

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement supérieur et de la Recherche  
Scientifique



المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

Ecole Nationale Polytechnique  
Département d'Electronique

**Mémoire de fin d'études**  
En vue de l'obtention du diplôme d'ingénieur d'état en  
Electronique

# Systeme de reconnaissance de texte manuscrit arabe

Dirigé par :  
Dr L. HAMAMI

Réalisé par:  
Melle BENDI Samah  
Melle DOUACHE Djamila

*Promotion : Juin 2003*

يصف هذا البحث نظام التعرف الآتوماتيكي على النص العربي نجد المشكل الأساسي لهذا النظام في إمكانية تجزئة النص إلى الحروف المكونة لها فلهذا الغرض استعملنا طريقة التجزئة حسب الحروف المتصلة ثم نقوم بتقسيم النص إلى حروف أما فيما يخص طريقة التعرف استعملنا شبكة عصبية اصطناعية متعددة الطبقات بحيث تمت مرحلة التعلم بالبرنامج التطبيقي MATLAB ثم ثبتنا النتيجة في تطبيقنا .  
مفاتيح : التعرف على الحروف ، الكتابة العربية ، الشبكة العصبية الاصطناعية .

### **Résumé :**

Ce travail décrit un système de reconnaissance de textes manuscrits arabes . L'étape de segmentation est la plus importante à la réalisation de notre travail .

Nous avons opté pour la segmentation en parties connexes , car elle s'adapte mieux à l'écriture manuscrite , ensuite nous avons développé un algorithme pour la segmentation en caractère.

Tandis que pour la reconnaissance , nous avons utilisé un réseau de neurones multicouches , tel que son apprentissage a été fait sous MATLAB , et implémenté sous BUILDER C++ .

**Mots clés :** Reconnaissance de caractères , écriture arabe , réseaux de neurones

### **Abstract :**

This work describes a recognition system for arabic handwriting .

The major problem is the segmentation ,to resolve it we use , the segmentation by connected parts , because it seems to be the adequate method for the handwriting text , after we cut out by character .

For the recognition we use , a multilayer neuronal network , the learning ( train ) is done with MATLAB and the result is include in our softwarewith BUILDER C++ .

**Key words :** character recognition , arabic writing , neuronal network .

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

En premier, nous adressons nos plus vifs remerciements à notre promotrice Mme HAMAMI, pour les conseils avisés, les connaissances dont elle nous a fait bénéficier, son suivi attentif et sa confiance qui nous a été très précieuse. Mais aussi nous tenons à la remercier pour nous avoir proposer le sujet et de nous avoir encadrer.

Nous remercions aussi Melle GUERTI, et M. BOUSSEKSOU de nous avoir fait l'honneur de bien vouloir examiner et juger notre travail, et de leur présence parmi le jury.

Nos remerciements à l'ensemble des enseignants qui ont contribués à notre formation.

Nos remerciements les plus sincères vont à Mme DOUACHE qui a contribué activement à l'élaboration de ce travail, par ses conseils et son soutien.

Enfin, nous remercions en particulier, toute personne ayant contribué de près ou de loin à l'élaboration de ce travail.

المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

**Dédicace :**

A mes parents qui m'ont toujours soutenu.  
A ma miminou chérie, ma seule et unique petite sœur.  
A toute ma famille.  
A ma Hanane.  
A ma binôme qui sans elle le travail n'aurait pas été fait.  
A tous mes amis et amies qui se reconnaîtront.

**Djamila**

A mes parents qui me sont très chers.  
A ma sœur Dounia et mes frères Salim et Mohamed Ali.  
A Karima et Riad , ainsi qu'a toute ma famille.  
A ma binôme .  
A tous mes amis et amies.

**Samah**

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# SOMMAIRE

Remerciements  
Liste des figures

المدرسة الوطنية المتعددة التخصصات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

Introduction Générale \_\_\_\_\_ 2

## 1ere PARTIE Généralités sur l'OCR

<b>CHAPITRE 1 Description d'un système de reconnaissance de caractères</b>	<b>6</b>
<b>I-1 Introduction :</b>	<b>7</b>
<b>I-2 Définition de chaque étape :</b>	<b>8</b>
I-2-1 L'acquisition	8
I-2-2 Le prétraitement	8
I-2-3 La segmentation	8
I-2-4 L'analyse	8
I-2-5 La reconnaissance	8
I-2-5-1 L'écriture en ligne (dynamique)	8
I-2-5-2 L'écriture hors ligne (statique)	8
<b>I-3 L'application de l'OCR :</b>	<b>9</b>
<b>I-4 Implémentation</b>	<b>9</b>
I-4-1 L'apprentissage	9
I-4-2 La décision	9
<b>I-5 Conclusion :</b>	<b>10</b>
<b>CHAPITRE II Acquisition et Prétraitement</b>	<b>10</b>
<b>II-1 Introduction</b>	<b>11</b>
<b>II-2 Techniques d'acquisition</b>	<b>11</b>
II-2-1 Le scanner	11
II-2-2 La caméra	12
II-2-3 La tablette graphique :	13
<b>II-3 Représentation des images</b>	<b>13</b>
<b>II-4 Quelques définitions générales (caractéristiques de l'image :</b>	<b>14</b>
II-4-1 Le niveau de gris	14
II-4-2 Le pixel	14
II-4-3 Le bruit	15
II-4-4 L'histogramme :	15
II-4-5 Le voisinage d'un Pixel	16
<b>II-5 Le prétraitement</b>	<b>17</b>
II-5-1 1ère étape : « Le filtrage	17
II-5-2 2ème étape : « La binarisation	20
<b>II-6 Conclusion:</b>	<b>21</b>
<b>CHAPITRE III La Segmentation et la Reconnaissance d'un texte</b>	<b>22</b>
<b>III-1 Introduction</b>	<b>23</b>
<b>III-2 La segmentation</b>	<b>23</b>
III-2-1 Approche frontière	23
III-2-2 Approche région	26

<b>III-3 La segmentation par seuillage</b>	<b>26</b>
<b>III-4 La segmentation par étiquetage en composantes connexes:</b>	<b>27</b>
<b>III-5 Extraction des paramètres :</b>	<b>27</b>
III-5-1 La squelettisation	27
III-5-2 Algorithme d'amincissement	28
<b>III-6 La reconnaissance</b>	<b>28</b>
<b>III-7 Conclusion</b>	<b>28</b>

## 2eme PARTIE Les Réseaux de Neurones

<b>CHAPITRE IV Généralités sur les réseaux de neurones</b>	<b>31</b>
<b>IV-1 Introduction</b>	<b>31</b>
<b>IV-2 Le modèle biologique :</b>	<b>31</b>
IV-2-1 Définition du neurone	31
IV-2-2 Anatomie du neurone :	31
<b>IV-3 Le modèle artificiel</b>	<b>32</b>
IV-3-1 Définition	32
IV-3-2 Historique	33
IV-3-3 Les Fonctions de bases :	34
IV-3-3-1 Fonction binaire à seuil :	34
IV-3-3-2 Fonction sigmoïde	34
<b>IV-4 Quelques définitions générales</b>	<b>36</b>
IV-4-1 Réseau de neurones :	36
IV-4-2 Mémoire et oubli	36
IV-4-3 Connectivité :	36
IV-4-4 Calcul des poids synaptiques:	36
IV-4-5 Les noeuds d'entrée	36
IV-4-6 Les noeuds cachés	37
IV-4-7 Les noeuds de sortie	37
<b>IV-5 Classifications des réseaux de neurones</b>	<b>37</b>
IV-5-1 Les réseaux de neurones non bouclés	38
IV-5-2 Les réseaux de neurones bouclés	38
<b>IV-6 L'apprentissage des réseaux de neurones</b>	<b>38</b>
IV-6-1 L'apprentissage supervisé	39
IV-6-2 L'apprentissage non supervisé	39
<b>IV-7 Les règles d'apprentissage</b>	<b>39</b>
IV-7-1 Règle d'apprentissage supervisée	39
IV-7-2 Règle d'apprentissage pour réseaux non supervisé	39
<b>IV-8 Différents type de réseaux de neurones</b>	<b>40</b>
IV-8-1 Les perceptrons	40
IV-8-2 Les perceptrons multicouches	41
IV-8-3 Réseaux de Hopfield	42
IV-8-4 Réseaux de Kohonen	42
<b>IV-9 Conclusion</b>	<b>43</b>
<b>CHAPITRE V Les Réseaux Multicouches</b>	<b>45</b>
<b>V-1 Introduction</b>	<b>45</b>

<b>V-2 Structure et fonctionnement</b>	<b>45</b>
<b>V-3 Apprentissage</b>	<b>46</b>
<b>V-4 Taille du réseau et de la base d'apprentissage</b>	<b>47</b>
<b>V-5 Les étapes de la conception d'un réseau</b>	<b>47</b>
V-5-1 Choix des échantillons	47
V-5-2 Elaboration de la structure du réseau	47
V-5-3 Apprentissage	48
V-5-4 Validation	48
<b>V-6 Conclusion</b>	<b>48</b>
<b>CHAPITRE VI Exemples d'application des réseaux de neurones</b>	<b>50</b>
<b>VI-1 Introduction</b>	<b>50</b>
<b>VI-2 Avantages et désavantages des réseaux de neurones</b>	<b>50</b>
<b>VI-3 Application des réseaux de neurones</b>	<b>51</b>
<b>VI-4 Conclusion</b>	<b>52</b>
<b>3eme PARTIE Approche Adoptée</b>	
<b>CHAPITRE VII Travail Effectué</b>	<b>55</b>
<b>VII-1 Introduction</b>	<b>55</b>
<b>VII-2 L'acquisition</b>	<b>55</b>
<b>VII-3 Module de prétraitement</b>	<b>56</b>
<b>VII-4 La segmentation d'un texte</b>	<b>59</b>
VII-4-1 La segmentation horizontale	59
VII-4-2 La segmentation verticale	61
VII-4-3 La segmentation par étiquetage en composantes connexes	63
VII-4-5 Segmentation en caractère ( Extraction des caractères)	68
<b>VII-5 Reconnaissance</b>	<b>70</b>
<b>VII-6 Reconnaissance par réseaux de neurones</b>	<b>70</b>
VII-6-1 Architecture du réseau de neurones	78
VII-6-2 Apprentissage du réseau	81
<b>CHAPITRE VIII Description du logiciel</b>	<b>83</b>
<b>VIII-1 Introduction</b>	<b>83</b>
<b>VIII-2 Configuration de l'application</b>	<b>84</b>
VIII-2-1 La barre des menus	84
VIII-2-1-1 Menu Fichier	85
VIII-2-1-2 Menu Fenêtre	86
VIII-2-1-3 Menu Prétraitement	86
VIII-2-1-4 Menu Filtrage	87
VIII-2-1-5 Menu Histogramme	87
VIII-2-1-6 Menu segmentation	88
VIII-2-1-7 Menu Reconnaissance	89
VIII-2-2 Fenêtre enfant	<b>90</b>



VIII-2-3 La barre de bouton	91
VIII-2-4 La barre d'état :	92
<b>VIII-3 Résultats</b>	<b>93</b>
<b>CONCLUSION GENERALE</b>	<b>97</b>
<b>ANNEXES</b>	<b>98</b>
Annexe 1 Algorithme de rétro propagation du gradient	99
Annexe 2 Structure d'un fichier bitmap	100
Annexe 3 Glossaire	102
<b>BIBLIOGRAPHIE</b>	<b>103</b>

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# LISTE DES FIGURES

**1 ere PARTIE**

**CHAPITRE II Acquisition et Prétraitement**

Figure II - 1 : Dispositif d'acquisition : Scanner	11
Figure II - 2 : Dispositif d'acquisition : Caméra	13
Figure II - 3 : Représentation des attributs d'une image	13
Figure II - 4 : Comparaison entre image de différentes couleurs	13
Figure II - 5 : représentation d'une image en niveaux de gris	14
Figure II - 6 : Représentation schématique du pixel	14
Figure II - 7 : Image en niveaux de gris : (a) bruitée , (b) filtrée .	15
Figure II - 8 : Histogramme en niveaux de gris	16
Figure II - 9 : voisinage du pixel : (a) avec 4 éléments connexes, (b) avec 8 éléments connexes	16

**CHAPITRE III La Segmentation et la Reconnaissance d'un texte** **31**

Figure III - 1 : Dérivation en présence d'un contour	25
(a) contour, (b) Gradient, (c) Laplacien.	25
Figure III - 2 : Segmentation par étiquetage en composantes connexes.	27

**2 eme PARTIE**

**CHAPITRE IV Généralités sur les Réseaux de Neurones** **39**

Figure IV - 1: neurone biologique	32
Figure IV - 2 : représentation d'un neurone artificiel	33
Figure IV - 3 : Dualité entre un neurone artificiel et biologique	33
Figure IV - 4 : Fonction binaire à seuil	34
Figure IV - 5 : fonction exponentielle	35
Figure IV - 6 : fonction tangentielle	35
Figure IV - 7 : Structure d'un réseau de neurones	37
Figure IV - 8 : Réseau de neurone bouclé	38
Figure IV-9 : Réseau à une couche	41
Figure IV-10 : Structure d'un réseau multicouche	41
Figure IV-11 : Carte Kohonen	42

**CHAPITRE V Les Réseaux Multicouches** **52**

Figure V-1 : Représentation générale d'un réseau multicouche	45
Figure V-2 Réseau de neurones à 3 couches	46

**3 eme PARTIE**

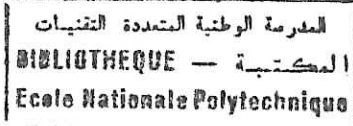
**CHAPITRE VII Travail Effectué** **62**

Figure VII - 1 a\ Portion agrandie d'une image bruitée.	57
b\ niveau de gris des pixels voisins.	57
Figure VII - 2 : Image d'un document scanné en couleur	57
Figure VII - 3: Image d'un document convertit en niveaux de gris	57
Figure VII - 4 : Passage d'un document en niveaux de gris à un document binarisé	58
Figure VII - 5 : Segmentation horizontale	61
Figure VII - 6 : Segmentation Verticale	62
Figure VII - 7 : segmentation horizontale-verticale(en mots)	62
Figure VII - 8 : Ligne médiane de chaque ligne de texte	63
Figure : VII - 9 exemple d'une image à segmenter	66
Figure : VII - 10 Image de départ initialisée	66
Figure : VII - 11 Image des étiquettes après le premier passage	67
Figure : VII - 12 Table d'équivalence correspondante	67
Figure : VII - 13 Table d'équivalence après résolution des équivalences	67
Figure : VII - 14 Image des étiquettes finales	67
Figure : VII - 15 segmentation par étiquetage en parties connexes	67
Figure : VII - 16 Ligne médiane de chaque partie connexe	68

Figure : VII - 17 Segmentation en caractère	70
Figure : VII - 18 Cadrage en caractère	70
Figure-VII-19: encadrement du caractère .	72
FigureVII-20 : longueur et largeur d'une étiquette	74
Figure VII-21 : Différentes positions possibles	74
FigureVII-22 : Mise en évidence de la ponctuation	75
Figure VII-23 : Différentes possibilités pour ndp	75
FigureVII-24 taux de remplissage	77
Figure VII-25 codes ASCII et de sortie de chaque caractère	78
Figure VII-26 représentation du réseau utilisé	79
Figure VII-27 étapes d'apprentissage	80
<b>CHAPITRE VIII Description du Logiciel</b>	<b>90</b>
Figure VIII-1 L'interface du logiciel	84
Figure VIII-2 Boite de dialogue pour ouvrir	85
Figure VIII-3 boite de dialogue pour la binarisation	86
Figure VIII-4 Représentation de l'histogramme horizontal	88
Figure VIII-5 Exemple de fenêtre enfant	90
Figure VIII-6 La barre de bouton	91
Figure VIII-7 la barre d'état	92
Figure VIII-8 représentation de l'étape de prétraitement	92
Figure VIII-9 Segmentation horizontale et verticale	93
Figure VIII-10 Segmentation en partie connexe et en caractère	94
Figure VIII-11 reconnaissance du texte	95

المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

# Introduction Générale



Les techniques liées au traitement de l'information connaissent actuellement un développement très actif en liaison avec l'informatique, et présentent un potentiel important dans le domaine de l'interaction Homme – Machine.

Les systèmes informatiques ne se contentent plus d'aider l'homme dans les tâches qualifiées, répétitives et fastidieuses, mais tentent plutôt de communiquer avec lui et d'acquérir les mêmes possibilités de perception.

Beaucoup pensent à la substitution de l'homme par la machine pour des situations complexes ou à automatiser certaines activités ; pour ce faire, il faudrait munir la machine d'un système de perception visuelle qui serait à la base de chaque traitement et analyse d'image.

La reconnaissance des formes s'intéresse à la conception et à la réalisation de ces systèmes de perception et d'interprétation des signaux captés dans le monde extérieur.

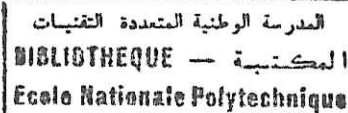
La reconnaissance de l'écriture s'inscrit dans le cadre de la reconnaissance des formes.

Un système de reconnaissance de caractères appelé OCR ( Optical Character Recognition) consiste à récupérer automatiquement des textes à partir d'un document numérisé, en évitant la phase fastidieuse de la saisie.

De nos jours, des systèmes très performants arrivent à des taux de reconnaissance très élevés pour l'écriture imprimée, alors que le manuscrit pose encore quelques problèmes au niveau de la cursivité des lettres et leur variabilité d'une personne à une autre.

Ce présent travail traite l'écriture arabe manuscrite. Ce thème a bénéficié de quelques recherches, car il reste un sujet important et utile mais assez complexe.

Pour certains, la reconnaissance de l'écriture représente la reconnaissance de la structure du document, c'est-à-dire de séparer les parties textuelles des images .... , mais dans notre cas, nous considérons le caractère comme base de travail, et pour arriver à ce stade, il a fallu séparer les lignes entre elles, les mots et finalement les caractères.



La procédure de reconnaissance adoptée pour notre travail suit les étapes suivantes :

- ✱ L'acquisition des données constituant un paragraphe de texte manuscrit
- ✱ Le filtrage et la segmentation par étiquetage en parties connexes
- ✱ La reconnaissance en utilisant les réseaux de neurones

Une application a été mise au point pour illustrer les étapes précédentes.

Ce mémoire est structuré en trois parties essentielles :

- ✱ Partie 1 : généralités sur l'OCR
- ✱ Partie 2 : les réseaux de neurones
- ✱ Partie 3 : le travail effectué

Dans la première partie, nous donnerons des généralités sur le traitement d'image ; la deuxième partie concerne les réseaux de neurones et leurs applications ; quant à la troisième partie, elle applique les notions précédentes à notre travail avec la description du logiciel et les résultats obtenus.

Nous concluons par une synthèse du travail et quelques perspectives.

# 1ere PARTIE

## GENERALITES SUR L'OCR

### CHAPITRE I

#### Description d'un système de reconnaissance de caractères

- 1) Introduction
- 2) Définition de chaque étape
- 3) Application de l'OCR
- 4) Implémentation d'un système OCR
- 5) Conclusion

### CHAPITRE II

#### Acquisition et Prétraitement

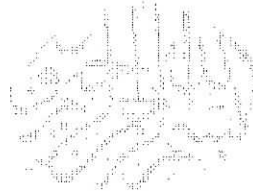
- 1) Introduction
- 2) Techniques d'acquisition
- 3) Représentation des images
- 4) Quelques définitions générales
- 5) Le prétraitement
- 6) Conclusion

### CHAPITRE III

#### La segmentation et la reconnaissance d'un texte

- 1) Introduction
- 2) Segmentation
- 3) Segmentation par seuillage
- 4) Segmentation par étiquetage en composantes connexes
- 5) Extraction des paramètres
- 6) Reconnaissance
- 7) Conclusion





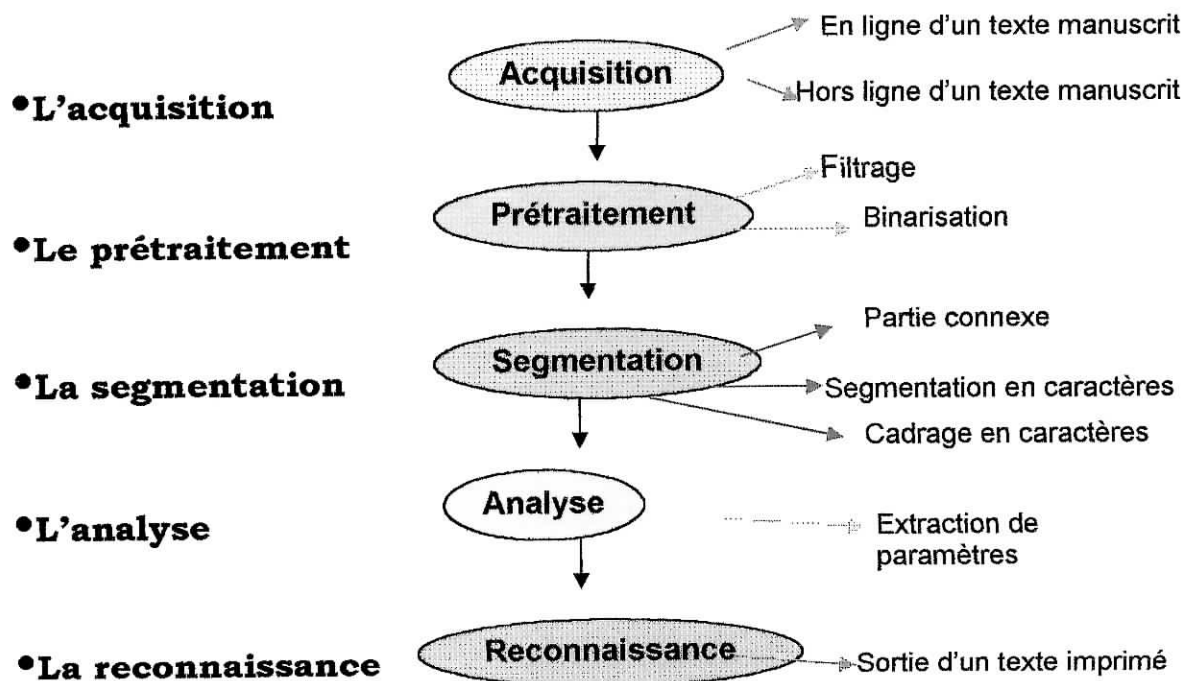
# Chapitre I

## Description d'un système de reconnaissance de caractère

## I-1 Introduction :

Le traitement d'image est né de l'idée et de la nécessité de remplacer l'observateur humain par la machine, car une image n'est que la représentation d'une scène ou d'un objet ; c'est une collection d'informations pour l'utilisateur qui pourra en extraire certains paramètres suivant que cette image soit un document officiel ou un autre type d'image (médicale, satellitaire....)

Nous pouvons représenter schématiquement les différentes étapes (ou modules) d'un système OCR comme suit :



## **I-2 Définition de chaque étape : [10]**

### **I-2-1 L'acquisition :**

L'acquisition représente la première étape pour le traitement d'une image. Elle consiste à transformer une image en matrice binaire. Ce qui revient à faire un passage d'un monde analogique vers un monde numérique autrement dit un *codage*.

### **I-2-2 Le prétraitement :**

Cette étape prépare les données à la phase suivante du traitement. Elle consiste à :

- Minimiser le bruit (opération de filtrage)
- Binariser l'image ( seuillage)
- Corriger les erreurs éventuelles.

### **I-2-3 La segmentation :**

Elle désigne la séparation de l'information en ses éléments constitutifs afin de les identifier isolément. Dans le cas de la reconnaissance des mots, les éléments constitutifs sont naturellement les lettres du mot.

### **I-2-4 L'analyse :**

Dont la définition littérale est « la décomposition d'un tout en ses parties », consiste généralement en l'extraction d'un ensemble d'attributs caractéristiques du texte.

Les éléments obtenus serviront à la reconnaissance.

### **I-2-5 La reconnaissance :**

A ce stade du traitement, notre texte manuscrit va être transformé (traduit) en une représentation compréhensible par une machine, essentiellement par un traitement de texte.

On classe souvent les méthodes de reconnaissance suivant le mode d'acquisition du texte.

#### I-2-5-1 L'écriture en ligne (dynamique)

Elle est obtenue par une saisie continue et se présente sous la forme d'une séquence de points ordonnée dans le temps.

## **I-5 Conclusion :**

Dans ce chapitre, nous avons défini ce qu'est un système de reconnaissance de l'écriture, en définissant les différentes étapes qui le constituent.

# Chapitre II

## Acquisition et Prétraitement

## **II-1 Introduction :**

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image, il est nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système.

## **II-2 Techniques d'acquisition : [7]**

L'acquisition n'est qu'une numérisation de l'image, et selon le capteur utilisé on obtient :

- Une image matricielle binaire donc à deux niveaux de gris dans le cas de l'utilisation d'un appareil à diode photosensible.
- Une image matricielle à plusieurs niveaux de gris si on utilise un scanner ou une caméra.

Parmi les moyens les plus utilisés pour l'acquisition des images :

- Hors ligne ⇒ les scanners  
Les caméras
- En ligne ⇒ les tablettes graphiques

### **II-2-1 Le scanner :**

C'est un des périphériques de l'ordinateur utilisé pour la capture, la numérisation de l'image, pour son traitement, la retoucher ou juste la sauvegarder. Son principe de fonctionnement est basé sur la réflexion d'un faisceau lumineux sur la page qu'on souhaite numériser grâce à un ensemble de capteurs CCD composant un couplage de charge.

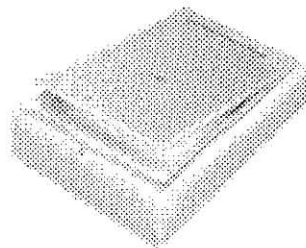


Figure II – 1 : Dispositif d'acquisition : Scanner

a) Le scanner à main :

Un scanner à main est utilisé pour numériser de petites surfaces ; il n'est pas adapté pour les grandes feuilles, et comme son nom l'indique, le déplacement le long du document se fait manuellement.

b) Le scanner à plat :

Ce type de scanner se charge de toute l'opération de numérisation ; il a un second nom qui est scanner A4 relatif au format scanner.

c) Le scanner à défilement :

Ce modèle n'utilise que des feuilles volantes, qu'il attire vers l'intérieur pour les numériser ligne par ligne.

### **II-2-2 La caméra :**

Pour l'acquisition hors ligne, on dispose d'un second appareil qui est la caméra. Cette dernière représente l'œil de l'ordinateur. Il existe trois types :

a) Les caméras à tube :

Son élément principal est une cible photoconductrice explorée par un faisceau lumineux par balayage ligne par ligne.

b) Les caméras CCD : (Changed Coupled Device)

C'est un assemblage de photodiodes qui délivrent une intensité proportionnelle à un point de l'image appelé pixel. La résolution d'une caméra est le couple constitué par le nombre de pixels disponibles en x et y.

c) Les caméras numériques :

Une image numérique n'est qu'une matrice bidimensionnelle de valeurs numériques.

Sa qualité dépend souvent de l'appareil utilisé.

Elles sont constituées d'un viseur à cristaux liquides pour bien visualiser ce qu'on est sur le point de photographier et d'effectuer les réglages nécessaires.

Ce viseur permet de consulter l'image, de l'effacer ou de l'archiver.

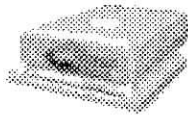


Figure II - 2 : Dispositif d'acquisition : Caméra

### **II-2-3 La tablette graphique :**

L'image obtenue n'est qu'un ensemble de points séparés ; on utilise un crayon tactile qui lorsqu'il y a contact avec la surface de la tablette, il émet les coordonnées des points qu'il décrit sachant que la rupture interrompt la transmission d'un caractère donné.

Donc, on obtient un ensemble de coordonnées séparées par des signes indiquant les moments où le crayon a cessé de toucher la tablette.

### **II-3 Représentation des images :**

Pour une représentation convenable, on peut stocker l'image dans un fichier. Ce dernier se compose de deux parties : l'entête et l'image elle-même.

Dans l'entête, on retrouve les informations générales de l'image (hauteur, largeur, nombre de couleurs utilisées .....).

Il existe un nombre important de formats de fichier images ; le plus utilisé est le bitmap ; il prend beaucoup de place mais est plus simple à utiliser par rapport au format JPEG qui est compressé.

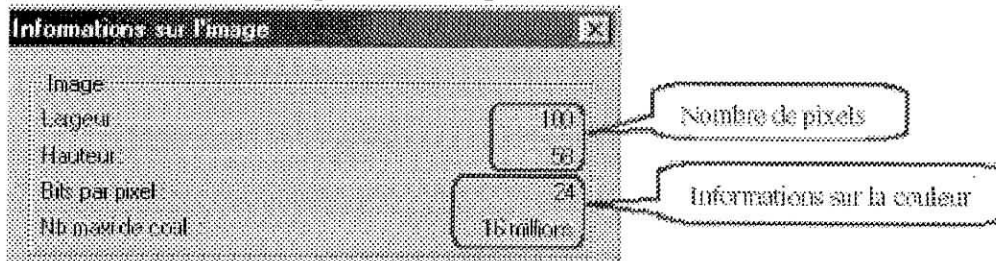


Figure II - 3 : Représentation des attributs d'une image

Dimensions de l'image	Image en noir et blanc	Image en 256 niveaux de gris	Image en true color
320x200	7.8Ko	62.5Ko	197.5 Ko
640x480	37.5Ko	300Ko	900Ko
800x600	50.6Ko	469.7Ko	1.4Mo
1024*768	96Ko	759Ko	2.3Mo

Figure II - 4 : Comparaison entre image de différentes couleurs



## **II-4 Quelques définitions générales (caractéristiques de l'image : [17]**

### **II-4-1 Le niveau de gris :**

C'est la valeur de l'intensité lumineuse en un point pour représenter une image en niveaux de gris. On donne à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée.

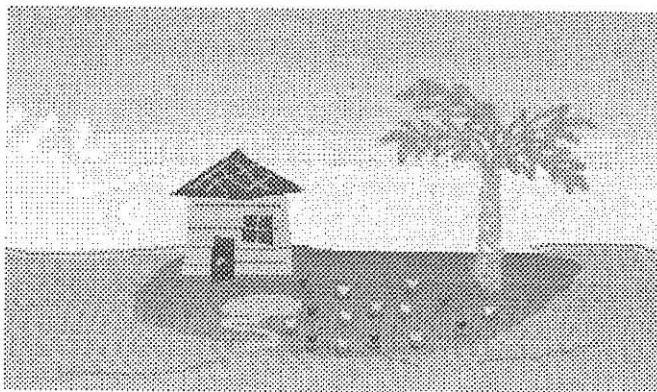


Figure II - 5 : représentation d'une image en niveaux de gris

### **II-4-2 Le pixel :**

C'est la plus petite valeur (composante) d'une image. Il peut être noir, blanc ou d'une autre couleur appartenant aux niveaux intermédiaires. En général, le noir est noté 0 et le blanc 0X00FFFFFF.

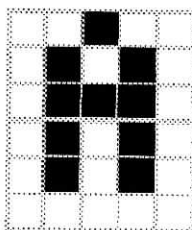
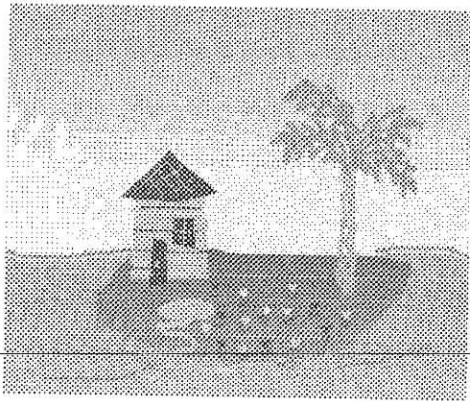


Figure II - 6 : Représentation schématique du pixel

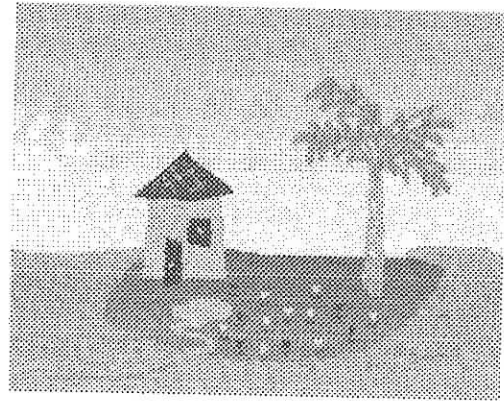
**II-4-3 Le bruit :**

Dans une image numérisée, le bruit est la brusque variation du niveau de gris d'un pixel isolé par rapport à ses voisins. On le considère aussi comme étant une fonction aléatoire qui se superpose additivement à l'image idéale.

Donc lorsque la qualité d'une image donnée est insuffisante pour qu'un algorithme de décomposition puisse s'appliquer de manière stable, on a recours à des méthodes d'atténuation du bruit dites de filtrage.



(a)



(b)

Figure II - 7 : Image en niveaux de gris : (a) bruitée , (b) filtrée .

**II-4-4 L'histogramme en niveaux de gris :**

L'histogramme est une fonction qui donne la fréquence d'apparition de chaque niveau de gris dans l'image. On aura en abscisse le niveau de gris allant de 0 à  $N=255$ , et en ordonnée, sera représenté le nombre de pixels affectés à chaque niveau de gris.

Cette fonction fournit un grand nombre d'informations sur la distribution des niveaux de gris qui va permettre de distinguer une image claire d'une autre foncée.

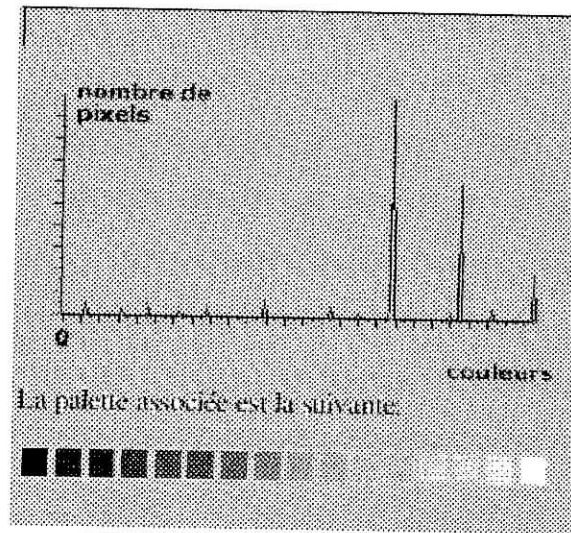


Figure II - 8 : Histogramme en niveaux de gris

#### **II-4-5 Le voisinage d'un Pixel :**

Le voisinage d'un pixel est composé de tous les pixels qui l'entourent immédiatement. Si  $p$  est un pixel d'une image  $D$ , alors le voisinage de  $p$  est le plus petit sous-ensemble de  $D$  qui contient  $p$ .

Dans une image numérique, on distingue deux types de connexités relatives au voisinage utilisé : la 4-connexité et la 8-connexité.

Le voisinage d'un pixel  $(i,j)$  est dit 4-connexe s'il est formé des quatre pixels de coordonnées spatiales  $(i+1,j)$ ,  $(i,j+1)$ ,  $(i-1,j)$ ,  $(i,j-1)$ , et il est dit 8-connexe s'il est formé des pixels de coordonnées spatiales  $(i+1,j)$ ,  $(i,j+1)$ ,  $(i-1,j)$ ,  $(i,j-1)$ ,  $(i+1,j-1)$ ,  $(i+1,j+1)$ ,  $(i-1,j+1)$ ,  $(i-1,j-1)$ .

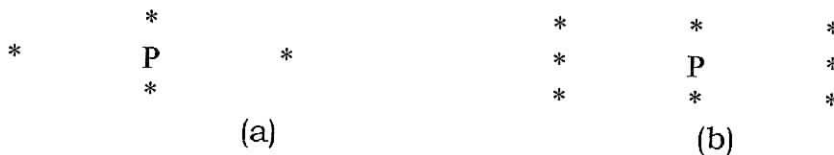


Figure II - 9 : voisinage du pixel : (a) avec 4 éléments connexes, (b) avec 8 éléments connexes

## II-5 Le prétraitement :

A ce stade du traitement, on a en notre possession, une image numérisée mais pas nécessairement très nette (parfaite) ; c'est pour cela qu'un prétraitement est nécessaire.

### II-5-1 1ère étape : « Le filtrage »:

L'action de « filtrer une image » signifie éliminer les composantes gênantes de cette image, et ceci, en lui appliquant une transformation mathématique qui va modifier la valeur du niveau de gris de la partie traitée (endommagée).

En général, les pixels voisins ont les mêmes caractéristiques physiques, donc une variation de l'un d'entre eux est remarquable mais surtout doit être éliminée par les traitements ultérieurs.

Il existe plusieurs types de filtres :

- Les filtres linéaires
- Les filtres non linéaires

#### a) Les filtres linéaires

Ce type de filtrage est caractérisé par une opération de convolution. De sorte que la fonction va prendre en considération les valeurs de gris au voisinage de chaque pixel et remplacer sa valeur par une combinaison linéaire de ses voisins.

$$B(i, j) = \sum_{k=-w}^{+w} \sum_{l=-v}^{+v} ch(k, l)A(i+k, j+l)$$

$B(i, j)$  = c'est la matrice filtrée au pixel  $(i, j)$  . C'est la combinaison de tous les pixels autour du point  $(i, j)$  .

$h(k, l)$  = ces valeurs donnent les poids = masque du filtre.

$A(k, l)$  = la valeur initiale du pixel  $(k, l)$  .

$w, v$  = les dimensions de la fenêtre en largeur et hauteur.

Parmi les filtres linéaires les plus connus, on a :

a-1) Le filtre moyen :

Ce filtre considère chaque pixel de l'image et fait la moyenne avec les huit pixels voisins dans le cas d'un masque 3x3 .On obtient une image adoucie en réduisant les brusques variations de niveaux de gris.

On peut utiliser le masque suivant :

$$H = \frac{1}{N} \begin{vmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{vmatrix}$$

tel que N= coefficient de normalisation  
= Σcoefficients non nuls

Chaque pixel est multiplié par le coefficient correspondant. les masques les plus courants sont :

$$H_1 = \frac{1}{9} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

$$H_2 = \frac{1}{10} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

$$H_3 = \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$$

On déplace le masque sur toute l'image et le pixel affecté par la transformation est le pixel central du masque.

*a-1-1 Filtre moyen square :*

$$H_1 = \frac{1}{9} \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

ce masque est régit par l'équation suivante :

$$I_F(x, y) = \sum_{i=-l}^{i=l} \sum_{j=-l}^{j=l} H[i+l, j+l] I_1(x+i, y+i)$$

*a-1-2 Filtre moyen cross :*

Les coefficients non nuls de ce masque forment un plus “ +”

$$H_2 = \frac{1}{5} \begin{vmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

a-1-3 Filtre moyen diagonal :

$$H_3 = \frac{1}{5} \begin{vmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix}$$

Ce masque a pour particularité que les diagonales sont non nulles (égales à 1) et le reste est nul.

a-2) Le filtre de Gauss :

Pour appliquer ce filtre, il suffit de convoluer l'image initiale à une gaussienne  $G(x, y, \sigma)$  à deux dimensions donnée par la formule suivante :

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Ce filtre effectue la moyenne des pixels voisins avec une pondération par des coefficients discrétisés d'une gaussienne.

b) Les filtres non linéaires :

Parmi les filtres non linéaires les plus connus nous citons le filtre médian ; son principe consiste à calculer en un point la valeur médiane dépendante du tri des niveaux de gris des points voisins.

Principe :

- On classe les pixels voisins du pixel concerné par la transformation, par valeurs croissantes.
- On prend la valeur médiane des pixels classés.
- On affecte cette valeur au pixel concerné.

Il existe d'autres types de filtres :

c) Les filtres morphologiques :

Utilisés souvent pour éliminer des pixels isolés dans une image binarisée (numérisée) qui sont considérés comme du bruit.

Parmi ces opérations morphologiques, nous avons :

c-1 La dilatation :

En appliquant une dilatation, les tâches blanches peuvent être éliminées en utilisant un «ET » logique des huit pixels voisins mais la taille de l'objet change.

c-2 L'érosion :

Cet opérateur permet d'éliminer les pixels noirs isolés grâce à un « OU » logique fait entre les huit pixels voisins. La taille de l'objet est aussi modifiée.

c-3 L'Ouverture :

L'ouverture est constituée par une opération d'érosion suivie d'une dilatation. Cet opérateur restitue la taille de l'objet.

c-4 La Fermeture :

La fermeture est une opération morphologique qui consiste à faire subir à l'image une opération de dilatation suivie d'une érosion. De même que pour l'ouverture, la fermeture restitue la taille de l'objet.

**II-5-2 2ème étape : « La binarisation »: [18]**

La binarisation fait partie des traitements ponctuels parmi les valeurs possibles attribuées aux pixels d'une image. On veut parfois sélectionner une plage plus restreinte, de sorte que toutes les intensités au-dessus d'un seuil  $\theta$  parmi les niveaux 0 à 255 soient gardées par exemple.

L'image à deux niveaux ainsi obtenue représente la zone sélectionnée pour le traitement.

Le choix du seuil doit être judicieux pour éviter au mieux les effets parasites de variations d'intensité ; à cet effet, l'histogramme des valeurs de pixels peut être utile.

Cette étape appelée aussi seuillage, nous permet d'isoler les objets en noir sur un fond blanc.

Donc on va se retrouver avec deux classes :

L'objet et le fond, mais le seuil reste toujours difficile à choisir, on peut faire la moyenne ou calculer la valeur médiane des pixels de l'image, mais la meilleure solution est l'utilisation de l'histogramme et prendre le seuil comme étant le point avec la plus petite valeur entre les deux pics.

L'algorithme de calcul du seuil d'une image à l'aide de l'histogramme est :

- 1) Calcul de l'histogramme  $H[i]$   $i = 0$  à  $255$  tel que :  
 $i$  = niveaux de gris  
 $H[i]$  = nombre de pixels ayant le niveau de  $i$
- 2) Calcul de la moyenne de l'image : c'est le seuil initial

$$T = \frac{1}{wx * hy} \sum_{i=0}^{wx-1} \sum_{j=0}^{hy-1} I(i, j)$$

$wx$  = nombre de lignes  
 $hy$  = nombre de colonnes

$I(i, j)$  = intensité de niveaux de gris du pixel  $(i, j)$

- 3) Calcul du  $T_{\text{objet}} = \frac{t_1}{2t_2}$

$$t_1 = \sum_{i=0}^T i * H[i]$$

$$t_2 = \sum_{i=0}^T H[i]$$

- 4) Calcul de  $T_{\text{fond}} = \frac{t'_1}{2t'_2}$

$$t'_1 = \sum_{i=T+1}^{255} i * H[i]$$

$$t'_2 = \sum_{i=T+1}^{255} H[i]$$

- 5) Si le seuil  $T' = T_{\text{objet}} + T_{\text{fond}}$ , alors  $T'$  est le seuil idéal  
 Sinon reprendre l'itération à partir de la troisième étape.

## II-6 Conclusion:

Dans ce chapitre nous avons défini une étape primordiale pour le traitement d'image en général, le *prétraitement*.

Nous avons présenté les différents filtres, l'algorithme de binarisation pour éliminer toutes les imperfections qui peuvent exister sur notre image.



## Chapitre III

# La Segmentation et la Reconnaissance d'un texte

### **III-1 Introduction :**

Pour pouvoir analyser une image, on doit la partager en un certain nombre de parties connexes ; cette étape est très importante en traitement numérique des images.

Lorsque deux points appartiennent à la même partie, c'est qu'ils possèdent une même propriété et qu'on peut les joindre par une chaîne continue de points voisins ayant cette même propriété.

### **III-2 La segmentation : [7]**

C'est l'étape médiane entre le prétraitement et la reconnaissance. Elle est définie par le voisinage et les caractéristiques de la partie traitée. Son but est l'extraction d'attributs caractérisant les objets faisant partie de l'image.

Ces attributs correspondent à des points d'intérêt (contours) ou des zones de l'image (régions).

On peut l'étudier suivant deux approches : frontières et régions

Approche frontière : la forme d'un objet dépend de son contour donc on s'intéresse essentiellement aux pixels de cette frontière.

Approche région : la région est la surface de l'objet. A cette étape, on applique la segmentation exclusivement sur les pixels de cette surface.

#### **III-2-1 Approche frontière :**

La détection du contour joue un rôle primordial dans les systèmes de reconnaissance.

Un contour est défini par un point frontalier de l'image ; c'est le lieu où l'on observe une discontinuité de la fonction des niveaux de gris.

Pour détecter le contour d'une image, il suffit de détecter les discontinuités, et pour cela, nous avons deux approches à notre disposition : le Gradient et le Laplacien.

Ces approches se basent sur l'effet de la dérivation en présence d'un contour.

##### a) Approche Gradient :

Le contour possède le maximum de points à amplitudes élevées du gradient. Mathématiquement, le gradient est défini comme suit :

$$\nabla f(x, y) = \left( \frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right) = |\nabla f(x, y)| \angle \arg(\nabla f(x, y))$$

$$\nabla f(x,y) = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

$$\arg(\nabla f(x,y)) = \arctg\left(\frac{\partial f(x,y)/\partial x}{\partial f(x,y)/\partial y}\right)$$

Dans le cas de notre image, il faut calculer le gradient de chaque pixel. Pour cela, une approximation du gradient est plus appropriée. Cette dernière n'est que le résultat de convolution d'une fenêtre de l'image avec les masques tel que :

Si I est l'image :

$$\text{Le gradient en } x \Rightarrow \frac{\nabla I(x,y)}{\partial x} = M_1 ** I(x,y)$$

$$\text{Le gradient en } y \Rightarrow \frac{\nabla I(x,y)}{\partial y} = M_2 ** I(x,y)$$

$M_1$  et  $M_2$  sont les masques utilisés ; les plus connus sont :

- Le masque de Perwitt :

$$M_1 = \frac{1}{6} \begin{vmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{vmatrix} \quad M_2 = \frac{1}{6} \begin{vmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & 1 \end{vmatrix}$$

- Le masque de Sobel :

$$M_1 = \frac{1}{6} \begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix} \quad M_2 = \frac{1}{6} \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}$$

Après le calcul du gradient en x et en y, il faut déterminer son module pour chaque point de l'image. Ensuite, il faut détecter les maxima locaux du gradient pour déterminer les points du contour.

Pour déterminer un maxima local, il faut calculer la norme de tous les pixels et faire leurs comparaisons. Pour des points appartenant à la même direction de gradient, la norme supérieure à celle des points adjacents représente le maximum local.

#### b) Approche Laplacien :

Pour cette approche, les points de contour correspondent aux valeurs nulles du Laplacien de l'image.

Le Laplacien est défini comme étant la dérivée seconde de notre signal.

$$L(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

On l'obtient par calcul du produit de convolution bidimensionnel de l'image avec un masque.

Pour déterminer les points de contour, il suffit de détecter les passages par zéro du Laplacien de l'image.

Pour cette méthode, des procédures de seuillage sont nécessaires car le Laplacien est très sensible au bruit.

Exemple de masque utilisé :

$$H_1 = \begin{vmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

$$H_2 = \begin{vmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & 1 \end{vmatrix}$$

$$H_3 = \begin{vmatrix} -1 & -2 & 1 \\ -2 & 1 & -2 \\ 1 & -2 & 1 \end{vmatrix}$$

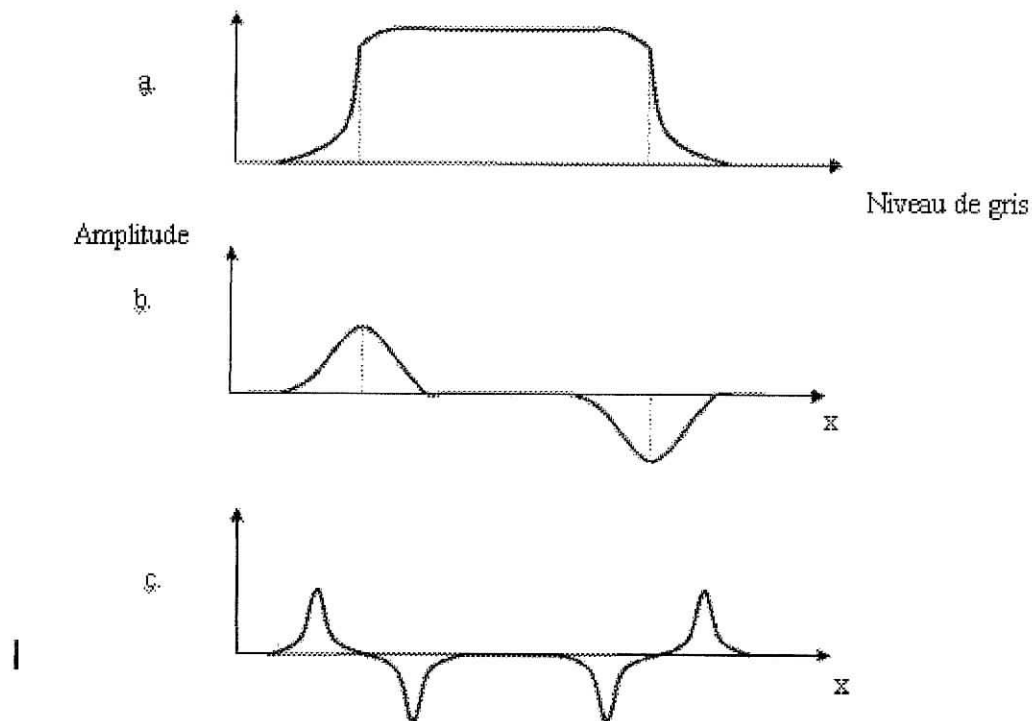


Figure III - 1 : Dérivation en présence d'un contour

(a) contour, (b) Gradient, (c) Laplacien.

### **III-2-2 Approche région :**

La segmentation en région vise à segmenter l'image I en se basant sur les propriétés de la région pour déterminer le critère de segmentation.

➤ Segmentation en utilisant les critères d'homogénéité :

L'homogénéité d'une région de l'image est le point déterminant des performances de la segmentation.

On la caractérise par le niveau de gris ou la couleur et la texture de l'image.

Il existe trois méthodes de segmentation en région :

a) Méthode par séparation :

- 1) ( test du critère d'homogénéité = = vrai)
- 2) stop ; arrêt de l'algorithme  
sinon :
- 3) décomposition de l'image en zones plus petites, et on recommence les étapes précédentes.

b) Méthode par fusion :

A partir d'une partition de l'image (parfois composée d'un seul pixel) , on fusionne les régions adjacentes si elles vérifient le critère d'homogénéité. Si le critère n'est plus vérifié, on s'arrête.

c) Méthode mixte :

Segmentation par division et rassemblement. C'est une combinaison des méthodes précédentes de sorte que si la région n'est pas homogène, elle est divisée et si elle est similaire, elle est fusionnée.

### **III-3 La segmentation par seuillage : [5]**

On associe pour chaque pixel de l'image la région à laquelle il appartient ; pour cela, il faut faire la comparaison de ses attributs à un ou plusieurs seuils. Cette approche est utilisée pour segmenter une image composée de plusieurs régions possédant des attributs différents.

La détermination du seuil se fait par calcul de l'histogramme de l'image.

### **III-4 La segmentation par étiquetage en composantes**

**connexes: [5]**

L'étiquetage en composantes connexes d'une image attribue à tous les pixels d'une même partie, la même étiquette ; mais si le voisin n'est pas étiqueté, on prend une nouvelle étiquette.

Donc la segmentation par étiquetage dépend de la position et de l'état des points voisins.

Le processus se répète jusqu'à l'obtention d'une image complètement étiquetée.

Cette méthode permet d'attribuer une étiquette à chaque élément similaire d'une image en niveaux de gris.

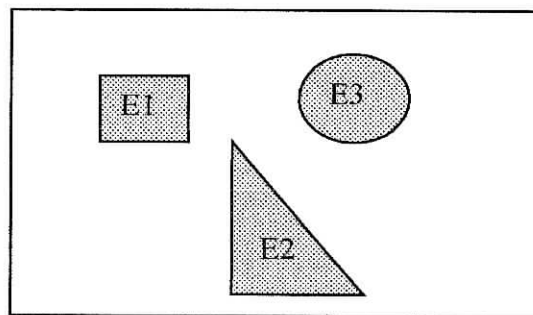


Figure III – 2 : Segmentation par étiquetage en composantes connexes.

### **III-5 Extraction des paramètres : [18]**

Une fois la segmentation terminée, on passe à l'extraction des paramètres. De notre image, on va extraire les informations pour décrire l'objet et / ou sa forme.

#### **III-5-1 La squelettisation :**

Cette opération est une étape importante de la reconnaissance des formes. Elle réduit le volume d'informations en représentant chaque objet par son ossature.

L'ensemble des points connexes représentent le squelette.

Il existe deux approches principales :

- Calcul de la distance à la frontière :  
On calcule les maximas locaux, tel que le squelette est l'ensemble des points.
- Amincissement : on épluche notre objet couche par couche de manière itérative tel que le squelette est le cœur de l'objet.

### **III-5-2 Algorithme d'amincissement :**

- 1) Binariser l'image
- 2) Epluchage : le pixel du fond est blanc (0)  
Le pixel du motif est noir (1)

On applique un balayage sur l'image avec les 16 masques 3x3 .  
Si l'une des configurations est vérifiée, le pixel central est mis à zéro (0).  
L'opération est réitérée jusqu'à stabilité.

- 3) Nettoyage : on utilise les masques de nettoyage si les frontières de l'objet sont bruitées. Dans ce cas là, le squelette comporte beaucoup d'embranchement.

### **III-6 La reconnaissance : [6]**

C'est la dernière étape d'un processus de traitement de l'image.  
Elle a pour but de classer l'objet traité en se référant à une base de données mémorisée au préalable.

Cette classification est considérée comme une identification partielle de l'information ; on peut déterminer une catégorie à laquelle elle appartient.  
La superposition avec l'un de ces modèles peut se faire de deux manières :

- Superposition de forme
- Détection de similitudes.

### **III-7 Conclusion :**

La segmentation et la reconnaissance sont les deux dernières étapes d'un système de reconnaissance de formes.

Dans ce chapitre nous avons pu introduire les différentes méthodes de segmentation, dont le choix dépend essentiellement de l'image et de ses caractéristiques.

Nous avons passé en vue quelques outils utilisés pour l'extraction des paramètres qui serviront par la suite au module de reconnaissance.

Ainsi, une définition générale d'un système de reconnaissance a été donnée ; nous passons à présent à la seconde phase de notre travail qui consiste à étudier les réseaux de neurones que nous avons choisi pour effectuer la phase d'identification et de reconnaissance.

# 2eme PARTIE

## LES RESEAUX DE NEURONES

### CHAPITRE 4

#### Généralités sur les réseaux de neurones

- 1) Introduction
- 2) Le modèle biologique
- 3) Le modèle artificiel
- 4) Quelques définitions générales
- 5) Classifications des réseaux de neurones
- 6) L'apprentissage des réseaux de neurones
- 7) Les règles d'apprentissage
- 8) Les modèles de réseaux de neurones
- 9) Conclusion

### CHAPITRE 5

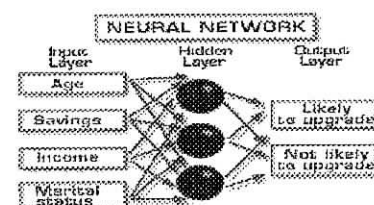
#### Les réseaux multicouches

- 1) Introduction
- 2) Structure et fonctionnement
- 3) Apprentissage
- 4) Taille du réseau et de la base d'apprentissage
- 5) Etape de la conception d'un réseau
- 6) Conclusion

### CHAPITRE 6

#### Exemples d'application des réseaux de neurones

- 1) Introduction
- 2) Avantages et inconvénients des réseaux neurones
- 3) Applications des réseaux de neurones
- 4) Conclusion





# Chapitre IV

## Généralités sur les réseaux de neurones

## **IV-1 Introduction :**

Le terme « Réseau de neurones artificiel » fait rêver beaucoup d'ingénieurs soucieux d'exploiter au maximum les données dont il dispose. Cet outil s'inspire essentiellement du processus de traitement de l'information effectué par le cerveau humain.

Mais avant d'aboutir à ce système, deux principales étapes ont été franchies :

- L'approche algorithmique basée sur la connaissance.
- L'intelligence artificielle qui se révèle être un moyen de stockage de la connaissance sous forme explicite.

## **IV-2 Le modèle biologique : [4]**

### **IV-2-1 Définition du neurone**

Le neurone représente l'élément fonctionnel de base du système nerveux. D'une espèce animale à une autre, le neurone se présente sous différentes formes anatomiques ainsi que pour différentes fonctions.

### **IV-2-2 Anatomie du neurone :**

Le neurone est une cellule constituée principalement de 3 parties : *les dendrites, le soma et l'axone.*

La première partie qui collecte les signaux venant de cellules extérieures forme *l'arbre dendritique.*

L'information arrive au niveau de point de contact appelé les *synapses*, et est acheminée vers le corps cellulaire ou *soma.*

Ce dernier permet de recueillir et concentrer l'ensemble des informations reçues par les dendrites et en effectue une sommation dite spacio-temporelle.

L'information envoyée vers l'extérieur passera par l'axone.

On retrouve souvent la même structure générale : dendrites soma axone mais avec des conformités variables.

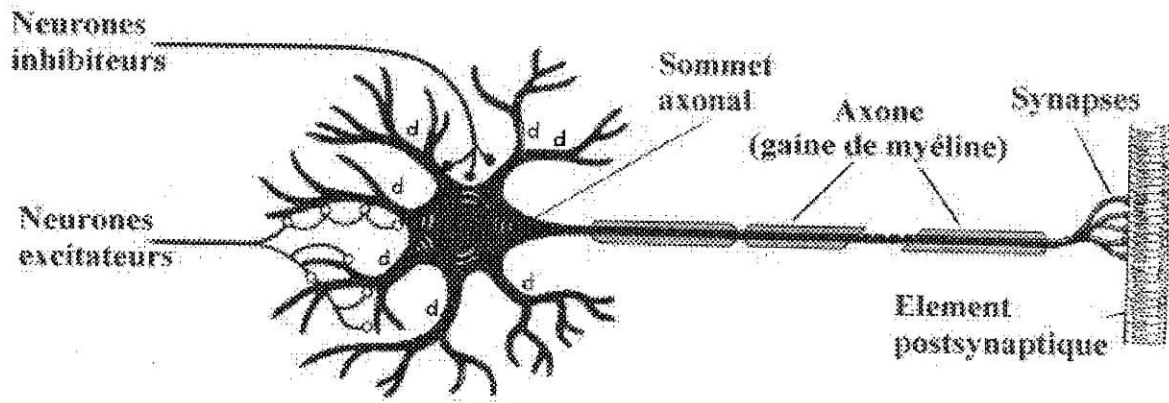


Figure IV - 1: neurone biologique

Pour certaines cellules, la distinction dendrites axone n'existe pas ; dans ce cas on a des inter neurones qui transmettent les signaux électriques d'une cellule à une autre de manière purement analogique.

### **IV-3 Le modèle artificiel : [11]**

#### **IV-3-1 Définition :**

La définition exacte d'un neurone artificiel diffère suivant les organisations et les associations de recherche dans le domaine des réseaux de neurones.

D'après S.Haykin membre de « A Comprehensive Foundation NY », un réseau de neurones est un processeur massivement distribué en parallèle qui a une propension naturelle pour stocker de la connaissance empirique et la rendre disponible à l'usage. Il ressemble au cerveau sur deux aspects :

- 1) La connaissance est acquise par le réseau à travers un processus d'apprentissage.
- 2) Les connexions entre les neurones, connues sous le nom de poids synaptiques, servent à stocker la connaissance.

On peut définir un neurone artificiel comme étant une fonction algébrique non linéaire, paramétrée à valeurs bornées.

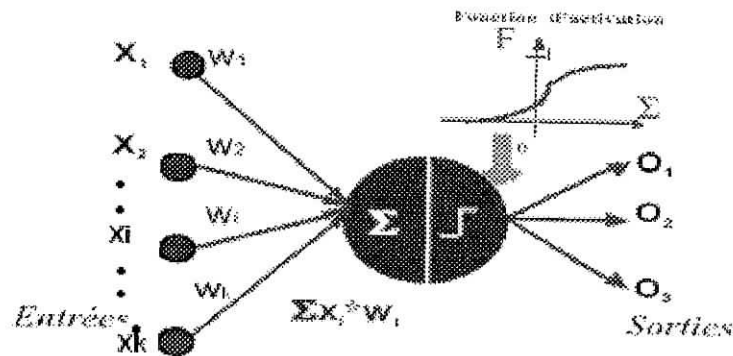


Figure IV - 2 : représentation d'un neurone artificiel

Son intérêt réside dans les propriétés qui résultent de son association en réseaux.

L'expression de la sortie s'écrit comme suit :

$$S = f\left(\sum_{i=1}^p w_i x_i\right)$$

tel que :  $w_i$  représente les poids .  
 $x_i$  représente les entrées.

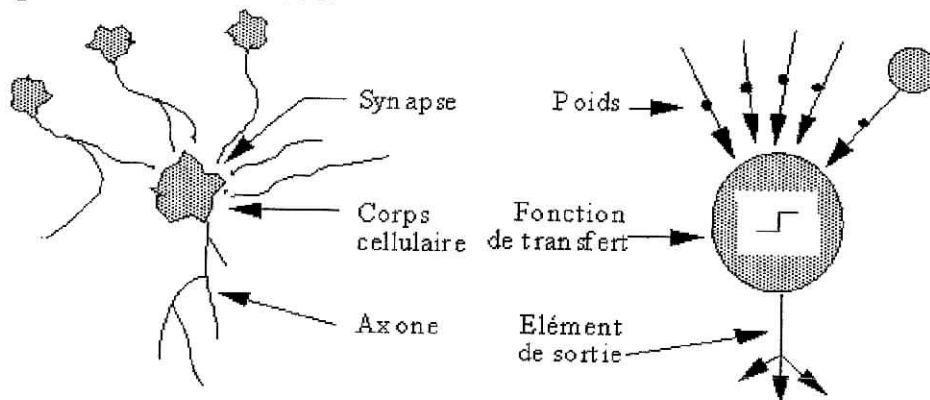


Figure IV - 3 :Dualité entre un neurone artificiel et biologique

### IV-3-2 Historique : [15]

La modélisation du premier neurone artificiel a été effectuée par McCulloch et Pitts en 1943, alors que le processus d'apprentissage fut mis au point en 1949 par Hebb.

Le perceptron réseau de neurones monocouche fut inventé par Rosenblatt en 1958 ; mais ce n'est que 11 ans après que Minsky a mis en exergue les limitations de ce réseau pour leur incapacité de simuler les OU exclusif.

Dix ans après, en 1980 les cartes de Kohonen ont été inventées pour résoudre le problème du perceptron, mais aussi avec un type d'apprentissage différent.

### **IV-3-3 Les Fonctions de bases :**

La fonction de base ou de seuillage  $f$  définit l'état interne en fonction de ses entrées et des poids.

On peut citer deux exemples de fonctions.

#### IV-3-3-1 Fonction binaire à seuil :

Ce type est caractérisé par deux états distincts : 0 = inactif, 1 = actif tel que le déclenchement de l'activité intervient si la somme des excitations dépasse un certain seuil propre au neurone.

*Fonction de Heaviside* :  $f(x) = 1$  si  $x > \text{seuil}$   
 $f(x) = 0$  sinon

Le modèle binaire a été perfectionné car c'est le modèle qui se rapproche le plus de la réalité.

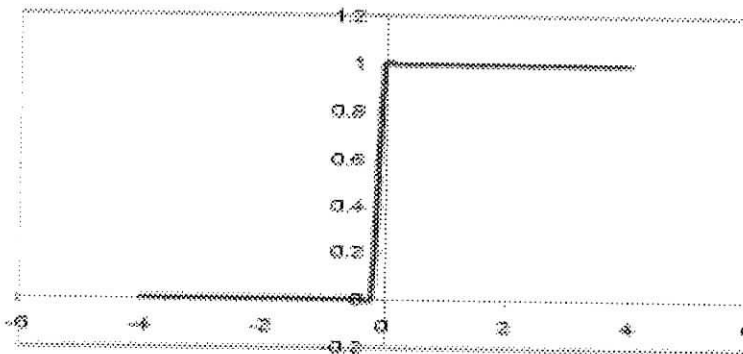


Figure IV - 4 : Fonction binaire à seuil

#### IV-3-3-2 Fonction sigmoïde :

On dit qu'une fonction est sigmoïde lorsqu'elle est monotone et croissante, dérivable et bornée.

Ce type de fonction combine l'avantage de l'effet de seuil et de la dérivabilité. On a plusieurs fonctions sigmoïdes fréquemment utilisées.

◆ la fonction exponentielle  $f(x) = \frac{1}{1 + \text{Exp}(-x)}$

$$f'(x) = f(x)(1 - f(x))$$

elle est la plus utilisée du fait que sa dérivée est simple à calculer

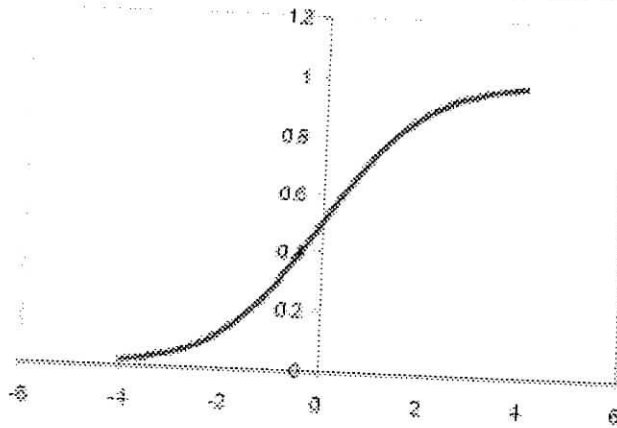


Figure IV - 5 : fonction exponentielle

◆ la fonction tangentielle  $f(x) = \tanh(x)$

$$f'(x) = 1 - f^2(x)$$

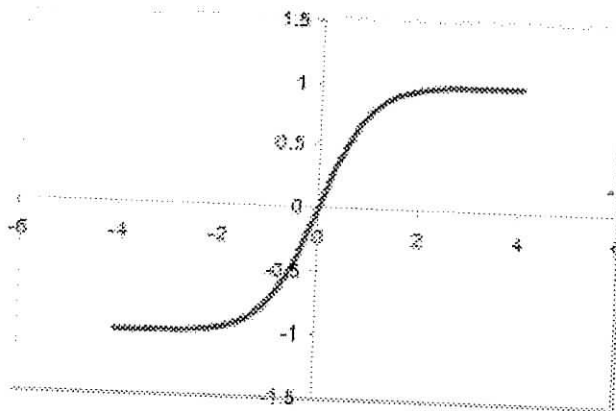


Figure IV - 6 : fonction tangentielle

## **IV-4 Quelques définitions générales: [15]**

### **IV-4-1 Réseau de neurones :**

Ce que l'on désigne habituellement par "réseau de neurones" (neural network), ou réseau neuromimétique est un réseau de neurones artificiels basé sur un modèle simplifié du neurone. Ce modèle permet certaines fonctions du cerveau, comme la mémorisation associative, l'apprentissage par l'exemple, etc. (cependant le neurone formel ne possède pas toutes les capacités des neurones biologiques, comme le partage de synapses, l'activation membranaire ou la structuration prénatale des neurones, par conséquent les réseaux de neurones actuels sont loin d'avoir les possibilités du cerveau).

### **IV-4-2 Mémoire et oubli**

Une caractéristique des réseaux de neurones est leur capacité à apprendre (par exemple à reconnaître une lettre, un son...). Mais cette connaissance n'est pas acquise dès le départ. La plupart des réseaux de neurones apprennent par l'exemple. Ils ont donc une capacité à classer, généraliser, à mémoriser et aussi à oublier. Un réseau de neurones reconnaîtra d'autant plus facilement un objet qu'il l'aura " vu " souvent.

### **IV-4-3 Connectivité :**

La connectivité des réseaux, c'est à dire la manière dont les neurones sont connectés, peut être totale (tous les neurones sont connectés entre eux) ou par couche (les neurones d'une couche ne sont connectés qu'aux couches suivantes - il existe des réseaux monocouches ou multicouches (*multi layer networks*)).

### **IV-4-4 Calcul des poids synaptiques:**

La rétro propagation (back propagation) est une méthode de calcul des poids pour un réseau à apprentissage supervisé qui consiste à minimiser l'erreur quadratique de sortie (somme des carrés de l'erreur de chaque composante entre la sortie réelle et la sortie désirée).

### **IV-4-5 Les neurones d'entrée :**

La première couche est appelée couche d'entrée. Elle recevra les données source que l'on veut utiliser pour l'analyse. Sa taille est directement déterminée par le nombre de variables d'entrées.

#### **IV-4-6 Les neurones cachés :**

La seconde couche est une couche cachée, en ce sens qu'elle n'a qu'une utilité intrinsèque pour le réseau de neurones et n'a pas de contact direct avec l'extérieur. Les fonctions d'activations sont en général non linéaires sur cette couche mais il n'y a pas de règle à respecter. Le choix de sa taille n'est pas implicite et doit être ajusté. En général, on peut commencer par une taille moyenne des couches d'entrée et de sortie mais ce n'est pas toujours le meilleur choix. Il sera souvent préférable pour obtenir de bon résultats, d'essayer le plus de tailles possibles.

#### **IV-4-7 Les neurones de sortie :**

La troisième couche est appelée couche de sortie. Elle donne le résultat obtenu après compilation par le réseau des données entrées dans la première couche.. Sa taille est directement déterminée par le nombre de variables qu'on veut en sortie.

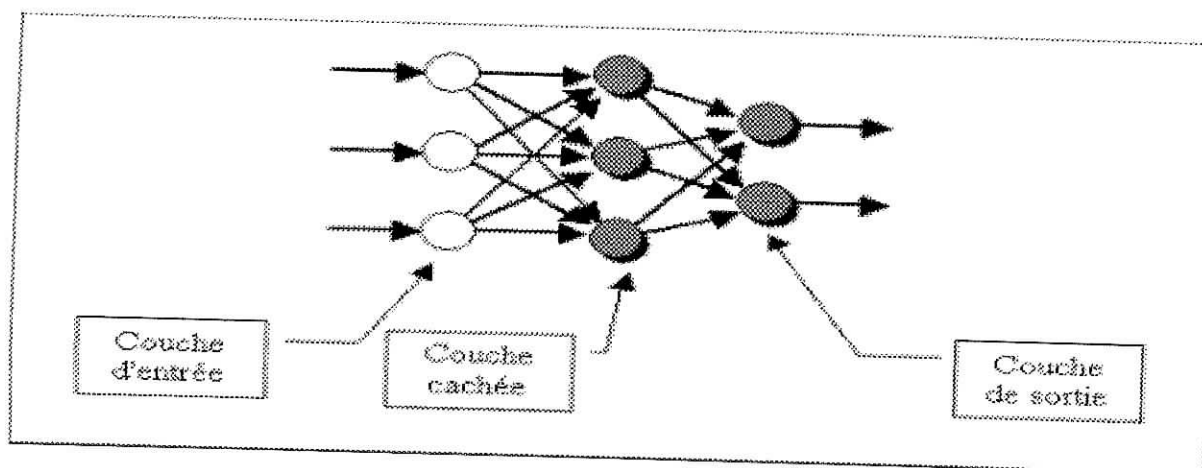


Figure IV - 7 : Structure d'un réseau de neurones

#### **IV-5 Classifications des réseaux de neurones : [11]**

On distingue deux types de réseaux de neurones :

- Les réseaux non bouclés
- Les réseaux bouclés



#### **IV-5-1 Les réseaux de neurones non bouclés :**

Ce type de réseau réalise une fonction algébrique de ses entrées par composition des fonctions réalisées par chacun des neurones.

Un réseau de neurones non bouclé est représenté graphiquement par un ensemble de neurones connectés entre eux ; mais l'information circule des entrées vers les sorties sans retour en arrière.

Ce type de réseau est dit acyclique car si l'on se déplace dans un réseau à partir d'un neurone quelconque suivant les connexions, on ne peut pas revenir au neurone de départ.

#### **IV-5-2 Les réseaux de neurones bouclés :**

C'est l'architecture la plus générale pour un réseau de neurones et la plus répandue ; les réseaux bouclés possèdent un graphe cyclique tel que lorsqu'on se déplace dans le réseau suivant le sens des connexions, on peut toujours revenir vers notre point de départ.

La sortie d'un neurone du réseau peut donc être fonction d'elle-même ; la notion de retard est donc introduite.

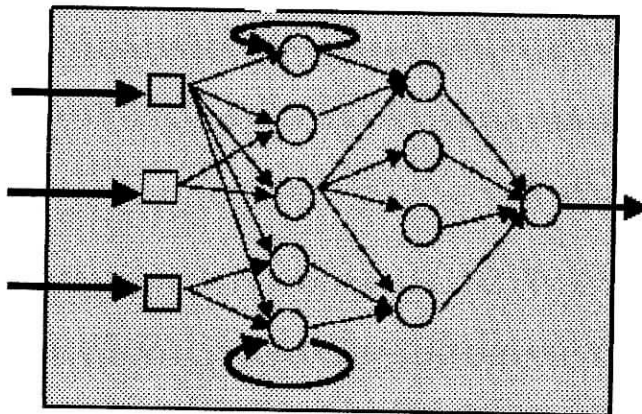


Figure IV - 8 : Réseau de neurone bouclé

### **IV-6 L'apprentissage des réseaux de neurones :[9]**

On appelle apprentissage des réseaux de neurones, la procédure qui consiste à estimer les paramètres des neurones du réseau, réalisé par des algorithmes de calcul dont le but est d'adapter ses poids en fonction des entrées du réseau.

On peut distinguer deux types d'apprentissages :

- L'apprentissage supervisé
- L'apprentissage non supervisé

#### **IV-6-1 L'apprentissage supervisé :**

Ce mode d'apprentissage est le plus courant, schématiquement, il consiste à évaluer les poids synaptiques minimisant l'erreur sur une « base d'apprentissage ».

La base d'apprentissage comprend un ensemble d'observations pour les quelles on connaît à la fois les entrées et les sorties souhaitées pour le réseau.

#### **IV-6-2 L'apprentissage non supervisé :**

Ce mode est moins intuitif. Il correspond aux cas où l'on ne dispose pas de base de données, donc, lorsqu'on ne peut pas déterminer si une sortie est valable ou non.

Cet apprentissage repose alors sur un « critère interne » de conformité du comportement du réseau par rapport à des spécifications générales et non sur des observations externes.

### **IV-7 Les règles d'apprentissage : [11]**

#### **IV-7-1 Règle d'apprentissage supervisé :**

Un apprentissage de réseaux de neurones supervisé revient à modifier les poids de ses connexions.

Ceci se fait en général en calculant  $\Delta w_{ij}$  tel que

$$w_{ij} = w_{ij} + \Delta w_{ij} .$$

Les calculs se font suivant des algorithmes comme celui de windrow Hoff [ annexe]avec une fonction de seuillage supposée dérivable.

#### **IV-7-2 Règle d'apprentissage pour réseaux non supervisé :**

Pour cette méthode d'apprentissage, la sortie n'est en aucun cas connue ; de ce fait, ces réseaux sont moins complexes mais moins précis aussi.

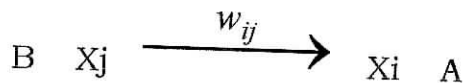
Ce type de réseau est concrètement utilisé pour les problèmes qui nécessitent un traitement en temps réel.

Pour faciliter le travail, cet apprentissage découpe l'espace d'entrée en plusieurs classes avec des régions de décision différentes.  
Il est principalement régit par la loi de Hebb.

La règle de Hebb :

C'est une méthode qualitative d'adaptation des poids synaptiques ; elle peut être exprimée de la manière suivante :

Si un neurone A est stimulé à maintes reprises par un autre neurone B pendant qu'il est actif, le neurone A devient plus sensible aux stimulations en provenance de B ; donc la stimulation de A par B se fera plus facilement à l'avenir.



Tel que  $w_{ij}$  est le poids synaptique qui relie B à A et  $x_i$  est l'activité du neurone.

$$\Rightarrow \Delta w_{ij} = S(X_i)S(X_j)$$

S = sortie = fonction non linéaire appliquée à son activité.

## **IV-8 Différents types de réseaux de neurones : [15]**

### **IV-8-1 Les perceptrons :**

Les perceptrons ont été introduit pour la première fois par Rosenblat en 1958 ; c'est l'un des premiers réseaux de neurones.

Par rapport à ses prédécesseurs basés sur le modèle de MC Culloch et Pitts, il est capable de modifier des poids en fonction des exemples appris.

C'est un réseau linéaire, monocouche avec sortie binaire, comparable aux sorties désirées.

Ce réseau est capable de résoudre des opérations logiques simples comme ET, OU, mais montre une incapacité à simuler la fonction OU exclusif.

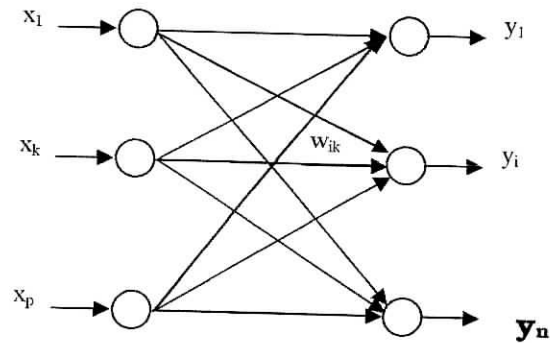


Figure IV-9 : Réseau à une couche

**IV-8-2 Les perceptrons multicouches :**

En prenant le perceptron précédent et en lui ajoutant quelques couches intermédiaires (cachés), on obtient le perceptron multicouches. Pour la modification de ses poids, il utilise l'algorithme généralisé de la règle de windrow Hoff, nommé algorithme de rétro propagation. Les perceptrons multicouches peuvent être utilisés pour la classification et le traitement de l'image car c'est un séparateur non linéaire. Malheureusement, le temps de calcul sur des problèmes de grande taille est assez important.

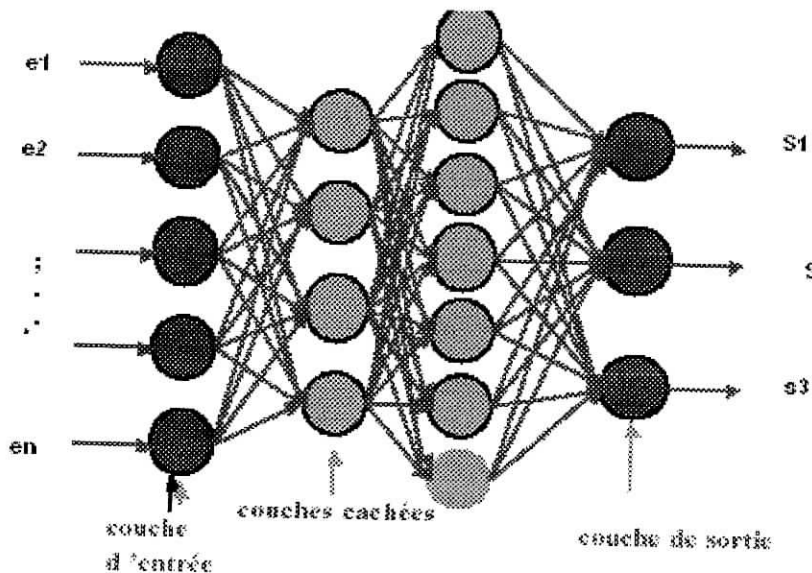


Figure IV-10 : Structure d'un réseau multicouche

**IV-8-3 Réseaux de Hopfield :**

Il dérive des réseaux de MC Culloch et Pitts (binaire) ; sa règle d'apprentissage est la règle de Hebb (définie plus haut).

Ce réseau réalise une mémoire adressable par son contenu, donc il est apte à mémoriser de l'information que l'on peut retrouver en fournissant des données relatives aux informations stockées.

C'est un réseau récurrent, chaque cellule est connectée à toutes les autres et le changement de valeurs s'enchaîne en cascade jusqu'à un état stable

Leur principale utilisation se trouve à régler des problèmes d'optimisation. Mais un inconvénient majeur les handicape, car un réseau peut apprendre une grande quantité d'informations mais les oublie si on lui ajoute un de plus.

**IV-8-4 Réseaux de Kohonen :**

Ce type de réseaux est inspiré des observations biologiques du fonctionnement des systèmes nerveux de perception des mammifères.

Cette carte est souvent comparée à un filet de pêcheur, tel que les nœuds sont les neurones.

L'algorithme d'apprentissage va déployer ce filet dans l'espace des données jusqu'à ce qu'il épouse au mieux la forme du nuage de points.

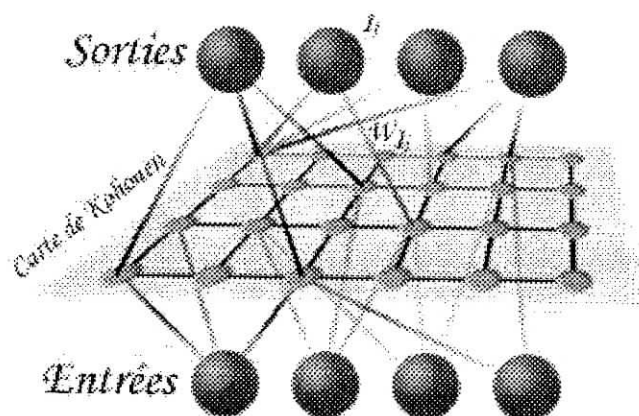


Figure IV-11 : Carte Kohonen

### **IV-9 Conclusion :**

Dans ce chapitre, nous avons présenté un aperçu de l'univers des réseaux de neurones. L'apparition de ces réseaux a révolutionné le monde informatique et électronique, car ils constituent un véritable moyen pour la résolution de plusieurs problèmes.

# Chapitre V

## Chapitre V

# Les réseaux multicouches

## V-1 Introduction :

Pour remédier aux limites du perceptron simple, le réseau multicouche fut mis au point. Un ensemble de cellules d'entrée est connecté à un ensemble de cellules de sortie par une ou plusieurs couches de connexions modifiables, et transparentes à l'utilisateur, qui permettent au réseau de réaliser des associations non linéaires entre l'entrée et la sortie. Chaque neurone n'est relié qu'aux neurones de la couche précédente.

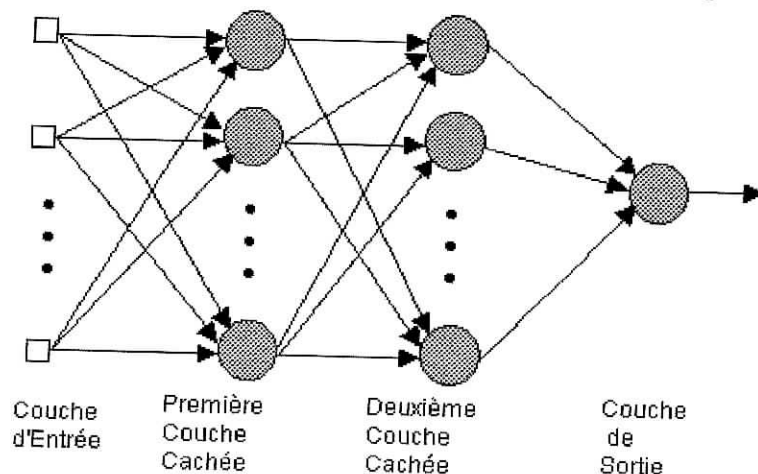


Figure V-1 : Représentation générale d'un réseau multicouche

## V-2 Structure et fonctionnement : [16]

L'utilisation des couches cachées dans les réseaux de neurones a été un pas décisif dans cette discipline.

Ils disposaient dès lors d'un grand degré de liberté pour résoudre les problèmes de classification ou d'association qui leurs étaient posés.

Les réseaux de neurones n'étaient plus limités à trouver une association directe entre les entrées et les sorties mais ils disposaient d'un ensemble de variables intermédiaires pour réaliser cette association.

Ces couches cachées constituaient de véritables représentations internes des problèmes posés.

La fonction de transfert utilisée en général est une sigmoïde qui peut être définie par l'équation suivante :

$$f(a) = \frac{(a \times \alpha - 1)}{(a \times \alpha + 1)} \quad \text{tel que : } \alpha \text{ est la pente de la sigmoïde .}$$

Par exemple pour un réseau à 3 couches : donc une couche d'entrée, une couche de sortie et une couche cachée.



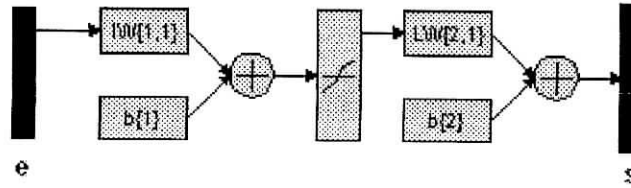


Figure V-2 Réseau de neurones à 3 couches

e représente l'ensemble de données à notre disposition.

s sont les sorties calculées par le réseau.

Pour la couche cachée, ses sorties se représentent en fonction des entrées et des poids de la couche précédente.

Le comportement collectif d'un ensemble de neurones permet l'émergence de fonctions plus importantes par rapport aux fonctions élémentaires du neurone.

### **V-3 Apprentissage : [16]**

L'apprentissage est supervisé : on associe une configuration d'entrée à une configuration de sortie ; l'algorithme de la rétro propagation est un algorithme de gradient itératif conçu pour minimiser un critère quadratique d'erreur entre la sortie obtenue et celle désirée.

Cette minimisation est réalisée par une configuration des poids adéquate.

Cet algorithme nécessite une fonction continue non linéaire et différentielle comme fonction de transfert du neurone.

L'algorithme est le suivant :

- 1/ Initialisation des poids à des valeurs aléatoires de faible grandeur;
- 2/ Sélection d'un exemple d'apprentissage (E, d) dans la base d'apprentissage
- 3/ Présentation de la forme d'entrée (E) sur la couche d'entrée du réseau;
- 4/ Calcul par propagation de la sortie obtenue (o);
- 5/ Si erreur en sortie alors pour tous les neurones i (depuis la sortie jusqu'à l'entrée)  
 Si i est un neurone de sortie alors  $y_i = 2 f'(a_i).(d_i - x_i)$  ;  
 Si i est un neurone caché (ou d'entrée) alors  $y_i = f'(a_i) \cdot \sum_k (w_{ki} \cdot y_k)$  ;  
 (k : neurones compris entre la couche actuelle et la couche de sortie)
- 6/ Application de la procédure de gradient.  $\mu$  est un gain fixé par l'utilisateur.  
 $w_{ij}(t+1) = w_{ij}(t) + \mu \cdot y_i \cdot x_j$  ;
- 7/ Tant que l'erreur est trop importante, retour à l'étape 2 (exemple suivant).

L'algorithme de la rétro propagation de gradient, bien que très simple à implanter, nécessite un ajustement des variables en fonction du problème traité.

Parmi ces variables à fixer, citons par exemple : la sélection des exemples pour l'apprentissage et le test, l'ordre de présentation et les distributions relatives des exemples dans la base d'apprentissage, le choix du codage des informations en entrée et en sortie, la structure du réseau (présence éventuelle de connexions directes de la couche d'entrée sur la couche de sortie pour traiter à ce niveau la partie linéaire du problème, limitation pratique du nombre de couches, taille de la couche cachée), la configuration initiale des poids, le nombre d'itérations d'apprentissage, ...

La généralisation de cet algorithme est donnée par la règle de Windrow Hoff . elle est plus appropriée pour ce réseau et est présentée en annexe.

#### **V-4 Taille du réseau et de la base d'apprentissage :[16]**

Pour un bon apprentissage il faut veiller à ne pas avoir un sur apprentissage , qui est dû à un nombre trop grand de paramètres dans le réseau par rapport au nombre d'exemples dans la base , au détriment d'une réponse pertinente pour des exemples non appris .

Sur la base d'apprentissage, l'erreur diminue toujours, alors que sur la base de test elle passe par un minimum. Si l'apprentissage se prolonge au-delà, les performances en test diminuent.

#### **V-5 Les étapes de la conception d'un réseau :**

Pour construire un réseau de neurones, la première chose à faire n'est pas de choisir le type de réseau mais de bien choisir son/ses échantillons de données d'apprentissage, tests et validation. Ce n'est qu'ensuite que le choix du type de réseau interviendra. Voici chronologiquement les quatre grandes étapes qui doivent guider la création d'un réseau de neurones.

##### **V-5-1 Choix des échantillons :**

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix des échantillons. Cette étape est cruciale et va aider le concepteur à déterminer le type de réseau le plus approprié pour résoudre son problème. La façon dont se présente l'échantillon conditionne : le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la façon dont il faudra mener l'apprentissage, les tests et la validation.

##### **V-5-2 Elaboration de la structure du réseau :**

La structure du réseau dépend étroitement du type des échantillons. Il faut d'abord choisir le type de réseau : un perceptron standard, un réseau de

Hopfield, un réseau à décalage temporel (TDNN), un réseau de Kohonen, un ARTMAP etc... Dans le cas du perceptron par exemple, il faudra aussi choisir le nombre de neurones dans la couche cachée. Plusieurs méthodes existent et on peut par exemple prendre une moyenne du nombre de neurones d'entrée et de sortie, mais rien ne vaut de tester toutes les possibilités et de choisir celle qui offre les meilleurs résultats.

### **V-5-3 Apprentissage :**

L'apprentissage est l'avant dernière phase d'élaboration d'un réseau de neurones. Elle consiste tout d'abord à calculer les pondérations optimales des différentes liaisons, en utilisant un échantillon. La méthode la plus utilisée est la rétro propagation : on entre des valeurs des cellules d'entrée et en fonction de l'erreur obtenue en sortie (le delta), on corrige les poids accordés aux pondérations. C'est un cycle qui est répété jusqu'à ce que la courbe d'erreurs du réseau ne soit pas croissante (il faut bien prendre garde à ne pas surentraîner un réseau de neurones qui deviendra alors moins performant).

### **V-5-4 Validation**

Une fois le réseau calculé, il faut procéder à des tests pour vérifier que le réseau réagit bien comme on le souhaite : c'est la validation. Il y a plusieurs méthodes pour faire cela. La méthode la plus simple étant de garder une partie de l'échantillon réservé à l'apprentissage, pour la validation et faire ainsi une validation hors échantillon.

### **V-6 Conclusion :**

Les réseaux multicouches entraînés pas l'algorithme de rétro propagation du gradient sont les modèles connexionnistes les plus utilisés à ce jour. Cependant, une utilisation efficace demande une analyse préalable aussi soigneuse que possible du problème. Le réseau apprend les exemples de la base d'apprentissage, dont les réponses désirées sont élaborées par le modèle que constitue le superviseur.

## Chapitre VI

# Exemples d'application des réseaux de neurone

## **VI -1 Introduction :**

Inspirée par le système nerveux biologique, la technologie des réseaux de neurones est utilisée pour résoudre un large éventail de problèmes scientifiques, d'ingénierie et économiques complexes. Ses applications commerciales incluent les opérations d'investissement, le data mining, le contrôle de processus, la suppression du bruit, la compression de données et la reconnaissance vocale autant de domaines différents les uns des autres.

Les réseaux de neurones sont particulièrement adaptés à de tels problèmes car, comme leurs équivalents biologiques, ils sont en mesure d'apprendre et peuvent donc être formés pour trouver des solutions, reconnaître des séquences, classifier des données et prévoir des événements futurs.

## **VI-2 Avantages et inconvénients des réseaux de neurones :**

Bien que les réseaux neuronaux possèdent une longue histoire qui remonte aux débuts de l'informatique et de la cybernétique. Ce n'est que dans les 20 dernières années que leurs évolutions ont eu vraiment lieu. L'approche connexionniste, à la quelle participe les réseaux neuronaux est fondée sur des modèles biologiques, semblables à ceux que l'on retrouve en physiologie.

Les réseaux de neurones modélisent l'architecture biologique du cerveau humain, leur état de développement est encore loin de simuler les raisonnements complexes de l'intelligence.

Voici quelques *avantages* que présentent les réseaux neuronaux :

- ❖ L'approche Connexionniste est un complément indispensable aux modèles symboliques de l'intelligence artificielle.
- ❖ Les réseaux neuronaux sont bien adaptés à un traitement sur des machines parallèles où chaque neurone est traité par un processeur indépendant des autres, ce qui accélère les calculs.
- ❖ Les réseaux neuronaux dépendent peu des connaissances préalables et de leurs modélisations, car ils apprennent des fonctions qui peuvent résoudre un problème donné, ce qui leur permet de traiter de nouvelles situations qui n'ont pas à être prévues au départ.
- ❖ Les problèmes non linéaires sont difficiles à résoudre mathématiquement. les réseaux neuronaux apportent une solution par leur capacité à traiter tout problème qui peut être représenté par patterns à transformer en d'autres patterns, même si la relation entrée sortie n'est pas linéaire.

Les réseaux neuronaux présentent aussi quelques *inconvenients* :

- ❖ Contrairement aux représentations symboliques de l'intelligence, la représentation par réseau neuronal se trouve dans les poids sur les liaisons entre neurones et dans les fonctions d'activation.
- ❖ Les réseaux neuronaux n'arrivent à simuler que les aspects très limités de l'intelligence humaine.
- ❖ L'entraînement d'un réseau neuronal est souvent une tâche longue et fastidieuse.

### **VI-3 Application des réseaux de neurones :**

C'est dans le domaine de la reconnaissance de caractères que les réseaux de neurones ont acquis leurs lettres de noblesse et ont prouvé qu'ils constituent des alternatives fiables à d'autres méthodes de classification.

Se trouvant à l'intersection de différents domaines (informatique, électronique, science cognitive, neurobiologie ...) l'étude des réseaux de neurones est une voie prometteuse de l'intelligence artificielle.

Voici quelques exemples d'applications :

➤ **En contrôle industriel :**

Les entrées du réseau peuvent être une mesure de la température en différents points d'une fournaise, les sorties calculent les valeurs des différents indicateurs qui alertent un opérateur et requièrent de sa part différentes actions.

➤ **Dans le domaine financier :**

On utilise les données historiques d'une action pour prédire un mouvement de prix futur de cette action.

➤ **En reconnaissance :**

● *De la parole :*

Les entrées peuvent être une segmentation de la fréquence d'un signal sonore, et les sorties identifient la valeur de tel ou tel phénomène associé à l'entrée. Ou bien on peut transcrire le langage parlé en texte ASCII.

● *De cibles :*

Applications militaires qui utilisent des données issues d'images vidéo ou infrarouge pour déterminer la présence d'une cible ennemie.

● *Codes postaux*

Cette application fût l'un des premiers problèmes de reconnaissance d'images assez complexe, en raison de la variabilité des styles d'écritures, mais en plus cette application est moins onéreuse que de rectifier une erreur de tri.

● *D'écriture :*

Qu'elle soit manuscrite ou imprimée, le réseau va être confronté à un problème de classification, par rapport à la base d'apprentissage qui constitue la base de données.

➤ **En médecine :**

Les entrées peuvent être un ensemble de valeurs résultant d'une liste de tests médicaux et en sortie les symptômes possibles de maladies infectieuses.

## **VI-4 Conclusion :**

La reconnaissance de formes , en particulier des écritures ou toute autre application citée plus haut sont entrées depuis un moment dans une phase d'applications industrielles grâce au développement de l'informatique . Les méthodes utilisées avec apprentissage comme les réseaux de neurones ne permettent pas de surpasser les méthodes structurales, mais elles sont utilisées surtout pour leurs rapidités de calculs et la puissance du traitement.

# 3eme PARTIE

## L'APPROCHE ADOPTEE

### CHAPITRE 7

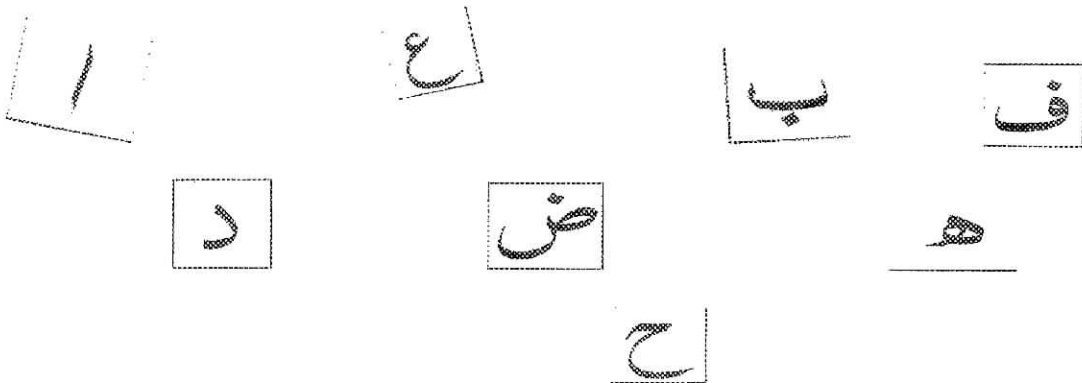
#### Travail effectué

- 1) Introduction
- 2) Acquisition
- 3) Modèle de prétraitement
- 4) Segmentation
- 5) Reconnaissance

### CHAPITRE 8

#### Description du logiciel

- 1) Introduction
- 2) Configuration de l'application
- 3) Résultats





# Chapitre VII

# Travail effectué

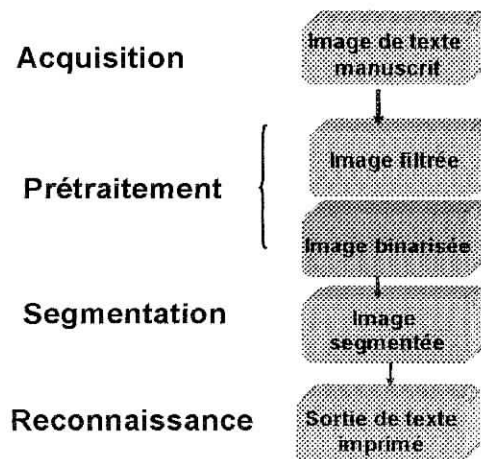
## **VII-1 Introduction :**

A l'issue des deux premières parties de ce mémoire, on a pu acquérir toutes les théories du traitement d'image et des réseaux de neurones. Le but maintenant est de mettre en pratique tout ceci, donc de mettre en œuvre des algorithmes pour les implémenter et réaliser le système de reconnaissance de l'écriture manuscrite arabe.

Des recherches dans ce domaine commencent à voir le jour ; plusieurs projets de fin d'études ont été proposés dans ce cadre, avec différentes méthodes de traitement et de reconnaissance, sachant que le manuscrit représente d'énormes irrégularités dans la forme, suivant le type de scripteur, son humeur ...

Dans cette dernière partie, les différents algorithmes, les méthodes utilisées ainsi que les résultats obtenus seront exposés.

Le schéma général du système se présentera comme suit :



## **VII-2 L'acquisition :** [6]

Afin d'avoir à notre disposition un texte manuscrit à traiter, nous avons numérisé des documents sous forme d'image matricielle.

Pour cela, un scanner de type scanner plat a été utilisé afin de permettre de choisir les paramètres de numérisation souhaités, par exemple : la résolution, la taille de l'image, le choix entre la couleur et le niveau de gris. Ainsi, nous avons obtenu une image de format *jpeg*, que l'on peut sauvegarder sous le format *bitmap* qui représente le format le plus simple à utiliser en traitement d'image.

### VII-3 Module de prétraitement : [6]

Le module de prétraitement prépare l'image issue de la phase d'acquisition à l'étape suivante de segmentation qui nécessite une image dénuée de bruit. En effet, le moindre pixel chargé, peut fausser toute l'analyse du texte et induire en erreur sa reconnaissance.

Donc, pour améliorer la qualité de l'image, on applique un prétraitement. Le module de prétraitement est composé de deux étapes, le filtrage et la binarisation.

#### ◆ L'étape de filtrage :

Parmi les différents filtres qui se présentent à nous, nous avons choisi le filtre médian qui est le plus approprié pour éliminer le bruit impulsionnel sans affecter la forme des caractères.

Le principe du filtre médian est caractérisé par :

- ◆ balayer l'image horizontalement
- ◆ pour chaque pixel  $i$ , prendre ses huit (8) pixels voisins
- ◆ classer les neuf (9) pixels obtenus par ordre croissant selon leur valeur de niveau de gris.

Cette dernière sera calculée avec l'équation fondamentale de la télévision :

$$N = 0,3 * R + 0,59 * V + 0,11 * B$$

Dès que les 9 pixels sont rangés, on remplace la valeur du pixel  $i$  par la valeur médiane.

Ce procédé est illustré par la figure suivante :

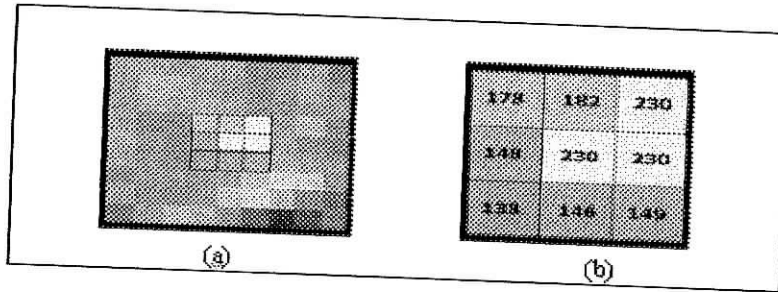


Figure VII - 1 a\ Portion agrandie d'une image bruitée.

b\ niveau de gris des pixels voisins.

La correction du pixel central de l'image de la figure VIII-1 se ferait donc de la façon suivante :

- Classer toutes les valeurs de luminance en ordre croissant : 138, 146, 148, 149, **178**, 182, 230, 230, 230
- Remplacer le pixel par la valeur médiane du classement. Dans ce cas 230 (intensité du pixel central) sera remplacée par 178

الطرف في جنات الجيرة التي تنقرأ فيها فماذا أدر  
 لستبط العلم طاقتهما سوى كامنة في ذرات المادة

Figure VII - 2 : Image d'un document scanné en couleur

الطرف في جنات الجيرة التي تنقرأ فيها فماذا أدر  
 لستبط العلم طاقتهما سوى كامنة في ذرات المادة

Figure VII - 3: Image d'un document converti en niveaux de gris

Après avoir diminué le bruit, on doit binariser l'image.

❖ L'étape de binarisation :

La binarisation consiste à éliminer tous les niveaux de gris transitoires entre le blanc et le noir, et ne garder que ces deux couleurs. Pour cela, deux approches ont été utilisées :

- ❖ L'approche manuelle où l'utilisateur fait son propre choix du seuil en utilisant une barre graduée entre 0 et 255.

Après introduction de cette donnée, on applique l'algorithme suivant :

**Début**

```

Image NG = Image en niveaux de gris
Imagebin = Image binarisé
pour i = 0 à largeur de l'image faire
  pour j = 0 à hauteur de l'image faire
    si ( ImageNG[i][j] < seuil ) faire
      Imagebin [i][j] = 0 ;
    Sinon Imagebin [i][j] = blanc ;
  
```

**Fin**

- ❖ La seconde approche est celle dite automatique où le calcul du seuil dépend de l'image elle-même. On calcule le seuil du fond de notre image en niveaux de gris, ensuite le seuil de l'objet présent sur l'image qui représente dans notre cas l'écriture.

Ces deux valeurs sont déterminées suivant l'histogramme de l'image en niveaux de gris et suivant l'algorithme énoncé au (chapitre II-4-2).

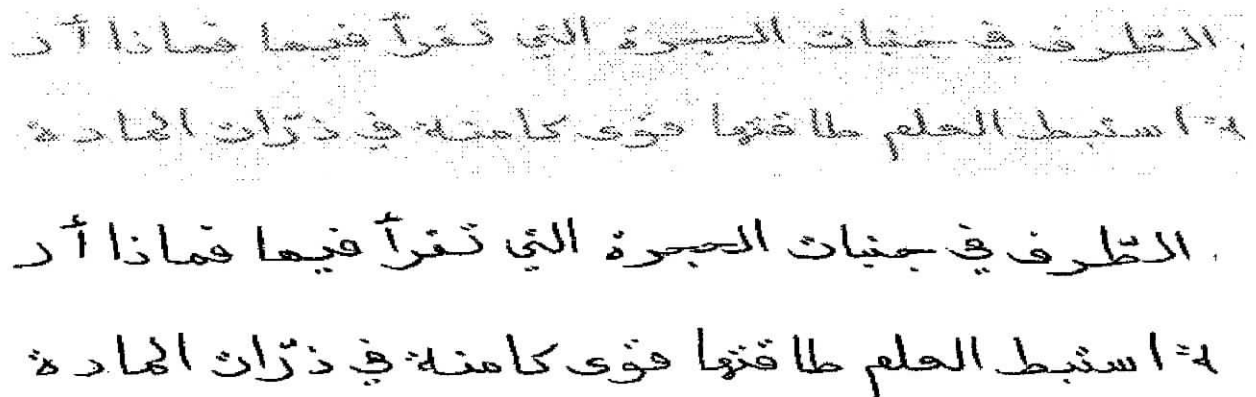


Figure VII – 4 : Passage d'un document en niveaux de gris à un document binarisé

Après la binarisation, on peut avoir des textes avec des trous au milieu.

Ceci est dû à une mauvaise acquisition (résolution trop basse) ; pour y remédier, on applique des filtres morphologiques comme la dilatation et l'érosion.

La dilatation va permettre de combler les trous qui subsistent dans le texte ; tandis que l'érosion va affiner l'écriture.

#### Algorithme de la dilatation :

**pour** le pixel  $i$  de la matrice Imagebin, //on récupère ses 8 voisins.

**Si** Imagebin[ $i$ ][ $j$ ] est blanc

on aura Imagedilat[ $i$ ][ $j$ ] = **ET** [ Imagebin[ $i$ ][ $j$ ] ]  
 $i \neq j$

**sinon**

Imagedilat[ $i$ ][ $j$ ] = Imagebin[ $i$ ][ $j$ ]

#### Algorithme de l'érosion :

// On récupère les pixels voisins.

**Si** Imagebin[ $i$ ][ $j$ ] est noir

on aura Imageerod[ $i$ ][ $j$ ] = **OU** [ Imagebin[ $i$ ][ $j$ ] ]  
 $i \neq j$

**sinon**

Imageerod[ $i$ ][ $j$ ] = Imagebin[ $i$ ][ $j$ ]

A la fin de toutes ces étapes, on obtiendra une image prête à être traitée et on pourra passer à l'étape suivante qui est la segmentation.

## **VII-4 Segmentation d'un texte : [10]**

Elle consiste à isoler chaque caractère de notre texte pour qu'il puisse être reconnu.

C'est une étape primordiale dans le système car l'essentiel du travail (reconnaissance) dépend de la qualité de la segmentation.

Pour effectuer la segmentation, nous avons défini plusieurs étapes.

### **VII-4-1 La segmentation horizontale :**

Le but principal de cette étape est de localiser les lignes de texte ; elle est réalisée en projetant horizontalement la page de texte sur un axe vertical.

Il est nécessaire de différencier entre la ligne de texte et la ligne pixel.

- ❖ La ligne pixel est un ensemble de pixels adjacents horizontalement pouvant prendre chacun la valeur 0 ou 1 selon qu'il y ait de l'écriture ou pas.
- ❖ La ligne de texte est un ensemble de lignes pixel se trouvant les unes au dessus des autres.

La segmentation horizontale se base sur l'histogramme horizontal dont on présente l'algorithme ci-dessous :

*Algorithme de l'histogramme horizontal :*

**Début :**

**Pour** toute l'image binarisée  
**si** aucun point noir  
**aller** à la ligne suivante  
**sinon** sauvegarder le début de ligne

**si** la ligne contient au max un point noir  
**aller** à la ligne suivante  
**sinon** sauvegarder la fin de la ligne  
**aller** à la ligne suivante

**Fin**

L'histogramme obtenu présente :

- ❖ des zones non nulles : les lignes pixels
- ❖ des zones blanches : les interlignes.

*Algorithme de la segmentation horizontale :*

**Début :**

**Pour** toute l'image binarisée  
**Faire** histogramme horizontal  $H[j]$   
**Pour** toutes les lignes faire  
**Si** ( $H[j] = 0$  et  $H[j+1] \neq 0$ )  
*Sauvegarder* début de ligne pixels .  
**Sinon**  
**Si** ( $H[j] \neq 0$  et  $H[j+1] = 0$ )  
*sauvegarder* fin de la ligne pixels.

**Fin**

Le résultat est schématisé ci-dessous :

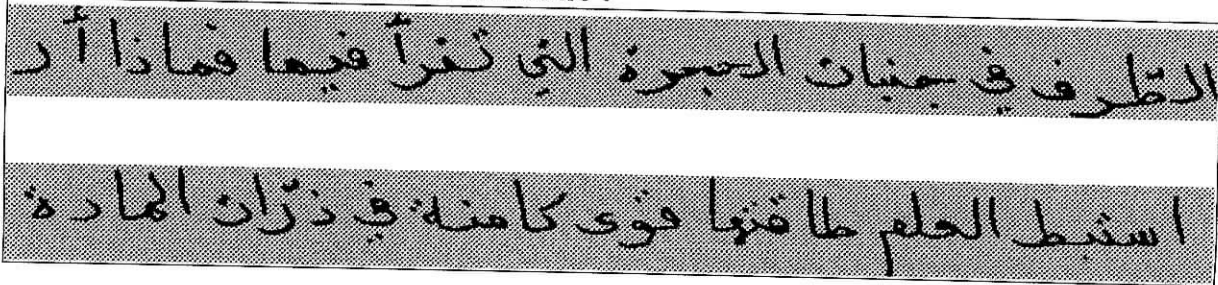


Figure VII – 5 : Segmentation horizontale

Le texte est séparé par des interlignes.

Remarque : On peut déterminer la ligne médiane qui représente la plus forte concentration de pixels.

#### **VII-4-2 La segmentation verticale :**

Après la segmentation horizontale, on obtient un découpage du texte ligne par ligne ; pour avoir un découpage par colonne, donc par élément de mot, la segmentation verticale s'impose. Sa qualité dépend essentiellement du calcul de l'histogramme vertical dont on présente l'algorithme ci-dessous :

*Algorithme de l'histogramme vertical :*

**Pour** chaque ligne de texte

**Tant que non** la colonne ne contient aucun point noir

{

**saut** vers la colonne suivante

}

**sauvegarder** la colonne

**Tant que non** la colonne contient au minimum un point noir

{

**saut** vers la colonne suivante

}

**sauvegarder** la colonne fin

à l'exécution de cet histogramme, on obtient pour chaque colonne le nombre de pixels allumés, donc une projection du texte sur les axes. Et c'est à l'aide de cet opérateur que l'algorithme de la segmentation verticale peut être réalisé.



Algorithme de la segmentation verticale :

**Début :**

**Pour** chaque ligne du texte issue de la segmentation horizontale

**Faire**

histogramme vertical  $H[i]$

**Pour** toutes les colonnes  $i$

**faire**

**Si** ( $H[i] = 0$  et  $H[i+1] \neq 0$ )

**Faire**

Début de la partie connexe  $N$  à sauvegarder

**Si** ( $H[i] \neq 0$  et  $H[i+1] = 0$ )

**Faire**

Fin de la partie connexe  $N$  à sauvegarder

**Fin**

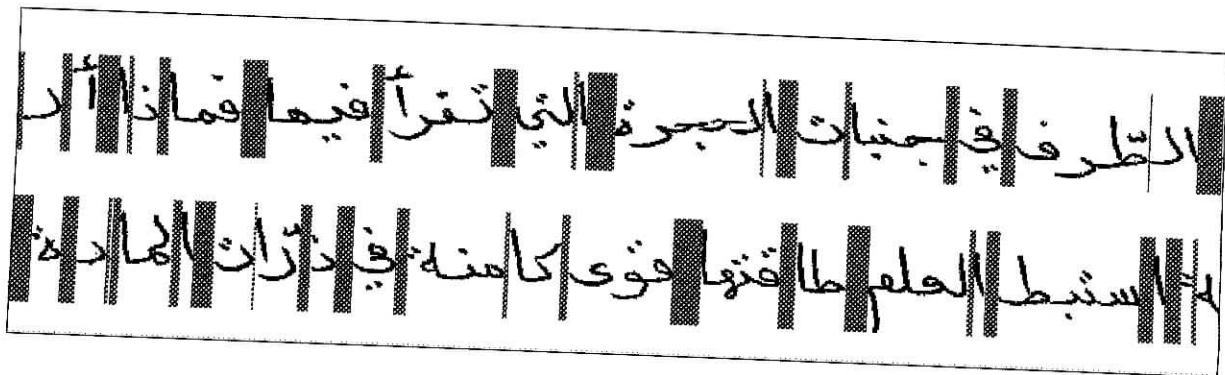


Figure VII - 6 : Segmentation Verticale

Lorsqu'on associe les deux algorithmes on réalise la segmentation horizontale-verticale comme illustrée ci-dessous :

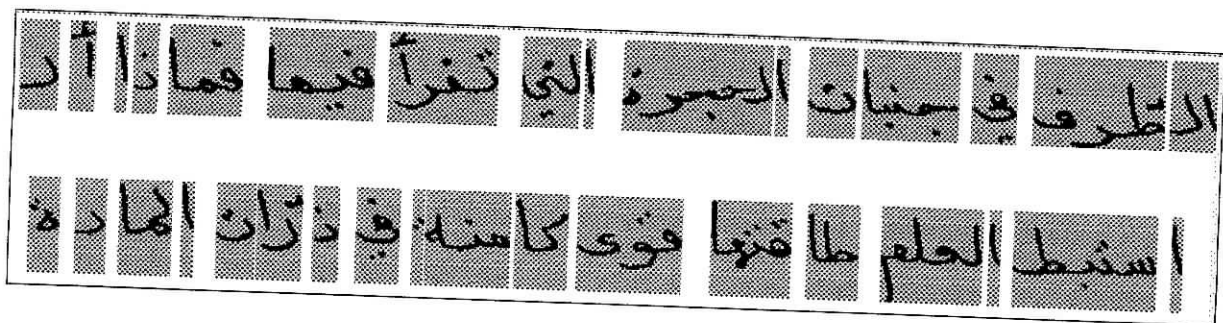


Figure VII - 7 : segmentation horizontale-verticale(en mots)

Figure VII – 8 : Ligne médiane de chaque ligne de texte

La segmentation en parties connexes par utilisation de l'histogramme vertical ne donne pas de bons résultats en écriture manuscrite (contrairement à l'imprimé) à cause des chevauchements de caractères n'appartenant pas à la même partie connexe. Donc la segmentation par étiquetage s'avère plus intéressante comme nous allons le voir tout de suite

#### **VII-4-3 La segmentation par étiquetage en composantes connexes : [5]**

L'extraction des composantes connexes est largement utilisée en reconnaissance des formes pour segmenter des images binaires. La technique consiste à regrouper les pixels voisins dans un ensemble appelé composante connexe.

Afin de réaliser notre segmentation, nous avons opté pour une méthode d'étiquetage qui passe par deux étapes essentielles :

- ❖ Un balayage de l'image ligne par ligne pour affecter une étiquette à chaque pixel
- ❖ Résolution des équivalences et un second balayage de l'image pour l'attribution des étiquettes définitives aux pixels.

Après le premier balayage, une région peut avoir plusieurs étiquettes ; pour éviter une superposition d'étiquettes, une mise en place d'une table d'équivalence est faite, pour la sauvegarde des étiquettes. L'étiquette finale pour chaque région connexe obtenue passe nécessairement par une table d'équivalence.

- Une étiquette a est dite équivalente à l'étiquette b s'il existe un chemin reliant le point a au point b dans l'image tel que le chemin ne passe pas par 0 .
- La table d'équivalence pour l'implémentation de cette structure, consiste à utiliser un tableau dynamique telle que la taille augmente pendant le parcours séquentiel de l'image.

## a) Processus d'étiquetage :

En parcourant une ligne horizontale de gauche à droite, on associe un numéro à chaque pixel de telle sorte que tous les pixels voisins portent le même numéro. Le numéro zéro est réservé pour un pixel vide.

Lorsque sur cette ligne, le voisinage est interrompu, puis reprend plus loin, le numéro est incrémenté de 1.

Lorsqu'une nouvelle ligne est commencée, on propage l'étiquetage de haut en bas tel que si le pixel qui se trouve au dessus du premier pixel de la nouvelle ligne n'est pas nul, ce dernier prend la même étiquette sinon une nouvelle est utilisée.

Lorsqu'un conflit se présente entre la propagation horizontale et la propagation verticale, on choisit la plus petite étiquette.

*Algorithme du premier balayage :*

```

Soit 0 = noir    255= blanc
Imgbin = la matrice de l'image binarisée ;
Etiq = 0 ;
Pour toutes les lignes de Imgbin faire
Tant que non fin de la ligne courante faire
{
  si (Imgbin != 255 ) alors
  {
    si ( tous les prédécesseurs de Imgbin sont à 255 )
    alors Imgbin = etiq++
    sinon
    si ( tous les prédécesseurs de Imgbin ont la même étiquette 'e' )
    alors Imgbin = e
    sinon
    {
      soit e1 = min ;
      Imgbin = e1
    }
  }
  Mise à jour de la table d'équivalence
  {
  }
}

```

Plus on avance dans notre balayage de l'image, plus on a de nouvelles étiquettes, plus la table d'équivalence s'élargit en colonne.

Cet algorithme est basé sur la 8 connexité donc chaque pixel possède trois prédécesseurs.

Cet algorithme fait appel à deux sous-programmes.

❖ le sous-programme min :

Il permet de calculer le minimum des trois étiquettes des prédécesseurs du point traité, en prenant le cas particulier où l'un d'eux est nul.

❖ Le sous-programme de la mise à jour de la table d'équivalence :

Lors du premier balayage de l'image, et en arrivant à un pixel à étiqueter dont les prédécesseurs sont différents, on prend la dernière étiquette de la colonne (au niveau de la table d'équivalence) correspondante à l'étiquette de l'un des trois prédécesseurs ; soit « a » cette étiquette.

Ensuite, on récupère les dernières étiquettes des colonnes correspondantes aux étiquettes des autres prédécesseurs, soient « b » et « c », ces étiquettes.

Après une comparaison à l'aide du sous-programme min, l'étiquette la plus basse est attribuée au pixel courant, et elle est insérée dans les colonnes des étiquettes des autres prédécesseurs dans l'ordre décroissant.

Ainsi pour une même colonne (à l'exception de la première étiquette), elles sont classées de la plus grande à la plus petite ; donc à la fin de chaque colonne, on obtient la plus petite étiquette équivalente à la première étiquette de la même colonne.

b) La résolution des équivalences :

Cette partie se fait directement au niveau de la table d'équivalence ; elle consiste à affecter à chaque étiquette la plus petite qui lui soit équivalente. Pour ce faire, nous avons utilisé l'algorithme suivant :

**Début :**

```

Pour i=0 → max-ligne faire
  Pour j=0 → max-eti faire
  {
    Si ( T[i+1][j] ≠ 255 ) alors
    {
      pour (k=0 → max-eti ) faire
      {
        si (T[0][k] =T[0][j] ) alors
          T[0][k] =T[i+1][j] ;
        }
      }
    }
  }

```

**Fin**

Tel que max\_ligne : est l'indice de la dernière ligne de la table d'équivalence.  
max\_etiq : est le nombre maximal d'étiquette.

$T[i][j]$  : est la matrice table d'équivalence.

Algorithme du deuxième balayage :

Il permet d'affecter à chaque pixel son étiquette équivalente.

**Début**

**Pour** toute l'image imgbin **faire**

```
{
  si ( imgbin [i][j] ≠ 255) alors
    img[i][j]=T[o][imgbin[i][j]] ;
}
```

**Fin**

**Processus de segmentation par étiquetage en parties connexes**

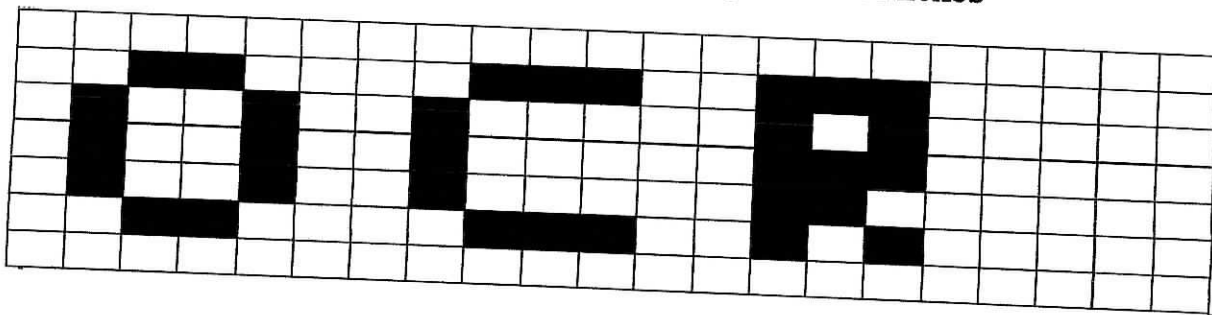


Figure : VII - 9 exemple d'une image à segmenter

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0
0	1	0	0	1	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0
0	0	1	1	0	0	0	0	1	1	1	0	0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure : VII - 10 Image de départ initialisée

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	2	2	2	0	0	3	3	3	0	0	0	0	0
0	4	0	0	1	0	0	5	0	0	0	0	0	3	0	3	0	0	0	0	0
0	4	0	0	1	0	0	5	0	0	0	0	0	3	3	3	0	0	0	0	0
0	4	0	0	1	0	0	5	0	0	0	0	0	3	3	0	0	0	0	0	0
0	0	4	4	0	0	0	0	5	5	5	0	0	3	0	3	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure : VII - 11 Image des étiquettes après le premier passage

1	2	3	4	5
			1	2

Figure : VII - 12 Table d'équivalence correspondante

1	2	3	1	2
---	---	---	---	---

Figure : VII - 13 Table d'équivalence après résolution des équivalences

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	2	2	2	0	0	3	3	3	0	0	0	0	0
0	1	0	0	1	0	0	2	0	0	0	0	0	3	0	3	0	0	0	0	0
0	1	0	0	1	0	0	2	0	0	0	0	0	3	3	3	0	0	0	0	0
0	1	0	0	1	0	0	2	0	0	0	0	0	3	3	0	0	0	0	0	0
0	0	1	1	0	0	0	0	2	2	2	0	0	3	0	3	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure : VII - 14 Image des étiquettes finales

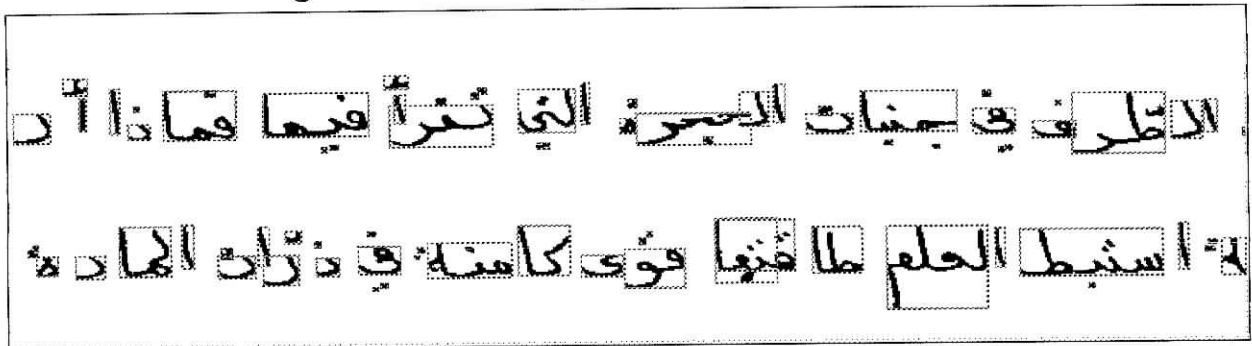


Figure : VII - 15 segmentation par étiquetage en parties connexes

استبط العلم طاقتموا قوى كامنه في ذرات المادة  
 لتظرف في جنات الصيرة التي تنفراً فيما فماذا أدر

Figure : VII - 16 Ligne médiane de chaque partie connexe

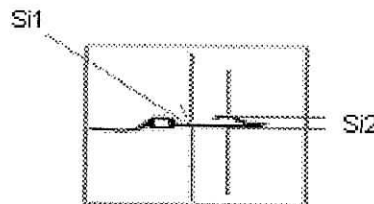
#### **VII-4-5 Segmentation en caractères ( Extraction des caractères ) :**

Les caractères arabes manuscrits sont un exemple de l'écriture cursive, car selon les caractères, on peut avoir des boucles, des branches ou des concavités.

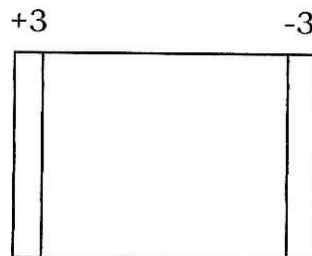
Toutes ces caractéristiques doivent être prises en considération durant le traitement.

On doit définir des paramètres essentiels à la segmentation en caractères :

- ★ La longueur  $S$  : est la distance entre le premier et le dernier pixel d'une même colonne.

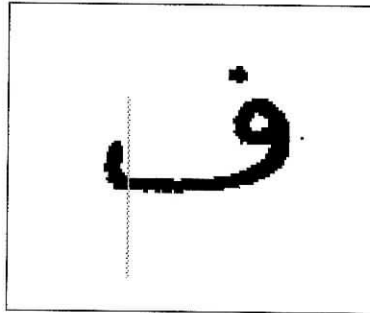
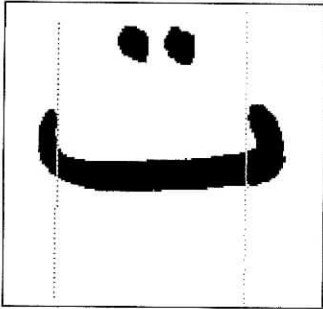


- ★ Le cadre réduit : on diminue chaque partie connexe de 3 pixels colonnes de chaque côté.



le carré réduit

Ces paramètres ont été calculé pour éviter le découpage des caractères comme suit :



En prenant compte de ces conditions , on n'aura pas de découpage, comme illustré sur la figure VII-17 .

*Algorithme de la segmentation en caractères :*

**Début**

**Pour** chaque étiquette **faire**

**Pour** chaque cadre réduit d'une étiquette **faire**

**Pour** chaque colonne **faire**

**Si** (  $S(i) \leq \text{seuil}$  et  $S(i+1) > \text{seuil}$  ) **alors**

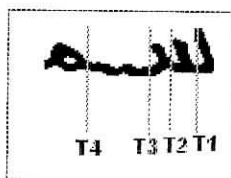
Couper

**Fin**

Le seuil est la valeur minimale la plus redondante de l'histogramme vertical.

Pour avoir un meilleur résultat, il faut rajouter une condition pour la lettre « SIN » .

On considère le SIN comme suit :



La condition repose sur le calcul de la distance entre les T tel que : les trois premières droites sont éliminé car la distance entre elles est petite par rapport à la distance avec T4 .

En appliquant ces conditions on aura :



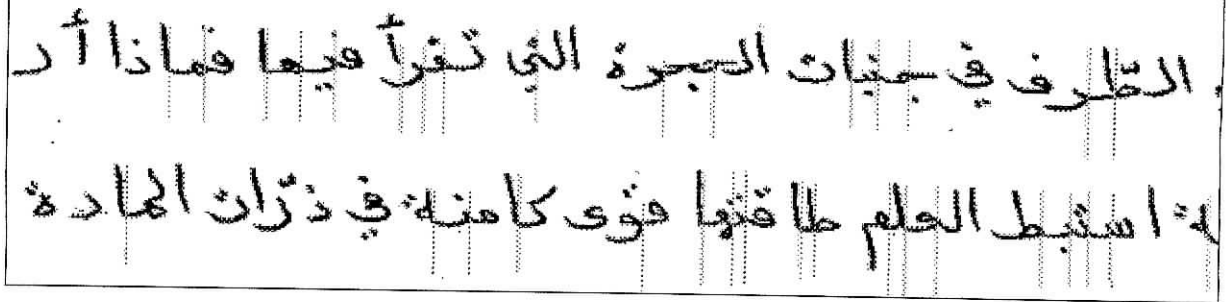


Figure : VII - 17 Segmentation en caractères

Après la segmentation en caractères, on effectue un cadrage de l'image pour localiser chaque caractère séparément.

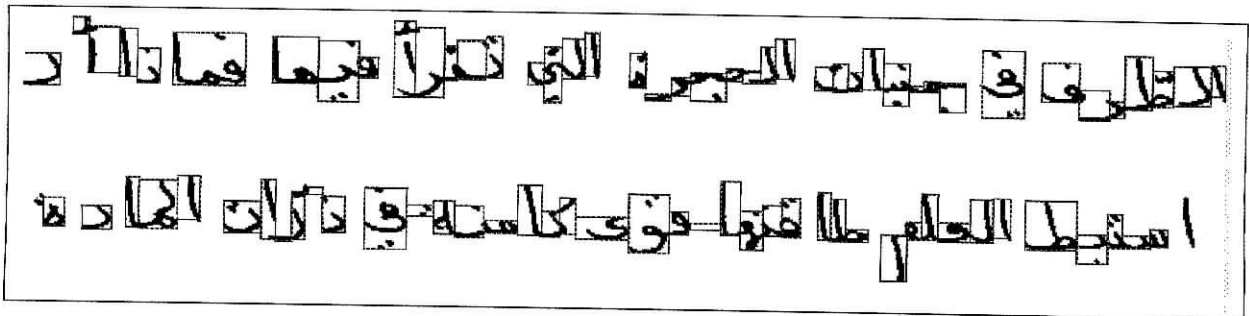


Figure : VII - 18 Cadrage en caractère

### **VII-5 Reconnaissance :**

La reconnaissance est la dernière étape du processus de traitement d'image, elle regroupe essentiellement deux tâches :

- ❖ L'apprentissage.
- ❖ La décision.

Leurs buts sont d'associer chaque caractère récupéré à un modèle de référence, et cela en se basant sur la comparaison des caractéristiques de l'élément à traiter et du modèle de référence.

### ❖ Caractéristiques de l'écriture manuscrite arabe :

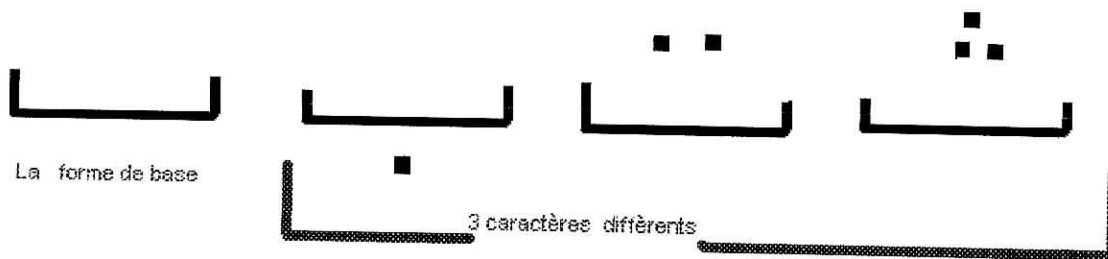
L'écriture arabe s'écrit de droite à gauche de manière cursive et attachée (connexe).

L'alphabet arabe comprend essentiellement 29 caractères de base, qui peuvent avoir 4 formes différentes selon leurs positions dans le mot, début, milieu, fin ou isolé.

L'arabe est une langue sans voyelles mais avec des signes ou des ponctuations qui peuvent être placées au dessus ou en dessous du caractère.

Pour une forme de caractère identique, le son change selon qu'on ait des points ou non et selon leurs positions.

Par exemple :



Certains caractères se chevauchent, donc il est impossible d'encadrer un caractère dans un rectangle sans croiser son successeur. Ce qui rend la séparation des caractères difficile, dans ce cas on parle de ligatures.

### ❖ Description de la méthode :

La méthode utilisée est basée sur les caractéristiques morphologiques du caractère arabe.

Elle consiste en l'extraction des caractéristiques principales, permettant le calcul des différentes classes comme : les boucles et les concavités ainsi que les caractéristiques secondaires telles que la forme et la ponctuation qui différencient les caractères entre eux.

### ❖ Extraction des primitives :

On a choisi de calculer six caractéristiques, qu'on appelle aussi primitives, pour l'étape de la reconnaissance.

**1) La classe :**

Elle nous permet de classer le caractère en prenant compte des caractéristiques morphologiques du tracé du caractère

On la calcule de la manière suivante :

$$\text{Classe} = (n_{\text{boucle}} * 4^0) + (n_{\text{CG}} * 4^1) + (n_{\text{CD}} * 4^2) + (n_{\text{CB}} * 4^3) + (n_{\text{CH}} * 4^4)$$

Tel que :

- ⊗  $n_{\text{boucle}}$  : nombre de boucles
- ⊗  $n_{\text{CG}}$  : nombre de concavités vers la gauche
- ⊗  $n_{\text{CD}}$  : nombre de concavités vers la droite
- ⊗  $n_{\text{CB}}$  : nombre de concavités vers le bas
- ⊗  $n_{\text{CH}}$  : nombre de concavités vers le haut

Une concavité est une partie du tracé du caractère courbée, et qui peut être orientée dans plusieurs sens.

*Principe de calcul des caractéristiques morphologiques :*

La phase de segmentation nous a permis de cadrer chaque caractère de l'image, la première étape consiste à chercher le plus petit rectangle qui peut contenir le caractère comme le montre la figure ci-dessous :



Figure-VII-19: encadrement du caractère.

On procède ensuite à un balayage horizontal et vertical de toute la surface contenant le caractère, pendant les balayages on sauvegarde les positions des lignes et colonnes qui présentent plus de deux paires de transitions (noir-blanc; blanc-noir) car l'existence de plus de deux paires de transitions signifie qu'on a croisé le caractère une fois ou plus donc il y a une concavité ou une boucle.

*Algorithme du balayage horizontal :*

Soit P le nombre de paires de transitions (noir-blanc ; blanc-noir) de la ligne courante.

Soit P1 le nombre de paires de transitions (noir-blanc ; blanc-noir) de la première ligne.

Soit  $x_i$  l'abscisse du centre de chaque paire de transitions.

**DEBUT**

Calculer  $P_1$  ;

**Si**  $P_1 \neq 0$  alors

Sauvegarder  $P_1$  ;

Sauvegarder  $j_1=j$  ; {  $j_1$  est l'ordonné de la nouvelle ligne }

Sauvegarder  $x_i$  ;

**FinSi**

**Pour** ( les lignes suivantes du cadre contenant le caractère) **faire**

Calculer  $P$  ;

**Si**  $(P = P_1)$  **alors**

Continuer ;

**Sinon**

**Si**  $(P_1 \neq 0)$  **alors**

$X_i = x_i$  ;

$Y_i = (j+j_1) / 2$  ;

**FinSi**

Sauvegarder  $P_1 = P$  ;

Sauvegarder  $j_1=j$  ;

Sauvegarder  $x_i$  ;

**FinSinon****Finpour****FIN .**

On utilise le même algorithme pour faire le balayage vertical, seulement on va traiter colonne par colonne et non pas ligne par ligne.

A la fin de chaque balayage on obtient un ensemble de points qui seront utilisés pour détecter les caractéristiques morphologiques du caractère.

Après les deux balayages , chaque concavité ou boucle sera représentée par plusieurs points, ce qui peut fausser la détection, pour cela il faut éliminer les points en plus et ne garder qu'un point représentative (point essentiel) par concavité et /ou boucle,

Pour cela on relie chaque deux points par une droite, si la droite ne coupe pas le tracé du caractère (les deux points représentent la même concavité ou boucle), on élimine l'un deux, sinon on les laisse.

A partir de chaque point essentiel on tire une droite dans les huit sens, et on voit quelles sont les droites qui coupent le tracé du caractère et celles qui ne le coupent pas, a partir de ces huit informations on identifie ce que représente le point (boucle, concavité vers le haut, concavité vers le bas, concavité vers la droite ou concavité vers la gauche).

## 2) Positions des points :

La position des points pour un caractère arabe détermine le son de ce dernier, c'est une caractéristique très importante mais difficile à extraire.

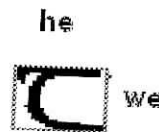
Pour déterminer cette primitive, on effectue un étiquetage sur chaque cadre contenant un caractère.

On récupère le nombre d'étiquettes présentes dans le cadre et la taille de chacun.

Pour déterminer les points, on définit une variable Pdp, qui prend les valeurs suivantes :

- Si le nombre d'étiquettes = 1  $\Rightarrow$  il n'y a pas de ponctuation  $\Rightarrow$  pdp = 0
- Si le nombre d'étiquettes est  $>1$  :

Pour une étiquette e on définit  $h_e$  et  $w_e$  tel que :



FigureVII-20 : longueur et largeur d'une étiquette

On compare les tailles des étiquettes. La plus grande sera cadrée avec la couleur verte, et le reste des étiquettes en rouge.

Exemple :



Figure VII-21 : Différentes positions possibles

- Si l'étiquette rouge est au dessus de l'étiquette verte  $\Rightarrow$  pdp=1
- Si l'étiquette rouge est à l'intérieur de l'étiquette verte  $\Rightarrow$  pdp=2
- Si l'étiquette rouge est en dessous de l'étiquette verte  $\Rightarrow$  pdp=3

### 3) Le nombre de points :

Après avoir déterminé la position des points, on définit la variable nombre de point  $ndp$ .

Si  $Pdp = 0 \Rightarrow ndp = 0$ .



FigureVII-22 : Mise en évidence de la ponctuation

Si  $pdp \neq 0$  donc existence d'une ponctuation :

- S'il y a 3 étiquettes rouges  $\Rightarrow ndp = 3$  points
- S'il y a 1 étiquette rouge tel que :  $we/he \geq 1.4 \Rightarrow ndp = 2$
- S'il y a 2 étiquettes rouges et pour l'une d'elles  $we/he \geq 1.4 \Rightarrow ndp = 3$  sinon  $ndp = 2$ .
- S'il y a 1 étiquette rouge avec  $0.8 < we/he < 1.2$  : on calcule le nombre de transitions verticales «  $ntv$  »
  - Si  $ntv \leq 1 \Rightarrow ndp = 1 \Rightarrow 1$  point
  - Si  $ntv > 1 \Rightarrow ndp = 4 \Rightarrow$  une hamza

Donc la variable  $ndp$  peut prendre les valeurs suivantes :

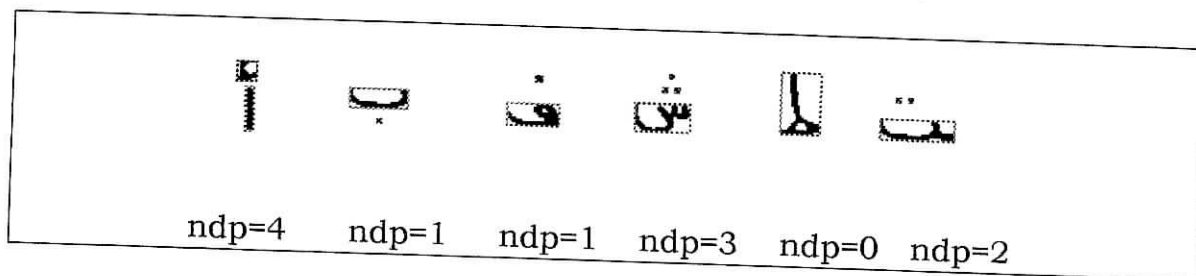


Figure VII-23 : Différentes possibilités pour  $ndp$

$Ndp = 0$  : il n'y a pas de ponctuation.

$Ndp = 1$  : il y a un point .

$Ndp = 2$  : il y a deux points .

$Ndp = 3$  : il y a trois points.

$Ndp = 4$  : il y a une hamza.

**4) La forme :**

La forme du caractère peut être carrée, debout ou allongée, pour calculer cette primitive, on calcule le rapport :

$$\text{r} = w/h$$

tel que :  $w$  = largeur du caractère.  
 $h$  = hauteur du caractère.

On définit la variable forme de la manière suivante :

forme=1 si la forme est allongée .

forme=2 si la forme est carrée.

forme=3 si la forme est debout.

On définit les constantes seuil1 et seuil 2 tel que : seuil1=0.8  
 Seuil2=1.2

**Si** ( $r > \text{seuil2}$ )  $\Rightarrow$  forme=1 ;

**Si** ( $r < \text{seuil1}$ )  $\Rightarrow$  forme=3 ;

**Si** ( $r \leq \text{seuil2}$  et  $r \geq \text{seuil1}$ )  $\Rightarrow$  forme=2 ;

Le choix des seuils, seuil1 et seuil2 a été fait après plusieurs essais.

**5) Le taux de remplissage :**

A partir des cadres obtenus après l'étiquetage en caractère, on peut calculer le taux de remplissage des quatre coins pour déterminer la répartition des pixels du caractère.

On définit la variable corvar et le taux de remplissage ( $S_i$ ) pour chaque coin.

Mais cette variable dépend de la primitive précédente : la forme tel que :

❖ si la forme est carrée ou allongée le coin  $S_i$  aura comme longueur =  $w/4$  donc une surface =  $w^2/16$ .

❖ Sinon la longueur =  $h/4$  donc une surface =  $h^2/16$

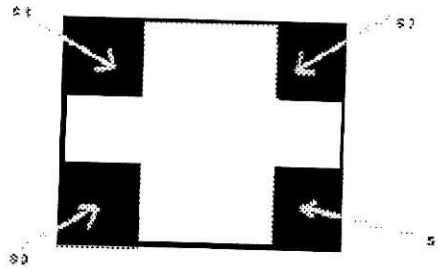
Pour chaque coin du carré réduit on aura :

$S_{si}$  = nombre de points noirs/surface du coin

Si  $S_{si} > 0.5$  donc  $S_{i=1}$

Sinon  $S_i = 0$

$$\text{corvar} = \sum_{i=1}^4 (S_i) * 2^{(i-1)} = S_1 + 2 * S_2 + 4 * S_3 + 8 * S_4$$



FigureVII-24 taux de remplissage

## VII-6 Reconnaissance par réseaux de neurones :

L'une des propriétés les plus importantes des réseaux de neurones est leur capacité de généralisation, qui leur permet de donner des réponses satisfaisantes aux exemples qui ne font pas partie de leur ensemble d'apprentissage. Pour notre système nous avons utilisé un réseau multicouche perceptron qui est le mieux adapté pour la classification.

### VII-6-1 Architecture du réseau de neurones :

Le réseau utilisé est un réseau perceptron multicouche composé de:

- 1) Une couche d'entrée à 21 neurones, car les caractéristiques de chaque caractère est codées sur 21 bits.
- 2) Une couche cachée de 21 neurones ( nombre choisi après plusieurs essais ).
- 3) Une couche de sortie à 6 neurones, qui représente le code de chaque caractère en binaire, comme illustré par le tableau suivant :

Code du caractère	Caractère Arabe	Code ASCII	Nom
000000		32	ESPACE
000001	آ	194	LETTRE ARABE ALIF MADDA EN CHEF
000010	أ	195	LETTRE ARABE ALIF HAMZA EN CHEF
000011	ؤ	196	LETTRE ARABE WAW HAMZA EN CHEF



000100	ا	197	LETTRE ARABE ALIF HAMZA SOUSCRIT
000101	أ	198	LETTRE ARABE YA' HAMZA EN CHEF
000110	ب	199	LETTRE ARABE ALIF
000111	ب	200	LETTRE ARABE BA'
001000	ة	201	LETTRE ARABE TA' MARBOUTA
001001	ت	202	LETTRE ARABE TA'
001010	ث	203	LETTRE ARABE THA'
001011	ج	204	LETTRE ARABE DJÎM
001100	ح	205	LETTRE ARABE HA'
001101	خ	206	LETTRE ARABE KHA'
001110	د	207	LETTRE ARABE DAL
001111	ذ	208	LETTRE ARABE DHAL
010000	ر	209	LETTRE ARABE RA'
010001	ز	210	LETTRE ARABE ZAY
010010	س	211	LETTRE ARABE SÎN
010011	ش	212	LETTRE ARABE CHÎN
010100	ص	213	LETTRE ARABE SAD
010101	ض	214	LETTRE ARABE DAD
010110	ط	216	LETTRE ARABE TA'
010111	ظ	217	LETTRE ARABE ZA'
011000	ع	218	LETTRE ARABE 'AYN
011001	غ	219	GHAYN
011010	ف	221	LETTRE ARABE FA'
011011	ق	222	LETTRE ARABE QAF
011100	ك	223	LETTRE ARABE KAF
011101	ل	225	LETTRE ARABE LAM
011110	م	227	LETTRE ARABE MÎM
011111	ن	228	LETTRE ARABE NOÛN
100000	هـ	229	LETTRE ARABE HA'
100001	و	230	LETTRE ARABE WAW
100010	ى	236	ALIF MAKSOURA
100011	ي	237	LETTRE ARABE YA'

Figure VII-25 codes ASCII et de sortie de chaque caractère

Les 21 neurones de la couche d'entrée ont été choisis, car les caractéristiques des caractères sont codés sur 21 bits tel que :

- ★ la classe sur dix bits,
- ★ la forme sur deux bits,
- ★ corvar sur quatre bits,
- ★ la position du point sur deux bits
- ★ et le nombre de points sur trois bits

Le réseau de neurones est représenté comme suit :

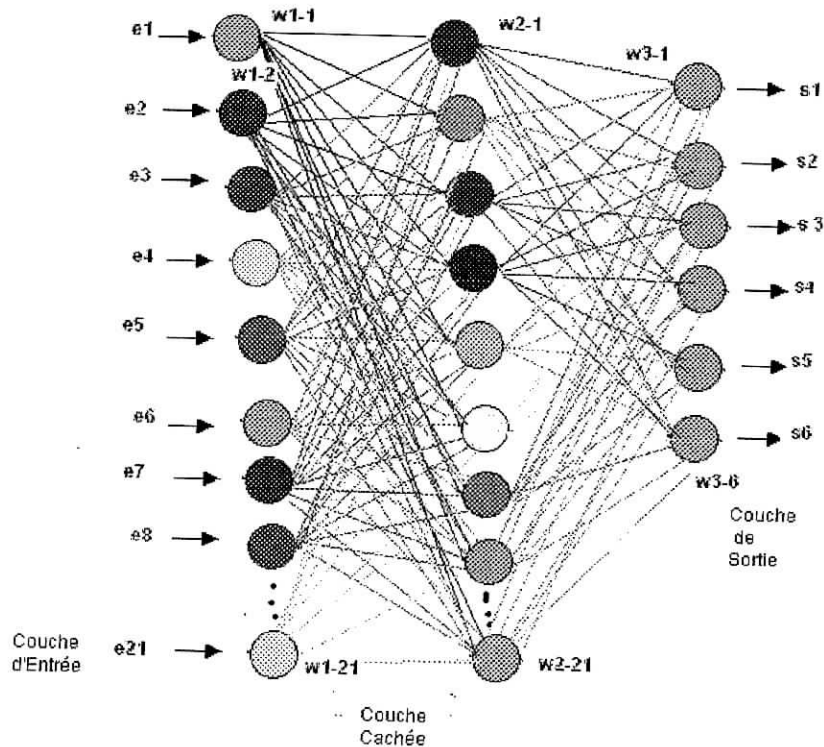


Figure VII-26 représentation du réseau utilisé

Après avoir déterminé le nombre de neurones nécessaire à notre système, on associe à chaque couche sa fonction :

- 1) Pour la fonction d'entrée : une fonction sigmoïde.
- 2) Pour la fonction de seuillage : une fonction sigmoïde.
- 3) Pour la fonction de sortie : la fonction identité.

Ces fonctions ont été choisies après plusieurs simulations sous MATLAB.

**VII-6-2 Apprentissage du réseau :**

Après avoir constitué notre réseau, la phase d'apprentissage peut commencer . Elle consiste à appliquer l'algorithme de rétro propagation, pour adapter les poids du réseau et rechercher les sorties les plus proches de celles souhaitées.

Nous avons utilisé MATLAB 6.5 pour ce calcul, et nous avons sauvegardé ces résultats dans un fichier pour l'implémenter dans notre application.

L'étape d'apprentissage s'effectue comme suit :

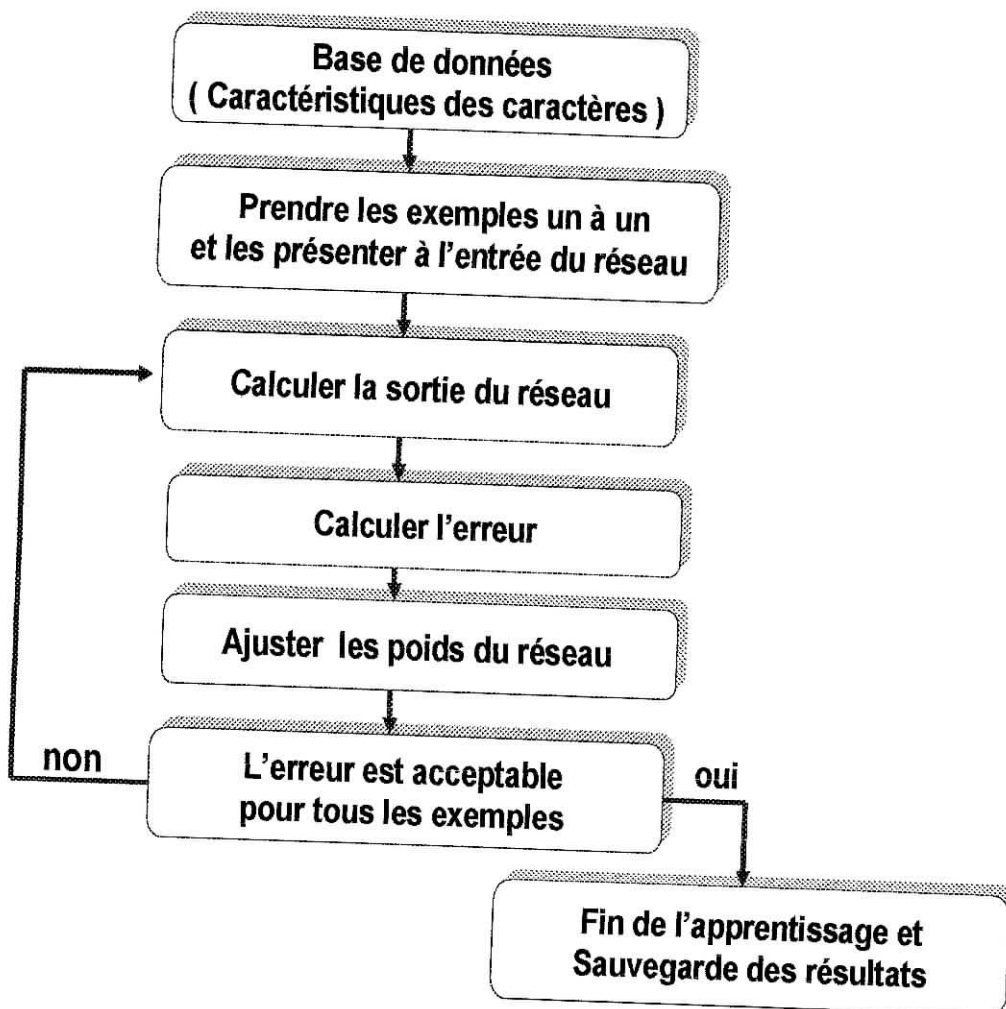


Figure VII-27 étapes d'apprentissage

Les méthodes d'extraction de caractères et de reconnaissance par réseaux de neurones ont donné des résultats acceptables,

tels que : on a un taux de segmentation égal à 86,72 %  
et un taux de reconnaissance égal à 70,45 %

# Chapitre VIII

## Description du Logiciel

## **VIII -1 Introduction :**

Dans ce chapitre nous allons présenter le logiciel que nous avons réalisé comme application à la reconnaissance de texte manuscrit arabe.

L'expansion de l'environnement WINDOWS a offert à l'utilisateur un ensemble d'outils performant et facile d'emploi.

Notre logiciel a été réalisé grâce à l'outil de programmation Builder C++ version 5 sous environnement Windows XP. Le choix du langage est justifié par la souplesse et la simplicité avec laquelle celui-là supporte la programmation orientée objet, et par le besoin d'un environnement de programmation évolué, permettant une description rapide des algorithmes en POO ( Programmation Orientée Objet).

C'est une application MDI (Multiple Document Interface) offrant ainsi à l'utilisateur la possibilité de visualiser plusieurs images en même temps. La présence d'un menu bien aménagé, et d'une barre à outils, facilite l'utilisation du logiciel.

Ce logiciel réalise essentiellement deux fonctions :

- Le prétraitement de chaque image donc document scanné
- La segmentation et la reconnaissance du texte.

Dans ce qui suit nous présenterons la fenêtre principale de l'application et les menus secondaires avec leurs différentes opérations.

## VIII-2 Configuration de l'application :

L'interface du logiciel est représentée ci-dessous par la figure suivante :

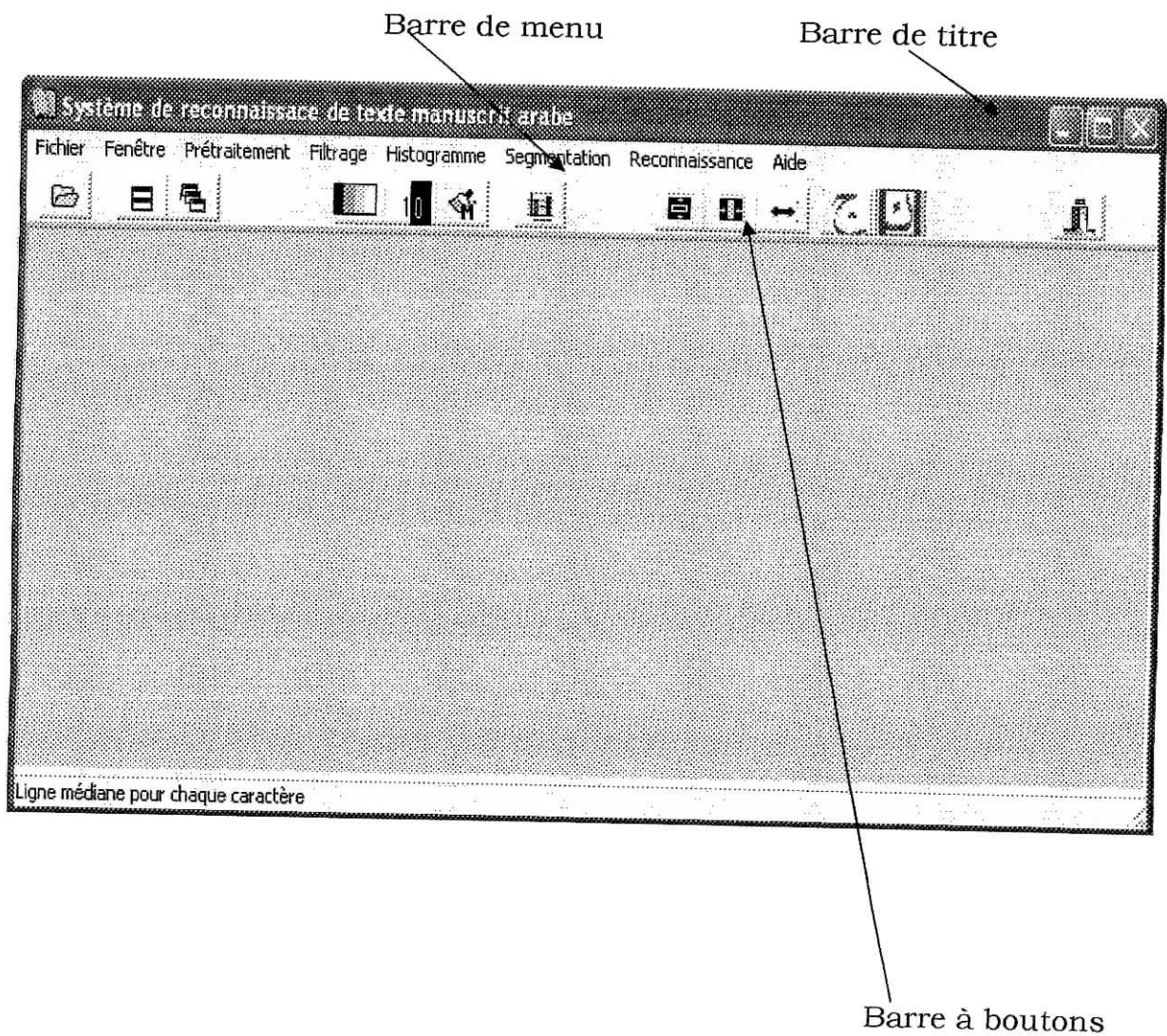
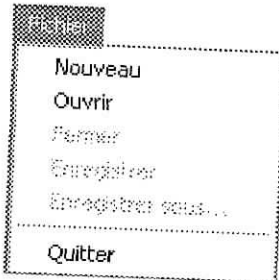


Figure VIII-1 L'interface du logiciel

**VIII-2-1 La barre des menus :**

C'est une barre contenant des articles qu'on peut sélectionner à partir de la souris. L'activation d'un article fait appel soit à une fonction, soit à une boîte de dialogue.

**VIII-2-1-1 Menu Fichier**



**Commande ouvrir**

A l'exécution de cette commande une boîte de sélection apparaît permettant à l'utilisateur de choisir le nom du fichier image, d'extension (\*.bmp), à ouvrir.

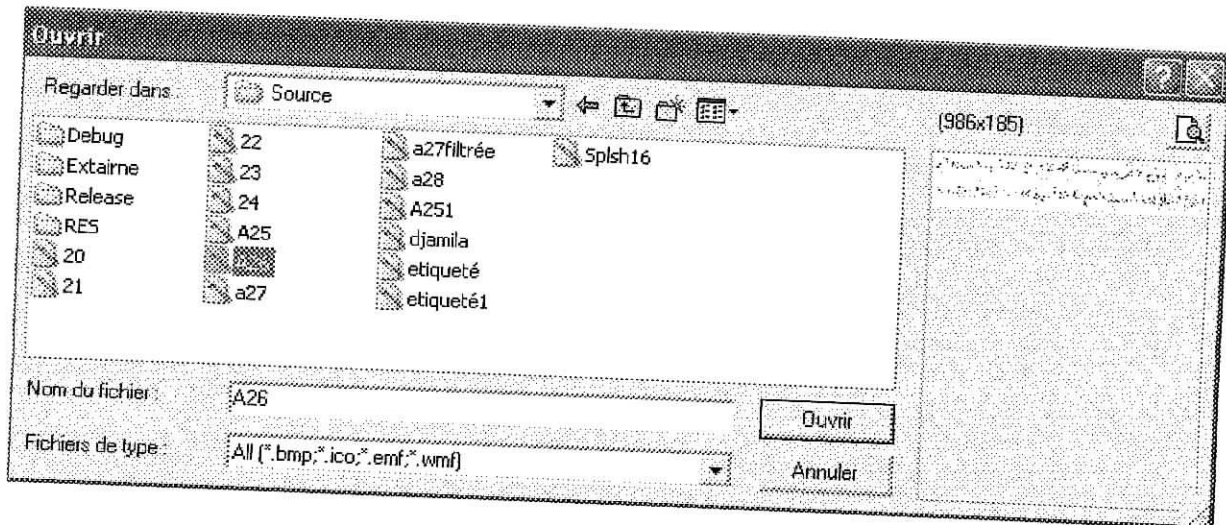


Figure VIII-2 Boite de dialogue pour ouvrir

● **Commande Enregistrer Sous :**

A l'exécution de cette commande une boîte de sélection apparaît permettant à l'utilisateur de sauvegarder l'image qui se trouve dans la fenêtre active, sous un autre nom de fichier.

● **Commande Enregistrer :**

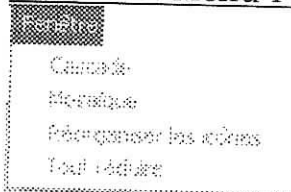


A l'exécution de cette commande une boîte de sélection apparaît permettant à l'utilisateur de sauvegarder l'image qui se trouve dans la fenêtre active.

● **Commande Quitter :**

Cette commande permet de mettre fin à l'application.

VIII-2-1-2 Menu Fenêtre :



Grâce à ce menu, on peut choisir la manière d'afficher nos différentes fenêtres en cascade, en mosaïque ou les réduire.

VIII-2-1-3 Menu Prétraitement :



● **Commande Niveaux de gris :**

Après avoir ouvert notre image , elle peut être en couleur, on lui applique cette commande pour l'avoir en niveau de gris.

● **Commande Binarisation :**

La binarisation est l'étape qui consiste à transformer l'image en seulement deux niveaux de gris le blanc et le noir et ceci se fait par un calcul automatique du seuil ou bien manuellement. Le choix de la méthode de calcul se fait grâce à la boîte de dialogue présentée ci-dessous :

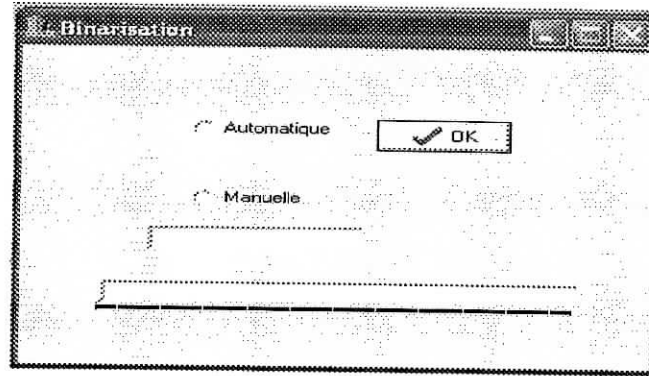
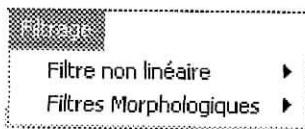
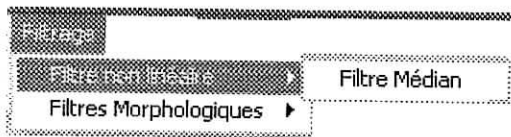


Figure VIII-3 boîte de dialogue pour la binarisation

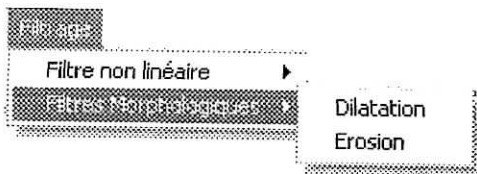
#### VIII-2-1-4 Menu Filtrage :



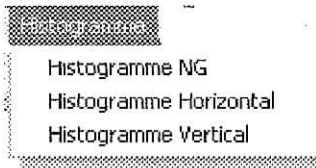
Ce menu se compose de deux sous menus déroulants : le filtre non linéaire et les filtres morphologiques.



Nous avons choisi le filtre médian comme filtre non linéaire pour réduire le bruit présent sur l'image.



Pour les filtres morphologiques, ils permettent de grossir ou d'affiner l'écriture selon nos besoins.

VIII-2-1-5 Menu Histogramme :

Dans ce menu, nous avons le choix entre les différents types d'histogrammes, soit en niveaux de gris, horizontal ou vertical.

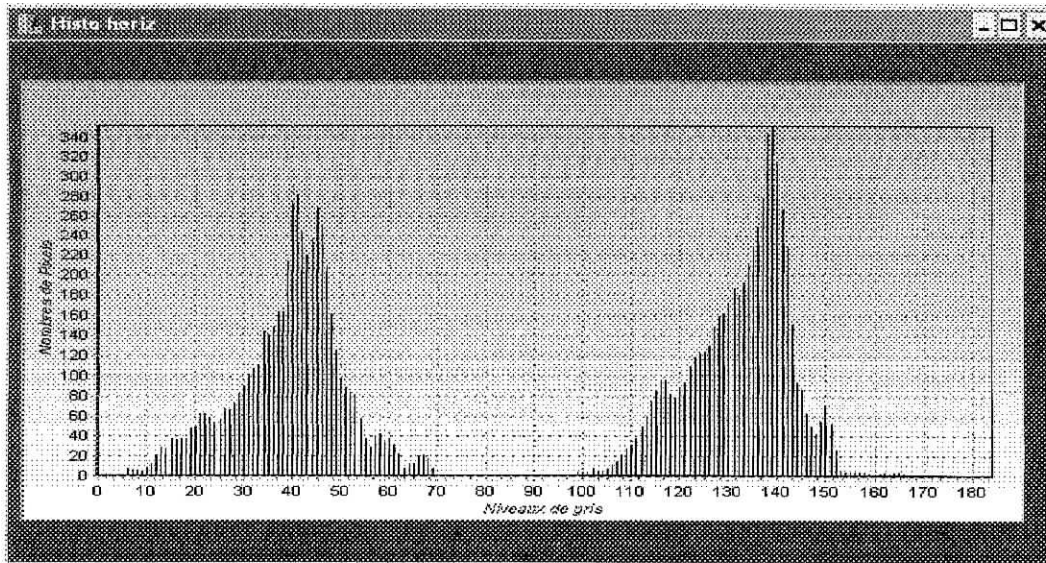
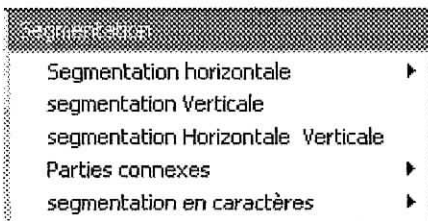


Figure VIII-4 Représentation de l'histogramme horizontal

Cet outil détermine le nombre de lignes texte, de notre document

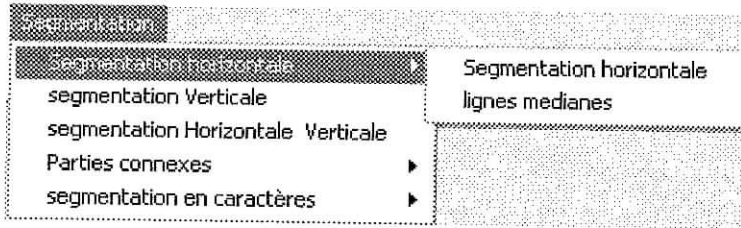
VIII-2-1-6 Menu segmentation :

Le menu segmentation est représenté comme suit :



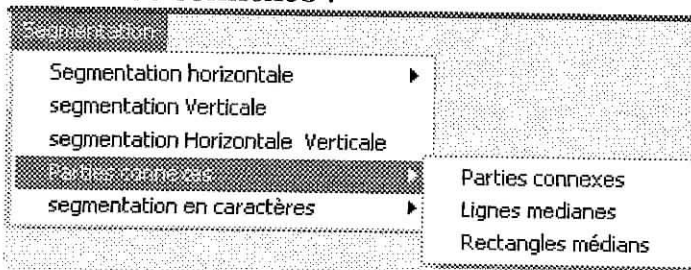
On a le choix entre une segmentation horizontale verticale et une segmentation en parties connexes.

● Segmentation horizontale :

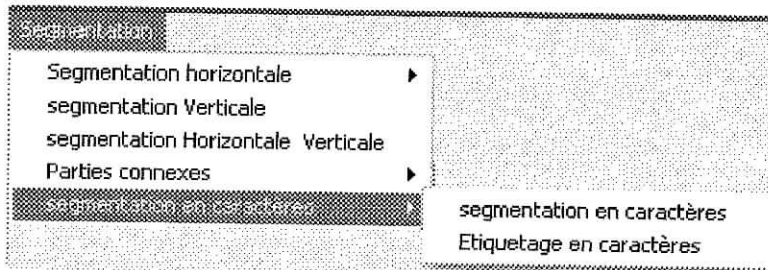


On peut déterminer aussi les lignes médianes.

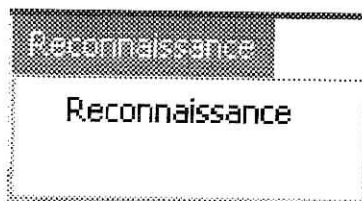
● Parties connexes :



● Segmentation en caractères :

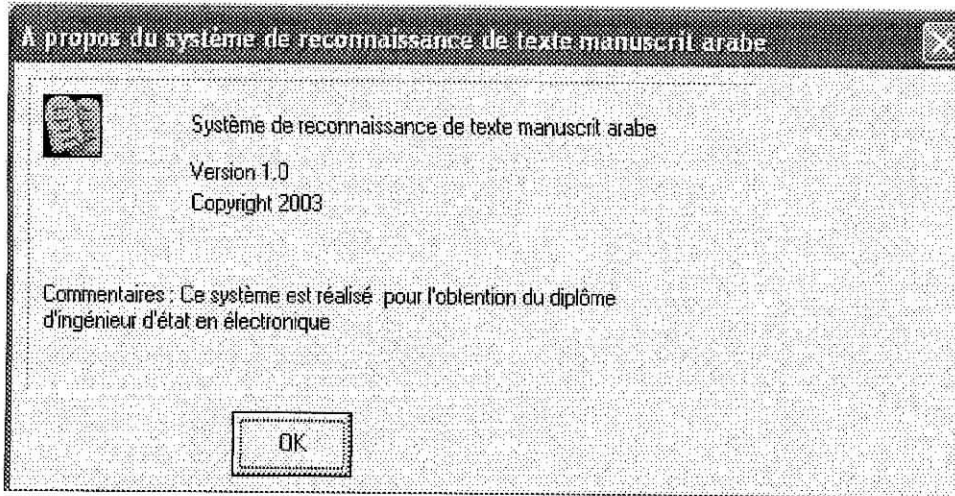


VIII-2-1-7 Menu Reconnaissance :



Lorsqu'on sélectionne cet onglet, on lance la reconnaissance pour notre texte.

VIII-2-1-8 Menu Aide :



L'a propos donne des informations sur le logiciel, et nous avons aussi l'onglet « L'Aide » qui permet d'avoir un support pour l'utilisation de l'application.

**VIII-2-2 Fenêtre enfant :**

Cette application est basée sur le modèle MDI ; ceci veut dire que chaque traitement se fait par rapport à l'image d'origine et s'affiche dans une nouvelle fenêtre enfant. Pour appliquer tous les traitements sur la même image on a rajouté un *pop menu* : qui est un menu déroulant détaché de la barre de menu avec tous les onglets possibles comme illustré ci-dessous :

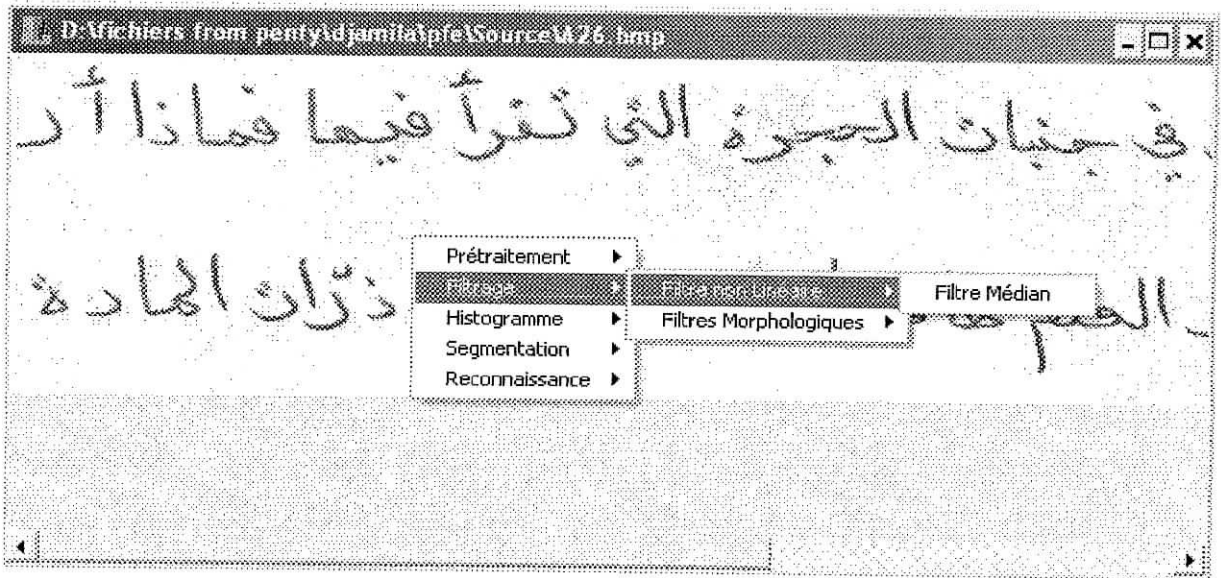
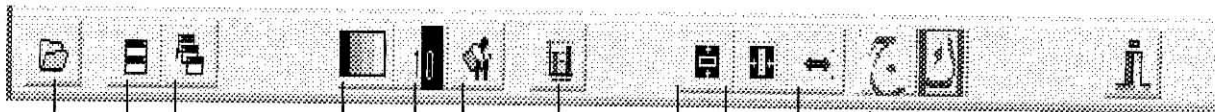


Figure VIII-5 Exemple de fenêtre enfant

### VIII-2-3 La barre de boutons:



- Quitter l'application.
- Etiquetage en caractères.
- Segmentation en caractères .
- Ligne médiane.
- Segmentation verticale.
- Segmentation Horizontale.
- Histogramme .

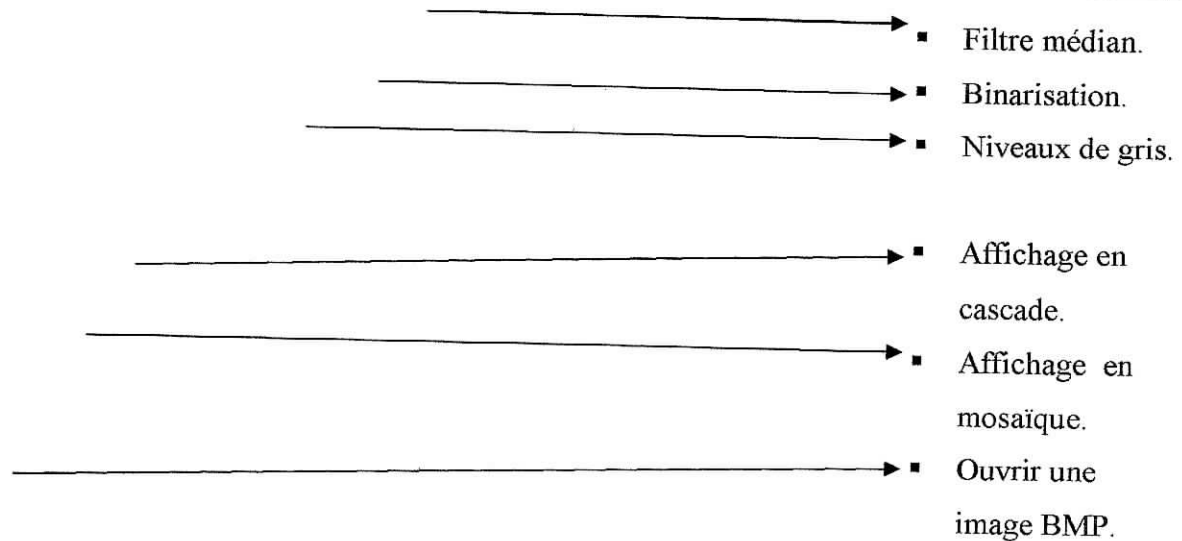


Figure VIII-6 La barre de boutons

### **VIII-2-4 La barre d'état :**

C'est une barre qui visualise le nom du traitement en cours : aide des commandes.

Segmentation

Figure VIII-7 la barre d'état

### **VIII-3 Résultats :**

Nous allons présentés quelques représentations des résultats obtenus à l'aide de notre logiciel :

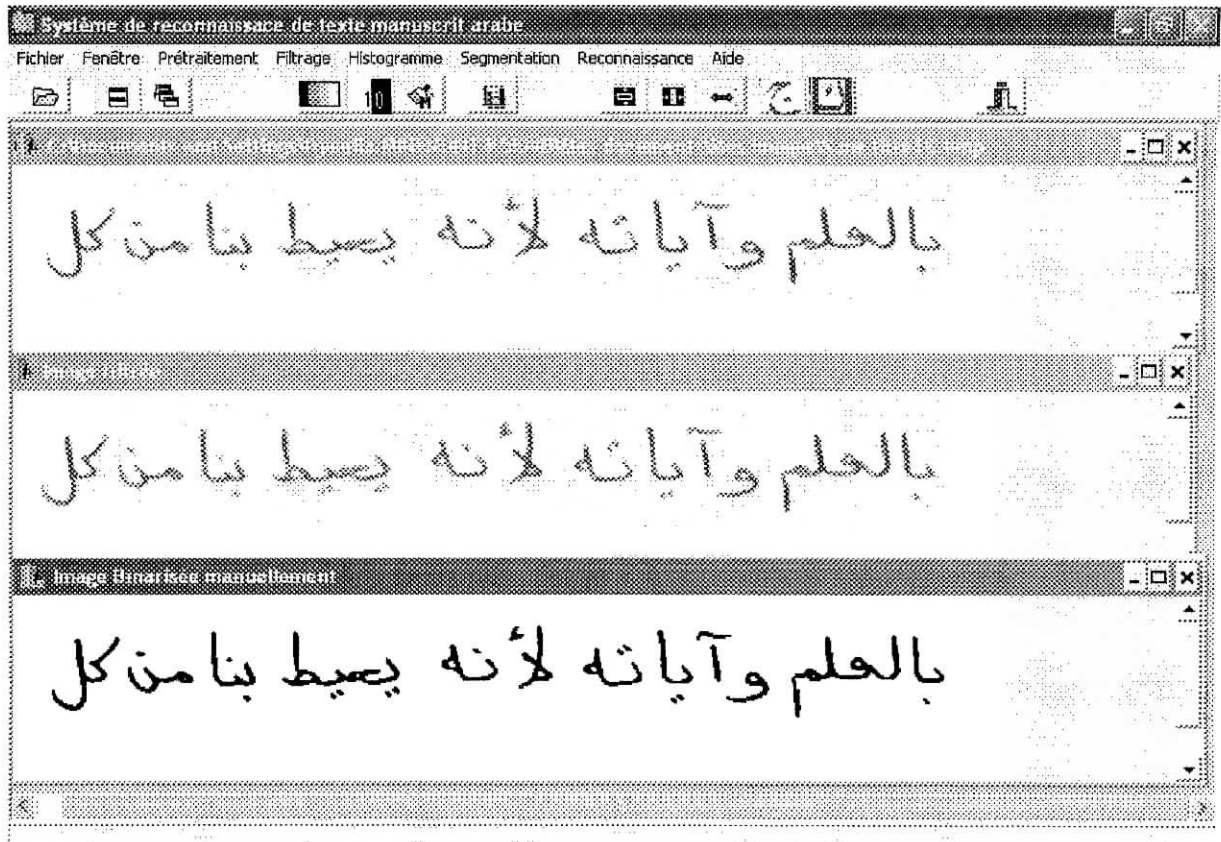


Figure VIII-8 Représentation de l'étape de prétraitement



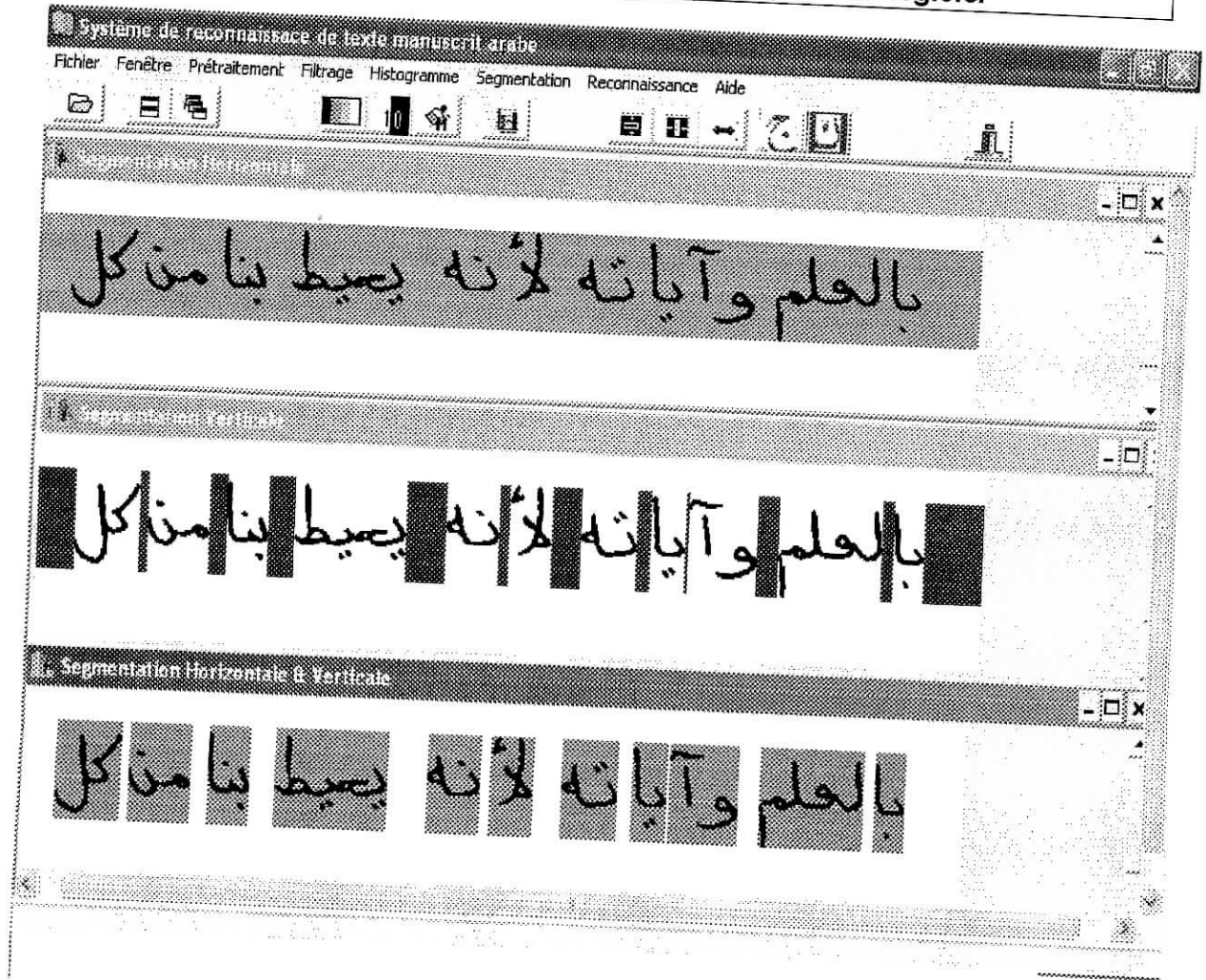


Figure VIII-9 Segmentation horizontale et verticale

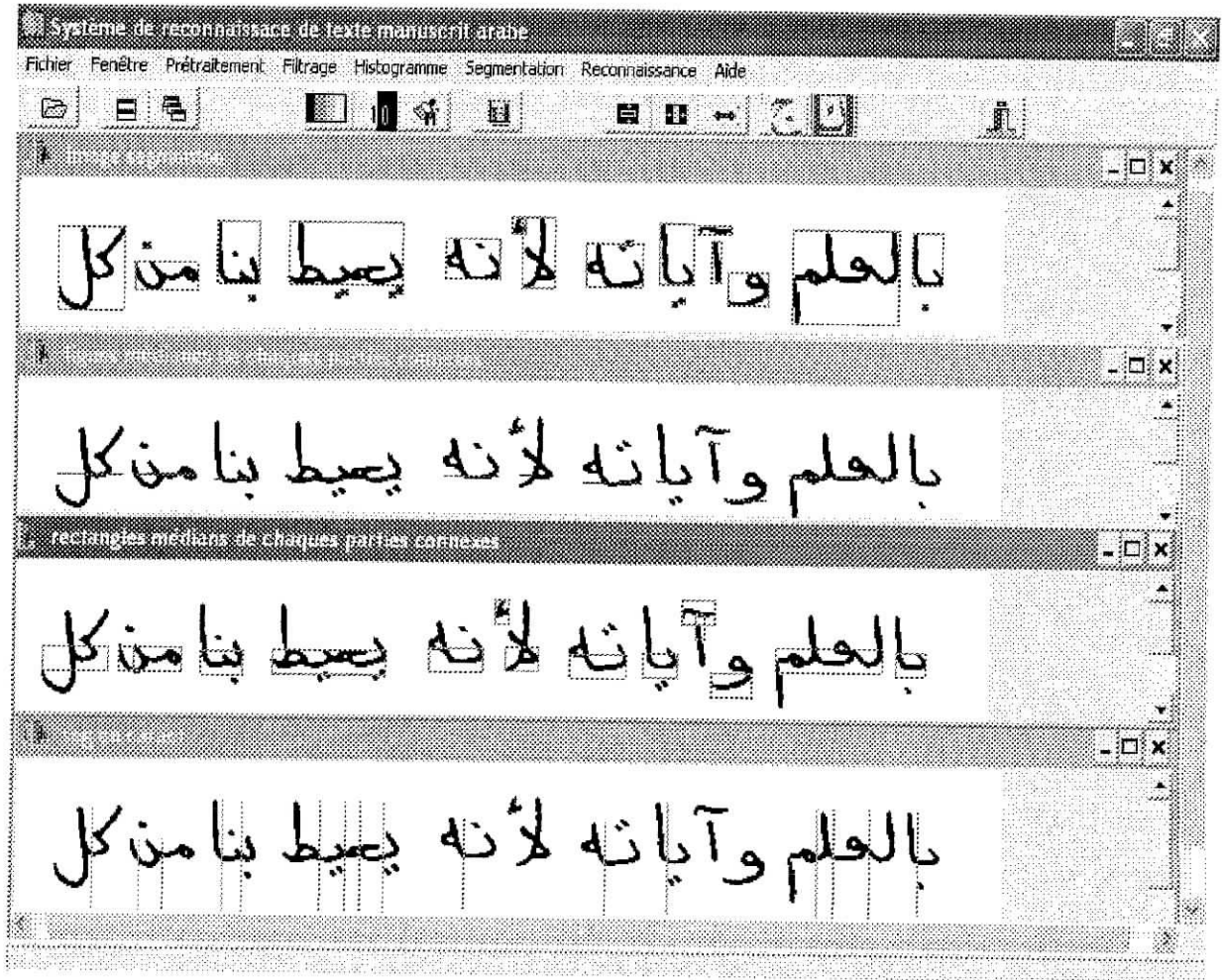


Figure VIII-10 Segmentation en partie connexe et en caractère

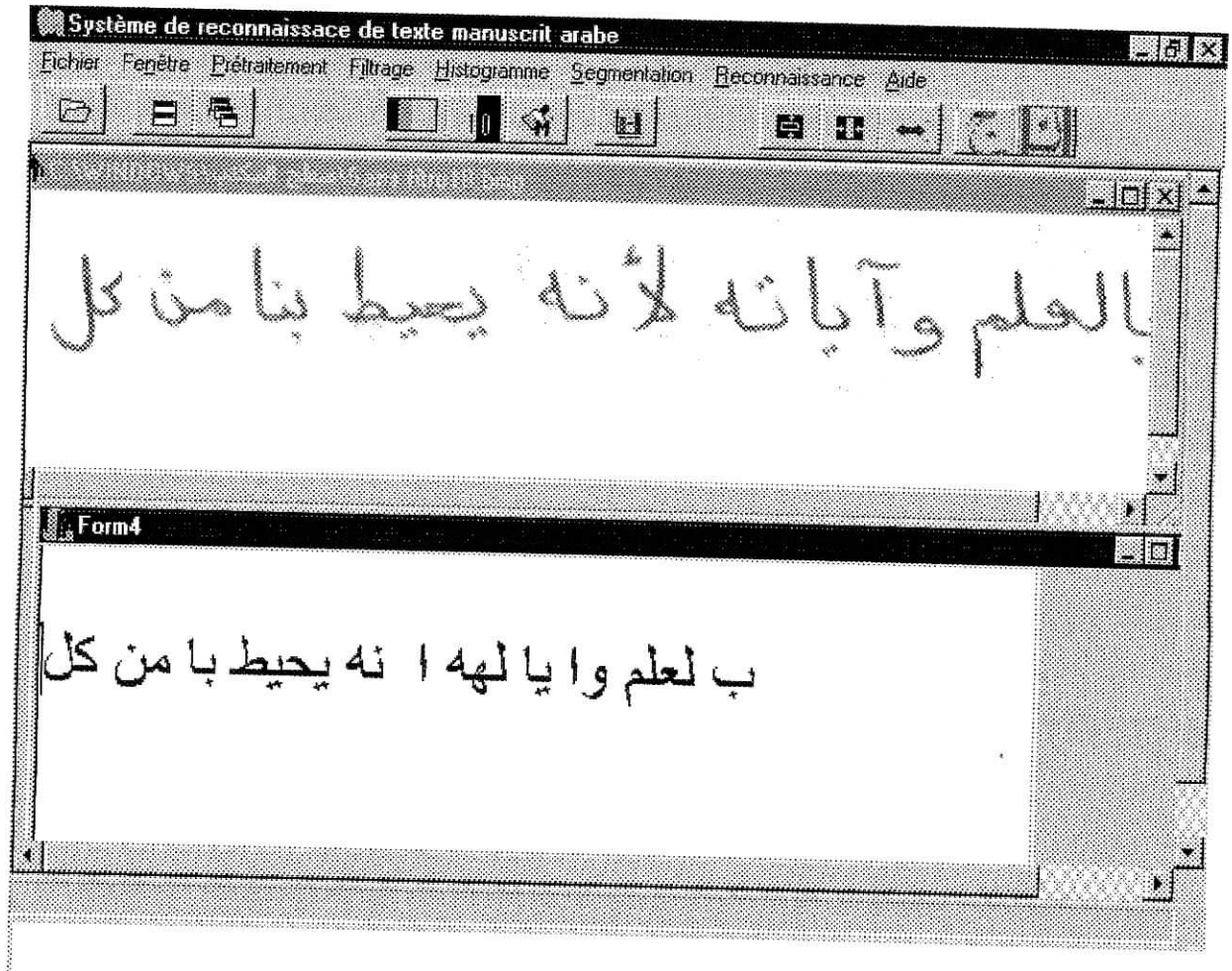


Figure VIII-11 reconnaissance du texte

# CONCLUSION GENERALE

Le traitement automatique de la langue arabe est un secteur qui comporte des enjeux de type industriel, économique, scientifique et technique.

Le travail que nous avons réalisé s'intitule : système de reconnaissance de texte manuscrit arabe, qui se trouve être un sujet assez complexe, vu la difficulté à segmenter l'écriture arabe en général et arabe manuscrite en particulier.

Nous avons découpé le problème en plusieurs modules.

Tout d'abord, nous avons scanné l'image, et nous lui avons appliqué un filtre médian pour minimiser le bruit et améliorer sa qualité, ensuite vient la binarisation pour ne laisser que deux niveaux distincts.

Le second module est la segmentation, c'est l'étape primordiale pour notre système. Nous avons procédé à une segmentation horizontale puis verticale pour localiser les blocs de mots mais cette méthode, plus appropriée à l'imprimé, a causé des problèmes vus les chevauchements des mots manuscrits.

Alors la solution adoptée a été de procéder à une segmentation en régions qu'on appelle « Etiquetage des composantes connexes » ; cette méthode permet de déterminer les parties connexes des mots, des points, des caractères isolés et de la hamza.

Nous traitons les mots par une autre méthode de segmentation qui permet de diviser les mots en caractères, en localisant la ligne d'écriture par un algorithme que nous avons développé. Après cette opération de segmentation, on effectue un cadrage de chaque caractère avec sa ponctuation.

Pour la reconnaissance, nous avons utilisé un réseau de neurones multicouches, son apprentissage a été fait sous MATLAB 6.5 et implémenté sous C++ BUILDER5.

Pour améliorer notre travail, et comme perspectives envisageables, nous proposons une révision de l'algorithme de segmentation développé permettant ainsi d'améliorer le taux de segmentation et en même temps celui de la reconnaissance.

En reconnaissance, il serait souhaitable de tester d'autres réseaux de neurones et en particulier celui de Kohonen ou encore utiliser la logique neuro-floue.

# ANNEXES

**Annexe 1 :****Algorithme de rétro-propagation du gradient (règle de Windrow Hoff) :**

Soit un réseau à  $n$  neurones recevant des vecteurs  $x$  à  $p$  composantes. Les  $p$  entrées  $x_k$  du réseau sont distribuées sur tous les neurones. La sortie du neurone  $i$  vaut :

$$y_i = f(P_i) = f\left(\sum_{j=1}^P w_{ij} x_j\right)$$

avec  $P_i$  = le potentiel du neurone ,  $C$  = combinaison linéaire des entrées pondérées par  $w_{ij}$  .

On veut obtenir  $y_d$  la sortie désirée ( c'est le but de l'apprentissage supervisé, la sortie est connue ).

On pourra calculer l'erreur quadratique :  $E = \frac{1}{2} \sum_{j=1}^n (y_i - y_{dj})^2$

Avec le calcul du gradient, on obtient :

$$\frac{\partial E}{\partial w_{ik}} = \sum_{j=1}^n (y_i - y_{dj}) \frac{\partial (y_i - y_{dj})}{\partial w_{ik}}$$

comme  $y_i = f(w_{ij})$

$$\text{et si } i = j \Rightarrow \frac{\partial (y_i - y_j)}{\partial w_{ik}} \neq 0$$

$$\Rightarrow \frac{\partial E}{\partial w_{ik}} = (y_i - y_{di}) \frac{\partial y_j}{\partial w_{ik}} = (y_i - y_{di}) x_k f'(P_i)$$

le gradient de l'erreur quadratique :

$$\frac{\partial E}{\partial w_{ik}} = \delta_i x_k \quad \delta_i = (y_i - y_{di}) f'(P_i)$$

$$\Delta w_{ik} = -a \delta_i x_k$$

$a$  = taux d'apprentissage

**Annexe 2 :****Structure d'un fichier BitMap**

Ce document décrit la façon dont est enregistré un fichier BitMap au format Windows.

Un BitMap est constitué de 4 parties :

◆ **LE HEADER :**

Il indique les caractéristiques principales du fichier : son type (BitMap), sa taille, et l'emplacement de l'image dans le fichier.

Position	Type	Description
00h	DB "BM"	Signature : BM indique que l'on se trouve en présence d'un fichier bitmap
02h	DD ?	Taille du fichier en octets
06h	DW 0	Réservé
08h	DW 0	Réservé
0Ah	DD ?	Offset de l'image

◆ **LA ZONE BITMAP INFO :**

Elle définit les caractéristiques essentielles de l'image, c'est pourquoi on la distingue du header, mais elle suit directement celui-ci.

Position	Type	Description
0Eh	DD ?	Taille de la zone BitmapInfo
12h	DD ?	Largeur de l'image en pixels
16h	DD ?	Hauteur de l'image en pixels
1Ah	DW 1	Nombre de plans (1)
1Ch	DW 1 4 8 24	Bits par pixel (1=monochrome, 4=16 couleurs, 8=256 couleurs, 24=16 M de couleurs)
1Eh	DD ?	Format de compression de l'image : 0 : non compressée 1 : compressée à 8 bits par pixel (RLE8) 2 : compressée à 4 bits par pixel (RLE4)
22h	DD ?	Taille de l'image en octets



26h	DD ?	Résolution horizontale en pixels par mètre
2Ah	DD ?	Résolution verticale en pixels par mètre
2Eh	DD ?	Nombre de couleurs utilisées. 0 = maximum possible. Si l'image est 24 bits, ce champ spécifie la taille de la palette de couleurs de référence utilisée pour optimiser les palettes Windows
32h	DD ?	Nombre d'index de couleurs importants pour l'affichage. 0 <=> toutes les couleurs importantes

### ◆ LA PALETTE :

C'est un tableau contenant la liste des couleurs pour le mode 8 bits. Il suit immédiatement la zone Bitmap Info, et n'est présent qu'en mode 8 bits. Une couleur est codée sur 4 octets : R:G:B:Réservé. La taille de la palette est donc de 1024 octets.

### ◆ L'IMAGE :

Enfin la voilà ! Elle suit normalement directement la zone BitMap Info (ou la palette, s'il y en a une), mais il est préférable d'y accéder via l'offset défini dans le header. Ici, pas de difficulté spécifique, le codage des pixels est similaire à celui dans la mémoire vidéo, il suffit donc de faire un transfert du fichier vers l'écran (en n'oubliant pas de revenir à la ligne, tout de même !)

**Annexe 3 :****Glossaire :**

- ❖ Algorithme du gradient : algorithme probabiliste permettant de déterminer le minimum d'une fonction.
- ❖ Architecture : structure d'un réseau de neurones, définition du nombre de neurones et de leurs connexions.
- ❖ Apprentissage : phénomène de modification temporelle des poids des connexions du réseau de façon à obtenir un fonctionnement particulier du réseau, un type de sortie prédéfinie.
- ❖ Apprentissage non supervisé : on ne connaît pas a priori la sortie correspondant à des entrées données.
- ❖ Apprentissage supervisé : pour une entrée donnée, la sortie est définie au préalable.
- ❖ Auto organisation : (ou respect du voisinage topologique) : qualité du réseau telle qu'à des entrées voisines correspondent des sorties voisines.
- ❖ Construction du réseau : choix des poids de connexion de tel sorte à réaliser l'apprentissage souhaité.
- ❖ Fonction d'activation : c'est la fonction qui lie la réponse du neurone formel aux informations qu'il reçoit en entrée. Ce peut l'identité, une fonction sigmoïde ..
- ❖ Mémoire associative : le réseau, va à partir d'une image partielle en entrée pouvoir renvoyer en sortie l'image complète.
- ❖ Mémoire distribuée : un objet est codé par une configuration du réseau et non par un neurone seul.
- ❖ Data mining : un processus itératif par lequel on extrait des connaissances valides, nouvelles, potentiellement utiles et compréhensibles .

# BIBLIOGRAPHIE

- [1] G. Leblanc ,  
Borland C++ Builder ,        Edition : Eyrolles ,1997 .
- [2] M. Mokhtari ,  
Matlab 5.2 et 5.3 et Simulink 2 et 3 pour étudiants et ingénieurs ,  
Edition : Springer ,2000 .
- [3] A. Belaid & Y. Belaid ,  
Reconnaissance des formes : Méthodes et applications ,  
Edition : InterEdition     , 1992
- [4] J. Hérault ,  
Réseaux neuronaux et traitement du signal ,  
Edition : Hèrmes ,1994
- [5] J.P. Cocquerez ,  
Analyse d'images : Filtrage et segmentation,  
Edition : Masson , 1997
- [6] M . Kunt,  
Traitement numérique des images : Volume 2 ,  
Edition : Presses polytechniques et universitaires romandes ,1993
- [7] Dr L. Hamami Mitiche ,  
Système de reconnaissance de l'écriture arabe imprimée ,  
Thèse de doctorat d'état en électronique,  
ENP ,2001/2002
- [8] J . Liberty ,  
Le langage C++ en 21 jours,  
Edition : Campus press , 2001
- [9] M. Brahim ,  
Réalisation d'un système de reconnaissance de l'écriture arabe  
imprimée ,  
Mémoire pour l'obtention du titre d'ingénieur d'état en électronique ,  
ENP , 2002
- [10] M.Younsi ,  
Contribution à la réalisation d'un système de reconnaissance de  
l'écriture arabe manuscrite ,  
Projet de fin d'étude pour l'obtention du diplôme d'ingénieur d'état en  
informatique ,  
USTHB , 2002
- [11] G . Dreyfus,  
Réseaux de neurones : Méthodologie et applications ,

Edition : Eyrolles ,2002

- [12] Light Speed Labeling : un nouvel algorithme d'étiquetage en composants connexes .  
[www.lis.jussieu.fr/lionel/publications/aaa2000\\_draft.pdf](http://www.lis.jussieu.fr/lionel/publications/aaa2000_draft.pdf)
- [13] Reconnaissance automatique de l'écriture et du document  
A . Belaid  
[www.loria.fr/abelaid/publi-ps/pour\\_la\\_science.pdf](http://www.loria.fr/abelaid/publi-ps/pour_la_science.pdf)
- [14] Introduction à C++ Builder  
Builder .PDF
- [15] Quelques types de réseaux de neurones  
[www.asi.insa-rouen.fr/pleray/ftp/rn.pdf](http://www.asi.insa-rouen.fr/pleray/ftp/rn.pdf)
- [16] E.Davalo ,  
Des réseaux de neurones,  
Edition : Eyrolles 1991
- [17] L'image numérique .  
[www.det.fundp.ac.be](http://www.det.fundp.ac.be)
- [18] N .Barkani & L.Taibi ,  
Système de classification et de détection d'anomalies numériques  
chromosomiques par réseaux de neurones,  
Mémoire du projet de fin d'études,  
USTHB ,2002