Democratic Republic of Algeria

Ministry of higher education and scientific research

Order's N° :
Serie's N° :

University Of Ghardaia

Sciences & Technologies Faculty
Department of sciences and technologies

A Thesis submitted in partial fulfillment of the requirements for the degree of

MASTER

Domain: Sciences & Technologies
Field : Automation
Academic Speciality : *Automation*

By: Benhamadi Abdelmadjid

Topic

ARABIC HANDWRITING RECOGNITION USING LOCAL APPROACH

Date : 06/06/2017

Supervisors:

R.Sadouni	MCB	Ghardaïa Univ	**President**
A.Kiffouche	MAA	Ghardaïa Univ	**Examiner**
B.Laadjal	MAA	Ghardaïa Univ	**Examiner**
M.Arif	MAA	Ghardaïa Univ	**Advisor**

Academic Year 2016/2017.

Acknowledgement

Firstly, I would like to express my sincere gratitude to my advisor **Mr. Arif Mohamed** for the continuous support of my Master study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Master study.

Dedication

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving mom whose words of encouragement and push for tenacity ring in my ears. My brothers & sister have never left my side and are very special.

I also dedicate this dissertation to my many friends who have supported me throughout the process. I will always appreciate all they have done, As well as my beloved for being always there, & helping me for the many hours of proofreading & rearranging texts.

I dedicate this work also to my lovely pet Messouda who has gone forever .

Index

Chapter I

Arabic handwriting recognition using local approach

Chapter II

State of art and pretreatment of an Arabic word recognition

Chapter III

Extraction of paramters & recognition's algorithm (SVM)

Chapter IV

Results & discussion

Figures & Tables

Chapter I

Chapter II

Chapter III

list of tables

Chapter I

Chapter III

General Introduction

One of the goals of computer research is to extend the boundaries of what is automatable. Repetitive, tedious tasks involving large volumes of data are good candidates. These include the processing of bank checks, the sorting of postal mail, the indexing of national archives (military archives, census forms, library funds, etc.), the indexing of private archives, the processing of incoming business mails, and so on. Although a machine is capable of performing complex calculations, in which it often exceeds human capabilities, it is nonetheless limited. We must still make use of a keyboard in order to communicate with it, a task that is painstaking for some, or at least unnatural.

For the time being, and although research in this field has continued for more than thirty years, the general solution to the problem of the automatic reading of cursive writing remains unknown. It seems, however, that cursive handwriting recognition has an important role to play in future recognition systems and hence that this field of research is still very much relevant.

The automation of this task requires the machine to be able to read the handwriting. However, the types of writing styles can vary considerably, depending on the writer. While reading is a relatively trivial act for a human, this activity nevertheless involves complex processes. Properly recognizing isolated symbols is not enough.

Studying how a human successfully performs this complex task could prove useful in teaching machines how to read handwritten texts. Which primitives are detected during reading? How do we access the information that allows us to understand the meaning of a word? Is the perception of a word constructed from the perception of its letters or from the perception of its general form?

For several years, researchers in the fields of biology, neurophysiology, cognitive psychology and linguistics have studied these questions, and reading models have resulted from these investigations. Although these models are still to be improved and although several of the theories put forward different points of view, we believe we can take advantage of their observations to develop a robust system of image recognition of isolated cursive words.

In fact, the primitives detected during reading can help us make an informed choice of primitives to be sought as a priority during the recognition process. The problem of lexical access can also influence the choice of architecture chosen for the method developed. However, most of these reading models have been developed from printed texts.

Few studies have been done on the mechanisms involved in the reading of cursive writing. Their authors conclude that even if the reading of cursive words differs at first sight from the reading of printed words, once what they call the “normalization of the cursive” is completed, printed and manuscript words seem to be subject to similar processing.

Recognition is called “on-line” when dynamic data is acquired during writing. We may think here of a tablet or an electronic paper where the user writes with a pen. On the other hand, it is called “off-line” when it comes to recognizing the image of a word obtained with a scanner.

The objective of this essay is to propose a system of recognition of off-line Arabic handwriting. This system is based on a structural segmentation method and uses Support Vector Machines (SVM) in the classification phase.

The first chapter is a reminder of some general concepts of Optical Character Recognition (OCR). As well as the necessary steps for the realization of a system of recognition, and a study of the OCR and the Arabic language, where we find a reminder on certain aspects of Arabic calligraphy, followed by notions of OCR on Arabic writing.

The second chapter is specific to the state of the art of segmentation of texts in the general case. For this purpose, we describe the process involved in the detection of objects in a page, the segmentation of the text blocks in rows then in word and then in characters. We focus on the methods used in this type of segmentation.

The third chapter will focus on the classification method, by studying the Support Vector Machines (SVM) method.

The fourth chapter constitutes our contribution. It is an algorithm allowing the segmentation of Arabic handwritten texts, followed by the tests and results obtained. We complete the work with a conclusion on the results obtained with our method, and finally the prospects of this work.

Chapter I

Arabic handwriting recognition using local approach

I.1 Introduction

The researchers of the Arabic handwritten characters’ recognition expose a domain that extends quickly and indefinitely, evoked by a place so important in the last two decades. This is how the Arabic characters' recognition is today a concern whose relevance is indisputable by the community of researchers who have devoted their efforts to reduce the constraints and to expand the realm of the recognition of Arab characters.

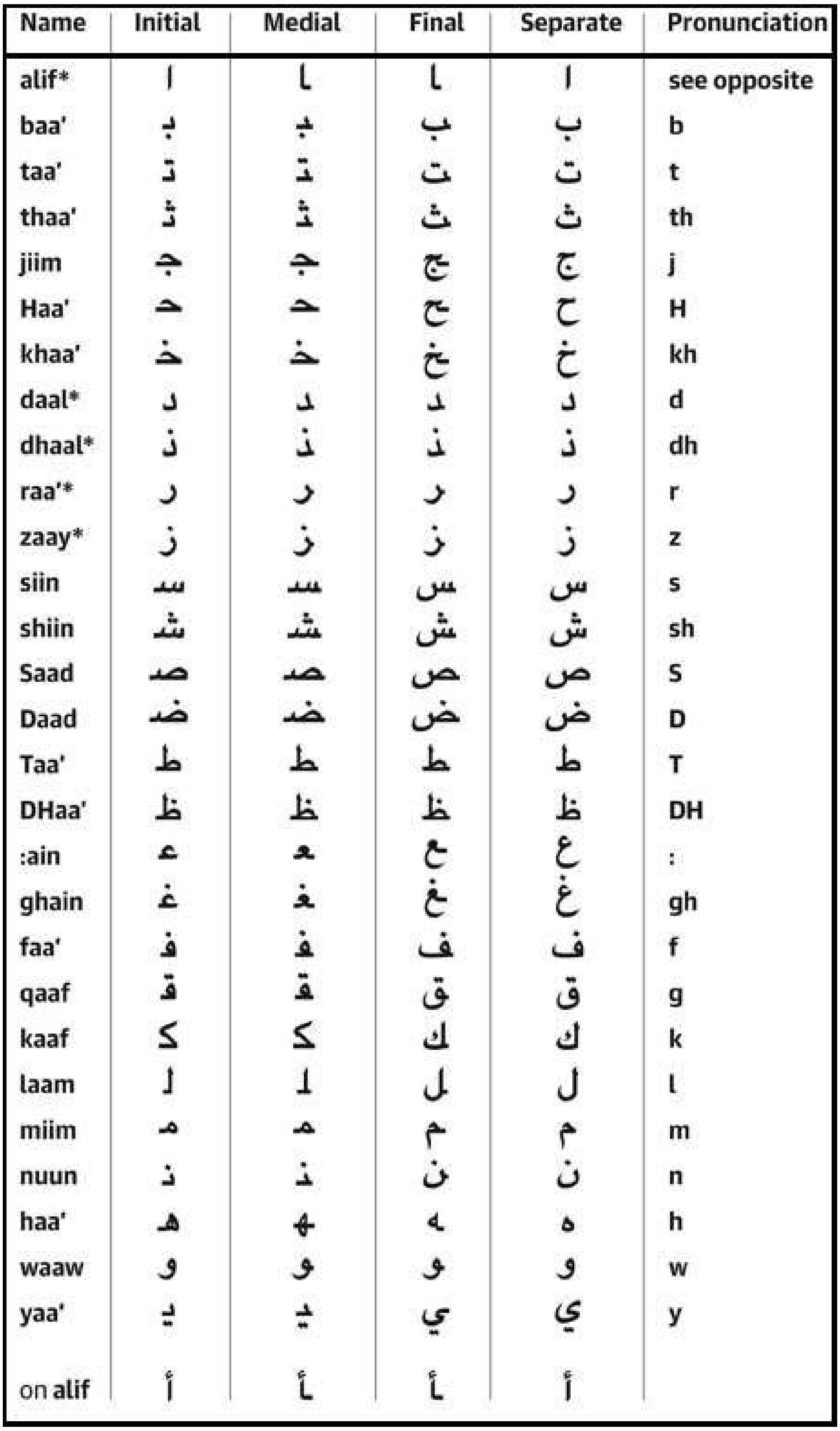
A handwriting recognition system should ideally locate, recognize and interpret any text or number written on a stand of an arbitrarily variable quality such as maps, forms, calendars, old manuscripts, etc.

Among the areas of application, we find the postal sector for the recognition of the code, the mailing address and the automatic reading of bank checks; the administrative area for the electronic management of document workflow; digital libraries for document indexing and finding information; biometrics for identification of the writer ... etc..

I.2 Features of the Arabic script

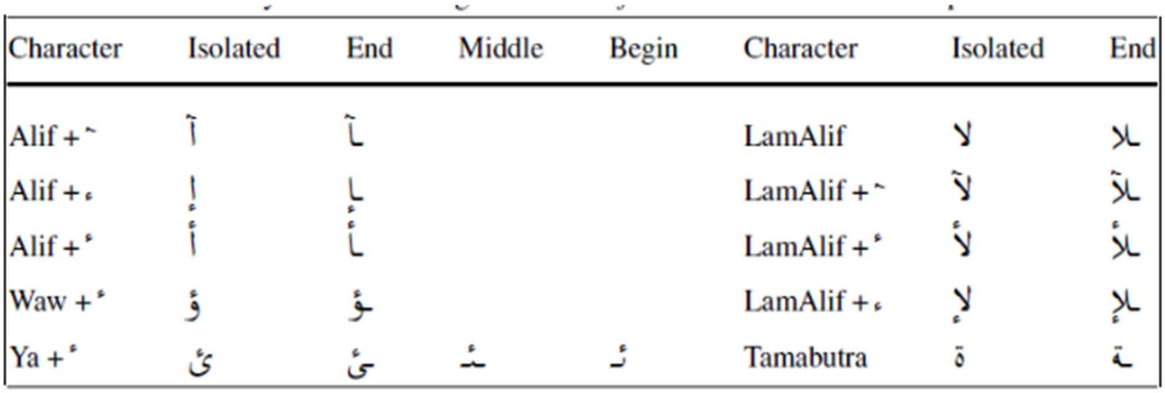
Arabic is a consonantal script that uses an alphabet of 28 letters (Table I.1), to which we should add the Hamza «ء» which is often considered as an additional sign [4]. The Hamza « ء» has a special spell that depends on grammatical rules, which multiplies the necessary forms to its representation, since it can be written alone or with the three vowels' support (alif, waw and ya) of which it follows the code (Table I.1).

Moreover, the Arabic alphabet includes other additional characters such as «ة» and «لا». Thus, several authors consider that the Arabic alphabet contains rather 31 letters instead of 29. The consideration of the symbol «~»which is written only to the support of the character «ا», suggests additional graphics (Table I.2) [5].



Name	Initial	Medial	Final	Separate	Pronunciation
alif*	ا	ـا	ـا	ا	see opposite
baa'	بـ	ـبـ	ـب	ب	b
taa'	تـ	ـتـ	ـت	ت	t
thaa'	ثـ	ـثـ	ـث	ث	th
jiim	جـ	ـجـ	ـج	ج	j
Haa'	حـ	ـحـ	ـح	ح	H
khaa'	خـ	ـخـ	ـخ	خ	kh
daal*	د	ـد	ـد	د	d
dhaal*	ذ	ـذ	ـذ	ذ	dh
raa'*	ر	ـر	ـر	ر	r
zaay*	ز	ـز	ـز	ز	z
siin	سـ	ـسـ	ـس	س	s
shiin	شـ	ـشـ	ـش	ش	sh
Saad	صـ	ـصـ	ـص	ص	S
Daad	ضـ	ـضـ	ـض	ض	D
Taa'	طـ	ـطـ	ـط	ط	T
DHaa'	ظـ	ـظـ	ـظ	ظ	DH
:ain	عـ	ـعـ	ـع	ع	:
ghain	غـ	ـغـ	ـغ	غ	gh
faa'	فـ	ـفـ	ـف	ف	f
qaaf	قـ	ـقـ	ـق	ق	g
kaaf	كـ	ـكـ	ـك	ك	k
laam	لـ	ـلـ	ـل	ل	l
miim	مـ	ـمـ	ـم	م	m
nuun	نـ	ـنـ	ـن	ن	n
haa'	هـ	ـهـ	ـه	ه	h
waaw	و	ـو	ـو	و	w
yaa'	يـ	ـيـ	ـي	ي	y
on alif	أ	ـأ	ـأ	أ	

Table I.1Arabic alphabet in its different forms



Character	Isolated	End	Middle	Begin	Character	Isolated	End
Alif + ~	آ	ـآ			LamAlif	لا	ـلا
Alif + ء	إ	ـإ			LamAlif + ~	لآ	ـلآ
Alif + ء	أ	ـأ			LamAlif + ء	لأ	ـلأ
Waw + ء	ؤ	ـؤ			LamAlif + ء	لإ	ـلإ
Ya + ء	ئ	ـئ	ـئـ	ئـ	Tamabutra	ة	ـة

Table I.2 Selection of special Arabic characters

Like noise reduction, segmentation, and binarization (conversion of the input image into a bi-level image) are performed. As the system should deal with a large number of different unknown writers, the next two blocks normalize the writing style with the goal to make it more robust against size, slant, skew, and line width variations of a word. The next block extracts a number of features from the normalized word image. An important precondition for this step is the estimation of baselines of each word to make the feature extraction more effective. Based on models and their parameters, which were fixed during an offline training process, and a word lexicon,the recognition process is performed by searching for models that fit best with a given feature vector sequence. The output of the recognizer is one or more word hypotheses.

A characteristic feature of the Arabic script is the presence of a horizontal baseline yet known line of reference or of writing. This is the case of the characters in the same chain (Figure I.1)

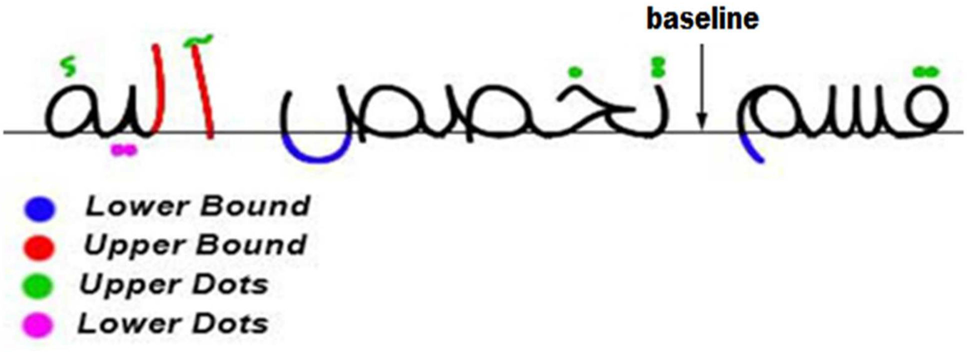


Figure I.1 Example of Arabic script showing the baseline

Certain characters can’t be matched to their left, so they can be only in an isolated or final position; which give, when they exist, compound words of one or more parties generally called PAW (Peace of Arabic Word) or pseudo-word. [45] A PAW corresponds to a chain of one or more characters (Table I.3).

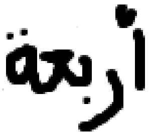
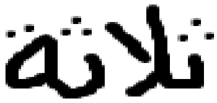
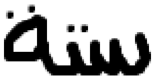
3PAW/word	2PAW/word	1PAW/word
		

Table I.3 Example compound words from the right to the left of 1, 2, 3 PAWs

Because of text justification and/or aesthetics, the horizontal ligatures can be extended by inserting between characters of the same chain one or more "madda" (or tatwil) elongations, corresponding to the symbol «ـ».

The elongation is always situated to the left of delivered character. If the elongation line is associated with a character at the beginning or final position, the character takes its shape from the middle and sees its escape increasing the number of inserted « madda » (Table I.3) [5].

At the level of PAW, the insertion of elongation lines affects only its width, the morphology remains the same [6]. Text editors such as Microsoft Word, insert into the text lines, the appropriate number of "Madda" for the left-right justification of an Arabic text.

With 6 maddas	With 3 maddas	With 1 madda	Without madda
قــــــ	قـــ	قـ	ق
تــــــ	تـــ	تـ	ت

Table I.4 Example of characters with and without madda

Arabic writing is semi-cursive in its printed form as well as handwritten.

The characters of the same chain (or pseudo-words) are horizontally ligated and sometimes vertically (in some fonts two, three and even four characters can be ligated vertically), making the characters's segmentation difficult.

قـ {جـ ، حـ ، خـ}	فـ {جـ ، حـ ، خـ}	لـ{جـ ، حـ ، خـ}	مـ {جـ ، حـ ، خـ}

Table I.5 Characters that may be vertically ligated

Mandatory ligatures of the letters: { خ م س ة}

Aesthetic ligature among the first 2 letters: خـــمسة

Aesthetic ligature among the first 3 letters: خمسة

Moreover, the shape of a character is different depending on its position in the pseudo-words and even in some cases; according to the phonetic context. In addition, more than half of Arab characters include in their shape diacritical marks.

These marks can be located above or below the character, but never up and down simultaneously.

Several characters can have the same body but a number and / or position of various diacritical marks.

The Arabic characters may be vowelized. The vowels also known as diacritical in certain documents and short vowels in others, such as [7], can be placed above or below the character. The vowels are of a later invention to the consonants.

In the ordinary and contemporary Arab, we write only the consonants and the long vowels.

The same word with different short vowels can be understood as verb, noun or adjective.

For example « علم» could mean: flag «عَلَمُ» or known: «عِلْمُ» or even Teach: «عَلَمَ» according to its vowelization.

There are 8 signs of vowelization that can be placed above the writing line, such as Fat'ha (ـَ) Dammah (ـُ), Soukoun (ـْ) and Chaddah (ـّ) which must be accompanied by one of the vowelizations: Fat'ha, Dammah or Kasrah, and those that can be placed below the line of writing like Kasrah (ــِـ). Also three "Tanween" can be formed from a double Fat'ha (ــً), from a double

Dammah (ٌ) or from a double Kasrah (ٍ).

Arabic writing contains a lot of fonts and models of writing, so it is sometimes difficult to separate one word from another, particularly when people write with calligraphy.

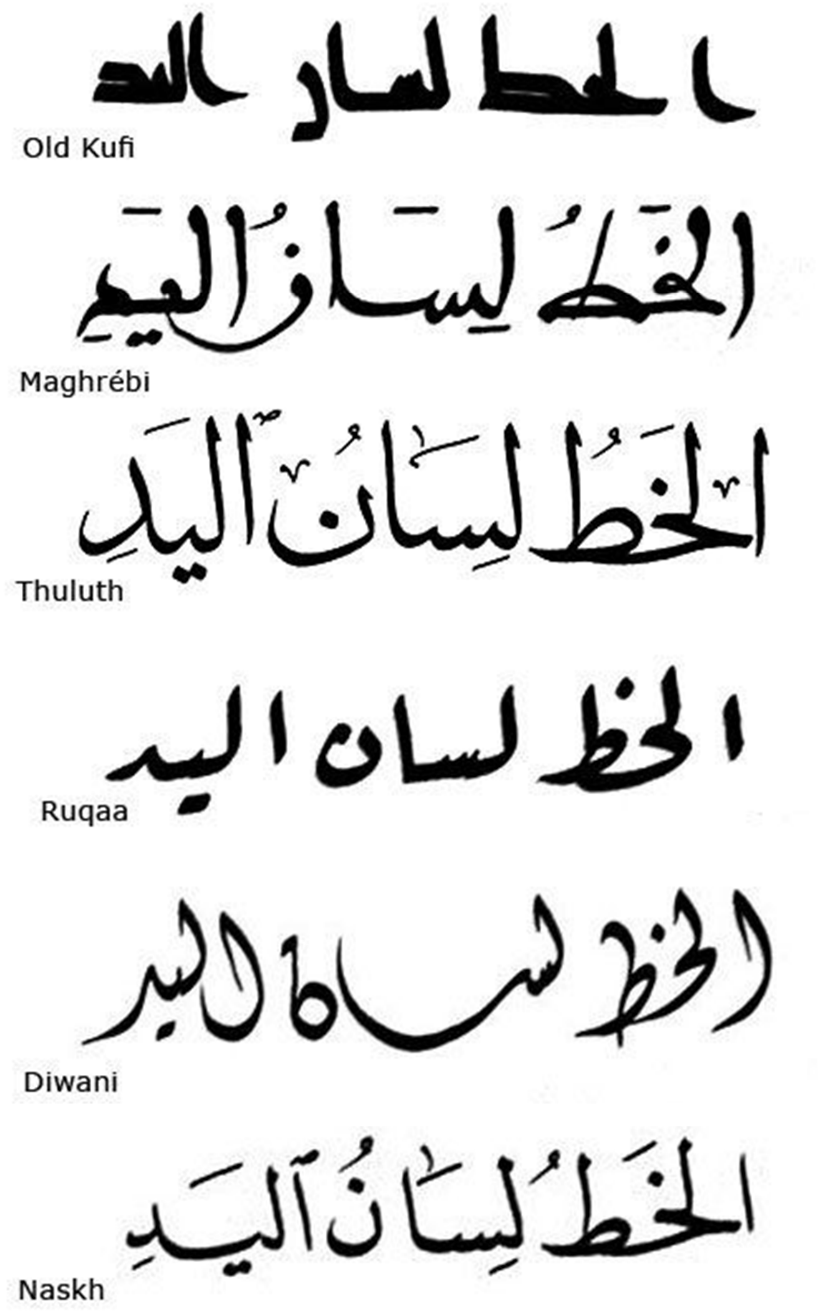


Figure I.2 Different Arabic sentences in different models [8]

Furthermore, the cursivity of the Arabic script shows a complexity of the characters' morphology, the elongations of the horizontal ligatures as well as vertical combinations of certain characters constitute the major problems related to the treatment of this script especially for the pseudo-words.

Indeed, these problems result in a considerable inertia on various levels including:

- The choice of relevant primitives describing the variability in the characters' morphology, knowing that certain topological features are sensitive to the degradation, particularly the diacritical marks and the curls.
- The segmentation method in characters or even pseudo-words (which can be superimposed especially in the case of the manuscript).

All these issues and many more are accentuated in the manuscript's case where other factors intervene (conditions of writing, fusion diacritical marks, overlapping pseudo-words, graphics unequally proportioned ...).

I.3 Different aspects of OCR (Optical Character Recognition)

There is no universal system for OCR that is sufficient to identify any character in any font. It depends on the type of data processed and obviously on the intended application [9]. There are several classifying modes for the OCR systems among which we can cite:

"Online" or "offline" qualified systems following the acquisition's method. The global and analytical approaches depending on whether the analysis is performed over the whole word, or by the characters' segmentation.

The statistical, structural or stochastic approaches relating to features extracted from considered forms.

The figure I.3 shows the various dichotomies of writing's type

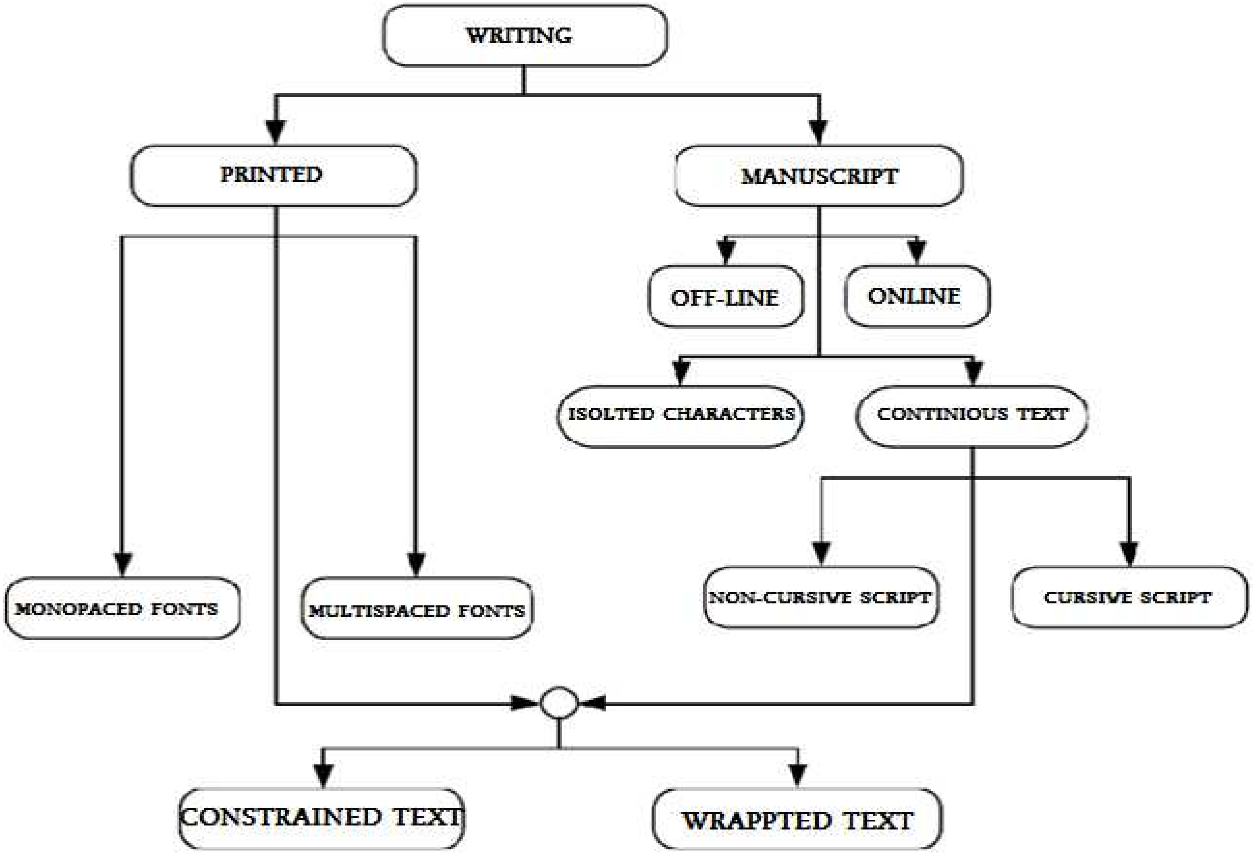


Figure I.3 Dichotomy of writing's types [3]

I.3.1 Recognition (On-Line and Off-Line)

These are two different modes of OCR, each with their own acquisition tools and their corresponding algorithms for recognition.

I.3.1.1 On-line Recognition

This method of recognition takes place in real time (during writing). The symbols are recognized progressively as they are written by hand.

This mode is generally reserved for handwriting; it is a "signal" approach in which the recognition is made on data of one dimension. The writing is represented as a group of points whose coordinates are in function of the time [10].

I.3.1.2 Off-line Recognition

Starting after the acquisition, it is appropriate for printed materials and for the already drafted manuscripts. This mode can be considered as the most general case of the writing's recognition. It approaches the mode of the visual recognition. The information's interpretation is independent from the generating source [11].

The offline recognition can be categorized in several types:

Recognition of text or document analysis: In the first case it is the recognition of a text from limited structure to a few lines or words. The research consists on a simple identification of words in the lines, then on a cutting of each word into characters [9].

In the second case (document analysis), it is a well-structured data whose reading requires the knowledge in typography and in the document's page layout. The approach is no longer a simple pretreatment, but an expert approach of document analysis: there are localization of areas, separation of graphics and photographic areas, semantic tagging of text areas from models, determination of the reading order and of the document's structure.

Recognition of the printed or of the manuscript: Approaches differ depending on whether the recognition of printed or handwritten characters. The printed characters are in the general case horizontally aligned and vertically separated; this is the reading phase [9]. The characters' shape is defined by a calligraphic style (font) which constitutes a model for the identification.

In the case of the manuscript, the characters are often ligated and their graphics is unequally proportioned, derived from the variability within and between writers. This usually involves the techniques' use of specific delimitation and frequently of the contextual knowledge to guide the reading [12].

In the case of the printed word, the recognition can be mono-font, multi-font or omni-font:

A system is mono-font if it can recognize only a single font at once, i.e. it knows graphics as a single font. This is the simplest case of printed characters' recognition [13].

A system is multi-font if it is able to recognize different types of fonts among a set of previously learned fonts [9].

An omni-font system is able to recognize any font, generally without any prior learning. However, this is practically impossible because there are thousands of fonts, some of which unreadable by humans (except from its designer) and with a font building software anyone can design a font in his own way.

In the case of the manuscript, the recognition can be mono-writer, multi-writer or omni-writer. The offline handwriting can be classified into two categories of writings: cursive and semi-cursive.

A system is Mono-writer (own to the writer): it is the fact that the system can recognize only one writing. All these elements influence the shape of the letters (writing bent, curly, rounded, straight, etc.) and of course on the shape of ligatures, sometimes compromising the identification of boundaries between letters.

A system is said Multi-writer (own handwriting): it is because the system can identify and recognize the writing for a certain number of writers. A system is said Omni-writer (specific to any handwriting): it is the fact of reducing the information contained in the image to the necessary minimum to accurately model the structure of characters [14].

I.4 Approaches for recognition

Two approaches are opposed in the word recognition: global and analytical.

I.4.1 Global approach

The global approach is based on a unique description of the word's image, seen as an indivisible entity. Having a lot of information, in fact, the discrimination of similar words is very difficult, and learning about styles requires a great amount of samples which is often difficult to bring.

I.4.2 Analytical approach

The analytical approach based on a division (segmentation) of the word. The difficulty of this approach was clearly evoked by Sayre in 1973 and can be summarized by the following dilemma:

"to recognize letters, we must segment the line and to segment the line, we must recognize the letters". It is followed that a recognition process according to this approach must necessarily be conceived as a relaxation process alternating the phases of segmentation and of segments identification. This approach is the only applicable one in the case of large vocabularies [15].

I.5 Conclusion

In this chapter, we have seen some basic concepts of OCR, the main recognition methods in general, in addition to the main problems encountered in this area. Then we discussed the different steps involved in designing a character recognition system. Subsequently, we have seen the main morphological and typographical properties of Arabic writing. The lack of standardization of typography has shown the complexity of adapting Arabic writing to modern technological requirements. In addition, we have seen the major problems in this field, which are reduced to the cursivity of writing and the sensitivity of certain topological characteristics of Arabic to degradation, in this case diacritical points and loops. Thus, Arabic optical character recognition remains an unresolved task.

CHAPTER II

State of art and pre-treatment of an Arabic word

Introduction

The words recognition of a written text consists in recognizing the different characters making up these words, which makes the segmentation phase a very important and crucial phase in the recognition process. It allows having a sequence of images representing the different characters from a source image containing the word to be recognized.

In this chapter, we will present & explain the stages of pretreatment as well as the segmentation methods, for different writings (i.e. cursive or non-cursive).

II.1 Process of recognition

A recognition system generally involves the following steps: acquisition, preprocessing, segmentation, feature extraction, classification, eventually attended by a post-processing phase.

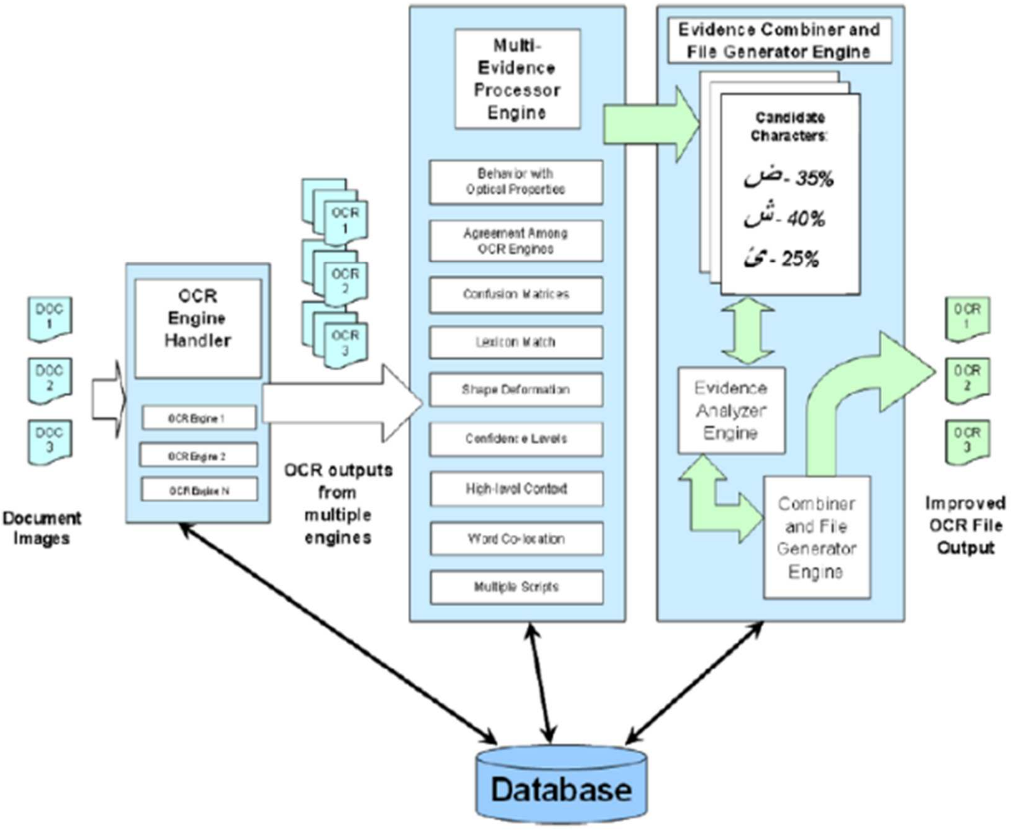


Figure II.1 General outline of the characters' recognition system.

II.2 Acquisition

This phase involves capturing an image of a text using physical sensors (scanner, camera ...) and to convert it into digital quantities adapted to the treatment system, with a minimum of potential degradation.

II.2.1 Acquisition phases

It consists of two phases:

- **Sampling (digitization)** of an image is spatial, by cutting in pixels.
- **Quantization (encoding)**: is a numerical value given to the light intensity; it is a gray level, called the dynamics of the image.

This dynamic is given as follows:2^m,where m is the number of bits, for example: the gray level 256 is encoded on 8 bits, the color image is coded on 24 bits (1 byte for each color (R,V, B))[16].

II.2.2 Concept of neighborhood

There are distinguished two types of neighborhood in the field of image processing:

Neighborhood in4: set of pixels 'p' which have one side in common with the considered pixel.

Neighborhood in8: set of pixels 'p' which have at least one connection point with the considered pixel.

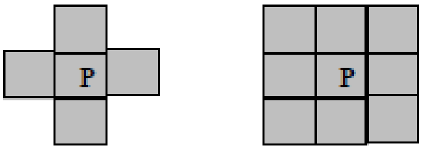


Figure II.2 Neighborhoods in 4 and Neighborhood in 8

II.3 Pretreatment phase

The pretreatment in preparing the data derived from the sensor to the next phase. This is basically to reduce the superimposed noise to the data and try to keep only the significant information of the shape shown. The noise may be due to the acquisition conditions (lighting, incorrect document formatting ...) or even more to the quality of the original document.

Among the pretreatment operations generally used, we may include: the binarization, the dilation, the erosion, the skeletonization and the normalization (Figure II.3).

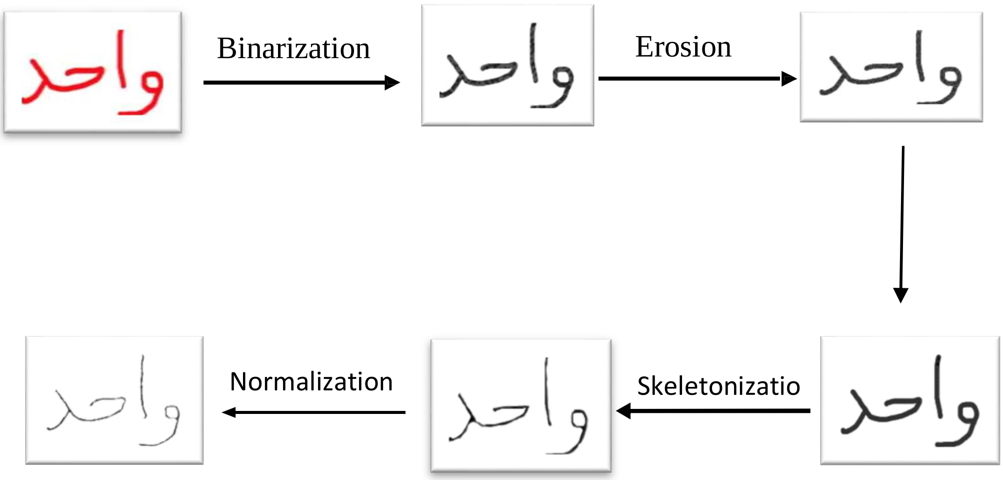


Figure II.3 Effects of certain pre-processing operations.

II.3.1 Binarization

Binarization is the passage of a color into image, defined by several gray levels in bitonal image (composed of two values 0 and 1) allowing a classification between the bottom (image of white paper) and the shape (lines of carvings and black characters).

Several techniques have been developed to convert an image to gray scale or color levels, in a binary image. All these techniques are based on the principle of thresholding as shown in the following equation:

$$I_b(x,y) = \begin{cases} 0 & if\ \ I_n(x,y) < T \\ 1 & if\ I_n(x,y) \geq T \end{cases} \qquad (2.1)$$

I_n(x,y) describes the intensity to the "n" gray scale at each point of the image, If (x, y) represents the intensity at two levels and T is the binarization threshold. If In (x, y) is greater than the threshold values, we attribute the pixel corresponding to the value of maximum intensity (white). Otherwise, the point is considered black and we attribute it the value of minimal intensity [17].

For images of gray scale, we can find in [18] a list of binarization's methods, offering adaptive thresholds (egg. adapting to the different gray scale’s distribution). The authors in [19] propose a solution for the images of postal addresses.

The threshold's search involves several stages: preliminary binarization based on a distribution of multimodal mixture, texture's analysis by means of histograms of lines' length, and threshold's selection from a decision tree.

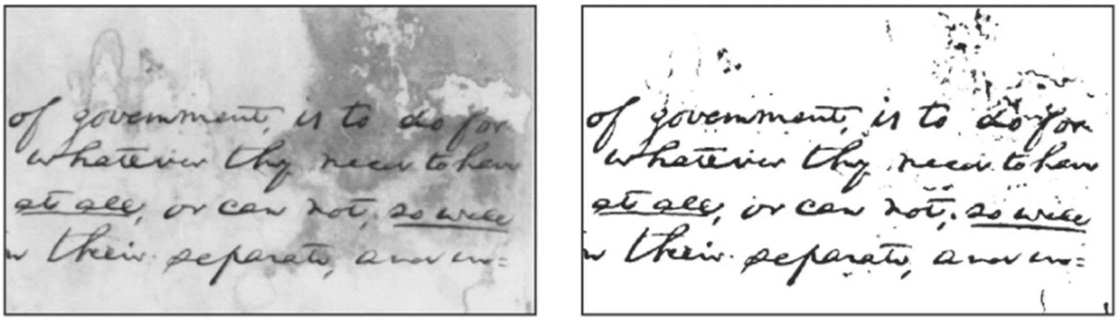


Figure II.4 Example of adaptive binarization [20]

II.3.2 Transformation by erosion

If a pixel 'p' is black (p (x, y) = 0), and there are at least 3 pixels or 7 pixels within 4-neighbors, or 8-neighbors respectively, which are white ($P_{neighbor}$ (x,y) =1), then we assign to this pixel the white color (p (x, y) = 1), i.e. we erase this pixel (This is a linking of one or two pixels of a related form) [17].

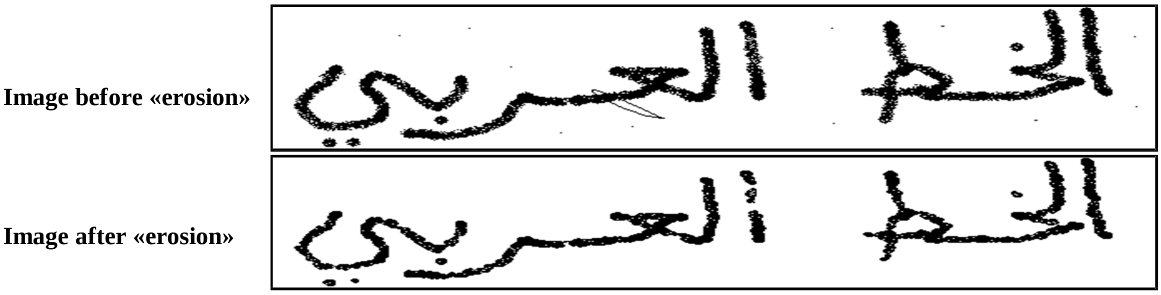


Figure II.5 Image processed by erosion

II.3.3 Transformation by dilatation

It is the opposite of the transformation by erosion, if a pixel 'p' is white (p (x, y) = 1), and there are at least 3 pixels or 7 pixels in the 4-neighbors, or 8-neighbors respectively, which are black (Pneighbor (x,y) =0), then we assign to this pixel the black color (p(x,y)=0).

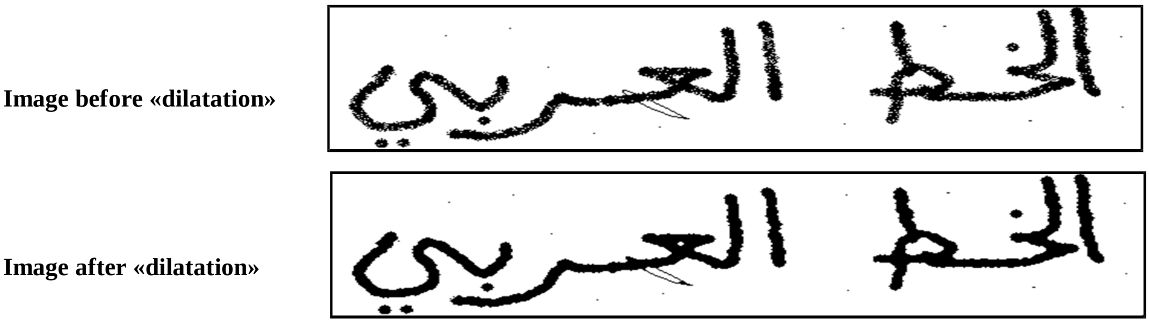


Image before «dilatation»

Image after «dilatation»

Figure II.6 Image processed by dilatation

II.3.4 Morphological opening

It is a combination of operations: erosion followed by a dilatation of an image with the same structuring element. For softening the contours, simplify the shapes by linking the bumps while maintaining the overall appearance [17].

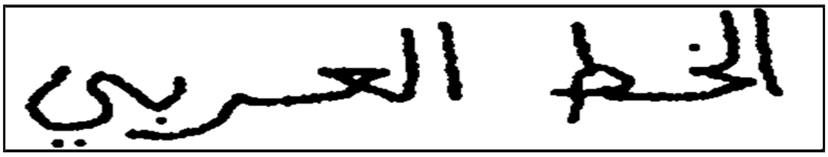


Figure II.7 Image processed by Opening

II.3.5 Morphological closing

This is the opposite of the opening, it consists on an operations' combination: dilation followed by an image's erosion of the same structuring element. It helps to simplify the forms, filling the spaces.

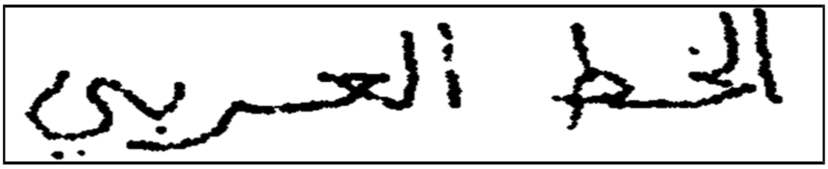


Figure II.8 Image processed by Closing

II.3.6 Skeletonization of an image

Widely used in the area of 'shape recognition', this operation consists of transforming a binary image into a 'skeleton'. The skeleton is a set of infinitely small line widths. Skeletonization must preserve the image's connectivity. In other words, this should neither separate the related items, nor link the unconnected elements. The aim is to simplify the character's image to an easier "line" one to handle by reducing it to the character's outline [21].

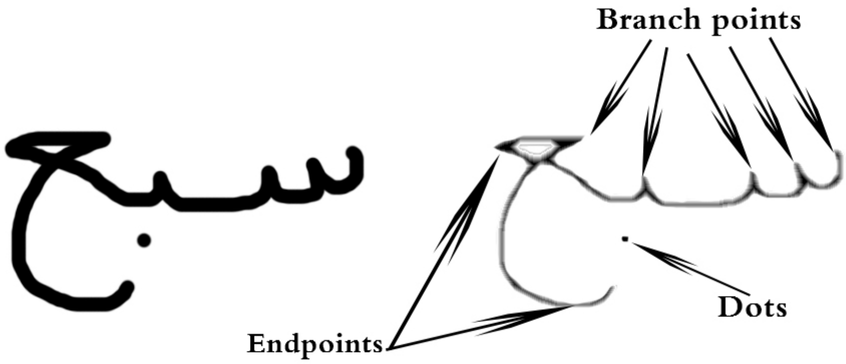


Figure II.9 Skeletonization of an Image [1]

There are several skeletonization's techniques among this we mention a tracking technique called semi-skeletonization. This technique is based on the detection for each column of pixels: the points of beginning, middle and end of the writing's layout (figure II.9).

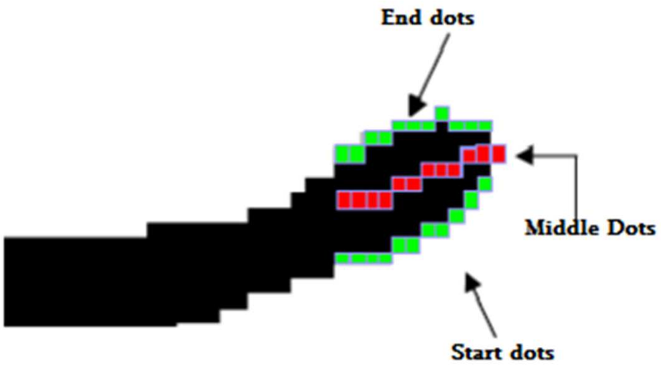


Figure II.10 Semi-skeletonization technique [2].

During the scanning, the milieu's extracted points are used to calculate the length and the angle of segment's deviation during treatment, whereas the endpoints are used to retrieve information about the segment's shape **[2]**.

II.3.7 Normalization

The discernment's study requires the elimination of conditions which can distort the results such as the size difference, so, it is necessary to achieve the normalization of the character size. After this operation, the images of all the characters are defined in a matrix of the same size, in order to facilitate the subsequent treatments.

This action usually introduces slight deformations in the images. However, certain characteristic features such as the shaft in characters (ط ظ لا for example) can be eliminated following the normalization, which may lead to confusions between certain characters[22].

An application should be defined to perform this task. This application must be reversible in order to move from one size to another and to return afterwards.

G:R^2 $\longrightarrow$ all colors

(x,y) $\longrightarrow$ color(black/white)

G is defined in the following way:

$$\begin{cases} G(x,y) = Black \quad if\ pixel\ (\mu 1, \mu 2\ Y) = Black \\ G(x,y) = White \quad Otherwise \end{cases} \qquad (2.2)$$

Such as: µ1 = (frame width of the character) / (frame width of the normalization)

µ2 = (frame height of the character) / (frame height of the normalization)

I.4 Segmentation phase

The segmentation is an operation applied to the image which consists of subdividing a real scene, into constituent parts or objects, projecting a real scene on a plan. It is the first operation to be performed in "shapes recognition". So, it must have a certain number of attributes, representative of areas that we are searching them to extract, to proceed to the individual classification of the points [6].

II.4.1 Techniques of the segmentation

There are two techniques to implement the segmentation. The first, known as implicit segmentation, and the second is the explicit segmentation.

A-Implicit segmentation

The implicit segmentation methods are based on the approaches used in the field of speech, where the signal is divided into regular time intervals, and proceed with a significant over-segmentation of the passive word's image (or a few pixels). This ensures a significant presence's level of the connection points between considered letters.

Segmentation is performed during the recognition which ensures its guide. The system searches in the image for components or groups of grapheme corresponding to its letters of classes. [23] Classically, it could be done in two ways:

- Either by windowing: the principle is to use a moving window of variable width (which is not easy to determine) in order to find sequences of potential segmentation's points that will be confirmed or not by characters' recognition. It requires two steps: generating segmentation's hypotheses (points' sequences obtained by the windowing); the second is the choice of the recognition's best hypothesis (validation).
- Either by research of primitives: it is to detect the primitives' combinations that will give the best recognition.

b- Explicit segmentation

This approach, often called dissection, is prior to recognition and is not called into question during the recognition phase. The characters' assumptions are determined from the low level of information contained in the image. These assumptions shall be final and must be highly reliable because the segmentation's slightest error challenges the eventual processing's totality.

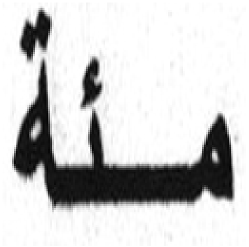
The explicit segmentation's approaches are based on a morphological analysis of the handwritten word to locate points of potential segmentation. They are particularly suitable to the analysis of the two-dimensional representation and thus often used in the systems of words' offline recognition.

Certain explicit segmentation's methods are based on a mathematical morphological analysis, exploiting the concepts of regularity and singularity of the line, analysis of higher/lower contours of the word. The potential segmentation's detected points are confirmed using various heuristics [23].

II.4.2 Segmentation's stages

a- Segmentation of the text into lines

Arabic processing's methods often use the horizontal projection to retrieve the lines. However, the presence of diacritical marks complicates this extraction and sometimes leads to the confusion of lines. [24] This problem occurs when the spacing is calculated by a simple average of the different spacing. To correct this problem, some authors such as [25] first identify the different lines of writing, then group the text blocks according to their proximity in relation to the already localized writing lines.



Tested Word from database to extract the Features

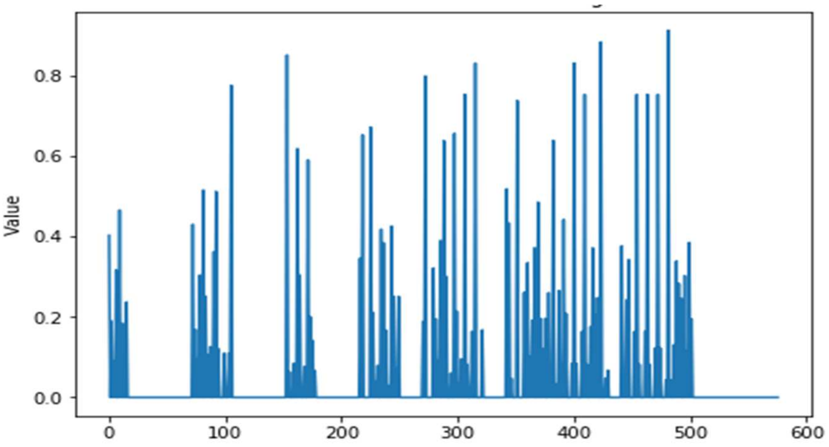


Figure II.11 Example of a horizontal histogram of a line of text [1].

As for Latin, a fusion of the lines is also possible because of the poles and the legs.

In case of fusion, an empirical correction method consists to first locate the line that contains the maximum black pixels [23].

The parts above and below this line are then analyzed on the basis of the black pixel densities of different lines.

For example, if the fusion took place in the upper part, the line having the minimum pixel density in this portion corresponds to the boundary between the fused lines.

b- Segmentation in Pseudo word (PAW: Peace of Arabic Word)

It is performed by determining the histogram of the vertical projections of the different lines of text. However, this method is not effective in the case where the PAWs are vertically overlapped. In this case, other techniques are used such as the determination of the contour, of the skeleton, or even of related components. The technique's choice is frequently guided by the analysis method [5].

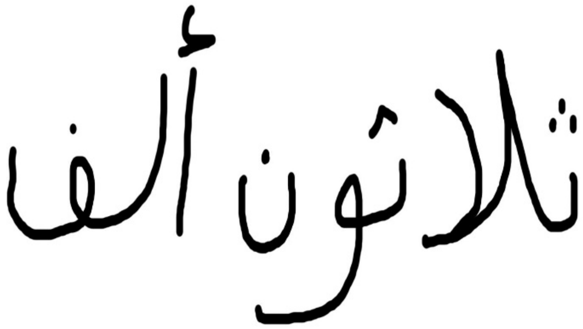


Figure II.12 Example of overlapping PAWs respectively from right to left between «ثلاثون» and «ألف».

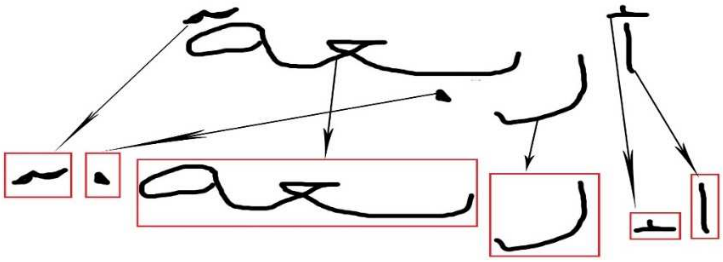


Figure II.12 Example of segmentationfrom our database [2]

c- Segmentation of the line of text into words

In the Arabic OCR, segmentation is often reserved for the extraction of PAWs, the word is instead considered in the post-treatment phase (if provided) to validate the results or correct recognition's errors. In addition, in order to prevent the problem of wrong word segmentation, some authors introduce into their systems a single word at the same time [23].

d-Word segmentation's into characters

The segmentation into characters (or into grapheme) constitutes the most difficult problem of the Arabic script's recognition. The difficulties on this level are of the same type as those faced during the recognition of the Latin manuscript (cursive), but often more complex because of the forms' diversity of the Arab character, of the shortest link that exists between the successive characters, the horizontal ligatures' elongation and the vertical ligatures' presence [5].

II.5 Extraction phase of characteristics

This is one of the most delicate and important stages in OCR. The characteristics' types can be classified into four main groups: structural characteristics, statistical characteristics, global transformations, and superposition of models and correlation [29].

II.5.1 Structural characteristics

The structural characteristics describe a shape in terms of topology and geometry by providing its local and global properties. Among these characteristics, we may mention:

- The lines and the handles in the different directions as well as their sizes.
- The endpoints.
- The points of intersection.
- The curls.
- The number of diacritical marks and their position in relation to the baseline.
- The vowelizations and zigzags (Hamza).
- The character's height and width.
- The shape's category (primary part or diacritical marks, etc.).

Many other features can be derived, depending on whether they are retrieved from a curve, a line or a contour segment.

II.5.2 Statistical characteristics

The statistical characteristics describe a shape in terms of a set of measurements taken from this shape. The characteristics used for the recognition of Arabic texts are: the zoning (zonning), the characteristics of locus (loci) and the moments [29].

The zoning consists of superimposing a grid (n × m) on character's image and calculate for each of the resulting areas, the points' average or percentage in gray scale, thus giving a characteristics' vector of size (n × m).

The Loci method is based on the number's calculation of the white and black segments along a vertical line through the shape, as well as their lengths [10].

II.5.3 Global transformations

Obviously, they are based on a global transformation of the image. The transformation consists in converting the pixel representation in a more abstract representation for reducing the characters' size, while keeping the maximum information on the shape to recognize. For example: the Hough transform, the Fourier transform, and the Zernike moments [23].

Zernike moments

The Zernike polynomials were first proposed in 1934. The formulation of these moments seems to be one of the most popular, surpassing the alternative solutions in terms of noise resilience, redundancy of the information and reconstruction's opportunities. Several studies also show the superiority of the Zernike moments in comparison to other approaches [30].

The Zernike moments are constructed using a set of complex polynomials which form a complete orthogonal set defined on the disk unit with: $(x^2+y^2)\leq \mathbf{1.}$

$$A_{mn} = \frac{m+1}{\pi} \sum_x \sum_y I\,(x,y)\,[V_{mn}(x,y)]\,dx\,dy \qquad (2.3)$$

Where m and n define the order of the moment and I (x, y) the grey scale of an image's pixel I on which we calculate the moment. The Zernike polynomials V_{mn} (x, y) are expressed in polar coordinates:

$$V_{mn}(r,\Theta)=R_{mn}(r)e^{-jn\Theta} \qquad (2.4)$$

$$R_{mn}(r) = \sum_{s=0}^{m-|n|}(-1)^s \frac{(m-s)!}{s!\left(\frac{m+|n|}{2}-s\right)!\left(\frac{m-|n|}{2}-s\right)!} r^{m-2s} \qquad (2.5)$$

Where R_{mn}(r)is the orthogonal radial polynomial:

The moments A_{mn} are invariant under rotation, translation and scale change (after normalization of the shape's size).

This representation is invertible; the image can be reconstructed in the following way:

$$I(x,y) = \lim_{n\to\infty} \sum_{n=0}^{n} \sum_m A_{mn}V_{mn}(x,y) \qquad (2.6)$$

The moments' order has a great influence on the conservation of angular information. The higher the order is, the higher the described angular variations are thin. The figure 1.17 illustrates this point.

(a) original image, (b) reconstruction of order 10, (c) reconstruction of order 20 (d) reconstruction of order 40 [32].

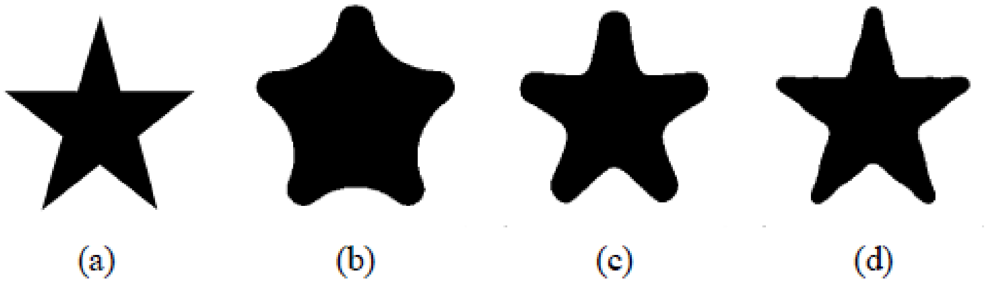


Figure II.13 Example of reconstruction from the Zernike descriptors.

II.5.4 Superposition of models (template matching) and correlation

The method of 'template matching' applied to a binary image (in grayscaleor skeletons), involves using the image of the shape as characteristics vector to be compared with a model (template) pixel by pixel in the recognition phase, and a similarity measure is calculated [29].

II.6 Classification's phase

The classification in an OCR system consists of two tasks: learning and recognition (decision). At this stage, the characteristics of the previous step are used to identify a character and to assign it to a reference model [4].

II.6.1 Learning

During this step, it's about to learn to the system the relevant properties of the used vocabulary and to organize it in reference models. The ideal would be to teach to the system as many samples as shapes of different scripts, but this is impossible because of the great variability of the writing that would lead to a combinatorial explosion of representation's models. So the trend is to replace the number with a better quality of the characteristic features [5]

The learning consists of two different concepts; training and adaptation. The training is to teach to the system the characters' description whereas the adaptation is used to improve the system performance by taking advantage of previous experiences. Some systems allow the user to identify a character when they fail to recognize it and they utilize the user's input each time the character is encountered [34].

The learning process differs depending on whether the recognition of printed characters or manuscripts or recognizing mono-font texts or multi-font ones. In a general way, there are two types of learning techniques: supervised and unsupervised.

The learning is called supervised if it is guided by a supervisor known as teacher. It is performed during recognition’s preliminary step by introducing a large number of reference samples. The professor indicates in this case the name of each sample.

The choice of reference characters is handmade based on the application.

The number of samples can vary from a few units to several dozen.

Unsupervised learning or without a teacher is to provide the system an automatic mechanism which is based on precise rules of grouping to find the reference classes with a minimal assistance. In this case, the samples are introduced by the user into a large number without indicating their class [36].

II.6.2 Recognition and decision

The decision is the final step of recognition. From the descriptionin parameters of the treated character, the recognition module searches among the reference models in the presence, those which are the nearest to it.

Hereinafter, we present a brief literature review of some discriminative models among the most famous for handwritten characters' recognition.

II.6.3 Bayesian decision

In a modeling approach, we seek a production model $p(x|c_{i)}$,that gives for each class of these shapes, the distribution of the data associated with it. In an approach by discrimination, we search to approximate the distribution $p(x|c_{i)}$,which represents the posterior probability of the data given the ci class. In practice, this information is not always provided [3].

Bayesian decision rule is a key theory in classification which estimates the posterior probability from the conditional probability and issue a belonging vote of the processed form. It is writte:

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{P(x)} \qquad (2.7)$$

For k classes, the Bayesian decision seeks the ci class that maximizes the posterior probability.

It is written:

$$\text{c(x)=}\mathit{arg}\text{max P(c}_{\text{i}}\text{|x)} \qquad (2.8)$$

The term P(x) is the prior probability of the ci class. It is particularly useful if the classes are not balanced in the sample of considered data. The observation's likelihood P(x) is a constant amount which can be omitted from the decision process.

II.6.4 The closest neighbor's method (KPPV)

The KPPV algorithm assigns an unknown shape to the class of its nearest neighbor by comparing the stored shapes in a reference class named prototypes. It returns the closest K shapes of the shape to be recognized according to a similarity criterion. A decision's strategy allows to assign the values of trust to each class in competition and to award the most similar class (depending on the chosen metric) to the unknown shape [5].

This method presents the advantage of being easy to implement and provides good results. Its main disadvantage is related to the classification's low speed due to the large number of distances to calculate.

II.6.5 Neural Networks

The neural networks have expanded significantly thanks to the propagation algorithm of the gradient for the error attributed to Werbos [37] Rumel hart [38] and Le Cun [39]. This classifier has found application in many areas such as characters' recognition, faces' recognition, classification of gene expressions, etc. It is able to infer any function of nonlinear decision by means of a single layer of hidden neurons and functions of sigmoidal activation [40].

In OCR, the retrieved primitives on a character's image (or of the chosen entity) constitute the network inputs. The activated network output corresponds to the recognized character. The choice of the network architecture is a compromise between the computational complexity and the recognition rate [36].

II.7 Post-processing phase

The objective of the post-processing is to improve the rate of recognizing words (as opposed to the character's recognition rate). This phase is usually implemented as a set of tools related to the character frequency of occurrence in a string, lexicons and other contextual information.

As classification can lead to several possible candidates, the post-processing aims to make a selection of the solution by using higher levels of information (syntactic, lexical, semantic ...) [5]. The post-processing is also responsible for checking whether the answer is correct (even if it is only one) based on other information not available to the classifier.

II.8 some work done in this area

The recognition of the Arabic script back to the years 70. Since then, several solutions have been proposed. We present here some recent work conducted in this Area.

In [41] Somaya Alma'adeeb et al proposed a recognition system that is based on HMM (Hidden Markov Models), which allowed having a recognition rate of 45.0% without post-processing.

Mustapha Kadri in [1] presents a complete system of offline recognizing the handwritten Arabic script, using the neural network SPIKE (SNN) (a pulse) and the separator Vaster Marge (SVM). The rate of recognition that he gets is 76% for the SVM method, and 69% for the NSS method (Modified Newton Method of solving system).

Another AOCR system based on a wavelet compression, gave a rate of 80.0% at the expense of analysis time [42].

The described system in [43], based on DWT (The Discrete Wavelet Transform) has achieved a recognition rate of approximately 90.0%, with a relatively low rate for the characters in the middle. These results are always detrimental to the system's speed due to the use of wavelets.

We find in [44] a system called ASCA (Analysis of variance – simultaneous component analysis) combined with an existing system RECAM (Review Command Assessment of project), based on morphological analysis of the word's contour. The topological features of the text are used to extract morphological rules.

In [2], a segmentation approach applied to Arabic handwriting, which allows offline reconstruction of a similar tracing way to that in the case of on-line. With the use of a semi-skeletonization technique for the tracking and the calculation of the characters' characteristics. With the application of SVM classifier in the classification phase, they reached some interesting recognition rates in reduced times.

II.9 Conclusion

We have presented in this chapter the concept of characters' recognition in a general way, with an update on the morphological characteristics of the Arabic script and the aspects of OCR (Optical character recognition) as well as the different phases of the recognition process. In the end, we presented some recent work in the field of Arabic writing's recognition.

We saw that the recursion of the Arabic script shows a complexity of the characters' morphology. Faced with this problem, there is a necessity of a robust modelization and an effective learning method to take into account all the morphological variations of the Arabic script.

We have also tried to expose the different methods used in the word segmentation. These methods have seen much progress in recent years. Various techniques influenced by developments in areas such as speech recognition and on-line character recognition have emerged.

Effective segmentation generally depends on several factors:

- The nature and quality of the document,
- The acquisition tool (scanner), and
- Pre-processing methods and selected segmentation algorithms.

For this reason, our choice was oriented towards the multi-class model SVM (support vector machines) which is very successful for the resolution of the learning problem in general and of the OCR in particular. This is the aim of the next chapter.

Chapter III

Extraction of parameters & recognition's algorithm (SVM)

III.1. Introduction

After the pretreatment phase, most OCR systems isolate the characters before recognizing them. This is done during the segmentation. Segmenting a text page can be divided into two steps [1]:

· Decomposition of the page, and

· Segmentation of words.

The decomposition of the page consists in separating the different elements of the page, thus producing rows and pseudo words (PAWs) from the text blocks [48]. When working with pages containing different types of objects such as graphics, mathematical formulas, text blocks ... etc.

The word segmentation consists in separating the characters of a pseudo word (PAW). The performance of an OCR system generally depends on how isolated characters [48]. Then, a classification step is made with the aim of each segmented character by giving a class label following the decision of a classifier [46].

III.1.1 Segmentation Approaches

The methods of segmentation of cursive Latin script have been studied extensive. Nevertheless, it is difficult to apply the algorithms and segmentation used for Latin scripts on Arabic writing.

Segmentation methods for Arabic words can be classified according to five approaches [47]:

1. The first approach, assumes that the input word is already segmented into characters.

2. The second approach consists in segmenting the input word into primitives smaller than the character.

3. The third approach, segments the input word into characters.

4. The fourth approach, considers is the word is recognized so that the segmentation or a sub-module of the recognition module.

5. The fifth approach treats the word without segmentation.

III.1.1.1 First approach: isolated characters

The approach is based on isolated characters, as it mainly deals with scripts in isolated characters. Although, isolated characters are rarely used in Arabic writing, except in a few words and in mathematical formulas. This Approach is only used for specific cases, and such systems under a segmentation system that identifies the characters inside the word, before recognizing it [49].

III.1.1.2 Second approach: primitives small than the character

In the PAW is segmented at all locations appear to be connection points. Then, it is possible that it primitives smaller than character, points of intersection, points of inflection and loops. The usual pattern of recognition recognizes the primitives and then combine them into characters .

This approach is used, for segmentation the two types on-line descriptions and offline thinning. It can identify set of connection points which can include all points of connection than to directly identify the segmentation points. Then, it is the position of the decision according to the a priori knowledge that it possesses. Particularly, this method is handwriting recognition characters are ambiguous [48].

III.1.1.3 Third approach: characters

The character-based approach attempts to correctly segment a word into characters then recognize them. So, the segmentation step has become the critical in the process of recognition. Many of the techniques used in this approach is like those used in the previous approach. Modified to prevent dissection of character in addition to a part.

There are methods that have a thinned word into characters by following the line base of the word, by detecting when the pixels descend below this line [48]. For example, the proposed IRAC II system

In [50], segments at the end of a tooth (ســ) several teeth (example 3 teeth in the case of the letter (س) and characters with a single tooth must have diacritical points above or below. The system examines the points around the tooth for decide whether to segment.

Other methods, search for segmentation points along the baseline to using vertical projection histograms. These points are defined as where the histogram falls below a certain threshold.

Researchers use different methods, because the same character may have inside several descents, to prevent breakage of a character in more than one share. For example, some researchers use heuristic rules that prevent the segmentation of characters of width less than a certain value.

Others modify the vertical projection histogram by multiplying each entry by the height of the column relative to the baseline. This has the effect of amplifying the distance from the baseline so as not to consider the points away from the baseline as connection points [50].

III.1.1.4 Fourth approach: module sub-module segmentation recognition

This approach recognizes the characters of a connected word on the prior segmentation). Some systems that adopt this method begin at the extreme right of a pseudo word and examine a set of columns (width equal to the closest character) and try to recognize it. If recognition fails. They repetitively add other columns, until character recognition. Once a character is recognized, it is removed from the sub word and the process is repeated [48].

This type of approach poses a problem when the system fails to know of a character in any part of the word and especially at the beginning the remainder of the word will not be treated. The solution is to use a left to right, triggered when the system fails to recognize a character in the middle [48].

III.1.1.5 Fifth approach: without segmentation

The word is recognized as an entity by the systems of this approach. Generally, global comparison techniques are used to compare the input words to others stored in a database. However, this approach is limited to the recognition of a predefined set of words (example: computer commands in pen-based computers), and is not practical for general recognition of rich vocabulary text [48].

After presenting the existing approaches, we note that the domain of character segmentation of words is very broad, and there are classes of methods to be able to scan a multitude of facets of the calligraphy of writing.

By examining the bibliography concerning the segmentation of characters of writing Arabic, we found five classes of methods. One class uses the basis of its segmentation the skeleton of the words of the text, a second uses as the outline of the words to be segmented into characters, a third vertical and horizontal projection histograms, a fourth class uses the storages, and a fifth using slippery windows.

Approach	Principle	Disadvantages
First approach	-used in the case of Isolated characters.	- cannot be used for The Arabic script, - requires a sub system Segmentation that Identifies characters to Within the word.
Second approach	- cutting the PAW into Smaller primitives That the character (such as The features, points Of intersection, points Of inflection and loops) - recognizes Primitives (for the Then combine characters).	- it is the role of phase Classification Decide which are the Segmentation points, Following knowledge to Priority it possesses.
Third approach	- the word is segmented Correctly in characters, - characters are recognized.	- changes to Methods of Segmentation of Second approach to Preventing dissection Character in addition to a part.
Fourth Approach	- the word is recognized on Place (without segmentation prior), - starting at the extreme Right of a pseudo word and By examining a Set of columns Width equal to the character the closest), - if recognition fails, He added iteratively Other columns up to Recognition of a character.	- if the system fails to Recognition of Character in any What part of the word and Especially at the beginning Rest of the word will not be treaty.
Fifth Approach	- the word is recognized As an entity, in Using techniques Of global comparisons (To compare the words Input to other stored In a database, of data).	- limited to the Recognition of Word set predefined, - is not practical for the General recognition From text to rich vocabulary.

Table III.1: Summary table of advantages and disadvantages of segmentation approaches.

III.1.2 Classification

In the literature, there are a variety of classifiers such as: KPV, Neurons, and MMC ... etc. Whatever the classifier chooses, the system of recognition keeps two important phases:

- Learning, and
- Decision.

Depending on the type of classifier, there is a difference in terms of characteristic vector, learning time, execution time or recognition time and hence a variation in classification certainty [51].

III.1.2.1 Apprenticeship

In artificial intelligence, learning is represented by two numerical and symbolic currents - which exploit respectively statistical and logical formalisms respectively. This is mainly the numerical aspect that we will consider here [44]. The examples are represented by a set of input / output pairs. The goal is to learn a function that corresponds to the examples seen and that predicts the outputs for the inputs that have not yet been seen. This requires choosing:

- Good examples,
- The kernel function and the appropriate parameters, etc.

Learning is a crucial phase because decision-making results are based on parameters set during this phase [52].

III.1.2.2 Decision

To make a classification is to determine a decision rule capable of from external observations, to assign an object to one of several classes. The simplest case is to discriminate two classes [53].

The following table illustrates the recognition rates for some classifiers and with different learning sets.

III.2 Choice and Proposal of Methods

In this section, we will see the proposed segmentation method, as well as the classifier chosen for it to be used in our system.

III.2.1 Proposed Segmentation Approach

We have seen in paragraph I.1 that for the segmentation of Arabic words it there were five different approaches. The first approach assumed that the words were already segmented at the entrance. The second segmented the words into primitives smaller than the character and that it was more appropriate to the case of the manuscript. The third segment segmented the word into characters to recognize them, adopted for printed character cases. The fourth approach recognizes words without prior segmentation, using morphological primitives and models for comparison. The problem with this approach is that the definition of the primitives depended on the size of the characters and their font, which limited the number of fonts and the font sizes. In the fifth approach, whole words were recognized without segmentation. This posed the problem of limiting vocabulary.

In this work, we present a simple structural segmentation method that belongs to the second class of approaches. The principle is simple, by scanning the image of the source PAW (see Figure III.3) in the reading direction (From right to left for Arabic) plus another sweep from bottom to top and following the following rules:

1. the first segment or the last segment detected is marked as segment bearer of character,

2. a division or merger with a segment marked as carrier of a simple character (level 1, see figure III.1), eliminates the mark,

3. A division or merger operation with a segment marked as complex character (level 2), launches the division into two segment and evaluated the set of segments under processing as of the level 3 segments, then a character is obtained.

The sequence in Figure III.2 - III.5 illustrates these steps.

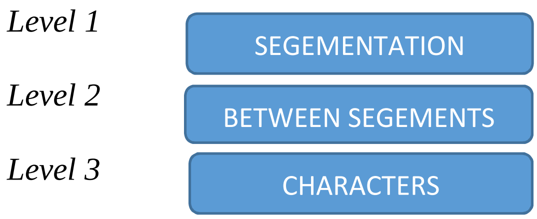


Figure III.1 different levels of features extraction

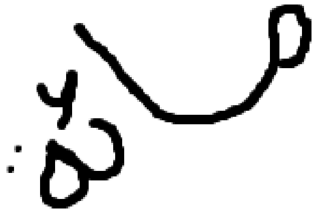


Figure III.2: Initial state of the word "مائة" م

For each step of the overall scanning cycle, a whole scan cycle from the purpose of extracting the definition of the points of division or Fusion, as shown in the following figure.

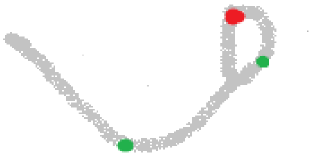


Figure III.3 Initial Paw of the word مائة

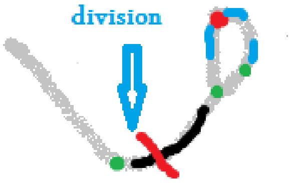


Figure III.4: The PAW "ما" after marking the character-carrying segments.

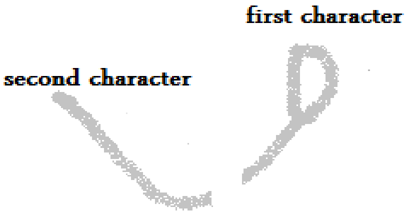


Figure III.5 Result after segmentation

The Extracting Character Characteristics is done by calculating values giving definition definitions (segment, segment, character) that respect the three axes presented in Figure III.6. Then the definitions obtained begin at level 1, giving characteristics of each segment separately, then the characteristics between level 2 segments and finally a description of the level 3 character (see Figure III.1).

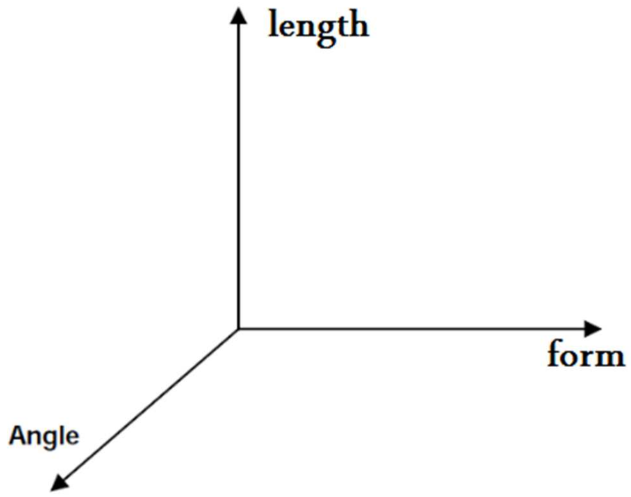


Figure III.6: Basic axes for characteristic extraction.

The present work does not follow a skeletonizing algorithm, but followed very simple, then:

- No loss of representation information;
- Gain of execution time;
- Great opportunity for effective recognition of writers, and Styles of writing.

The extracted midpoints are used to calculate during the scan the length and angle of deflection of the segment being processed. Dots of ends are used to extract information about the shape of the segment.

III.2.2 Selected classification method

In this work, we have chosen to use an SVM classifier to have a compromise of choice between: a characteristic vector of large size, time reduce learning, and an execution or recognition time reasonable, also, a classification accuracy is targeted.

III.3 Why Support Vector Machines (SVM)?

The support vector machine algorithm was developed in the 1990s by the Russian Vladimir Vapnik. Initially, SVMs were developed as a supervised binary classification algorithm. It is particularly effective in that it can deal with problems involving large numbers of descriptors, that it provides a single solution (no local minimum problems as for neural networks) and provides good results on real problems [53].

The algorithm in its initial form consists in seeking a linear decision boundary between two classes, but this model can be considerably enriched by projecting itself into another space allowing to increase the separability of the data. We can then apply the same algorithm in this new space, which results in a non-linear decision boundary in the initial space [53].

III.4 Statistical learning and SVM

Since the concept of learning is important, we will start by making a reminder. Learning by induction leads to conclusions by examining specific examples. It is divided into supervised and unsupervised learning. The case of SVM is supervised learning. The specific examples are represented by a set of input / output pairs. The goal is to learn a function that corresponds to the examples seen and that predicts the outputs for the inputs that have not yet been seen. The entries can be descriptions of objects and the outputs the class of objects given as input [52].

III.4.1 Objective of statistical learning

Executing a classification involves determining a decision rule capable of, from external observations, assigning an object to one of several classes. The simplest case is to discriminate two classes. In a more formal way, the bi-class classification amounts to estimating a function f: x → {+ 1, -1} from a learning set consisting of pairs (Xi, Yi), that we suppose i.i.d. according to an unknown probability distribution P (X, Y), such as

$$(Xi,\ Yi) \in X \times Y \text{ while } i = 1 \quad Nx \text{ and } Y = \{+1,-1\} \qquad (3.1)$$

So that f correctly classifies unknown examples (xt, yt). For example, we can assign xt to the class (+1) if f (Xt) ≥0, and to class (-1) otherwise. The unknown examples are assumed to follow the same probability distribution P (x, y) as those of the learning set. The best function f is that obtained by minimizing the risk:

$$R[f] = \int L\ [f(x),\ y]\ d\ P(x,\ y) \qquad (3.2)$$

Where L denotes a cost function, for example:

$$L\ [f(\mathrm{x}),y] = (f(x)-y)^2 \qquad (3.3)$$

Unfortunately, the risk (4.2) cannot be directly minimized insofar as the underlying probability distribution P (x, y) is unknown. Also, we will look for a decision function close to the optimal one from which we have the learning set and the function class F to which the solution f belongs. To do this, we approximate the minimum of the theoretical risk by the minimum of the empirical risk which is written:

$$R_{emp}[f] = \frac{1}{N_x} + \sum_{i=1}^{N_x} L(\mathrm{f}(X_i).\,y_i) \qquad (3.4)$$

It is possible to give conditions to the classifier in order that asymptotically (Nx→∞), the empirical risk (4.2) converges towards the risk (4.2). However, if there are few examples for learning (Nx small), there is a risk of over-learning (Figure III.7). To avoid over-learning, we can restrict the complexity of class F to which f belongs. Intuitively, a simple decision function (the simplest class consisting of linear functions) capable of correctly discriminating data is preferable to a complex function. To do this, we introduce a regularization term to limit the complexity of the functions of F.

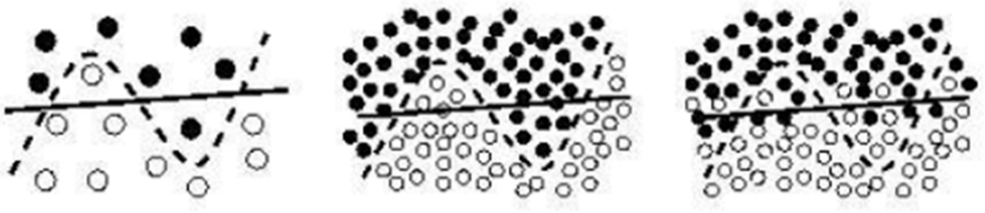


Figure III.7: Example of the over-learning problem.

Given a small learning set (left diagram), two discrimination boundaries (represented by continuous and discontinuous lines) are possible. The discontinuous line is more complex but minimizes the empirical risk. Only a larger set of examples makes it possible to determine the best of the two decision boundaries. If it is the discontinuous line, then the continuous line is not sufficiently discriminating (middle pattern); If the line is continuous, then the discontinuous line is not suitableand characterizes one on learning (right diagram) [53].

III.4.2 Theory of Vapnik-Chervonenkis

One way to control the complexity of a function class is given by the Vapnik-Chervonenkis (VC) theory and the minimization of structural risk principle. Here, the concept of complexity of the decision function f is expressed by the dimension of VC (denoted h) of the class of functions F to which f belongs. Roughly, the VC dimension measures how many samples in the learning set can be separated by all possible classifications from class functions [53].

Consider a nested family of function classes:

$F_1 \subset F_2 \subset \ldots \subset F_k$;

With a non-decreasing VC dimension, and ... $f_{1\ldots}$ $f_k$the functions minimizing the empirical risk in each of these classes.

The minimization of the structural risk consists in choosing the class Fi (and the function fi) so that an upper bound of the generalization error can be minimized (for example, by the following theorem) [53].

Theorem 1: Let h be the dimension of VC of the function class F, Remp [f] the empirical risk defined by (4.2) with the loss function 0/1 (ie L [f (xi), yi] Yf (x)) where H denotes the Heaviside function). For every δ> 0 and f∈F, the inequality limiting the risk

$$R[f] = R_{emp}[f] + \sqrt{\frac{h\left(ln\frac{2N_x}{h}+1\right)-\ln\left(\frac{\delta}{4}\right)}{N_x}} \qquad (3.5)$$

It is true with a probability of less (1- δ) for Nx> h [12].

This terminal is only an example and similar formulations have been demonstrated for other loss functions and other complexity measurements. The aim here is to minimize the generalization error R [f] by obtaining a low empirical risk Remp [f] while keeping the smallest class of functions possible.

The inequality (4.5) reveals two extreme cases:

- A very small class of functions (for example F1) causes the complexity term (the square root) to rapidly decrease, but the empirical risk remains high,
- A very large class of functions (for example f_k) implies a small empirical risk, but the term of complexity explodes.

The best class of functions is generally intermediate between the smallest and the largest, since we seek a function that best explains the data while preserving a low empirical risk (Figure III.8) [53].

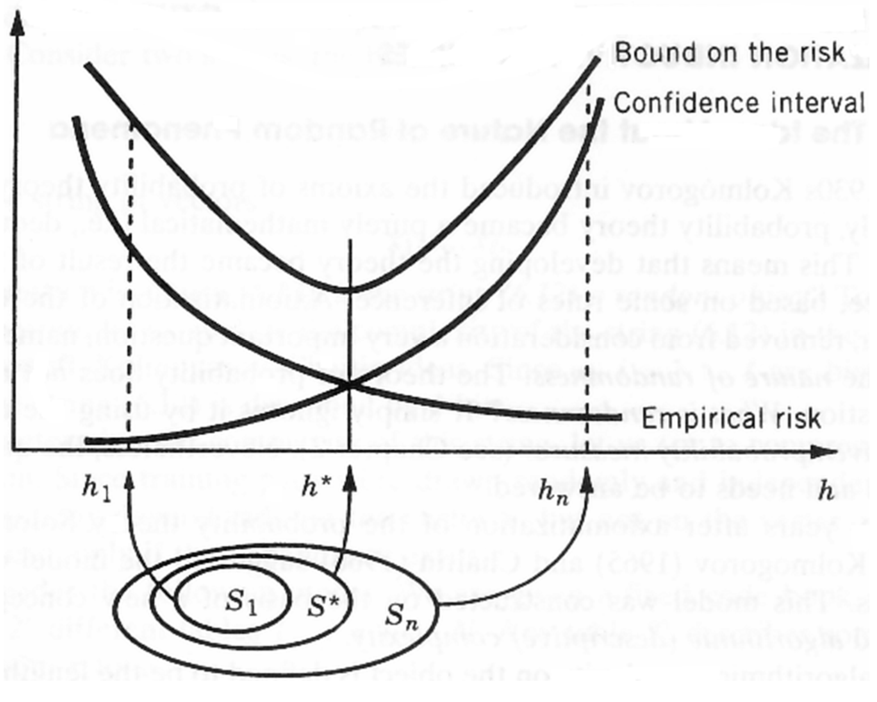


Figure III.8: Inequality illustration (4.3) [53].

The increasing curve, called confidence, corresponds to the upper bound of the complexity term. The behaviors of the complexity term and empirical error are clearly opposite. Therefore, we search for the best compromise between complexity and empirical error [53].

III.4.3 VC margin and dimension

Suppose for the moment that the samples of the learning set are separable by a hyperplane (Figure III.8), i.e. we choose decision functions of the form:

$$f(x)= w, x* +b \qquad (3.6)$$

The margin is the minimum distance between the samples of the learning set and the decision boundary.

It has been shown that for the class of hyperplanes, the dimension of VC can be bounded as a function of the margin. The margin can in turn be measured using the weight vector w: since we assume that the samples are separable, we can redefine w and b so that the samples x closest to the hyperplane satisfy | <W, x> + b | = 1.

Consider now two samples x1 and x2 of different classes such that we have <w, x1> + b = + 1 and <w, x2> + b = -1. The margin γ then corresponds to the distance between x1 and x2 measured perpendicular to the hyperplane:

$$\gamma= w / ||w||,x1 - x2 * =2/||w|| \qquad (3.7)$$

The results linking the VC dimension of the class of separation hyperplanes to the margin and the length of the weight vector we are respectively given by the following inequalities:

Where R is the radius of the smallest ball encompassing the data. Thus, by limiting the margin of the function class, one can control its dimension of VC [54].

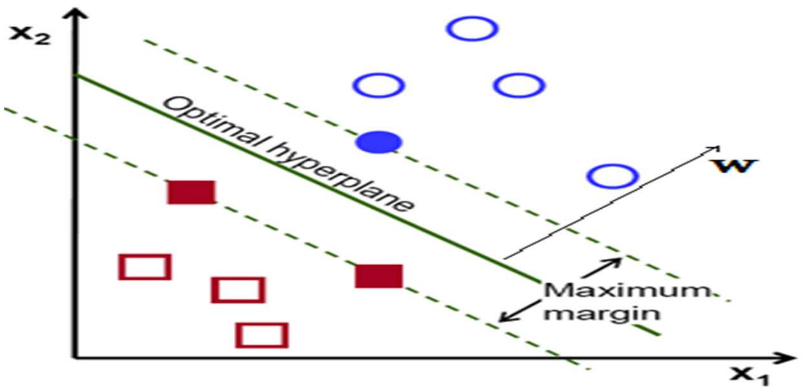


Figure III.9: Linear classifier and margin [34].

A linear classifier is defined by a vector normal to the hyperplane w and a bias b: the decision boundary is {x | <w, x> + b = 0} (continuous line). Each of the two sub-spaces separated by the hyperplane corresponds to a class, i.e. f (x) = sign (<w, x> + b). The margin of the linear classifier is the minimum distance between the samples of the learning set and the decision boundary. In the diagram, it is the distance between the continuous line and the discontinuous lines [53].

III.5. SVM general operating principle

III.5.1 Basics: Hyperplane, margin and vector support

For two classes of given examples, the goal of SVM is to find a classifier that will separate the data and maximize the distance between these two classes. With SVM, this classifier is a linear classifier called hyperplane.

In the following diagram, we determine a hyperplane which separates the two sets of points [52].

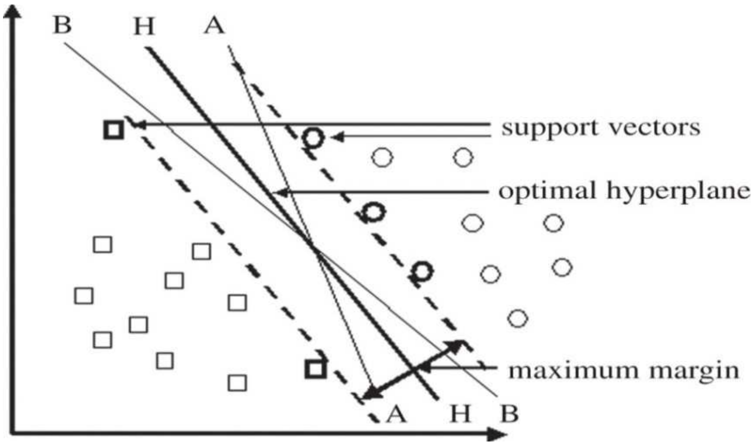


Figure III.10: Example of a hyperplane separator [52].

The closest points, which alone are used to determine the hyperplane, are called support vectors.

It is obvious that there is a multitude of valid hyperplane but the remarkable property of the SVM is that this hyperplane must be optimal. We will therefore also search among the valid hyperplanes, that which passes "in the middle" of the points of the two classes of examples. Intuitively, this consists in seeking the <<safest >> hyperplane.

Indeed, suppose that an example has not been described perfectly, a small variation will not modify its classification if its distance to the hyperplane is large. Formally, this is equivalent to finding a hyperplane whose maximum distance from the learning examples is maximal [55].

This distance is called <<margin>> between the hyperplane and the examples. The optimal separator hyperplane is the one that maximizes the margin. As we try to maximize this margin, we will speak of wide margin separators [55].

III.5.2 Why maximize margin?

Intuitively, having a wider margin provides more security when classifying a new example. In addition, if we find the classifier with the better performance in relation to the learning data, it is clear it will also be the one allowing the best classification of the new examples. In the following diagram, the right side shows us that with an optimal hyperplane, a new example remains well classified when it falls within the margin. We see on the left side that with a smaller margin, the example is poorly classified [52].

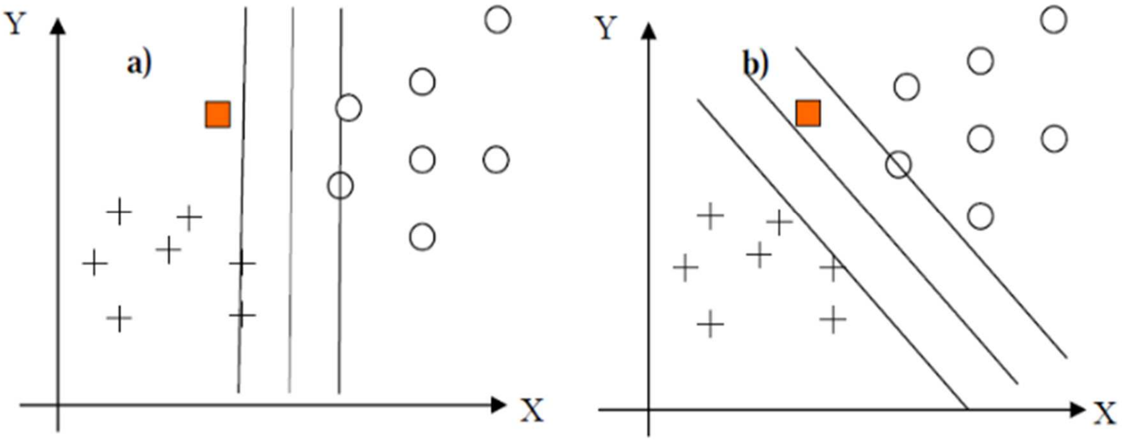


Figure III.11: a) Hyperplane with low margin, b) Best separator hyperplane [52].

In general, the classification of a new unknown example is given by its position with respect to the optimal hyperplane. In the following diagram, the new item will be classified as "+".

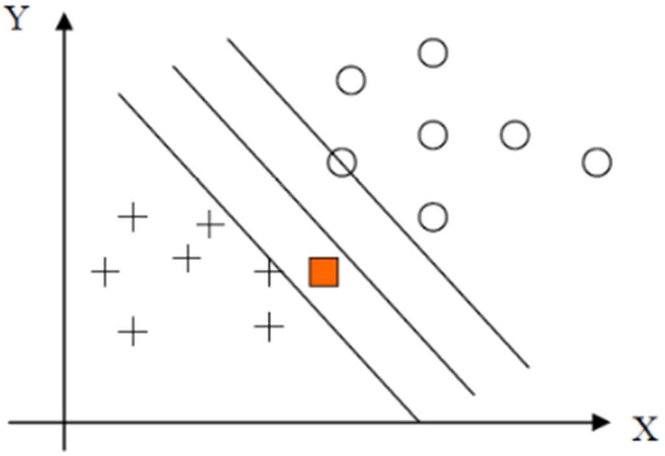


Figure III.12: Example of a new element classification.

III.5.3 Linearity and non-linearity

Among the SVM models, we find the cases linearly separable and the cases not linearly separable. The first ones are the simplest of SVM because they make it easy to find the linear classifier. In most real problems, there is no possible linear separation between the data, the maximum margin classifier cannot be used because it works only if the learning data classes are linearly separable [52].

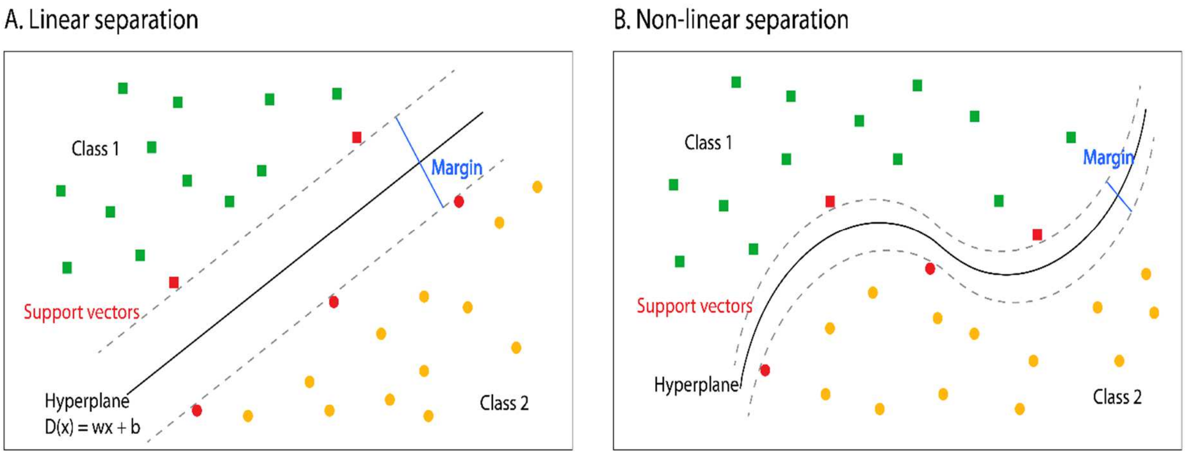


Figure III.13: a) Linearly separable case, b) Non-linearly separable case [52].

III.5.4 Nonlinear Case

To overcome the disadvantages of non-linearly separable cases, the idea of SVM is to change the data space. The nonlinear transformation of the data can allow linear separation of the examples in a new space. So, we're going to have a change in size. This new dimension is called a "re-description space". Intuitively, the larger the dimension of the re-description space, the greater the probability of finding a separating hyperplane between the examples. This is illustrated by the following diagram [52]:

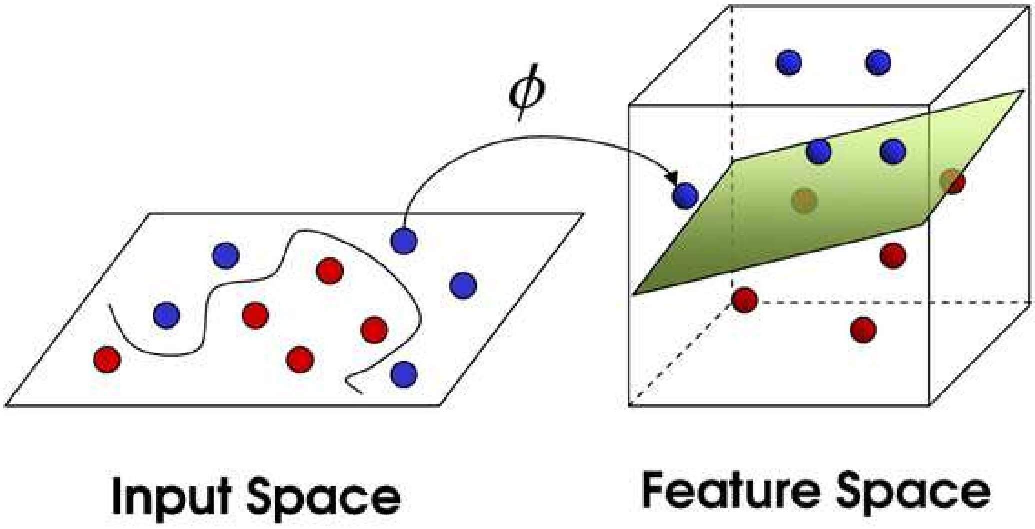


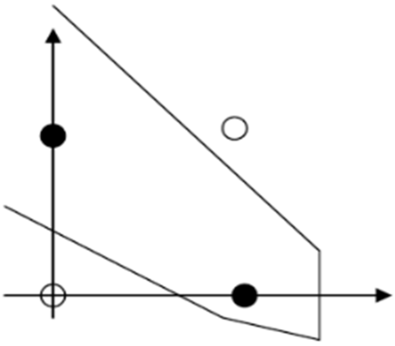
Figure III.14: Example of changing the data space.

Thus, we have a transformation of a nonlinear separation problem in the representation space into a linear separation problem in a larger dimensional re-description space. This non-linear transformation is performed via a kernel function [52].

In practice, some families of parametrized kernel functions are known and it is up to the SVM user to carry out tests to determine which one is best suited for its application. We can cite examples of the following cores: polynomial, Gaussian, sigmoid and Laplacian [52].

III.5.5 Illustration of non-linear case transformation: the XOR case

The case of XOR is not linearly separable, if we place the points in a two-dimensional plane, we obtain the following figure:



Coordinates of points: (0,0); (0.1); (1.0); (1,1)

Figure III.15: Non-linearly separable case illustration (XOR case) [52].

If we take a polynomial function (x, y) → (x, y, x.y) that passes from a space of dimension 2 to a space of dimension 3, we obtain a problem in three dimensions linearly separable:

(0,0) → (0,0,0)

(0.1) → (0.1.0)

(1.0) → (1.0.0)

(1,1) → (1,1,1)

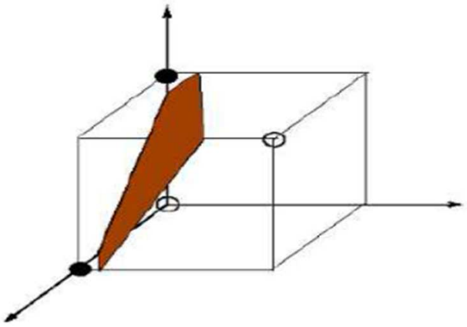


Figure III.16: Changeover from a 2D space to a 3D space illustration [52].

III.6. Mathematical Foundations

We will detail in the paragraphs below the mathematical principles on which SVM is based.

III.6.1 Learning Problem

We are interested in a phenomenon f (possibly non-deterministic) which, starting from a certain set of inputs x, produces an output y = f (x).

The goal is to find the function f from the single observation of several input-output pairs {(xi, yi), i = 1, ..., n} to "predict" other events.

We consider a pair (X, Y) of random variables with values in X x Y. Only the case Y = {-1, 1} (classification) is of interest here (we can easily extend to the case card (Y) = m> 2 and the case Y = A. The joint distribution of (X, Y) is unknown.

Knowing that we observe a sample S = {(X1, Y1),...,(X, Yn)} of n copies independent of (X, Y), we want to: construct a function h: X → Y such that P (h (X)! = Y) is minimal [56].

Illustration:

Find a decision boundary that separates space into two regions (not necessarily related).

Knowing h, we can deduce the classification of the new points, i.e. find a decision boundary.

The problem is to find a boundary rather distant from the points of different classes. This is one of the major classification problems with SVMs [56].

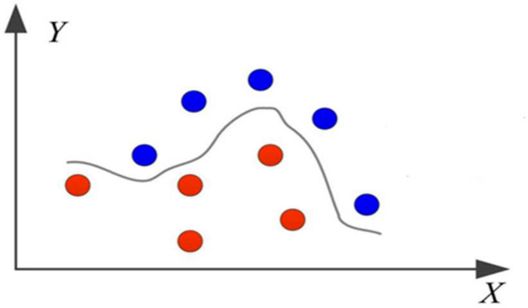


Figure III.17: Border determination problem rather far from the points of different classes [56].

Over and under-learning:

If the data is generated by a quadratic model:

The linear model is in a situation of under-learning.

The high-level model is in a situation of over-learning (learning by heart).

Therefore, an agreement must be found between data adequacy and complexity in order to be able to generalize.

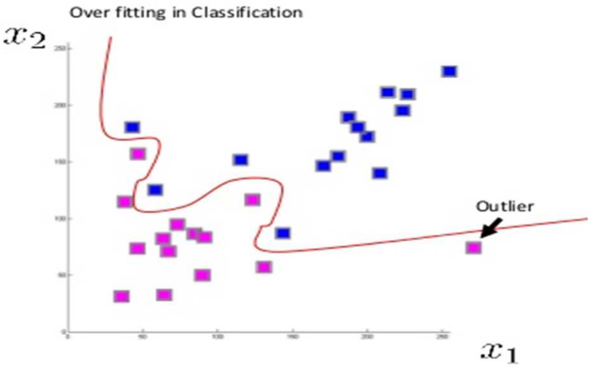


Figure III.18: Over and under-learning illustration [56].

III.6.2 Real-valued classification

R: X → R (set of real numbers). The class is given by the sign of f; H = sign (f).

The error is calculated with P (h (X)! = Y) = P (Yf (X) ≤0). This gives some idea of confidence in the classification. Ideally, | Yf (X) | is proportional to P (Y | X). Yf (X) represents the margin of f in (X, Y). The goal is to construct f and h. We will see how to achieve this [52].

III.6.2.1 Entries transformation

It may be necessary to transform entries in order to treat them more easily. X is an given space of objects. We transform the entries into vectors in a space (feature space) by a function: Φ: X → F; F is not necessarily finite but has a scalar product (Hilbert space). The Hilbert space is a generalization of the Euclidean space which can have an infinite number of dimensions. Nonlinearity is treated in this transformation, so we can choose a linear separation (we shall see later how to reduce a nonlinear problem to a classical linear problem) [52].

Therefore, it is a matter of choosing the optimal hyperplane which correctly classifies the data (where possible) and which is as far as possible from all the points to be classified.

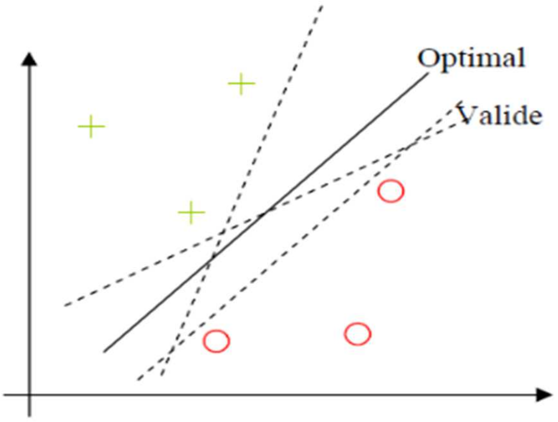


Figure III.19: Example of an optimal hyperplane search [52].

But the chosen separator hyperplane must have a maximum margin.

III.6.2.2 Maximizing the margin

The margin is the distance from the nearest point to the hyperplane.

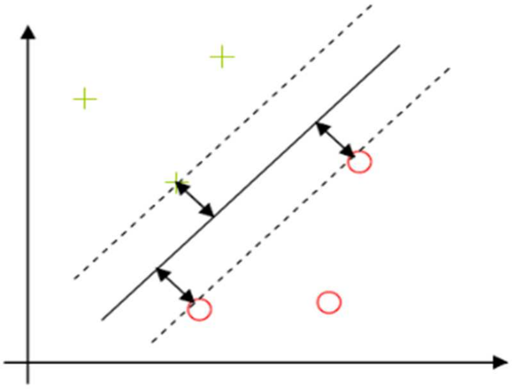


Figure III.20: Illustration of the relation between margin, support vector points and optimal hyperplane [56].

In a linear model (figure above), we have f (x) = w.x + b. The separating hyper plane (decision boundary) therefore has the equation w.x + b = 0. The distance from a point to the plane is given by d (x) = | w.x + b | / ‖ w | The optimal hyper plane is the one for which the distance to the nearest points (margin) is maximum. Let x1 and x2 be the points of different classes (f (x1) = +1 and f (x2) = -1) (w.x1) + b = +1 and (w.x2) + b = (X1 - x2)) = 2 Hence: (w / ‖ w ‖. (X1 - x2)) = 2 / ‖ w ‖ [56].

Therefore, it can be deduced that maximizing the margin consists in minimizing ‖ w ‖ under certain constraints which we shall see in the following paragraphs.

III.6.2.3 Primal problem

A point (x, y) is well classified if and only if f (x)> 0. Since the pair (w, b) is defined with a multiplicative coefficient, we define f (x) ≥ 1. From this we derive (using also the previous paragraph) the problem of minimization under the following constraints:

$$\begin{cases} min\frac{1}{2}||w||^2 \\ \forall i, y_i(x\,.x_i+b)\geq 1 \end{cases} \qquad (3.8)$$

It may be easier to minimize || w || ² rather than directly || w || [57].

III.6.2.4 Dual Problem

We pass from the primal problem to the dual problem by introducing Lagrange multipliers for each constraint.

Here we have a constraint example of learning: Tapez une équation ici.

$$\begin{cases} \max\sum_{i=1}^{n}.\ \alpha-\frac{1}{2}\sum_{ij}\alpha_i\alpha_j\gamma_i\gamma_j x_i.x_j \\ \forall i, 0\leq\alpha_i\leq c \\ \sum_{i=1}^{n}\alpha_i\,\alpha_i=0 \end{cases} \qquad (3.9)$$

It is a quadratic programming problem of dimension n (number of examples). We define the following matrix called "hessian matrix": (xi.xj) i, j which represents the product matrix of inputs X (Matrix notation allows to solve the problem in computing in an easier way) [52].

It is shown that if the α^*_i a are solutions of this problem then we have:

$$w^*=\sum_{i=1}^{n,}\alpha_i^* y_i x_i \qquad (3.10)$$

Only the α_i a corresponding to the closest points are non-zero. We speak of support vectors.

Therefore, the associated decision function is:

$$f(x)=\sum_{i=1}^{n,}\alpha_i^* y_i x_i\ .x+b \qquad (3.11)$$

However, there are cases where the entries cannot be classified linearly [55].

III.6.3 Nonlinearity (non-separable case / soft margin)

We start from the primal linear problem and we introduce "spring" variables to soften the constraints [55]:

$$\begin{cases} min\frac{1}{2}||w||^2 + c\sum_{i=1}^{n}\varepsilon_i \\ \forall i, y_i(x\,.x_i + b) \geq 1 - \varepsilon_i \end{cases} \qquad (3.12)$$

It is penalized by the exceeding of the constraint.

From this we deduce the dual problem which has the same form as in the separable case [55]:

$$\begin{cases} \max\sum_{i=1}^{n}.\ \alpha - \frac{1}{2}\sum_{ij}\alpha_i\alpha_j\gamma_i\gamma_j x_i.x_j \\ \qquad \forall i, 0 \leq \alpha_i \leq c \\ \qquad \sum_{i=1}^{n}\alpha_i\,\alpha_i = 0 \end{cases} \qquad (3.13)$$

The only difference is the upper bound C on the α.

III.6.3.1 Core function (kernel)

In the linear case, the data could be transformed into a space where classification would be easier. In this case, the most commonly used re-description space is R (set of real numbers). It is found that for nonlinear cases, this space is not sufficient to classify the entries. Therefore, we passe into a space of great dimension [56]. With card (F)> d.

Example:

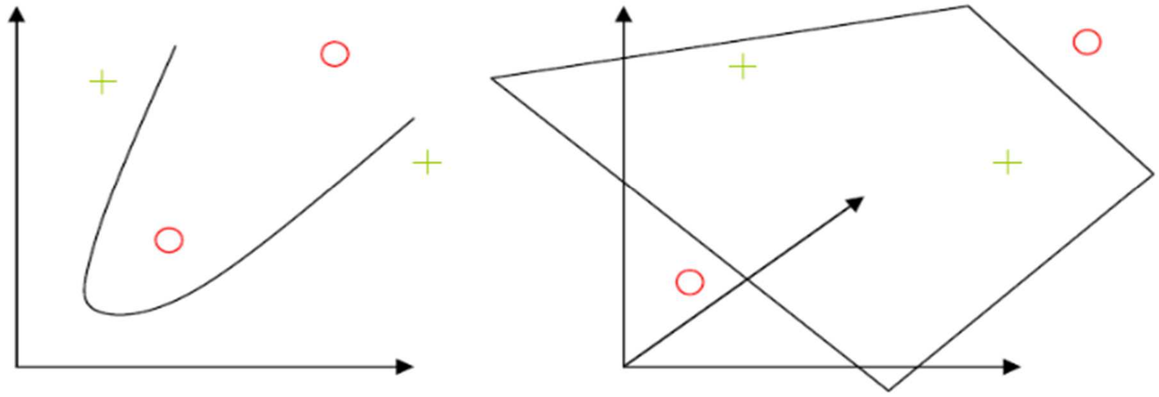


Figure III.21: Illustration of transition to R3 [56]

The transition in $F = R^3$ makes linear separation of data possible. Therefore, we have to solve:

$$\begin{cases} \max \sum_{i=1}^{n} \alpha_i - \frac{2}{2} \sum_{i.j} a_i a_j y_i y_j \emptyset\left(x_j\right) . \emptyset(x_j) \\ \forall j, 0 \leq a_i \leq c \\ \sum_{i=1}^{n} a_i y_i = 0 \end{cases} \qquad (3.14)$$

And the solution has the form:

$$f(x) = \sum_{i=1}^{n} . a_i^* y_i \emptyset(x_i) . \emptyset(x) + b \qquad (3.15)$$

The problem and its solution depend only on the scalar product ø(x) ×ø(x’). Rather than choosing the non-linear transform: ø : X F, we choose a function K: XxX R (real numbers) called the kernel function.

It represents a scalar product in the intermediate representation space. Thus, k is linear (which allows us to reconcile with the linear case of the preceding paragraphs). Therefore, this function expresses the distribution of the examples in this space k (x, x ') = Φ (x) .Φ (x'). When k is well chosen, it is not necessary to calculate the representation of the examples in this space to calculate Φ .

Example:

$$x = (x_1, x_2) \; and \emptyset \; (x) = (x_1^2, \sqrt{2} \; x_1 x_2, {x_2}^2) \qquad (3.16)$$

Let be:

In the intermediate space, the scalar product gives

$$\emptyset(x). \emptyset(x^{\cdot}) = x_1^2 \; x_1^{\cdot 2} + 2 x_1 x_2 x_1^{\cdot} x_2^{\cdot} + x_2^2 x^2 \qquad (3.17)$$

$$= (x_1 x_1^{\cdot} + x_2 x_2^{\cdot})^2$$

$$= (x \, . \, x^{\cdot})^2$$

Therefore, we can calculate ø(x) ×ø(x') without calculating f: k (x, x ') = (x.x') ². Therefore, K will represent the kernel for the corresponding entries, however it must fulfill certain conditions [58].

III.6.3.2 Mercer Condition:

A symmetric function k is a kernel if (k (xi, xj)) i, j is a defined matrix positive. In this case, there exists a space F and a function f such that k (x, x ') = F (x) × f (x,) [56].

Problems:

· This condition is very difficult to verify

· It gives no indication for the construction of core

· It does not make it possible to know ø is.

Examples of kernels:

· Linear k (x, x ') = x. x '

· Polynomial k (x, x ') = (x, x') d or (c + x. X ') d

· Gaussian k (x, x ') = e ‖ x-x' ‖ 2 / s

· Laplacian k (x, x ') = e ‖ x-x' ‖ 1 / s.

It is generally observed that the Gaussian nucleus gives better results and data groups into net packets. In practice, simple nuclei are combined for in obtaining more complex [56]. In practice, simple coreare combined to obtain more complexity.

III.6.4 Computing time and convergence

III.6.4.1 Complexity

We will evaluate the complexity (computation time) of the SVM algorithm. She does not depends on the number of inputs to be classified (d) and the number of training data (n).

We show that this complexity is polynomial in n.

$dn^2 \leq$Complexity $\leq dn^3$

Size of the hessian matrix = n^2

Indeed, we must at least browse all elements of the matrix as well as all entries. For a very large number of learning data, the calculation time is exploding.

Therefore, SVMs are practical for "small" problems of Classification [52].

III.6.4.2 Why SVM works?

The previous kernels which are the most used, fulfill the conditions of mercer (easy to check once one has the kernel). Normally, the class (number) of the hyperplanes of Rd is dH = d +1.

Class of margin hyperplanes 1 / ‖ w ‖ Such that ‖ w ‖ $^2\leq$ c is bounded by: dH≤Min (R^2c, d) + 1 Where R is the radius of the smallest sphere encompassing the training sample S, dH may be much smaller than the dimension d of the input space X; It is therefore always possible to find one, which is why [52].

III.7 Areas of application

SVM is a classification method that shows good performance in the resolution of various problems. This method has shown its effectiveness in many fields of application such as image processing, text categorization or medical diagnostics, even on very large data sets.

The realization of an SVM learning program is reduced to an optimization problems solving in a system of consequent space dimension. The use of these programs is mainly to select a good family of core functions and to adjust the parameters of these functions. These choices are most often made by a cross validation technique, in which the performance of the system is estimated by measuring it on examples that have not been used during training.

The idea is to look for the parameters to achieve maximum performance. If the implementation of an SVM algorithm is generally inexpensive in time, it is necessary to expect that the search for the best parameters may require rather long test phases [52].

III.8 Histogram.

The histogram is a quick way to study the distribution of a variable. It can be used in quality management when the data are obtained during manufacture.

Examples:

- Diameter of a shaft after machining,
- Hardness of a series of parts after a heat treatment,
- Concentration of an element in the composition of alloys produced by a foundry,
- Mass of food preparation in a can,
- Distribution of the brightness of the pixels in a photograph.

The histogram is a "visual" tool that allows to detect certain anomalies or to make a diagnosis before starting an improvement process. Used in this framework, the histogram is a "qualitative" tool. To be able to conduct the study of the dispersion of a variable using one or several histograms, one must have a good knowledge of the variable studied. It is also necessary to know the conditions for collecting the data: measurement frequency, measuring tool used, possibility of mixing batches, sorting capability etc.

III.8.1 Collection of data

The first phase is the collection of data during manufacturing. This collection can be carried out either in an exceptional way during the study of the variable or by using an automatic or manual survey made during a control carried out as part of the monitoring of the manufacturing process.

Without it being possible to give a minimum number, the number of values must be sufficient. The higher the number of values, the easier the interpretation.

III.8.2 Number of classes

The choice of classes, their number and width, is not univocal. It is appropriate to consider both the nature of the distribution and the number of data points. Often, in an analysis of this type, classes of identical width are used.

Many suggestions for choosing the number of classes can be found in the literature. For example:

- That of Herbert Sturges (1926), which for N points of data distributed with an approximately normal distribution, suggests a few classes K obtained with the following formula:

$$K = 1 + log_2 N \approx 1 + \frac{10}{3} log_{10} N \qquad (3.18)$$

Rule of Sturges can be readily consulted in this regard.

- The alternative to the previous rule is the so-called rule of Rule 3 where:

$$K = 2 N^{\frac{1}{3}} \qquad (3.19)$$

The simple choice of the square root:

$$K = \sqrt{N} \qquad (3.20)$$

In any case, since the histogram is a visual tool, it is possible to vary the number of classes. This makes it possible to see the histogram with a different number of classes and thus find the best compromise that will facilitate the interpretation. The use of a dedicated software or, more simply, a spreadsheet facilitates this operation.

III.8.3 Class Intervals

The (minimum) amplitude w of the histogram is

W = Maximum value– Minimum value.

However, it may be of interest to obtain a more meaningful histogram of choosing an amplitude wider than the minimum amplitude.

H=W/K

The theoretical amplitude h of each class is then:

This value should be rounded to a multiple of the resolution of the measuring instrument (rounded to excess).

III.8.4 Interpretation

The distribution of many industrial parameters often corresponds to a normal distribution. The histogram obtained is often compared to the "bell" profile of the normal distribution. This comparison is visual and although it may be a first approach, it does not constitute a test of "normality". To do this, you must run a test, one of the most classic of which is Henry's right.

The distribution according to the normal law, if it is extremely frequent, is not systematic. It will be checked that the distribution does not correspond to a distribution of shape defect (example: measurement of the excentration in a tube, position of objects thrown in the direction of a wall some of which bounce on this wall).

Interpretation can, for example, yield the following results:

1. Histogram showing a mixture of two batches.
2. Histogram showing a mixture of two lots but with a near average. In this case, it is also necessary to vary the number of classes to verify that this is not a construction problem.
3. Histogram showing that the batch has been sorted. All elements for which the value of the measured parameter was lower were deleted.

In the case of a histogram showing a mixture of two batches having a different average, there are cases where the dispersion presents this appearance without incriminating a mixture. This is the case, for example, for the measurement of a cylindrical part but which has an ovalization-type defect. The two averages then represent the large diameter and the small diameter. It is the knowledge of the process and / or the product that makes this type of interpretation possible.

III.9. Otsu Method

In computer vision and image processing, the Otsu method is used to perform automatic thresholding from the shape of the histogram of the image, or the reduction of a grayscale image to an image binary. The algorithm assumes that the image to be binarized contains only two classes of pixels (the first plane and the background) and then calculates the optimal threshold separating these two classes so that their Intraclass variance is minimal. The extension of the original method to multi-level thresholding is called Multi Otsu method. The name of this method comes from the name of its initiator, Nobuyuki Otsu.

III.9.1. Method

In the Otsu method, the threshold that minimizes intra-class variance is sought from all possible thresholds:

$$\sigma_{\omega}^{2}(t)=\omega_{1}(t)\sigma_{1}^{2}(t)+\omega_{2}(t)\sigma_{2}^{2}(t) \qquad (3.21)$$

The weights M represent the probability of being in the ith class, each being separated by a threshold T. Finally, the g are the variances of these classes.

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t) \quad \omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \tag{3.22}$$

Otsu shows that minimizing the intra-class variance amounts to maximizing the inter-class variance.

Which is expressed in terms of the probabilities of class ω_i and the mean of classes $\mu_i$which in turn can be updated iteratively. This idea leads to an efficient algorithm.

III.9.1.1 Algorithm

Calculate the histogram and the probabilities of each intensity level

Define the initial $w_i(0)$ and (0)

Browse all possible thresholds t=1…. intensity max

1. Updatew_i and u_i
2. Calculate $\sigma_b^2(t)$

The desired threshold is the maximum $\sigma_b^2(t)$.

III.10 Conclusion

In this chapter, we have tried to present in a simple and complete way for segementation, as well as the concept of learning system introduced by Vladimir Vapnik, the support Vectors. We have given a general vision and a purely Mathematics of the SVM. This method of classification is based on the search for a hyperplane which allows to separate data sets as well as possible. We have presented the cases linearly separable and non-linearly separable cases that require the use of kernel function to change space. This method is applicable for two-class classification tasks, but there are extensions for multi class classification.

We then looked at the different fields of application. It exists extensions that we have not presented, including the use of SVM for regression tasks, prediction of a continuous variable as a function of other variables, as is the case, for example, in the prediction of electricity consumption according to the time of year, temperature, etc.

The scope of the SVM is therefore broad and represents a method of interesting classification.

CHAPTER IV

Results & discussion

IV-1. Introduction

In Chapter III, we presented the results of the segmentation of the handwritten words (numbers) obtained using local approach. We have already mentioned that, in proposing this approach, we sought to consistently segment several words without prior knowledge of the word itself.

The extraction of the parameters of the segmentation is carried out in a fully automatic manner. It consists in extracting the static characteristics of the word for each stroke of it, considering the results presented and the algorithm discussed (SVM)

In chapter IV, we will carry out the tests to extract the best settings for recognition.

IV-2. Description of used database

Our database has 1564 unique images for training, 24 classes as described below:

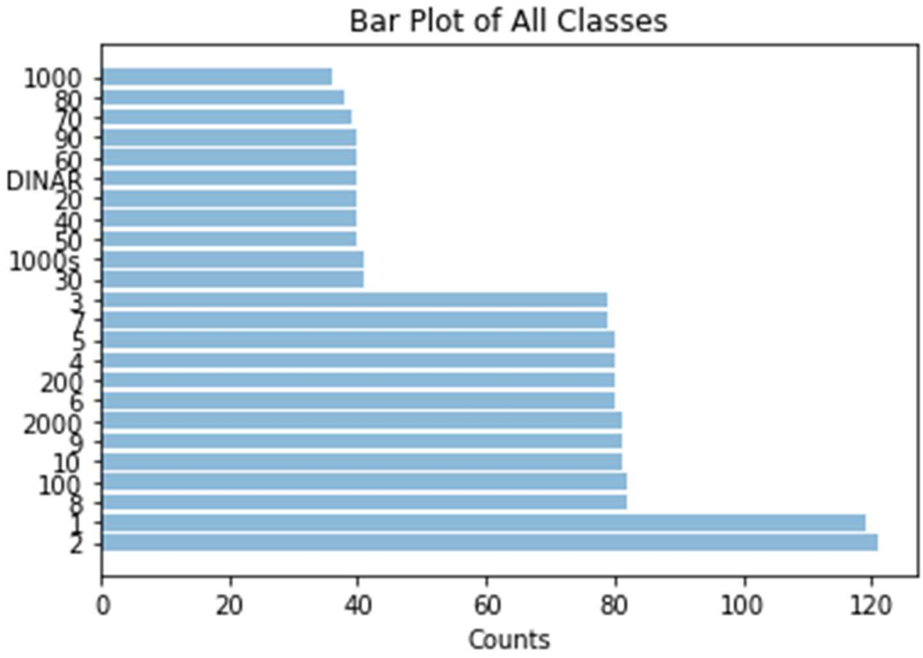


Figure IV .1 Bar plot of all classes

IV-3. Calculating False Acceptance Rate:

In this study, we have calculated the quantities: FAR (False Acceptance Rate) and FRR (False Rejection Rate) for the five individual extracted parameters of our SVM kernel

For testing purpose, we have picked 6 pictures for bulk processing.

- **Achieved Results**

We will address the plotted curves (FAR, FRR) obtained from the combinations of two type of kernels. In order to finalize the best settings for our SVM parameters.

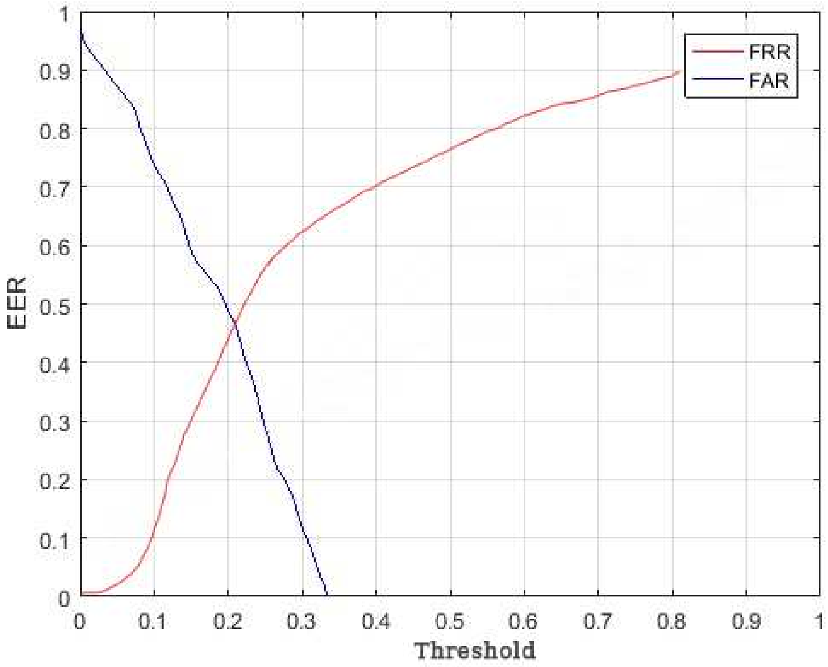


Figure IV-2: The FRRs and FARs curves for parameters I.

Parameters I: For (Figure IV-2) , we have used Kernel : RBF Gamma=0.001 C=10

we deduce that the first parameter has no great influence on our approach.

EER =0.473

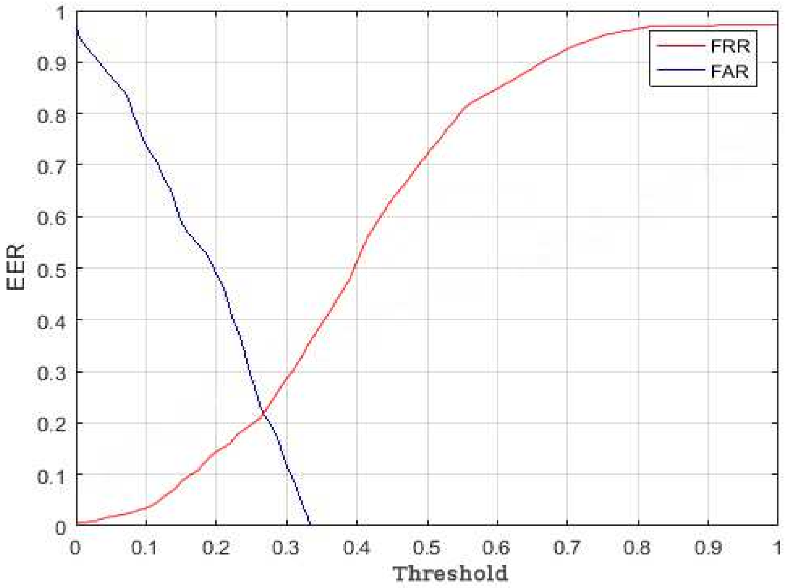


Figure IV-3: The FRRs and FARs curves for parameters II.

Parameters II: For (Figure IV-3) , we have used Kernel : Linear C=1000

We see that our system improved in terms of accuracy

EER=0.224

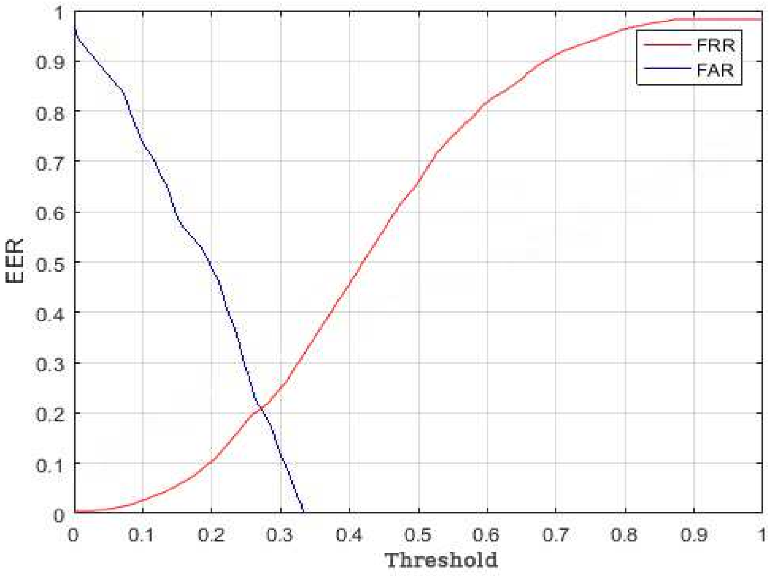


Figure IV-4: The FRRs and FARs curves for the combination of parameters I & II.

Parameters I & II combined: For (Figure IV-4) , we have used Kernel : RBF C=100

We see that our system improved in terms of accuracy

EER=0.215

This observation is the same for the ROC (Receiver Operating Characteristic) curves, which represent only the figures of the best results of the combinations mentioned above. All scales for the threshold, the equal error rate (EER) or the following for FAR and FRR are normalized in the interval (0.1).

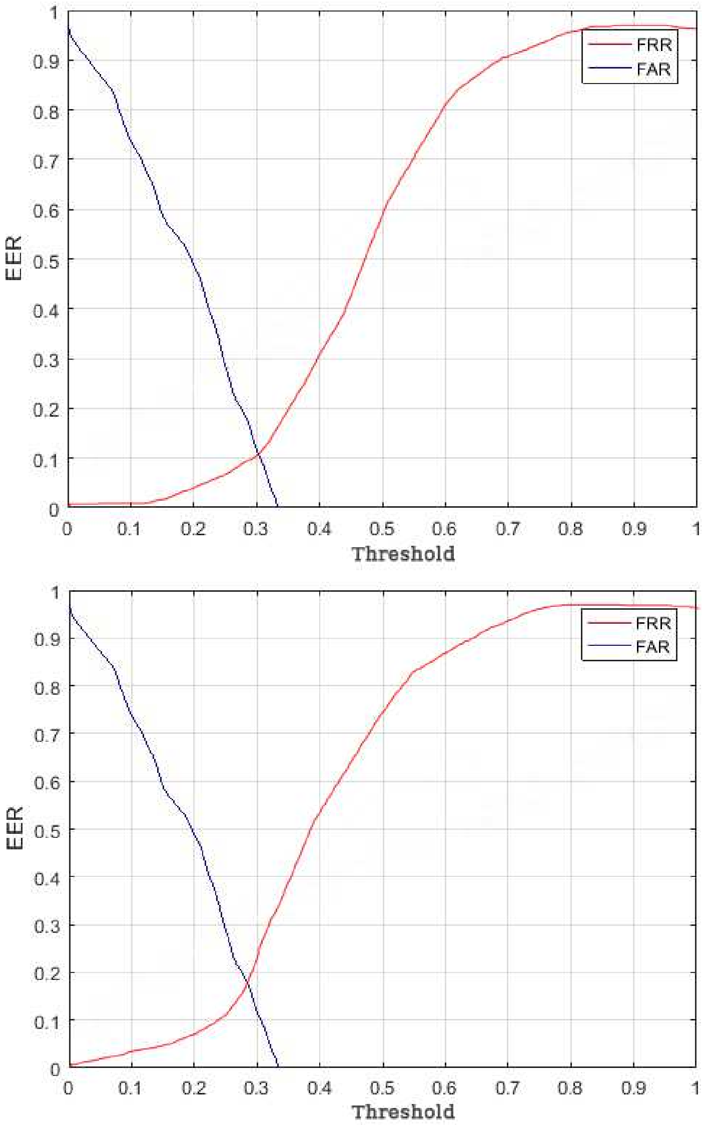


Figure IV-5: ROC curves for the best combinations of parameters.

EER=0.18 & 0.113 in the respective order

Different Kernel , High Gamma scale which is equal to 1000

These results show that the error rate with a combination of the three parameters (The kernel type, The gamma rate & the cost of classification aka C) is equal to 11.3% & the recognition rate is 88.7% better than the combination of the three first parameters.

IV-4. Conclusion

In this Chapter we have presented the different SVM parameters that can lead to a high accuracy rate in terms of Arabic handwriting recognition, using two completely different kernels.

The results of the SVM classification obtained can be improved by

Optimizing the following parameters:

· C and gamma parameters;

·The size of dataset

-The Right kernel Type

General conclusion

Despite all intensive efforts in the field optical recognition of writing, no OCR system is considered to be 100% reliable. But, step by step, image processing scientists try to improve the scores to achieve better results.

In the case of our study, we presented a structural segmentation method that proved to be efficient in terms of recognition rate. However, the major problems influencing the AOCR research are:

- the lack of standardization of Arabic calligraphy,
- the lack of in-depth studies on the classification of fonts from the point of view of calligraphy and body,
- the lack of tools such as dictionaries, databases and statistics related to Arabic writing.

The solution to these problems would be of significant help, both for the simplification of the AOCR's task and for the validation and portability of the products.

Through this work, we hope to have covered much of the field of research in the segmentation of Arabic characters and to have contributed to its evolution, despite the fact that efforts today are intensifying and each day new articles dealing with this subject are published

ملخص

التعرف على الكتابة هام جدا في وقتنا الحالي، فهو قادر على حل مشاكل معقدة كما يسهل العديد من الأنشطة التي يقوم بها الإنسان. التعرف على الكتابة العربية يعود إلى السبعينات، أين العديد من الحلول اقترحت وهي متعددة كتعدد الحلول بالنسبة للغات اللاتينية.

يمثل هذا العمل دراسة حول مجال التعرف على خط اليد العربي باستعمال التقسيم المحلي للكلمة حيث تم طرح دراسة عامة حول أنظمة التعرف على الكتابة، وبعد ذلك تم التطرق لمرحلة مهمة من مراحل التعرف على الكتابة وهي مرحلة التجزئة (حيث يتم تقسيم صورة الكلمة إلى أحرف).

قمنا بتقديم لمحة عن مختلف طرق التجزئة، يليها تقديم مجال التعرف على الكتابة، اللغة العربية، و بعض مشاكل الضبط في الكتابة العربية بإيجاز.

بعد عرض مقارنة لمختلف طرق التجزئة لخط اليد العربي، قدمنا مقترح مشاركة في هذا المجال من خلال خوارزمية تجزئة.

كلمات دلالية: التجزئة، الحروف العربية، جزء، (OCR) التعرف على الكتابة، SVM، كلمة ، المعالجة، الم□نف.

Résumé

La reconnaissance de caractère joue un rôle très important dans le monde actuel. Elle est capable de résoudre des problèmes complexes et rendre les activités de l'homme plus simple.

La reconnaissance de l'écriture arabe remonte aux années 70, depuis plusieurs solutions ont été proposées. Elles sont aussi variées que celles utilisées pour le latin.

Le présent travail porte sur une étude concernant le domaine de reconnaissance optique de caractères arabes manuscrits basée sur une approche locale. Une étude générale sur les systèmes de reconnaissance de l'écriture a été développée, puis elle a été affinée par un intérêt particulier à une phase considérée comme cruciale dans le procédé de reconnaissance la phase de segmentation.

Nous avons présenté un état de l'art des méthodes de segmentation des caractèresensuite nous avons présenté la langue arabe et le domaine de l'OCR, nous avons soulevé certains problèmes de normalisation dans l'écriture arabe.

Après que nous ayons une comparaison de méthodes de segmentation de caractères arabes manuscrits, nous avons proposé une contribution par un algorithme de segmentation.

Mots clés: OCR, segmentation, caractères arabes, pseudo mot, post-traitement, SVM.

Abstract

Optical Character Recognition (OCR) has a main role in the present time. It's capable to solve many serious problems and simplify human activities. The OCR yields to 70's, since many solutions has been proposed, but unfortunately, it was supportive to nothing but Latin languages.

This work proposes a system of recognition of an off-line Arabic handwriting. This system is based on a structural segmentation method and uses Support Vector Machines (SVM) in the classification phase.

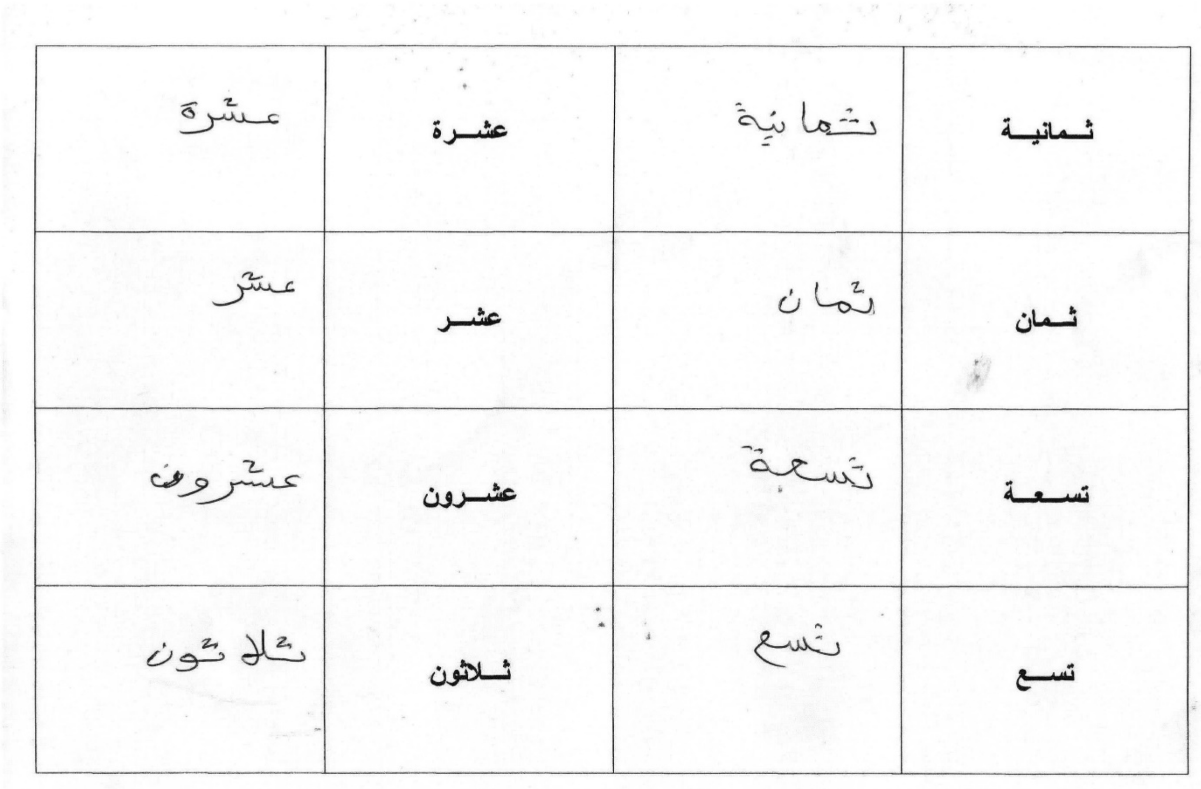
We have presented a state of art of the characters segmentation methods, after that a view of the OCR area, Also we will address the normalization problems we went through.

After a comparison between the Arabic handwritten characters & the segmentation methods, we had introduced a contribution through a segmentation algorithm.

Keywords: OCR, segmentation, Arabic characters, PAW, post-processing, SVM.

ANNEX

I-Sample of Our Database :



عشرة	عشــرة	ثما نية	ثـمانيــة
عشر	عشــر	ثما ن	ثـمان
عشرون	عشـــرون	تسعة	تسـعــة
ثلا ثون	ثــلاثون	تسع	تســع

Figure Annex-I Sample of Our database image before splitting

The dataset contains 1000 images like the one above, we collected it ourselves & used 1000 different volunteers

Each image got split into 16 small ones in order to contain one word a time, the size of every image was set to be 128 X 128 px

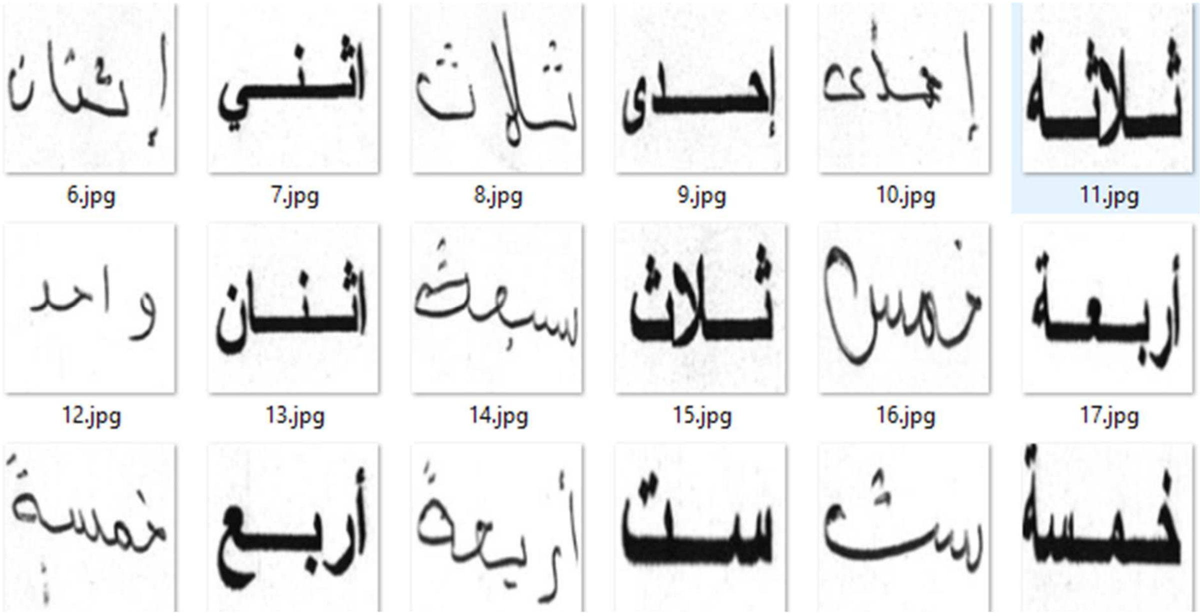


Figure Annex –II Sample of Our database image after splitting

II –Explaining the main concept of The Script

```
# Initializing imgs[] list and reading image data
# and appending them to this list.
imgs = []
for imfile in imfiles:
    img = imread(imfile, as_grey = True)
    img = preprocessing(img)
    imgs.append(np.array(img.reshape(1, imwidth * imheight)))
print('%d images are loaded.'%(len(imgs)))
```

```
#Then the Script Proceeds the binarization part
```

```
img = imgs[1201]
# (16384) => (128,128)
image_width = 128
image_height = 128
image = img.reshape(image_width, image_height)
plt.figure(num = None, figsize = (3, 3), dpi = 90, facecolor = 'w', edgecolor = 'k')
plt.axis('off')
plt.imshow(image, cmap = 'binary')
plt.show()
```

```
# Features Extraction
# Number of features = (128/ppc)^2 * orientation
```

```
#So basically the script split the image into 64 Blocks
ppc = 16
orientation = 9
imfeatures = []
for img in imgs:
    fv = hog(img.reshape(imwidth, imheight), orientations = orientation,
             pixels_per_cell = (ppc, ppc), cells_per_block = (1, 1),
                               visualise = False)
    imfeatures.append(fv)
hog_features = np.array(imfeatures, 'float64')
print('Features are extracted successfully!')

#Hog Feature
k = 150
x = np.arange(len(hog_features[k]))
y = hog_features[k]
data = {'xx' : x, 'yy' : y}
data = pd.DataFrame(data)
plt.plot(hog_features[150])
 #SVM
# Initialize the classifier.
# Percentage of data that will be used to test the classifier.
test_size = 0.30

# Constructing training and testing sets.
train, test, target_train, target_test = train_test_split(hog_features, list(labels["labels"]),
test_size = test_size);

# Building SVM classifier
classifier = svm.SVC(kernel = 'linear', probability = False, C = 1)
```

```
train_labels = pd.DataFrame(target_train)
train_labels.columns = ["labels"]
train_labels_name = list(train_labels["labels"].value_counts().index.values)
train_labels_count = list(train_labels["labels"].value_counts().values)

# Plot number of observations in each class and check to see that
# if there is any imbalance in our dataset. cross your fingers

y_pos = np.arange(len(train_labels_name))
plt.barh(y_pos, train_labels_count, align = 'center', alpha = 0.5)
plt.yticks(y_pos, train_labels_name)
plt.xlabel('Counts')
plt.title('Bar Plot of Training dataset Labels')
plt.show()
```

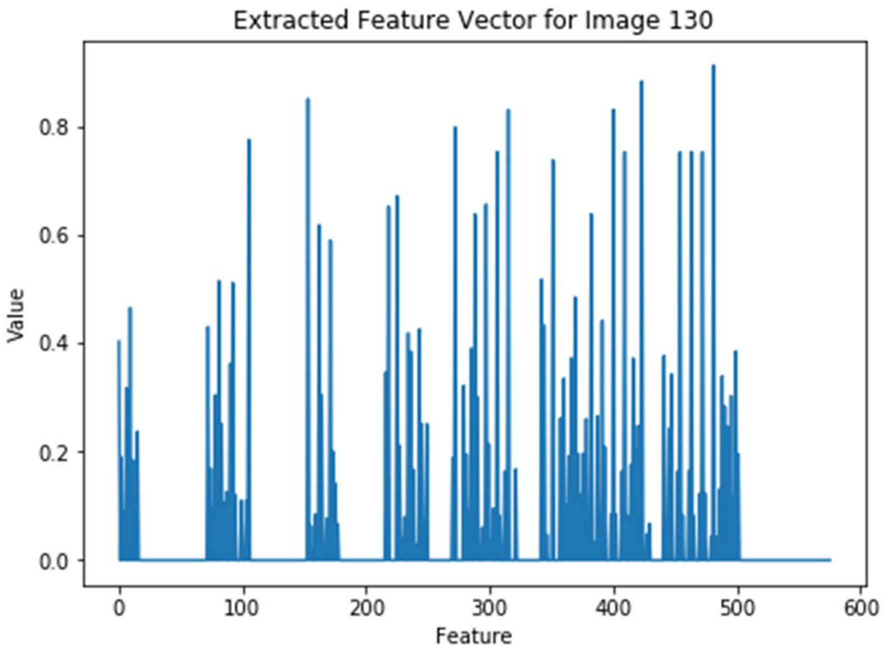


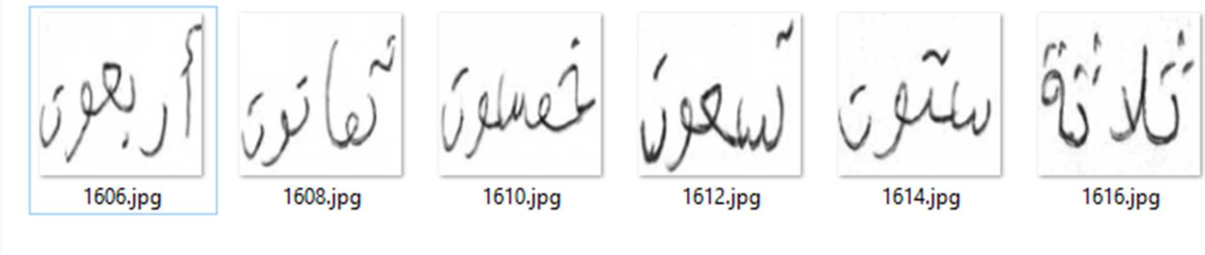
Figure Annex-3 Extracted Feature of an image vector [130]

```
#Prints the result
```

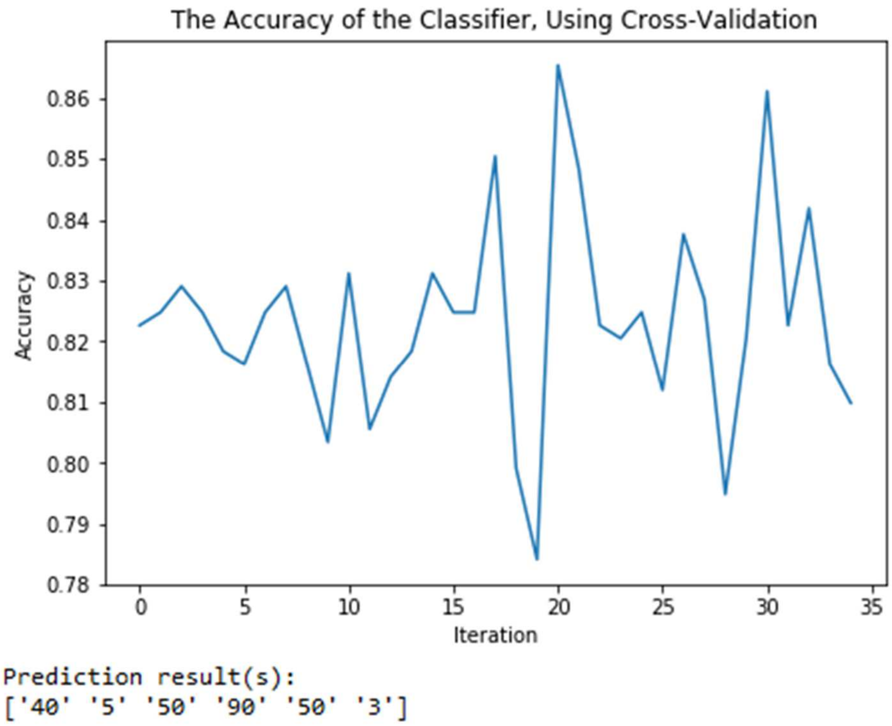
```
impath_ = 'imgs/test/*.jpg'
imfiles_ = sorted(glob.glob(impath_), key = numericalSort)
imgs_ = []
for imfile in imfiles_:
    img = imread(imfile, as_grey = True)
  img = preprocessing(img)
  imgs_.append(np.array(img.reshape(1, imwidth * imheight)))
imfeatures_ = []
for img in imgs_:
    fv = hog(img.reshape(imwidth, imheight), orientations = orientation,
             pixels_per_cell = (ppc, ppc), cells_per_block = (1, 1), visualise = False)

    imfeatures_.append(fv
hog_features_ = np.array(imfeatures_, 'float64')
pred = classifier.predict(hog_features_)
print('Prediction result(s):')
print(pred)
```

<u>**Tested Images :**</u>



Results



```
Prediction result(s):
['40' '5' '50' '90' '50' '3']
```

Bibliography & References

[1] Mustapha Kadri, « Méthode de reconnaissance de l'écriture arabe manuscrite en utilisant les réseaux neuronaux » Mémoire de magister Université des Sciences et des Technologies Mohamed Boudiaf d'Oran, 2010.

[2] Zaïz Faouzi et al « SVM pour la reconnaissance de caractères manuscrits arabes » Laboratoire LESIA, Département d'Informatique, Université Mohamed Khider Biskra, 2010.

[3] Nedjem eddine Ayat, « sélection de modèle automatique des machines à vecteurs de support: application à la reconnaissance d'images de chiffres manuscrits » Thèse de doctorat montréal, le 20 janvier 2004.

[4] Muhammad Sarfraz, « Computer -Aided intelligent recognition techniques and applications » (section : Offline Arabic Character Recognition) King Fahd University of Petroleum and Minerals, Kingdom of Saudi Arabia, 2005.

[5] S. Snoussi maddouri, « Modèle perceptif neuronal à vision globale-locale pour la reconnaissance de mots manuscrits arabes », 2002.

[6] Haitaamar Schahrazed, « Segmentation de textes en caractères pour la reconnaissance optique de l'écriture arabe ». Thèse de magister université Elhadj-Lakhdar Batna, 2007.

[7] Richard Alan Peters II, « Image processing (digital image and matlab, frequency filtering) », 2007.

[8] Somaya A. S. Al-Ma'adeed, « Recognition of Off-line Handwritten Arabic Words », University of Nottingham for the degree of Doctor of Philosophy, June 2004.

[9] N. Ben Amara « Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée ». These de doctorat, specialite Genie Electrique, Universite des sciences, des Techniques et de medecine de Tunis II, 1999.

[10] B. Al-Badr, S.A. Mahmoud: « Survey and bibliography of Arabic optical text recognition ». Signal processing, vol. 41, pp. 49-77, 1995.

[11] I.R. Tsang: « Pattern recognition and complex systems ». These de doctorat, université d'Anterwerpen, 2000.

[12] Fahmy, S.Al Ali: « Automatic recognition of handwritten Arabic characters using their geometrical features ». Studies in informatics and control journal (SIC journal), vol. 10, No 2, 2001

[13] J. Anigbogu : « Reconnaissance de textes imprimes multi-fontes a l'aide de modèles stochastiques et métriques ». thèse de doctorat, Université de Nancy I, 1992.

[14] Riadh Bouslimi, « Système de reconnaissance hors-ligne des mots manuscrits arabe pour multi-scripteurs » mémoire de master soutenue le 21/10/2006.

[15] Abdel Belaïd, « Reconnaissance automatique de l'écriture et du document » LORIACNRS Campus scientifique B.P. 239 54506 Vandoeuvre-lès-Nancy, 2002

[16] A. Belaïd, « Reconnaissance de l'Écriture et Analyse de Documents : Numérisation, Prétraitements », 2006.

[17] Maged Mohamed Mahmoud Fahmy, « Automatic Recognition Of Handwritten Arabic Characters Using Their Geometrical Feature », 2000.

[18] O. D. Trier and T. Taxt. « Evaluation of binarization methods for document images, On Pattern Analysis and Machine Intelligence », vol. 11, n.12, pp. 312- 314, December 1995.

[19] Y. Liu and S. Srihari. « Document image binarization on texture features, On Pattern Analysis and Machine Intelligence », vol. 19, n.5, pp. 540-544, May 1997.

[20] H. Emptoz, F. Lebourgeois, V. Eglin, Y. Leydier. « La reconnaissance dans les images numérisées : OCR et transcription, reconnaissance des structures fonctionnelles et des métadonnées », 2003.

[21] Sylvain Chevalier, « Reconnaissance d'écriture manuscrite par des techniques markoviennes: une approche bidimensionnelle et générique ». Thèse de doctorat de l'Université René Descartes - Paris 5, 2004.

[22] A. Belaïd, « Analyse et reconnaissance des documents », http://www.loria.fr/~abelaid/Teaching (support de cours), 2006.

[23] Dr Suici-Meslati, « Reconnaissance de formes et écriture arabe manuscrite », support de cours 2008.

[24] Najoua Ben Amara et al, « Modélisation Pseudo Bidimensionnelle pour la Reconnaissance de Chaînes de Caractères Arabes Imprimés », 1999.

[25] Yousef Al-Ohali, MohamedCheriet et Ching Suen, « Databases for recognition of handwritten Arabic cheques ». Pattern Recognition Society 36 111 – 121. 2003.

[26] Najoua Ben Amara et al, « Une méthode stochastique pour la reconnaissance de l'écriture arabe imprimée », 1996.

[27] Philippe BOLON ET AL, « Analyse d'image, Traitements de bas niveau » (support de cours), 1998.

[28] M. Mansour, M. Benkhadda & A. Benyettou, « Optimized Segmentation Techniques for Handwritten Arabic Word and Numbers Character Recognition », 2005.

[29] S. Kermi : « Classifieur neuronal base connaissances, application a la reconnaissance des caractères arabes isoles manuscrits ». Thèse de magister, université Badji Mokhtar, Annaba, Algérie 1999.

[30] C.H. TEH et R.T. CHIN: « On image analysis by the methods of moments ». IEEE Trans. Pattern Analysis and Machine Intelligence, 10(4):496–513.

[31] A. Choksuriwong et al, « Etude Comparative de Descripteurs Invariants d'Objets » ENSI de Bourges – Université d'Orléans Laboratoire Vision et Robotique - UPRES EA 2078, 2005

[32] Denis Arrivault, « Apport des Graphes dans la Reconnaissance Non-Contrainte de Caractères Manuscrits Anciens » Thèse de doctorant de l'université de Poitiers, 2006

[33] Muhammad Sarfraz, « Computer-Aided intelligent recognition techniques and applications » (section: Offline Arabic Character Recognition) King Fahd University of Petroleum and Minerals, Kingdom of Saudi Arabia, 2005.

[34] Trevor Hastie and Patrice Y. Simard, « Metrics and Models for Handwritten Character

Recognition ». Statistical science Vol. 13 N° 1, 54 -65, 1998.

[35] Sander Bohte, « Spiking neural networks », Thesis University Leiden.2003,

[36] Antoine Cornuéjol et Laurent Miclet, Livre : « Apprentissage artificiel : concepts et algorithmes », Préface de Tom Mitchell. Édition EYROLLES deuxième tirage 2003.

[37] P. J. Werbos. Beyond regression : « New Tools for Prediction and Analysis in the Behavioral Sciences ». PhD thesis, Masters Thesis, Harvard University, 1974.

[38] D. E. Rumelhart et al. « Learning internal representations by error backpropagation ». In J. L. McClelland In D. E. Rumelhart and the PDP research group, editors, Parallel distributed processing: explorations in the microstructure of cognition, pages 318–362. 1986.

[39] Y. LeCun et al. « Backpropagation applied to handwritten zip code recognition ». Neural Computation, 1(4):541–551, Winter 1989.

[40] A. Kolmogorov. « Three approaches to the quantitative definition of information ». Problems Inform. Transmission, 1(1) :1–7, 1965.

[41] S. Alma'adeed, C. Higgens, D. Ellima,« Recognition of Off-Line Handwritten Arabic Words Using Hidden Markov Model Approach », ICPR, University of Nottingham, 2002.

[42] Aburas A. A., Rehiel S. M. A., « Off-line Omni-style Handwriting Arabic Character Recognition System Based on Wavelet Compression », International Islamic University Malaysia, Electrical and Computer Engineering, Malaysia, 2007.

[43] Jannoud I. A., « Automatic Arabic Hand Written Text Recognition System ». Damascus University, Damascus, Syria and Al-zaytoonah University, Amman, Jordan, American Journal of Applied Sciences 4 (11): 857-864, 2007.

[44] Sari T., Souici L., Sellami M., « Off-line Handwritten Arabic Character Segmentation Algorithm: ACSA », Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), Algeria, 2002.

[45] Abderrazak Zahour,et AL , « Contribution à la segmentation de textes manuscrits anciens » ,2004

[46] G. Lebrun, C. Charrier, O. Lezoray :" Réduction du temps d'apprentissage des SVM par Quantification

Vectorielle ". LUSAC EA 2607, groupe Vision et Analyse d'Image, 120 Rue de l'exode,

F-50000 Saint-Lô, France

[47] J. Mary :" Méthodes d'Apprentissage Avancées, SVM ". equipe TAO, LRI, Janvier 2006.

[48] S. Haitaamar :" segmentation de texte en caractère pour le reconnaissance optique de l'écriture arabe".

Université EL-HADJ LAKHDHAR Batna, Juillet 2007

[49] A. Rachidi , M. El Yassa, D. Mammass :" A Pretopological approach for handwritten isolated Arabic

characters recognition ". IRF – SIC, Ecole Nationale de Commerce et de Gestion, IRF-SIC,

Faculty of Sciences, Ibn Zohr University Agadir, Morocco

[50] I. A. Jannoud :" Automatic Arabic Hand Written Text Recognition System ". Damascus University,

Damascus, Syria and Al-zaytoonah University, Amman, Jordan, American Journal of Applied

Sciences 4 (11): 857-864, 2007

[51] M. Eldawy :" A Survey of Classification techniques ". Presentation, May 2006.

[52] Mohamadally Hasan,Fomani Boris :" SVM machine à vecteurs de support ou séparateur à vaste marge ".

BD Web, ISTY3,Versailles St Quentin, France, janvier 2006.

[53] P. Mahé :" Noyaux pour graphes et Support Vector Machines pour le criblage virtuel de molécules ".

Rapport de stage, DEA MVA 2002/2003,Septembre 2003

[54] G. Loosli et al :" Une boîte à outils rapide et simple pour les SVM* ".

CAP, RSISE&NICTA à Canberra, Australie, Novembre 2004.

[55] A. Cornuéjols :" Une nouvelle méthode d'apprentissage : Les SVM. Séparateurs à vaste marge".

Université de Paris-Sud, Orsay, France, Juin 2002

[56] O. Bousquet :" Introduction aux ' Support Vector Machines '(SVM) ". Centre de Mathématiques

Appliquées Ecole Polytechnique, Palaiseau, Orsay, Novembre 2001.

[57] P. Mahé, L. Ait-Ali :" Projet d'apprentissage statistique SVM pour l'apprentissage non supervisé".

DEA MVA, Février 2003

[58] P. Vincent :" Modèles à noyaux à structure locale ". Thèse présentée à la Faculté des études supérieures en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.) en informatique, Université de Montréal, Octobre 2003