

Holistic Farsi handwritten word recognition using gradient features

Z. Imani^{1*}, A.R. Ahmadyfard¹ and A. Zohrevand²

1. Electrical Engineering Department, University of Shahrood, Shahrood, Iran.

2. Computer Engineering & Information Technology Department, University of Shahrood, Shahrood, Iran.

Received 16 July 2015; Accepted 2 August 2015

*Corresponding author: z.imani13@gmail.com (Z Imani).

Abstract

In this paper we address the issue of recognizing Farsi handwritten words. Two types of gradient features are extracted from a sliding vertical stripe which sweeps across a word image. These are directional and intensity gradient features. The feature vector extracted from each stripe is then coded using the Self Organizing Map (SOM). In this method each word is modeled using the discrete Hidden Markov Model (HMM). To evaluate the performance of the proposed method, FARSA dataset has been used. The experimental results show that the proposed system, applying directional gradient features, has achieved the recognition rate of 69.07% and outperformed all other existing methods.

Keywords: *Handwritten Word Recognition, Directional Gradient Feature, Intensity Gradient Feature, Hidden Markov Model, Self-organizing Feature Map, FARSA Database.*

1. Introduction

Due to being fraught with such difficulties as high variability of handwritten words' style and shape, uncertainty of human-writing, skew or slant writing, segmentation of words into characters, and the size of lexicon, cursive handwritten word recognition has become a challenging area in pattern recognition [1]. Farsi handwritten recognition, which this paper addresses, is very similar to Arabic in terms of strokes and structure. The only difference is that Farsi has four more characters. Therefore, Farsi word recognition system can also be used for Arabic words [2].

Handwritten word recognition systems may work online or offline. In the online system, words are written using special tools like pen and tablet. In this way available information such as writing direction and writing speed facilitates the process of recognition. In the online method, handwritten recognition is occurs concurrently with writing [3,4,5], while in off-line method handwritten words exist prior to the recognition. The data were first collected using a pen and a paper and then scanned. Since in the offline method, additional tools for collecting data are not used, the recognition is more complex. In this paper we address the problem of recognizing Farsi

handwritten words in an offline manner. In recent years, several studies have been conducted in this area, as reported below.

In [6] a holistic system for recognition of Farsi/Arabic handwritten words using the right-left discrete Hidden Markov Models (HMM) was proposed. The histogram of chain-code directions for vertical stripes on word image was used as feature vector [6]. The extracted feature vectors were used as the input data to the Kohonen self-organization vector quantization. A database including 17000 images of 198 words was used for evaluating this system. Finally by smoothing the probabilities of observation symbols, recognition rate of 65% was achieved [6].

In [7] Dehghan et al generated a fuzzy codebook using the same feature vectors. The recognition rate of 67% was reported for this system, using an HMM classifier [7].

In [2], a method for recognizing handwritten Iranian cities' name was proposed. In this method K-means clustering was used for vector quantization and words were modeled using the discrete HMM. For this method a recognition rate of 80.75% was reported [2]. In [8] the authors proposed an off-line Arabic/Farsi handwritten

recognition algorithm using RBF network. The features utilized in this method were wavelet coefficients being extracted from the profiles of smoothed word image in four standard directions. A database including 3300 images of 30 common Farsi names was used to evaluate this system. This method achieved 96% recognition rate.

AlKhateeb et al [9] dealt with the problem of offline Arabic word recognition. They used a set of intensity features to train an HMM classifier for each word. The results were re-ranked using structure-like features (including a number of sub-words and diacritical marks) to improve recognition. This method achieved 89% recognition rate using re-ranking.

Imani et al [10] used chain-code and distribution of stoke pixels across a sliding frame as a hybrid method for feature extraction stage, then they used an HMM for classification. This method was applied on FARSA database [11] and a recognition rate of 68.88% was achieved.

In this paper we addressed the off-line Farsi handwritten words recognition problem. To do so, we extracted gradient features from a sliding window, which sweeps across the word image. In this paper, we proposed two recognition systems. In one of the systems, the magnitude of image gradient was used and in the other one, the direction of image gradient was utilized in order for representing the word image. In either of the two systems, the feature vector, being extracted from a sliding window, was quantized through using a vector quantization algorithm. Then, for each word class some sequences of code indexes, (that were taken from its sample image), from the sample images of that class were used to train its discrete Hidden Markov Model. As it was shown in the experimental results, directional gradient features outperform intensity gradient features in terms of recognition rate. In fact, the intensity gradient features are performed in typewritten recognition in [12]. However using features based on the direction of script gradient improves the recognition rate. FARSA is an appropriate dataset of handwritten Farsi words which was introduced by Imani et al [11]. We evaluate the proposed methods in this paper using FARSA database.

The rest of the paper is organized as follows. In the next section, the proposed word recognition system will be introduced. In section 3, the employed classification features are explained. The vector quantization for feature coding vector is presented in section 4. In section 5, the system's classification method will be described. The experimental results are reported in section 6. Finally, the paper will be drawn to a conclusion in section 7.

2. Proposed method

The word recognition system contains three stages: preprocessing, feature extraction and classification. The block diagram of the proposed Farsi handwritten word recognition is demonstrated in figure 1.

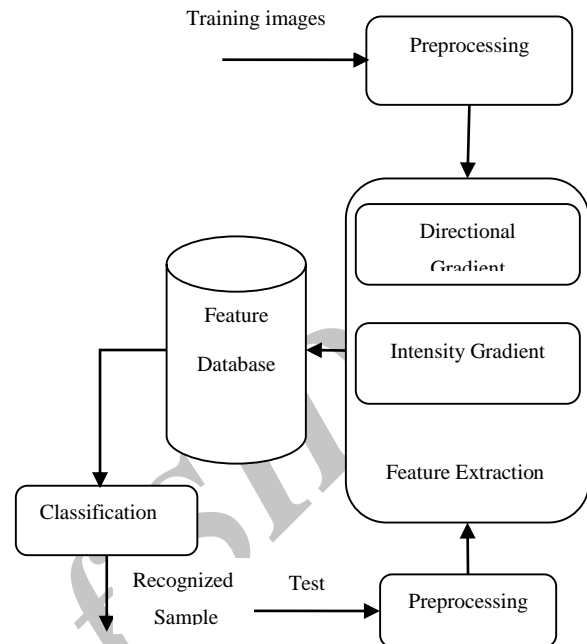


Figure1. Block diagram of the proposed system.

2.1. Preprocessing

The pre-processing stage plays a very crucial role in word recognition in order to represent various samples of a word in an invariant manner [3]. The stage consists of the following steps:

- Binarization and noise removal: The gray level image of a word is binarized by thresholding. Then those connected components with an area less than the predefined threshold are deleted.
- Cropping: In order to increase the processing speed and to decrease memory usage, the binary image is cropped by its bonding box. Then, the image is resized by 45 pixels in height.
- Skeletonization and dilation: First, the word image skeleton is extracted (Figure 2.b). Then, it is dilated (Figure 2.c) by a 4×4 square structure element. This step aims at making the system independent of stroke width [3].

3. Extracting feature from word image

In order to represent the word image as a pattern to the recognition system we need to represent it using discriminative features. We use a number of white pixels, horizontal and vertical edges, together with the histogram of gradient directions,

within a sliding window, to describe the word image.

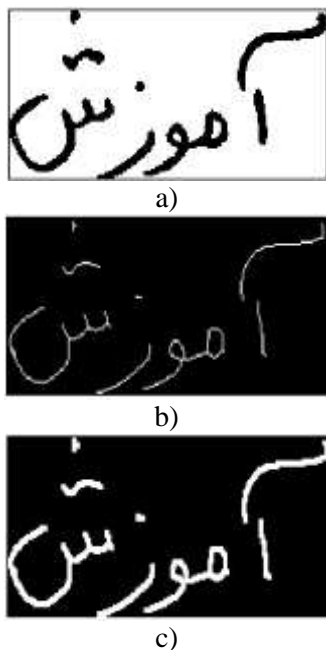


Figure2. An example of skeletonization and dilation. a) Original image. b) Word skeleton. c) Dilation on word skeleton.

3.1 Sweep word image using sliding stripe

A sliding stripe is used to scan word image from right to left. Accordingly each word image is divided into several narrow windows with an overlap of 50%. Using this strategy, word image can be represented as a sequence of script primitives [10]. Each window is twice as wide as each stroke in the word image as suggested in [6]. Thus, in this study we widen the window to 8 pixels and heighten it to the word image height. A set of simple features are extracted from pixels falling within that window. In order to extract proper features from each window of the word image, the window is divided horizontally into a number of zones. In this paper two types of gradient features are extracted from each window zone.

3.2. Gradient feature extraction

By applying a low pass Gaussian filter, we convert the binary image of Farsi script into a gray scale image. Here we made use of horizontal and vertical Sobel operators to determine the gradient of word image in x and y directions namely G_x and G_y . These two Sobel operators are shown in figure 3. We also calculated the gradient's direction:

$$\theta(x, y) = \tan^{-1}(G_y/G_x) \tag{1}$$

-1	-2	-1	-1	0	1
----	----	----	----	---	---

0	0	0	-1	0	2
1	2	1	-1	0	1

(a) (b)

Figure3. Sobel masks, (a) horizontal mask and (b) vertical mask.

3.3. Using intensity and magnitude of gradient components as a feature set

In order to describe image inside a sliding window, the window is divided into 9 horizontal cells and 3 features are extracted from each individual cell. The features are: image intensity, horizontal component, and vertical components extracted from the Sobel operators (Figure 4).

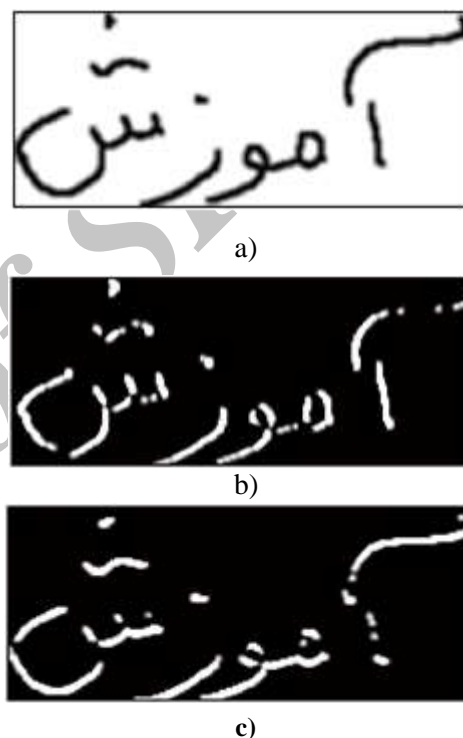


Figure4. a) Input word image. b) Vertical derivative of (a). c) Horizontal derivative of (a).

The intensity feature represents the number of white pixels within each sliding window cell. The number of horizontal and vertical edges is counted in each sliding window cell [12]. Accordingly, there are 3 features for each cell and each sliding window has 9 horizontal cells, so 27 feature values for each sliding window are extracted.

3.4. Gradient direction as feature

In (1), $\theta(x, y)$ returns the direction of gradient vector (G_x, G_y) in the range of $[-\pi, \pi]$. The gradient direction at each pixel is quantized to 8 intervals of $\pi/4$ each (Figure 5). Each sliding window, as introduced in the previous section, is divided into 5 horizontal cells whose histograms are shown in four normalized directions. Thus we represent a

sliding window on image word using a feature vector including 20 components.

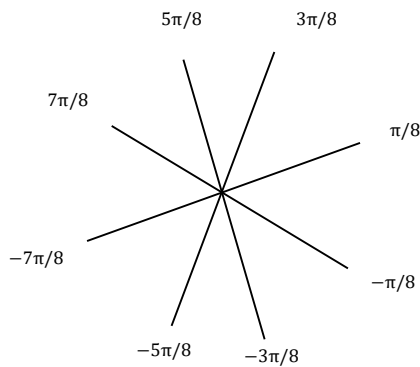


Figure 5. Quantizing the gradient direction with four symbols.

4. Vector quantization

In this research for 198 classes of database, 533000 extracted frame from the training set is used as input to the Kohonen Self Organization vector quantization (SOM in Neural Network MATLAB Toolbox) to obtain a codebook with 49 symbols. After generating the codebook a given feature vector is mapped to a symbol from 1 to 49, which is the closest code word by the Euclidean distance measure. The histogram distribution of feature vectors in codebook has been shown in figure 6. Thus, each word image is now identified by an observation sequence. The sequences are given as input to the Hidden Markov Models.

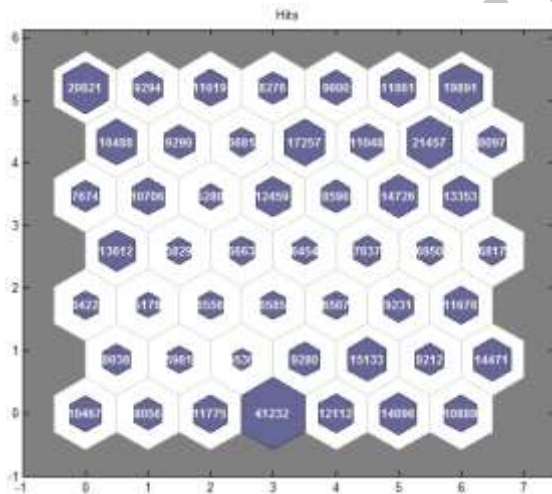


Figure6. Histogram distribution of feature vectors.

5. Hidden Markov Model classification

The Hidden Markov Models (HMMs) are widely used for text recognition. The HMMs are statistical models being originally used for speech recognition efficiency. Since the HMM has put in a good performance in speech recognition, and because of the similarities between speech recognition and cursive handwriting, it has been

extended to Farsi handwriting word recognition [3,13,14].

HMM can be represented using three parameters as follows:

$$\lambda = (\pi, A, B) \tag{2}$$

λ stands for an HMM model, π is the vector of the initial state probabilities, A is the state transition matrix and B is the matrix of observation symbol probabilities:

$$\pi = \{\pi_i | \pi_i = P(S_1 = i)\} \tag{3}$$

$$A = \{a_{ij} | a_{ij} = P(S_t = j | S_{t-1} = i)\} \tag{4}$$

$$B = \{b_j(o_k) | b_j(o_k) = P(O_t = o_k | S_t = j)\} \tag{5}$$

where, a_{ij} is the probability that system at time t is in j^{th} state assuming that system at time t-1 was in i^{th} state. In (5), $b_j(o_k)$ is the probability that we observe o_k at time t, $o_t = o_k$, assuming that system at this time is in j^{th} state. Here, N equals to the number of states, T to the length of the sequence of observations to the number of possible observations (form the training set), and $S = \{s | 1 \leq s \leq N\}$ [15].

To design an HMM classifier, several procedures are needed to be performed including (i) deciding the number of states and observations, (ii) choosing HMM topology, (iii) model training using selected samples and (iv) testing and evaluation [9].

The number of chosen states is proportioned to the word length. Indeed for each word, the number of states is considered proportional to the minimum number of frames in the word picture. In this paper, a right-to-left HMM is employed. Each state could have a self-transition, or a transition to the next or two next states. In the training stage, the model is optimized using the training data through an iterative process. The Baum-Welch algorithm, a variant of the Expectation Maximization (EM) algorithm, is utilized to maximize the observation sequence probability $p(O | \lambda)$ of the chosen model, $\lambda = (\pi, A, B)$, for optimization [9].

It is well known that if adequate training data is not provided, the HMM parameters especially the observation symbol probabilities, are usually poorly estimated. As a result, the recognition rate is degraded even if a very slight variation in the testing data occurs. A proper smoothing of the estimated observation probability can overcome this problem without a need for more training data. The parameter smoothing method as proposed in [16] is used. After training all of the HMMs by the Baum-Welch algorithm, the value

of each observation probability in each state is raised by adding a weighted-sum of the probabilities of its neighboring nodes in the SOM as follows:

$$b_j^{new}(k, l) = b_j^{old}(k, l) + \sum_{\substack{(k,l) \neq \\ (p,q)}} W_{(k,l),(p,q)} b_j^{old}(p, q) \quad (6)$$

where, the weighting coefficient $W_{(k,l),(p,q)}$ defined as the function of distance between two nodes (p, q) and (k, l) in the map:

$$W_{(k,l),(p,q)} = SF \cdot c^{(d_{(k,l),(p,q)}-1)} \quad (7)$$

(c) Is a constant chosen to be equal to 0.5. The smoothing factor (Sf) controls the degree of smoothing, and $d_{(k,l),(p,q)}$ is the hexagonal distance between two nodes with the coordinates (k, l) and (p, q) in the codebook map.

In testing stage, an observed sequence of test images is given to all of the HMM models to find the best model that can generate the data. Viterbi's algorithm is used to match a single model to the observed sequence of symbols. The reader is referred to [17] for more details about HMM.

6. Experimental results

In this section, firstly, the collected database, namely FARSA, is introduced. Then, the results of applying the proposed method to the database are reported.

6.1. FARSA database

Farsi scripts have four more characters than the Arabic ones in their character set. Therefore, the most accurate recognition result can be obtained only by using the proper dataset for the language [18]. So the standard Arabic database cannot be used for Farsi and there is no proper handwritten database available in Farsi.

A proper database must include a significant number of samples for each class of words in the dictionary. We prepare a database including 30000 images of 300 formal words which are common in Farsi Language. The handwritten words are scanned with 300-dpi resolution and 256 gray levels. The database is called FARSA [11].

6.2. Experiments and results

In this paper, 198 word classes of the FARSA were used to evaluate the proposed system. The number of word classes was chosen to compare the proposed method with the methods in [6, 10]

which had used the same number of word classes. A subset including 19800 samples of the images in the FARSA database was chosen. Out of which about 70% of samples were chosen randomly as the training database and the rest were utilized for the test.

All of the HMM models are initialized using the same parameters. The performance of the word recognition system is illustrated in table1 by a top-n recognition rate (the percentage of test words recognized as true class lies among the first n positions in the candidate list). The criterion for ranking of a HMM model is the log-likelihood of a given test image being produced by the HMM model.

Table 1 reports the results achieved through comparing two types of gradient feature extraction. As can be seen, directional gradient features are more appropriate than intensity gradient features for handwritten word recognition. The amplitude of gradient performed very well as feature for typewritten recognition in [12]. But in handwritten, due to a variety of text and change the font stretch as seen in table 1, the directional gradient features work better. As shown in table 1 the recognition rate using the angle of gradient improves the recognition rate about 16%.

As mentioned previously, due to the variation of handwritten words in a class and because of the limited number of training data, the recognition rate is not considerable. The proposed method was repeated after smoothing the observation probabilities of the HMMs. As reported in [10] the appropriate smoothing factor is 0.001. In this experiment, we report the results of smoothing the HMM parameters with proper smoothing factor in table1. In another experiment, we consider three codebook sizes. Figure 7 shows the results of an experiment in which directional gradient feature extraction method is adopted. As can be seen, the recognition rate of a codebook being 49 in size outdoes that of the one being 36 in size. But the results for 49 and 64 are almost identical. On the other hand, figure 8 shows that the computational complexity and training system time for 64 is much higher than 49. So we choose 49 as the codebook size.

On the other hand, figure 8 shows that the computational complexity and training system time for 64 is much higher than 49. So we choose 49 as the codebook size.

Table1. Comparison of recognition results with two types feature extraction.

Lexicon size = 198	Directional Gradient feature	Intensity Gradient features	Feature vector size	Top1	Top5	Top10
Without smoothing	✓		20	58.83	76.81	81.80
With smoothing SF=0.001	✓		20	69.07	88.35	92.73
Without smoothing		✓	27	47.17	68.50	75.35
With smoothing SF=0.001		✓	27	53.48	77.89	85.39

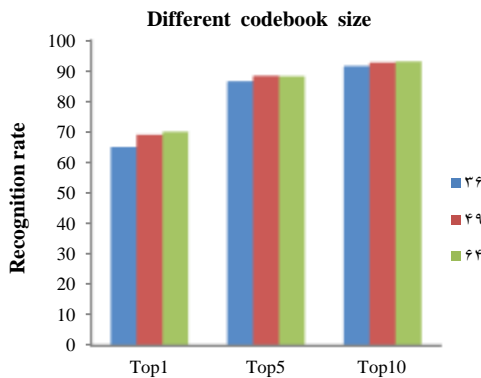


Figure7. Comparison of recognition results with different codebook size.

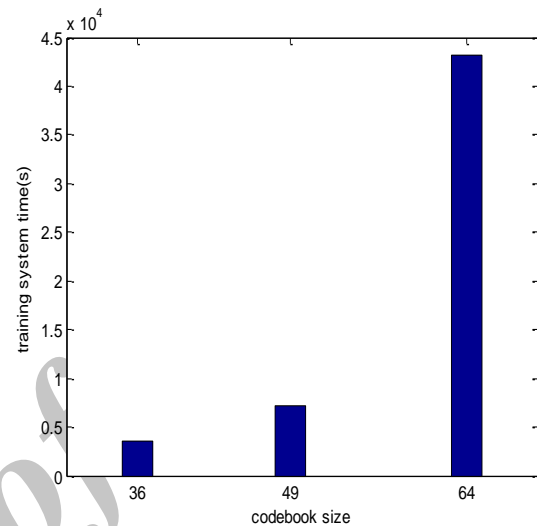


Figure8. Training system time for different codebook size.

Table 2 shows that the performance of the proposed word recognition systems compared with three existing methods [6,7,10].

In an earlier work [10], a combination of chain code histogram features and intensity features were used during the extraction stage and the feature vectors were 25 in length.

It can be seen that, despite in this work we use less features in compare with an earlier work [10], the performance has been improved.

Table2. Recognition rate of proposed method compared to other method.

Method	lexicon size	Number of training images	Number of testing images	Top-1	Top-5	Top-10
DHMM+smoothing [6]	198	10200	6800	65.05	86.08	90.83
FVQ/HMM[7]	198	10200	6800	67.18	87.55	92.06
The previous method [10]	198	13800	5900	68.88	87.54	91.75
The proposed system	198	13800	5900	69.07	88.35	92.73

The proposed system was evaluated using a newly prepared database, namely FARSA database. The experimental results for directional gradient features outperformed intensity gradient features

7. Conclusion

We proposed an offline system for recognizing of Farsi handwritten words. Directional and intensity gradient features were employed. The extracted feature vector was coded using a vector quantization algorithm. The codes were utilized as an observation in order to train the HMM model for each word class.

and the results achieved through adopting the proposed method were far better than those achieved through employing the existing methods.

References

- [1] Sagheer, M.W., He, C. L., Nobile, N. & Suen, C. Y. (2010). Holistic Urdu Handwritten Word Recognition Using Support Vector Machine. Proceedings of the 20th International Conference on Pattern Recognition, Istanbul, Turkey, 2010.
- [2] Vaseghi, B., Shahpour, A., Ahmadi, M. & Amirfattahi, R. (2008). Off-line Farsi /Arabic Handwritten Word Recognition Using Vector Quantization and Hidden Markov Model, Proceedings of the 12th IEEE International Multi topic Conference. Karachi, Pakistan, 2008.
- [3] Märgner, V. & ElAbed, H. (2012). Guide to OCR for Arabic Scripts. London: Springer-Verlag.
- [4] Ghods, V., Kabir, E. & Razzazi, F. (2013). Effect of delayed strokes on the recognition of online Farsi handwriting. Pattern Recognition Letters, vol. 34, no. 5, pp. 486–491.
- [5] Ghods, V., Kabir, E. & Razzazi, F. (2013). Decision fusion of horizontal and vertical trajectories for recognition of online Farsi subwords. Journal of Engineering Applications of Artificial Intelligence, vol. 26, no. 1, pp. 544–550.
- [6] Dehghan, M., Faez, K., Ahmadi, M. & Shridhar, M. (2001). Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM. Pattern Recognition, vol. 34, no. 5, pp. 1057–1065.
- [7] Dehghan, M., Faez, K., Ahmadi, M. & Sridhar, M. (2001). Unconstrained Farsi handwritten word recognition using fuzzy vector quantization and hidden Markov models. Pattern Recognition Letter, vol. 22, no. 2, pp. 209–214.
- [8] Bahmani, Z., Alamdar, F., Azmi, R. & Haratizadeh, S. (2010). Off-line Arabic/Farsi handwritten word recognition using RBF neural network and genetic algorithm. IEEE International Conference on Intelligent Computing and Intelligent Systems, Xiamen, China, 2010.
- [9] AlKhateeb, J. H., Ren, J., Jiang, J. & Al-Muhtaseb, H. (2011). Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking. Pattern Recognition Letters, vol. 32, no. 5, pp. 1081–1088.
- [10] Imani, Z., Ahmadyfard, A. R., Zohrevand, A. & Alipour, M. (2013). Offline Handwritten Farsi cursive text recognition using Hidden Markov Models. 8th Iranian conference on machine vision and image processing, Zanzan, Iran, 2013.
- [11] Imani, Z., Ahmadyfard A.R. & Zohrevand A. (2013). Introduction to Database FARSA: digital image of handwritten Farsi words. 11th Iranian Conference on Intelligent Systems in Persian, Tehran, Iran, 2013.
- [12] Khorshed, M. S. (2006). Mono-font cursive Arabic text recognition using speech recognition system. In Structural, Syntactic, and Statistical Pattern Recognition, Springer Berlin Heidelberg pp.755-763.
- [13] Khorshed M. S. (2002). Off-line Arabic character recognition – a review. Pattern Analysis and Application, vol. 5, no. 1, pp.31-45.
- [14] Günter, S. & Bunke, H. (2004). HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and Gaussian components. Pattern Recognition, vol. 37, no. 10, pp. 2069–2079.
- [15] Al-Muhtaseb, H. A., Mahmoud, S. A. & Qahwaji, R.S. (2008). Recognition of off-line printed Arabic text using Hidden Markov Models, Signal Processing journal, vol. 88, no. 12, pp. 2902–2912.
- [16] Zhao, Z., & Rowden, C. G. (1992). Use of Kohonen self-organising feature maps for HMM parameter smoothing in speech recognition. In IEE Proceedings F (Radar and Signal Processing), IET Digital Library, vol. 139, no. 6, pp. 385–390.
- [17] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286.
- [18] Haghighi, P. J., Nobile N., He, C. L. & Suen, C.Y. (2009). A New Large-Scale Multi-purpose Handwritten Farsi Database, Proc. International Conference on Image Analysis and Recognition, Halifax, NS, Canada, 2009.

بازشناسی کلمه دستنوشته فارسی به صورت کلی نگر با استفاده از ویژگی‌های گرادیان

زهرا ایمانی^{۱*}، علیرضا احمدی فر^۱ و عباس زهره‌وند^۲

^۱دانشکده مهندسی برق، دانشگاه صنعتی شاهرود، شاهرود، ایران.

^۲دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی شاهرود، شاهرود، ایران.

ارسال ۲۰۱۴/۰۶/۱۰؛ پذیرش ۲۰۱۵/۰۸/۰۲

چکیده:

ما در این مقاله به مسئله تشخیص کلمات دستنویس فارسی پرداخته‌ایم. دو نوع ویژگی مبتنی بر گرادیان از نوار کشویی عمودی که در سراسر تصویر کلمه حرکت می‌کند استخراج می‌شود این ویژگی‌ها گرادیان شدتی و گرادیان جهتی هستند. بردار ویژگی که از هر نوار عمودی استخراج می‌شود، سپس با استفاده از یک شبکه عصبی خود سازمان‌ده کوانتیزه می‌شود. در این روش هر کلمه با استفاده از مدل مخفی مارکوف گسسته، مدل می‌شود. برای ارزیابی کارایی روش پیشنهادی پایگاه داده فارسی مورد استفاده قرار می‌گیرد. نتایج تجربی نشان می‌دهد که روش پیشنهادی، با اعمال ویژگی‌های گرادیان جهتی، به نرخ بازشناسی ۶۹,۰۷ درصد دست می‌یابد و دارای عملکرد بهتری نسبت به تمام روش‌های موجود است.

کلمات کلیدی: بازشناسی کلمه دستنویس، ویژگی گرادیان جهتی، ویژگی گرادیان شدتی، مدل مخفی مارکوف، نگاشت ویژگی خود سازمان‌ده، پایگاه

داده فارسی