Feature Extraction – Arabic OCR Case Study

Gheith A. Abandah

Feature extraction is extracting from the raw data the information which is most relevant for classification purposes.

The following subsections describe the techniques and algorithms used to extract an assortment of features used in this research.

- We start by detecting the secondary components of the Arabic letters and extracting features from these components.
- Then we remove the secondary components and extract additional features from
    - the main body,
    - the main body’s skeleton, and
    - the main body’s boundary.

3.1 Secondary Components Detection and Removal

More than half the Arabic letters are composed of *main body* and *secondary components*. The secondary components are letter components that are disconnected from the main body. For example, **Beh** (ب) has a dot under its main body, **Teh** (ت) has two dots above its main body, and **Kaf** (ك) has a zigzag enclosed within the main body. Table 2 lists the secondary types that are encountered in written Arabic samples.

Table 2 Types of the Secondary Components

No	Secondary Type	Examples
1	No Secondary	ا و ه م ل ع ط ص س ر د ح ء
2	One Dot	ن ف غ ظ ض ز ذ خ ج ب
3	Two Dots	ي ق ة ت
4	Three Dots	ش ث
5	Zigzag	ك إ ؤ أ
6	Vertical Bar [a]	ط
7	Vertical bar and a dot [a]	ظ
8	Long Stroke [b]	ک

[a] This secondary is encountered when the upper vertical stroke is drawn disconnected from the loop of **Tah** and **Zah**.

[b] This secondary is encountered when the upper stroke is drawn disconnected from the lower part of initial **Kaf**.

The type of the secondary components and their position are very important features for recognizing Arabic letters. For example, two dots below the main body are sufficient to recognize the letter **Yeh** (ي) because **Yeh** is the only letter with these features. Furthermore, some letter forms can only be distinguished by the type of secondary components as in medial **Teh** and medial **Theh** (ﺘ ﺜ), or the secondary position as in medial **Teh** and medial **Yeh** (ﺘ ﻴ). Table 3 shows the possible secondary components positions of Arabic letters.

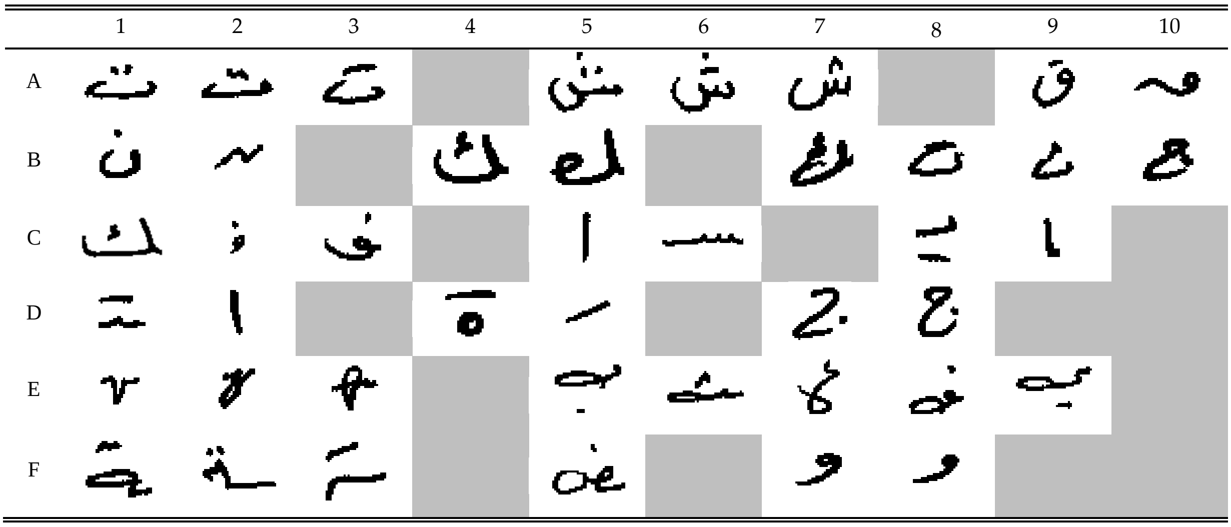
Table 3 Possible Positions of Secondary Components

No	Secondary Position	Examples
1	No Secondary	ا و ه م ل ع ط ص س ر د ح ء
2	Above	ن ق ف غ ض ش ز ذ خ ث ة ت ؤ أ
3	Within	ك ظ ج
4	Below	ي ب إ

Detecting the secondary components can be done after segmenting the binary image of the letter into its disconnected components using the *connected component labeling* techniques [12, 13]. Then the main body is easily identified as it is usually the largest component and is closer to the baseline than the secondary components. The position of the secondary components is then easily found relative to the main body. Finally, the number and position of the secondary components play important role in finding their type. However, our approach in classifying the secondary components also utilizes other features extracted from the secondary components such as size, orientation, elongation, and spatial distribution (see Section 3.2.).

There are important variations in drawing the secondary components; mostly in drawing two dots and three dots. As shown in Table 4-Samples A1, A2, and A3, the two dots come in three variations: two disconnected dots, two connected dots, and horizontal dash. Samples A5, A6, and A7 show three variations in drawing the three dots: three disconnected dots, one dot above horizontal dash, and inverted “v” shape. The secondary components classification process should take these variations into consideration.

Table 4 Samples Showing Variations in Handwritten Letters


1
2
3
4
5
6
7
8
9
10
A
B
C
D
E
F

It is important to note that some writers use styles that replace the secondary components of isolated and final forms with main body curves. Table 4 shows some examples: Samples A9 and A10 show how the two dots of isolated **Qaf** are replaced, Samples B1 and B2 show how the one dot of isolated **Noon** is replaced, and Samples B4 and B5 show how the zigzag of final **Kaf** is replaced.

One difficulty in recognizing the secondary components comes when hasty writers draw them connected to the main body. For example, Sample B7 shows the zigzag connected to **Kaf**'s body, Sample B8 shows the two dots connected to **Teh**'s body, Sample B9 shows the three dots connected to **Theh**'s body, and Sample B10 shows the dot connected to **Jeem**'s body.

After detecting and classifying the secondary components, we remove them from the letter image and pass the main body to the feature extraction stages described below.

3.2 Main Body Features

Main body features are mainly statistical features. They are found from the letter image after removing the secondary components. The following paragraphs define some of these features, such as area, width, height, pixel distribution, moments, orientation, roundness, and number of loops.

3.2.1 Size

We use a threshold function to convert the 2-dimentional image into a binary image $B(x,y) \in (0,1)$; black pixels are the foreground pixels and take the value 1 [12]. The *area A* of the letter body is found by

$$A = \sum_x \sum_y B(x,y) . \qquad (1)$$

To find the main body's width *W* and height *H*, the image is clipped into a rectangular shape such that all four borders have at least one black pixel. We also derive a scale-invariant feature; the width to height ratio *W/H* [14].

3.2.2 Distribution

We partition the clipped image into four equal quadrants and find the fraction of black pixels in each quadrant relative to the area *A*. The resulting four fractions are: upper-right *UR/A*, lower-right *LR/A*, lower-left *LL/A*, and upper-left *UL/A*. We also find the fractions of the four halves relative to *A*: upper *U/A*, right *R/A*, lower *Lo/A*, and left *Lt/A*.

3.2 Moments

The *moments* of order $(u+v)$ of the binary image [15, 16] can be found by

$$m_{uv} = \sum_x \sum_y x^u y^v B(x,y) \qquad u,v = 0,1,2,3,\ldots \qquad (2)$$

Note that *m₀₀* is the body's area *A*, and the image's center of mass $(\bar{x}, \bar{y})$ is found from

$$\bar{x} = \frac{m_{10}}{m_{00}} \text{ and } \bar{y} = \frac{m_{01}}{m_{00}}. \qquad (3)$$

The center of mass is dependant on the origin selection and the body's scale. In order to normalize for these two factors, we compute the normalized center of mass $(\bar{x}_N, \bar{y}_N)$ using

$$\bar{x}_N = \frac{\bar{x} - (W-1)/2}{W/2} \text{ and } \bar{y}_N = \frac{\bar{y} - (H-1)/2}{H/2}. \qquad (4)$$

The *central moments,* which are translation invariant, can be found by

$$\mu_{uv} = \sum_x \sum_y (x - \bar{x})^u (y - \bar{y})^v B(x,y). \qquad (5)$$

Finally, the *normalized central moments*, which are translation and scale invariant, are derived from the central moments as follows

$$\eta_{uv} = \frac{\mu_{uv}}{(\mu_{00})^k}, \qquad (6)$$

where $k = 1 + (u+v)/2$ for $u + v \geq 2$.

Hu has defined a set of seven moments which are invariant under the actions of translation, scaling, and rotation [17]. Using the feature evaluation techniques described in Section 5, we found that the normalized central moments η_{uv} give better results than Hu's moment invariants, suggesting that the added computational overhead of Hu's moment invariants does not give added value for our samples that do not have rotational variations.

3.2.4 Orientation

The *orientation* θ of an elongated object is the orientation of the elongation axis [12]. The axis of least inertia is the elongation axis. The inertia of the elongation axis is found by

$$\chi^2 = \sum_x \sum_y r^2 B(x,y), \qquad (7)$$

where *r* is the perpendicular distance from point (*x*, *y*) to the elongation axis. Using polar coordinates and utilizing the fact that the elongation axis passes through the center of mass, the inertia is found from the second-order central moments by

$$\chi^2 = \frac{1}{2}(\mu_{20} + \mu_{02}) - \frac{1}{2}(\mu_{20} - \mu_{02})\cos 2\theta - \mu_{11}\sin 2\theta . \qquad (8)$$

The orientation of the elongation axis can be found by solving the minimization problem of (8) with respect to θ. The orientation θ then can be found by solving

$$\sin 2\theta = \pm \frac{2\mu_{11}}{\sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}} \text{ and} \qquad (9)$$

$$\cos 2\theta = \pm \frac{(\mu_{20} - \mu_{02})}{\sqrt{4\mu_{11}^2 + (\mu_{20} - \mu_{02})^2}} . \qquad (10)$$

3.2.5 Roundness

The positive and negative values for sine and cosine of 2θ in (9) and (10) can be plugged in (8) to find the minimum and maximum inertia values, respectively. The object *elongation E* (or eccentricity) is found by

$$E = \frac{\chi_{\max}}{\chi_{\min}} . \qquad (11)$$

The object *roundness R*, defined using (12), is a ratio between 0 for a straight line and 1 for a circle.

$$R = \frac{\chi_{\min}^2}{\chi_{\max}^2} \qquad (12)$$

Table 5 shows the averages of the some of the statistical features described above. These averages are found for the features extracted from handwritten letter samples of the four forms. These samples are described in Section 4. The first three averages indicate that final and isolated forms are larger than initial and medial forms. Samples C1 and C2 of Table 4 show two extremes; the final **Kaf** is much larger than the initial **Feh**. Moreover, Samples C2 and C3 show that the initial and final forms of **Feh** have totally different sizes.

Table 5 Average Values of Some Statistical Features for the Four Letter Forms

No	Feature	Isolated	Initial	Medial	Final
1	Area *A* (in pixels)	145	116	120	173
2	Width *W* (in pixels)	23	20	22	29
3	Height *H* (in pixels)	19	15	13	18
4	Ratio *W/H*	1.40	1.51	2.09	1.75
5	*UR/A*	0.28	0.31	0.22	0.23
6	*LR/A*	0.24	0.26	0.29	0.24
7	*LL/A*	0.33	0.32	0.32	0.34
8	*UL/A*	0.15	0.11	0.17	0.19
9	*U/A*	0.43	0.43	0.39	0.42
10	*R/A*	0.52	0.57	0.51	0.47
11	*Lo/A*	0.57	0.57	0.61	0.58
12	*Lt/A*	0.48	0.43	0.49	0.53
13	$\bar{x}_N$	0.02	0.09	0.01	-0.05
14	$\bar{y}_N$	-0.09	-0.08	-0.12	-0.11
15	Orientation θ	37°	34°	22°	27°
16	Roundness *R*	0.24	0.23	0.25	0.22

From the width and height averages, we can conclude that Arabic letters are generally elongated in the horizontal direction. Also note that the ratio *W/H* of medial and final forms is larger than that of isolated and initial forms. Sample C5 shows isolated **Alef**, which has small *W/H* ratio. And Sample C6 shows the medial **Seen**, which has large *W/H* ratio.

By studying the averages of fractions of pixel distributions and the normalized center of mass, we can reach some interesting conclusions about the characteristics of handwritten Arabic letters. In general, Arabic letters have more mass in the lower half of the clipped letter image. However, on average initial forms have more mass in the right half, and final forms have more mass in the left half. Sample C8 shows initial **Yeh** that demonstrates an example of large relative mass in the right half, and Sample C9 shows final **Alef** that demonstrates an example of large relative mass in the left half. Both these samples have most of their respective masses in the lower half.

In general, the Arabic letters go from right to left and up to down. The average orientation is 30°. However, the four forms have different orientation averages. The medial form’s average is the closest to the horizontal direction and the isolated form’s average is the farthest. Sample D1 shows medial **Teh** which has a small orientation angle and Sample D2 shows isolated **Alef** which has a large orientation angle.

Finally, Table 5 shows that the average Arabic letter is far from the rounded shape. However, Sample D4 shows the isolated **Teh** (closed form) which is the closest form to perfect circle. On the other hand, Sample D5 shows isolated **Reh** which is almost a straight line.

3.2.6 Loops

The number of main body loops is a structural feature. While more than half the Arabic letters are usually written without loops (see Table 6), ten other letters are usually written with one loop in all four forms, two letters are written with one loop in the medial and final forms only, and three letters are written with or without a loop according to the writing style. For example, **Jeem** (ج) is written without a loop and with a loop as shown in Samples D7 and D8, respectively.

Table 6 Existence of Loops in Arabic Letters

No	Loop Existence	Examples
1	No loops	ي ا ن ل ك ش س ز ر ذ د ث ت ب ء
2	One loop in all forms	و ه م ق ف ظ ط ض ص ة
3	One loop in some forms	ـغـ ـغـ
4	One loop in some styles	ح خ ج

Medial **Heh** has large style variation; Samples E1, E2, and E3 show that this form has styles with no loops, one loop, and two loops, respectively. Moreover, some writing styles introduce additional loops to the isolated and final forms by extending the curve of the letter's end. Examples are Letters **Beh** (Sample E5), **Teh**, **Theh** (E6), **Ain** (E7), **Ghain**, **Feh** (E8), **Qaf**, **Kaf** (B4), **Noon**, curly **Alef**, and **Yeh** (E9). Some writers don't close the loop of the final forms of closed **Teh** and **Heh**, as illustrated in Samples F1-F3.

We have noticed that some samples of the isolated and final forms of the letters that have a rounded cusp have unexpected loops when the cusp is drawn completely closed. We have noticed this observation with some samples of Letters **Seen**, **Sheen**, **Sad**, **Dad** (see Sample F5), and **Noon**. Also we have noticed that many samples of letters that have a small loop are drawn with a filled loop that is hard to discover. This was frequently noticed with samples of Letters **Feh**, **Qaf**, **Meem**, and **Waw**. Samples F7 and F8 show how the **Waw** loop is drawn punctured and filled, respectively. Note also that Sample E8 shows final **Feh** drawn with a filled loop.

There are many techniques to find the number of loops in an image. We used the connected component labeling algorithm to find the number of loops. The number of background components (white components) minus one is the number of loops.

For example, Sample D8 has one loop since is has two background components; the large background component surrounding the letter (always present) and the small component enclosed within the loop of the upper part of the letter.

3.3 Skeleton Features

Thinning is usually a pre-processing stage in character recognition where the character image is reduced to a simplified one-pixel wide skeleton. Fig. 1 shows an isolated **Seen** before thinning and the resulting skeleton after thinning. The skeleton allows extracting a variety of character features as described below.

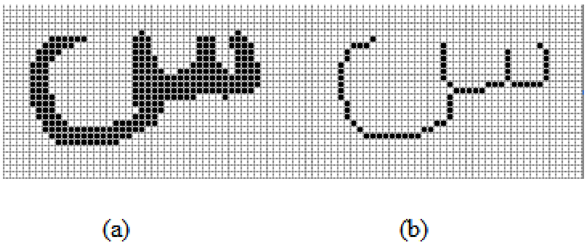
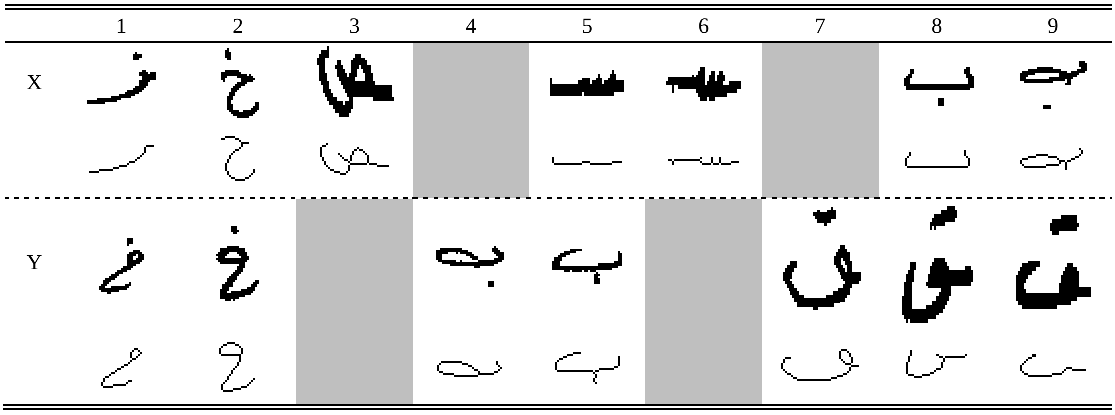


Fig. 1 An isolated Seen (a) before thinning and (b) its skeleton after thinning.

There are many serial and parallel algorithms for thinning character images [18, 19]. We have found that the simplified version of Rutovitz's thinning algorithm [20] as described by Stefanelli and Rosenfeld [21] generates good skeletons for our samples. However, this algorithm has a problem with diagonal lines where it sometimes generates two-pixel wide diagonal lines. We therefore adopted Deutsch's thinning algorithm [22] that has a complete set of rules to solve the diagonal lines problem and to get symmetric thinning. Table 7 shows some sample letters and the respective main body skeletons using this algorithm.

Table 7 Letter Samples and Respective Skeletons


1
2
3
4
5
6
7
8
9
X
Y

We used the skeleton of the main letter's body to extract five features. These features are the numbers of vertical and horizontal crossings and the feature points as described below.

3.3.1 Vertical and Horizontal Crossings

The *vertical* and *horizontal crossings* are found by counting the number of white-black-white transfers when scanning the image's pixels on a vertical line and a horizontal line, respectively. These lines are the two lines that pass through the center of mass of the main body's skeleton. These features are signs of the letter's complexity. For example, Samples X1, X2, and X3 in Table 7 show the simple final **Zain** that has one vertical and one horizontal crossing, isolated **Khah** that has three vertical crossings and one horizontal crossing, and the complex final **Sad** that has two vertical crossings and four horizontal crossings.

Elongated letters have large variance in the number of crossings in the elongation direction. For example, Samples X5 and X6 of the medial Seen, which is horizontally elongated, have one vertical crossing and two and five horizontal crossings, respectively. These two samples illustrate another problem; Seen has three small teeth that are often lost through the thinning process.

Decorative loops in the isolated and final forms increase the number of crossings. Samples X8 and X9 illustrate that the vertical crossings of isolated **Beh** increase from one to two when this letter is written with a decorative loop. Also handwriting variations introduce variance in the number of crossings. Samples Y1 and Y2 show two more samples of the isolated **Khah**; Sample Y1 has two vertical crossings because it is written with the loop shifted to the back, and Sample Y2 has four vertical crossings because it is written with the loop hanging to the front.

3.3.2 Feature Points

Three important feature points can be easily found from the skeleton by examining the eight immediate neighbors of every black pixel: *end point* is a point with one black neighbor, *branch point* has three black neighbors, and *cross point* has four black neighbors. Fig. 2 shows the skeleton of isolated **Sad** that demonstrates three end points, one branch point, and one cross point.

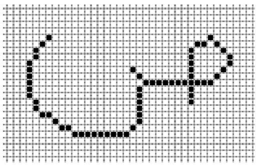


Fig. 2 The Skeleton of isolated Sad has three end points, one branch point, and one cross point.

The number of feature points is affected when decorative loops are added to the isolated and final forms. Although isolated **Beh** has only two end points as illustrated by Sample X8, adding a decorative loop adds a cross point, or eliminates an end point and adds a branch point as illustrated by Samples X9 and Y4, respectively.

The number of feature points is also affected when the secondary objects touch the main body. Sample Y5 shows an isolated **Beh** with its dot touching the main body. As a result, the main body of isolated **Beh** gets one more end point and one branch point.

Variations in drawing loops also affect the number of feature points. Samples Y7 and Y8 show two final **Qaf** letters with punctured and filled loops, respectively. The punctured loop feature gives one cross point, whereas the filled loop gives one branch point and one end point. However the thinning process may dissolve the filled loop completely and end up with no feature points as illustrated in Sample Y9.

Moreover, the thinning process may remove the teeth of **Seen**, **Sheen**, **Sad**, and **Dad**, as illustrated in Sample X5. The removal of every tooth eliminates one branch point and one end point.

Table 8 shows the averages of the features extracted from the skeleton for the four letter forms. The averages of the medial and final forms are larger than the averages of the isolated and initial forms, which is an indication that medial and final forms are more complex. Note that the averages of the number of end points is around two or larger. Simple letters have two ends unless one end is a loop as in isolated **Waw** (و). The complex forms have more end points, branch points, and cross points.

Table 8 Average Values of the Skeleton Features for the Four Letter Forms

No	Feature	Isolated	Initial	Medial	Final
1	Vertical Crossings	1.66	1.55	1.68	1.58
2	Horizontal Crossings	1.75	1.59	1.84	1.91
3	End Points	1.96	2.00	2.47	2.41
4	Branch Points	0.71	0.88	1.20	0.99
5	Cross Points	0.06	0.05	0.08	0.07

We noticed that the number of cross points is smaller than the expected number. For example, we expected that the cross point feature would be found in 6 medial forms out of 23 (averaging 0.26). But the extracted average was only 0.08. The reason is that cross points are often lost through the thinning process and are converted to pairs of neighboring branch points as Fig. 3 illustrates. Here the main body of medial **Ain** has one perceptible cross point at the base of the loop. But the thinning process converts the cross into two pixels that are two adjacent branch points.

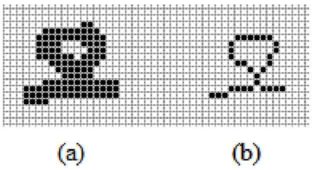


Fig. 3 Medial **Ain**: (a) Main Body, (b) Skeleton after Thinning

3.4 Boundary Features

Boundary finding is another pre-processing stage in character recognition where the character outer contour is found [12]. Fig. 4 shows isolated **Seen** before and after finding its boundary.

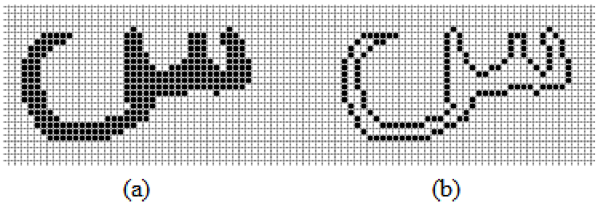
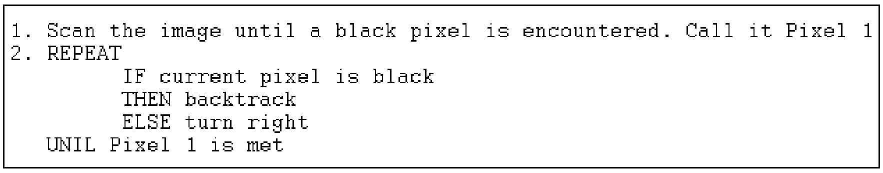


Fig. 4 Isolated **Seen** (a) Original Main Body and (b) Its Boundary

Fig. 5 shows the algorithm we used for finding the boundary [23]. This algorithm first finds one boundary pixel, then it traces the boundary pixels in a clockwise fashion until it gets back to the first boundary pixel.



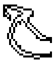
```
1. Scan the image until a black pixel is encountered. Call it Pixel 1
2. REPEAT
         IF current pixel is black
         THEN backtrack
         ELSE turn right
   UNIL Pixel 1 is met
```

Fig. 5 Boundary Tracing Algorithm

Table 9 shows some sample letters and the respective main body boundary using this algorithm.

We used the boundary of the main letter's body to extract several features. These features are the number of boundary pixels, perimeter length, perimeter to diagonal ratio, bending energy, compactness ratio, and elliptic Fourier descriptors as described below.

Table 9 Letter Samples and Respective Boundaries

	1	2	3	4	5	6	7	8	9
Z									
									

3.4.1 Boundary Pixels

The number of *boundary pixels m* is directly found by counting the boundary pixels $(x_i, y_i), i = 1, 2, \ldots, m$ traced using the above algorithm. Then Freeman chain code is used to compactly encode the boundary pixels [24]. The direction from every boundary pixel to the next boundary pixel is put in the chain. The direction from the last pixel to the first pixel is the last code in the chain. The direction codes $f_i \in [0, 7]$ are as shown in Fig. 6.

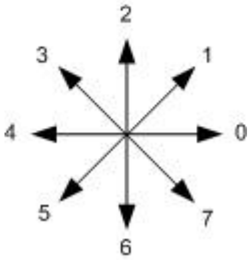


Fig. 6 Freeman Chain Code for the Eight Directions from One Boundary Pixel to the Next

3.4.2 Perimeter Length

The *perimeter length T* is found by summing the distances from one pixel to the next. Formally, it is found from the chain code using

$$T = \sum_{i=1}^{m} \mathrm{L}(f_i), \quad \text{where} \quad \mathrm{L}(f_i) = \begin{cases} 1 & f_\mathrm{i} \text{ is even} \\ \sqrt{2} & f_\mathrm{i} \text{ is odd} \end{cases}. \quad (13)$$

3.4.3 Perimeter to Diagonal Ratio

We also used a scale-invariant feature which is the ratio of half the perimeter length to the diagonal of the clipped main body rectangle *T/2D*. For simple shapes like **Alef**, this ratio is 1, and this ratio is larger than 1 for more complex shapes.

$$T/2D = \frac{T/2}{\sqrt{W^2 + H^2}} \qquad (14)$$

Sample Z1 in Table 9 shows isolated **Reh** that has small *T/2D* ratio. Sample Z2 shows final **Khah** that has large *T/2D* ratio.

3.4.4 Compactness Ratio

Another derived feature from the perimeter length and the area is the *compactness ratio* or roundness ratio which is found by (15) [15].

$$\gamma = \frac{T^2}{4\pi A} \qquad (15)$$

This ratio is 1 for a filled circle and is larger than 1 for distributed complex shapes. Samples Z4 and Z5 show isolated **Teh** and final **Sheen**, which are two extreme examples of small and large compactness ratios, respectively.

3.4.5 Bending Energy

The *bending energy E* is a measure of how curly the boundary curve is [15]. It can be found from the chain code by summing the squares of the direction changes from one boundary pixel to the next.

$$E = \frac{1}{T} \sum_{i=1}^{m} \left(\frac{\pi}{4} \right)^2 \left(\mathrm{IF}(k_i > 4, 8 - k_i, k_i) \right)^2 , \qquad (16)$$

$$\text{where}\quad k_i = \begin{cases} \operatorname{mod}(f_{i+1} - f_i, 8) & i < m \\ \operatorname{mod}(f_1 - f_m, 8) & i = m \end{cases}. \qquad (17)$$

Small rounded shapes tend to have large bending energy factor. One example is the initial **Feh** shown in Sample Z7. As the isolated **Ain** shown in Sample Z8 has rounded and coarse boundary, it also has a relatively large bending energy. The isolated Lam shown in Sample Z9 is an example large letter that has smooth boundary and low pending energy.

Table 10 shows the averages of the above five features that are extracted from the boundary for the four letter forms. The averages of the number of boundary pixels and the perimeter length indicate that the final and isolated forms are larger than medial and initial forms.

Table 10 Average Values of the Boundary Features for the Four Letter Forms

No	Feature	Isolated	Initial	Medial	Final
1	Boundary Pixels	79	56	63	92
2	Perimeter Length	90	63	71	105
3	Perimeter to Diagonal Ratio	1.5	1.2	1.3	1.5
4	Compactness Ratio	4.6	2.9	3.4	5.2
5	Bending Energy	0.41	0.47	0.49	0.42

The averages of *T/2D* and γ indicate that the final and isolated forms are more complex and spread than the medial and initial forms. Finally, the averages of the bending energy indicate that the medial and initial forms have slightly more curly boundaries than the final and isolated forms.

3.4.6 Elliptic Fourier Descriptors

The piecewise linear curve that passes through all boundary pixels is a closed outer contour curve. This curve passes through the points $\left(x(t), y(t)\right), t = 1, 2, \ldots, m$ and can be approximated using the *elliptic Fourier descriptors* (EFD) of Kuhl and Giardina [25]. These descriptors are useful features [14, 26] and are used to approximate the curve as follows

$$\hat{x}(t) = A_0 + \sum_{n=1}^{N} \left[a_n \cos \frac{2n\pi t}{T} + b_n \sin \frac{2n\pi t}{T} \right], \qquad (18)$$

$$\hat{y}(t) = C_0 + \sum_{n=1}^{N} \left[c_n \cos \frac{2n\pi t}{T} + d_n \sin \frac{2n\pi t}{T} \right], \qquad (19)$$

where T is the perimeter length and $\left(\hat{x}(t), \hat{y}(t)\right) \equiv \left(x(t), y(t)\right)$ in the limit when $N \to \infty$. These descriptors are found by

$$A_0 = \frac{1}{T}\int_0^T x(t)\,\mathrm{d}t \qquad (20)$$

$$C_0 = \frac{1}{T}\int_0^T y(t)\,\mathrm{d}t \qquad (21)$$

$$a_n = \frac{2}{T}\int_0^T x(t)\cos\frac{2n\pi t}{T}\,\mathrm{d}t \qquad (22)$$

$$b_n = \frac{2}{T}\int_0^T x(t)\sin\frac{2n\pi t}{T}\,\mathrm{d}t \qquad (23)$$

$$c_n = \frac{2}{T}\int_0^T y(t)\cos\frac{2n\pi t}{T}\,\mathrm{d}t \qquad (24)$$

$$d_n = \frac{2}{T}\int_0^T y(t)\sin\frac{2n\pi t}{T}\,\mathrm{d}t \qquad (25)$$

Since the functions $x(t)$ and $y(t)$ are piecewise linear, then the discrete evaluation of these descriptors is

$$a_n = \frac{T}{2n^2\pi^2}\sum_{i=1}^{m}\frac{\Delta x_i}{\Delta t_i}\left[\cos\phi_i - \cos\phi_{i-1}\right] \qquad (26)$$

$$b_n = \frac{T}{2n^2\pi^2}\sum_{i=1}^{m}\frac{\Delta x_i}{\Delta t_i}\left[\sin\phi_i - \sin\phi_{i-1}\right] \qquad (27)$$

$$c_n = \frac{T}{2n^2\pi^2}\sum_{i=1}^{m}\frac{\Delta y_i}{\Delta t_i}\left[\cos\phi_i - \cos\phi_{i-1}\right] \qquad (28)$$

$$d_n = \frac{T}{2n^2\pi^2}\sum_{i=1}^{m}\frac{\Delta y_i}{\Delta t_i}\left[\sin\phi_i - \sin\phi_{i-1}\right] \qquad (29)$$

where

$$\phi_i = 2n\pi t_i / T,$$
$$\Delta x_i = x_i - x_{i-1}, \qquad \Delta y_i = y_i - y_{i-1}, \qquad (30)$$
$$\Delta t_i = \sqrt{\Delta x_i^2 + \Delta y_i^2}, \qquad t_i = \sum_{j=1}^{i} \Delta t_j,$$
$$T = t_m = \sum_{j=1}^{m} \Delta t_j,$$

These descriptors can be normalized for phase and rotation [14]. However, the feature evaluation described in Section 5 proved that the raw descriptors give better results than the normalized descriptors, suggesting that the added computational overhead of the normalization does not give added value. Particularly because our samples are not rotated and the boundary start pixel is always found by scanning the images from the same direction.

References

[1] Mori S., Nishida H., Yamada H.: Optical character recognition. Wiley, New York (1999)

[2] Khorsheed M.: Off-line Arabic character recognition – a review. Pattern Anal. & Applications, 5(1), 31-45 (2002)

[3] Plamondon R., Srihari S.: On-line and off-line handwriting recognition: a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. 22(1), 63-84 (2000)

[4] Al-Emami S., Usher M.: On-line recognition of handwritten Arabic characters. IEEE Trans. Pattern Anal. Mach. Intell. 12(7), 704-710 (1990)

[5] Arica N., Yarman-Vural F.: Optical character recognition for cursive handwriting. IEEE Trans. Pattern Anal. Mach. Intell. 24(6), 801-813 (2002)

[6] Lorigo L., Govindaraju V.: Offline Arabic handwriting recognition: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 28(5), 712-724 (2006)

[7] Pechwitz M., Snoussi Maddouri S., Märgner V., Ellouze N., Amiri H.: IFN/ENIT-database of handwritten Arabic words. In: Proc. 7th Collque Int'l Francophone sur l'Ecrit et le Document (CIFED 2002), Hammamet, Tunis, pp. 129–136 (2002)

[8] Märgner V., Pechwitz M., ElAbed H.: ICDAR 2005 Arabic handwriting recognition competition. Proc. Int'l Conf. Document Anal. and Recogn., pp. 70-74 (2005)

[9] Amin A.: Arabic character recognition. In Bunke H., Wang P. (eds) Handbook of character recognition and document image analysis, World Scientific, pp. 397-420 (1997)

[10] Abandah G., Khundakjie F.: Issues concerning code system for Arabic letters. Dirasat Eng. Sci. J. 31(1), 165-177 (2004)

[11] The Unicode Consortium: The Unicode Standard. v. 1.0, Addison-Wesley, Reading, MA, USA (1992)

[12] Jain R., Kasturi R., Schunck B.: Machine vision. MacGraw-Hill, New York (1995)

[13] Rosenfeld A., Kak A.C.: Digital picture processing. Academic Press (1976)

[14] Trier O., Jain A., Taxt T.: Feature extraction methods for character recognition - a survey. Pattern Recogn. 29(4), 641-662 (1996)

[15] Theodoridis S., Koutroumbas K.: Pattern recognition. 3rd ed., Academic Press (2006)

[16] Reiss T. H.: The revised fundamental theorem of moment invariants. IEEE Trans. Pattern Anal. Mach. Intell. 13(8), 830-834 (1991)

[17] Hu M.: Visual pattern recognition by moment invariants. IRE Trans. Inf. Theory, 8(2), 179-187 (1962)

[18] Lam L., Lee S-W., Suen C.: Thinning methodologies - a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. 14(9), 869-885 (1992)

[19] Lam L., Suen C.: An evaluation of parallel thinning algorithms for character recognition. IEEE Trans. Pattern Anal. Mach. Intell. 17(9), 914-919 (1995)

[20] Rutovitz D.: Pattern recognition. J. Royal Statist. Soc. 129, Series A, 504-530 (1966)

[21] Stefanelli R., Rosenfeld A.: Some parallel thinning algorithms for digital pictures. J. of ACM, 18(2), pp. 255-264 (1971)

[22] Deutsch E.: Thinning algorithms on rectangular, hexagonal, and triangular arrays. Comm. of the ACM, 15(9), 827-837 (1972)

[23] Ha T., Bunke H.: Image processing methods for document image analysis. In: Bunke H., Wang P. (eds) Handbook of character recognition and document image analysis, World Scientific, pp. 1-47 (1997)

[24] Freeman H.: On the encoding of arbitrary geometric configurations. IRE Trans. Electronic Computers, 10(2), 260-268 (1961)

[25] Kuhl F., Giardina C.: Elliptic Fourier features of a closed contour. Computer Graph. and Image Process. 18(3), 236-258 (1982)

[26] Mezghani N., Mitiche A., Cheriet M.: On-line recognition of hand-written Arabic characters using a Kohonen neural network. In: Proc. 8th Int'l Workshop on Frontiers in Handwriting Recogn. pp. 490-495 (2002)

International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), 2002