Evaluation of Skeletonization Methods for Arabic/Farsi Handwriting Recognition

M. M. Haji
Shiraz University
Department of Computer Science and Engineering
mehdi.haji@gmail.com

S. D. Katebi
Shiraz University
Department of Computer Science and Engineering
katebi@shirazu.ac.ir

ABSTRACT
A goal-directed evaluation of four skeletonization methods for recognizing Arabic/Farsi handwritten words is presented. Two of which are well established and the third one has recently been reported. The fourth one is a novel parallel method proposed by the authors. This new method with the addition of a postprocessing step is shown to be the best taking into the consideration the fact that the handwriting recognition system using it yields higher recognition rates. The recognition is based on continuous hidden Markov models (CHMMs) with structural features.

Keywords
Skeletonization, Thinning, Postprocessing, Structural Features, Arabic/Farsi Script, Handwriting Recognition, Evaluation.

1. INTRODUCTION
Skeletonization due to its numerous applications has been one of the most surveyed problems in machine vision. A skeletonization algorithm, also referred to as medial axis transform, homotopic thinning or topology preserving shrinkage, transforms a shape into arcs and curves of unitary thickness which is called skeleton. Ideally, the skeleton must retain basic structural properties of the original shape; it must be well-centered, well-connected (preserve connectivity information), robust and also allows a precise reconstruction [1]. Over years, it has been found to be difficult to design an algorithm that satisfies all of the requirements. Also there is no unique definition for skeleton, so different algorithms with different definitions can produce different skeletons for the same shape.

Skeletons have been proved to be effective in pattern recognition problems such as character recognition, fingerprint recognition and chromosome recognition. Skeletons provide compact representations that allow structural analysis of objects. By diminishing variability and distortion of the instances which belong to one class and reducing the amount of the data to be handled, skeletonization simplifies classification.

Skeletonization methods can be divided into two major categories [2]: direct and indirect. A direct method produces the skeleton by directly removing pixels from the pattern. Direct methods can be further classified into iterative and non-iterative. An iterative direct method computes the skeleton by iteratively deleting removable boundary pixels either sequentially [3] or in parallel [4], until it causes no further changes to the image. In iterative methods, any pixel is tested and marked to be removed if its neighbors (usually 8-neighbors) satisfy certain conditions. In sequential methods, all pixels are tested in a fixed order in all iterations and removing a pixel in any iteration depends both on the resultant image of the previous iteration and the previous operations of the current iteration. But in parallel algorithms, removing a pixel only depends on the resultant image of the previous iteration, so all pixels can be tested independently in any iteration. Iterative methods yield thin and geometrically representative but not necessarily well-centered skeletons.

A non-iterative method produces the skeleton by connecting pixels having special properties. A pixel with special properties may be the middle pixel of a component of a scan line, or the parts of polygonal regions where a pattern is divided into a set of regular or irregular polygons, etc. [2].

An indirect method [2] is very similar to a non-iterative one as it does not construct the skeleton by removing or changing pixels, but it does the task by computing appropriate logical properties such as distributions of pattern pixels. Indirect methods are proved to perform better in some applications but usually slower than direct ones. Therefore, indirect methods are usually neglected in favor of direct ones.

Skeletonization algorithms have been notorious for irrational memory and CPU usage, but it should be noted that such problems are only pronounced for large images. Neither memory nor CPU inefficient usage is important for a typical word image of size 300×300 or so; by using fast ubiquitous hardware, almost all algorithms are practical for text recognition.

There are hundreds of skeletonization algorithms in the literature [5, 6]. Of course it is not practical to implement and experiment all of them. For the skeletonization of Arabic/Farsi script, four algorithms were implemented: two classical methods [3,4], the proposed method which is designed for Arabic/Farsi script and Huang et al.'s fully parallel method [7]. Most skeletonization algorithms, including the four ones evaluated here, work on binary images, so each input image is first binarized [8], and then skeletonization and feature extraction are performed. However, document image binarization may be avoided by using a grey-scale skeletonization approach [9] which directly works on gray-scale images.

In the rest of this paper, each of these methods is briefly described, then a simple and effective skeleton postprocessing procedure is presented. Finally, the proposed method is shown to be the best for the structural recognition of Arabic/Farsi handwriting.

2. EVALUATED ALGORITHMS
2.1 SPTA
The Safe-Point Thinning Algorithm (SPTA) [3] is a sequential one. Like other iterative algorithms, it consists of iteratively deleting edge-points (those along the edges of a shape) while keeping end-points (those at the ends of a stroke), also the shape connectedness should not be broken and excessive erosion (iteratively removing a stroke) should not be occurred.

Hereafter, it is assumed that in a binary input image, the shape pixels are represented by black pixels and background pixels by white pixels. For a point p with the coordinate (x,y), the set of points with the coordinates $(x+1,y)$, $(x-1,y)$, $(x,y-1)$ and $(x,y+1)$ are called its 4-neighbours, and its 8-neighbors are the set of points with the coordinates (x+1,y), (x+1,y-1), (x,y-1), (x-1,y-1), (x-1,y), (x-1,y+1), (x,y+1) and (x+1,y+1) (see Figure 1).

n_3	n_2	n_1
n_4	p	n_0
n_5	n_6	n_7

Figure 1. A point p and its 8-neighbors (n_0 to n_7). The points n_0, n_2, n_4 and n_6 are referred to as the 4-neighbors of p.

In the SPTA, an edge-point is defined as a black pixel with at least one white 4-neighbor, an end-point is defined as a black point with at most one black 8-neighbor and a break-point is defined as a point whose deletion would break the connectedness of the pattern. The algorithm in each pass flags a point if it is an edge-point but not an end-point, nor a break-point, and nor must its possible deletion cause excessive erosion. All flagged points are removed at the end of a pass, and if there is no flagged point the procedure stops. An edge-point can be of one or more of the following types: 1) a left-edge point, having its left neighbor n_4 white; 2) a right-edge point, having its right neighbor n_0 white; 3) a top edge-point, having its top neighbor n_2 white; and 4) a bottom edge-point having its bottom neighbor n_6 white.

By examining different combinations of the 8-neighbors of a left-edge point p the authors have concluded that p can be safely removed (without breaking connectedness, end-point deletion and excessive erosion) if the boolean expression S_4 is true:

$$S_4 = n_0.(n_1 + n_2 + n_6 + n_7).(n_2 + \overline{n}_3).(n_6 + \overline{n}_5) \qquad (1)$$

A boolean variable has the true value if its corresponding point is black and unflagged. Similarly, for a right-edge point, trueness of the expression S_0, for a top-edge point, trueness of the expression S_2 and for a bottom-edge point, trueness of the expression S_6 are sufficient conditions for safe deletion of the corresponding edge-points.

$$S_0 = n_4.(n_5 + n_6 + n_2 + n_3).(n_6 + \overline{n}_7).(n_2 + \overline{n}_1) \qquad (2)$$

$$S_2 = n_6.(n_7 + n_0 + n_4 + n_5).(n_0 + \overline{n}_1).(n_4 + \overline{n}_3) \qquad (3)$$

$$S_6 = n_2.(n_3 + n_4 + n_0 + n_1).(n_4 + \overline{n}_5).(n_0 + \overline{n}_7) \qquad (4)$$

Each pass in the SPTA involves two scans, where all black points (the shape points) are examined in each scan. The scanning sequence can be either row-wise or column-wise. The first scan of each pass, flags safely removable left-edge points and safely removable right-edge points. In the second scan of the pass, safely removable top-edge points and safely removable bottom-edge points are flagged. At the end of the pass, all flagged points are removed (become white) and the next pass starts.

2.2 Zhang-Suen's Algorithm

This algorithm has been used as the basis of comparison for skeletonization algorithms for many years. It is a fast and simple parallel iterative algorithm, meaning that the new value for a pixel can be calculated using only the values from the previous iteration.

Each pass in the algorithm involves two sub-iterations, where in each sub-iteration, certain points are flagged, and at the end of the sub-iteration if there is no flagged point the algorithm stops; otherwise the flagged points are removed and the next sub-iteration starts. In the first sub-iteration, a pixel is flagged if it satisfies all of the four following conditions:

1. Its connectivity number is one. The connectivity number C_n of a pixel p can be defined as the number of transitions from black (foreground) to white (background) within the pixel 8-neighbors. It has a value in the range of zero to four.
2. It has at least two and at most six black neighbors.
3. At least one of n_0, n_4 and n_6 is white.
4. At least one of n_0, n_2 and n_6 is white.

Now, if there is no flagged point the algorithm stops; otherwise, all flagged point are removed and the second sub-iteration starts which is the same as the first sub-iteration except for conditions 3 and 4:

3. At least one of n_0, n_2 and n_4 is white.

4. At least one of n_2, n_4 and n_6 is white.

Zhang-Suen's algorithm has been widely used in general machine vision application. It has also been used in a recent study on the structural recognition of Arabic handwriting [10], but as it will be shown later, the algorithm sometimes can remove the letter dots, which carry the necessary information to distinguish certain Arabic/Farsi letters from each other. In fact, it always removes 2×2 squares and sometime cause excessive erosion. Therefore, this skeletonization is actually a source of error and must not be used in the context of text recognition.

2.3 Proposed Algorithm

To overcome the problems of Zhan-Suen's algorithm, a new parallel iterative algorithm is developed for Arabic/Farsi script. The basic idea of this algorithm has been initiated in [11] and further developed and implemented in this work. The algorithm involves four sub-iterations in each pass. All shape (black) pixels are examined in each sub-iteration. Certain points are flagged within a sub-iteration; at the end of the sub-iteration if there is no flagged point the algorithm stops; otherwise the flagged points are removed and the next sub-iteration starts.

In the first sub-iteration, each left-edge point for which the boolean expression D_4 is true is flagged. In the second sub-iteration, each bottom-edge point for which the boolean expression D_6 is true is flagged. In the third sub-iteration, each right-edge point for which the boolean expression D_0 is true is flagged. In the forth (last) sub-iteration, each top-edge point for which the boolean expression D_2 is true is flagged. Where the definitions of the left-edge, right-edge, top-edge and bottom-edge points are the same as those of the SPTA and:

$$D_0 = S_0.(n_7 + n_1 + (n_2 + n_6 + n_3 \oplus n_5).(n_3 + n_5 + \overline{n}_2.\overline{n}_6)) \qquad (5)$$

$$D_2 = S_2.(n_1 + n_3 + (n_4 + n_0 + n_5 \oplus n_7).(n_5 + n_7 + \overline{n}_4.\overline{n}_0)) \qquad (6)$$

$$D_4 = S_4.(n_3 + n_5 + (n_6 + n_2 + n_7 \oplus n_1).(n_7 + n_1 + \overline{n}_6.\overline{n}_2)) \qquad (7)$$

$$D_6 = S_6.(n_5 + n_7 + (n_0 + n_4 + n_1 \oplus n_3).(n_1 + n_3 + \overline{n}_0.\overline{n}_4)) \qquad (8)$$

These expressions have been derived by processing all the 8-neighbours configurations in which the central pixel is neither an end-point nor a break-point and also its deletion does not cause excessive erosion.

From experiments it is observed that the algorithm becomes faster if the values of D_0, D_2, D_4 and D_6 for all 256 combinations of the 8-neighbors (n_0 to n_7) are stored at first in a table rather than evaluating them from the explicit expressions at runtime.

2.4 Huang et al.'s Algorithm

Huang et al. have recently proposed a fully parallel thinning algorithm [7] which involves one iteration in each pass. It uses the information of 3×3 windows (i.e. the state of 8-neighbors) as the previous iterative algorithms do, but in order to preserve connectivity, 3×4, 4×3 and 4×4 windows are used also. The algorithm has been shown to be efficient and robust to border noise.

All of the following rules are applied simultaneously to each pixel p to determine whether it should be flagged or not:

• If p has zero, one or eight black neighbors, it is not flagged.

• If p has two black neighbors, it is flagged if the two neighbors are consecutive, i.e. n_0 and n_1 are black, or n_1 and n_2 are black, or n_2 and n_3 are black, …, or n_7 and n_0 are black.

• If p has three black neighbors, it is flagged if the three neighbors are consecutive, or if they match any of the following templates:

0	1	0
1	*p*	0
1	0	0

0	1	0
0	*p*	1
0	0	1

1	1	0
0	*p*	1
0	0	0

0	1	1
1	*p*	0
0	0	0

Where 1 denotes a black and 0 denotes a white pixel.

• If p has four black neighbors, it is flagged if the four neighbors are consecutive, or if they match any of the following templates:

1	1	0
0	*p*	1
0	0	1

0	1	1
1	*p*	0
1	0	0

• If p has five black neighbors, it is flagged if the five neighbors are consecutive.

• If p has six black neighbors, it is flagged if the six neighbors are consecutive.

• If p has seven black neighbors, It is flagged if its white neighbor is a 4-neighbor.

The above rules remove two-pixel-width rectangular patterns, resulting in loss of information or pattern connectivity. To obviate this problem, the pixel p is preserved (not flagged) if it matches any of the following templates:

x	0	x
1	*p*	1
1	1	1
x	0	x

x	0	0
1	1	0
0	*p*	0
0	0	x

x	0	0
0	*p*	0
0	1	1
0	0	x

0	0	0	0
0	*p*	1	0
0	1	1	0
0	0	0	0

x	0	0	0
0	*p*	1	0
0	0	1	x

x	1	1	x
0	*p*	1	0
x	1	1	x

0	0	0	x
0	1	*p*	0
x	1	0	0

At the end of a pass, if there is no flagged pixel the algorithm stops; otherwise the flagged pixels are removed and the next pass starts.

3. POSTPROCESSING

A skeletonization algorithm usually produces a distorted skeleton with some spurious branches which need a postprocessing step to be removed. An example of a handwritten word is given in Figure 2. As shown in Figure 3, the skeletons of this image have unwanted branches resulting from the border noise. However, the four algorithms behave very differently: the skeleton obtained using Zhang-Suen's algorithm has no spurious branch, but the skeleton obtained using the proposed approach has nine spurious branches. By many other experiments it was concluded that without postprocessing Huang et al.'s and Zhang-Suen's algorithms were the robustest against border noise while the proposed algorithm was the least robust one.

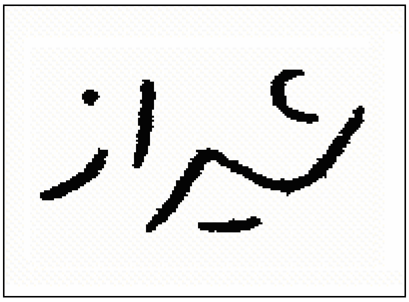


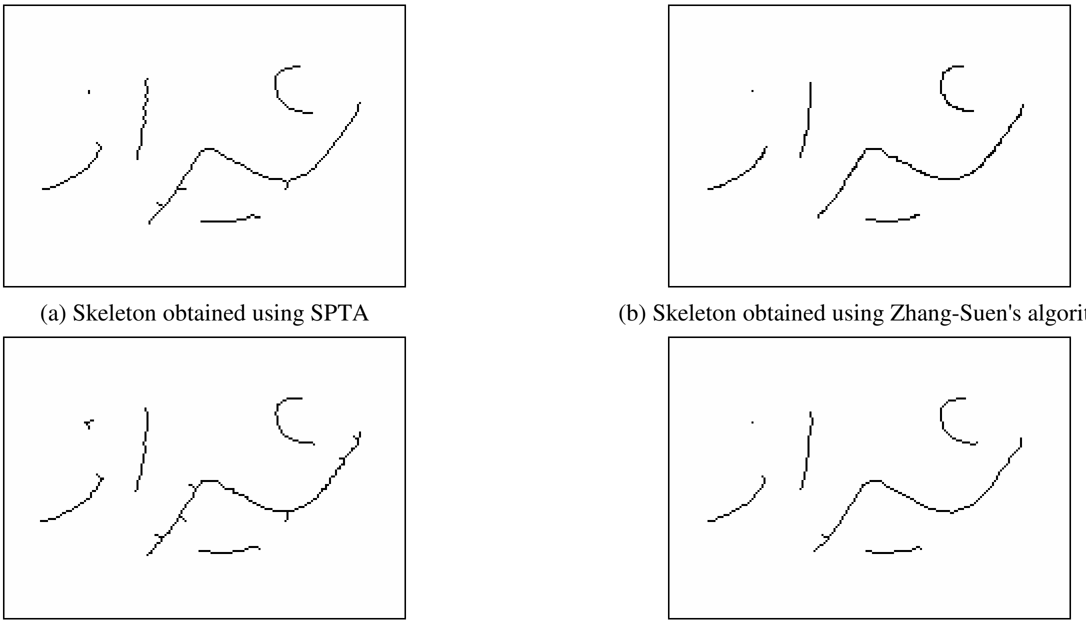
Figure 2. Example of a handwritten word.

The simple technique described here uses the maximum circle idea. Since the local features of the pattern are affected, the original pattern is also used to modify the skeleton. In order to explain the postprocessing procedure clearly, the following definitions are given first:

Definition 1. A feature-point is a black pixel in the skeleton with a connectivity number other than two; i.e. p is a feature-point if and only if $C_n(p)\neq 2$.

Definition 2. An end-point is a feature-point with a connectivity number equal to one; i.e. p is an end-point if and only if $C_n(p)=1$. An end-point can be deleted without affecting the pattern connectivity.

The postprocessing procedure is as follows: first, for each end-point ep, the radius R_{ep} of the largest circle of black pixels within the original image that is centered at ep is evaluated. Then, the nearest non end-point nep to ep is found, and the link between ep and nep is removed if $\mathrm{dist}(ep,nep) < R_{ep} + R_{nep}$. Where R_{nep} is the radius of the largest circle of black pixels within the original image that is centered at nep.



(a) Skeleton obtained using SPTA

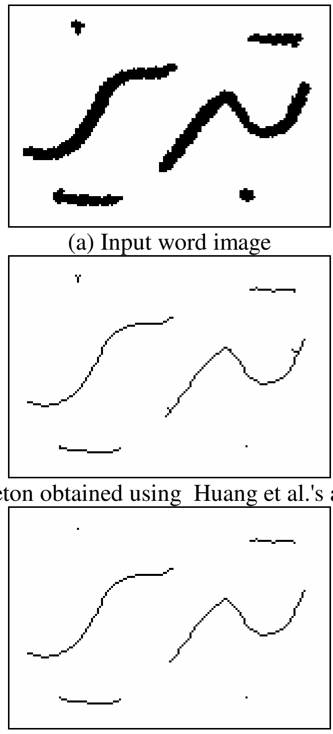
(b) Skeleton obtained using Zhang-Suen's algorithm

(c) Skeleton obtained using proposed algorithm

(d) Skeleton obtained using Huang et al.'s algorithm

Figure 3. Robustness of implemented skeletonization algorithms against border noise.

The advantage gained by the postprocessing step is shown in Figure 4. The input word image has a great deal of border noise, and Huang et al.'s algorithm, which is robust to border noise, is used for skeletonization. As shown in Figure 4(b), there are still some spurious branches in the skeleton which are totally removed after the postprocessing step.



(a) Input word image

(b) Skeleton obtained using Huang et al.'s algorithm

(c) Skeleton after postprocessing

Figure 4. Removing spurious branches by postprocessing.

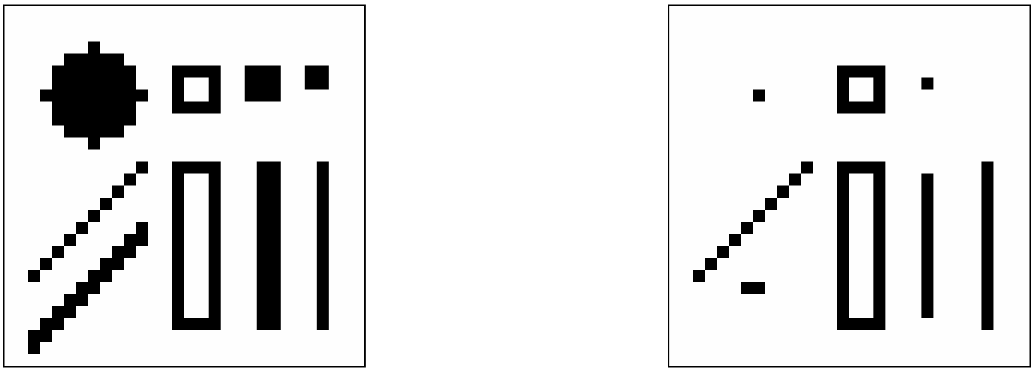
4. EXPERIMENTAL RESULTS

For visual judgment about the quality of a skeletonization algorithm, the following criteria are considered: the width and connectivity of the skeleton, excessive erosion and robustness to border noise. Obviously for recognition purposes, a skeletonization algorithm must also have the following properties: preserving text characteristics, such as not removing dots, and producing well-connected skeletons of unitary thickness. Rather than going into a long detailed explanation, ineffectiveness of Zhang-Suen's algorithm can simply be shown by actual examples. For the input image of Figure 5(a), the skeleton obtained using Zhang-Suen's algorithm does not have the desired properties. All other algorithms produce acceptable outputs for this input image. The algorithms are also applied to Farsi (Figure 6(a)) and English (Figure 6(c)) character set. As shown in Figure 6(b), Zhang-Suen's algorithm removes some of the dots, so some letters have the same skeleton, for example 'ت' and 'ث', which leads to misidentification. Also, notice the skeleton of 'K', in the image of Figure 6(d), which has been excessively eroded. Thus, Zhang-Suen's algorithm is suitable neither for Arabic/Farsi nor for English. The other three algorithms produce acceptable skeletons for both character sets.

Each implemented skeletonization algorithm is evaluated by the recognition rate attained when it is incorporated in the feature extraction step of a handwriting recognition system [12]. In other words, different recognition systems with every identical aspects but different skeletonization algorithms are trained and evaluated on the same dataset. The dataset contains 100 city names (i.e. there are 100 classes) of Iran with 180 samples for each city. 120 randomly selected samples of each class were used for training and the remaining 60 ones were used for evaluating the performance.

Table 1. N-best recognition rate of system for different skeletonization methods with and without postprocessing

	with postprocessing						without postprocessing					
	N=1	N=2	N=3	N=4	N=5	N=10	N=1	N=2	N=3	N=4	N=5	N=10
SPTA	.819	.890	.917	.931	.940	.961	.667	.771	.826	.861	.891	.949
Zhang Suen's	.811	.883	.907	.922	.931	.953	.781	.860	.890	.906	.919	.951
Huang et al.'s	.825	.896	.920	.933	.942	.963	.811	.882	.909	.923	.934	.959
Proposed	.835	.902	.923	.937	.947	.967	.607	.720	.773	.807	.830	.890



(a) Input image (b) Skeleton obtained using Zhang-Suen's algorithm

Figure 5. Applying Zhang-Suen's algorithm to a binary image containing simple geometrical objects. The 2×2 square is completely removed and the two-pixel-width slanted line is excessively eroded.

A handwriting recognition system can provide a list of hypotheses rather than a single hypothesis for each word image. Thus, the performance of such a system is usually expressed by a criterion called N-best recognition rate. A word image is said to be N-best recognized when the correct word hypothesis is one of the first (most probable) N hypotheses for the minimum value of N. When it is said that the N-best recognition rate of a system is α it means that in α percent of the experiments the correct word hypothesis is one of the first N hypotheses generated by the system. Obviously N-best recognition rate increases with N and definitely will be 100% when N equals to the lexicon size.

Table 1 shows the N-best recognition rate of the system for different skeletonization algorithms with and without postprocessing. As seen the postprocessing step in all cases improves the recognition rate. Overall, of the four algorithms, the proposed one with the addition of the postprocessing step results in highest recognition rates, however without postprocessing this algorithm is by far the worst one because it often produces many spurious branches. As expected, without postprocessing Huang et al.'s algorithm is the best because it produces hardly any spurious branches. Ineffectiveness of Zhang-Suen's algorithm, which was previously shown by examples, is verified here as the lowest recognition rates are due to this algorithm. With postprocessing, Huang et al.'s algorithm performs slightly better than the SPTA.

The implementation of the skeletonization algorithms and the recognition system has been carried out in C++. All algorithms studied in this paper are very fast and suitable for real-time applications, so the execution time was not considered in the evaluation. The average execution time for the skeletonization of an input image of size 256×256 is tens of milliseconds running on an AMD Athlon XP 2800+ processor.

5. CONCLUSION

A goal-directed evaluation of four skeletonization methods, namely the SPTA, Zhang-Suen's, the proposed method and Huang et al.'s algorithm for recognizing Arabic/Farsi handwritten words was presented. A summary of each algorithm was given. A simple and effective skeleton postprocessing procedure was also proposed. It was shown that Zhang-Suen's algorithm is not suitable for recognition purposes. Of the other three algorithms, Huang et al.'s was shown to be the robustest with respect to border noise as it produces skeletons with the smallest number of spurious branches. But overall, the proposed algorithm with the addition of the postprocessing step resulted in highest recognition rates.

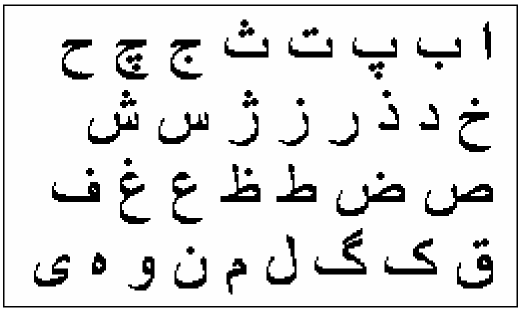
6. REFERENCES

[1] Ivanov, D., Kuzmin E., and Burtsev, S. An Efficient Integer-Based Skeletonization Algorithm. *Computers and Graphics*, vol. 24 (2000), 41-51.

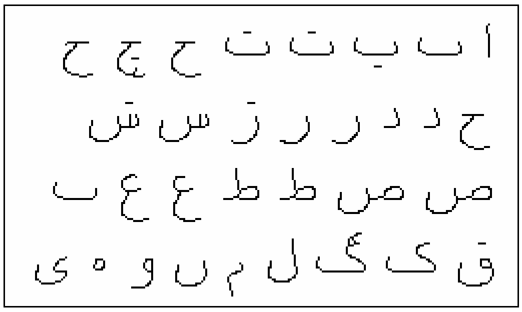
[2] Ahmed, P. A Neural Network Based Dedicated Thinning Method. *Pattern Recognition Letters*, vol. 16 (1995), 585-590.

[3] Naccache, N. J., and Shinghal, R. SPTA: A Proposed Algorithm for Digital Pictures. *IEEE Trans. on Systems, Man and Cybernetics*, vol. SMC-14(3) (1984), 409-418.

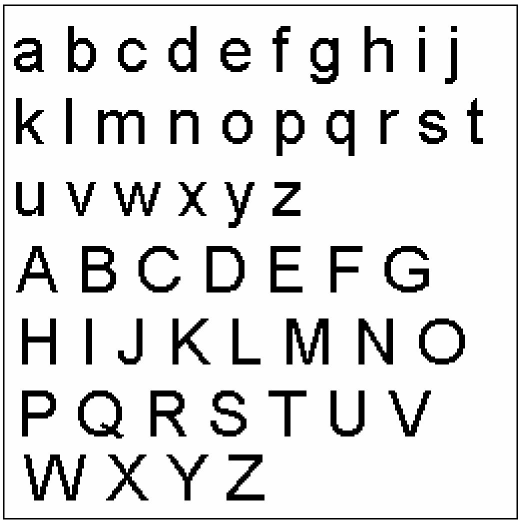
[4] Zhang, T. Y., and Suen, C. Y. A Fast Parallel Algorithm for Thinning Digital Patterns. *Comm. ACM*, vol. 27(3) (1984), 236-239.



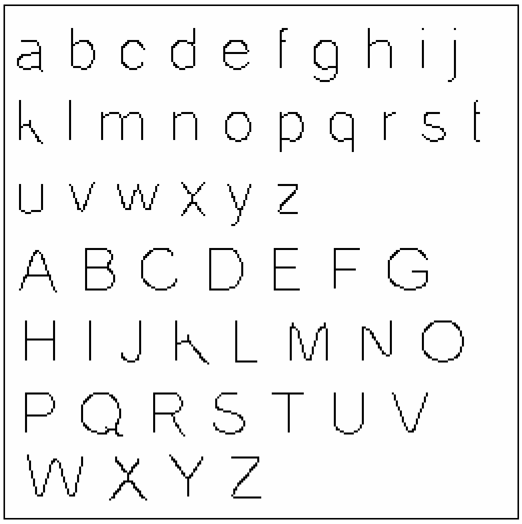
(a) Farsi character set, as input image



(b) Skeleton obtained using Zhang-Suen's algorithm



(c) English character set, as input image



(d) Skeleton obtained using Zhang-Suen's algorithm

Figure 6. Applying Zhang-Suen's algorithm to English and Farsi character set.

[5] Lam, L., Lee, S. W., and Suen, C. Y. Thinning Methodologies: A Comprehensive Survey. *IEEE Trans. on PAMI*, vol. 14(9) (September 1992), 869-885.

[6] Suen, C. Y., and Wang, P. S. P. *Thinning Methodologies for Pattern Recognition*. Series in Machine Perception and Artificial Intelligence, vol. 8, World Scientific, 1994.

[7] Huang, L., Wan, G. and Liu, C. An Improved Parallel Thinning Algorithm. *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)* (2003), 780-783.

[8] Liu, Y., and Srihari, S. N. Document Image Binarization Based on Texture Features. *IEEE Trans. on PAMI*, vol. 19(5) (May 1997), 540-544.

[9] Dyer, C. R., and Rosenfeld, A. Thinning Algorithms for Grey-Scale Pictures. *IEEE Trans. on PAMI*, vol. 1 (1979) 88-89.

[10] Khorsheed, M. *Automatic Recognition of Words in Arabic Manuscripts*. Ph.D. Thesis, Churchill College, University of Cambridge, June 2000.

[11] Sajjadi, M. R. *Skeletonization of Persian Characters*. M.Sc. Thesis, Computer Science and Engineering Department, Shiraz University, Shiraz, Iran, Oct. 1996.

[12] Haji, M. M. *Farsi Handwritten Word Recognition Using Continuous Hidden Markov Models and Structural Features*, M.Sc. Thesis, Computer Science and Engineering Department, Shiraz University, Shiraz, Iran, Jan. 2005.