PERGAMON

Pattern Recognition 34 (2001) 1057–1065

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM

M. Dehghan[a], K. Faez[a], M. Ahmadi[b,*], M. Shridhar[c]

[a]Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran
[b]Electrical and Computer Engineering Department, University of Windsor, Windsor, Ont, Canada N9B 3P4
[c]Electrical and Computer Engineering Department, University of Michigan, Dearborn, MI 48128, USA

Received 22 December 1999; received in revised form 9 February 2000; accepted 9 February 2000

Abstract

A holistic system for the recognition of handwritten Farsi/Arabic words using right–left discrete hidden Markov models (HMM) and Kohonen self-organizing vector quantization is presented. The histogram of chain-code directions of the image strips, scanned from right to left by a sliding window, is used as feature vectors. The neighborhood information preserved in the self-organizing feature map (SOFM), is used for smoothing the observation probability distributions of trained HMMs. Experiments carried out on test samples show promising performance results. © 2001 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Handwritten word recognition; Hidden Markov model; Self-organizing feature map; Parameter smoothing; Farsi/Arabic handwriting recognition

1. Introduction

During the past decade, remarkable progress has been achieved in the field of handwritten word recognition. Many papers dealing with applications of handwritten word recognition to automatic reading of postal addresses, bank checks and forms (invoices, coupons and revenue documents, etc.) have been published [1–4]. However, most of the published work deals with the recognition of Latin and Chinese scripts. The progress in Farsi/Arabic script recognition has been slow mainly due to the special characteristics of these scripts. The reader is referred to Refs. [5,6] for more details on the state of the art of Arabic character recognition.

Generally holistic or character based methods have been used for handwritten word recognition. In the former approach, a word is treated and identified as an entity. In the second approach, a word is considered as a sequence of smaller components-like characters or graphemes. The word is identified by extracting and recognizing its constituent components. The choice of a specific approach for a particular application is strongly influenced by the size of the available vocabulary (lexicon). The holistic approach can be used if the size of the vocabulary is small (as would be the case in a typical application such as the recognition of the legal amount in cheques provided the words in the phrase are well isolated). The character-based approach is generally the preferred method for recognition applications that involve large-size vocabularies. In applications such as recognition of city and street names from the address blocks, both approaches are feasible. Although the vocabulary is almost large in this case, contextual information like ZIP codes can be used to prune the size of vocabulary.

Hidden Markov model (HMM) [7] has been successfully used for recognition of handwritten words [1–3], using both the aforementioned approaches. In HMM word recognition paradigm, there are two principal methods: the model discriminant method, and the path discriminant method [1]. In the model discriminant approach, a separate HMM is used for each word class. However, in the path discriminant approach, which is

* Corresponding author. Tel.: + 1-519-253-3000; fax: + 1-519-971-3695.

E-mail address: ahmadi@uwindsor.cn (M. Ahmadi).

PII: S0031-3203(00)00051-0

analogous to the character based method, only one HMM is used for all the word classes, and different paths in the model distinguish one word class from the others. This paper deals with recognizing handwritten city names extracted from postal address blocks. Since the vocabulary of this application is limited (198 in our case), and also the segmentation of handwritten Farsi (Arabic) words is a very crucial problem; a holistic approach based on model discriminant discrete HMM is chosen as the recognition engine. The Kohonen self-organizing feature map [8] is used for constructing the codebook and also smoothing the observation probability distribution. The method was implemented using a database consisting of more than 17,000 images of 198 city names of Iran.

2. Farsi handwriting characteristics

Since the characteristics of Farsi handwriting is different from the Latin one and some of the readers maybe unfamiliar with Farsi script, a brief description of the important aspects of Farsi script will be presented. Farsi text is inherently cursive both in handwritten and printed forms and is written horizontally from right to left. Farsi writing is very similar to Arabic in terms of strokes and structure. Therefore, a Farsi word recognizer can also be used for recognition of Arabic words. The only difference between Farsi and Arabic scripts is in the character sets. Farsi character set, shown in Fig. 1, comprises all of the 28 Arabic characters plus four additional ones (marked with the * in Fig. 1).

A Farsi character is written as a single main stroke and in most cases is completed with other complementary strokes such as dot(s), zigzag bars, etc. The complementary strokes might be placed above, below, or in the middle of the main stroke. Some Farsi characters have a unique main stroke (overall shape); however they are distinguished from each other only by the presence/absence, position or number of some secondary strokes. An example of different characters with similar main stroke is shown in Fig. 2. Ambiguous writing of these secondary strokes sometimes causes a word image to be read in many various forms with completely different meanings.

خ	ح	چ	ج

Fig. 2. An example of different Farsi characters with a unique overall shape.

In contrast to English, Farsi characters are not divided into upper and lower case categories. Instead, a Farsi character might have several shapes depending on its relative position in a word. The shape of a character should be changed if it is located at the beginning of the word, in the middle of the word, at the end of the word, and in isolation. An example is shown in Fig. 3.

Table 1 shows a complete set of Farsi character variants.

The shape of some Farsi characters might be changed in a handwritten word depending on the previous character in that word (articulation effect). Fig. 4 shows some examples of articulation effect in handwritten word.

In addition to the cursive nature of Farsi script, the articulation effect makes the segmentation of handwritten Farsi words into its constituent characters very difficult. Therefore, in this paper, a holistic approach is used for handwritten Farsi word recognition. Fig. 5 shows some images of city names of Iran. It is easy to see the similarity in shapes of two very different words (second row of Fig. 5). It is this similarity that poses the biggest challenge in Farsi word recognition.

3. The word recognition system

Fig. 6 shows a block diagram of the word recognition system proposed in this paper. This block diagram consists of the following steps:

Image acquisition: An image of postal envelope is captured using a scanner with 300-dpi resolution and 256 gray levels. Then the name of the city is extracted from the image and assigned an appropriate label between 1 to 198. Our database consists of 17820 word images of the cities in Iran.

16	15	14*	13	12	11	10	9	8	7*	6	5	4	3*	2	1
ش	س	ژ	ز	ر	ذ	د	خ	ح	چ	ج	ث	ت	پ	ب	ا
32	31	30	29	28	27	26*	25	24	23	22	21	20	19	18	17
ی	ه	و	ن	م	ل	گ	ک	ق	ف	غ	ع	ظ	ط	ض	ص

Fig. 1. Isolated Farsi character set.

ع	ـع	ـعـ	عـ
In isolation	At the end of a word	In the middle of a word	At the beginning of a word

Fig. 3. An example of different shapes of a Farsi character.

Table 1
The complete Farsi character set

	4	3	2	1	
Character	Isolate	End	Middle	Beginning	
Alef	(آ) ا	ـا	ـا	(آ) ا	1
Be	ب	ـب	ـبـ	بـ	2
Pe	پ	ـپ	ـپـ	پـ	3
Te	ت	ـت	ـتـ	تـ	4
Se	ث	ـث	ـثـ	ثـ	5
Jim	ج	ـج	ـجـ	جـ	6
Che	چ	ـچ	ـچـ	چـ	7
He	ح	ـح	ـحـ	حـ	8
Khe	خ	ـخ	ـخـ	خـ	9
Dal	د	ـد	ـد	د	10
Zal	ذ	ـذ	ـذ	ذ	11
Re	ر	ـر	ـر	ر	12
Ze	ز	ـز	ـز	ز	13
Zhe	ژ	ـژ	ـژ	ژ	14
Sin	س	ـس	ـسـ	سـ	15
Shin	ش	ـش	ـشـ	شـ	16
Sad	ص	ـص	ـصـ	صـ	17
Zad	ض	ـض	ـضـ	ضـ	18
Ta	ط	ـط	ـطـ	طـ	19
Za	ظ	ـظ	ـظـ	ظـ	20
Ayn	ع	ـع	ـعـ	عـ	21
Ghayn	غ	ـغ	ـغـ	غـ	22
Fe	ف	ـف	ـفـ	فـ	23
Ghaf	ق	ـق	ـقـ	قـ	24
Kaf	ک	ـک	ـکـ	کـ	25
Ghaf	گ	ـگ	ـگـ	گـ	26
Lam	ل	ـل	ـلـ	لـ	27
Mim	م	ـم	ـمـ	مـ	28
Noon	ن	ـن	ـنـ	نـ	29
Waw	و	ـو	ـو	و	30
He	ه	ـه	(ـهـ) ـهـ	هـ	31
Ye	ی	ـی	ـیـ	یـ	32

Preprocessing: The preprocessing consists of the following steps:

- *Binarization*: The gray level image of a word is binarized at a threshold determined by a modified version of maximum entropy sum and entropic correlation methods (Fig. 7(a) and (b)) [9].
- *Noise removal*: The binarized image often has spurious segments which are removed by a morphological closing operation followed by a morphological opening operation both with a 3×3 disk as the structure element [1].
- *Slope correction and baseline estimation*: The baseline in Arabic text usually has the maximum value in the horizontal projection histogram (as shown in Fig. 7(b) and (c)). Therefore from the horizontal projection histogram of word image and rotated images of word by small degree of $\pm\theta$ and using some heuristic rules, the baseline is estimated. Then the skewed image is corrected based on the estimation of the slope of the baseline.
- *Stroke width estimation*: The approximate stroke width of the word image is iteratively estimated by counting the average run-lengths of black pixels in each column of the image. The run-lengths of black pixels that exceed 150% of the average stroke width are excluded from the calculation [4].

Image representation and feature extraction: The main goal of this phase is converting the word image to an appropriate sequential form suitable for the HMM recognizer module. A feature extraction method based on the word image contour is used for this purpose. The following steps are carried out in this phase:

- *Stroke width compensation*: Before feature extraction, a variant of stroke width compensation proposed by Hu et al. [10] is applied to the word image to make the image stroke widths at least three pixels wide to ensure proper contour generation.
- *Baseline centering and contour generation*: The image is enlarged virtually with some blank rows so that the baseline of the word is located in the middle of the image. The chain code of the image contour is then traced. The coordinates (x, y) and the chain-code direction (slope) of each pixel of the contour is saved in a data structure (Fig. 7(d)).
- *Feature extraction*: The area of the image is first divided into a set of vertical fixed-width frames (strips) from right to left. There is a 50% overlap between two consecutive frames. The width of a frame is twice of the average stroke width of the word image (estimated in the preprocessing phase). The overlap allows for

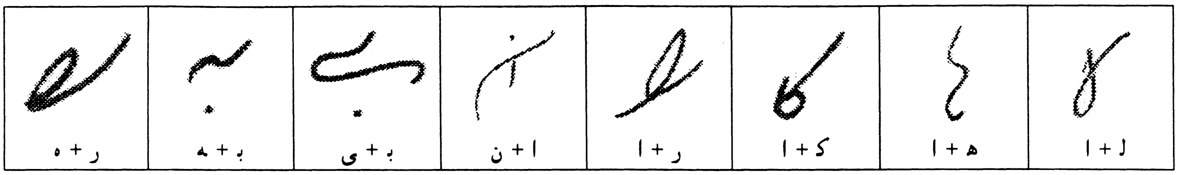

Fig. 4. Examples of the articulation effect in Farsi words.

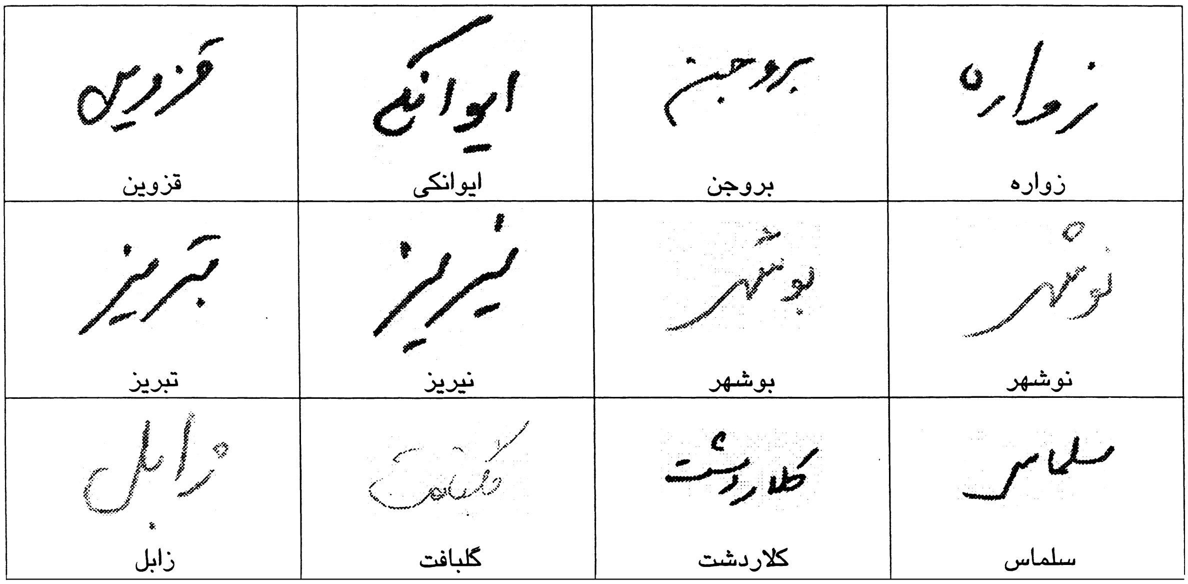

Fig. 5. Examples of Farsi words extracted from our database of city names.

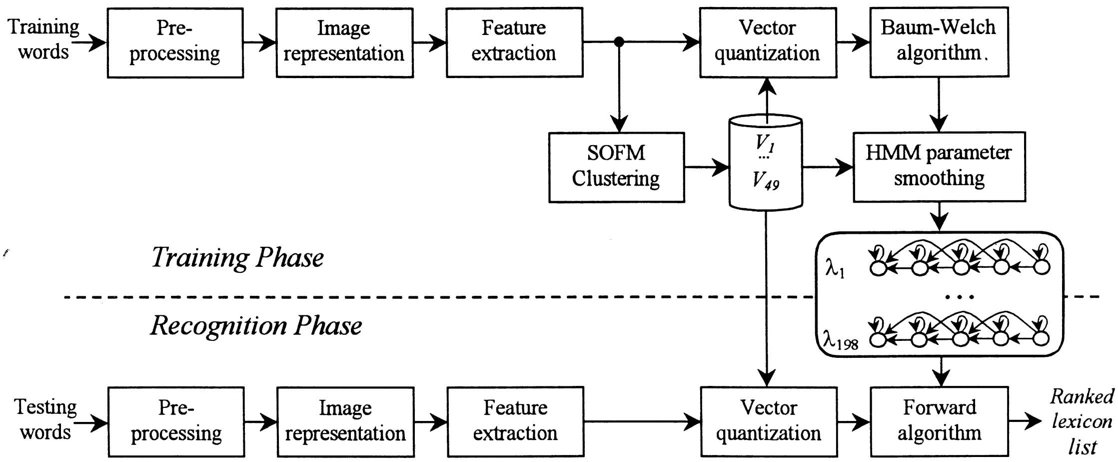

Fig. 6. An overview of proposed handwritten word recognition system.

a smooth transition as one proceeds from one frame to the next. The overlap is a desirable feature for handwritten Farsi words since it can robustly handle normal variations in writing styles of the writers. Then each frame is divided horizontally into five zones with equal height as shown in Fig. 7(d). The choice of five zones was found to yield the best performance. The authors found that comparable results were obtained

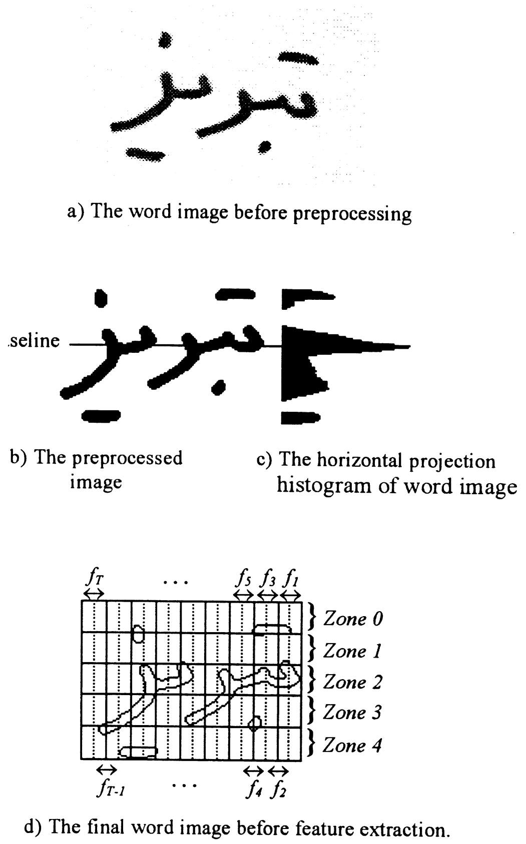

Fig. 7. An example of word image during processing.

with four zones, however, consistently better performance was obtained with five zones. In each zone, a local histogram of the contour chain codes is calculated. Since the contour direction assumes one of the four possible slopes (0, 45, 90, 135°), a histogram in each zone has four components. The components were normalized by dividing the histogram by the height of the zone. This procedure ensures invariance of the components to the height of the character. In this way, each frame is represented as a 20-dimensional feature vector as shown in Fig. 8.

Self-organizing feature map (*SOFM*) *clustering*: The feature space must be quantized into a set of codeword vectors in order to limit the number of observation symbols in discrete hidden Markov model training. The Kohonen self-organization feature map was used to construct the codebook vectors. The extracted feature vectors from more than 400,000 word image frames (strips) are used as the input data file to the Kohonen SOFM clustering program (SOM PAK available via anonymous FTP [8]). The parameters used in SOFM clustering are shown in Table 2.

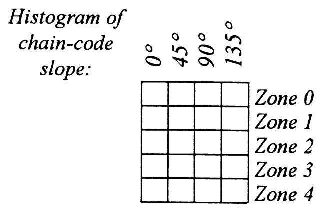

Fig. 8. Chain-code histogram feature vector for a frame.

Table 2
Parameter values used in SOM PAK program

Parameters	Value
Map topology	7×7 Hexagonal
Neighborhood function	Bubble
Initial learning rate in first phase	0.1
Initial learning rate in second phase	0.05
Initial neighborhood radius in first phase	10
Initial neighborhood radius in second phase	3
Learning rate function	Inverse-time type

Fig. 9 shows the best map with the minimum quantization error rate in 10 trials. The weight vectors of the map can be used as the code words of the codebook. Now, a feature vector of image frame can be assigned to the position index (Row, Column) of a node in the map

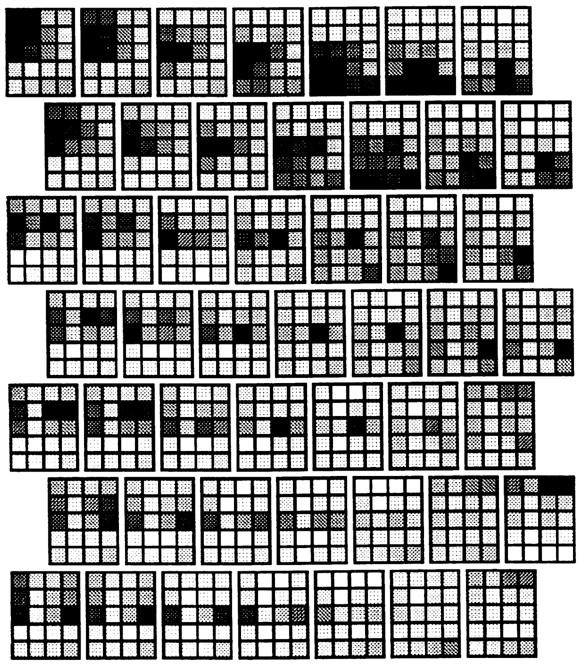

Fig. 9. The SOFM represents codebook vectors.

which represents the closest code word according to the Euclidean distance. Therefore, each image word after this phase will be represented by a sequence of two-dimensional code word positions in the map.

Fig. 9 illustrates the results of SOFM clustering. The authors observed that the neighborhood characteristics of the feature vectors were preserved in SOFM. This property was found to be very useful for smoothing the trained HMM parameters.

4. Hidden Markov model recognizer

The hidden Markov model is a double stochastic process, which can efficiently model the generation of sequential data [7]. HMMs have been successfully used in speech and handwriting recognition. There are two different approaches to model sequential data by HMM [1]. In the model discriminant approach, a separate HMM is used for each class of patterns, while in the path discriminant approach, only one HMM models all of the pattern classes and the different paths in the model distinguish one pattern class from the others. A model discriminant discrete HMM was chosen as the recognition engine that is suitable for this limited-vocabulary application. Therefore, each city class (ω_c) is modeled by a single right–left HMM (λ_c) which is trained using a set of training samples of the same class (c). Each word is represented as a sequence of observation symbols (coordinates of the SOFM codebook). Fig. 10 shows the general structure of a word model.

In the following, first the notations and then the algorithms for manipulating a discrete HMM (λ_c) is described in detail:

- The set of states in the model

  $$S = \{S_1, S_2, \ldots, S_N\}, \tag{1}$$

  where N (the number of states) is the set for each class, and is proportional to the average number of frames of the training samples of that class. The state at time t is denoted as q_t.
- The observation symbol set in each state:

  $$V = \{V_{x,y}\}, \quad 1 \leqslant x \leqslant X, 1 \leqslant y \leqslant Y, \tag{2}$$

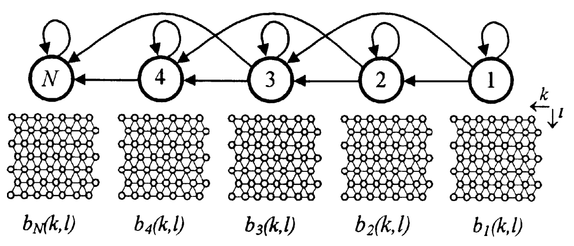

Fig. 10. Sample structure of a right-left HMM.

  where $V_{x,y} = (x, y)$ represents the node with the (x, y) coordinate in the SOFM codebook; and X and Y are the height and width of the map, respectively.
- The initial state distribution $\Pi = \{\pi_i\}$, where

  $$\pi_i = P[q_1 = S_i] = \begin{cases} 0, & i \neq 1, \\ 1, & i = 1. \end{cases} \tag{3}$$

- The last state distribution $\Gamma = \{\gamma_i\}$, where

  $$\gamma_i = P[q_T = S_i] = \begin{cases} 0, & i \neq N, \\ 1, & i = N. \end{cases} \tag{4}$$

- The state transition probability distribution $\mathbf{A} = \{a_{ij}\}$, where

  $$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leqslant i, j \leqslant N, \tag{5}$$

  and

  $$a_{ij} = 0 \quad \text{if } (j < i) \text{ or } (j > i + \Delta). \tag{6}$$

  The maximum number of forward jumps in each state (Δ) is chosen experimentally to be between 2 and 4 for each class during training.
- The observation symbol probability distribution $\mathbf{B} = \{b_j(x, y)\}$, where

  $$b_j(x, y) = P[V_{x,y} \text{ at } t | q_t = S_j], \quad 1 \leqslant j \leqslant N, 1 \leqslant x \leqslant X, 1 \leqslant y \leqslant Y. \tag{7}$$

Let O be the set of K training samples used for the word model λ_c.

$$O = \{O^{(1)}, O^{(2)}, \ldots, O^{(K)}\}, \tag{8}$$

where the training word $O^{(k)}$ is represented as a sequence of T_k observation symbols.

$$O^{(k)} = (O_1^{(k)}, O_2^{(k)}, \ldots, O_{T_k}^{(k)}) \tag{9}$$

and each observation symbol represents the coordinates of a node in the codebook map.

The forward variable for a given word sample k is calculated as

$$\alpha_t^{(k)}(j) = \begin{cases} \pi_j \cdot b_j(O_t^{(k)}), & t = 1, \\ \left[\sum_{i=1}^{N} \alpha_{t-1}^{(k)}(i) \cdot a_{ij}\right] \cdot b_j(O_t^{(k)}), & t = 2, \ldots, T_k, \end{cases} \quad 1 \leqslant j \leqslant N. \tag{10}$$

Similarly, the backward variable for a given word sample k is calculated as

$$\beta_t^{(k)}(j) = \begin{cases} \gamma_i, & t = T_k, \\ \sum_{i=1}^{N} a_{ji} \cdot b_j(O_t^{(k)}) \cdot \beta_{t+1}^{(k)}(i), & t = T_k - 1, \ldots, 1, \end{cases} \quad 1 \leqslant j \leqslant N, \tag{11}$$

The probability of generating a word sample k by the model λ_c is obtained by:

$$P_k = P(O^{(k)}|\lambda_c) = \sum_{i=1}^{N} \alpha_{T_k}^{(k)}(i) \cdot \gamma_i. \tag{12}$$

Prior to recognition, each word model λ_c should be trained independently by Baum–Welch algorithm [7] to maximize the overall probability of all of the K observation sequences. In the right–left HMM, the initial state distribution (Π) and the last state distribution (Γ) are predefined as shown in Eqs. (3) and (4). The state transition probability distribution (**A**) and the observation probability distribution (**B**) are re-estimated as follows:

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K}(1/P_k)\sum_{t=1}^{T_k-1}\alpha_t^{(k)}(i) \cdot a_{ij} \cdot b_j(O_{t+1}^{(k)}) \cdot \beta_{t+1}^{(k)}(j)}{\sum_{k=1}^{K}(1/P_k)\sum_{t=1}^{T_k-1}\alpha_t^{(k)}(i) \cdot \beta_t^{(k)}(i)},$$
$$1 \leqslant i, j \leqslant N, \tag{13}$$

$$\bar{b}_j(x, y) = \frac{\sum_{k=1}^{K}(1/P_k)\sum_{t=1,\ s.t.\ O_t^{(k)}=V_{x,y}}^{T_k}\alpha_t^{(k)}(j) \cdot \beta_t^{(k)}(j)}{\sum_{k=1}^{K}(1/P_k)\sum_{t=1}^{T_k}\alpha_t^{(k)}(j) \cdot \beta_t^{(k)}(j)},$$
$$i \leqslant j \leqslant N, 1 \leqslant x \leqslant X, 1 \leqslant y \leqslant Y. \tag{14}$$

It is well known that if sufficient training data is not provided, HMM parameters, especially the observation symbol probabilities, are usually poorly estimated. Consequently, the recognition rate is adversely affected by even a slight variation in the testing data. This fact is clearly revealed in the experimental result shown in Table 3. An appropriate smoothing of the estimated observation probability can overcome this problem without the need for more training data. The parameter smoothing method proposed by Zhao et al. [11] was used in this study. In this method, smoothing of the observation symbol probabilities is achieved by using the neighborhood information preserved in the SOFM codebook. After training all of the HMMs by Baum–Welch algorithm (Eqs. (13) and (14)), 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 self-organization map. Details are provided below:

$$b_j^{new}(x, y) = b_j^{old}(x, y) + \sum_{(u,v) \neq (x,y)} W_{(x,y),(u,v)} \cdot b_j^{old}(u, v),$$
$$1 \leqslant j \leqslant N, 1 \leqslant x \leqslant X, 1 \leqslant y \leqslant Y, \tag{15}$$

where the weighting coefficient $W_{(x,y),(u,v)}$ is a function of the distance between two nodes (x, y) and (u, v) in the map

$$W_{(x,y),(u,v)} = Sf \cdot \zeta^{(d_{(x,y),(u,v)} - 1)}, \quad 0 < \zeta < 1 \tag{16}$$

and ζ is a constant, chosen to be equal to 0.5. The smoothing factor (Sf) controls the degree of smoothing, and $d_{(x,y),(u,v)}$ is the hexagonal distance between two nodes with the coordinates (x, y) and (u, v) in the codebook map.

Table 3
Recognition result before smoothing HMM parameters

Top-n	1	2	5	10	20
Recognition Rate	32.04	44.72	69.47	86.41	93.63

5. Experimental results

More than 17,000 images of 198 city names of Iran were collected in a database. The images were manually labeled to one of the 198 classes. A subset of 60% of images was randomly chosen for building the training data set and the remaining images were used as the testing data set. The Kohonen SOFM clustering program [8] was used to construct a codebook from more than 400,000 chain-code histogram feature vectors extracted from vertical frames of the training data set. Fig. 9 shows the resulting codebook. For each class c, the best right–left HMM (λ_c) was chosen from the trained HMMs using Baum–Welch algorithm under different conditions such as:

- Initializing parameters with random or equal values.
- Different topology parameters (the number of states (N), the connectivity of states (Δ)).

Then, each word image in the test data set was represented as a sequence of T observations, $O = \{o_t\}$. The probability that O has been generated by each word model, $P(O|\lambda_c)$, $1 \leqslant c \leqslant 198$, was computed by the forward algorithm (Eqs. (10) and (12)), and a sorted list of candidate classes were obtained. The performance of the word recognition system is illustrated in Table 3 by a top-n recognition rate measure (the percentage of samples that the true class is among the first n positions in the candidate list).

As previously mentioned, due to the problem of insufficient training data, the recognition rate is low. The above procedure was repeated after smoothing the HMMs with a different smoothing parameter (Sf) and the recognition results are shown in Table 4.

From Table 4, it is seen that the top choice recognition rate increased significantly from 32.04% without smoothing, to 65.05% with a smoothing factor equal to 0.01.

Since the standard maximum-likelihood optimization is used for estimating the HMM parameters, each word model (λ_c) is trained only by observation sequences of the same word class (c). When the probabilities of generating an unknown word image by a model are calculated for all

Table 4
Recognition result after smoothing HMM parameters

Smoothing factor (Sf)	Recognition rate (Lexiconsize = 198)				
	Top-1	Top-2	Top-5	Top-10	Top-20
0.1	62.96	74.13	84.40	89.67	94.56
0.01	**65.05**	**76.09**	**86.08**	**90.83**	**95.00**
0.001	62.68	74.04	85.69	91.35	94.89
0.0001	58.75	71.66	84.29	90.46	94.72

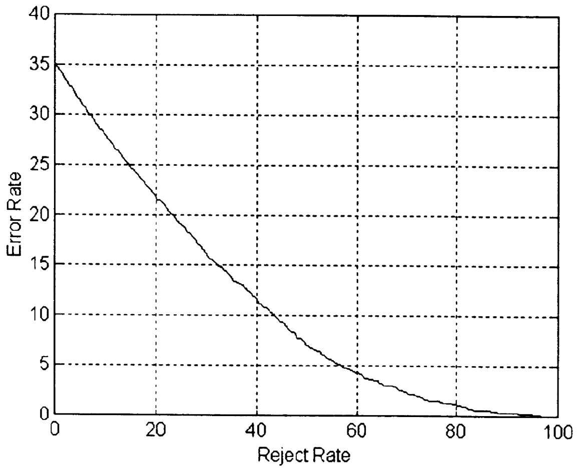

Fig. 11. Error rate (in top-1 case) versus rejection rate.

of the 198 classes in the recognition phase, there is a possibility that in addition to the correct class, some other classes also have high observation probabilities. The existence of some very similar lexicon entries (e.g. samples in the second row of Fig. 5) increases this possibility. In order to improve the reliability of the recognition process, a rejection feature is incorporated. Thus a test image, whose difference of the observation probabilities of the best two word models is less than a threshold, is rejected. Another classifier using contextual information can handle the rejected cases. Fig. 11 shows the error rate versus rejection rate obtained using different values of the threshold.

The misclassified words were examined to determine the main sources of the error. Besides the existence of very similar lexicon entries, the incorrect estimation of baseline, existence of some difficult-to-read samples (even by a human observer) are the principal causes of the error. Fig. 5 illustrates images that were not recognized correctly.

It is very difficult to give comparative results with other Farsi/Arabic handwritten word recognition systems because of the unavailability of such systems to the authors. Instead, the results were compared with those of other Latin word recognition systems in Table 5. The

Table 5
Comparisons with other word recognition systems in the literature

Method	Lexicon size	Recognition rate		
		Top-1	Top-2	Top-20
CDVDHMM, Chen [1]	271	67.0	75.5	88.3
Proposed system	**198**	**65.0**	**76.1**	**95.0**
DP, Kim [4]	100	84.6	91.2	99.0
KNN + HMM, Guillevic [2]	30	86.7	94.6	99.9

results obtained are very promising when one takes into account

- the number of classes in our lexicon (198 classes).
- the specific characteristics of Arabic/Farsi script, which make the recognition more complicated such as cursiveness, and dependence of the Farsi scripts to the location of the dots.
- No explicit character pattern is extracted in the proposed system. The recognition is, in this sense, holistic.

6. Summary and conclusions

This paper establishes the feasibility of holistic recognition of Farsi handwritten words using hidden Markov models (HMM). Although the accuracy of recognition may be considered relatively low (65%), in comparison with other reported results, the authors note that the recognition of Farsi handwritten words is considerably more difficult than the recognition of words based on the English alphabet, due to ambiguities in writing that can lead to mis-recognition. However, using heuristics and contextual information, significantly higher accuracy can be achieved. The authors did not use contextual information in this study, as this will mask the effectiveness of the HMM approach.

In conclusion, the following observations are noted with regard to the proposed recognition system:

(1) Recognition of Farsi handwritten words is shown to be feasible.
(2) Top choice accuracy of 65% without rejection is comparable to other published results for Latin words. It is worth noting that these results were achieved without using contextual information.

References

[1] M.Y. Chen, A. Kundu, S.N. Srihari, Variable duration hidden Markov and morphological segmentation for handwritten word recognition, IEEE Trans. Image Process. 4 (12) (1995) 1675–1688.

[2] D. Guillevic, C.Y. Suen, HMM-KNN word recognition engine for bank cheque processing, Proceedings of International Conference on Pattern Recognition, Vol. 2, Brisbane, Australia, August 1998, pp. 1526–1529.
[3] A.W. Senior, A.J. Robinson, An off-line cursive handwriting recognition system, IEEE Trans. Pattern Anal. Mach. Intel. 20 (3) (1998) 309–321.
[4] G. Kim, V. Govindaraju, A lexicon driven approach to handwritten word recognition for real-time applications, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 366–379.
[5] A. Amin, Off-line Arabic character recognition: the state of the art, Pattern Recognition 31 (5) (1998) 517–530.
[6] B. Al-Badr, S.A. Mahmoud, Survey and bibliography of Arabic optical text recognition, Signal Processing 41 (1995) 49–77.
[7] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.
[8] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, SOM_PAK: The Selp-Organizing Map Program Package, Version 3.1, Helsinki University of Technology, Finland, 1995.
[9] P. Sahoo, C. Wilkins, J. Yeager, Threshold selection using Renyi's entropy, Pattern Recognition 30 (1) (1997) 71–84.
[10] J. Hu, D. Yu, H. Yan, Algorithm for stroke width compensation of handwritten characters, Electron. Lett. 32 (24) (1996) 2221–2222.
[11] Z. Zhao, C.G. Rowden, Use of Kohonen self-organising feature maps for HMM parameter smoothing in speech recognition, IEE Proceedings, Part F 139 (6) (1992) 385–390.

About the Author—MEHDI-DEHGHAN received his B.Sc. in Electrical Engineering from Esfahan University of Technology in Esfahan, Iran in 1992 and M.Sc. in Electrical Engineering from Amirkabir University of Technology in Tehran, Iran in 1995. He is currently a candidate for Doctoral degree in Amirkabir University of Technology. Mr. Dehghan has been a visiting scholar in the Department of Electrical and Computer Engineering at the University of Windsor during February 1999 to January 2000 through a graduate student exchange program. Mr. Dehghan's Ph.D. thesis is in the area of handwritten Farsi word recognition using Hidden Markov Model.

About the Author—K. FAEZ received his B.Sc. in Electrical Engineering from Tehran Polytechnique in Tehran, Iran in 1973, M.Sc. and Ph.D. in Computer Science from University of California, Los Angeles in 1977 and 1980 respectively. Dr. Faez is currently Professor of Electrical Engineering at Amirkabir University of Technology in Tehran, Iran. Dr. Faez's area of research includes Fault-Tolerant System Design, Pattern Recognition, Image Processing, Hardware Design and Computer Network. He has published over fifty papers in the above areas.

About the Author—M. AHMADI received B.Sc. (EE) degree from Arya Mehr University in Tehran, Iran, and Ph.D. degree from Imperial College of London University, London, U.K., in 1970 and 1977, respectively. He has been with the Department of Electrical Engineering, University of Windsor, Canada, since 1980 and holds the rank of Professor. His research interests include design, stability and realization of 2-D filters. Pattern Recognition and Machine Vision. Dr. Ahmadi co-authored the book on *Digital Filtering in 1-D and 2-D Design and Applications*, published by Plenum, 1989. He has published over 280 articles in the above area. Dr. Ahmadi is a Fellow of the Institution of Electrical Engineers in England (FIEE) and is a Senior Member of IEEE.

About the Author—M. SHRIDHAR received his M.Sc. in Electrical Engineering from Polytechnic Institute of Brooklyn, New York, in 1968 and Ph.D. in Electrical Engineering from the University of Aston in Birmingham, U.K. in 1970. From 1969 to 1985, Dr. Shridhar was a faculty member in the Electrical Engineering Department of the University of windsor, Canada. since 1986, he has been with the University of Michigan-Dearborn where he is currently a Professor and chairman of the Electrical and Computer Engineering Department. His research interests are in the area of image processing, pattern recognition and computer vision.