I.J. Intelligent Systems and Applications, **2017, 12, 29-45**
Published Online December 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2017.12.04

Handwritten Character Recognition on Focused on the Segmentation of Character Prototypes in Small Strips

Ali Benafia[1]
[1]University of Batna, Algeria
E-mail: ali_bnfa@yahoo.fr

Smaine Mazouzi[2]**, Benafia Sara**[3]
{[2]University of Skikda, [3]ENAF of Batna} Algeria
E-mail: mazouzi_smaine@yahoo, mouradasma23@gmail.com

Received: 02 April 2017; Accepted: 15 September 2017; Published: 08 December 2017

Abstract**—**The automatic recognition of handwriting is a particularly complex operation. Until now, there is no algorithm able to recognize handwriting without that; there are assumptions to take in advance to facilitate the task of the process. A handwritten text can contain letters lowercase, uppercase letters, characters sticks and digits. Therefore, it is capital to know extract and separate all these different units in order to process them with specific algorithms to their class handwriting.

In this paper, we present a system for unconstrained handwritten text recognition, which allows to achieve this operation thanks to an intelligent segmentation based on an iterative cutting in a multi-script environment.

The results obtained from the experimental protocol reach an "equal error rate" (EER) neighboring to 6%. These calculations were calculated with a relatively small base; however this same rate can be decreased with great bases. Our results are extremely encouraging for the simple reason that our approach is situated in a more general context unlike other approaches which define several non-rigid assumptions; this clearly makes the problem simpler and may make it trivial.

Index Terms**—**Character recognition, feature extraction, off-line handwriting analysis, segmentation.

I. Introduction And Motivation

Handwriting recognition has been one of the most challenging research areas in the field of pattern recognition in recent years. Due to the variability in writing style and sizes, recognition of handwritten scripts is even more challenging than printed scripts [12], [22]. In general, handwriting recognition can be defined as the task of transforming text represented in spatial form of graphical marks into its symbolic representation [5]. There are two kinds of systems, on-line and off-line systems [23]. The main difference between them is that in an on-line system the recognition was performed at the time of writing (e.g., tablet, smart phone) while the off-line handwriting recognition is performed after the writing is completed (e.g. scanned document).

The importance and need of handwriting recognition has been arising in many real world applications such as postal addresses [1], bank checks processing [38], hand-filled forms processing [29], conversion of field notes and historical manuscripts where a vast amount of handwriting historical manuscripts can be easily converted into an electronic representation for sharing and searching [24].

In the last few years the field of handwriting recognition has become much more popular. Not only are more researchers trying to tackle the problems that it presents, but solutions to these problems are slowly becoming available and are actually being sold as useful products of late, pen computers have become available with handwriting recognition software for isolated characters and more recently for cursive script.

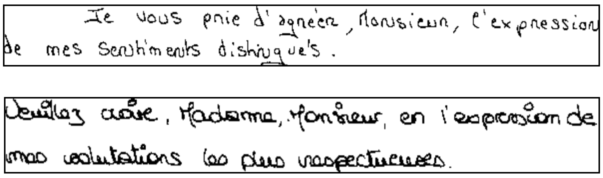
II. Manuscript Recognition Problem

Handwriting Recognition is a difficult problem due to the high variability of character shapes and handwritten strokes produced by different writers. In general, this problem depends on substantial criteria [12], [17]:

-Conditioning of information: the recognition of a handwritten text can be more or less conditioned by the presence of pre-cased (case of forms, address, postal code), frames (amounts of the checks) or lines (lines of an address block, the literal amount of a check), or unconditioned (free documents).

-Writing style: the off-line handwriting recognition is comparatively difficult, as different people have different handwriting styles, like script writing with spaced characters, writing free scripts, cursive writing etc.

-The number of writers: the reduction of the number of potential writers allows to reduce the variability and to learn the different styles of writing.



Je vous prie d'agréer, Monsieur, l'expression
de mes Sentiments distingués .

Veuillez croire, Madame, Monsieur, en l'expression de
mes salutations les plus respectueuses.

Fig.1. Variability of manuscript handwriting

A lot of works of handwritten character recognition [36], are done at a character level, which requires to determine character boundaries. This problem is difficult and may be resolved by jointly segmenting and recognizing the characters [27], [37], such a problem can be formulated as a multi-class classification problem with a large range of small snippets extracted from the different prototypes of characters [25].

The remainder of the paper contains 9 sections. Section 2 briefly describes the problem of handwriting recognition systems, Section 3 reviews some recent approaches and achievements, A proposed methodology is developed (in detail) in Sections 4,5,6, and 7 , experimental results are presented in Section 8, finally a conclusion and prospect are drawn in Section 9.

III. Related Work

The handwriting recognition problem has been addressed by many researchers for a long period of time, in this context; we present a state of art in the various works in connection with our study.

In [28], the authors have presented a method for off-line writer identification using the contours of fragmented connected components in mixed-style handwritten samples of limited size. The writer considers to be characterized by a stochastic pattern generator, producing a family of character fragments named fraglets. Using a codebook of such fraglets from an independent training set, the probability distribution of fraglet contours was computed for an independent test set. Results revealed a high sensitivity of the fraglet histogram in identifying individual writers on the basis of a paragraph of text. Large-scale experiments on the optimal size of Kohonen maps with fraglet contours were performed, showing usable classification rates within a non-critical range of Kohonen map dimensions.

Bertolami et Al [3] have proposed a recognition system implementing an implicit segmentation using a sliding window for extracting a vector representation of the image. The features used here are relatively simple; this is the density of black pixels in a window represented from each frame. A frame is thus subdivided into sixteen smaller windows having a dimension of 4 pixels by 4 pixels. This representation is very simple and inexpensive in computing time. It also allows to absorb a certain variant of the writing: the density of each small window allows to smooth the irregularities avoiding a spatial description too exaggerated that may be prone to noise.

As per [36], each character is determined by the probabilistic model, taking into account both of the recognizer outputs and the geometric confidence besides with logical constraint. The global optimization of this method is conducted to find an optimal cutting path, taking a weighted average of character confidences as objective function. Experiments on handwritten Arabic documents with various writing styles show the proposed method is effective.

As per [15], the authors have presented a method for performing offline writer identification by using K-adjacent segment (KAS) features in a bag-of-features framework to model a user’s handwriting. This approach achieves a top 1 recognition rate of 93% on the benchmark IAM English handwriting dataset, which outperforms current state of the art features. Results further demonstrate that identification performance improves as the amount of training samples increase, and additionally, that the performance of the KAS features extend to Arabic handwriting found in the MADCAT dataset.

M. Edwin et Al [9] have presented a novel model that allows to represent a handwritten character into string graph representation. This type of graph consists of several edges that indicate the connected vertices. The vertices are the curves that make the character and the curve is extracted by analyzing the character's chain code, and it's string feature is created using certain rules. The similarity distance between graph is measured using approximate subgraph matching and string edit distance method. The recognition experiment conducted by comparing both methods on alphabet and number character images taken from ETL-1 AIST Database. The experiment results obtained validate the effectiveness of this approach.

IV. Preprocessing

A. Cleaning of the background

Binarization is a crucial task for document analysis method since such methods rely on features extracted from foreground pixels. Hence, an inaccurate binarization tends to propagate the noise through the whole system.

Text binarization has been used in many applications such as text observation, segmentation, character detection. It allows us to eliminate noise, faint characters and to ease the feature extraction. It is then necessary to overcome these tasks to increase the correct detection rate.

1) Modeling of estimation criterion: We propose a new image binarization method that uses a simple standard deviation approach and gives us very good results. It allows to remove large signal dependent noise in the background (shadows, non-uniform illumination, low contrast etc.) in order to maintain the dynamic grayscale images modeling the cursive writing strokes to be used in the latter stages of character recognition.

For this, we assume that the gray scale image of the document is divided into two classes C0 and C1 containing N0 and N1 pixels respectively. Each class is modeled by a Gaussian distribution with a0 and a1 are the mains and t the standard deviation. We wanted to determine the threshold S0 which minimizes the probability of error:

For this, we assume that the grayscale image of the document is divided into two classes C0 and C1 who contain N0 and N1 pixels respectively. Each class is modeled by a Gaussian distribution with a0 and a1 who are the mains and t represent the standard deviation. We wanted to determine the threshold S0 which minimizes the probability of error:

$$Pe = Pe_0.P(C_0) + Pe_1.P(C_1) \qquad (1)$$

with P(C0) and P(C1) are respectively the probabilities of belonging to the C0 and C1 class.

$$\text{With } Pe_0 = \frac{1}{\sqrt{2\pi t}} \int_{S0}^{\infty} f(S)dS \qquad (2)$$

$$f(S) = exp^{(-(S-a0).(S-a0)/2.t)} \qquad (3)$$

and Pe_0 is the number of pixels belonging to class C0 , but detected in class C1 and

$$Pe_1 = \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{S0} f(S)dS \qquad (4)$$

with

$$f(S) = exp^{(-(S-a1).(S-a1)/2.t)} \qquad (5)$$

and Pe_1 is the number of pixels belonging to class C1 but detected in class C0.

Minimize Pe, with respect to s, amounts to calculating the minimum threshold that minimizes the error probability where:

$$\frac{dPe}{dS} = 0 \Rightarrow S_0 = \frac{a0+a1}{2} + \frac{t}{a1-a0}.ln\frac{P(C0)}{P(C1)} \qquad (6)$$

Algorithm: We describe below the algorithm for finding a threshold for two class C0 and C1 corresponding to the background and the dynamics of different stroke shapes used to form characters

```
$S_0 \leftarrow 0$
$S_1 \leftarrow$ nbag   /* nbag is the average number of gray in image*/
While $S0 \neq S1$?Do
Begin
   Threshold (C0, S0); Threshold (C1, S1)
   Calculate $a_0$ , $a_1$ ,P(C0) , P(C1), t
   $S_0 \leftarrow S_1$
   $S_1 \leftarrow \frac{a0+a1}{2} + \frac{t}{a1-a0}.ln\frac{P(C0)}{P(C1)}$
End
```

The calculation of P (C0) and P (C1) are recurrent, these two forms can be obtained in a more general manner by:

$$P(R_k) = n_k / n \qquad (7)$$

where R_k is the kth gray level and n_k is the number of pixels whose gray level is R_k and n is the total number of pixels in the image.

B. Image transformation into skeleton

Our character representation model includes a hierarchical level named structural level, which is composed of elementary structures (straight lines, arcs, curves...). The skeletonization process in this case is necessary because it allows to pass from the visual perception of image to the structural perception of the image.

There are many methods of thinning [35], we have chosen for the case of text manuscripts, the method of Hilditch [39], and we made some improvements to this method:

-The homotopy in Hilditch algorithm [13] is not assured, we then traveled the image with a structuring element (3 * 3) and verified the correspondence between the neighbors of a given pixel (value 0) to the configuration of the structuring element with 8 rotations and in case there is a match, the pixel will be set to 1. The selected structuring element is of the form:

0 0 0
* 1 *
1 1 1

-The Hilditch algorithm generates many feathers, we then calculated the distance for each leg of the skeleton between the extremity of the branch and its junction with another branch. We then set a threshold and we removed all the branches whose length are less than the threshold corresponding to small parts of the skeleton.

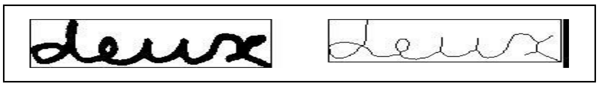


Fig.2. Skeletal characters

C. Detecting baseline

Process baseline is useful to detect the line which the letters connect to each other.

1) Hypotheses: The problem of regularity of handwritten text has been extensively studied by many authors and many satisfactory solutions have been proposed [12].

The manuscripts are characterized by lines that’s the length is more or less fluctuating. Their treatment need to overcome several difficulties that are: nesting lines, overlapping characters, deformation of ligatures, presence of heads and jambs, fragmentation of character due to binarization etc .

To determine the baselines of handwritten text, we define some hypotheses that will serve us as a starting base for research alignments lines:

-The text can contain multiple lines.
-There is no overlapping of characters
-The text may contain distortions of ligatures.
-Presence of legs (lower, upper).
-A connected component is not necessarily a character or word.
-The text was supposed already binarized and skeletonized.
-To this stage of processing, the characters are not yet segmented..

2) Process description: For the detection of the basic lines of text, the best solution is to cut the text using the RLSA method (Run Length Smoothing Algorithm). Such a method is very relevant to regions containing printed characters because the character components are evenly spaced and aligned, however, it cannot apply to handwritten characters. That is why we proposed some heuristics to solve this problem.

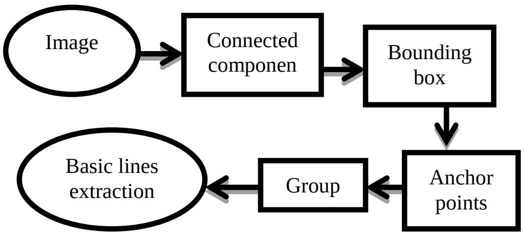


Fig.3. Sequence of steps involved for Identification and creation of baselines

The process of detecting of the text baseline passes through several stages:

Stage 1: Extraction of related components
We use for this purpose a recursive algorithm.
-Pick any pixel p in the image and assign it a label.
-Assign same label to any neighbor pixel adjacent to p.
-Continue labeling neighbors until no neighbors can be assigned this label.
-Choose another label and another pixel not already labeled and continue, if no more unlabeled image points then stop.

Stage 2: Calculation of bounding boxes
We chose the method of rectangles enclosing because all related components have a uniform shape (text is a sequence of homogeneous characters).
The minimal bounding rectangle chosen here is a minimum area rectangle with an arbitrary orientation, that contains the vertex set S of each connected component. It has two pairs of parallel edges, given by the max and min over S of two linear expressions

$$f1 = \text{a1.x+b1.y} \tag{8}$$

and

$$f2 = \text{a2.x+b2.y} \tag{9}.$$

In 2D, there is a well-known algorithm for finding the minimum rectangle using a technique referred to as the "rotating calipers [28]. This technique applies to a convex polygon, and can find the minimal rectangle in O (n) time. For a non-convex object, one must first compute its convex hull, which can be done in $O\,(n\log - n)$ time.

Stage 3: Filtering of anchor points
The connected components having a the connected components having a privileged direction are considered as anchor points .For the identification of these points we were inspired by the human visual system [28]. Indeed, the configuration of receptor fields of some cells makes them sensitive to the orientation of the line segments. By analogy with the receptive field, we define masks discretizing the space of connected components in different directions.
Each mask contains an excitation zone noted e_m sensitive to the writing density in the area.
Let Is C_i a connected component and Si its bounding box, the response of a connected component i to mask m is given by:

$$A_m(C_i) = P(C_i, e_m) / P(C_i, S_i) \tag{10}$$

with $P(C_i, S_i)$ is the number of pixels of C_i in the component bounding S_i.
The anchor points are then the components having a sufficient response $(A_m(C_i) \geq \alpha$, α being a chosen threshold), which correspond to shapes of homogeneous connected components

Stage 4: Group
The anchor points are at the origin of the alignments. At an anchor point , we associate the neighboring components located on each side to form a group.
Let C1 and C2 be two connected components with their bounding boxes (X_{min1} , X_{max1} , Y_{min1} , Y_{max1}) and (X_{min2} , bounding boxes (X_{min1} , X_{max1} , Y_{min1} , Y_{max1}) and (X_{min2} , X_{max2} , Y_{min2} , Y_{max2}) with X_{min1}<X_{min2} , the grouping rules are defined as follows:

Table 1. Grouping rules for anchor points

Condition	Type of fusion to perform
$X_{max1} > X_{max2}$	Complete recovery
$X_{min2} - X_{min1} < s$	Shift left
$X_{max2} - X_{max1} < s$	Shift right
$X_{max1} - X_{min2} < s$	partial recovery

of group .The best compromise combining the over-segmentation and sub-segmentation was obtained in our case with

$$s = | s_{max} - s_{min} | \tag{11}$$

where

$$s_{min} = (X_{max1} - X_{min1}) / 2 \tag{12}$$

and

$$s_{max} = (X_{max2} - X_{min2}) / 2 \tag{13}$$

V. Normalization

In our case, the normalization process for handwriting recognition comprises two main stages:

Stage 1: Removing diacritics and punctuation
By studying the positions of the encompassing rectangles with respect to the baselines, we can deduce that all the connected components situated above the seating line are considered as diacritics (accent…,) and connected components located above are considered as punctuations, It is preferable to delete them. Some manuscripts contain special separators, it is also necessary to remove them, for that we can say that a connected component Ci is

considered as one of these separators if it satisfies the inequality: connected components located above are considered as punctuations, It is preferable to delete them.

Some manuscripts contain special separators, it is also necessary to remove them, for that we can say that a connected component Ci is considered as one of these separators if it satisfies the inequality:

$$Max\ \left(w(C_i),\ h(C_i)\right) /\ Min\left(w(C_i), h(C_i)\right)\ \geq\ s \qquad (14)$$

with w and h successively the width and height of the connected component Ci, s is a threshold that can take values 5, 6 ...

The choice of these values depends on certain parameters related to the connected components such as the total number of connected components, the horizontal distance separating two related components etc. All these related components will no longer be retained in the rest of our treatments.

Stage 2: Baseline slope correction

It is considered as a major task and important factor in optical character recognition. Handwriting, containing skewed text lines can disrupt the definition of areas and the value of features extracted from these lines. An overall correction of the slope is insufficient because it assumes that writing is always based on a straight line. For the systems which cut lines into words, an estimation and a correction of the skew by cutting is possible [3] It is this principle that we will base ourselves here. The proposed skew detection method is based on straight-line fitting. It used on "the nearest neighbors and of center gravities" with the concept of centroids .After the relations between the successive centroid in every text line within a suitable sub-region were analyzed, the centroids most possibly laid on the baselines are selected as samples for the straight-line fitting.

We describe below the algorithm from these steps:

-The reading of the image containing the manuscript and its conversion into a binary image

-Calculation of connected components

-Computation of horizontal and vertical histograms for each connected component

-Obtain the centroid and its neighbor for each of the connected components

-Detecting the angle of inclination

-Calculate the average of all angles and rotate the image using obtained angle

We noticed that this task produces significant gains in the experimentation of skew correction.

VI. Proposition of a Model for Strips of Character Prototypes

The structural representations of forms based on the graphs are popular for the recognition of image structures because they allow modeling very finely the structural dimensions and topological of objects. In some areas of graphic imagery, as in recognition of symbols and chemical formulas, these techniques have found a boom in recent years.

The representations based on graph models [5] have the advantage of preserving the topological and dispositional properties of internal segments present in the writing. We have chosen to represent the structural properties of writing by focusing on its skeleton which alone contains all the morphological indications that we wish to retain

The vertices of the graph correspond to the structural points extracted by analyzing local configurations from the skeleton and edges. Contextual morphological information of each vertex in relation to its nearest neighbors is described in the context of forms (SC: Shape Context) [20] which captures the distribution contours neighborhood of a point in a rich and informing vector.

This descriptor has been exploited in similar contexts for recognition of graphic forms [2], [7]. It is here associated with each vertex of the graph. The edges of the graph are also described by the length of segments, estimated from the counting skeletal points between two adjacent peaks.

A. Formal definition

The handwriting is often ligated and their graphic is unequally proportioned due the variability of writers. This usually requires an adequate form of representation and ad hoc contextual knowledge to guide the reading and understanding of these characters. For this, we propose a relevant model, of which we will explain all the specificities.

The proposed graph model can be defined as follows:

$$G = (S_m, S_h, f, g, ad)$$

where $\mathbf{S_m}$ represents the peaks corresponding to positions named "structural points", $\mathbf{S_h}$ is the set of shapes between different points linked to each other, **f** is the function that associates with each vertex its context shape (SC : 5 concentric levels and 12 angular sectors then we have 60 dimensions), **g** is the function that associates to each graph shape a length expressed in number of pixels (estimated between two peaks) and **ad** is the function that returns for a given vertex a set of adjacent vertices placed continually on the same trajectory .

This same definition is valid also for character prototypes as shown in the following figures:

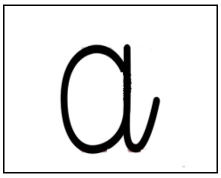


Fig.4. a. Prototype of character

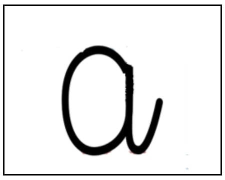


Fig.4.b. The same character after removal superfluous parts (faint parts)

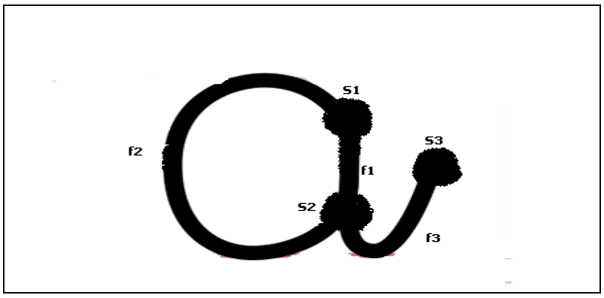


Fig.4.c. Character representation in our model

B. Concepts

- ✓ Notion of structure points: We distinguish here three types of structure points in each graph:

-Extreme point which defines the extremities 'start' and 'end' of character components (S3 in Fig 4.c).

-Crossing point with the portions of forms defining the character (S1 and S2 in Fig 4.c).

-Point of curvature which defines the contour of a shape linking two points (extremum or crossing) between them (f1 and f2 in Fig 4.c).

✓ Chain associated to graph:This is the sequence defined from different alternations between "put the pen" and "lift the pen" in the writing of the manuscript character. In the previous example, the string associated with the graph is S2 f2 S1 f1 f3 S3 while the two strings S1 f2 S2 f1 and f1 S3 S3 by combining them, also represent a chain associated with the graph but the latter requires the "put to pen" to pass from first to second sequence.

Note : The same model can also be applied for a part taken by cutting from a given character (notion of the strip).

C. *Hierarchical levels of model*

Like the other models based on hierarchical description [17], the proposed model has three levels of perception:

-Internal level: It corresponds to the higher and lower edges of characters and describes a character as a sequence of segments extracted from the character image.

-Intermediate level: It uses a Structured schema named “structural representation” constituted by a typology of elementary forms such as closed loops, lines, inflection points, intersection points, endpoints...

-External level: It defines a symbolic and visual description, including flagpoles, jambs...

D. *Main features*

Features associated to shape characters have a determining role in the character recognition problem [6].

In this context, we denote by F1, F2 ... F6 the different attributes defined in our model which will allow us to structure the stroke representations of characters.

F1 (Position): It describes the location of each point belonging to the graph. For any point p(x,y) in the plane, this feature is defined by:

$$Pos(p) = \int_0^l \sqrt{f_x^1(t) + f_y^1}\,(t)dt \qquad (15)$$

where f_x^1 and f_y^1 are respectively the x-axes coordinate and y-axes coordinate

F2 (Curve)**:** The position is not sufficient to define correctly the edge of the graph of our model .We add this kind of feature which describes the curvilinear shape of a snippet .This feature is defined by:

$$Allure(p) = \int_0^l arctan\frac{f_y^{1'}(t)}{f_x^{1'}(t)}dt \qquad (16)$$

Where $f_y^{1'}(t)$ and $f_x^{1'}(t)$ are are respectively the x-axes coordinate derivative and y-axes coordinate.

F3 (Shape signature): For the graph of our model, this feature is inspired by the concept of shape context used in [20]. It is composed of a set of n vectors. Each vector admits a configuration with respect to an edge point, which is represented by a bidimensional polar histogram. For this, it is necessary to discredit the radial and angular distance of all vectors .We use a constant number of bins in the application of these two measures. For a point p_i, the shape context corresponding hi is given, using the relative coordinates of other n - 1 points, by:

$$h_i(k) = Card\{q \neq p_i such\ as\ (q - p_i) \in bin(k)\} \qquad (17)$$

where q is a point belonging to the graph and k denotes a bin on a radial sector in a polar plane or log-polar

The bins were uniformly chosen in the log-polar space. Therefore, greater sensitivity is achieved with the closest points. A point of contour corresponding to graph in our model is then characterized by two measures:

$$\theta_n = (nb_\alpha - 1) * \frac{\theta}{2.\pi} \qquad (18)$$

(radial angle)

$$r_n = (nb_r - 1).log(1 + r / r_o) / log(1 + r_m / r_o) \qquad (19)$$

(normalized radial distance)

Where nb_r and nb_α respectively, are the number of bins chosen to measure the rays and angles, r denotes the radial distance of a contour point at the reference point pi, r_m is the maximum distance of all configurations and r_o designate the mean radius calculated from n2 distances between the set of points describing the graph. We took 5 bins and 12 angular sectors with a total of 60 dimensions).

F4 (Bounding): The bounding feature considered here is obtained in the following way:

-Calculate the minimum bounding rectangle for the graph modeling the character

-Subdivide a white image with a numeral written in black to several blocks or zones which are square or rectangular.

-Counting the number of white pixels in each zone.

-Converting the image to a vector with the number of components equal to that of zones.

Each graph of our model representing a bounding will be characterized by a vector of 9 components.

$$Descriptor_{zoning}(G) = \begin{pmatrix} SquareNpixelsZ1\ , & SquareNpixelsZ2, & SquareNpixelsZ3\ , \\ SquareNpixelsZ4\ , & SquareNpixelsZ5, & SquareNpixelsZ6, \\ SquareNpixelsZ7, & SquareNpixelsZ8, & SquareNpixelsZ9 \end{pmatrix} \qquad (20)$$

With SquareNpixelsZi is the number of pixels equal to value 1 in the square number (zone)

F5 (Jambage): It defines the position on relative to baselines; this knowledge of character shape (or strip shape) allows to determine its class. The different values associated with this feature are:

1: for average character (or strip)
2: for character (or strip) containing an upper jamb
3: for character (or strip) containing a low jamb
4: for character (or strip) containing an upper jamb and a low jamb
5: for character (strip) containing a tall silhouette

F6 (Snippet curve chain)**:** The calculation of this feature is to browse the contour of the chain and generate structural points corresponding to strip shape. The sequence of these points must be observed in the order of the trace of writing (loop, bridge ...).

In the proposed model, the terminal and junction points are the nodes of the graph. The arcs are the union of the curvature points starting from a terminal point to the point nearest junction called shape in our case. We model this feature by the sum of the flow of the points which is an approximation of the length of the contour that generates this shape. For each shape considered, we calculate the total flow and we assign as label this value. More formally, if s is the strip shape to model by our graph and x_i an arbitrary point of shape, we have:

$$Flux(s) = \sum_{i \in s} \nabla f\,(x_i) \qquad (21)$$

Note that ∇ denotes the gradient operator and $\mathbf{f}$ **($\mathbf{x_i}$)** defines a vector field of a point $\mathbf{x_i}$ on the inside of the shape **s** [14]. The flow of the vector field associated with the graph is the sum of the flows associated with each form of the graph representing the character.

In sum, we summarize the proposed descriptors along with the dimensionality of each.

Table 2. Dimension of model features

Feature	*Dimension*
F1	1
F2	1
F3	60
F4	9
F5	1
F6	128

VII. Manuscript Recognition Process

A. Principle

The process of automatic handwriting recognition considered here is characterized by a process with two passes:

✓ The first pass cuts the text in small sections (strips).

Each strip is treated in sequence in a sliding window, then it is added to the strips already examined, if this set is recognized as receivable, it is submitted to the character recognition system by resetting all strips to empty state otherwise we progress the reading of strips. The method used for the character recognition is that of KNN (k nearest neighbors) which generates the k most probable strips. We iterate this process until a character is detected and in this case we pass the second pass..

✓ The second pass receives in input a trellis obtained as a result of the first pass. This trellis contains all the nodes obtained from the k probable characters provided to each time that a character prototype is submitted for recognition. We dispose all the combinations of decompositions that can lead to the construction of a character. We apply in this passage the Viterbi algorithm which generates the most probable character that is to say the most likely sequence of strips.

B. General algorithm

We present below the general algorithm of our approach:

```
1.  Take a line of text in the sliding window
2.  While it exists lines in the text do
3.  Begin
4.    Initialize variables
5.    Get a strip from sliding window
6.    While it exists empty strip do
7.      Begin
8.      Get a strip from sliding window
9.      EndWhile
10.   If there are no more strips to treat then
11.   Goto 1
12.   Else
13.   If the current strip is a ligature  then
14.     Begin
15.       Applying the Viterbi algorithm to identify the
16.       handwritten character
17.       Store the character,  thus identified in the line
          variable  (line is a string of words)
18.     End
19.   Else
20.     Begin
21.       Extract strip features (We use the Moore Algorithm)
22.       Determine k nearest neighbors to the current strip
23.       Insert the k candidate strips (closest to the current
          strip)  into the trellis by linking them with the
          current strip
24.       Prune all paths of strips that are not compatible
          candidate characters from the root to the
          reached in trellis
25.     End
26.   Goto 3
27. EndWhile
```

Note: The sliding window is cut into small strips where each strip is intended to contain a stroke of character.

C. Cutting on a fixed pitch

The cutting considered here is based on an implicit segmentation using a sliding window. Each character of handwritten text being cut into small sections equal to the surface of the slide window. This method achieves a high capacity of performance, provides robustness against the noises and the partial recoveries.

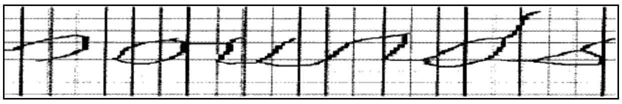


Fig.5. Extraction of small overlapping in strips

The principle is to move over the entire width of the sliding window the different characters of words constituting the text line to treat. This width is fixed in pixel size with a pitch smaller than the width of the window. We then define two parameters that are the width of the window and the overlap between two successive positions.

The different configurations of width and overlap have been adapted to the recognition chain because the choice of the widest window imposes significant overlap and induce a significant redundancy of information. Conversely, the choice of a window too small could eventually provide weakly discriminating attributes.

Each image is cut into **n** same size strips that can generate bending points of the graph of our model. To pass from one image to strips sequence is a problem of the choice of strip size.

These strips are not to be too small to be statistically significant and not too large so as not to exceed the size of a character. It is indeed important that a stripe is a subpart of a single character, this condition is necessary to build a word model built on the basis of the concatenation character models.

D. Learning for the recognition strip

The classification method chosen is the k-nearest neighbor method that is a classical supervised method in the field of matching learning based on statistical data. Several approaches have used this method in the resolution of the problem of the handwriting recognition problem [18],[25].

In our case, we choose this method for several advantages

-There are no explicit training phases or it is very minimal, once the date is loaded into memory it began to classify and this means the training phase is pretty fast.

-The data set has been labeled that is to say the selected neighbors of the test sample have been classified correctly.

-Independently of any data distribution, it only needs to be adjusted or assigned an integer parameter k.

✓ Data structures

- *Learning base:* It is defined by the following structure:

<CP, S, R, F1, F2, F3, F4, F5, F6, CL >

With CP: character prototype, S: strip stemming from the segmentation, R: Rang in the sequence of strips that makes up the character prototype, F1… F6: features of the strip and CL: class of characters corresponding to character prototype.

Note: we have 62 classes with 26 uppercase letters, 26 lowercase letters and 10 digits.

-*Character prototype base:* It is defined by the following structure:

< CP, NS, CL>

With CP: character prototype, NS: number of strips forming the character prototype and CL the class of characters corresponding to character prototype.

Example :

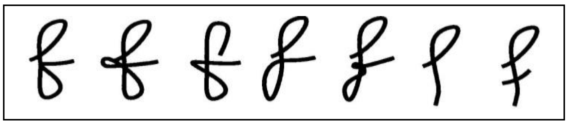


Fig.6. Different prototypes of character “f”

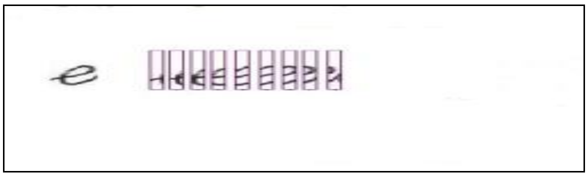


Fig.7. Image illustrating different strips constituting the character prototype “e”

The two bases described above, thus described constitute the ground truth; their construction procedure is described in the experimental part.

✓ *Metric for proximity*

The choice of metric plays an important role in the performance of the nearest neighbor classifier for a given sample size n.

Different distance metrics exist in the literature and we can be used when calculating the distance between the object points. As the set of features considered here are heterogeneous then we associate with each attribute type, the measure most suitable and appropriate.

We present below for each feature, the metric distance function for pairs of objects.

-F1: When we get the shape of character from the sequence of strips appearing in the window, we compare for each of these strips with all contour shapes located in the learning base. The distance metric for this feature is the average Euclidean distance between two points of the contour C1 and C2:

$$DF1(C1,C2) = \int_0^l \sqrt{(f_x^1(t) - f_x^2(t))\hat{}2 + (f_y^1(t) - f_y^2(t))\hat{}2dt} \quad (22)$$

-F2: The position is not a sufficient a clue to distinguish between two contours, we add the orientation which provides flexibility. The pairing using the notion of orientation uses the average angle between two spaces of contour C1 and C2:

$$DF2(C1\,,C2) = \int_0^0 |arctan(\frac{f_y^{1'}(t)}{f_x^{1}(t)}) - arctan\left(\frac{f_y^{2'}(t)}{f_x^{2}(t)}\right)|dt \tag{23}$$

-F3: Let P_i and P_j be, respectively, the point on the first shape and the second shape. The correspondence found using bipartite matching with costs defined by the distance between histograms [2]:

$$DF3\left(P_i,\ P_j\right) = \frac{1}{2}\sum_{m=1}^{n}\frac{\left(h_i(m)-h_j(m)\right)^\wedge 2}{h_i(m)+h_j(m)} \tag{24}$$

Where hi(m) and hj(m) denote the n-bin normalized histogram at Pi and Pj , respectively.

While the cost DF3ij can include additional terms based on the local appearance similarity at the points Pi and Pj Where $h_i(m)$, $h_j(m)$ denote the n-bin normalized histogram .

-F4: The Euclidean distance between two vectors X and Y represents two bounding features for two strips BS1 and BS2 is:

$$DF4\left(BS1, BS2\right) = \sqrt{\sum_{i=1}^{9}\left(x_i - y_i\right)^2} \tag{25}$$

-F5: Given two images containing respectively the strips S1 and S2, the distance involving the jamb attribute is defined as follows:

$$DF5\left(C1,C2\right) = \begin{cases} V1\ if\ \ S1\ and\ \ S2\ are\ medians \\ V2\ if\ \ S1\ and\ \ S2,\ each\ have\ an\ upper\ jamb \\ \ but\ no\ low\ jamb \\ V3\ if\ \ S1\ and\ \ S2,\ each\ have\ a\ low\ jamb \\ but\ no\ upper\ jamb \\ V4\ if\ \ S1\ and\ \ S2{,}each\ have\ a\ low\ jamb \\ \quad and\ \ a\ upper\ jamb \\ \infty\ \ if\ \begin{pmatrix} one\ of\ \ strips\ has\ an\ upper\ jamb \\ while\ the\ other\ has\ not \end{pmatrix} or \\ \qquad \begin{pmatrix} one\ of\ \ strips\ has\ a\ low\ jamb \\ and\ \ the\ other\ \ has\ \ not \end{pmatrix} \end{cases} \tag{26}$$

The different values V1, V2, V3 and V4 are chosen such that $V4 < V2 < V3 < V1$

-F6: This feature describes the silhouette of strips using three main elements that are extreme point (begin and end of a shape), crossing point (starting point of a shape) and structure point (line segment, bow ...). The basic rules about matching associated with two graphs St1 and St2 use a measure based on a set of operations (replace, insert, delete, merge, split and replace 2 pairs of strips). For each operation is associated a cost Ct defined as follows:

$$Ct\left(St1\ ,\ St2\ \right) = \begin{cases} 0\ \ if\ \ the\ two\ strings\ are\ identical \\ V\ \ if\ \ the\ two\ strings\ are\ similar \\ V3\ if\ \ S1\ and\ S2,\ each\ have \\ a\ \ low\ jamb\ but\ \ no\ \ upper\ jamb \\ \infty\ \ Otherwise \end{cases} \tag{27}$$

The distance defined for this descriptor is then

$$DF6\left(St1,\ St2\right) = \sum_{i=1}^{l} Ct_i\left(S_k\,,S_m\right) \tag{28}$$

With

$$Ct_i\left(S_k\,,S_m\right) = \left|flux\left(S_k\right) - flux\left(S_m\right)\right| \tag{29}$$

Where Ct_i is the cost corresponding to the ith operation performed during the process of comparing of strings St1, St2 and k, m are two forms (substrings) taken from chains St1 and St2.

Note:

-For each string is associated a flow of a vector field that is the sum of the flow of each form (point of the structure) of the graph modeling the strip.

-All the features presented above are also applicable at the global level for character prototypes.

- ✓ *Modeling of the learning process* : Let S = $(S_1,S_2\ldots S_n)$, the set of the strips of character prototypes where each S_i is described by a set of features and let $C(S_1)$, $C(S_2)$ …$C(S_n)$ the classes associated with each these elements S_i and $Sim(S_{win},S_i)$ a strictly positive function measuring the similarity between a strip of a character prototype extract of the sliding window S_{win} with its corresponding in the learning base (S_i).

When a new small strip S_{win} arrives at the base of examples this strip is compared using a distance, with all examples of the base in function to attributes defined (6 in number). We determine the k closest examples, each of then k examples voting so that the strip is stored in its own class and the class that received the highest votes “wins” and will be selected. Hence the chosen class can be expressed as:

$$Strip\_selected\left(S^*\right) = \underset{S_j\ \in\ Base}{ArgMax}\ Sim\left(S^*,S_j\right) \tag{30}$$

With

$$Sim\left(S^*,\ S_j\right)\ = \Sigma(w_i * D_i\left(S^*,\ S_j\right) * \delta\left(C\left(C_j\right),\ i\right)) \tag{31}$$

Where w_i , the weights assigned to the various features , D_i define the distances described in section V (paragraph D) and δ the Kroneker function that's equal to 1 if both arguments are equal, 0 if not. We choose the class Class i whose vote is equal to ArgMax.

Note: The values of w_i depend on the features of S_{win} , each feature has its own distance.

E. Ligature detection

- ✓ Principle: Our ligature detection algorithm uses two kinds of filters:

-The first filter, which consists to select, according to a set of attributes in learning database (associated with a set of ligatures constituting a benchmark database), the closest ligature to strip extracted from the sliding window.

-The second filter allows to take the selected strip (strip candidate) and verify according to a heuristic rule if this strip is a ligature or not.

- ✓ Ligature features: Ligature is a small stroke that is used to connect joined/cursive characters.

The proposed method for ligature identification does take into account the detailed facts of the ligatures strokes and its structure according to the attributes that describe it.

The proposed features are extremely simple to extract and analyze. These features are suggested:

-A1: Measurement from side to side of ligature strip
-A2: Measurement for the tallness of ligature strip
-A3: Previous strip
-A4: Following strip
-A5: Density feature of shape strip (number of pixels divided by the window size of shape strip)
-A6: Aspect ratio (Ratio of height divided by the width of a ligature strip)
-A7: Sum of side pixels of a ligature strip
-A8: Number of sides of a ligature strip
-A8: Product of width and height of ligature strip
-A9: Perimeter to area ratio (Division of sum of side pixels by area of ligature strip)
-A10: Slope between start and end point (The slope of the diagonal line when connecting the start point to the end point)

- ✓ Structure of database learning (Benchmark database) of ligature strips: This database is constructed from a set of ligatures extracted from 2345 Latin manuscript texts, 970 of which are written by a set of writers we have chosen. Each ligature strip is an element of this database , it is structured as follows:

<Ref, A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,Class>

Where *Ref* is an address of the ligature strip location, the A_i represents different features of ligature strip and *Class* the class of characters to which the ligature belongs.

- ✓ Strip ligature identification algorithm: The disclosed system captures the ligature information by describing the previous and the following strip using the features that describe them in association with the features of strip analyzed. A greater degree of recognition accuracy is therefore attained by integrating the information found in the center of the current strip with the "grouping" information of the neighboring strips (The features “previous strip" and "following strip" are two very interesting indicators in the identification of ligatures).

Our algorithm uses the concept of vertical histogram that is calculated by counting black pixels in each strip horizontal line. The vertical histogram of a binary strip, in our case, is sufficient to distinguish the difference between a ligature strip and a non-ligature. In this context, a vertical histogram is used to identify the position of ligature in sequence of strips forming the character.

The problem of “touched /overlapped characters” caused for miss-segmentation errors are very hard to deal with ligature detection algorithm. Where two characters are tight together ligatures cannot be found and therefore they cannot be segmented. Nevertheless, touching characters is a rare case in cursive handwriting. Similarly, the problem of overlapping characters has been dealt with in one of our preliminary treatments. Hence, overall correct segmentation results are not significantly affected.

Note: the value of feature F8 is defined by:

$$Ct\left(St1\ ,\ St2\ \right)=\begin{cases}0\ \ if\ \ strip\ is\ composed\ by\ a\ single\ side\\ \\ \infty\ \ if\ \ strip\ is\ composed\ by\, or\ more\ sides\ a\ ligature\ strip\\ \end{cases} \tag{32}$$

Algorithm description:

```
-Input: *Learning database consisting of m strips with their
descriptions (A sample of m strips with their features)
        *The strip X to be identified (ligature or not)
-Output: Deciding whether a strip is a ligature or not
Begin
-Calculate the features of strip X
-Determine the nearest strip of X by calculating the distances in
database learning of strips (first filter)
-Construct the vertical histogram of the strip X
-Take the previous (XPREV) and following (XFOL) strip of X in
learning database
-Construct the vertical histogram for XPREV and XFOL
-Calculate average (number of pixels) of XPREV and XFOL
-If number of pixels of X is small to average number of pixels of
XPREV and XFOL, it means that X is a strip ligature (second filter)
End
```

Note:

-The baseline detection: we assume that the junction between characters is usually located on the baseline, which is a fictive line used by writers to align their handwritings.

-Vertical histogram for any ligature strip is computed after taking the sum of pixels along each row.

F. Generation of trellis

The above algorithm allows to generate a structure in the form trellis containing all the possible sequences of the most probable strips (candidate strips). This structure is created gradually by adding to each execution of the algorithm the k edges corresponding to the k-nearest neighbors.

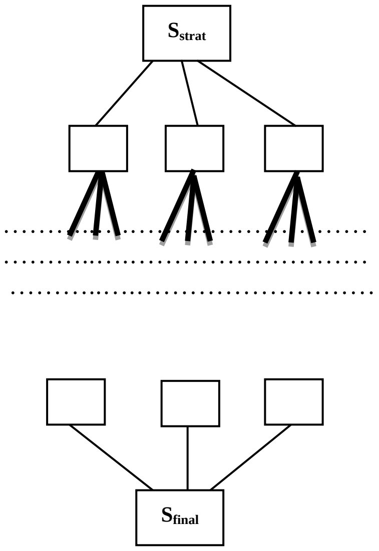


Fig.8. Trellis contains the k nearest neighbors of strips

From each vertex of level i of this trellis, a set of arcs goes to other vertices of level i + 1. These arcs are weighted with scores computed from K-nearest neighbor classification and allow to determine the k candidate vertices.

G. Ultimate step of the recognition

✓ Na ïve method: The dashed paths in the trellis are the different strips pruned, that is to say the strips which do not correspond to character prototypes. The other paths remaining in this trellis , then represent sequences of strips which are candidates and which may constitute the character to be recognized. The depth of this proliferation of nodes in the trellis corresponds to the number of sections resulting from the segmentation process of a considered character.

```
Score_previous ← 0
For each sequence of candidate strips s extract from trellis do
Begin
Calculating cumulative scores for the sequence s in score_s
If score_s > Score_previous Then
Begin
Save the sequence s in the variable result
score_previous ← score_s
EndIf
EndFor
```

The set of strips which locate in the variable ***result*** represent the desired character.

✓ Viterbi method: The Viterbi algorithm allows to determine for a given sequence of candidate strips ***s*** if it is or is not a valid sequence by calculating an optimal path from a trellis containing all probable solutions of strip paths.

To describe the algorithm for our type of situation we need two matrixes:

a) Matrix of strip transition:

Table 3. Matrix of transition probabilities

	S_0	**S_1**	**S_2**	**…**	**S_j**	**S_{i+1}**	**…**	**S_m**	**S_{m+1}**
S_0									
S_1									
S_2									
…									
S_i					**a_{ij}**				
S_{i+1}									
…									
S_m									
S_{m+1}									

With Si, the state of transition to the **ith** strip and **a_{ij}** the probability of transition from the ith strip to the **ith** strip, this probability is denoted by **P (S_i \ S_j)**.

b) Matrix of emission:

Table 4. Matrix of emission probabilities

	S_0	**S_1**	**S_2**	**…**	**S_j**	**S_{i+1}**	**…**	**S_m**	**S_{m+1}**
C_0									
C_1									
C_2									
…									
C_i					**b_{ij}**				
C_{i+1}									
…									
C_{61}									

With **C_i** , the set of lowercase letters, upper case letters and digits and **b_{ij}** , the emission probability of the ith character to the **jth** strip in the sequence, this probability is denoted **P (C_i \ S_j)**

c) Search algorithm to find the optimal sequence of strips:

```
Initialize α_0(S) for S the start state and α_0(S)=0 for all other states
For i: = 1 to n do   /* n is number of strip constituting the Character to be analyzed */
Begin
Calculate
    $\alpha_i(S) = Max_{Si-1}\ P(S_i \setminus S_{i-1}) P(_{Ci-1} \setminus S_{i-1})\ \alpha_{i-1}(S_{i-1})$   (33)
```

```
Calculate
    $\beta_i(S) = ArgMax_{Si-1}\, P(S_i \setminus S_{i-1}) P(_{Ci-1} \setminus S_{i-1}) \alpha_{i-1}(S_{i-1})$    (34)
EndFor
From the end state we can read off the Viterbi sequence (following the back traces through to the start state) and its probability.
```

This algorithm requires an amount of storage proportional to the number of states and an amount of computation proportional to the number of transitions.

Example: The image given below represents a sequence of strips resulting from the cutting of the prototype letter "d".

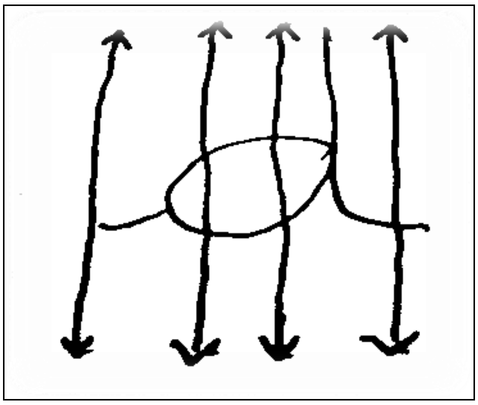


Fig.9. Cutting of letter “d” (prototype) in 3 strips

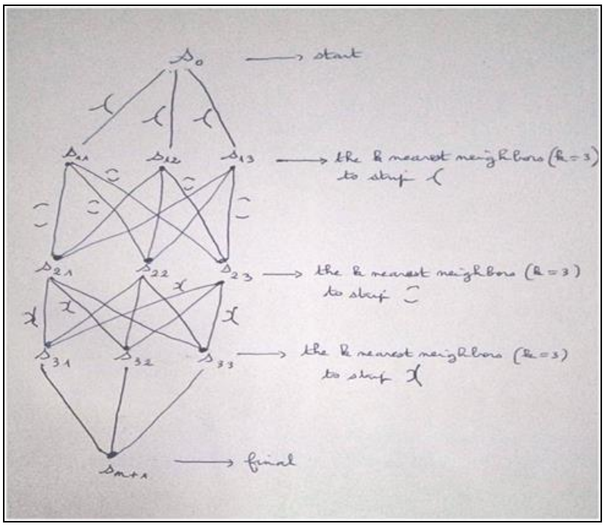

s0 → start
s11 s12 s13 → the k nearest neighbors (k=3) to strip (
s21 s22 s23 → the k nearest neighbors (k=3) to strip ⁐
s31 s32 s33 → the k nearest neighbors (k=3) to strip (
sm+1 → final

Fig.10. Trellis containing strip sequences similar to the different strips of the character “d”

The set of strips most similar to the input sequence (three strips extract for cutting process) is given by:

S_0S_{11} ,S_0S_{12} , S_0S_{13} , S_0 S_{11} S_{21} , S_0 $S_{11}S_{22}$, S_0 $S_{11}S_{13}$, $S_0S_{12}S_{21}$, $S_0S_{12}S_{22}$, $S_0S_{12}S_{23}$, $S_0S_{13}S_{21}$, $S_0S_{13}S_{22}$, $S_0S_{13}S_{23}$, .)..

The insertion of the new candidate strips deep into the trellis continuously generates several proliferations of strips. A solution to this problem is to add a pruning procedure which eliminates all strips paths except the paths whose corresponding strips belong to the same character. This solution simplifies the structure highlighted and makes it more compact.

The set of strips most similar to the input sequence is given by:

s_0 s_{11} $(s_{21})_1$, s_0 s_{11} $(s_{22})_1$, s_0 s_{11} $(s_{23})_1$, s_0 s_{12} $(s_{21})_2$, s_0 s_{12} (s_{22}) 2, $s_0s_{12}(s_{23})_2$, $s_0s_{13}(s_{21})_3$, s_0 s_{13} $(s_{22})_3$, s_0s_{13} $(s_{23})_3$}.

The insertion of the new candidate strips deep into the structure continuously generates prohibitive proliferations of strips. A solution to this problem is to add a pruning procedure which eliminates all strips paths except the paths whose corresponding strips belong to the same character. This solution simplifies the structure highlighted and makes it more compact.

VIII. Experimental Evaluation

A first experimental study allows us to measure the performance of our algorithm for handwritten character recognition. The proposed algorithm uses a successive cutting of each prototype of character in equal parts (sections) until we reach the ligature linking two juxtaposed characters. Our approach is based on the use of learning base developed manually from manuscripts randomly chosen of the Rimes base. Each prototype of the character is segmented into tiny sections and stored in the database with all these attributes.

A. Choice of training corpus

We have chosen the Rimes database which is a database of documents written by hand. These documents are letters intended for public Administrations. The letters were produced by thousands of volunteers following the various scenarios: change of the personal information, request for information, opening or closure of account, contract amendment, command, complaint, payment difficulties, reminder letter or declaration of a disaster. It was created to provide data relative to advanced mailrooms, and the panel of documents offers a large variety and makes the database a challenging one. These letters are written on white paper and then scanned at a resolution of 300 dpi, which claimed to reduce background noise.

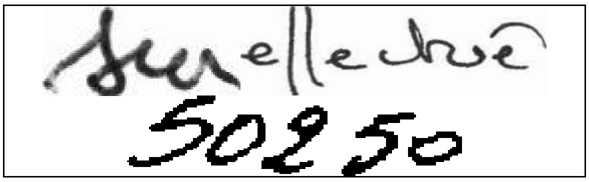


Fig.11. Three images taken from database Rimes

B. Benchmark Datasets with Ground Truth

The high variety of writing styles increases the effort needed to collect a representative dataset.

In [17], the quality of a handwritten character recognition system depends on the kind of learning base used. In this context, we associate, to this learning base, a ground truth which is considered as an excellent practice to characterize assessment tasks and entire assessments as having the potential to elicit evidence for our approach.

There are three types of methods to create a ground truth for this kind of evaluation [4]:

-The methods which allow to generate synthetic models based on a process model collection, which enrich the learning base with prototype characters.

-The most widespread methods for manually annotating of corpora

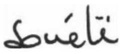
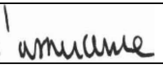
-The specialized methods in the creation of perceptual information, such as color and shape, which are often related to the quality of an image (qualification of the degradation, legibility …).

For our case study, we have chosen the manual annotation of our own learning base obtained from the base rhymes.

a) Construction of a first base: the lexicon

We have taken randomly 685 manuscripts of the Rimes database and are trying to build a lexicon from a set of words that we extract from this database. This task is easy to achieve, it is sufficient to take a word written in a form discursive and associate it with its corresponding typed.

Table 5. Inseparable characters (handwriting) extracted from a text with its corresponding writing in Latin letters printed

	soci ét é
	d’assurance

Note: here, the term "*d’assurance* " (equivalent to de assurance) is treated as a sequence of indivisible characters (Whole entity).

b) Construction of a second database: base of character prototype *images*

We always take the same manuscripts (685) of the Rimes database and we extract manually these *character prototype* images for bootstrapping a recognition system.

```
For each line L of the collection C Do
  Begin
    For each word W of the line L Do
      Begin
        Delimit and extract the characters from the word W
        Store them in the database with their corresponding
        class
      EndFor
EndFor
```

This procedure is long and tedious; it requires an effort of concentration .We took a lot of time to reach to create approximately 15,000 entries.

It should be noted that all the characters which have slight differences in the shape (form) are considered similar and will be associated with the same prototype of character (they will be assigned to the same class). There have, on average, 27, 34% of the set of the characters in the collection that represent these slight differences. The database constructed contains several instances of character prototypes with different shapes such that they are written on the manuscripts.

c) Construction of a third base: Base of “character prototypes fragmented”

This cutting operation of character prototypes is proving as extreme difficulty, especially with the objective of providing a fragmentation more refined.

```
For each entry in the base of character prototypes Do
Begin
  Take the prototype C
  While the cutting of the prototype C is always possible Do
    Begin
      -Obtain the fragment $F_i(C)$ for this prototype C
      -Calculate the features of $F_i(C)$
      -Store these features as well as the class of the
       prototype C in this base
    EndWhile
EndFor
```

This database contains all features defined in VI (D) that describe the different strips of all character prototypes considered in the database. The number of associated strips for each character is not the same for the others prototypes; we have an average of 10 for the set of prototypes which gives in any around 150000 entries for the learning base. The following fields illustrate the structure of the database:

(Character prototype , Shape of Fragment , Rang in the sequence of fragments that makes up the character prototype , F1 , F2, F3 , F4 , F5 , F6 , Class of printed character)

The specificities of this base is:

-All fragments have been standardized whether it is for those who have been selected from the Rimes database or those who are located in the learning base in order to ensure the homogeneity.

-The fragments of all prototypes have the same size

C. Evaluation parameters

The evaluation criteria that we have used are directly derived from the criteria used to assess the handwritten character recognition systems, namely Equal Error Rate (EER), precision and recall rates.

In our study, these rates are defined from this matrix of confusion:

Table 6. Confusion matrix

	Predicted positive	**Predicted negative**
Positive cases	Targets belonging to the "ground truth" classified "positive " by our classification algorithm (TP)	Targets do not belong to the "ground truth" classified "positive" by our classification algorithm (FN)
Negative cases	Targets belonging to the "ground truth" classified "negative " by our classification algorithm (FP)	Targets do not belong to the "ground truth" classified "negative" by our classification algorithm (TN)

With:

$$Precision = TP \ / \left(TP + FP\right) \qquad (35)$$

and

$$Recall = TP / \left(TP + FN\right) \qquad (36)$$

$$F - Measure = 2 * \left(Precision * Recall\right) / \ \left(Precision \ + \ Recall\right) \qquad (37)$$

It is possible to calculate all these indicators for each class of character prototype. The average of each class of these indicators provides global indicators on the quality of the classifier.

$$Precision \ = (1/k).\sum_{i=1}^{n} TP_i/(TP_i + FP_i) \qquad (38)$$

$$\mathrm{Re}\,call = (1/k).\sum_{i=1}^{n} TP_i/(TP_i + FN_i) \qquad (39)$$

$$F - Measure = 2 * \left(Precision * Recall\right) / \ \left(Precision \ + \ Recall\right) \qquad (40)$$

On the other hand, there are two ways of measuring performance in terms of “false acceptance rate “(TFA) and “false rejection rate “(TFR), defined as follows:

To validate our system for manuscript character recognition, we performed significant experiments according to two substantial parameters which are, the size of the sliding window (s) and the number of close neighbors strips (k).

Several tests have been carried out which finally allow to determine the best values of these two parameters in order to construct a robust recognition system.

We have varied the parameter **s** (size of the sliding window) from 5 pixels up to 10 pixels, while the number of close neighbors **k** received 1, 2, 3, 4 and 5.

Note: the values of s greater than 10 give bad results idem for k greater than 5 which give prohibitive results.

$$\mathrm{TFA} = Number \ of \ accepted \ characters \ \left(NAC\right) / \ Total \ number \ characters \ to \ be \ considered \qquad (41)$$

$$TFR = Number \ of \ characters \ rejected \ \left(NCR\right) / \ Total \ number \ of \ characters \ to \ be \ considered \qquad (42)$$

A perfect verification of identity (*NAC=0* and *NCR=0*) is not feasible in practice. However, any of these two rates (NAC, NCR), it can be reduced to a small arbitrary value by changing the decision threshold, with the disadvantage of increasing the other rate. A single measurement can be obtained by combining these two rates of errors in the rate error EER:

$$EER = \ \left(NAC + NCR\right)) / \ Total \ number \ of \ characters \ to \ be \ considered. \qquad (43)$$

Table 7. Calculation of Equal Error Rate (EER) from the different values of s and k

s	k	EER	Precision	Recall	F-Measure
5	1	08.80	70.31	72.58	71.48
	2	07.74	70.91	71.35	71.13
	3	07.00	71.53	68.27	69.86
	4	06.54	72.05	70.90	71.47
	5	06.19	72.40	69.15	70.73
6	1	06.22	72.15	70.50	71.31
	2	06.07	72.70	70.01	71.33
	3	06.00	73.22	71.87	72.54
	4	06.22	73.17	70.94	72.04
	5	06.11	74.29	74.65	75.08
7	1	05.51	77.90	75.46	76.66
	2	06.07	78.20	76.30	77.24
	3	05.27	79.10	78.71	78.90
	4	05.19	78.86	79.27	79.06
	5	05.08	79.84	80.53	80.18
8	1	05.74	77.76	78.09	77.92
	2	05.56	77.58	77.25	77.41
	3	05.71	77.91	76.55	77.22
	4	05.45	76.97	75.80	76.38
	5	05.39	78.10	77.64	77.87
9	1	06.28	64.54	65.85	65.18
	2	06.79.	65.70	63.90	64.79
	3	06.65	66.40	65.46	65.92
	4	06.37	65.88	62.25	64.01
	5	06.25	67.38	66.77	67.49
10	1	06.94	58.30	56.38	57.32
	2	06.80	55.04	56.65	56.20
	3	06.51	57.61	55.50	56.53
	4	06.48	57.45	57.00	57.22
	5	06.62	58.02	59.48	58.74

This part reviews some results obtained for the values of parameters s and k which represent the substantial criteria for our approach evaluation. The above table gives the recognition rates for each pair (s, k). A deep analysis of these results leads to the following observations:

-The large values of k parameter reduce the effect of noise on the classification and thus the risk of over-learning.

-The large values of parameter k make the borders between classes less distinct.

-The results obtained show that our approach achieves high performances when we choose s = 7 pixels with increasing values for k. (We do not discuss the optimal solution in this case).

-The related state of the art using different evaluation strategies shows that the manuscript recognition rate is always lower on the restricted learning bases; this is the case for our protocol experiments.

-Approach using “sliding windows” allows to avoid the problems encountered in the majority of current work and it also to absorb the disparities between the data.

-There are certain difficulties which represent a handicap for our approach. They relate to the recognition of certain letters such as the letter l with the letter b, v with w, etc. All these ambiguities represent a major handicap for our handwriting recognition approach.

-The equal error rate (EER) obtained is close to 6%, but it has been calculated with a relatively small base. This same rate can be decreased with great bases.

-Our results are slightly lower than the results observed in the state of art, for the simple reason that our approach is situated in a more general context , contrary to other approaches , some simplifications have been made in order to reduce the severity of difficulties and to make their systems more flexible and accessible.

IX. Conclusion and Prospects

We cannot establish a comparison of our approach with the solutions brought in the state of the art due to the corpora used in the training set, which were different from the corpora of the other works and especially the way in which they were treated.

The results obtained by our approach in a Multi scripter context of mixed writing are promising and encouraging. However, in order to achieve higher performance, several difficulties must be overcome:

-Reduce the number of characters missed (not recognized) by the system.

-Treat the confusions introduced by the amalgam of lowercase letters with upper letters without forgetting the digits.

-Define another parameter representing the overlap between two successive positions in the sliding window. This parameter allows to solve the problem of overlaps and entanglements between the different strips.

-Enrich the proposed model with the integration of new attributes allowing to increase our handwritten character recognition.

As prospects, we propose as solutions to the problems mentioned above the following:

-The first problem was dealt with in our study by collecting samples (prototypes of characters) for learning which is a long and costly step. It would therefore be very interesting to be able to artificially increase the number of prototypes available to better cover all the variants of the writing on the one hand and by increasing the number of algorithms put in combination on the other hand.

-The second problem can be dealt with by interacting segmentation and character recognition with cursive letters. To do this, several solutions can be envisaged. The simplest way is to systematically double the recognition of cursive letters by introducing a decision model placed downstream , which must make it possible to optimize the results by selecting the best candidate given the recognition scores obtained. Finally, the introduction of fuzzy logic should make it possible to improve the overall performance of the system by delaying the final decision in order to take account of all the information coming from the different tasks involved in the recognition process.

-As for the problem of enrichment of the proposed model, we recommend to add relevant features in agreement with the configuration of the proposed model without forgetting to adapt Hough transforms and Harris detectors to identify any shape, strip, parameterized. We are therefore working on these last points.

References

[1] T. Akiyama et al , (2004) , Handwritten Address Interpretation System Allowing for Non-use of Postal Codes and Omission of Address Elements , Proceedings of the 9th Int’l Workshop on Frontiers in Handwriting Recognition (IWFHR-9- 2004) , IEEE *Xplore.*

[2] S. Belongie, J. Malik & J. Puzicha. (2008), Shape Matching and Object Recognition Using Shape Context. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pages 509–522, April 2002.

[3] R. Bertolami and H. Bunke, (2008), Hidden Markov Model Based Ensemble Methods for Offline Handwritten Text Line Recognition, Pattern Recognition, vol. 41, pp3452- 3460.

[4] A. Borkar, M. Hayes, and M. T. Smith, (2010), An efficient method to generate ground truth for evaluating lane detection systems, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 1090–1093.

[5] H. Bunke and K. Riesen (2011). Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recognition*, 44(5):1057–1067.

[6] L. P. Cordella , C. D. Stefano, F.,Fontanella , C. Marrocco ,(2010), A feature selection algorithm for handwritten character recogntion . In Pattern Recgnition(ICPR) , the 19th International conference on , pages 1-4.

[7] G. Costagliola , M. de Rosa and V. Fuccella (2011) , Improving shape context matching for the recognition of sketched Symbols in Proceedings of Int. Conf. Distributed Multimedia Systems.

[8] L. V. Der Maaten , (2009), A new Benchmark dataset for handwritten character recognition , Technical Report TICC TR , Tilburg University.

[9] M. Edwin, W. Putra,(2015),Structural Off-line Handwriting Character Recognition Using Approximate

Subgraph Matching and Levenshtein Distance, Procedia Computer Science Elsevier, Vol 59 , Pages 340-349.

[10] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen and H. Bunke (2013) , A fast matching algorithm for graph-based handwriting recognition. In Lecture Note in Computer Science, volume 7887, pages 194–203.

[11] H. Fujisawa, (2009), Robustness design of industrial strength recognition systems, in: B.B. Chaudhuri (Ed.), Digital Document Processing: Major Directions and Recent Advances, Springer, London, 2007, pp. 185-212 , Wuthrich.

[12] S. Hamrouni , N. Vincent , F. Cloppet (2014) , Handwritten and Printed Text Separation: Linearity and Regularity Assessment International Conference Image Analysis and Recognition ICIAR 2014: Image Analysis and Recognition pp 387-394 , DOI: 10.1007/978-3-319-11758-4_42

[13] C. Hilditch (1969) , Linear Skeletons form square cupboards , In B , Mertzer and D. Michie , editors , machine intelligence , vol 4 , chap 22 , pp 403-420 University Edinburgh .

[14] R. Hu, M. Barnard and J. Collomosse, (2010) , Gradient field descriptor for sketch based retrieval and localization. In Proceedings of the 17th Int. Conf. on Image Processing, pages 1025–1028

[15] R. Jain and D. Doermann. (2011). Offline Writer Identification using K-Adjacent Segments. International Conference on Document Analysis and Recognition, pp. 769-773.

[16] A. L. Koerich, A. S. Britto Jr., L. E. S. Oliveira and R. Sabourin , (2006) ,Fusing high- and low-level features for handwritten word recognition. In Proceedings of the 10th Int. Workshop on Frontiers in Handwriting Recognition, pages 151–156.

[17] A. L. Koerich, R. Sabourin, and C. Suen, (2011), Large vocabulary offline handwriting recognition: A survey. Pattern Analysis and Applications, vol 6 pp.97–121.

[18] M. Kumar , M.Jindal , R. Sharma , (2011) , KNN based offline handwritten Gurmukhi character recognition . In Image Information Processing (IcIIP) , The International Conference on , pages 1-4 , .

[19] M.M. Luqman, J-Y. Ramel, J. Lladós, and T. Brouard, (2011), Subgraph Spotting through Explicit Graph Embedding: An Application to content spotting in graphic document images, in ICDAR, 2011, pp. 870–874.

[20] T-O. Nguyen, S. Tabbone and O. R. Terrades, (2008) ,Symbol descriptor based on shape context and vector model of information retrieval. In Proceedings of Int. Workshop Document Analysis Systems.

[21] O. K. Oyedotun, E. O. Olaniyi, A. Khashman (2014) , Handwritten Numeral Recognition using Feature Selection approach, International Journal of Intelligent Systems and Applications (IJISA)(MECS Press) , vol.6,no.12,pp.40-47,2014,DOI:10.5815/ijisa.2014.12.06

[22] R. Plamondon ,S.R. Srihari (2000) , on-line and off-line handwriting recognition : a comprehensive survey IEEE transacrtions on pattern analysis and machine intelligence , 22(1) ,63-84 .

[23] H. Pesch (2011), Advancements in Latin script recognition thesis, RWTH Aachen University, Aachen, Germany, Nov. 2011.

[24] T. M. Rath. (2005), Retrieval of handwritten historical document images. PhD thesis, University of Massachusetts Amherst.

[25] R. Rathi, R. Krishan , Pandey , M. Jangid, (2012) ,Offline handwritten Devanagri vowels recognition using KNN classifer . International Journal of Computer Applications , Vol 49 page 11-16 .

[26] M. Rusiñol, D. Aldavert, R. Toledo and J. Lladós , (2011) Browsing heterogeneous document collections by a segmentation-free word spotting method, In Proceedings of the 11th Int. Conf. on Document Analysis and Recognition, pp 63-67.

[27] D. Salvi , J. Zhou , J. Waggoner and S. Wang , (2013), Handwritten text segmentation using longest path algorithm . in Application of computer Vision (WACV) , IEEE Workshop on , pages 505-512 .

[28] L Schomaker, M, Bulacu (2004) , Automatic Writer Identification Using Connected-Component Contours and Edge-Based Features of Uppercase Western Script, vol. 26, pp.787-798, PAMI.

[29] D. V. Sharma , Books Computing , Internet Media, (2012) , Form Processing System for Hand-Filled Forms in Gurmukhi Script Paperback – Import,,

[30] I. Siddiqi and N. Vincent, (2008) , Combining Global and Local Features for Writer Identification, In Proc of 11th Int'l Conference on Frontiers in Handwriting Recognition (ICFHR), Canada.

[31] M. Sreeraj , M. I. Sumam, (2010) k-NN based On-Line Handwritten Character recognition system,First International Conference on Integrated Intelligent Computing.

[32] S. Uchida, R. Ishida, A. Yoshida, W. Cai, and Y. Feng (2012). Character image patterns as big data. In Frontiers in Handwriting Recognition (ICFHR), The 13th International Conference on, pages 479–484.

[33] A. Verma, P. Singh ,N.S. Chaudhari ,Devanagri Handwritten Numeral Recognition using Feature Selection Approach , I.J. Intelligent Systems and Applications, 2014, 12, 40-47.

[34] H. Wei , K. Chen , R. Ingold , and M. Liwicki , (2014) , Hybrid feature selection for historical document layout analysis. In Frontiers in Handwriting Recognition (ICFHR) , The 14th International Conference on , pages 87-92.

[35] A. R. Widiarti (2011) “Comparing Hilditch, Rosenfeld, Zhang-Suen, and Nagen-Draprasad -Wang-Gupta Thinning”, International Scholarly and Scientific Research & Innovation, Vol5, 2011.

[36] S. Wong, S. Uchida and M. Liwicki , (2011), Comparative study of part-based handwritten character recognition methods . In document Analysis and Recognition (ICDAR), <The 11th International conference on , pages 814-818.

[37] P. Xiu, L. Peng, (2006). ,Offline handwritten arabic character segmentation with probabilistic model. Document Analysis Systems VII, Springer: 402-412.

[38] Q. Xu, L. Lam, & CY. Suen, (2003) , Automatic segmentation and recognition system for handwritten dates on Canadian bank cheques in M. fairhurst & A downto (Edo) , proceedings of the seventh international conference on document analysis and recognition (pp 704-709) Edinburgh UK IEEE Computer Society Press.

[39] http://cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton.html

Authors' Profiles

Name: Ali Benafia:

Affiliation: University of Hadj Lakhdar (University Batna 1).

Address: 05 Avenue Chahid Boukhlouf, Batna (05000), Algeria.

Brief Biographical History: Ali Benafia got his Engineer degree from the University of Constantine; he also received his Masters and Ph. D. in computer science from the same university. Actually, he is a conference master in the University of Batna – Hadj Lakhdar University.

Main Works: His main interests are Knowledge Engineering, Artificial Intelligence, Vision and Image Processing , Machine learning , Multi-Media Intelligent Information Systems and Natural Language Processing.

Name: Smaine Mazouzi

Affiliation: Associate professor at 20 aout 1955 university of Skikda, Algeria, head of the laboratory of informatics and communication of Skikda University

Address: BP 26, El Hadaik Road, 21000 Skikda, Algeria

Brief Biographical History:

1996 - MS degree in computer science - Robotic vision

2008- Phd degree in computer science - Collaborative vision systems

2012-Informatics and Communication Laboratory Head - Skikda University

Main Works:

A New Distributed Approach for Range Image Segmentation. CIARP 2011: 296-303

Une approche multi-agent pour la segmentation d'images de profondeur à base d'objets polyédriques. Une nouvelle approche de segmentation d'images. Technique et Science Informatiques 28(3): 365-393 (2009)

A Multi-agent Approach for Range Image Segmentation with Bayesian Edge Regularization. ACIVS 2007: 449-460

Sara Benafia obtained her Master's degree 2 in Computer Science from the University of Biskra. She is a group leader at with responsibility for consulting services, managed services where she has worked since 2010.

Main Works: She has a great interest in software development, analysis and design. She already has experience in this area through her professional experiences and her internships during its university studies. The main interests are machine learning, Specifications for web applications, Semantic Web Service and Web engineering techniques.

How to cite this paper: Ali Benafia, Smaine Mazouzi, Benafia Sara, "Handwritten Character Recognition on Focused on the Segmentation of Character Prototypes in Small Strips", International Journal of Intelligent Systems and Applications(IJISA), Vol.9, No.12, pp.29-45, 2017. DOI: 10.5815/ijisa.2017.12.04