3

Recognition of Tifinaghe Characters Using Dynamic Programming & Neural Network

Rachid El Ayachi, Mohamed Fakir and Belaid Bouikhalene
Sultan Moulay Slimane University/
Faculty of Sciences and Techniques
Morocco

1. Introduction

Optical Character Recognition (OCR) is one of the most successful applications of automatic pattern recognition. The field of characters recognition is very important. Several studies have been conducted on Latin, Arabic and Chinese characters (Bozinovic and Shihari, 1989; Brown, 1983; Fakir and Sodeyama, 1993; Fakir, 2001; Chaudhuri and al, 2002; Blumenstein and al, 2002; Miyazaki and al, 1974; Mezghani and al, 2008; Lallican and al, 2000; Burr, 1982) and various commercial applications have been produced such as bank cheque processing, postal automation, documents. However, for Amazigh characters, called Tifinaghe, few studies have been published in the literature. Among these researches, we find (El ayachi and Fakir, 2009; Amrouch et al, 2009; Es saady, 2009; Fakir et al, 2009; El ayachi et al, 2010).

Because, characters are sensitive to noise the main problem in this field how to extracts strokes. This may be solved by the selection of the useful features customarily defined in the automatic character recognition as two types: global and local features. The principle of global features is based on the transformation of the character matrix into a new domain to extract features. The selection of local features is based on geometrical and topological properties of the character, such as strokes direction, strokes density, strokes length and position, etc.

Unlike Latin characters, Tifinaghe characters are formed by loops, lines and curves. This makes it difficult to describe a character in one parametric form. In this study, invariant moments, modified invariant moments and Walsh transform are used as features for the recognition of Tifinaghe characters. Fig.1 illustrates the block diagram of the proposed recognition system. Tifinaghe texts were transferred to the computer through an image scanner.

The process consists of three phases. After preliminary pre-processing of position normalization, noise reduction and skew correction), a text is segmented into lines and lines into what to be characters in the second phase. In the third phase features extraction methods are applied. In the last phase the recognition procedure is completed. In this phase a Multilayer Neural Network and Dynamic Programming Technique are used to classifier characters. These phases are described in the following sections, but before that a brief explanation about the characteristics of Tifinaghe characters is given.

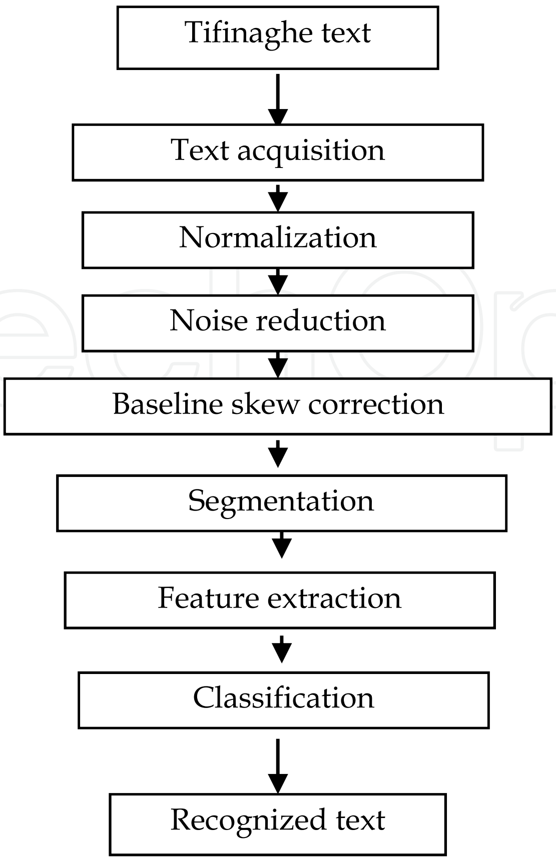


Fig. 1. Block diagram of the recognition system

2. Tifinaghe characters

The Tifinaghe script is used by approximately 20 million people who speak varieties of languages commonly called Berber or Amazigh. The three main varieties in Morocco are known as Tarifite, Tamazighe, and Tachelhite (El ayachi and Fakir, 2009; Es saady, 2009).

In Morocco, more than 40% of the population speaks Berber. In accordance with recent governmental decisions, the teaching of the Berber language, written in the Tifinaghe script, will be generalized and compulsory in Tifinaghe is an alphabetic writing system. It uses spaces to separate words and makes use of Western punctuation.

The earliest variety of the Berber alphabet is Libyan. Two forms exist: a Western form and an Eastern form. The Western variety was used along the Mediterranean coast from Kabylia to Morocco and most probably to the Canary Islands. The Eastern variety, old Tifinaghe, is also called Libyan-Berber or old Tuareg. It contains signs not found in the Libyan variety and was used to transcribe Old Tuareg (El ayachi and Fakir, 2009, rachidi, 2009).

Historically, Berber texts did not have a fixed direction. Early inscriptions were written horizontally from left to right, from right to left, vertically (bottom to top, top to bottom); boustrophedon directionality was also known. Modern-day Berber script is most frequently written in horizontal lines from left to right; therefore the bidirectional class for Tifinaghe letters is specified as strong left to right. Displaying Berber texts in other directions can be accomplished by the use of directional over rides or by the use of higher level protocols.

The encoding consists of four Tifinaghe character subsets: the basic set of the " Institut Royal de la Culture Amazighe (IRCAM) ", the extended IRCAM set, other Neo-Tifinaghe letters in

Character number	Character	Character number	Character	Character number	Character
1	ⴰ	12	ⵃ	23	ⵕ
2	ⴱ	13	ⵄ	24	ⵖ
3	ⴳ	14	ⵅ	25	ⵙ
4	ⴳⵯ	15	ⵇ	26	ⵚ
5	ⴷ	16	ⵉ	27	ⵛ
6	ⴹ	17	ⵊ	28	ⵜ
7	ⴻ	18	ⵍ	29	ⵟ
8	ⴼ	19	ⵎ	30	ⵡ
9	ⴽ	20	ⵏ	31	ⵢ
10	ⴽⵯ	21	ⵓ	32	ⵣ
11	ⵀ	22	ⵔ	33	ⵥ

Table 1. Tifinaghe characters adopted by IRCAM

use, and modern Tuareg letters. The first subset represents the set of characters chosen by IRCAM to unify the orthography of the different Moroccan modern day Berber dialects while using the historical Tifinaghe script.

The alphabet Tifinaghe adopted by IRCAM ([El ayachi and Fakir, 2009) is composed of thirty-three characters representing consonants and vowels as shown in Table 1. Fig.2 illustrates a Tifinaghe text.

Tifinaghe characters are consisted by loops, curves and lines segments. In addition some characters have the same shape which differs only by the addition of secondary parts (for example character 3&4, 10&9). This increases the complexity of recognizing Tifinaghe characters. Another problem is the rotation; some characters can be obtained from each other only by rotation of 90° (for example character 13&26, 19&30, 2&11) or 30° (example 23 & 26).

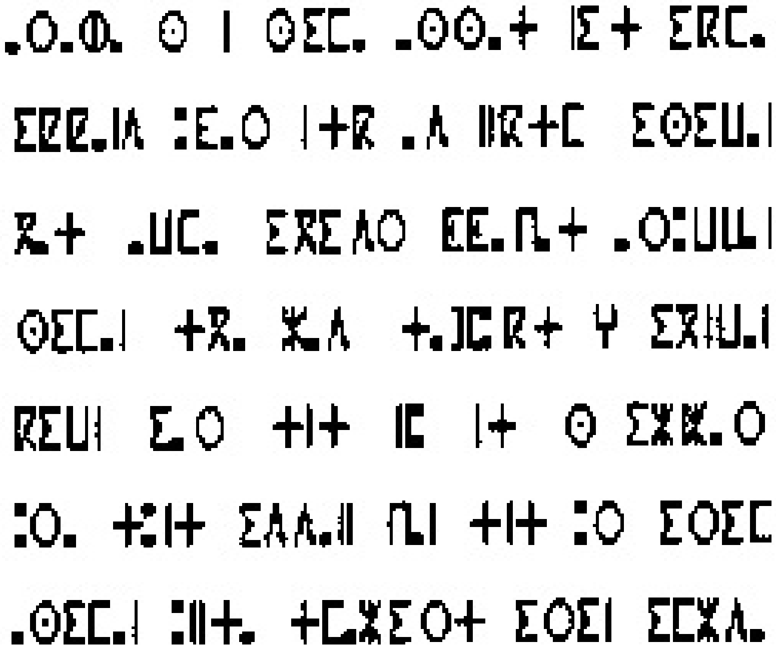


Fig. 2. An example of Tifinaghe text

3. Pre-processing

Pre-processing algorithms provides the required data suitable for further processing. In other words, it establishes the link between real word and recognition engine. Pre-processing steps consist of digitization of image document and cleaning it (by medium filter for example), converting the gray-scale image into binary image, normalizing the text (El ayachi et al, 2010), detecting and correcting baseline skew (Kayallieratou and al, 1999; Mnjunath et al, 2007), and segmenting (Casy et Lecolinet, 1996; Choisy et Belaid, 2002; Hadjar et Ingold, 2003) the text into lines and the lines into characters.

3.1 Normalization of the position

The position normalization is designed to eliminate unwanted areas and reduce the processing time. For this operation we use the histogram given by the following form:

$$HistV = \sum\sum pixel(x,y) \tag{1}$$

Where pixel(x, y), is the intensity of the pixel with the coordinates (x, y).

In this operation, firstly, we compute the horizontal and vertical histograms, secondly, we scan the horizontal histogram in two directions: from top to bottom and bottom to top respectively until the first meeting of black pixels, finally, we scan the vertical histogram in two directions: from left to right and right to left respectively until the first meeting of black pixels. After obtaining the positions of first black pixels, unwanted areas are eliminated in the image as shown in (Fig. 3).

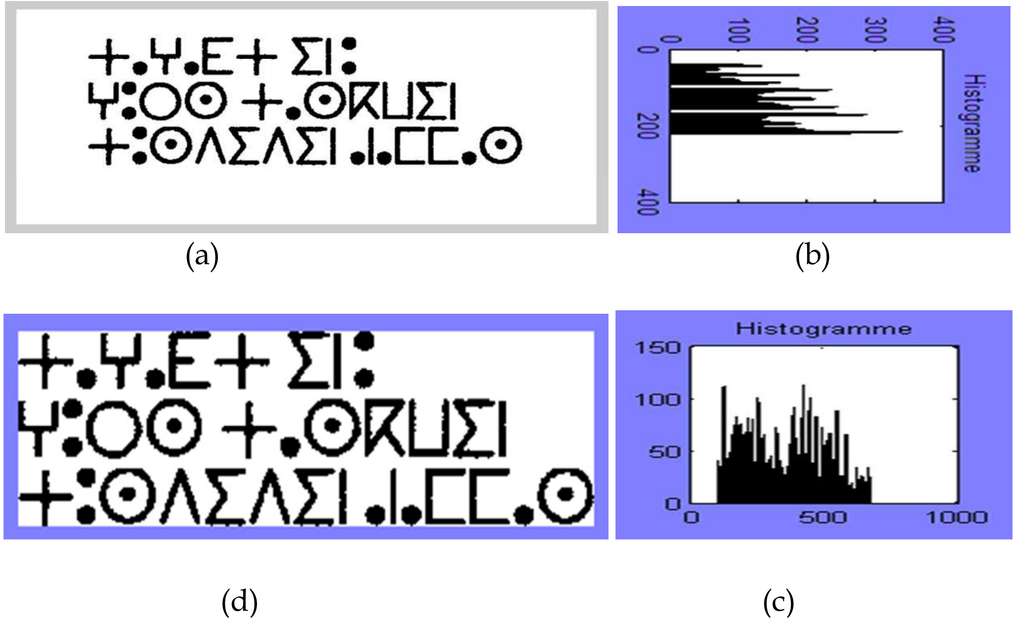


Fig. 3. (a) Before normalization, (d) after normalization, (b) Horizontal histogram and (c) Vertical histogram

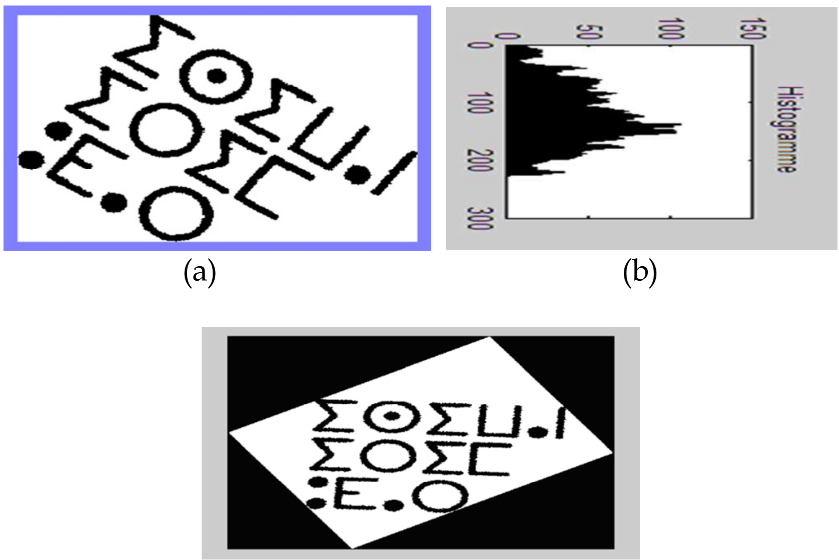
3.2 Baseline skew detection and correction

In many document analysis systems, printed material is scanned and stored as an image. It is later retrieved for character and graphics extraction and recognition. During the scanning process, the document may be skewed and the text lines in the image may not be strictly horizontal. The skew may cause problems in text baseline extraction and document layout structure analysis. Several methods have been developed by many researchers for skew angle detection.

A skew angle is the angle that the text lines of the document image make with the horizontal direction. The skew correction is necessary for the success of many OCR systems. It consists of the extraction of the skew angle θs corresponding to baseline using Hough transform. The baseline is considered as the line corresponding to the maximum points in the horizontal projection profile. The skew angle θs is detected by observing high valued cell in the accumulative matrix in the Hough transform space. The image is then rotated by θs in the opposite direction so that the scripts become horizontal. Fig. 4(a) and Fig. 4(c) respectively show a text before and after baseline skew correction.

3.3 Segmentation

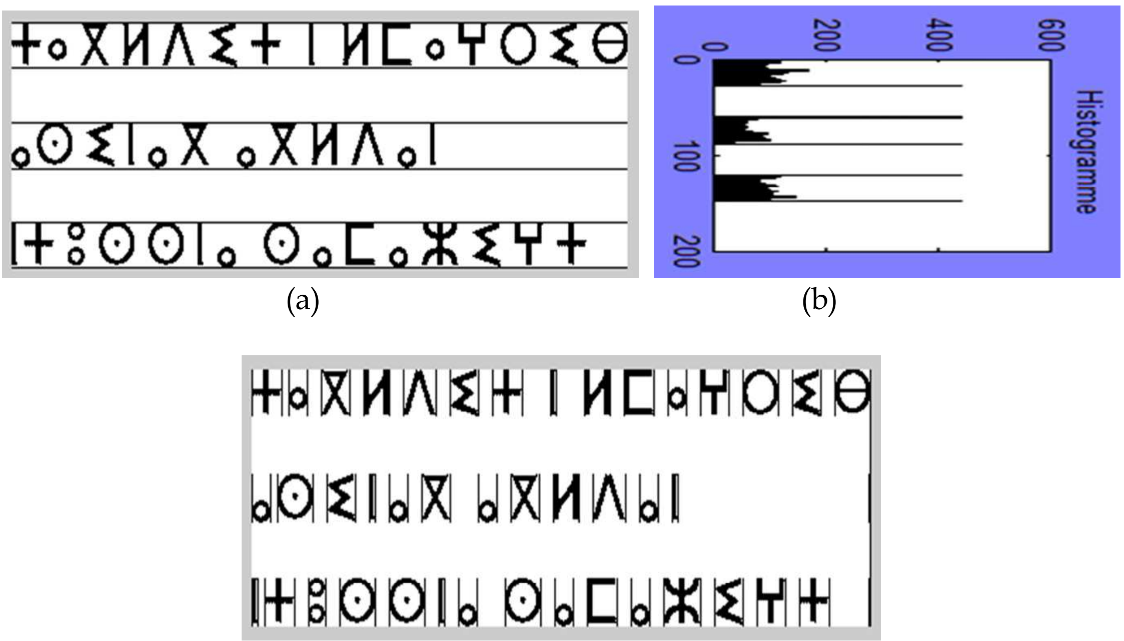
In this phase, the proposed OCR system detects individual text lines and then segments lines into characters. The lines of a text are segmented by using the horizontal histogram; we browse from top to bottom until the first line containing at least one black pixel, the line is the beginning of the first line of text, then we continue traverse until a line that contains only white pixels, this line corresponds to the end of the first line of text. With the same way, we continue to detect other text lines (Fig.5 a). The same principle is used in the vertical histogram to detect characters in each line of text (Fig.5 b).



(a) (b)

(c)

Fig. 4. (a) Before correction, (c) After correction and (b) Horizontal histogram



(a) (b)

(c)

Fig. 5. (a) lines detection, (b) Horizontal histogram and(c) Characters segmentation

4. Features extraction

The next stage in the Tifinaghe character recognition is the feature extraction stage. Feature extraction represents the character image by a set of numerical features. These features are used by the classifier to classifier the data. In our work moments and other shape descriptors by Hu have been utilized to build the feature space. Using nonlinear combinations of geometric moments Hu derived a set of invariant moments which has the desired property of being invariant under image translation, scaling and rotation.

In addition to the invariant moments derived by Hu(Ibrahim, 2005), the modified invariant moments (Fakir and al, 2000), computed from the shape boundary for each character, and Walsh transform (Fazekas and Hajdu, 2001) are used as features.

4.1 Invariant moments

Let $f(x,y)$ be 1 over a closed and bounded region R and 0 otherwise. Define the $(p,q)th$ moment as:

$$m_{pq} = \iint_R x^p y^q f(x,y)dxdy \text{ , where } p,q = 0,1,2,... \tag{2}$$

The central moments can be expressed as

$$\mu_{pq} = \iint_R (x-\overline{x})^p (y-\overline{y})^q f(x,y)dxdy \tag{3}$$

Where

$$\overline{x} = \frac{m_{10}}{m_{00}} \text{ , } \overline{y} = \frac{m_{01}}{m_{00}} \tag{4}$$

However for digital images the continuous image intensity f(x,y) is replaced by a matrix where x and y are the discrete locations of the image pixels. The integral in equation 3 is approximated by the summation

$$\mu_{pq} = \sum_{(x,y)\in R} \sum (x-\overline{x})^p (y-\overline{y})^q f(x,y) \tag{5}$$

It can be verified that the central moments up to the order p+q≤ 3 may be computed by the following formulas:

$$\mu_{00} = m_{00} \tag{6}$$

$$\mu_{10} = 0 \tag{7}$$

$$\mu_{01} = 0 \tag{8}$$

$$\mu_{11} = m_{11} - \overline{y}m_{10} \tag{9}$$

$$\mu_{20} = m_{20} - \overline{x}m_{10} \tag{10}$$

$$\mu_{02} = m_{02} - \overline{y}m_{01} \tag{11}$$

$$\mu_{12} = m_{12} - 2\overline{y}m_{11} - \overline{x}m_{02} + 2\overline{y}^2 m_{10} \tag{12}$$

$$\mu_{21} = m_{21} - 2\overline{x}m_{11} - \overline{y}m_{20} + 2\overline{x}^2 m_{01} \tag{13}$$

$$\mu_{30} = m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2 m_{10} \tag{14}$$

$$\mu_{03} = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01} \tag{15}$$

The central moments are invariant to translation. They can also be normalized to be invariant to scaling change by the following formula. The quantities in equation (6) are called normalized central moments

$$\alpha_{pq} = \frac{\mu_{pq}}{\mu_{00}{}^{\gamma}} \tag{16}$$

Where

$$\gamma = \frac{p+q}{2} + 1 \quad \text{, for } p+q = 2,3,\ldots \tag{17}$$

Invariant moments derived by Hu (Hu, 1962) were frequently used as features for shape recognition and were shown to be invariant to scaling, translation and rotation (Tables 2, 3, 4).

$$\varphi_1 = \alpha_{20} - \alpha_{02} \tag{18}$$

$$\varphi_2 = (\alpha_{20} - \alpha_{02})^2 + 4\alpha_{11}^2 \tag{19}$$

$$\varphi_3 = (\alpha_{30} - \alpha_{12})^2 + (3\alpha_{12} - \alpha_{03})^2 \tag{20}$$

$$\varphi_4 = (\alpha_{30} + \alpha_{12})^2 + (\alpha_{21} + \alpha_{03})^2 \tag{21}$$

$$\begin{aligned}\varphi_5 = {} & (\alpha_{30} - 3\alpha_{12})(\alpha_{30} + \alpha_{12})[(\alpha_{30} + \alpha_{12})^2 - 3(\alpha_{21} + \alpha_{03})^2] \\ & + (3\alpha_{21} - \alpha_{03})(\alpha_{21} + \alpha_{03})[3(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2]\end{aligned} \tag{22}$$

$$\begin{aligned}\varphi_6 = {} & (\alpha_{20} - \alpha_{02})[(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2] \\ & + 4\alpha_{11}(\alpha_{30} + \alpha_{12})(\alpha_{21} + \alpha_{03})\end{aligned} \tag{23}$$

$$\begin{aligned}\varphi_7 = {} & (3\alpha_{21} - \alpha_{30})(\alpha_{30} + \alpha_{12})[(\alpha_{30} + \alpha_{12})^2 - 3(\alpha_{21} + \alpha_{03})^2] \\ & + (3\alpha_{12} - \alpha_{03})(\alpha_{21} + \alpha_{03})[3(\alpha_{30} + \alpha_{12})^2 - (\alpha_{21} + \alpha_{03})^2]\end{aligned} \tag{24}$$

The φ_i 's have dynamic values. Thus it was found that it was more practical to deal with the logarithm of magnitude of φ_i. Thus the seven moment invariants used in the proposed system are replaced by their logarithm values. For each character issued from the segmentation process the above moment invariant descriptors are calculated and fed to the classifiers.

$\log(\varphi_i)$	Original image	Reduced image	Mirror image	Rotated image by (90°)
φ_1	-1.3666	-1.3413	-1.3666	-1.3598
φ_2	-6.6891	-6.5944	-6.6891	-6.7362
φ_3	-7.1587	-7.3184	-7.1587	-7.3915
φ_4	-12.1029	-10.7623	-12.1029	-11.1215
φ_5	-24.2394	-20.1474	-24.2394	-20.4762
φ_6	-16.1004	-14.5595	-16.1004	-14.5128
φ_7	-22.1628	-21.5532	-21.7188	-21.2735

Table 2. Invariant Moments

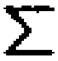
$\log(\varphi_i)$	Original image	Reduced image 	Mirror image	Rotated image by (90°)
φ_1	-1.5010	-1.5031	-1.5112	-1.5185
φ_2	-11.9233	-8.9845	-10.0586	-11.1259
φ_3	-7.7360	-7.7739	-7.4510	-7.3024
φ_4	-9.1054	-9.6056	-9.2274	-9.2739
φ_5	-17.9265	-18.5968	-17.9453	-17.9095
φ_6	-15.8471	-14.2235	-14.2585	-14.8929
φ_7	-18.6203	-19.4092	-18.9059	-18.7813

Table 3. Invariant Moments

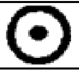
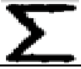
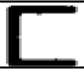
$Log(\varphi_i)$				
φ_1	-1.2978	-1.5010	-1.7512	-1.5141
φ_2	-8.1338	-11.9233	-6.3559	-4.0690
φ_3	-12.2658	-7.7360	-14.2564	-8.4575
φ_4	-12.8021	-9.1054	-13.4338	-8.3696
φ_5	-25.3674	-17.9265	-27.6235	-16.8251
φ_6	-17.7442	-15.8471	-17.5999	-10.5131
φ_7	-27.0270	-18.6203	-27.5615	-16.9178

Table 4. Invariant Moments for different characters

4.2 Modified Invariant Moments

In this phase features are extracted from the external contour of the character (Fig.6). In order to differentiate between the characters illustrated in Fig.7that have the same external contour, we extract other features such C_ext, H_ext, V_ext, C_int, H_int and V_int , where

- C_ext is the number of externals contours;

- H_ext is the horizontal histogram. This feature is used to compute the number of externals contours;
- V_ext is the Vertical histogram. It is used to compute the number of externals contours;
- C_int is the number of internals contours;
- H_int is the horizontal histogram; it is used to compute the number of internals contours;
- V_int is the Vertical histogram; it is used to compute the number of internals contours.

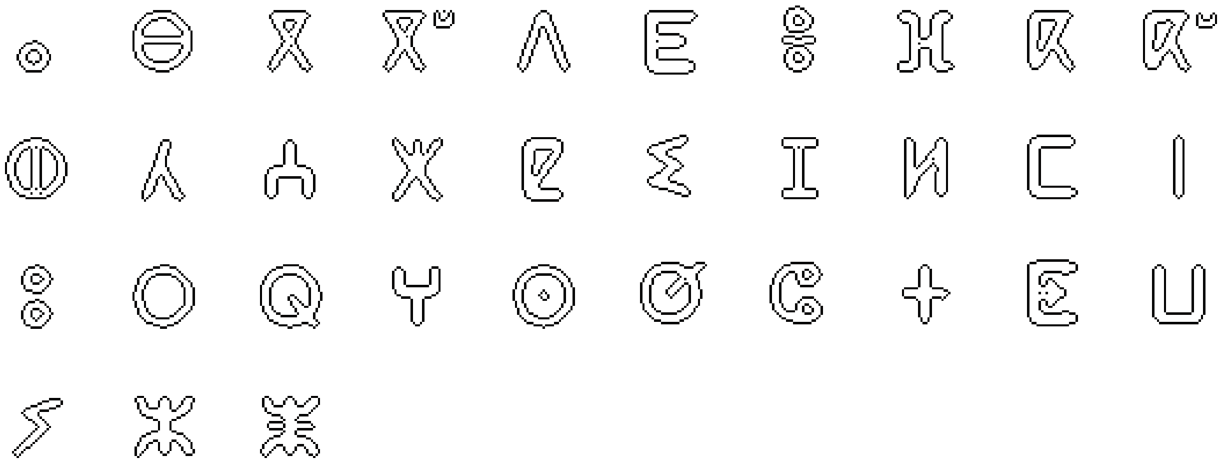


Fig. 6. Tifinaghe characters Contours

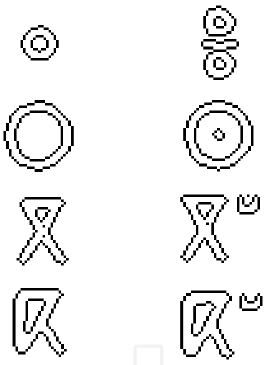


Fig. 7. Externals contours used to calculate modified invariant moments

We modified the moment definition in equation (2) using the character boundary only.

$$m_{pq} = \int_C x^p y^q f(x,y) ds \text{ , for p, q = 0, 1, 2, …} \quad (25)$$

Where $\int_C$ is a line integral along the curve C, and

$$ds = \sqrt{((dx)^2 + (dy)^2)} \quad (26)$$

The modified central moments can be similarly defined as:

$$\mu_{pq} = \int_C (x - \bar{x})^p (y - \bar{y})^q ds \quad (27)$$

Where

$$\bar{x} = \frac{m_{10}}{m_{00}} \;,\; \bar{y} = \frac{m_{01}}{m_{00}} \tag{28}$$

For a digital character, equation (9) becomes

$$\mu_{pq} = \sum_{(x,y)\in C} (x-\bar{x})^p (y-\bar{y})^q \tag{29}$$

The modified central moments are invariant to translation.
Theorem 1: For μ_{pq} defined in equation (9),

$$\alpha'_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{p+q+1}} \text{ is scaling invariant for p, q = 2, 3, } \ldots \tag{30}$$

Proof: Suppose C is a smooth curve in the plane, C′ is the curve obtained homogeneously by rescaling the coordinates by a factor r, then

$$\mu'_{pq} = \int_C (x(s'))^p (y(s'))^q \, ds' = r^{p+q+1} \mu_{pq} \tag{31}$$

Since

$$\mu_{00} = \int_C ds = |C| = \text{length of curve C} \tag{32}$$

$$\mu'_{00} = \int_C ds' = |C'| = \text{ length of curve C}'\text{=r } |\text{C}| \tag{33}$$

Then, for any r > 0, we have

$$\frac{\mu'_{pq}}{(\mu'_{00})^{p+q+1}} = \frac{r^{p+q+1}\mu_{pq}}{r^{p+q+1}|C|} = \frac{\mu_{pq}}{(\mu_{00})^{p+q+1}} \tag{34}$$

The quantity in equation (34) is invariant to a homogenous scaling.
Theorem 2: Suppose C is a smooth curve in the plane and C′ is the curve obtained by rotating C an angle θ clockwise, then

$$\varphi'_i = \varphi_i \;\text{ for } 1 \le \text{ i} \le 7 \tag{35}$$

Where φ'_i is defined as in equations (18, 19, 20, 21, 22, 23 and 24) by using α'_{pq} instead of α_{pq} for $p+q = 2,3,...$
Proof:

$$\mu'_{pq} = \int_C (x(s'))^p (y(s'))^q \, ds' = \int_C \left\{ \left[x(s)\cos\theta - y(s)\sin\theta\right]^p \left[y(s)\sin\theta + x(s)\cos\theta\right]^q \right\} ds \tag{36}$$

Note that

$$ds' = ds \tag{37}$$

We shall prove $\varphi_1^{'} = \varphi_1$ and $\varphi_2^{'} = \varphi_2$ as examples.
For 3 ≤ j ≤ 7,

$$\varphi_j^{'} = \varphi_j \tag{38}$$

can be similarly proved after computations by using the trigonometric identities:

$$\cos^2\theta + \sin^2\theta = 1 \tag{39}$$

And

$$(\cos^3\theta - 3\cos\theta\sin^2\theta)^2 + (\sin^3\theta - 3\cos^2\theta\sin\theta)^2 = 1 \tag{40}$$

Table 5 represents the seven elements of the vector calculated using modified invariant moments for one character with four transformations.

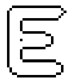
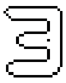
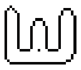
$\log(\varphi_i)$				
φ_1	-0.0229	-0.0311	-0.0229	-0.0229
φ_2	-2.3107	-2.4123	-2.3107	-2.3107
φ_3	-3.4907	-3.5817	-3.4907	-3.4907
φ_4	-5.4983	-5.5664	-5.4983	-5.4983
φ_5	-10.2246	-10.3329	-10.2246	-10.2246
φ_6	-6.8287	-6.9195	-6.8287	-6.8287
φ_7	-10.6272	-10.7364	-10.6272	-10.3445

Table 5. Modified Invariant Moments

4.3 Walsh transform

The Walsh transformation is given by:

$$W(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)g(x,y,u,v) \tag{41}$$

Where $f(x,y)$ is the intensity of the pixel with the coordinates (x,y) in the original binary image. The size of image f is $N * N$, and $u,v = 0,...,N-1$, thus we compute N^2 Walsh transforms, $g(x,y,u,v)$ is the Kernel function given by the following form:

$$g(x,y,u,v) = (1/N)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-i-1}(u)+b_i(y)b_{n-i-1}(v)} \tag{42}$$

Where $b_i(x)$ is the ith bit in the binary expansion of x (it is equal either 0 or 1).
Table 6 represents the seven first elements of the vector Walsh calculated for one character with four transformations. While Table 7 illustrates Walsh coefficient calculated for different characters.

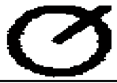
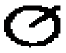
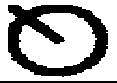
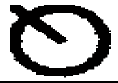
w_i				
w_1	-0.2500	-0.5000	-0.2500	-0.3750
w_2	-0.8750	-1	-0.8750	-1
w_3	-1.5000	-1.6250	-1.5000	-1.5000
w_4	-2.1250	-2.2500	-2.1250	-2.1250
w_5	-2.8750	-3	-2.8750	-2.8750
w_6	-3.3750	-3.6250	-3.5000	-3.6250
w_7	-3.8750	-4.1250	-4	-4.3750

Table 6. Walsh coefficients

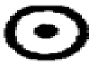
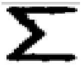
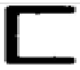
w_i				
w_1	-0.0020	0	0	-0.0039
w_2	-0.0049	-0.0029	-0.0034	-0.0078
w_3	-0.0078	-0.0064	-0.0069	-0.0118
w_4	-0.0098	-0.0098	-0.0103	-0.0157
w_5	-0.0118	-0.0132	-0.0137	-0.0196
w_6	-0.0142	-0.0167	-0.0172	-0.0211
w_7	-0.0172	-0.0201	-0.0206	-0.0211

Table 7. Walsh coefficients

5. Characters recognition

In the character recognition system, the recognition is the last phase which is used to identify the segmented character. Where we use two techniques: the Neural Network (Hamza, 2008; Gu and al, 1983; Alnsour and Alzoubady, 2006; Asiri and Khorsheed, 2005) and Dynamic Programming (Sylvain and al, 2003).

5.1 Neural network

Characters are classified according to their computed features by means of artificial neural networks. Among the many applications that have been proposed for neural networks, character recognition has been one of the most successful. Many neural network architectures have been used in optical character recognition implementation. MLP is usually a common choice. Unfortunately, as the number of inputs and outputs grow, the MLP grows quickly and its training becomes very expensive. In addition, it is not easy to come up with a suitable network design and many trail-and-error cycles are required.
The Neural Network (Fig.8) represents an example of Multilayer Neural Network which contains one hidden layer. It has:

- An input layer of 7 (invariant moment's vector) inputs cells $E_i = X_i$ where the cells represents the inputs E_i of Network.
- A hidden layer of 3 activations Neural Y_j .
- An output layer of 6 activations Neural Z_k .
- 7×3 connections between input layer ant hidden layer, each weighted by V_{ji} .

- 3×6 connections between hidden layer and output layer, each weighted by W_{kj} .
- X_0 , Y_0 are initialled values which are scalars.

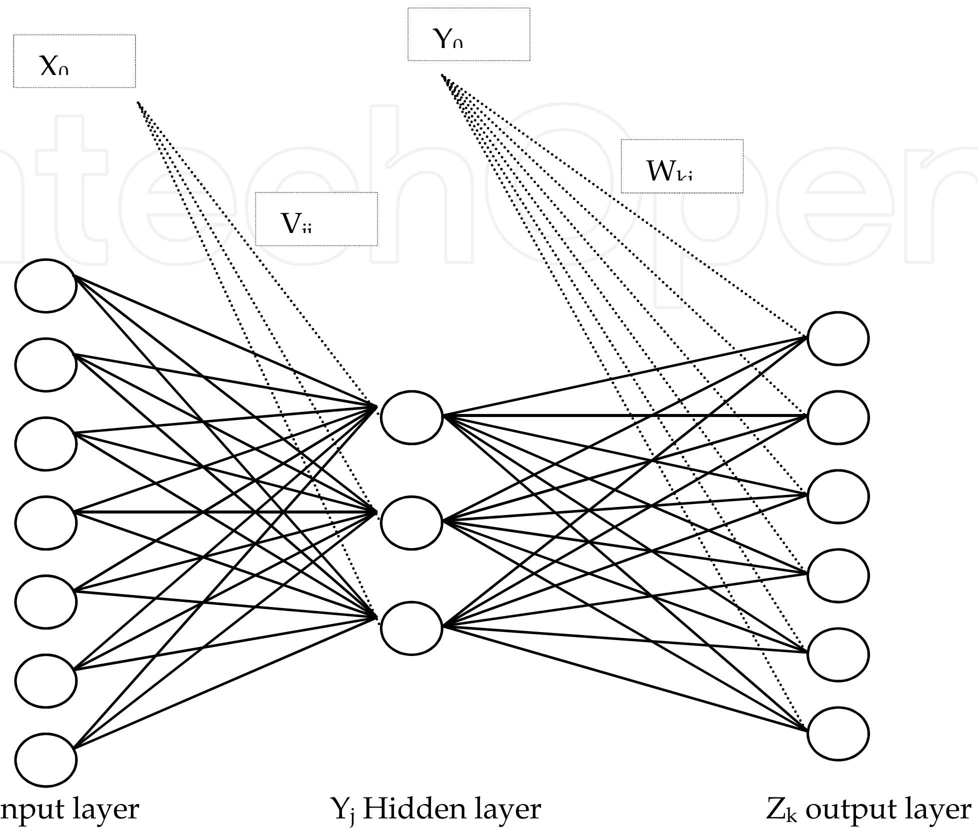


Fig.8 Neural Network

The processing of Neural Network consists of five steps:

Step 1. Initializing of weights connections
The weights are randomly selected.

Step 2. inputs propagation
The inputs E_i are presented to input layer: $X_i = E_i$.
We propagate to hidden layer as follows

$$Y_j = f\left(\sum_{i=1}^{7} X_i V_{ji} + X_0\right) \tag{43}$$

After hidden layer, the output layer is given by

$$Z_k = f\left(\sum_{j=1}^{3} Y_j W_{kj} + Y_0\right) \tag{44}$$

where the values X_0 and Y_0 are scalars and f is the activation function given by

$$f(a) = 1 \Big/ 1 + \exp(-a) \tag{45}$$

Step 3. Error back propagation
For each example of applied learning base input of the network, we calculate the error at output layers by the

$$E_k = Z_k\left(1 - Z_k\right)\left(S_k - Z_k\right) \tag{46}$$

where S_k desired output and Z_k the actual output.
In the next, we propagate this error on the hidden layer; the error of each neuron of the hidden layer is given by

$$F_j = Y_j\left(1 - Y_j\right)\sum_{k=1}^{6} W_{kj}.E_k \tag{47}$$

Step 4. Correction of connections weights
We change the weights connections as follows

- Between input layer and hidden layer:

$$\Delta V_{ji} = \eta.X_i.F_j \tag{48}$$

and

$$\Delta Y_0 = \eta.F_j \tag{49}$$

- Between hidden layer and output layer:

$$\Delta W_{kj} = \eta.Y_j.E_k \tag{50}$$

and

$$\Delta X_0 = \eta.E_k \tag{51}$$

Where $(\eta = 0.9)$ is the learning parameter determinate experimentally.

Step 5. (Loop)
Loop in step two allows computing the error threshold (0.0001), and the number of terations (50000).
After the learning of Network and the execution of Tifinaghe characters recognition system to recognize a text, we use the Euclidian distance to identify characters.

$$d\left(t_k, o\right) = \left(\sum_{i=1}^{6}\left(t_{ki} - o_i\right)^2\right)^{1/2} \tag{52}$$

With, t_k is a desired output and o is the output of Network.

5.2 Dynamic programming

Dynamic Programming was applied to speech recognition by Tappert et al 1982, by Sakoe and Chiba, 1978 to recognize Latin character and by El ayachi and al, 2010 to recognize Tifinaghe characters. The Dynamic Programming strategy is a useful technique for the problem of optimization. It is often used to find the shortest path from one place to another and solve the comparative problem between two strings.

The operation of Dynamic Programming (for invariant moments) is based on the following steps:

Step 1. Compute the matrix d between the vector of segmented character V_{car} and each one of the Tifinaghe characters vectors V_{ref} .
The matrix d is given by:

$$d(x,y) = \left| V_{car}(x) - V_{ref}(y) \right| \tag{53}$$

Where $x,y = 1,2,...,7$

Step 2. Calculate the optimal path from point $(1,1)$ to point (x,y) by the following recursive relationship:

$$S(x,y) = d(x,y) + \min \begin{Bmatrix} S(x-1,y), \\ S(x-1,y-1, \\ S(x,y-1) \end{Bmatrix} \tag{54}$$

Where
$S(x,y)$ is4 the cumulative distance along the optimal path from point $(1,1)$ to point (x,y) .
$S(x,y)$ is eva4luated on the area $[1,7]*[1,7]$ that is browsed column by column or row by row starting from point $(1,1)$.

Step 3. Calculate the dissimilarity indices using the following form:

$$D(V_{car},V_{ref})=S(7,7)/7 \tag{55}$$

6. Experiments results

Experiments have been performed to test the above system. The developed Tifinaghe text recognition system has been tested on printed text. The system designed in Matlap for the recognition. The image to be tested is captured using a scanner. Before the recognition phase a database of 360 images is made. All tests are applied on 124 characters.

The system is working fine and showing a good recognition rate (Table 8). It has been noticed that the extracted features of the images produced from segmentation module deviate a lot from the respective results in the training set. It seems that the resolution differences are affecting the geometric moments of the image, making them highly variant. It is expected that the recognition rate of the system can be improved by normalizing the training set as the characters that result after the segmentation phase. The system has been implemented on Intel (R) Core (TM) 2 Duo, CPU T5870 @ 2.00 Ghz, with a RAM: 2.00 Go. The system is still under development.

	Dynamic Programming		Neural Network	
	Recognition rates	Error rates	Recognition rates	Error rates
Invariant Moments	93.55%	6.45%	94.35%	5.65%
Modified Invariant Moments	92.32%	7.68%	93.63%	6.37%
Walsh Transform	92.75%	7.25%	97.58%	2.42%

Table 8. Recognition rates & Error rates

7. Conclusion & perspectives

Two recognition methods based on neural network and dynamic programming has been presented. The system recognition consists of three phases including pre-processing, features extraction and recognition. The pre-processing includes position normalization, baseline skew correction and segmentation. The skew angle is determined by using Hough transform. The segmentation process consists of two steps: (a) segmentation of text into lines using horizontal histogram, (b) segmentation of lines into characters using vertical histogram. In features extraction phase a set of 7 invariant moments descriptors, 7 modified invariant moments descriptors and 7 Walsh coefficients have been used to represent the numerical features of the character extracted. Finally the numerical features are passed to the classifiers to recognize the character.

The programs were written using Matlap. As mentioned previously, no efficient technique has been found for Tifinaghe scripts recognition. This field is of importance for future research.

8. References

Alnsour, A.J. and L.M. Alzoubady. (2006). *Arabichandwritten characters recognized by neocognitron artificial neural network*. J. Pure Appl. Sci., 3: 1- 17.

Asiri, A. and M.S. Khorsheed. (2005). *Automatic processing of handwritten Arabic forms using neural networks*. Proceeding of the World Academy of Science, Engineering and Technology, Aug. 2005, pp: 313-317.

Attila Fazekas and Andras Hajdu. (2001). *Recognizing Type set Documents using Walsh*, JCIT-CIT 9, 2-2001, pp: 101-112.

B B Chaudhuri, U Pal and M Mitra. (2002). *Automatic recognition of printed Oriya script*, Sädhanä Vol. 27, Part 1, February 2002, pp. 23–34. © Printed in India.

Casey & Lecolinet . (1996). *A survey of methods and strategies in character segmentation*. IEEA Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 7, pp: 690-706 July 1996.

Ch. Choisy and A. Belaid. (2002). *Cross- learning in analytic word recognition without segmentation*, in Int. Journal on documentAnal. & Recognition IJDAR, 4(4): 281-289, 2002

C. Tappert. (1982). *Cursive Script Recognition by Elastic Matching*, IBM J. Res. Develop., Vol. 26, No. 6, 1982, pp. 765-771.

D. J. Burr. (1982). *A Normalizing Transform For Cursive Script Recognition*, Proc. 6th Int. J. Conf. on Pattern Recognition Munich (1982), pp. 1027–1030.

E.Kavallieratou, N.Fakotakis, and G.Kokkinakis. (1999). *Skew Angle Estimation in Document Processing Using Cohen's Class Distributions*, PRL(20), No. 11-13, November 1999, pp. 1305-1311.

Hadjar, K. and R. Ingold. (2003). *Arabic newspaper segmentation*. Proceeding of 7th International Conference on Document Analysis and Recognition, Aug. 3-6, IEEE Computer Society, pp: 895-899.

Hamza, Ali A. (2008). *Back Propagation Neural Network Arabic Characters Classification Module Utilizing Microsoft Word*; Journal of Computer Science 4 (9): 744-751, 2008.

H. Sakoe and S. Chiba. (1978). *Dynamic Programming Algorithm Optimization for Spoken Word Recognition*, IEEE Trans. Acoust., Speech and Signal Processing, Vol. ASSP-26, No. 1, 1978, pp. 401-408

Ibrahim S. I. Abuhaiba. (2005). *Arabic Font Recognition Using Decision Trees Built From Common Words*, JCIT-CIT 13, 3-2005, 211-223

M. Amrouch, Y. Es saady, A. Rachidi, M. El Yassa and D. Mammass. (2009). *Printed Amazigh Character Recognition by a Hybrid Approach Based on Hidden Markov Models and the Hough Transform*, 978-1-4244-3757-3/09/$25.00 ©2009 IEEE

M. Blumenstein & C. K. Cheng & X. Y. Liu. (2002). *New Preprocessing Techniques for Handwritten Word Recognition*, in Proceedings of the Second IASTED International Conference on Visualization, Imaging and Image Processing (VIIP 2002), ACTA Press, Calgary, pp. 480-484.

M. Fakir. (2001). *Reconnaissance des Caractères Arabes Imprimés*, Thesis, 2001, pp : 28-36.

M. Fakir, B. Bouikhalene and K. Moro. (2009). *Skeletonization Methods Evaluation for the Recognition of PrintedTifinaghe characters*, SITCAM'09, Agadir-Maroc

M. Fakir and C. Sodeyama. (1993). *Recognition of Arabic printed Scripts by Dynamic Programing Matching Method*, IECICE Trans. Inf & Syst, Vol. E76- D, No.2 Feb. 93, pp: 31-37

M. Fakir, M.M. Hassani and C.Sodeyama. (2000). *On the recognition of Arabic characters using Hough transform technique*, Malysian Journal of Computer Science Vol. 13, No.2, Dec.2000, pp: 39-47.

M. K. Brown. (1983). *Pre-processing techniques for cursive word recognition*, Pattern Recognition, Vol.13, N°.5, pp: 447-451, 1983

M. K. Hu. (1962). *Visual pattern recognition by moment invariants*, IRE trans. Infor. Theory TT-8, pp: 179-187, 1962.

Mnjunath Aradhya et al. (2007). *Skew estimation technique for binary document images based on thinning and moments*, engineering letters, 14:1, EL_14_1_22, 2007.

N. Mezghani, A.Cheret and N.Mitiche. (2008). *Bayes classification of online arabic characters by Gibbs modeling of class conditional densities*, IEEE Trans PAMI Vol 30, issue 7, july 2008, pp: 1121-1131

N. Miyazaki et al. (1974). *Recognition of handprint katakana characters*, Annual conference of inf. Process. Society of Japan, 1974.

P.M. Lallican, C. Viarp-Gaudin, S. Knerr. (2000). *From off-line to on-line handwriting recognition*. Proc. 7th workshop on frontiers in handwriting recognition, pp.303-312, Amsterdam 2000.

R. El ayachi and M. Fakir. (2009). *Recognition of Tifinaghe Characters Using Neural Network*, 978-1-4244-3757-3/09/$25.00 ©2009 IEEE

R. El ayachi, K. Moro, M. Fakir and B. Bouikhalene. (2010). *On The Recognition Of Tifinaghe Scripts*, JATIT, vol. 20, No. 2, pp: 61-66, 2010.

R. M. Bozinovic and S. N. Shihari. (1989). *Off Line Cursive Script Word Recognition*, IEEE Trans.Pattern Anal. Mach. Intell. PAMI 11, 1989, pp. 68- 83

Sylvain Chevalier, Edouard Geoffrois, and Franc¸oise Prêteux. (2003). *A 2D Dynamic Programming Approach for Markov Random Field-based Handwritten Character Recognition*, Proceedings IAPR International Conference on Image and Signal Processing (ICISP' 2003), Agadir, Morocco, 2003, p. 617-630.

Y. Es saady, M.Amrouch, A. Rachidi, M. El Yassa and D. Mammass. (2009). *Reconnaissance de caractères Amazighes Imprimés par le Formalisme des Automates à états finis*, SITCAM'09, Agadir-Maroc,12-13 December 2009, pp:48-57.

Y.X. Gu et al. (1983). *Application of a multilayer tree in computer recognition of Chinese character*, IEEE Trans. On PAMI-5, N°.1, pp: 83-89, 1983.

Recent Advances in Document Recognition and Understanding
Edited by Dr. Minoru Mori

ISBN 978-953-307-320-0
Hard cover, 94 pages
Publisher InTech
Published online 17, October, 2011
Published in print edition October, 2011

In the field of document recognition and understanding, whereas scanned paper documents were previously the only recognition target, various new media such as camera-captured documents, videos, and natural scene images have recently started to attract attention because of the growth of the Internet/WWW and the rapid adoption of low-priced digital cameras/videos. The keys to the breakthrough include character detection from complex backgrounds, discrimination of characters from non-characters, modern or ancient unique font recognition, fast retrieval technique from large-scaled scanned documents, multi-lingual OCR, and unconstrained handwriting recognition. This book aims to present recent advances, applications, and new ideas that are relevant to document recognition and understanding, from technical topics such as image processing, feature extraction or classification, to new applications like camera-based recognition or character-based natural scene analysis. The goal of this book is to provide a new trend and a reference source for academic research and for professionals working in the document recognition and understanding field

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821