2004.

[13] B.V.Dhandra and Mallikarjun Hangarge, "On Separation of English Numerals from Multilingual Document Images", *Journal of Multimedia*, vol. 2, no. 6, November 2007.

[14] A.L. Spitz, "Determination of the Script and Language Content of Document Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 235-245, Mar. 1997.

[15] Shijian Lu, Chew Lim Tan, and Weihua Huang, "Language Identification in Degraded and Distorted Document Images", *in proc. Of 21st National Conference on Artificial Intelligence*, pp. 769-774, 2006.

[16] Malike Khoddami and Alireza Behrad, "Farsi and Latin Script Identification using Curvature Scale Space Features" " *in proc. 10th Symposium on Neural Network Applications in Electrical Engineering, (Neurel 2010),* 2010.

[17] Mehdi Salehpour and Alireza Behrad, "Cluster Based Weighted SVM for the Recognition of Farsi Handwritten Digits" *in proc. 10th Symposium on Neural Network Applications in Electrical Engineering, (Neurel 2010),*2010.

[18] 16 F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature based shape representation for planar curves", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 789-805, Aug. 1992.

[19] F. Mokhtarian, "Convergence properties of curvature scale space representations", *Proc. of the 1995 British conf. on Machine vision,* vol. 1, pp. 357 – 366, 1995.

[20] N. Otsu, "A Threshold Selection Method from Gray-Level Histogram", *IEEE Trans. on Systems*, *Man and Cybernetics*, vol. 9, pp. 62-66, 1976.

[21] M. Turk, and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, March 1991.

[22] S. Abe, *Support vector machines for pattern recognition*, Springer Verlog London limited, 2005.