

P.N. 6/02

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE**



ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

Mémoire de fin d'études

**En vue de l'obtention du diplôme d'ingénieur d'Etat en
Electronique**

**REALISATION D'UN SYSTEME DE
RECONNAISSANCE D'ECRITURE
ARABE IMPRIMEE**

Dirigé par :
M^{me} L.HAMAMI

Réalisé par :
**M^r BRAHIM Mohamed
M^r CHAFAI SIDALI**

Promotion : Juin 2002

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
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'Etat en
Electronique**

**REALISATION D'UN SYSTEME DE
RECONNAISSANCE D'ECRITURE,
ARABE IMPRIMEE**

Dirigé par :
M^{me} L.HAMAMI

Réalisé par :
**M^r BRAHIM Mohamed
M^r CHAFAI SIDALI**

Promotion : Juin 2002

DEDICACES

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

*A*ux yeux de mes yeux :

Ma mère et mon père

A Mes chers grand-père et grand-mère.

A Mon cher oncle Krimo et mes chères tantes et cousin Islam.

A Mes chers frères et sœurs.

A Neïla.

A Mehdi.

A Tous ceux et celles qui me sont chers.

A Mes amis : Fethi , Lyes , Sidali , Radia , Samia , Amine , Salim,
Messaoud, Bilel, Mehdi, Lotfi, Cherif, Wissam, Rafik, Djamel,
Youcef et tous les autres...

Je dédie ce modeste travail.

MOHAMED

DEDICACES

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

A Mes très chers parents pour leurs sacrifices , leurs encouragements et leurs présences a tout moment.

A Mes deux chères sœurs Rima et Bouchra.

A Mes chers grand-père et grand-mère.

A Mon cher oncle Sidali.

A Mehdi.

A Tous ceux et celles qui me sont chers.

A Mes amis : Fethi , Lyes , Sidali , Radia , Samia , Amine , Salim et tous les autres...

Je dédie ce modeste travail.

SIDALI

Remerciements

Nous remercions Dieu de nous avoir permis de mener à bien notre entreprise. Que son nom soit glorifié.

Nos remerciements vont pour Mme Hamami qui a placé sa confiance en nous en nous guidant et en nous encourageant tout au long de notre travail.

Un grand remerciement aussi pour Sofiane pour la précieuse aide qu'il nous a fourni.

Nous remercions Monsieur MEHENNI, pour l'honneur qu'il nous a fait en acceptant de présider le jury.

Nous remercions également Madame MOUSSAOUI qui nous a fait l'honneur de faire partie du jury de soutenance.

Nous n'oublierons pas de rendre hommage à Ami Salah ainsi qu'aux responsables du centre de calcul pour leurs aides durant tout notre cursus.

Enfin, que toutes les personnes ayant contribué de près ou de loin à la réalisation de ce projet trouvent ici, l'expression de notre profonde gratitude.

عرفت العلاقة بين الإنسان و الحاسوب تطورا ملحوظا، أراد الإنسان أن يتصل مع الحاسوب بطريقة سهلة و طبيعية لتسهيل التبادل المعلومات، و أراد أن يجعل هذه الآلة قادرة على القراءة و الرؤية و التنقل و أن تحلل المعلومات بسرعة.

العمل المقدم من خلال هذه المذكرة يمثل تصميم برنامج للتعرف على الأحرف العربية المطبوعة. قمنا في البداية بتقييم الصورة وجعلها على شكل مصفوفة ، ثم نقوم بتصفية هذه الصورة من الشبهات ، و الهدف من هذه العمليات لتقسيم هذه الكتابة سطرا بسطر ثم كلمة بكلمة و أخيرا حرف بحرف.

و لتعرف على هذه الأحرف استعملنا ثلاثة طرق ، طريقة الأولى ، طريقة المقارنة و طريقتين باستعمال الشبكة العصبية.

Résumé

La relation entre l'homme et l'ordinateur a connu un développement important. L'homme veut communiquer avec l'ordinateur avec une méthode facile et naturelle pour faciliter l'échange de l'information, et il veut rendre cette machine capable de lire, voire, déplacer et analyser plus rapidement les informations.

Le travail effectué présente la conception d'un logiciel pour la reconnaissance de l'écriture arabe imprimée.

Au départ Nous avons commencé par la numérisation de l'image scannée et la rendre sous forme de matrice, nous avons, par la suite, filtré cette image. Dans le but de segmenter l'écriture scannée ligne par ligne, mot par mot et en dernier caractère par caractère.

Pour la reconnaissance de ces caractères nous avons utilisé trois méthodes, la première concerne la comparaison des chaînes, les deux autres utilisent les réseaux de neurones.

Abstract

The relation between the man and the computer experienced a significant development. The man wants to communicate with the computer with an easy and natural method for facility the exchange of information, and he wants to make this machine able to read, even, to move and analyze more quickly information.

Carried out work presents the design of software for the recognition of the printed Arab writing.

At the beginning We started with the digitalization of the scanned image and to return it in the form of matrix, us, thereafter, filtered this image. With an aim of segmenting the scanned writing line by line, word by word and in last character by character.

For the recognition of these characters we used three methods, the first relates to the comparison of the chains, the two others use the networks of neurons.

Mots clé

Reconnaissance, écriture, imprimé, arabe

Introduction.....	1
I-Généralités.....	2
I-1-historique :.....	3
I-2-APPLICATION :	3
I-3- LES DIFFERENTS ASPECTS DE L'OCR :	3
I-4-LES technique de l'OCR :	4
I-5-OBJECTIF DE LA RECONNAISSANCE :	6
II: Description d'un système de reconnaissance des formes.....	7
II-1-Chaîne d'acquisition :	9
II-2-Module de prétraitement :	10
II-2-1-Filtrage :	11
II-2-2-Conclusion.....	14
II-3-Segmentation	15
II-4-Module d'apprentissage et décision :	15
II-4-1-L'apprentissage :	15
II-4-2-Décision :	16
III LES RESEAUX DE NEURONES.....	17
III-1-Introduction:	18
III-2-Fondements biologiques :	19
III-2-1-Le neurone biologique.....	19
III-2-2-Définitions :	19
III-2-3-Structure du neurone biologique:.....	19
III-3-Les réseaux de neurones artificiels :	20
III-3-1-Qu'est-ce qu'un neurone artificiel ?	20
III-3-2-Le neurone formel :	21
III-3-3-La fonction de seuillage:.....	22
III-4-Un peu de vocabulaire.....	24
III-5-Architecture des réseaux :	25
III-6-Apprentissage supervisé et non supervisé :	27
III-6-1-Réseaux à apprentissage supervisé :	27
III-6-2-Réseaux à apprentissage non-supervisé :	33
III-7-Exemples d'utilisation des réseaux de neurones :	35
III-8-Propriétés et limites des réseaux de neurones:	36
III-9-Conclusion :	36
IV: Application.....	38
IV-1-Prétraitement.....	39
IV-1-1-Introduction :	40
IV-1-2-Filtrage.....	40
IV-2-Segmentation.....	42
VI-2-1-Introduction	43
VI-2-2-Segmentation Horizontale :	43
IV-2-3-Segmentation Verticale :	47
VI-2-4-Segmentation en Caractères :	52
VI-2-5- Problèmes de la segmentation et les Solution proposées :	59
IV-3-RECONNAISSANCE.....	60
IV-3-1-La reconnaissance :	61
IV-3-2-Caractéristiques de l'alphabet arabe :	61
IV-3-3-Description de la méthode :	62
1-Extraction des primitives :	62
2-Reconnaissance et Apprentissage:	74
2-1Reconnaissance par comparaison des chaînes:.....	74
2-2Reconnaissance par réseaux de neurones	78
V: Présentation Du Logiciel.....	88
CONCLUSION GENERALE.....	97
ANNEXE.....	99
BIBLIOGRAPHI.....	102

INTRODUCTION

Introduction:

Les techniques liées au traitement de l'information connaissent actuellement un développement très actif en liaison avec l'informatique et présentent un potentiel de plus en plus important dans le domaine de l'interaction Homme-machine. L'Homme veut communiquer avec l'ordinateur de la façon la plus simple, la plus naturelle pour faciliter et accélérer l'interaction et l'échange d'informations. Il cherche à rendre ces machines accessibles par la voix, capables de lire, de voir, de se déplacer...et de traiter et d'analyser rapidement l'information reçue.

Ecrire pour communiquer a été de tous les temps une préoccupation première de l'Homme. L'écrit a été, et restera, l'un des grands fondements des civilisations et le mode par excellence de conservation et de transmission du savoir. Malgré les avancées d'autres moyens de communication tel que l'audio visuel, nombreuses sont les applications dont l'existence commence sur le papier, plus particulièrement dans la bureautique (saisie, indexation et archivage automatique de documents), en publication assistée par ordinateur (pour faciliter la composition à partir d'une sélection de plusieurs documents), dans la poste (lecture des adresses et tri automatique), dans les banques (traitement des chèques, des factures).... Cependant malgré les progrès technologiques, le clavier reste encore un moyen indispensable de communication avec l'ordinateur.

Notre application est un logiciel de reconnaissance de l'écriture arabe imprimée, ce dernier s'articule autour de trois modules, un module de pré traitement, un module de segmentation et un module de reconnaissance, notre logiciel propose trois modules de reconnaissance, le premier utilise une méthode de comparaison de chaînes, les deux autres utilisent des méthodes basées sur des réseaux de neurones, les résultats obtenus par ces trois méthodes sont très satisfaisants.

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

I-Généralités

I-1-HISTORIQUE

La reconnaissance de l'écriture est mieux connue sous le nom d'OCR (Optical Character Recognition) du fait de l'emploi des procédés d'acquisition optiques.

Son origine remonte aux années 1900 au cours desquelles on inventa le scanner à balayage pour la télévision et les machines à lire. "TUYRIN" développa alors la première application d'aide aux handicapés visuels, mais il a fallu attendre jusqu'à 1940 pour voir la première version d'informatique de cette application.

I-2-APPLICATION

L'OCR est utilisé dans plusieurs domaines parmi lesquelles on peut citer:

- **les banques**

Pour l'authentification des chèques.

- **la poste**

Par la lecture des adresses et le tri automatique du courrier.

- **la police et la sécurité**

Pour la reconnaissance des empreintes, l'authentification du manuscrit et l'identification du scripteur.

- **les affaires et l'industrie**

Pour la lecture automatique des documents administratifs [10].

I-3- LES DIFFERENTS ASPECTS DE L'OCR

Il existe plusieurs systèmes OCR selon le type de données à traiter et de l'application visée. Voici quelques aspects de L'OCR :

- **Reconnaissance de l'imprimé ou du manuscrit :**

L'approche n'est pas la même selon qu'il s'agisse de l'imprimé ou du manuscrit. Dans le manuscrit.

- **Reconnaissance monofonte, multifonte, omnifonte :**

La question se pose pour un texte imprimé, un système est dit monofonte s'il ne traite qu'une fonte à la fois, il est dit multifonte s'il est capable de reconnaître un mélange de fontes préalablement apprises, enfin un système est dit omnifonte s'il est capable de reconnaître toute fonte sans l'avoir apprise.

- **Reconnaissance en ligne (on-line) ou hors ligne (off-line) :**

La première dynamique se fait pendant l'écriture, elle permet de corriger ou de modifier l'écriture de manière directe et instantanée, tandis que la deuxième démarre après la fin de l'acquisition du document entier, elle permet d'analyser un grand nombre de caractères, résultant d'un pré traitement coûteux [11].

I-4-LES TECHNIQUE DE L'OCR

a) Méthode de corrélation ou de masquage :

Le principe de cette méthode est fondé sur l'examen d'un certain nombre de pixels dans l'image du caractère et l'étude de leur appartenance .

Une forme est reconnue si l'ensemble de ses pixels est présent .

Cependant à cause de la rigidité de ces méthodes (conçues à partir d'emplacements fixes de pixels) chaque masque de la taille du motif recherché est déplacé sur toute la région où le motif est supposé être et on identifie le motif à partir du masque qui réalise le maximum de correspondances.

Cette méthode est la plus ancienne et n'est plus employée aujourd'hui que pour des produits spécifiques (banque, poste), elle représente l'avantage d'être très rapide, relativement fiable et surtout très facile à mettre en œuvre , mais elle correspond à une série de caractères bien particuliers avec une fonte fixe .

b) Méthodes structurelles :

Elles sont fondées sur les structures propres du caractère, la structure est exprimée en terme de composants primitifs correspondant à des formes élémentaires telles qu'un rebroussement, un changement d'orientation, un accroissement ou un décroissement de pente, ces composantes sont appelées primitives, parmi ces méthodes on peut citer :

b-1) Méthodes de tests :

Elles consistent à appliquer sur chaque caractère entré des tests de plus en plus fins sur l'absence ou la présence de ces primitives.

Le processus le plus répandu ou habituel consiste à diviser à chaque test, l'ensemble des choix en deux jusqu'à n'obtenir qu'une seule forme correspondant au caractère entré .

b-2) Comparaison des chaînes :

Les caractères sont présentés par des chaînes de primitives, la comparaison du caractère test avec un modèle de référence consiste à mesurer la ressemblance entre les deux chaînes en calculant des distances.

b-3) Méthode syntaxique :

Chaque caractère est représenté par une phrase dans un langage où le vocabulaire est constitué de primitives .

Les caractères d'une même famille sont représentés par une grammaire, la reconnaissance consiste à déterminer si la phrase de description du caractère peut être générée par la même grammaire, on définit la distance d'une phrase à une grammaire comme étant la distance minimale de modification à faire subir à la phrase pour que le caractère (ou la phrase) soit accepté par la grammaire .

c) Méthode des experts :

Chaque expert a la responsabilité de son caractère, l'avantage de cette technique est la rapidité ; dès qu'un caractère est identifié, on passe à l'autre, le seul problème de cette méthode est la complexité de programmation d'un expert.

Un système expert est un logiciel issu des recherches en intelligence artificielle (I A) destiné à remplacer l'être humain dans toutes les tâches d'expertise sur un domaine ou un problème en tentant de reproduire le raisonnement d'un ou plusieurs experts humains [12].

b) Méthode heuristique :

Les méthodes heuristiques sont des procédures basées sur l'intuition et l'expérience de l'automaticien.

Les performances et la structure de ces systèmes en dépendront.

Cette approche présente les avantages suivants :

- Elle peut simplifier la résolution de problèmes complexes.
- Elle est souple.

Les règles et procédures utilisées dans cette technique sont spécifiques aux problèmes à résoudre et ne peuvent donc pas être généralisées à d'autres .

I-5-OBJECTIF DE LA RECONNAISSANCE :

L'objectif de la reconnaissance d'un document est l'obtention d'une description syntaxique des différents éléments qui le constituent à partir de la masse d'informations qu'il contient à l'état brut ; c'est d'abord un moyen économique qui permet de baisser considérablement le coût de la saisie et augmenter sa vitesse.

La reconnaissance de caractères permet à une machine d'approcher le comportement humain de la lecture, au départ le document est sur un support papier et après passage par un système OCR, ce dernier va créer une copie numérisée du document à traiter .

De manière générale, un système OCR est composé de :

- Un module d'acquisition et de prétraitement.
- Un module de segmentation.
- Un module de reconnaissance.

**Chapitre II:
Description d'un
système de
reconnaissance des
formes**

Chapitre II : DESCRIPTION D'UN SYSTEME DE RECONNAISSANCE DES FORMES

Introduction :

Un système de reconnaissance des formes peut être représenté schématiquement par le figure II-1.

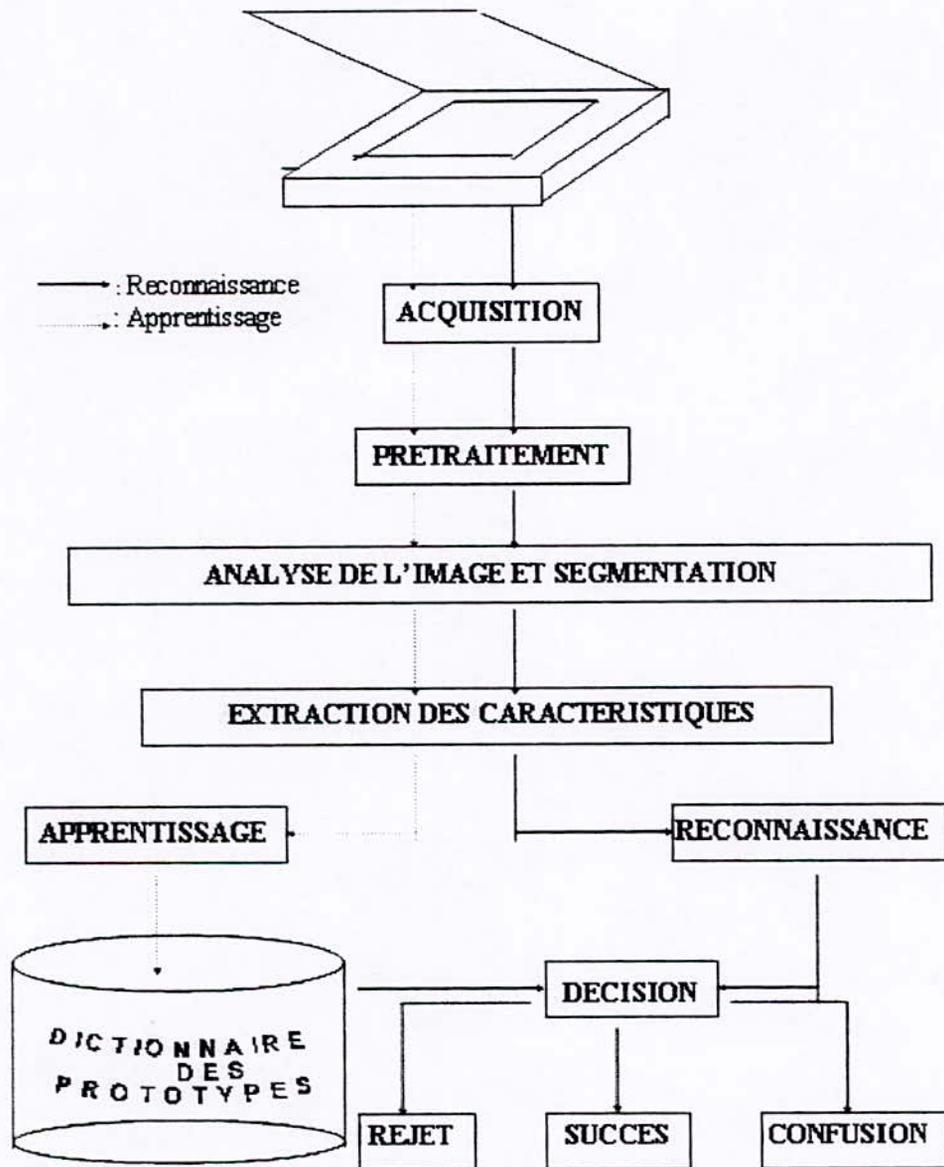


Figure -II-1

Architecture d'un système de reconnaissance des formes :

Chapitre II : DESCRIPTION D'UN SYSTÈME DE RECONNAISSANCE DES FORMES

II-1-Chaîne d'acquisition :

L'acquisition est l'opération qui permet de capter les informations du monde extérieur auquel le système est destiné, c'est donc l'interface entre le système et l'environnement extérieur, il n'a aucun pouvoir décisionnel, mais il peut avoir des conséquences considérables sur les traitements ultérieurs.

Les structures des autres modules dépendent étroitement du capteur utilisé, ainsi que de la nature de l'information qu'il délivre. En effet, l'opération d'acquisition permet le passage de l'information du monde réel au monde numérique de l'ordinateur. Ce passage nommé aussi numérisation ou digitalisation, n'est autre que la transformation d'un signal analogique en un signal numérique.

Dans notre cas la chaîne d'acquisition est constituée d'un scanner et d'un ordinateur .

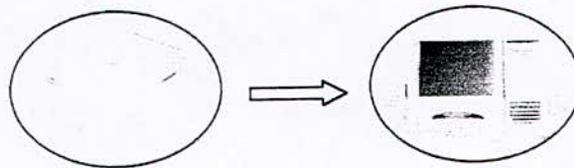


Figure -II-2 : Dispositifs d'acquisition

➤ Il existe différents types de scanners dont :

- **Le scanner à main :**

Un scanner à main ne peut travailler que sur de petites fenêtres, ce qui exige plusieurs passages pour digitaliser des feuilles de plus grandes dimensions. Le déplacement le long du document à numériser se fait manuellement.

Chapitre II : DESCRIPTION D'UN SYSTÈME DE RECONNAISSANCE DES FORMES

- **Le scanner à plat :**

Un scanner à plat est un scanner complet qui se charge tout seul de l'opération de numérisation (il n'a pas besoin d'intervention extérieure pour parcourir la surface à digitaliser). Il dispose à cet effet d'un mécanisme qui lui permet de scanner d'un seul coup, des pages de format A4 d'où le second nom de



Figure -II.3- : scanner à plat

scanner A4.

- **Le scanner à défilement :**

Ce type de scanner n'accepte que des feuilles volantes. Son mécanisme, à l'inverse du scanner à plat, ne déplace plus les sources lumineuses sur la feuille à scanner mais attire plutôt cette dernière vers l'intérieur, à la manière des imprimantes, et la digitalise ligne par ligne.

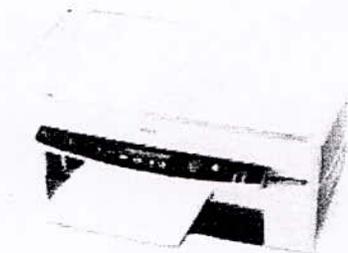


Figure -II.4- : Scanner à défilement

II-2-Module de prétraitement :

Le rôle du prétraitement est de «préparer» les données reçues du capteur à la phase suivante d'analyse. Cette phase n'est possible et surtout fiable que si les données du capteur sont dénuées de bruit, corrigées de leurs erreurs éventuelles, homogénéisées et réduites à l'essentiel. Toutes les techniques élaborées dans ce sens se gardent de modifier les propriétés essentielles des formes, ce qui pourrait conduire, dans le cas contraire à de graves erreurs d'analyse et plus tard de reconnaissance.

Chapitre II : DESCRIPTION D'UN SYSTÈME DE RECONNAISSANCE DES FORMES

II-2-1-Filtrage :

Des bruits d'origines diverses provoquent une certaine dégradation de l'image. Ils peuvent être introduits au cours de plusieurs phases du procédé d'acquisition, comme la numérisation.

Afin de réduire l'influence de ces bruits d'origines diverses, un filtrage s'avère nécessaire pour améliorer les traitements ultérieurs. Il consiste à améliorer l'aspect visuel de l'image en diminuant le bruit. Les pixels d'une image ont lorsqu'ils sont voisins, les mêmes caractéristiques physiques ; donc le bruit est un phénomène de brusque variation d'un pixel par rapport à ses voisins.

Citons quelques types de filtres en traitement numérique des images :

a- Filtrage linéaire : [2]

Ce type de filtrage est caractérisé par une opération de convolution qui est le remplacement de la valeur d'un pixel par une combinaison linéaire de ses voisins.

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

$B(i, j)$: Est la valeur de la matrice filtrée au pixel (i, j) , elle représente la somme pondérée des valeurs des pixels, dans la matrice non filtrée, autour de (i, j) .

$h(k, l)$: Ces valeurs donnent les poids.

$A(k, l)$: La valeur initiale du pixel (k, l) .

w, v : Déterminent les dimensions de la fenêtre $((2w+1)(2v+1))$.

La formule précédente peut être écrite d'une autre manière :

$$B = A * h$$

[: Symbolise l'opérateur de convolution.

Les valeurs de 'h' portent indifféremment les noms de : poids, noyaux, masque du filtre, et elles portent aussi le nom de la réponse impulsionnelle qui caractérise les filtres linéaires.

Chapitre II : DESCRIPTION D'UN SYSTÈME DE RECONNAISSANCE DES FORMES

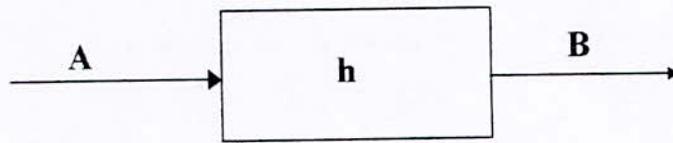


Figure -II.5- : Filtrage linéaire

Parmi les filtres linéaires les plus répandus nous citons :

- **Filtre moyen :**

Le filtre moyen consiste à considérer chaque pixel de l'image et d'en faire la moyenne avec les huit pixels (dans le cas du masque 3*3) qui lui sont voisins. Ceci va avoir pour effet d'adoucir l'image en réduisant les fluctuations des niveaux de gris [15]. Ce filtre peut être réalisé en utilisant un masque H :

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

Où N : représente le coefficient de normalisation.

N = somme des coefficients non nuls.

Chaque pixel est multiplié par le coefficient correspondant, les masques les plus couramment utilisés sont :

$$H_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, H_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, H_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

On va alors déplacer le masque choisi sur toute l'image, le pixel affecté par la transformation étant le pixel central du masque.

Plusieurs types de filtres moyens existent, entre autres : [9]

Chapitre II : DESCRIPTION D'UN SYSTÈME DE RECONNAISSANCE DES FORMES

➤ **Moyen Diagonal** : Ce filtre considère le masque suivant :

$$H = \frac{1}{5} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

On remarque que les coefficients diagonaux sont à 1 et le reste à zéro.

➤ **Moyen Square** : Ce filtre considère le masque suivant :

$$H = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

➤ **Moyen Cross** : Ce filtre considère le masque suivant :

$$H = \frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

On remarque que les coefficients non nuls forment un «+» (Cross).

b- Filtrage non-linéaire :

Parmi les plus importants des filtres non-linéaires on trouve :

➤ **Le filtre Médian :**

Ce filtre agit sur les pixels selon une loi non-linéaire, qui consiste à :

- Classer les pixels voisins par ordre croissant.
- Affecter la valeur médiane au pixel courant.

Le filtre médian semble très efficace pour une image dégradée par une source de bruit impulsionnel, donc pour les variations brusques de pixels isolés, la taille du filtre influe beaucoup sur sa qualité, plus la taille du filtre est grande, plus le filtre paraît efficace.

**Chapitre II : DESCRIPTION D'UN SYSTEME DE RECONNAISSANCE
DES FORMES**

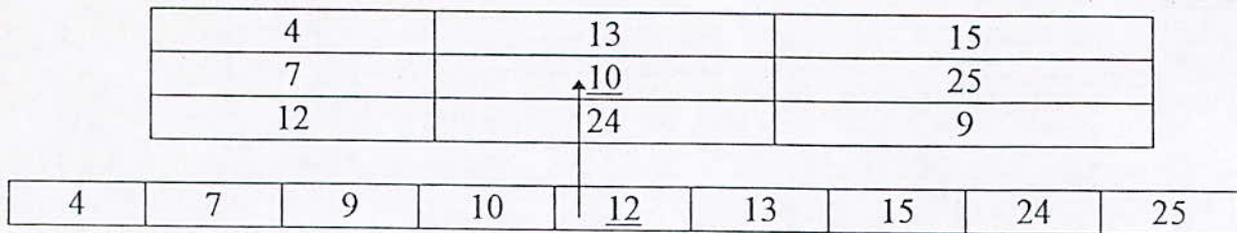


Figure -II.6- : Principe du filtre Médian

a- Filtrage morphologique :

Ce filtre est applicable sur des images binaires (2 niveaux de gris), ce filtre met en correspondance chaque pixel et son voisin par une fonction logique plus ou moins compliquée. Parmi les opérateurs morphologiques, nous citons :

➤ **La Dilatation :**

Elle permet d'éliminer les pixels noirs isolés au milieu des parties blanches de l'image, on effectue le **OU** logique des huit voisins du pixel considéré. En appliquant une dilatation ces tâches noires peuvent être éliminées mais la taille des objets présents dans l'image change, entraînant une déformation de certains objets.

➤ **L'Erosion :**

L'érosion permet d'éliminer les pixels blancs isolés. On effectue le **ET** logique des huit voisins du pixel considéré. En appliquant l'érosion, les tâches blanches dans les zones noires sont éliminées mais elles ont été entourées par d'autres points noirs, de plus l'épaisseur du trait sera augmentée.

➤ **L'Ouverture :**

L'ouverture est constituée par une opération d'érosion suivie d'une dilatation.

➤ **La Fermeture :**

La fermeture est une opération morphologique qui consiste à faire subir à l'image une opération de dilatation suivie d'une érosion.

Chapitre II : DESCRIPTION D'UN SYSTEME DE RECONNAISSANCE DES FORMES

II-3-Segmentation [5] :

Le but de la segmentation est de partitionner l'image originale en plusieurs régions distinctes. Le succès de cette opération dépend du degré de corrélation qui existe entre les entités du monde réel se trouvant sur l'image tels que : les objets, les surfaces ou les parties d'objets et les régions extraites par la segmentation comme c'est le cas dans de nombreux problèmes de vision.

Il est souvent difficile de mettre au point un algorithme de segmentation qui fonctionne correctement dans tous les cas et aussi bien que le système visuel humain.

Il existe plusieurs approches de la segmentation qui sont liées :

- A la nature de l'image (Bruit, Texture, Contours, flous, etc...).
- Aux opérations situées en aval (Reconnaissance de formes, Localisation, etc...).
- Aux primitives à extraire (Contours, Segments de droite, Régions, etc...).
- Aux contraintes d'exploitation de la méthode (Complexité algorithmique, Taille mémoire, etc...).

Cette étape est très importante car le résultat de la reconnaissance en dépend Très fortement.

II-4-Module d'apprentissage et décision :

II-4-1-L'apprentissage :

L'étape d'apprentissage est l'étape clés dans le processus de reconnaissance de formes, son rôle est de définir des modèles de référence ou de caractériser des classes de décision les plus adéquates vis-à-vis des règles de modélisation choisies , pour éclairer la décision à l'aide des connaissance apprises.

Chapitre II : DESCRIPTION D'UN SYSTEME DE RECONNAISSANCE DES FORMES

II-4-2-Décision :

La décision est l'étape de reconnaissance proprement dite, son rôle est d'identifier l'objet, la méthode de décision est souvent exhibée par l'apprentissage, ce qui veut dire que les critères utilisés pour la comparaison sont les mêmes que ceux utilisés pour l'apprentissage, en effet, il est évident que le choix de critères différents pour la décision, n'assurant pas l'uniformité de la description, ne conduisent pas avec certitude à des résultats cohérents.

Parmi les techniques utilisées certaines sont fondées sur la notion de proximité et nécessitent le calcul d'une distance ou d'une probabilité de ressemblance avec les modèles définis.

D'autres sont fondées sur l'analyse de la structure de la forme et essaient plutôt de vérifier une certaine cohérence dans les relations entre les différentes sous-structures.

La réponse de la décision peut être selon le cas, le nom de la forme en cas de bonne reconnaissance, plusieurs noms en cas d'ambiguïté ou bien le rejet de la forme en cas d'incompatibilité de description avec les formes de référence.

**CHAPITRE III
LES RESEAUX DE
NEURONES**

III-1-Introduction:

L'informatique est la science du traitement automatique de l'information. Son développement est souvent confondu avec celui des machines de traitement : les ordinateurs.

Cependant, l'augmentation de la puissance de l'ordinateur ne permet pas de toujours résoudre les problèmes d'une application informatique dans un domaine particulier. L'idée s'est donc installée que ce n'était peut être pas tant le matériel que le logiciel qui pêchait par manque de puissance. La construction de logiciels s'appuie sur plusieurs approches. Deux parmi les plus utilisées sont l'approche algorithmique et l'approche basée sur la connaissance.

Une **approche algorithmique** nécessite l'écriture de l'algorithme à suivre pour résoudre le problème. Lorsque le problème est complexe, ce peut être une étape coûteuse ou impossible. D'autre part, les ordinateurs sont des machines complètement logiques qui suivent à la lettre chacune des instructions du programme. C'est un avantage lorsque tous les cas ont été prévus à l'avance dans l'algorithme. Ce n'est hélas pas toujours possible.

La seconde approche possible est celle de l'**intelligence artificielle (appelée I.A.)**, avec pour applications les plus connues les systèmes experts. Ici, la résolution du problème est confiée à un ensemble de règles données par l'expert humain du domaine. Il n'en demeure pas moins que toutes les règles doivent avoir été exprimées préalablement au traitement. Les cas qui n'ont pas été prévus par l'expert ne seront pas correctement traités. En fait, l'approche basée sur la connaissance se limite à des domaines d'application où la modélisation de la connaissance, par exemple sous forme de règles, est possible. Ces domaines sont souvent ceux des sciences dites "exactes" comme l'électronique, la mécanique, la physique par opposition aux sciences dites "humaines" comme la médecine, la psychologie, la philosophie où la connaissance est plus empirique. L'I.A. se révèle donc être principalement un moyen commode de stocker de la connaissance sous forme explicite.

Ces deux approches ne suffisent pas à répondre à tous les problèmes existants. Citons les domaines de la reconnaissance de formes (images ou signaux), du diagnostic, du contrôle des moteurs, de la traduction automatique, de la compréhension du langage, depuis longtemps explorés à l'aide des approches algorithmiques et à base de connaissances, qui n'ont pas rencontré le succès escompté. Pourtant, des êtres vivants relativement simples sont capables de réaliser certaines de ces opérations apparemment sans difficulté. Il suffit pour s'en rendre compte de lever les yeux, suivre le vol de la mouche et essayer de la capturer.

Une troisième approche au traitement automatique de l'information semble donc s'offrir à nous, où l'on cherche à s'inspirer du traitement de l'information effectué par le cerveau. D'où la naissance des **réseaux de neurones artificiels**.

III-2-Fondements biologiques :

III-2-1-Le neurone biologique [7]:

La connaissance de la structure interne du cerveau et la compréhension de son fonctionnement a été d'une extrême importance, pour le développement des réseaux de neurones artificiels. Nous allons, dans ce qui suit, décrire succinctement l'élément de base du système nerveux central : le neurone.

III-2-2-Définitions [8], [7] :

Le neurone est l'unité fonctionnelle de base du système nerveux. Le cerveau se compose d'environ 10^{12} neurones (mille milliards), avec 1000 à 10000 synapses (connexions) par neurone. Les neurones sont caractérisés par cinq fonctions de base qui sont :

- Recevoir des signaux en provenance des neurones voisins.
- Intégrer ces signaux.
- Engendrer un flux nerveux.
- Le conduire.
- Le transmettre à un autre neurone capable de le recevoir.

III-2-3-Structure du neurone biologique [7] :

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone. L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La jonction entre deux neurones est appelée la synapse (Figure III.1).

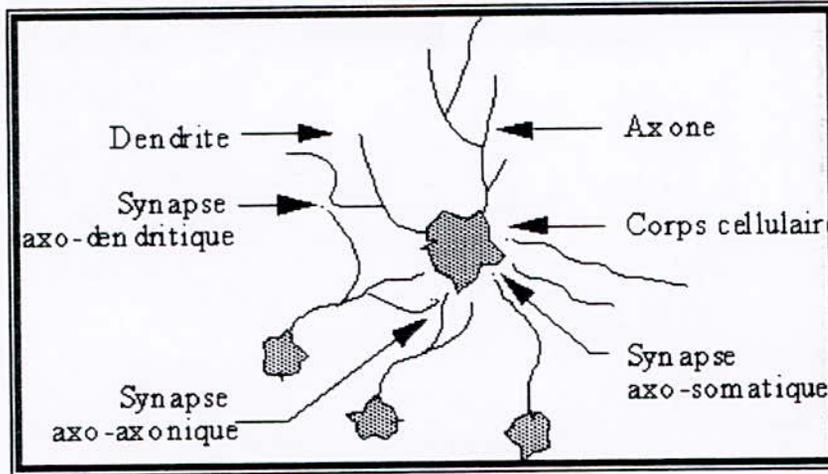


Figure III-1: Un neurone avec son arborisation dendritique [7].

III-3-Les réseaux de neurones artificiels :

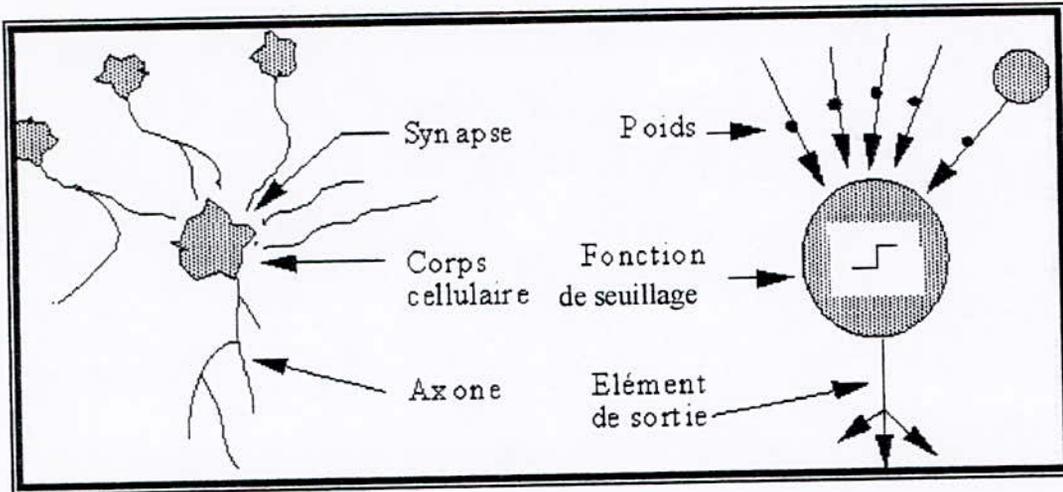
Copier le cerveau restera pour longtemps une ambition exagérée, mais vouloir s'inspirer des architectures et des fonctions du système nerveux n'est pas un rêve inaccessible ; c'est une réalité.

Les réseaux de neurones biologiques réalisent facilement un certain nombre d'applications telles que la reconnaissance de formes, le traitement du signal, l'apprentissage par l'exemple, la mémorisation, la généralisation. Ces applications sont pourtant, malgré tous les efforts déployés en algorithmique et en intelligence artificielle, à la limite des possibilités actuelles. C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau que les réseaux de neurones artificiels se sont développés. Les réseaux de neurones artificiels sont des modèles, à ce titre ils peuvent être décrits par leurs composants, leurs variables descriptives et les interactions des composants.

III-3-1-Qu'est-ce qu'un neurone artificiel ?

Ce que l'on désigne habituellement par «réseau de neurones » ou réseau neuromimétique est un réseau de neurones artificiels basé sur un modèle simplifié de neurone. Ce modèle permet certaines fonctions du cerveau, comme la mémorisation associative, l'apprentissage par l'exemple, le travail en parallèle, etc. [1].

La (Figure III.2) montre la structure d'un neurone artificiel. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associée un poids w abréviation de weight (poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associée un poids [7].



FigureIII-2: Mise en correspondance neurone biologique/neurone artificiel [7].

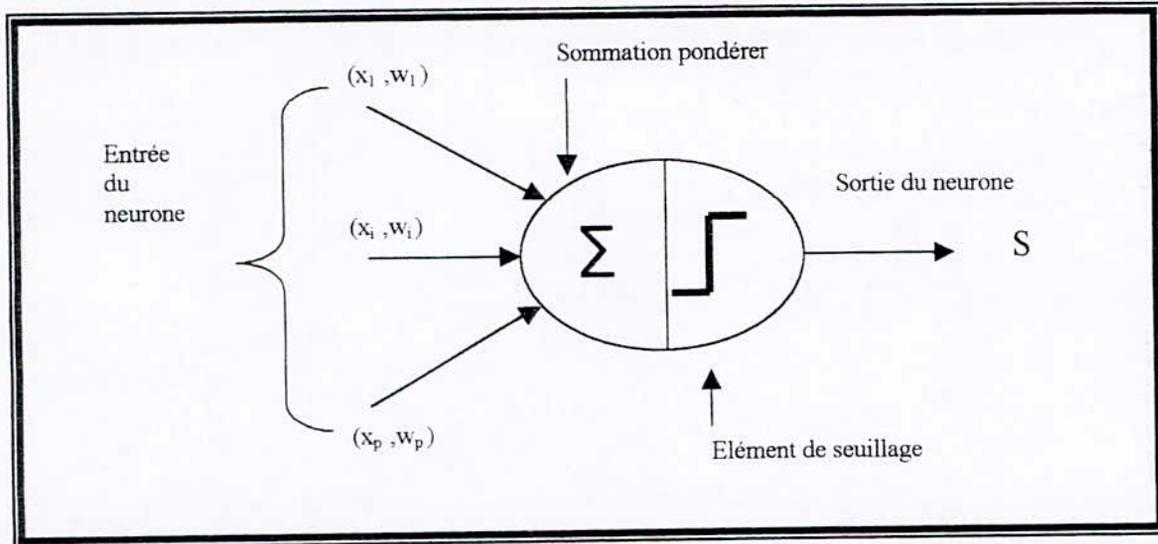
Ces réseaux sont capables de séparer les entrées en différentes classes, en leur présentant successivement des exemples, sans avoir besoin de lui apprendre des règles logiques ni de stocker des données.

III-3-2-Le neurone formel [3],[8]:

En 1943, Mc Culloch et Pitts inventèrent le premier modèle de neurone artificiel (FigureIII.3). Inspiré du neurone biologique, ce modèle n'était rien d'autre qu'un produit scalaire entre un vecteur d'entrées $X(x_1, \dots, x_i, \dots, x_p)$ et un vecteur poids $W(w_1, \dots, w_i, \dots, w_p)$, avec p :le nombre d'entrées, suivi d'une fonction f de seuillage.

Voici comment le neurone est modélisé mathématiquement :

$$S = f(\sum_{i=1,p} w_i * x_i) \quad \dots(1) \quad \text{Avec } f : \text{fonction de seuillage.}$$



FigureIII-3: Le modèle de neurone formel Mc Culloch et Pitts [3].

III-3-3-La fonction de seuillage[3]:

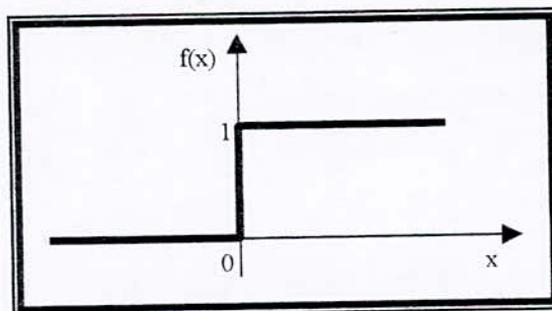
La fonction de seuillage f (d'activation ou d'état) définit son état interne en fonction de ses entrées totales. La fonction d'activation peut être "Discrète" ou "Continue".

- Fonction binaire à seuil (figures III.4 et III.5):

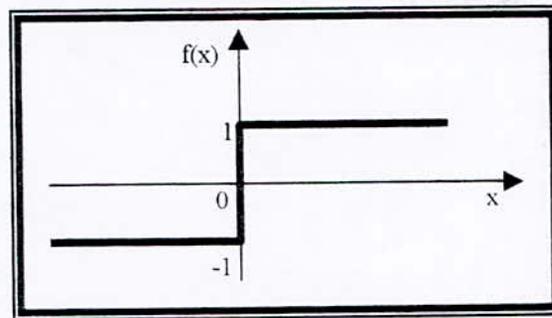
Les premiers modèles de neurones étaient caractérisés par une fonction d'activation à seuil simple (*binaire* : 0=*inactif*, 1=*actif*). Le déclenchement de l'activité intervient si la somme des excitations (activité des neurones excitateurs pondérée par les poids synaptiques) dépasse un certain seuil propre au neurone.

Fonction de Heaviside : $f(x) = 1$, si $x > \text{SEUIL}$.

$f(x) = 0$, sinon.



FigureIII-4: Fonction de Heaviside.



FigureIII-5: Fonction signe.

Le modèle binaire a été rapidement perfectionné (fonctions sigmoïdes...), mais l'effet de seuil reste souvent recherché car il confère une certaine efficacité aux réseaux et parce qu'il semble correspondre à **un modèle proche de la réalité**.

- Fonction sigmoïde :

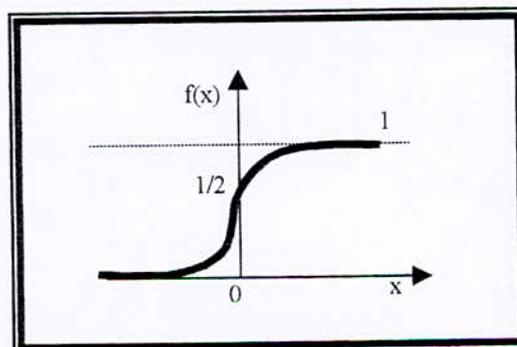
Lorsque la fonction d'activation est dérivable, il est possible d'utiliser des algorithmes d'apprentissage performants, basés sur la recherche d'un minimum de la fonction d'erreur, notamment la "rétropropagation du gradient" pour les perceptrons multicouches (voir plus loin).

Une fonction est "Sigmoïde" lorsqu'elle est monotone croissante, dérivable et bornée. Ce type de fonction combine l'avantage de l'effet de seuil et de la dérivabilité. Les fonctions sigmoïdes sont donc très utilisées, notamment dans le cas du perceptron multicouches.

Fonction sigmoïde exponentielle (ou logistique) : $f(x) = \frac{1}{1 + \text{EXP}(-x)}$

Dérivée : $f'(x) = f(x) \cdot (1 - f(x))$

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



FigureIII-6: Fonction logistique.

Fonction sigmoïde tangentielle : $f(x) = \text{TANH}(x)$

Dérivée : $f'(x) = 1 - f(x)^2$

III-4-Un peu de vocabulaire [3],[1] :

III-4-1-Apprentissage, mémoire et oubli :

Une caractéristique des réseaux de neurones est leur capacité d'apprendre (par exemple à reconnaître une lettre, un son...). Mais cette connaissance n'est pas acquise dès le départ, ces réseaux apprennent par l'exemple de la même manière qu'un enfant apprend à reconnaître un oiseau à partir d'exemples d'oiseaux qu'il a vus. Ils ont donc une capacité à classer, généraliser, à mémoriser et à oublier. Un réseau de neurones reconnaîtra d'autant plus facilement un objet qu'il l'aura vue souvent.

Définition : L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. t.

III-4-2-Connectivité :

La connectivité des réseaux, est la manière dont les neurones sont connectés. Elle peut être totale (tous les neurones sont connectés entre eux) ou par couches (les neurones d'une couche ne sont connectés qu'aux couches suivantes). Il existe des réseaux multicouches (Figure III.7) ou monocouches (Figure III.8).

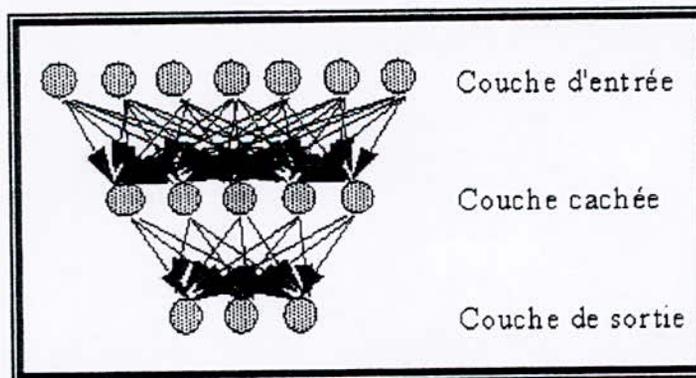
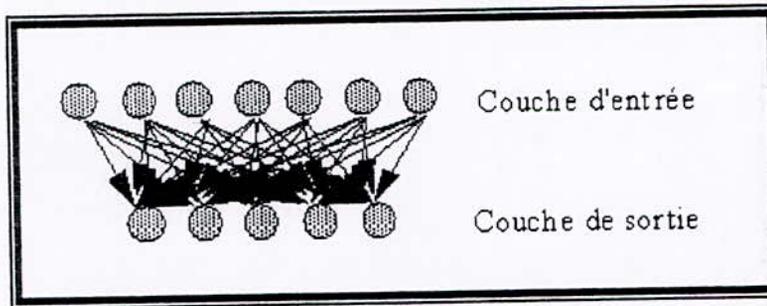


Figure III-7: Définition des couches d'un réseau multicouche.



FigureIII-8: Définition des couches d'un réseau monocouche.

III-4-3-Calcul des poids synaptiques :

La rétropropagation 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).

D'autres méthodes de modification des poids sont plus "locales", chaque neurone modifie ses poids en fonction de l'activation ou non des neurones proches. C'est le cas des réseaux suivant la règle de Hebb ou les réseaux de Kohonen (apprentissage non-supervisé).

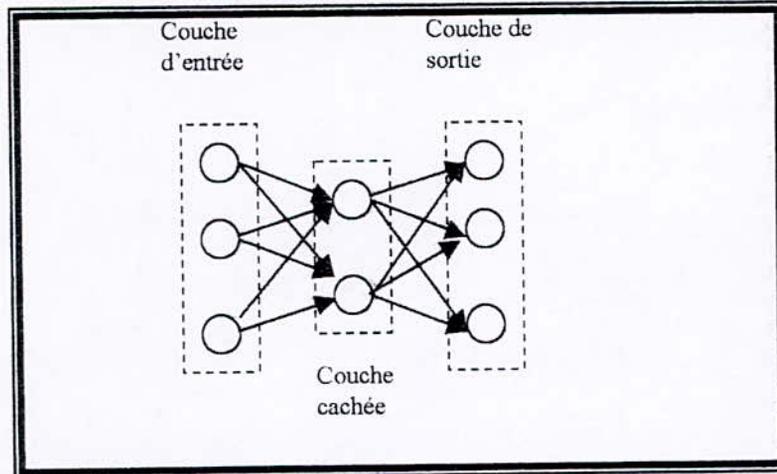
III-5-Architecture des réseaux [8] ,[3]:

Les structures utilisées sont très variées. Si l'on se réfère aux études biologiques du cerveau, on constate que le nombre de connexion est énorme. D'une manière générale, l'architecture d'un réseau définit son fonctionnement. Il existe trois types de réseaux à architectures différentes.

III-5-1-Réseaux statiques :

Ce type de réseau est organisé en couches de neurones (Figure III.9). Chaque neurone d'une couche reçoit ses entrées à partir des neurones de la couche précédente. Dans de tels réseaux il n'existe pas de retour d'information appelé «feed-back ». Ces réseaux peuvent être utilisés pour des problèmes de classification.

La première couche s'appelle couche d'entrée, la dernière couche de sortie, les autres s'appellent couches cachées.

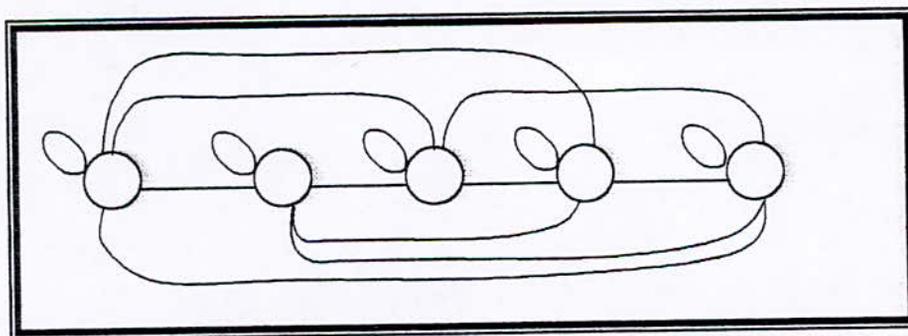


FigureIII-9: Réseaux statique (à couches)[3].

III-5-2-Réseaux dynamiques :

L'introduction d'une structure appelée feed-back entre les neurones a rendu le réseau dynamique(Figure III.10), ce qui a nécessité l'étude de la stabilité de ce dernier.

Ces réseaux sont exploités pour l'identification ou la commande de systèmes dynamiques.



FigureIII-10: Réseau dynamique [3].

III-5-3-Réseaux à architecture évolutive et réseaux Auto-Organisés :

- Les réseaux sont dits évolutifs si la dimension du réseau change pendant l'apprentissage ou l'entraînement. Ainsi le nombre de neurones augmente ou diminue.
- Les réseaux sont dits auto-organisés s'ils changent leurs structures internes pendant l'utilisation. Ainsi les neurones se regroupent topologiquement suivant la représentation des exemples issus de l'espace d'entrée.

III-6-Apprentissage supervisé et non supervisé :

Les réseaux de neurones se divisent en deux principales classes, les réseaux à apprentissage supervisés et les réseaux à apprentissage non supervisés.

Pour les réseaux à apprentissage supervisés (Perceptron, Adaline, etc...), on présente au réseau des entrées, et en même temps les sorties que l'on désirerait pour cette entrée. Par exemple on lui présente en entrées des caractéristiques d'un caractère et en sortie un code correspondant à ce dernier. Le réseau doit alors se reconfigurer, c'est à dire calculer ses poids afin que la sortie qu'il donne corresponde bien à la sortie désirée.

Pour les réseaux à apprentissage non supervisé (Hopfield, Kohonen, etc...), on présente une entrée au réseau et on le laisse évoluer librement jusqu'à ce qu'il se stabilise.

III-6-1-Réseaux à apprentissage supervisé :

Les réseaux à apprentissage supervisé représentent la classe de réseaux de neurones qui a fait l'objet du plus grand nombre de travaux de recherche et qui a connu une grande évolution depuis les années 1980.

L'histoire de ce type de réseaux commence par le perceptron de Rosenblatt en 1957 puis ce fut Widrow et Hoff avec l'ADALINE en 1960. Après une époque très difficile, c'est Rumelhart, Hinton et Williams qui ont ouvert une nouvelle ère avec l'algorithme de rétropropagation du gradient et les réseaux multicouche perceptron en 1986. C'est à partir de cette date que les réseaux à apprentissage supervisé ont connu un grand essor qui dure à nos jours [8].

III-6-1-1-Règles d'apprentissage :

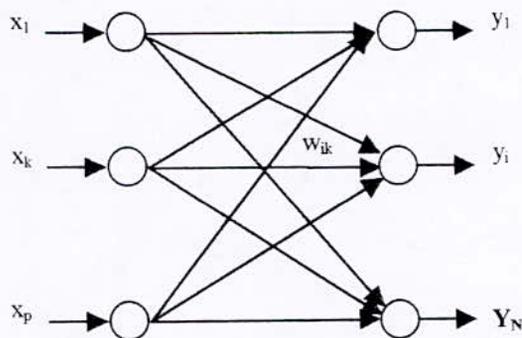
L'apprentissage d'un réseau de neurones supervisé revient à modifier les poids de ses connexions. En général, la modification d'un poids w_{ij} se fait de la façon suivante : $w_{ij} = w_{ij} + \Delta w_{ij}$. Dans la suite, nous allons expliquer comment ce fait la réadaptation de ces poids (Règle de Widrow-Hoff et l'algorithme de rétropropagation du gradient), c'est à dire comment calculer le terme Δw_{ij} . Les calculs seront conduits de façon générale avec une fonction de seuillage f simplement supposée dérivable (on utilise fréquemment avec ce type de réseaux une fonction de type sigmoïde).

a) Règle de Widrow-Hoff [4]:

Ce type de règle est utilisé pour les réseaux à une couche. On considère un réseau constitué de n neurones recevant des vecteurs X à p composantes $\{x_1, \dots, x_p\}$ (figure III.11). Les p entrées x_k du réseau sont distribuées sur tous les neurones. La sortie du neurone i vaut :

$$y_i = f(p_i) = f\left(\sum_{j=1}^p w_{ij} x_j\right) \dots\dots\dots(1)$$

Avec p_i le potentiel du neurone i.



FigureIII-11: Réseau à une couche.

A un vecteur d'entrée X, on veut associer un vecteur de sortie Yd (y_{d1}, \dots, y_{dn}) (sortie désirée). Si les poids w_{ik} ont des valeurs quelconques (ce qui est le cas initialement et avant la fin de l'apprentissage). le vecteur de sortie y observé est a priori différent de yd. On peut associer à cette différence l'erreur quadratique :

$$E = \frac{1}{2} \sum_{j=1}^n (y_j - y_{d_j})^2 \dots\dots\dots(2)$$

Calculons le gradient de cette erreur par rapport à w_{ik} :

$$\frac{\partial E}{\partial w_{ik}} = \sum_{j=1}^n (y_j - y_{d_j}) \frac{\partial (y_j - y_{d_j})}{\partial w_{ik}} \dots\dots\dots(3)$$

La dérivée partielle du membre de droite est nulle sauf pour $j=i$ car seul la sortie y_i est fonction du poids w_{ik} . De plus, la sortie désirée, fournie par le superviseur, ne dépend d'aucun poids.

On arrive finalement à :

$$\frac{\partial E}{\partial w_{ik}} = (y_i - yd_i) \frac{\partial y_j}{\partial w_{ik}} = (y_i - yd_i) x_k f'(p_i) \dots\dots\dots(4)$$

où f' dérivée de f .

En posant $\delta_i = (y_i - yd_i) \cdot f'(p_i)$, le gradient de l'erreur quadratique E s'écrit :

$$\frac{\partial E}{\partial w_{ik}} = \delta_i \cdot x_k \dots\dots\dots(5)$$

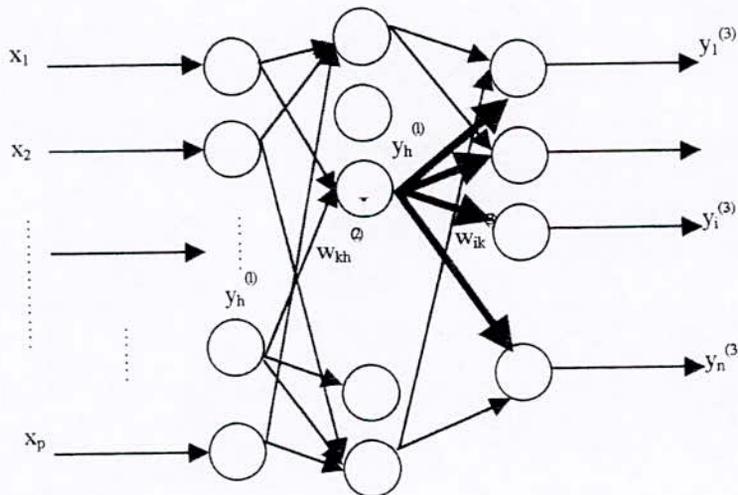
et la mise à jour du poids, selon le principe du gradient, s'écrit :

$$\Delta w_{ik} = -a \cdot \delta_i \cdot x_k \dots\dots\dots(6)$$

où a est un gain d'adaptation positif appelé **taux d'apprentissage**.

b) Algorithme de rétropropagation du gradient [4] :

Considérons le réseau multicouche (2 couches) de la (figure III.12). On veut comme précédemment associer un vecteur de sortie y_d à un vecteur d'entrée x . La sortie observée est $y \neq y_d$. On peut y associer le terme d'erreur quadratique E comme précédemment.



FigureIII-12: Réseau de neurones multicouches. Le numéro de la couche est indiqué en indice situé entre parenthèse et en haut.

Nous allons montrer que nous pouvons modifier les poids du réseau en minimisant cette erreur quadratique de façon analogue à celle de Widrow-Hoff. Cependant, si cette méthode est directement applicable pour ajuster les poids de la dernière couche, elle ne l'est pas pour les couches internes car on ne connaît pas la sortie désirée pour ces couches, et par conséquent on ne connaît pas directement le terme d'erreur associé à chaque couche interne. Il faut donc exprimer l'erreur à la sortie de chaque neurone d'une couche quelconque à partir de l'erreur de la dernière couche, seul erreur mesurable.

Ceci est possible par un calcul différentiel simple. On peut calculer l'erreur δ_k^j associée à un neurone k d'une couche quelconque j à partir des erreurs des neurones de la couche suivante(j+1) pondérées par les poids $w_{ik}^{(j+1)}$ des connexions partant du neurone k vers les neurones de la couche suivante (en gras sur la (figure III.12)). L'erreur $\delta_k^{(j)}$ s'exprime par :

$$\delta_k^{(j)} = f'(p_k^{(j)}) \cdot \sum_{i \in \text{couche } j+1} w_{ik}^{(j+1)} \cdot \delta_i^{(j+1)} \dots\dots\dots(7)$$

Avec le potentiel qui s'exprime par : $p_k^{(j)} = \sum_{h \in \text{couche } j-1} w_{kh}^{(j)} y_h^{(j-1)}$

Pour plus de détail sur le calcul du terme $\delta_k^{(j)}$ voir l'annexe .

En appliquant la méthode du gradient, la modification des poids se fait selon la règle :

$$\Delta w_{kh}^{(j)} = -a \cdot \delta_k^{(j)} \cdot y_h^{(j-1)} = -a \cdot \left[f'(p_k^{(j)}) \cdot \sum_{i \in \text{couche } j+1} w_{ik}^{(j+1)} \cdot \delta_i^{(j+1)} \right] \cdot y_h^{(j-1)} \dots\dots\dots(8)$$

La mise à jour de l'ensemble des poids d'une couche nécessite la connaissance des erreurs associées à chaque neurone de la couche suivante. On appliquera l'algorithme d'ajustement des poids en partant de la dernière couche (pour laquelle les erreurs sont connues) vers la première. D'où l'appellation de cet algorithme : **algorithme de rétropropagation du gradient de l'erreur**
 Nous allons dans ce qui suit donner l'algorithme de rétropropagation du gradient pour un réseau à 2 couches.

1)Présenter un exemple X_1 à l'entrée du réseau.

2)Calculer les potentiels des neurones de la couche cachée (couche numéro 2)

$$p_k^{(2)} = \sum_{h \in \text{couche } 1} w_{kh}^{(2)} x_h \dots\dots\dots(9)$$

3)Calculer les sorties des neurones de la couche cachée (couche numéro 2)

$$y_k^{(2)} = f(p_k^{(2)}) \dots\dots\dots(10)$$

4) Calculer les potentiels des neurones de la couche sortie (couche numéro 3)

$$p_i^{(3)} = \sum_{k \in \text{couche.2}} w_{ik}^{(3)} y_k^{(2)} \dots\dots\dots(11)$$

5) Calculer les sorties des neurones de la couche de la sortie (couche numéro 3)

$$y_i^{(3)} = f(p_i^{(3)}) \dots\dots\dots(12)$$

6) Calculer les termes d'erreur sur chaque unité de sortie :

$$\delta_i^{(3)} = (y_i^{(3)} - yd_i) \cdot f'(p_i^{(3)}) \dots\dots\dots(13)$$

7) Calculer les termes d'erreur sur chaque neurone de la couche cachée :

$$\delta_k^{(2)} = f'(p_k^{(2)}) \cdot \sum_{i \in \text{couche.3}} w_{ik}^{(3)} \delta_i^{(3)} \dots\dots\dots(14)$$

8) Calculer les valeurs de réadaptation des poids :

Pour la couche 3 : $\Delta w_{ik}^{(3)} = -\alpha \delta_i^{(3)} y_k^{(2)} \dots\dots\dots(15)$

Pour la couche 2 : $\Delta w_{kh}^{(2)} = -\alpha \delta_k^{(2)} x_h \dots\dots\dots(16)$

9) S'il s'agit d'un Data Adaptive Learning (voir définition plus loin) :

Aller à 11.

10) S'il s'agit d'un Block Adaptive Learning (voir définition plus loin) :

Si $l \neq M$ (M étant le nombre d'exemple d'entraînement) retourner à 1.

Sinon aller à 11.

11) Réadapter les poids :

$$w_{ij}^{(k)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)} \dots\dots\dots(17)$$

12) Répéter de 1 à 11 autant de fois que nécessaire jusqu'à ce que l'erreur quadratique E_1 de chaque exemple X_1 devient acceptable pour tous les exemples (inférieur à un seuil d'erreur fixé) avec :

$$E_1 = \frac{1}{2} \sum_{i \in \text{couche.3}} (y_i^{(3)} - yd_i)^2 \dots\dots\dots(18)$$

- **Data Adaptive Learning [8]** : en utilisant cette technique, les poids sont réadaptés au passage de chaque exemple. De cette manière, le processus devient plus sensible à chaque exemple individuellement, ce qui le rend donc, facilement influençable par les bruits que peuvent contenir ces entrées durant l'entraînement. Cette technique est utilisée pour un apprentissage en temps réel.
- **Block Adaptive Learning [8]**: dans ce cas, on ne réadapte les poids qu'après passage de tous les exemples d'entraînement. La réadaptation est plus «prudente ». Le réajustement se fait suivant la moyenne de tous les exemples, ce qui rend la méthode moins sensible aux bruits que peuvent contenir les exemples. La méthode est plus robuste si l'application ne nécessite pas un apprentissage en temps réel.

c) Lissage de la règle d'adaptation :

L'algorithme proposé précédemment est rarement utilisé tel quel en pratique. La vitesse de convergence est assez lente. L'amélioration la plus couramment utilisée consiste à ajouter un terme de filtrage sur les incréments d'adaptation. Ce terme est appelé «momentum » dans la littérature [4]. La règle d'apprentissage $w_{ij}^{(k)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)}$ proposé précédemment devient

$$w_{ij}^{(k)}(t) = w_{ij}^{(k)}(t) + \Delta w_{ij}^{(k)}(t) + \gamma(w_{ij}^{(k)}(t) - w_{ij}^{(k)}(t-1)). \dots\dots\dots(19)$$

Avec γ le momentum, en pratique, on le prend inférieur et voisin de 1.

III-6-1-2-Quelques modèles de réseaux à apprentissage supervisé [1]:

a) Les perceptrons :

C'est un des premiers réseaux de neurones, conçu en 1958 par Rosenblatt. Il est linéaire et monocouche. Il est inspiré du système visuel. C'est un réseau de neurones très simple. Les sorties des neurones sont binaires, elles ne peuvent prendre que deux états (-1 et 1 ou 0 et 1). Le procédé d'apprentissage est supervisé, ce qui veut dire que les sorties du réseau sont comparées avec des sorties désirées.

La règle de modification des poids utilisée est la règle de Widrow-Hoff : si la sortie (celle d'une cellule de décision donc) est égale à la sortie désirée, le poids de la connexion entre ce neurone et le neurone d'association qui lui est connecté n'est pas modifié. Dans le cas contraire le poids est modifié en fonction de l'entrée.

Le réseau est capable de résoudre des opérations logiques simples comme AND ou OR. En 1969, Papert et Minsky (un des pères des systèmes experts) démontrent les limites du perceptron classique, incapable, par exemple de simuler la fonction ou exclusif (xor).

b) Les perceptrons multicouches :

C'est le modèle le plus utilisé. Appelé aussi réseau de neurones à rétropropagation. Il est une amélioration du perceptron comprenant une ou plusieurs couches intermédiaires dites couches cachées. Il utilise, pour modifier ses poids, un algorithme de rétropropagation du gradient, qui est une généralisation de la règle de Widrow-Hoff. Il s'agit toujours de minimiser l'erreur quadratique, ce qui est assez simple quand on utilise une fonction f dérivable (la sigmoïde par exemple). On propage la modification des poids de la couche de sortie jusqu'à la couche d'entrée. Les perceptrons multicouches agissent comme un séparateur non linéaire et peuvent être utilisés pour la classification, le traitement de l'image ou l'aide à la décision. Ils permettent d'obtenir les résultats les plus satisfaisants. Il est très gros consommateur de temps de calcul sur des problèmes de grande taille.

III-6-2-Réseaux à apprentissage non-supervisé :

Les réseaux à apprentissage non-supervisé traitent des échantillons en entrée sans avoir d'informations sur les sorties que ceux-ci doivent générer. Ces réseaux présentent moins de complexité de calculs mais sont moins précis que ceux à apprentissage supervisé.

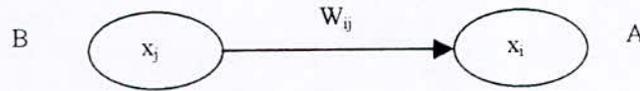
L'utilisation de ces réseaux est pratique dans les problèmes qui nécessitent un apprentissage rapide en temps réel. Après l'échec du perceptron linéaire et avant l'apparition de l'algorithme de rétropropagation au gradient assurant l'apprentissage supervisé des réseaux multicouches, c'est vers ces réseaux que beaucoup de travaux se sont orientés [8].

III-6-2-1-La loi de Hebb, le fondement de l'apprentissage non-supervisé [8] :

La loi de Hebb (1949) est le fondement de l'apprentissage non-supervisé. Elle s'exprime de la façon 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 ; le lien synaptique de B vers A devient plus significatif. Ainsi B pourra stimuler A plus facilement à l'avenir.



Cette règle a été développée sous des aspects mathématiques qui traduisent le principe initial.

Ainsi si w_{ij} est le poids synaptique reliant le neurone B au neurone A, la réadaptation de ce dernier peut s'écrire sous la forme discrète suivante :

$$\Delta w_{ij} = s(x_i) s(x_j)$$

où x_i représente l'activité du neurone et $s(.)$ est sa sortie, représentée par une fonction non linéaire appliquée à son activité.

III-6-2-1 Quelques modèles de réseaux à apprentissage non-supervisé :

a) Réseau de Hopfield :

Un réseau de Hopfield réalise une mémoire adressable par son contenu. Ils sont capables de mémoriser de l'informations que l'on peut ensuite retrouver non pas par leurs adresse comme dans une mémoire classique mais en fournissant des données même incomplètes ou bruitées relatives aux informations stockés. Il s'agit d'un réseau constitué de neurones de McCulloch et Pitts (à deux états -1 et 1 ou 0 et 1), dont la loi d'apprentissage est la règle de Hebb, qui veut qu'une synapse améliore son activité si et seulement si l'activité de ses deux neurones est corrélée (C'est à dire que le poids w_{ij} d'une connexion entre un neurone i et un neurone j augmente quand les deux neurones sont activés en même temps).

Ce type de réseau est utilisé dans les problèmes d'optimisation, tels le problème du voyageur de commerce (il s'agit de trouver le plus court chemin reliant N villes en passant une fois et une seule par une ville).

Les limites du réseau de Hopfield :

- Les prototypes doivent vérifier une condition d'orthogonalité assez contraignante et qui n'a pas de sens vis-à-vis d'un problème réel : comment peut-on assurer que des numéros de téléphone sont orthogonaux ?
- Oubli catastrophique : un réseau peut apprendre n exemples sans problème et les oublier tous dès lors que l'on souhaite lui en faire apprendre un de plus.

b) Le modèle de Kohonen :

Contrairement aux réseaux de Hopfield où les neurones sont modélisés de la façon la plus simple possible, on recherche ici un modèle de neurone plus proche de la réalité.

Ces réseaux sont inspirés des observations biologiques du fonctionnement des systèmes nerveux de perception des mammifères. Une loi de Hebb modifiée (tenant compte de l'oubli) est utilisée pour l'apprentissage. La connexion est renforcée dans le cas où les neurones reliés auraient une activité simultanée, et diminuée dans le cas contraire.

Une loi d'interaction latérale (observée biologiquement) est aussi modélisée. Les neurones très proches (physiquement) interagissent positivement (le poids des connexions est augmenté quand une synapse est activée), négativement pour les neurones un peu plus loin, et pas du tout pour les neurones éloignés. Ceci crée un "amas" de neurones activés et contribue à spécialiser certains neurones : pour une entrée donnée, une sortie particulière sera activée et pas les autres. On utilise aussi parfois des lois de concurrence entre les neurones (création et destruction de neurones selon certains critères).

Les réseaux de Kohonen permettent de résoudre les problèmes d'optimisation, tels le problème du voyageur de commerce (comment relier n villes par le chemin le plus court).

Les réseaux de Kohonen ont des applications dans les problèmes difficiles de la reconnaissance des formes comme la reconnaissance de la parole où les perceptrons multicouches sont limités dans leurs performances.

III-7-Exemples d'utilisation des réseaux de neurones [1] :

Se trouvant à l'intersection de différents domaines (informatique, électronique, science cognitive, neurobiologie et même philosophie), l'étude des réseaux de neurones est une voie prometteuse de l'Intelligence Artificielle, qui a des applications dans de nombreux domaines :

- **Industrie** : contrôle qualité, diagnostic de panne, corrélations entre les données fournies par différents capteurs, analyse de signature ou d'écriture manuscrite
- **Finance** : prévision et modélisation du marché (cours de monnaies...), sélection d'investissements, attribution de crédits.

- **Télécommunications et informatique** : analyse du signal, élimination du bruit, reconnaissance de formes (bruits, images, paroles), compression de données.
- **Environnement** : évaluation des risques, analyse chimique, prévisions et modélisation météorologiques, gestion des ressources.

III-8-Propriétés et limites des réseaux de neurones [3]:

III-8-1-les propriétés :

l'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification dans les quelques propriétés fascinantes qu'il possède :

➤ **la capacité d'adaptation :**

Elle se manifeste par la capacité d'apprentissage qui permet de tenir compte de nouvelles contraintes ou de nouvelles données du monde extérieur.

➤ **Le parallélisme :**

Les réseaux de neurones sont considérés comme un ensemble d'entités élémentaires qui travaillent simultanément. Le parallélisme permet une rapidité de calcul supérieure mais exige de poser différemment les problèmes à résoudre

➤ **La capacité de généralisation :**

La capacité de généralisation d'un réseau de neurones est sa capacité à donner une réponse satisfaisante à une entrée qui ne fait partie de son ensemble d'apprentissage.

III-8-2-Limites :

Les principales limites actuelles sont :

- La plupart des réseaux de neurones sont simulés sur des machines séquentielles. Ce qui entraîne des temps de calculs importants dès que la taille du problème devient grande.
- Incapacité d'expliquer les résultats qu'ils fournissent. Les réseaux de neurones se comportent comme des boîtes noires dont les règles de fonctionnement sont inconnues.

III-9-Conclusion :

L'apparition des réseaux de neurones est l'une des révolutions les plus importantes que le monde de l'informatique a connu. Leur utilisation dans les différentes applications a augmenté durant ces dernières années grâce à l'augmentation de la vitesse des processeurs et de la mémoire.

A travers ce chapitre, nous avons constaté que les réseaux de neurones sont une famille d'outils très diversifiés. Chaque modèle a ses propres caractéristiques et peut être destiné à des opérations spécifiques.

Chapitre IV:

Application

IV-1-Prétraitement

IV-1-1-Introduction :

Lorsque l'acquisition est réalisée, la plupart des systèmes OCR comportent une étape de prétraitement, généralement les prétraitements ne sont pas spécifiques de la reconnaissance du texte, mais sont des prétraitements classiques en traitement d'images.

Le prétraitement peut comprendre des dizaines d'opérations différentes effectuées sur l'image numérisée, cette dernière peut présenter des distorsions dues à la qualité du scanner et/ ou du papier [13].

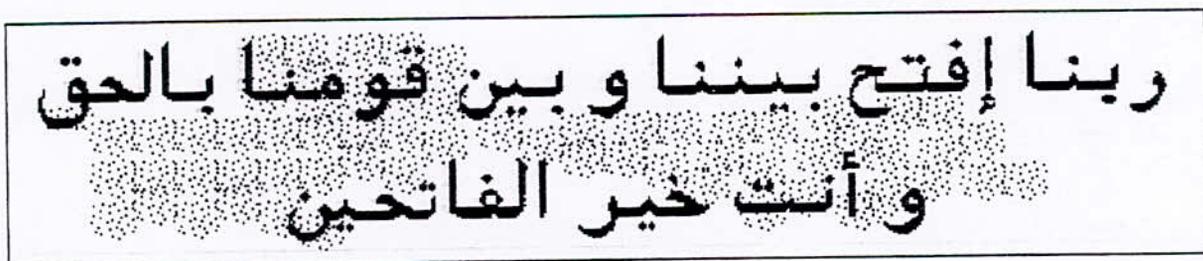
A cette phase de l'analyse, le but est de préparer les étapes ultérieures de la reconnaissance. Parmi ces opérations on peut citer le problème du bruit, l'élimination du fond dans le cas de la lecture d'un texte sur un fond texturé (chèque, papier trame, etc..).

On inclut généralement dans le prétraitement des opérations de type redressement des écritures penchées, détections et éventuellement redressement des lignes de bas [14].

Cette étape comporte deux phases qui sont :

IV-1-2-Filtrage :

Après l'étude comparative des filtres, et comme le bruit qui existe dans le document numérisé est un bruit impulsionnelle , on a opté pour le filtre médian qui élimine le bruit sans changer la forme des caractères, ce qui n'est pas le cas pour les autres filtres (linéaire, morphologique). La figure IV.1 donne un exemple d'une image filtrée par un filtre médian .



a- Image originale

ربنا إفتح بيننا و بين قومنا بالحق
و أنت خير الفاتحين

b- Image filtrée par le filtre médian
Figure IV.1 : Résultat du Filtre médian

IV-2-Segmentation

VI-2-1-Introduction

Une des premières étapes dans la conception d'un système de reconnaissance de l'écriture est la segmentation du texte en lignes. Cette opération est relativement facile si le texte n'est pas incliné, et si les lignes ne se chevauchent pas. Ces conditions peuvent être satisfaites pour l'imprimé. Dans le cas du manuscrit sans contrainte, l'écriture fluctue et peut présenter une inclinaison importante par rapport à l'horizontale: deux lignes adjacentes peuvent se chevaucher.

VI-2-2-Segmentation Horizontale :

la segmentation horizontale sépare les lignes dans un paragraphe (ou elle détecte le début et la fin de chaque ligne)

pour détecter le début et la fin d'une ligne nous calculons l'histogramme horizontal pour chaque ligne (ou la somme des pixels non blanc)

a- Algorithme de la segmentation horizontale

En premier lieu, on détecte le début de la $i^{\text{ème}}$ ligne, il faut que l'histogramme horizontal (ou la somme des pixels non blancs) soit supérieur ou égal à un. Ensuite on doit détecter la fin de la $i^{\text{ème}}$ ligne, il faut que l'histogramme horizontal (ou la somme des pixels non blancs) soit égal à zéro.

b- Organigramme de la segmentation horizontale(Figure IV.2.1)

Notation :

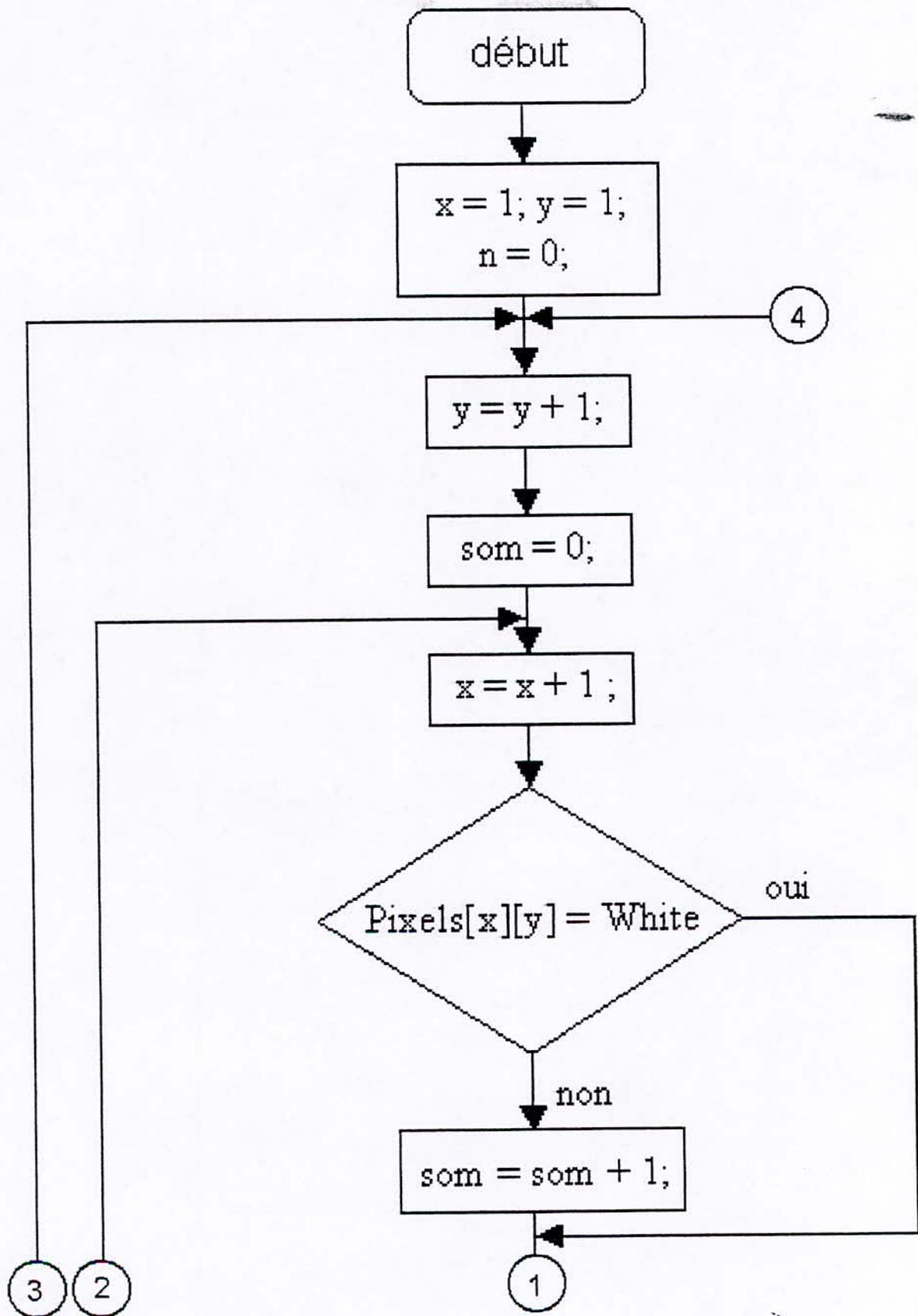
length = la longueur de l'image ;

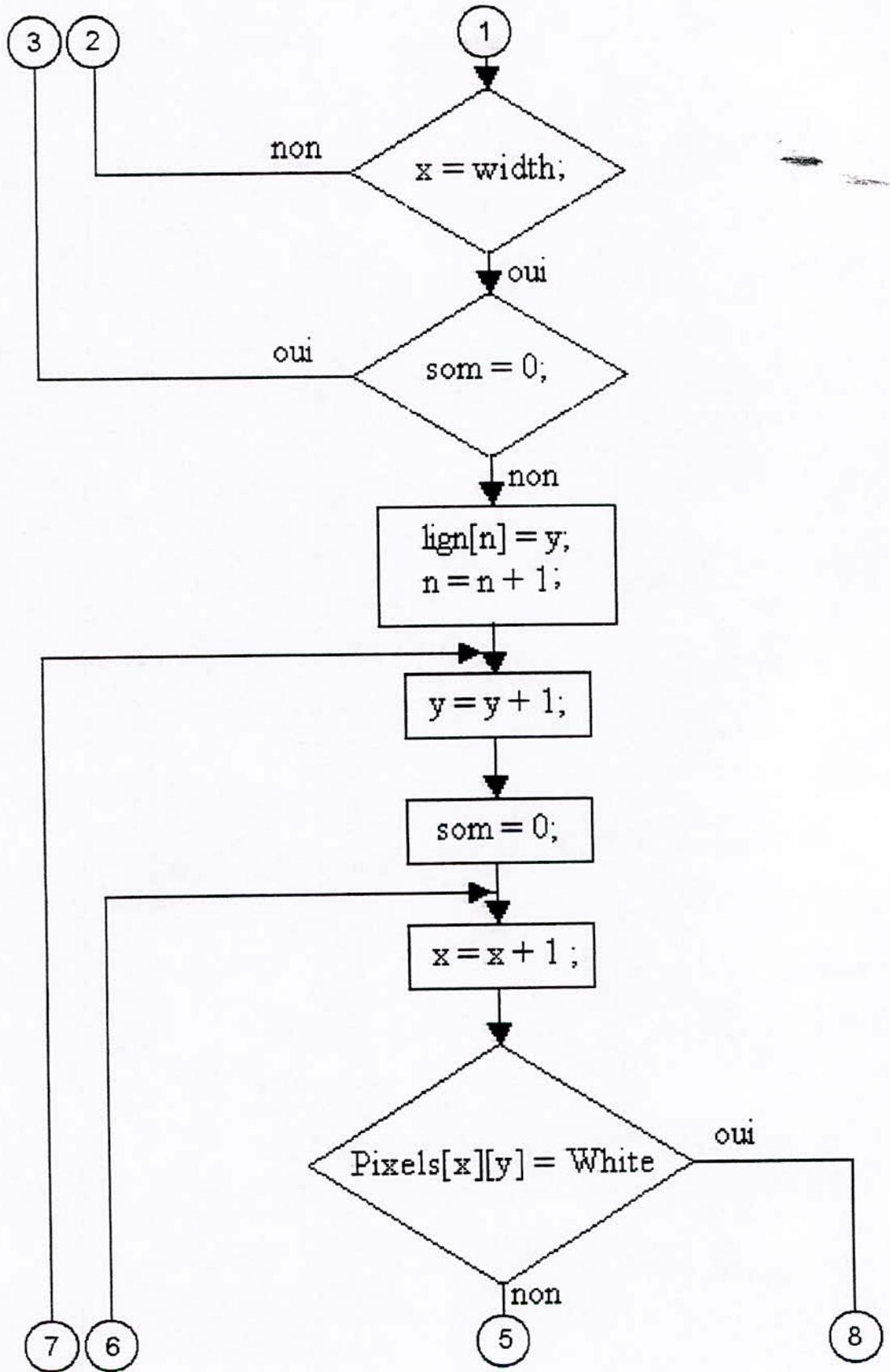
width = la largeur de l'image ;

pixels[x][y] = la valeur de la chrominance d'un pixel de coordonnées (x , y) ;

lign[2*(n-1)] = début de la ligne n ;

lign[2*n+1] = fin de la ligne n ;





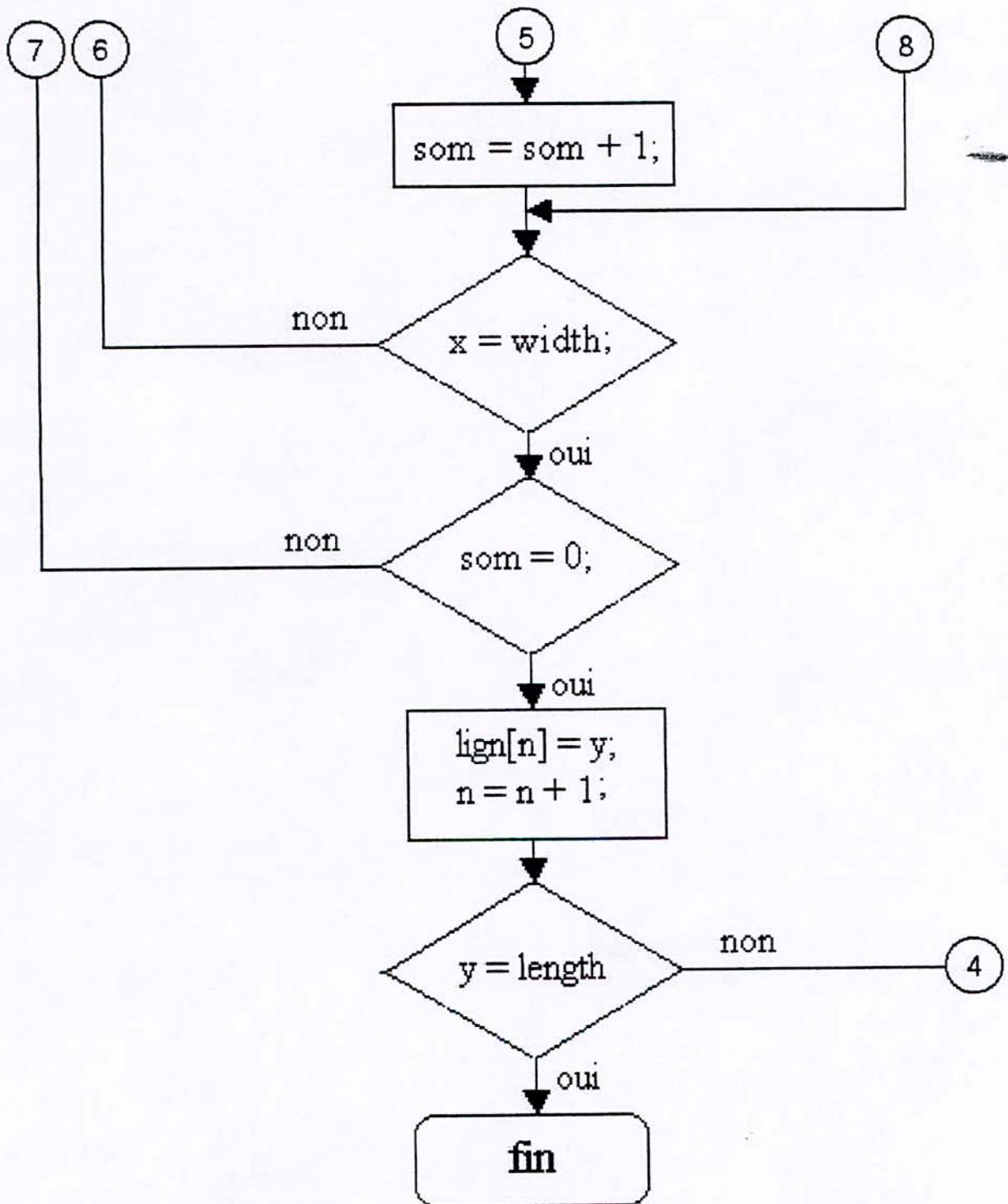


Figure IV.2.1 : Organigramme de la segmentation horizontale.

IV-2-3-Segmentation Verticale :

La segmentation verticale divise chaque ligne en mots ou en parties connexes (elle détecte le début et la fin des mots ou des parties connexes de chaque ligne)

Pour détecter le début et la fin d'un mot ou d'une partie connexe, nous calculons l'histogramme vertical de chaque ligne (ou la somme des pixels non blancs)

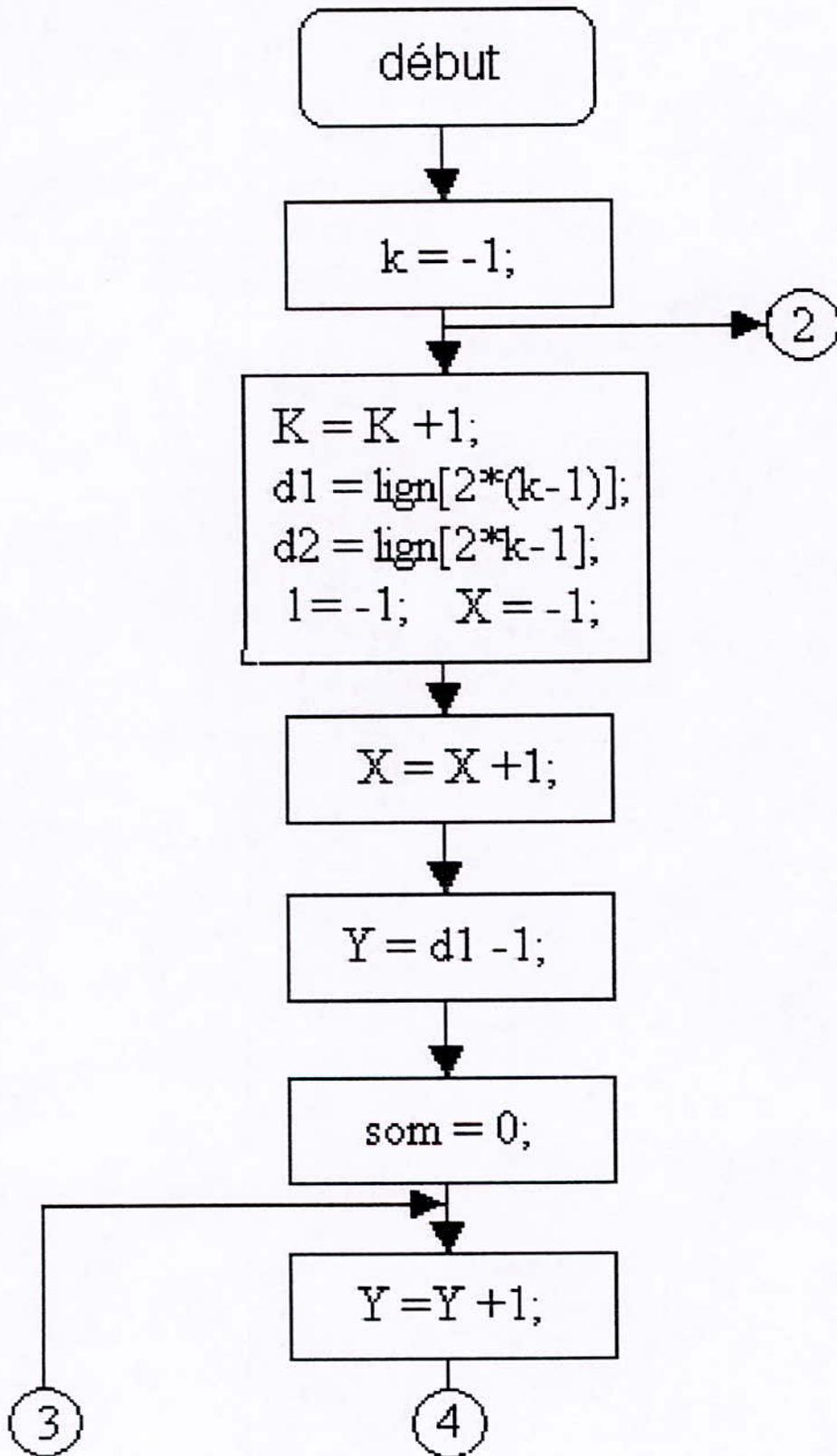
a- L'algorithme de la segmentation verticale

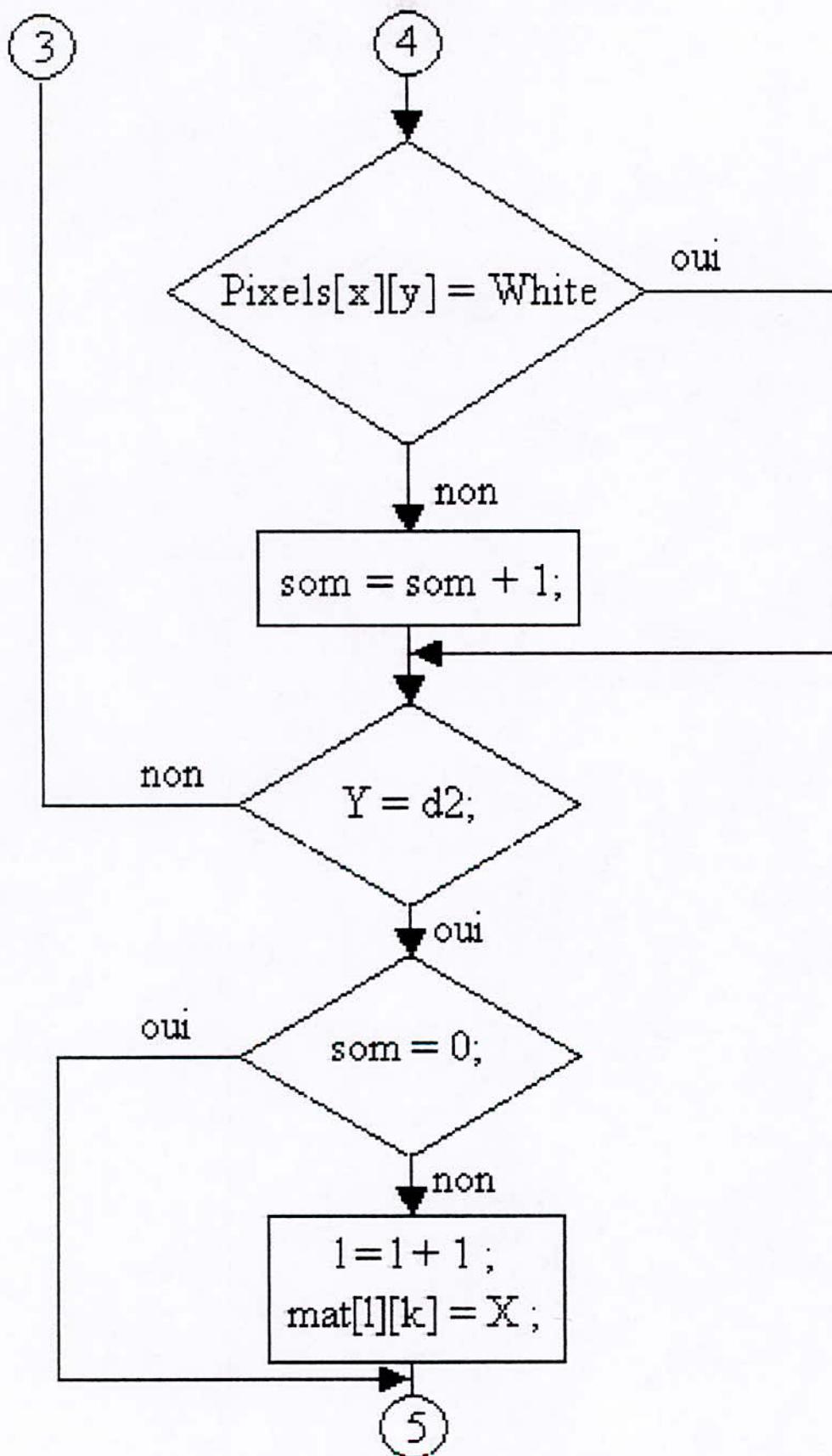
On calcule l'histogramme vertical de chaque ligne (ou la somme des pixels non blancs), jusqu' à ce que l'histogramme vertical (ou la somme des pixels non blancs) soit supérieur ou égale à un, on aura alors le début du $j^{\text{ème}}$ mot de la $i^{\text{ème}}$ ligne, ensuite on calcul l'histogramme vertical de la $i^{\text{ème}}$ ligne (ou la somme des pixels non blancs), jusqu' à ce que l'histogramme vertical (ou la somme des pixels non blancs) soit égal à zéro, on aura alors la fin du $j^{\text{ème}}$ mot de la $i^{\text{ème}}$ ligne.

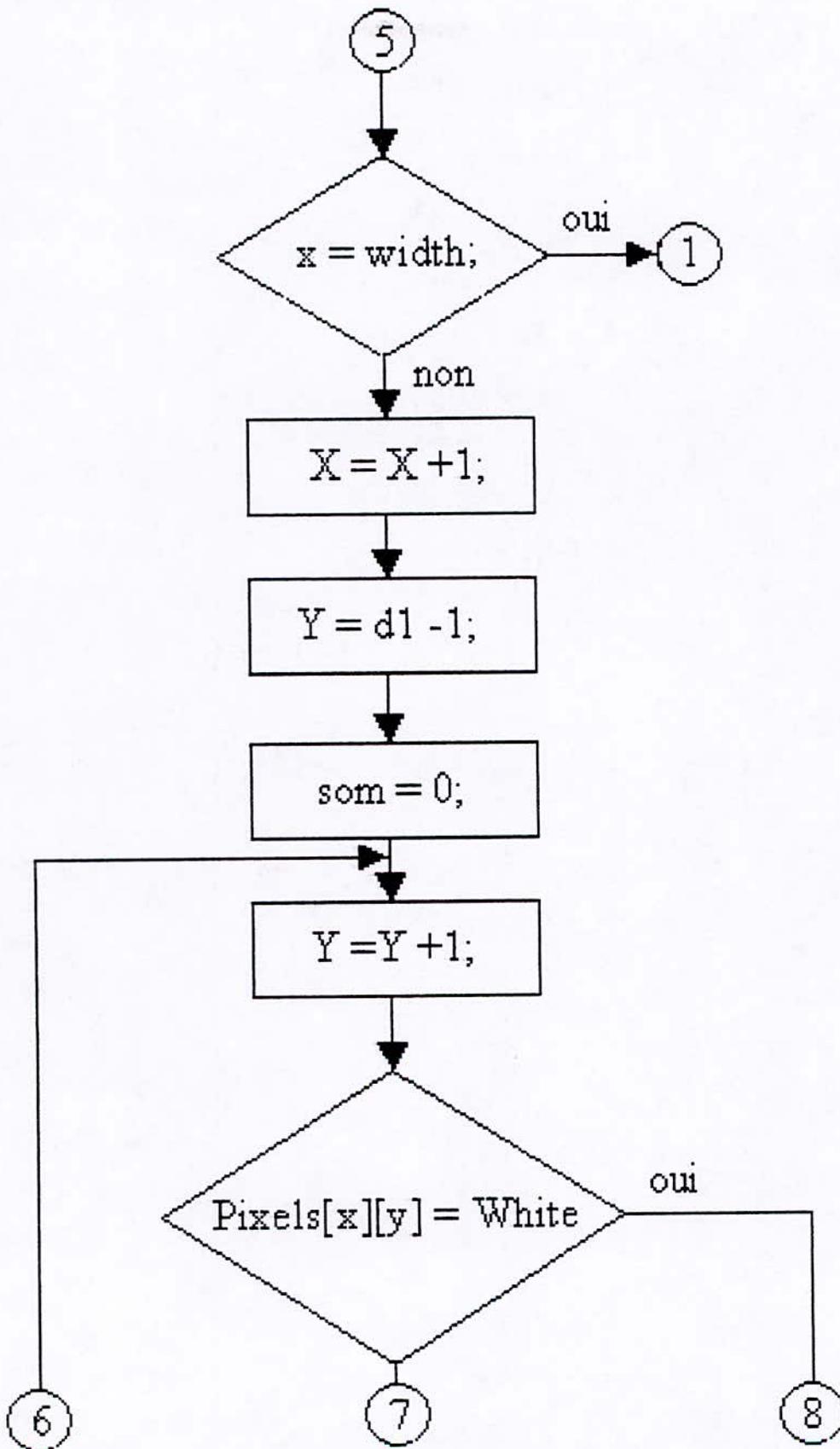
b- Organigramme de la segmentation verticale(Figure IV.2.2)

notation :

length = la longueur de l'image ;
width = la largeur de l'image ;
pixels[x][y] = la valeur de la chrominance d'un pixel de coordonnées(x , y) ;
lign[2*(i-1)] = début de la ligne i ;
lign[2*i+1] = fin de la ligne i ;
mat[2*(i-1)][k] = début de la partie connexe i de la ligne k ;
mat[2*i+1][k] = début de la partie connexe i+1 de la ligne k ;
n = le numéro de la ligne ;







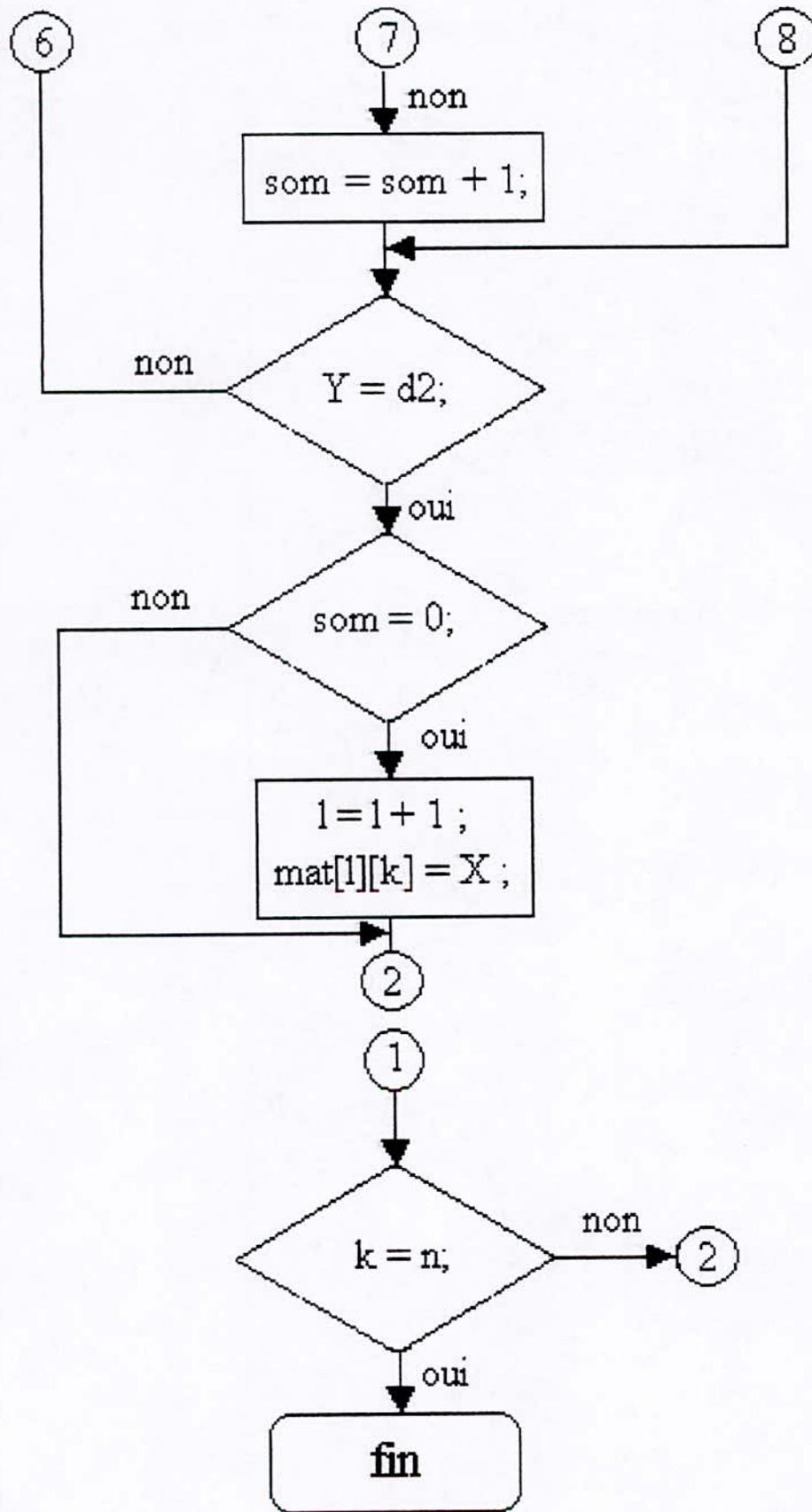


Figure IV.2.2 : Organigramme de la segmentation verticale.

VI-2-4-Segmentation en Caractères :

La segmentation en caractères est l'étape la plus difficile, elle sert à séparer les caractères dans une partie connexe (elle détecte le début et la fin des caractères de chaque mot dans une partie connexe)

a- Algorithme de la segmentation en caractères

La segmentation en caractères comporte trois étapes qui sont :

1-La détection de la ligne médiane :

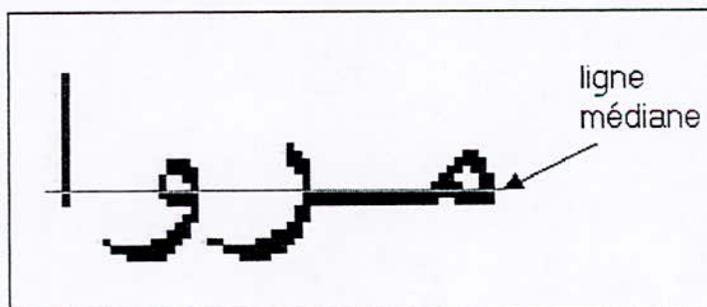
La ligne médiane constitue la région qui a la plus forte concentration des pixels en ligne. Pour détecter la ligne médiane on procède par un balayage horizontal des lignes, donc nous utilisons l'histogramme horizontal.

Remarque

La détection de la ligne médiane n'est pas faite pour chaque partie connexe mais pour chaque ligne, parce que chaque partie connexe a sa propre ligne médiane et elle n'est pas forcément proche de la ligne médiane.

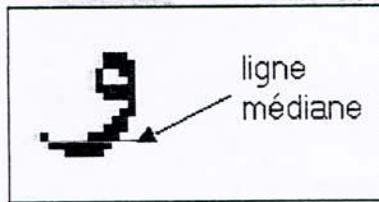
Exemple :

Si on a plusieurs parties connexes comme suit :



On remarque que la ligne médiane est au milieu de la ligne.

Si on prend par exemple le caractère (و)

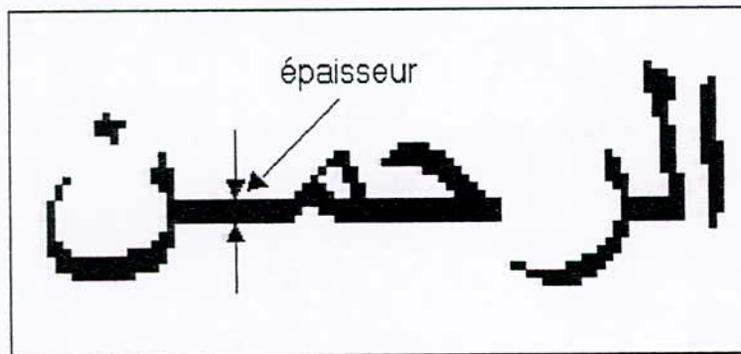


On remarque que la ligne médiane est en dessous du milieu de la ligne.

2-calcul de l'épaisseur de la ligne de jonction :

L'épaisseur de la ligne de jonction permet de connaître la taille de l'écriture, on prend chaque ligne et on calcule l'histogramme vertical de cette ligne, par la suite on met le nombre des pixels non blancs dans un vecteur (la dimension de ce vecteur est égale à la longueur de l'image traitée). Par conséquent, l'épaisseur est égale à l'élément qui se répétera plusieurs fois.

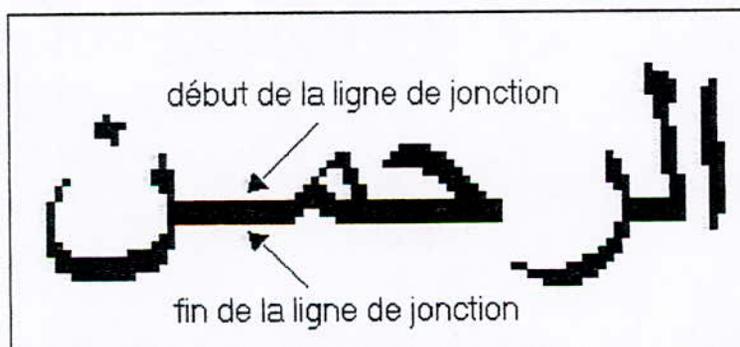
Exemple : l'épaisseur d'une ligne :



3- Séparation des caractères

Après avoir calculé l'épaisseur et la ligne médiane on déduit le début et la fin de la ligne de jonction

Exemple : le début et la fin de la ligne de jonction pour une ligne

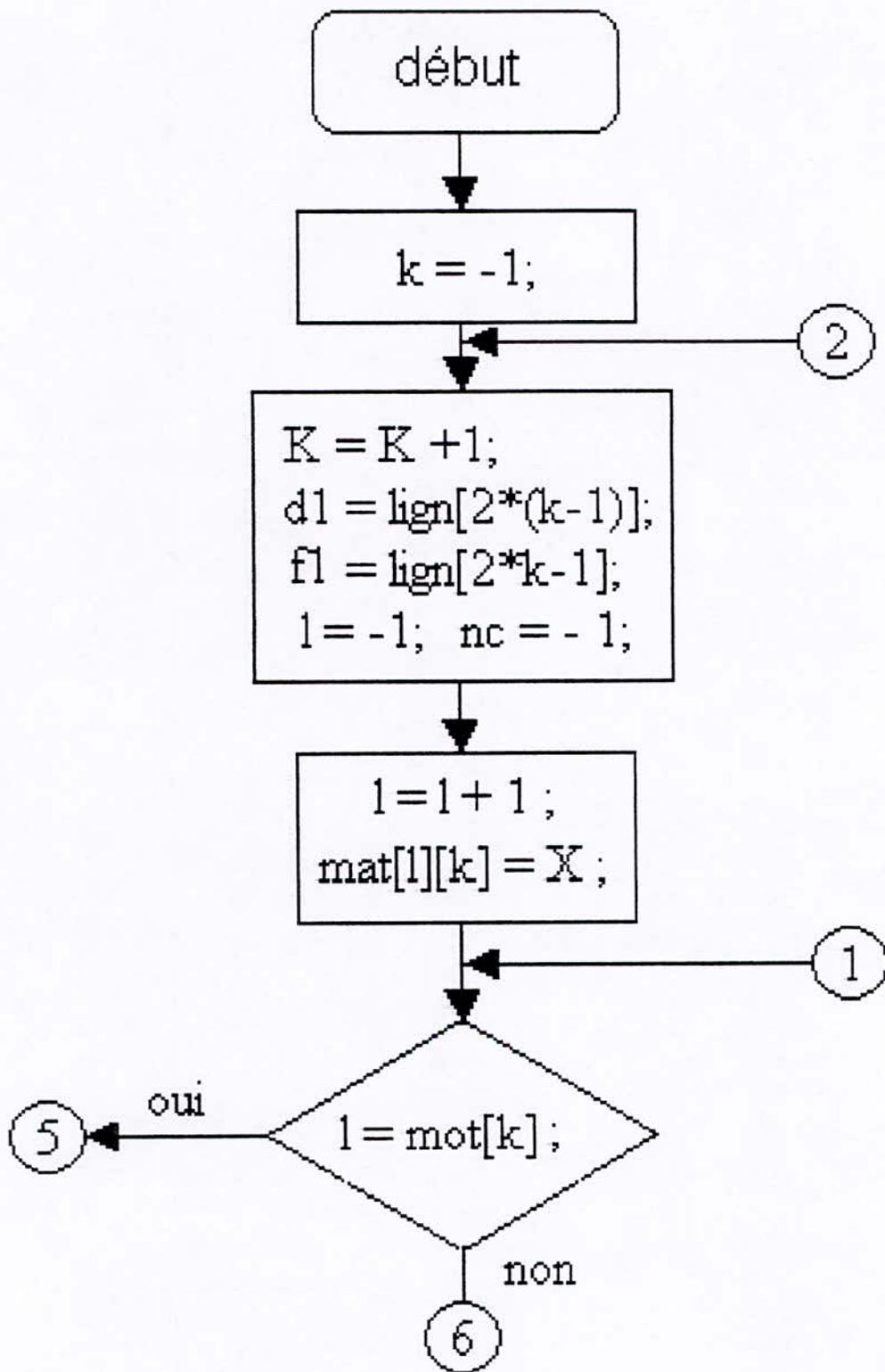


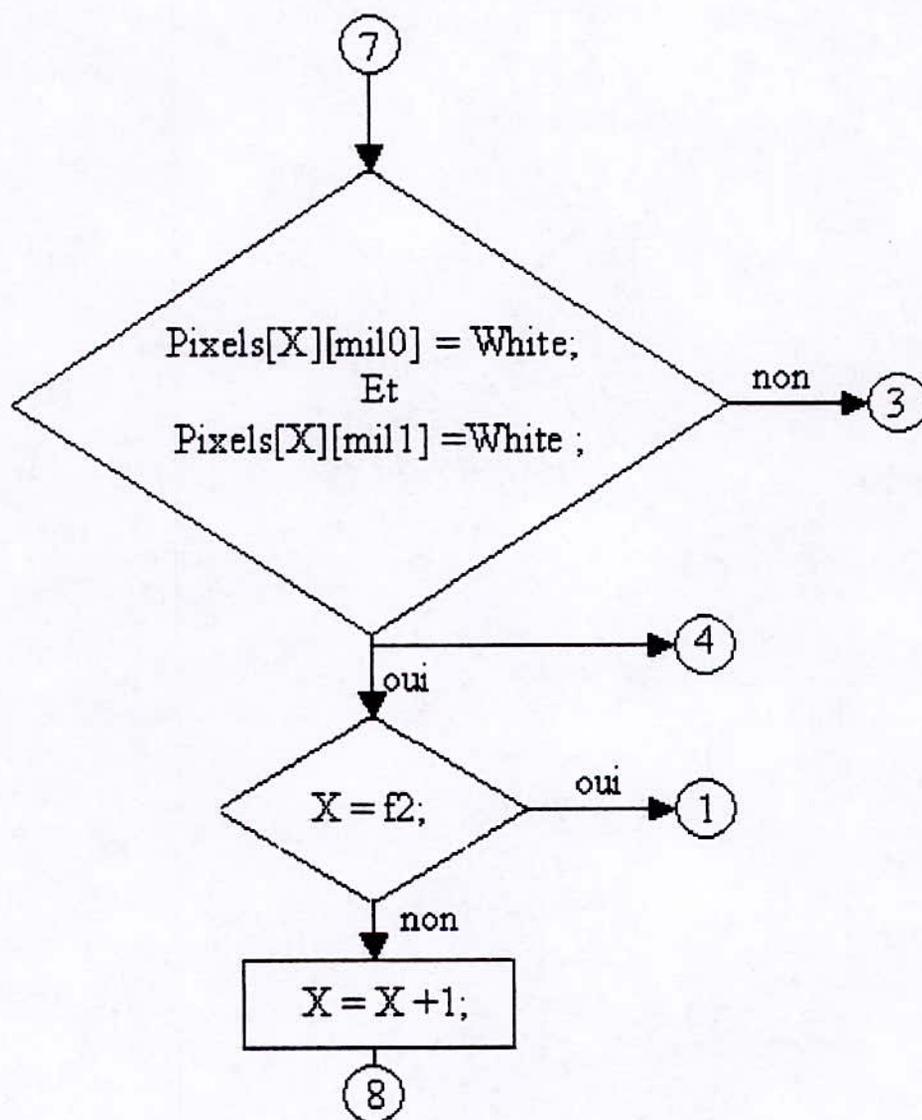
Pour séparer les caractères on fait deux balayages horizontaux de droite à gauche pour chaque partie connexe du début à la fin de la ligne de jonction. Si on détecte un pixel non blanc on a un début du $i^{\text{ème}}$ caractère et une fin du $(i-1)^{\text{ème}}$ caractère, ensuite on fait le même balayage jusqu'à ce qu'on détecte un pixel blanc etc....

b- Organigramme de segmentation en caractères(Figure IV.2.3)

notation :

length = la longueur de l'image ;
width = la largeur de l'image ;
pixels[x][y] = la valeur de la chrominance d'un pixel de coordonnées (x , y) ;
lign[2*(i-1)] = début de la ligne i ;
lign[2*i+1] = fin de la ligne i ;
mat[2*(i-1)][k] = début de la partie connexe i de la ligne k ;
mat[2*i+1][k] = début de la partie connexe i+1 de la ligne k ;
n = le nombre de lignes ;
ep1[k] = épaisseur de la ligne k ;
mil = la ligne médiane ;
mil0 = début de la ligne de jonction ;
mil1 = fin de la ligne de jonction ;
elph[i][k]= début ou fin d'un caractère de la ligne k ;
mot[k] = le nombre des parties connexes pour la ligne k ;





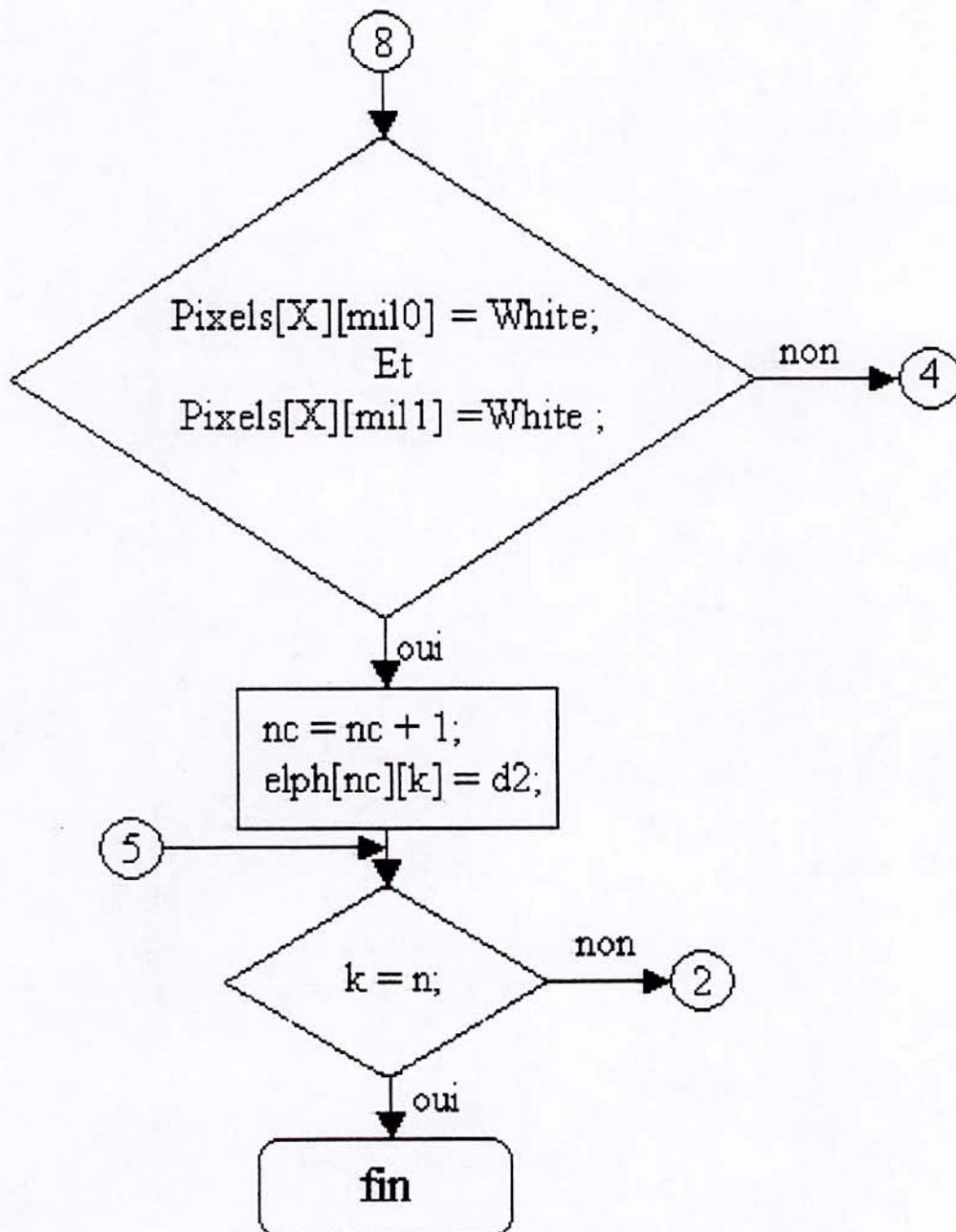


Figure IV.2.3 : Segmentation en caractères.

VI-2-5- Problèmes de la segmentation et les Solutions proposées :

Nous avons testé cette méthode sur plusieurs textes arabes, et nous avons découvert plusieurs problèmes qui sont les suivants:

- Le caractère "س" est segmenté en trois portions



La solution proposée :

Notre algorithme segmente le caractère "sin" en trois parties, et on remarque que la distance entre le début et la fin de la première partie est très petite devant les autres caractères, donc si la longueur d'une partie est petite par rapport à un certain seuil ($2 * (\text{épaisseur de la ligne médiane})$) on ignore la première et la deuxième partie.

- les caractères "م" et "ط".

Notre algorithme segmente ces caractères en deux parties



La solution proposée :

On remarque que tous ces caractères sont bouclés, donc si on détecte une boucle fermée on ignore la partie détectée.

- les caractères "ص" et "ض"

Notre algorithme segmente ces caractères en trois parties



La solution proposée :

On remarque que ce problème, est une addition entre les deux problèmes précédents, donc si on détecte une boucle fermée on ignore la première et la deuxième partie détectée.

IV-3-Reconnaissance

IV-3-1-La reconnaissance [6] :

C'est la dernière étape du processus d'analyse d'images. La reconnaissance est un problème de classement d'un objet donné parmi un ensemble de classes, dont chacune est caractérisée par un jeu de paramètres essentiellement mathématiques. Elle se compose d'une phase d'apprentissage et d'une phase de reconnaissance.

Avant de pouvoir reconnaître une forme, il faut d'abord connaître un modèle représenté par un ensemble de caractéristiques afin de pouvoir faire une comparaison. Ces données devront être mémorisées dans une base de données. La superposition avec l'un de ces modèles peut se faire de deux manières :

- Superposition de formes.
- Détection de similitudes.

La première solution consisterait à superposer l'image à reconnaître avec un modèle préalablement appris et stocké dans une base de données d'images. L'inconvénient est que la méthode est bonne pour des objets simples, bien définis et se présentant de la même façon (même orientation, même échelle). De plus, la base de données est grande. Ceci nous amène à utiliser la notion de distance entre images (mesure de dissemblance entre l'image et le modèle). L'image est identique au modèle si la distance est inférieure à un certain seuil ou inférieure à la distance entre la même image et chacun des autres modèles.

L'inconvénient est que la distance n'est pas invariante vis-à-vis d'un déplacement (rotation ou translation) de l'image, d'une déformation.

La seconde solution consiste à passer par une représentation des connaissances, ce sont donc des systèmes à base de connaissances. Chaque objet est défini comme un ensemble de primitives ou paramètres qui permettent de déterminer ce que représente l'objet de l'image. Ces paramètres sont considérés comme une signature de l'objet ; c'est cette dernière méthode qui a été adoptée.

IV-3-2-Caractéristiques de l'alphabet arabe :

L'alphabet arabe comporte 28 caractères de base plus la hamza qui a un rôle très important dans la phonétique et le lem-alif qui est en réalité un caractère composé de lem et alif.

Chaque caractère de l'alphabet arabe peut s'écrire sous 4 formes selon sa position dans le mot : début, milieu, fin ou isolé.

Contrairement aux caractères latins, l'Arabe n'a pas de voyelles mais plutôt des ponctuations ou signes diacritiques qui peuvent être placées au-dessus ou bien au-dessous du caractère.

La présence de groupes de points et leurs positions jouent un rôle prépondérant pour la discrimination des caractères possédant le même tracé et appartenant à la même classe.

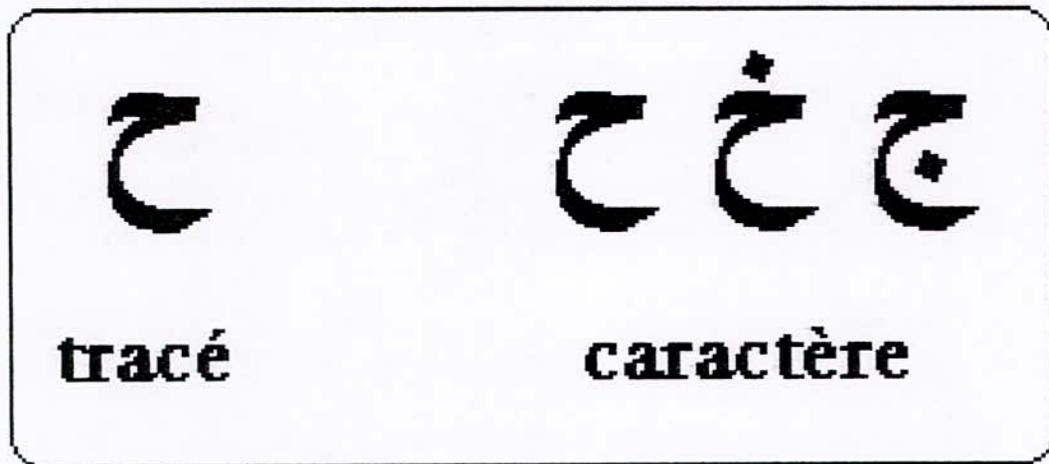


Figure-IV-3-1: caractères ayant le même tracé principal

IV-3-3-Description de la méthode :

La méthode procède en deux niveaux, on commence par calculer les primitives puis on passe à l'apprentissage où, à la reconnaissance.

1- Extraction des primitives :

On a choisi six primitives : une principale et cinq secondaires qui sont :

- La classe (primitive principale).
- Position des points.
- Nombre de points .
- La forme du cadre du caractère.
- Le taux de remplissage des 4 coins.

- La position du caractère dans le mot.

a- Calcul de la classe :

La classe nous permet de classer le caractère en prenant compte des caractéristiques morphologiques du tracé du caractère, elle est calculée de la façon suivante (convention) :

$$\text{Classe} = BC + (G*4) + (D*16) + (B*64) + (H*256)$$

Où :

BC : représente le nombre de boucles.

G : représente le nombre de concavités vers la gauche.

D : représente le nombre de concavités vers la droite.

B : représente le nombre de concavités vers le bas.

H : représente le nombre de concavités vers le haut.

Une concavité est une partie du tracé du caractère qui est courbée vers l'intérieur, elle peut être orientée dans plusieurs sens : bas, haut, gauche ou droit comme il est montré sur la figure IV.3.2.

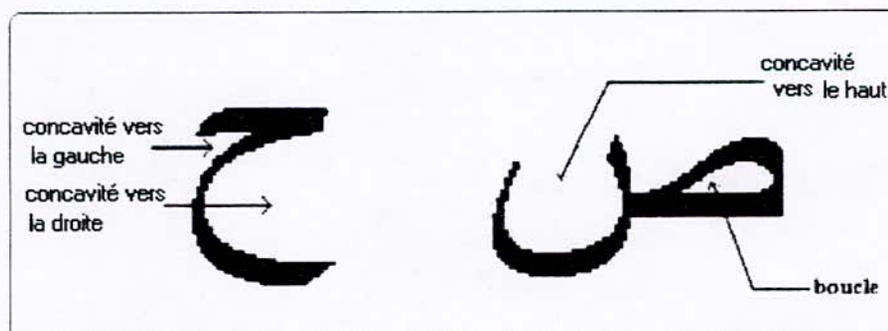


Figure-IV-3-2: Exemple de concavités

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 IV.3.3:

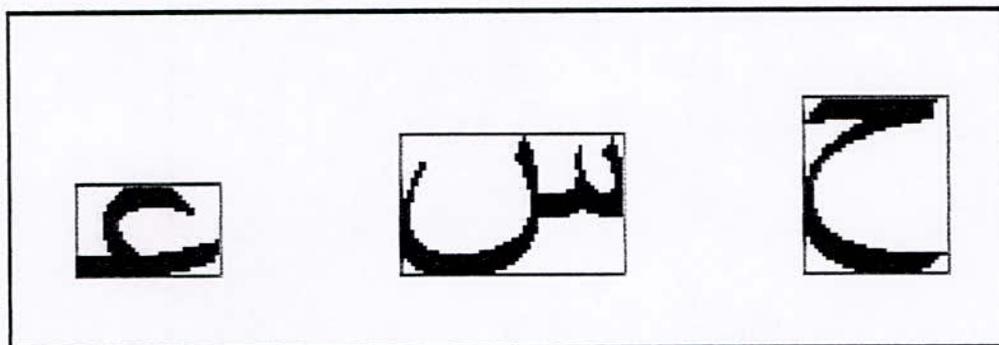


Figure-IV-3-3: encadrement du caractère.

On procède ensuite à un balayage horizontal puis vertical de toute la surface contenant le caractère, pendant les balayages on sauvegarde les positions des lignes et colonnes qui présentent plus de deux de transitions (noir-blanc; blanc-noir) car l'existence de plus de deux 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 P_1 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 $P_1 \diamond 0$ alors

Sauvegarder P_1 ;

Sauvegarder $j_1=j$; { j_1 est l'ordonnée 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 ($P_1 < 0$) **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.

Les deux exemples des figures IV.3.4 et IV.3.5 montre le résultat des deux balayages:

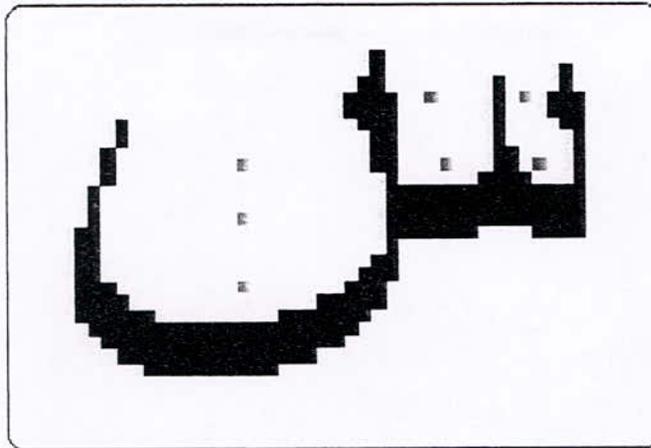


Figure-IV-3-4: Points trouvés après le balayage horizontal.

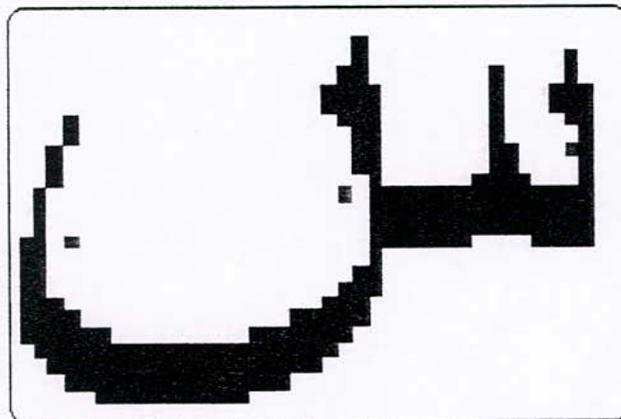


Figure-IV-3-5: Points trouvés après le balayage vertical.

Après les deux balayages chaque concavité et/ou boucle seront représentés par plusieurs points, ce qui pourra fausser la détection ; pour cela il faut éliminer les points en plus et ne garder qu'un point représentatif (point essentiel) par concavité et/ou boucle.

Pour garder ça 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.

L'exemple de la figure IV.3.6 montre les points essentiels trouvés :

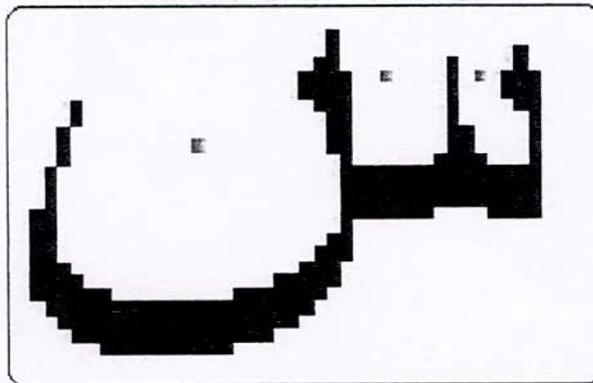


Figure-IV-3-6: Points essentiels trouvés.

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 coupe 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).

Exemple :

Si on croise le caractère dans les huit sens donc le point représente une boucle.

Si on croise le caractère dans les sens haut droit, bas droit et droit alors le point représente une concavité vers la gauche.

Une fois que l'on a identifié ce que tous les points essentiels représentent, on calcule alors la classe du caractère.

Cette méthode de classification a été utilisée par M.B EL KURDY,

Problème rencontré:

Cette méthode présente un problème avec les caractères de petite taille où le contour du caractère présente une discontinuité, comme on peut le voir sur le figure IV.3.7:

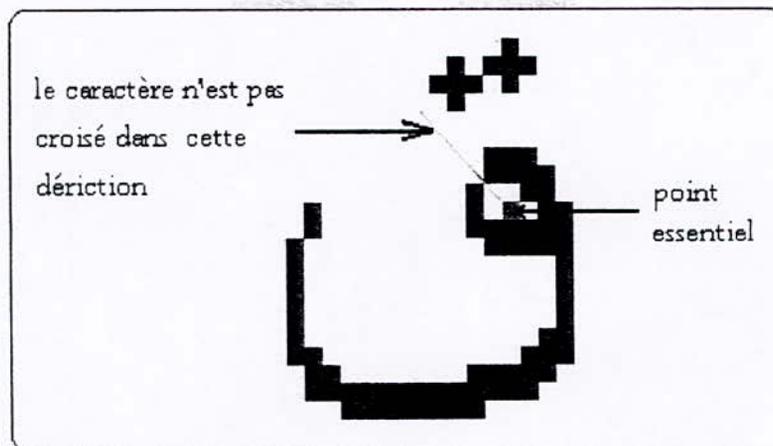


Figure-IV-3-7: problème des contours discontinus.

Pour surmonter ce problème on a utilisé un remplissage du contour, à chaque fois qu'il y a deux pixels voisins sur la même diagonale noire est sur l'autre diagonale les deux pixels sont blancs alors on remplace les pixels blancs par des pixels noirs.

b- Position des points :

Cette caractéristique comme les autres caractéristiques est très importante, dans certains cas elle est très difficile à extraire, pour cela on procède en deux étapes :

Si dans la première étape on détecte un point, deux points, trois points, ou une hamza, alors on ne passe pas à la seconde étape si non on passe à la deuxième étape.

Dans la première étape on fait un balayage de la surface du rectangle qui contient le caractère (le rectangle est obtenu après cadrage du caractère), si on trouve une ligne vide où tous les pixels sont éteints alors on peut conclure que le caractère se compose de deux entités, donc il y a présence de points ou d'une hamza.

Si après balayage on ne trouve aucune ligne vide ou tous les pixels sont éteints, on passe à la deuxième étape où/ on va voir s'il y a un point ou une hamza à l'intérieur du corps du caractère, comme c'est le cas pour les caractères djim ou kef isolé.

Dans cette étape, on fait un étiquetage [2] sur la surface du cadre qui limite le caractère pour voir s'il y a une partie connexe à l'intérieur du caractère.

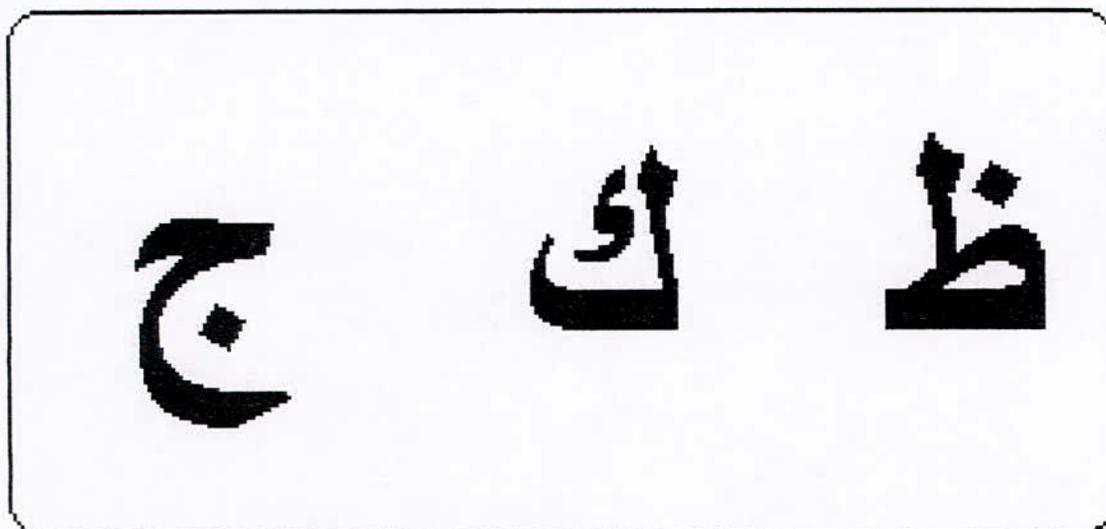


Figure-IV-3-8:caractères ayant leur point au milieu du corps

Si après étiquetage on trouve deux parties connexes on conclue qu'il y a un point ou une hamza dans le cas contraire on peut dire que le caractère n'est pas ponctué.

Si le caractère est ponctué, on calcule la surface des deux entités , l'évidence est que les points appartiennent à la plus petite surface , donc on compare les deux surfaces.

On définit une variable qu'on appelle Posit comme suit :

Posit=1 si le point est situé au dessus du caractère.

Posit=2 si le point est situé au milieu du caractère.

Posit=3 si le point est situé au dessous du caractère.

Posit=0 si le caractère n'est pas ponctué.

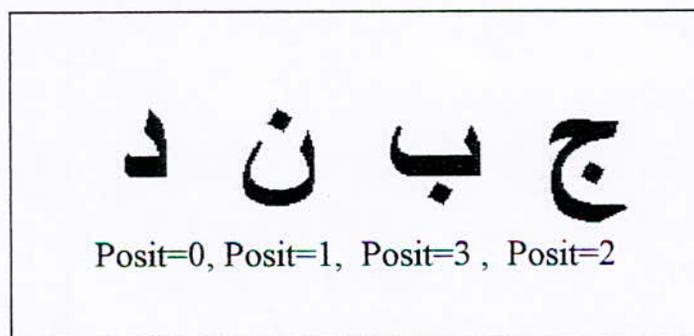


Figure-IV-3-9:Les différentes positions du point.

b- Nombre de points :

la deuxième primitive secondaire est le nombre de points du caractère. Pour extraire cette primitive représentée par la variable n-point, on calcule deux variables nbth nbtv qui représentent le nombre maximale de transitions (0-1 et 1-0) horizontales et verticales, dans l'entité représentant la ponctuation, si cette dernière existe (Posit>0).

n-point peut prendre les valeurs suivantes:

n-point=0 s'il y a pas de ponctuation.

n-point=1 s'il y a un point.

n-point=2 s'il y a deux points.

n-point=3 s'il y trois points.

n-point=4 s'il y a une hamza.

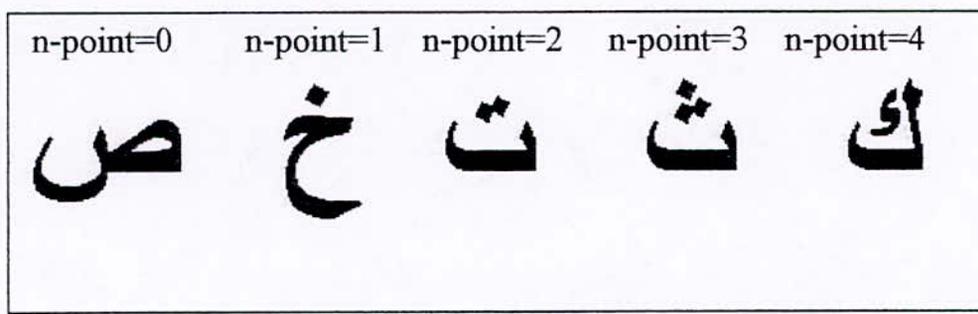


Figure-IV-3-10: Les différents nombres de points.

- **Cas d'un point :**

Pour un point on a nbth=2 et nbtv=2.

- **Cas de deux points :**

Pour deux points on a nbth=4 et nbtv=2.

- **Cas de trois points :**

Pour trios points on a nbth=4 et nbtv=4.

• Cas de hamza :

Pour la hamza on a nbth=2 et nbtv=4.

c- La forme:

La forme du caractère peut être carrée, debout ou allongée.
 Pour extraire cette primitive on calcule le rapport :

$$R=W/H.$$

Où W et H sont respectivement la largeur et la hauteur du caractère

On définit une variable form qui exprime la forme du caractère comme suit :

form=1 si la forme du caractère est allongée.

form=2 si la forme du caractère est carrée.

form=3 si la forme du caractère est debout.

La variable form est liée à la variable R par les relations suivants :

form=1 si $R > \text{seuil2}$.

form=3 si $R < \text{seuil1}$.

form=2 si $\text{seuil2} < R < \text{seuil1}$.

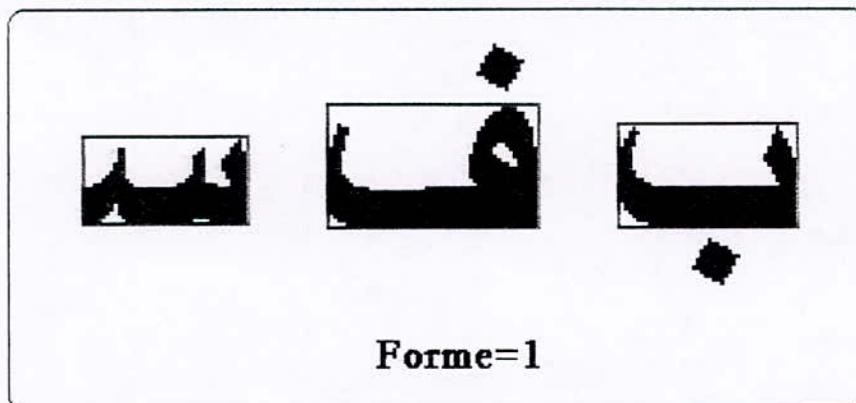


Figure-VI-3-11:Caractères allongés.

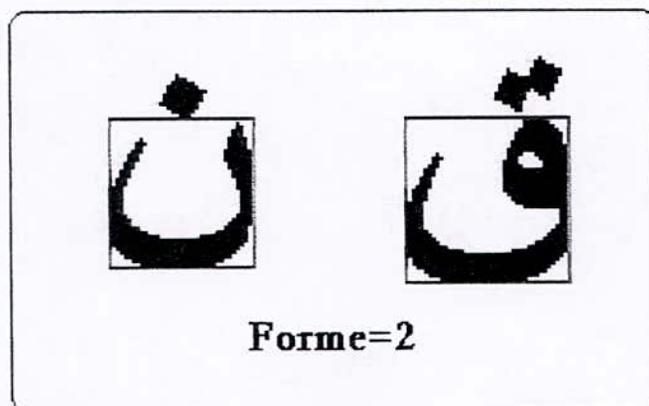


Figure-VI-3-12:Caractères carrés.

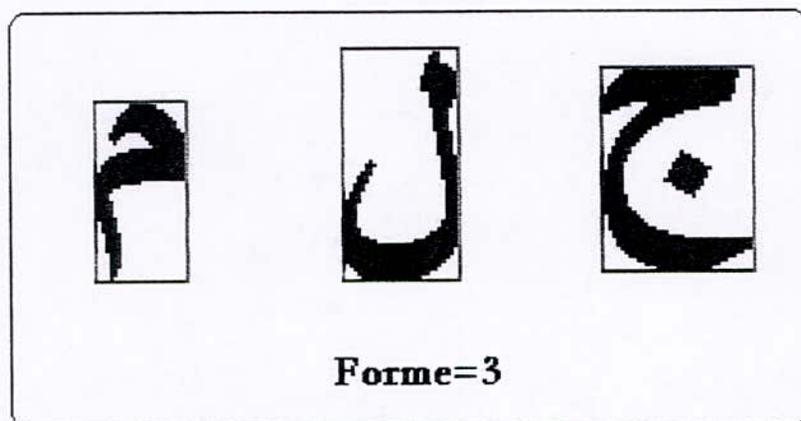


Figure-VI-3-13:Caractères debout.

Les seuils sont estimés après plusieurs expériences à :

- seuil 1=0.8
- seuil 2=1.2

d- Taux de remplissage des 4 coins :

Cette caractéristique nous donne une idée sur la répartition du caractère dans le plus petit rectangle qui le contient.

On définit une variable qui appelée corvar et qui est évaluée de la façon suivante :

On considère 4 carrées de dimension :

- $(W/4)*(W/4)$ si la forme du caractère est debout.
- $(H/4)*(H/4)$ si la forme du caractère est allongée ou carrée.

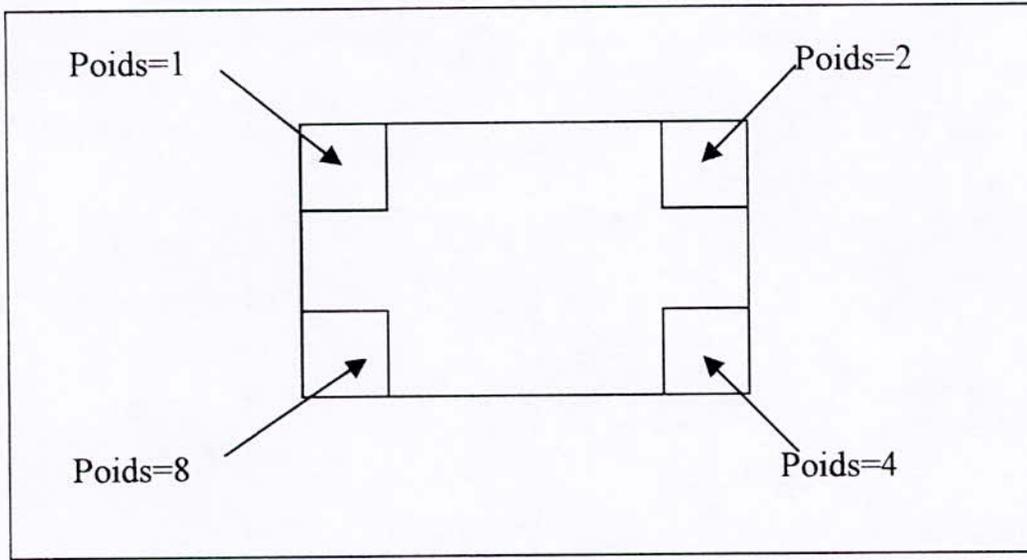


Figure-VI-3-14: Corvar, Taux de remplissage des 4 coins

Le taux de remplissage d'un coin est défini comme étant le rapport entre le nombre de pixels noirs sur le nombre total de pixels dans ce carré, un coin est considéré plein si le taux de remplissage est ≥ 0.5 , ce seuil est estimé après plusieurs tests .

On évalue la variable corvar comme suit (convention):

$$\text{Corvar} = \text{taux}(\text{haut_gauche}) + 2 * \text{taux}(\text{haut_droit}) + 4 * \text{taux}(\text{bas_droit}) + 8 * \text{taux}(\text{bas_gauche}).$$

Il est à noter que la variable sera comprise entre 0 et 15.

e- Position du caractère dans le mot :

Un caractère arabe peut avoir au maximum 4 positions : au début du mot, au milieu, à la fin ou isolé.

On définit une variable pos qui prend les valeurs suivante :

Pos = 0 si le caractère est isole.

Pos =1 si le caractère est au debout du mot.

.Pos =2 si le caractère est au milieu du mot.

Pos =3 si le caractère est à la fin du mot

Avec ces primitives on définit un vecteur de six éléments.

V[0]	V[1]	V[2]	V[3]	V[4]	V[5]
------	------	------	------	------	------

Où :

V[0] = classe, V[1] = n_point, V[2] = posit, V[3] = pos, V[4] = corvar,

V[5] = forme .

2- Reconnaissance et Apprentissage:

Dans cette dernière partie nous avons utilisé deux méthodes, la première est une reconnaissance par comparaison de chaînes et la deuxième est une reconnaissance par réseaux de neurones.

2-1-Reconnaissance par comparaison de chaînes:

Dans cette méthode on utilise un apprentissage supervisé, on prend chaque caractère, on extrait ses caractéristiques (primitives) puis on le range dans un dictionnaire.

La reconnaissance est basée sur la notion de distance, qui est calculée entre le caractère à reconnaître et les caractères prototypes (de référence) qui sont stockés dans le dictionnaire.

Le caractère inconnu est identifié au prototype qui lui ressemble le plus, c-a-d celui qui a la plus petite distance avec lui.

Pour cela on définit quatre distances Dist1, Dist2, Dist3 et Dist4 comme suit :

- **Dist1:**

$$\text{Dist1}[i] = \sum_{j=1}^5 \text{abs} (V[j] - V_i[j])$$

Où :

$V[j]$: la $j^{\text{ème}}$ caractéristique du caractère inconnu.

$V_i[j]$: la $j^{\text{ème}}$ caractéristique du $i^{\text{ème}}$ prototype.

- **Dist2:**

$$\text{Dist2}[i] = \text{abs} (V[5] - V_i[5])$$

Où :

$V[5]$: la forme du caractère inconnu.

$V_i[5]$: la forme du $i^{\text{ème}}$ prototype.

Pour que la forme d'un prototype ressemble à la forme du caractère inconnu il faut que :

$$\text{Dist2}[i] \leq 1$$

La condition (\leq) est due au fait qu'un caractère qui s'écrit sous une forme allongée (forme = 1) peut être trouvé sous forme carrée (forme = 2) mais pas sous forme debout (forme = 3), l'inverse est aussi vrai.

- **Dist3:**

La troisième distance est calculée à partir de deux variables qui sont : la corvar du caractère inconnu ($V[4]$) et celle du prototype. Comme ces deux variables sont binaires, cette distance est calculée de la façon suivante :

$$\text{Dist3}[i] = F(A[i])$$

Où

$$A[i] = V[4] \text{ XOR } V_i[4]$$

- **Dist4:**

$$\text{Dist4}[i] = \sum_{j=1}^3 \text{abs}(V[j] - V_i[j])$$

a)-Description de la méthode de reconnaissance:

- extraction des primitives du caractère à reconnaître.
- chargement de tous les prototypes qui ont la même classe que le caractère inconnu.
- calcul de la 1^{ère} distance entre le caractère et chaque prototype chargé.
- chargement de tous les prototypes qui vérifient la condition: $\text{Dist1}[i]=0$, dans une liste appelée list1.
- voir si le nombre d'éléments (k_1) de la list1 est égal à '1' donc le caractère est identifié.

Si $k_1 > 1$ alors on vérifie si tous les prototypes qui vérifient $\text{Dist1}[i]=0$ décrivent le même caractère, si oui alors le caractère est reconnu, sinon il y a ambiguïté.

Si $k_1 = 0$ alors on calcule les distances: Dist4 , entre le caractère et chaque prototype chargé et on met dans une 2^{ème} liste appelée list2, tous les prototypes qui vérifient la condition : $\text{Dist4}[i]=0$.

Si le nombre d'éléments (k_2) de la list2 est égal à '1' donc le caractère est reconnu.

Si $k_2 > 1$ alors on vérifie si tous les prototypes qui vérifient les deux conditions: $\text{Dist2}[i] \leq 1$ et $\text{Dist3}[i] \leq 1$, décrivent le même caractère si oui alors le caractère est reconnu, sinon il y a ambiguïté.

Si $k_2 = 0$ alors le caractère est rejeté.

b)-Architecture du dictionnaire:

Le dictionnaire est une base de données qui contient les prototypes qui seront utilisées dans le programme de reconnaissance. Pour éviter que le caractère à identifier soit comparé avec tous les prototypes du dictionnaire, on a partitionné ce dernier en classes comme le montre la figure IV.3.15.

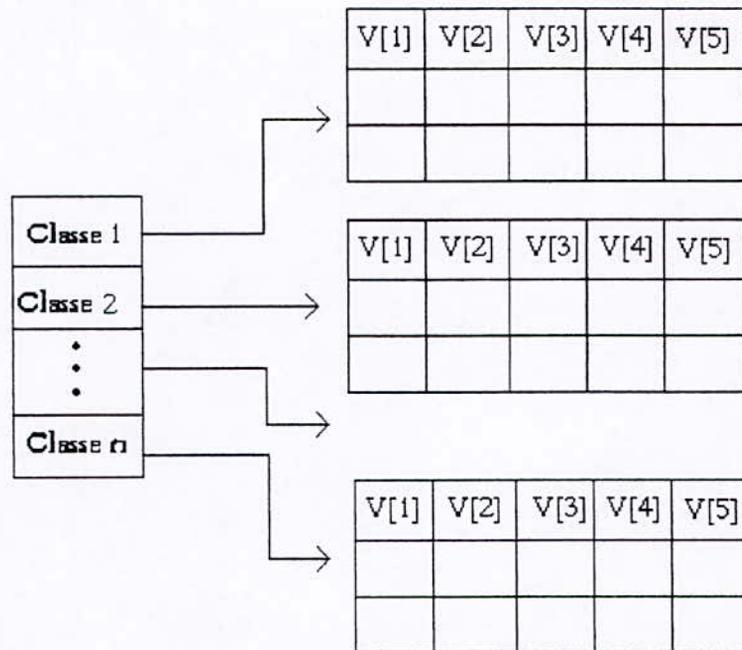


Figure-VI-3-15:architecture du dictionnaire

2-2-Reconnaissance par réseaux de neurones

Pour la reconnaissance par réseaux de neurones nous avons utilisé deux méthodes, l'une utilise un réseau de neurones global, et l'autre méthode utilise cinq réseaux, chaque réseau devant reconnaître un type bien précis de caractères.

1- Première méthode

Introduction:

Dans cette partie nous avons simulé un réseau de neurones multicouche, avec une couche cachée. Après avoir extrait les caractéristiques de chaque caractère, l'algorithme d'apprentissage consiste à adapter les poids et les biais du réseau pour que ses sorties soient suffisamment proches des sorties souhaitées (code du caractère), pour cette simulation on a utilisé le MATLAB, une fois

l'apprentissage terminé nous enregistrons les valeurs des poids de pondérations et les biais dans un fichier.

Finalement l'étape de décision ou de la reconnaissance proprement dite, consiste à conserver les mêmes paramètres du réseau simulé (nombre de couches, poids de pondérations et biais) et implémenté se réseau dans notre logiciel.

1-1-Préparation de l'ensemble d'apprentissage:

Au début de notre simulation du réseau nous avons utilisé un réseau à 5 entrées, où chaque entrée est la valeur en décimal d'une caractéristique du caractère, durant l'apprentissage (sur MATLAB) le réseau n'a pas convergé, nous avons donc converti les entrées décimales en binaire, et ainsi les entrées du réseau sont passées de 5 à 21, car on a codé la classe sur dix bits, la forme sur deux bits, la corvar sur quatre bits, la position du point sur deux bits et le nombre de points sur trois bits.

1-2-Architecture du réseau:

Le réseau utilisé a la structure suivante :

La couche d'entrée est formée de 21 neurones.

La couche cachée est aussi formée de 21 neurones.

Les neurones de ces deux couches ont les caractéristiques suivantes:

La fonction d'entrée est linéaire.

La fonction de seuillage est une sigmoïde centrée.

La fonction de sortie est la fonction identité.

la couche de sortie est formée de six neurones, ces derniers ont une fonction de seuillage identité, le choix de cette fonction a été fait après plusieurs simulations sur MATLAB .

la sortie du réseau est une valeur binaire qui correspond au code du caractère(voir tableau).

Caractère	Code	Code ASCII
	000000	10
	000001	205
	000010	27
	000011	28
	000100	29
	000101	218
Espace	000110	32
	000111	193
	001000	199
	001001	207
	001010	209
	001011	211
	001100	213
	001101	216
	001110	223
	001111	225
	010000	227
	010001	229
	010010	230

6	010011	236
5	010100	4
4	010101	200
3	010110	12
2	010111	206
1	011000	30
0	011001	31
7	011010	219
8	011011	204
9	011100	208
10	011101	210
11	011110	214
12	011111	217
13	100000	221
14	100001	223
15	100010	228
16	100011	5
17	100100	202
18	100101	201
19	100110	222

	100111	237
	101000	6
	101001	203
	101010	212
	101011	1
	101100	3
	101101	195
	101110	196
	101111	197
	110000	223

2- Deuxième méthode

Introduction:

Dans cette deuxième méthode nous avons simulé cinq réseaux de neurones multicouche avec une couche cachée, après avoir extrait les caractéristiques de chaque caractère, nous avons divisé ces caractéristiques en cinq groupes, chaque groupe englobe tous les caractères ayans le même nombre de points.

2-1-Architecture des cinq réseaux:

Les réseaux qui reconnaissent les caractères à 1,2 et 4 points ont 18 entrées qui sont les valeurs binaires des caractéristiques classe, forme, corvar et position du point.

Les réseaux qui reconnaissent les caractères à 0 et 3 points ont 16 entrées car on a supprimé la caractéristique position du point, qui est la même pour tous les caractères de même groupe (0 pour les caractères à 0 point, 1 pour les caractères à 3 points).

L'apprentissage a été fait sur MATLAB pour les cinq réseaux ; les poids de pondération et les biais de chaque réseau sont conservés dans un fichier qui va être utilisé pour l'implémentation des réseaux sur notre logiciel.

La structure des neurones des cinq réseaux est la suivante:

Les neurones de la couche d'entrée et de la couche cachée ont une fonction d'entrée linéaire, une fonction de seuillage sigmoïde et une fonction de sortie identité, sauf pour la couche de sortie qui a une fonction de seuillage identité.

2-2-Tables de codage des caractères suivant leurs groupes:

Premier réseau (caractère à 0 point):

Caractère	Code	Code ASCII
	000000	10
	000001	205
	000010	27
	000011	28
	000100	29
	000101	218
Espace	000110	32
	000111	193
	001000	199
	001001	207
	001010	209
	001011	211
	001100	213

Caractère	Code	Code ASCII
	001101	216
	001110	223
	001111	225
	010000	227
	010001	229
	010010	230
	010011	236

Pour le deuxième réseau (caractère à 1 point):

Caractère	Code	Code ASCII
	000000	4
	000001	200
	000010	12
	000011	206
	000100	30
	000101	31
	000110	219
	000111	204
	001000	208

Caractère	Code	Code ASCII
ا	001001	210
ب	001010	214
ت	001011	217
ث	001100	221
ج	001101	223
ح	001110	228

Le troisième réseau (caractère à 2 points):

Caractère	Code	Code ASCII
ا:	000000	5
ب:	000001	202
ت:	000010	201
ث:	000011	222
ج:	000100	237

Le quatrième réseau (caractère à 3 points):

Caractère	Code	Code ASCII
ا::	000000	6
ب::	000001	203
ت::	000010	212

Le cinquième réseau (caractère avec hamza : considérée à 4 points):

Caractère	Code	Code ASCII
	101011	1
	101100	3
	101101	195
	101110	196
	101111	197
	110000	223

Conclusion :

Dans ce chapitre on a exposé les trois modules qu'on a introduit dans notre logiciel à savoir : le module de pré-traitement, le module de segmentation et le module de reconnaissance.

Pour le pré-traitement, on a utilisé un filtre médian pour éliminer le bruit impulsionnel causé par le scanner et la qualité du papier, sans modifier la forme du caractère.

Pour la segmentation, une méthode algorithmique a été proposée et a donné des résultats satisfaisant.

Enfin, pour la reconnaissance, on a implémenté trois méthodes qui ont donné pour chacune d'elles un taux de reconnaissance proche de 97% ce qui est très satisfaisant.

Chapitre V : Présentation Du Logiciel

V-1 Introduction :

Le but principal de ce modeste travail, est la conception d'un logiciel de reconnaissance de caractères arabes imprimés. Pour atteindre ce but, nous avons été amené à utiliser deux types de logiciels de programmation à savoir : le logiciel Builder C++ 5.3 et le logiciel Matlab 6.1

Dans ce chapitre, nous allons tout d'abord présenter brièvement les différents logiciels utilisés, en expliquant leurs choix et les avantages qu'ils présentent. Puis nous passerons à la présentation de notre logiciel. Dans cette partie, nous allons exposer les différentes palettes (pages) qui constituent le logiciel et expliquer le rôle de chacune d'entre elles.

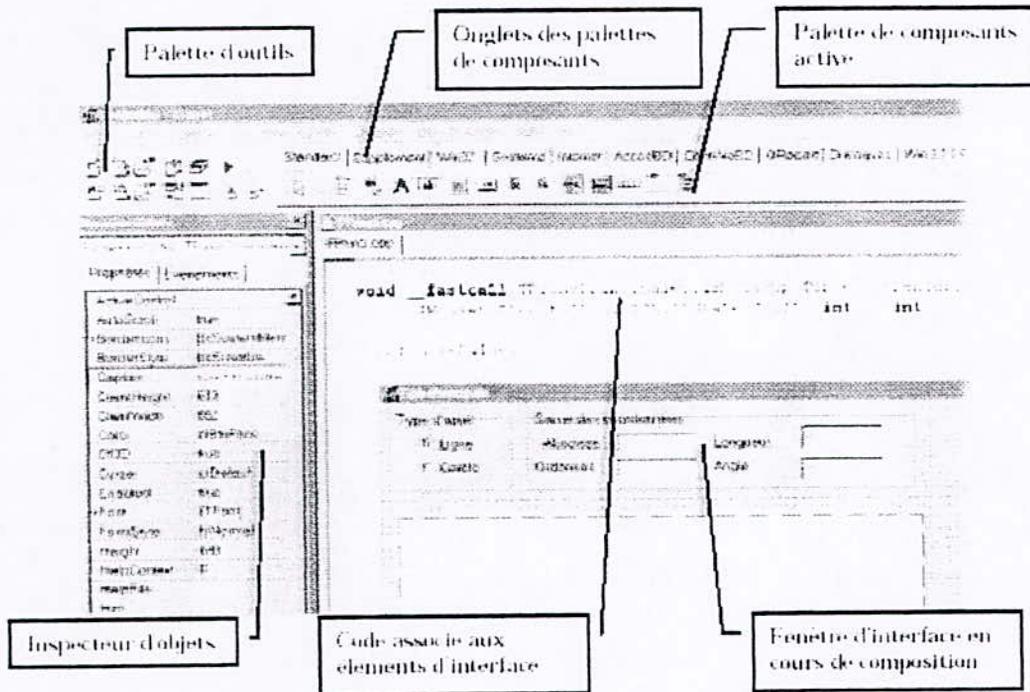
V-2 Logiciels utilisés :

V-2-1 Builder C++:

Tout d'abord C++ est un outil RAD, c'est à dire orienté vers le développement rapide d'applications (Rapid Application Development) sous Windows. En un mot, C++ Builder permet de réaliser de façon très simple l'interface des applications et de relier aisément le code utilisateur aux événements Windows, quelle que soit leur origine (souris, clavier, événement système, etc.) Pour ce faire, C++ Builder repose sur un ensemble très complet de *composants visuels* prêts à l'emploi. La quasi-totalité des contrôles de Windows (boutons, boîtes de saisies, listes déroulantes, menus et autres barres d'outils) y sont représentés, regroupés par famille. Leurs caractéristiques sont éditables directement dans une fenêtre spéciale intitulée *éditeur d'objets*. L'autre volet de cette même fenêtre permet d'associer le code au contrôle sélectionné. Il est possible d'ajouter à l'environnement de base des composants fournis par des sociétés tierces et même d'en créer soit même. Un outil RAD c'est également un ensemble de *squelettes* de projets qui permettent de créer plus facilement une application SDI () ou MDI (), une DLL (), des objets OLE (), etc. A chacun de ces squelettes est habituellement associé un *expert* qui par une série de boîtes de dialogues permet de fixer une partie des options essentielles à la réalisation du projet associé.

Ainsi, c'est pour la simplicité de son utilisation et pour la diversité des objets (prés à l'emploi) qu'il offre, que nous nous sommes intéressés au langage

Builder C++. La figure 5.1 représente l'espace de travail du langage Builder C++.



FigureV.1 : Espace de travail du langage Builder C++

V-2-3 Matlab :

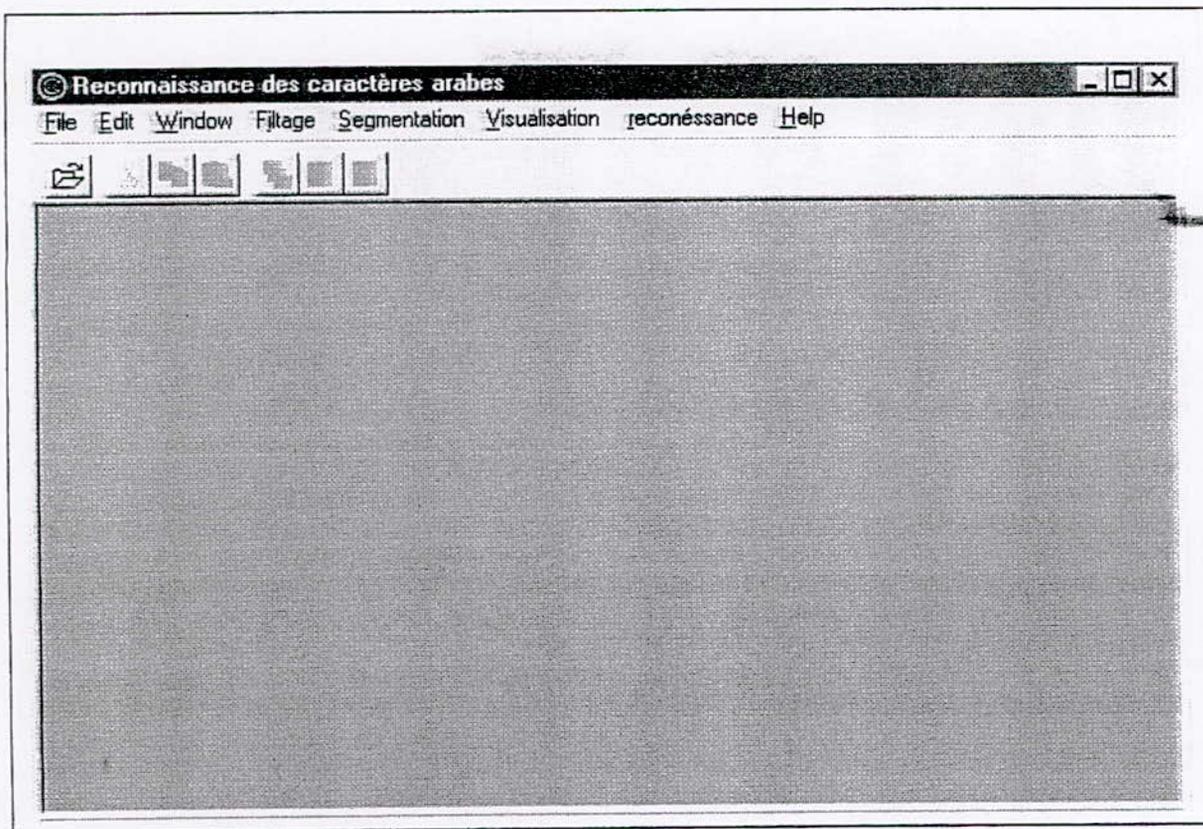
Le logiciel Matlab consiste en un langage interprété qui s'exécute dans une fenêtre dite d'exécution. L'intérêt de Matlab tient, d'une part, à sa simplicité d'utilisation : pas de compilation, déclaration implicite des variables utilisées et, d'autre part, à sa richesse fonctionnelle : arithmétique matriciel et nombreuses fonctions de haut niveau dans de nombreux domaines (analyse numérique, graphique, ...). La programmation sous Matlab consiste à écrire des *scripts* de commandes Matlab, exécutables dans la fenêtre d'exécution. En outre, grâce aux diverses *Toolboxes* spécialisés (ensemble de scripts Matlab), Matlab s'enrichit au fur et à mesure.

V-3-Interface développée :

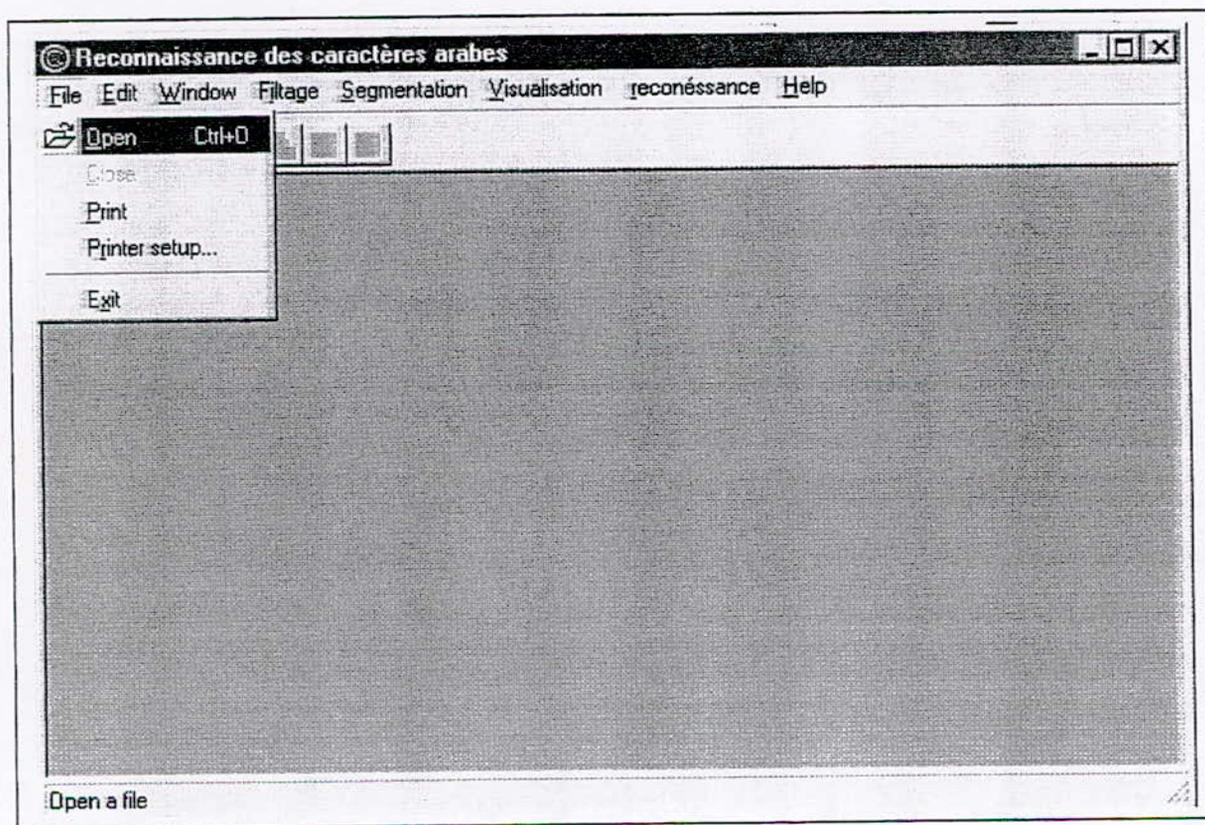
Ce logiciel est un système de reconnaissance d'écriture arabe imprimée, il traite les fichiers BMP, EMF et WMF .

Ce programme s'exécute sous Windows ce qui permet la facilité d'utilisation.

On donne ci dessous le résultat de l'exécution de notre programme.



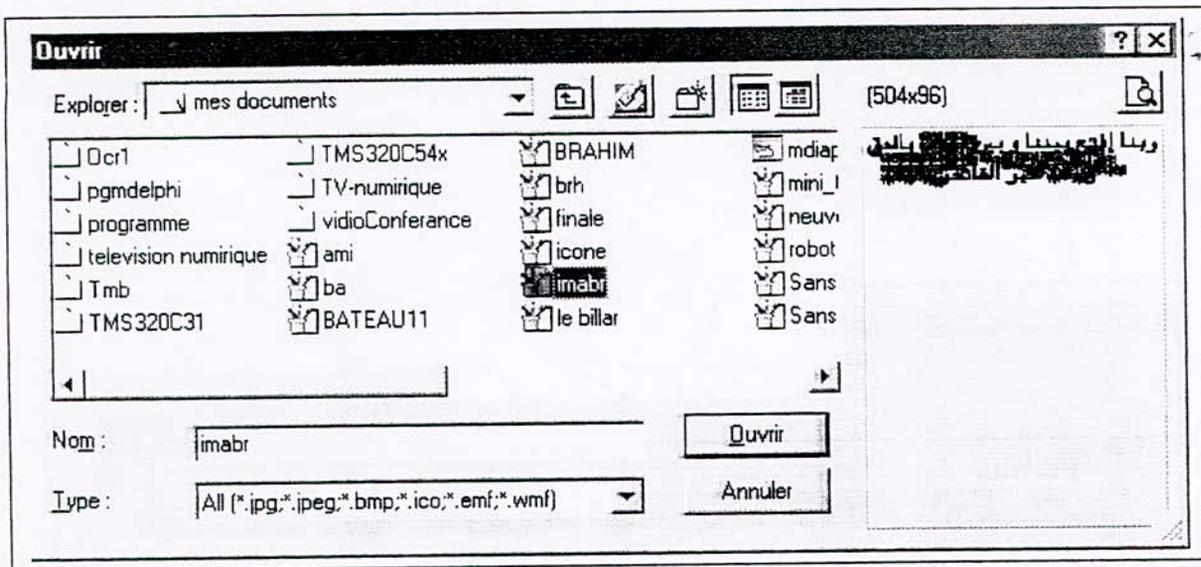
Un click sur le bouton OK fait apparaître la fenêtre principale du logiciel



Fenêtre principale du menu

Menu Fichier

Pour le chargement des fichiers, on clique sur l'article Open



Boîte de sélection de l'image à traiter

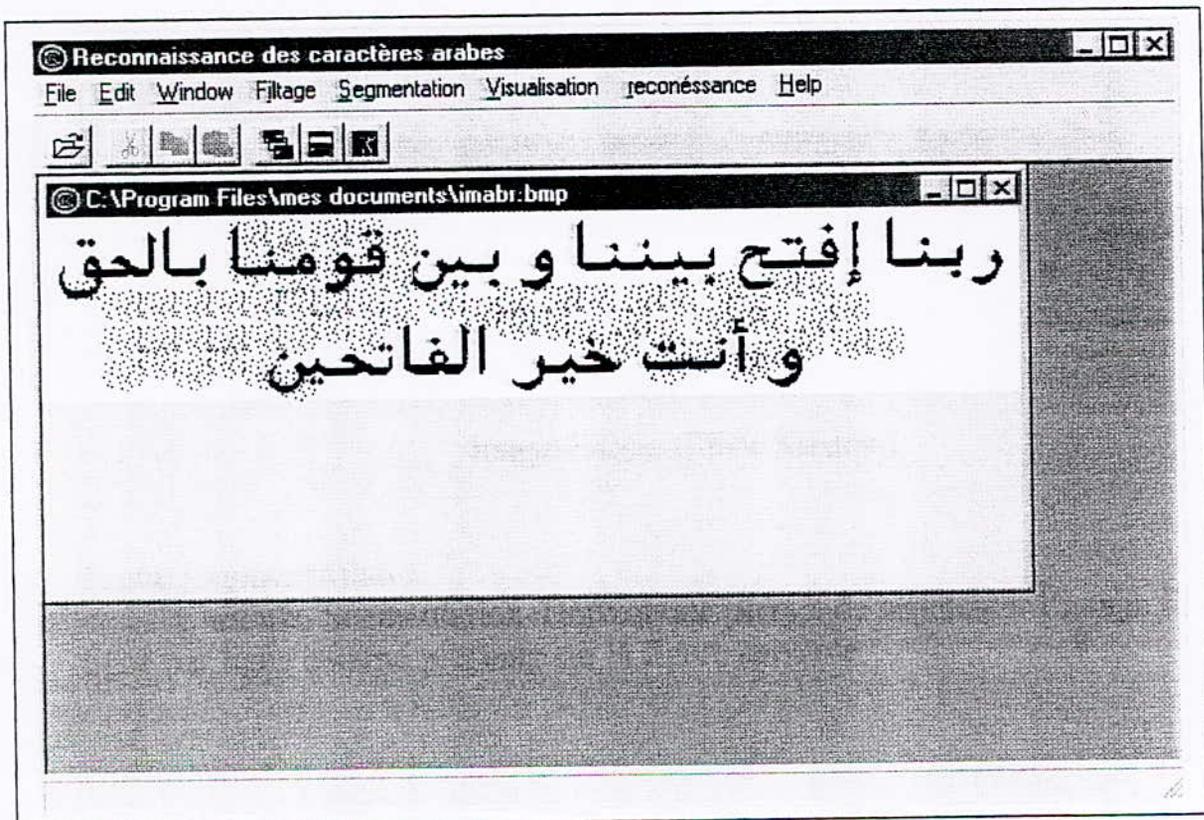
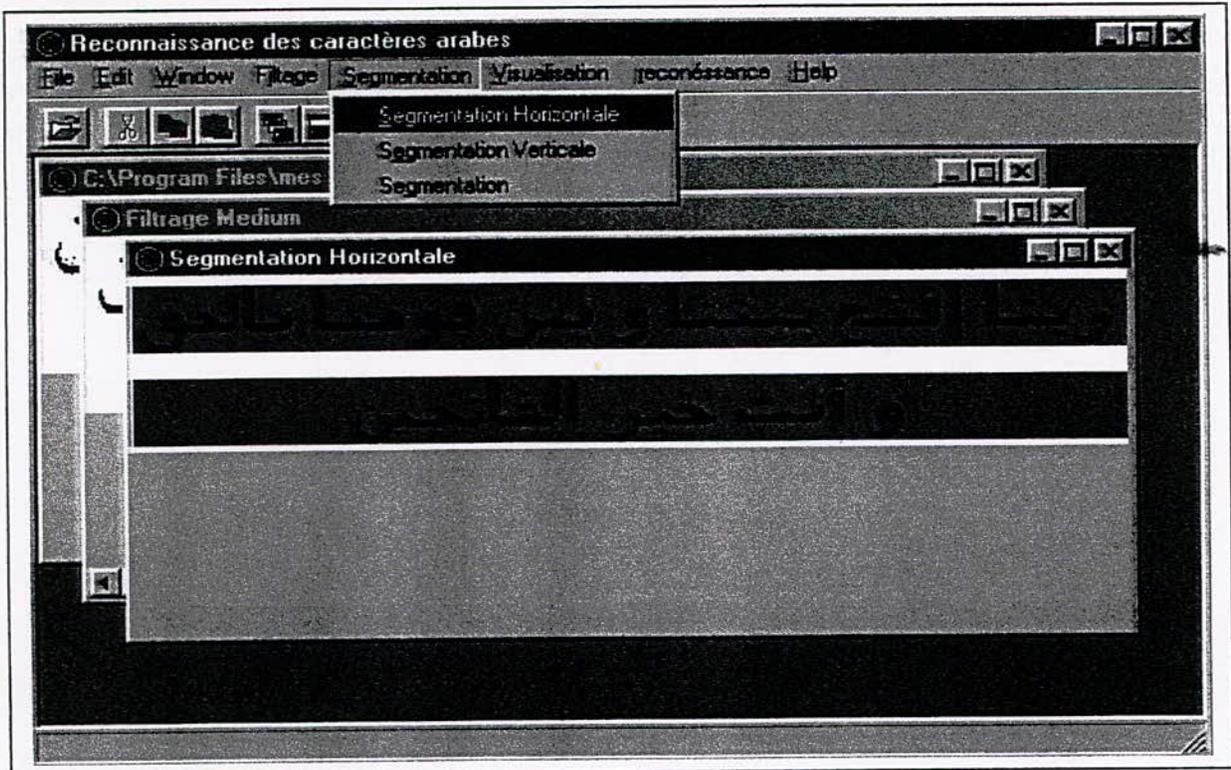
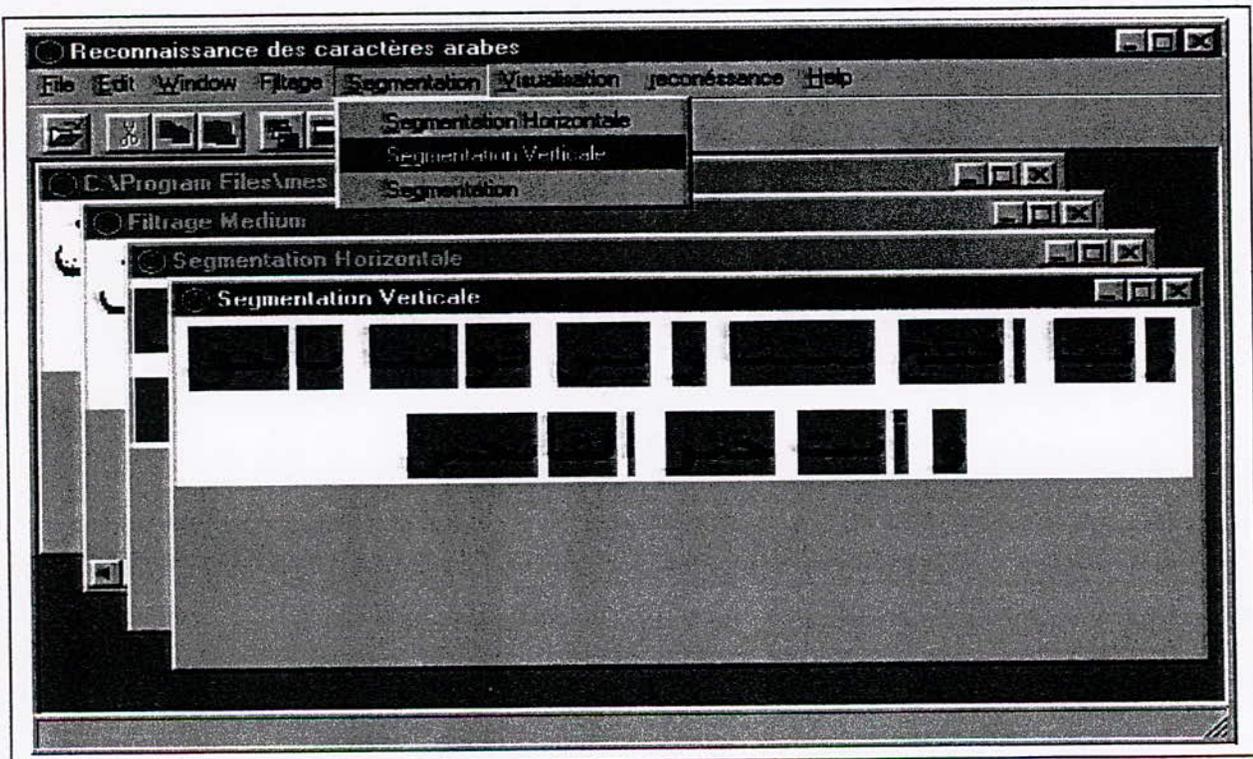


Image Originale.



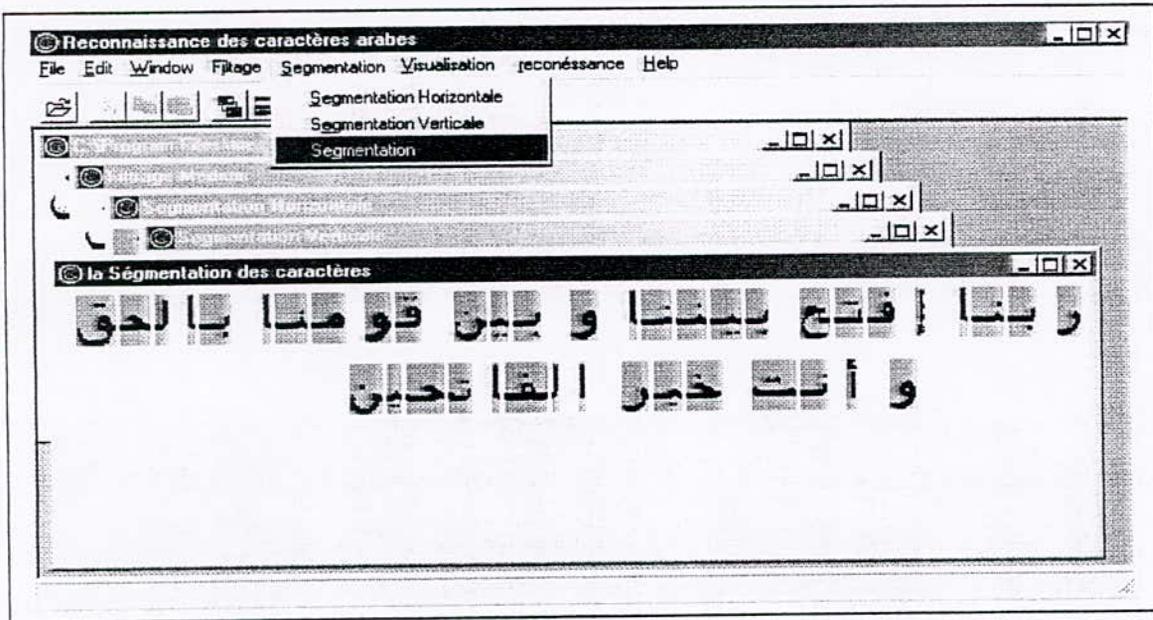
Résultat de la Segmentation Horizontale

L'article Segmentation Verticale permet de segmenter les lignes obtenues précédemment en mots ou parties connexes.



Segmentation des lignes en parties connexes

L'article Segmentation permet de segmenter l'image obtenu précédemment en caractères.

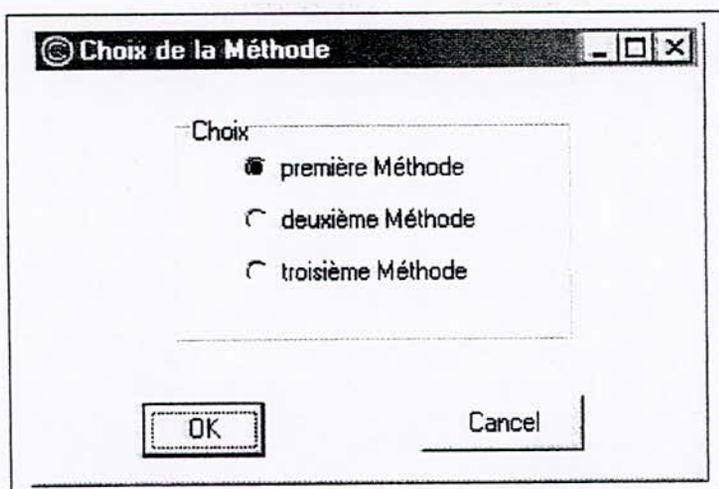


Segmentation en caractères

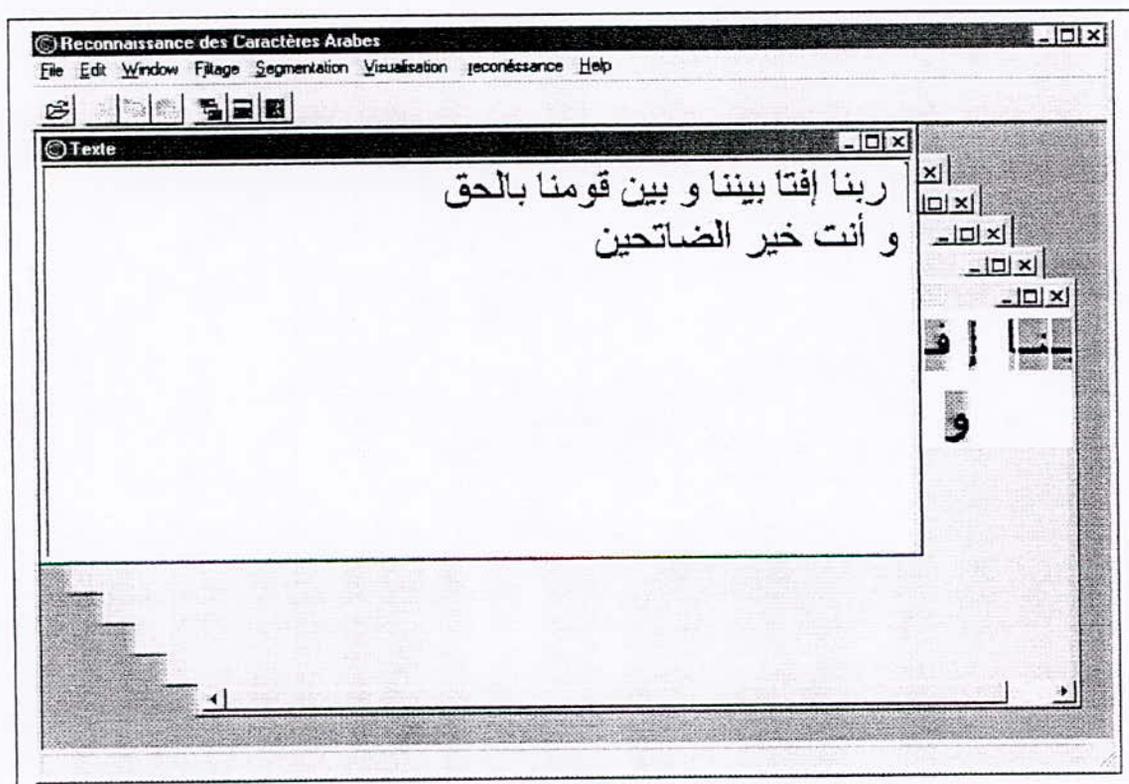
Menu Reconnaissance :

L'article Reconnaissance permet de reconnaître les caractères après le choix de la méthode, il présentera en résultat le texte reconnu.





Fenêtre permettant le choix de la Méthode



Résultat de la reconnaissance avec la première méthode

CONCLUSION GÉNÉRALE

CONCLUSION GÉNÉRALE

CONCLUSION GÉNÉRALE

Notre travail concerne essentiellement le prétraitement, la segmentation et la reconnaissance de l'écriture arabe imprimée.

Nous avons commencé par une présentation d'un système de reconnaissance des formes, puis nous avons décrit l'opération de prétraitement puis de segmentation et enfin de reconnaissance.

Les différents traitements décrits dans cette thèse ont été implémentés en utilisant le langage de programmation Borland C++Builder5.

Le document en arabe est scanné, un filtre médian est effectué sur la copie numérisée du document afin d'éliminer les bruits.

La segmentation utilisée s'effectue en trois étapes, la première est une segmentation horizontale qui segmente le texte en lignes, la deuxième est une segmentation verticale qui segmente les lignes en mots et la dernière segmente les mots en caractères.

Pour la reconnaissance, on a implémenté trois méthodes, la première est une méthode de comparaison de chaînes basée sur la notion de distance, les deux autres méthodes utilisent les réseaux de neurones. L'apprentissage de ces réseaux se fait par le logiciel MATLAB.

Une perspective envisageable de ce travail serait d'améliorer notre application de manière à pouvoir traiter les documents manuscrits, pour cela il faudra changer le module de segmentation qui utilise une méthode algorithmique, par un autre module basé sur des méthodes adaptatives comme les réseaux de neurones.

ANNEXE

Annexe

La règle de modification des poids (la méthode de rétropropagation du gradient de l'erreur)[4]

Nous allons dans ce qui suit détailler les calculs qui ont permis d'obtenir $\delta_k^{(j)}$, c'est à dire, la règle(7 du chapitre III)de modification des poids selon la méthode de rétropropagation du gradient de l'erreur. En reprenant l'équation (2 du chapitre III), l'erreur quadratique sur la sortie s'écrit :

$$E = \frac{1}{2} \sum_{i=1}^n (y_i - yd_i)^2$$

calculons le gradient de cette erreur par rapport au poids $w_{kh}^{(2)}$ associé à la connexion venant du neurone h de la couche 1 vers le neurone k de la couche 2 (l'indice de la couche cible est indiqué en haut) :

$$\frac{\partial E}{\partial w_{kh}^{(2)}} = \sum_{i \in \text{couche.3}} \frac{\partial E}{\partial p_i^{(3)}} \frac{\partial p_i^{(3)}}{w_{kh}^{(2)}} \dots\dots\dots(1)$$

Calculons d'abord le premier terme :

$$\frac{\partial E}{\partial p_i^{(3)}} = \frac{\partial E}{\partial y_i^{(3)}} \frac{\partial y_i^{(3)}}{\partial p_i^{(3)}}$$

avec :

$$\frac{\partial E}{\partial y_i^{(3)}} = (y_i^{(3)} - yd_i) \dots\dots\dots(2)$$

De la relation $y_i^{(3)} = f\left(\sum_{j \in \text{couche.2}} w_{ij}^{(3)} \cdot y_j^{(2)}\right)$, on obtient :

$$\frac{\partial y_i^{(3)}}{\partial p_i^{(3)}} = f'(p_i^{(3)}) \dots\dots\dots(3)$$

on tire, en tenant compte de l'expression de E :

$$\frac{\partial E}{\partial p_i^{(3)}} = (y_i^{(3)} - yd_i) f'(p_i^{(3)}) \dots\dots\dots(4)$$

Calculons maintenant le second terme de (1). Le potentiel du neurone i de la couche 3 est la somme des sorties $y_j^{(2)}$ de la couche 2 pondérées par les poids $w_{ij}^{(3)}$ des neurones de la couche3. On peut donc écrire :

Annexe

$$\frac{\partial p_i^{(3)}}{\partial w_{kh}^{(2)}} = \frac{\partial}{\partial w_{kh}^{(2)}} \left[\sum_{j \in \text{couche.2}} w_{ij}^{(3)} \cdot y_j^{(2)} \right] \dots\dots\dots (5)$$

Dans le second membre de cette expression, le seul terme de type $y_j^{(2)}$ dépendant du poids $w_{kh}^{(2)}$ est $y_k^{(2)}$, celui obtenu pour $j=k$. la relation (5) devient :

$$\frac{\partial p_i^{(3)}}{\partial w_{kh}^{(2)}} = w_{ik}^{(3)} \frac{\partial}{\partial w_{kh}^{(2)}} [y_k^{(2)}] \dots\dots\dots (6)$$

Dans laquelle on a fait sortir les poids $w_{ik}^{(3)}$.

En remplaçant $y_k^{(2)}$ (sortie du neurone k de la couche 2) par son expression, on arrive à :

$$\frac{\partial p_i^{(3)}}{\partial w_{kh}^{(2)}} = w_{ik}^{(3)} \cdot \frac{\partial}{\partial w_{kh}^{(2)}} \left[f \left(\sum_{m \in \text{couche1}} w_{km}^{(2)} \cdot y_m^{(1)} \right) \right] \dots\dots\dots (7)$$

Le seul terme de la somme sur m dont la dérivée est non nulle est celui qui contient $w_{kh}^{(2)}$, car il correspond à $m=h$. En calculant la dérivée, on obtient :

$$\frac{\partial p_i^{(3)}}{\partial w_{kh}^{(2)}} = w_{ik}^{(3)} \cdot f'(p_k^{(2)}) \cdot y_h^{(1)} \dots\dots\dots (8)$$

on peut réécrire à partir des relations (4) et (8) le gradient de l'erreur

$$\frac{\partial E}{\partial w_{kh}^{(2)}} = \sum_{i \in \text{couche3}} (y_i^{(3)} - yd_i) \cdot f'(p_i^{(3)}) \cdot w_{ik}^{(3)} \cdot f'(p_k^{(2)}) \cdot y_h^{(1)} \dots\dots\dots (9)$$

En posant $\delta_i^{(3)} = (y_i^{(3)} - yd_i) \cdot f'(p_i^{(3)})$ et en sortant les termes $y_h^{(1)}$ et $f'(p_k^{(2)})$ de la somme sur i :

$$\frac{\partial E}{\partial w_{kh}^{(2)}} = \left[\sum_{i \in \text{couche.3}} \delta_i^{(3)} \cdot w_{ik}^{(3)} \right] \cdot f'(p_k^{(2)}) \cdot y_h^{(1)} \dots\dots\dots (10)$$

Si on pose :

$$\delta_k^{(2)} = \left[\sum_{i \in \text{couche.3}} \delta_i^{(3)} \cdot w_{ik}^{(3)} \right] \cdot f'(p_k^{(2)}) \dots\dots\dots (11)$$

La relation (11) est l'expression de l'erreur affectée à un neurone k de la couche 2. Elle se généralise pour n'importe qu'elle couche (j) du réseau de neurones.

$$\delta_k^{(j)} = \left[\sum_{i \in \text{couche.3}} \delta_i^{(j+1)} \cdot w_{ik}^{(j+1)} \right] \cdot f'(p_k^{(j)}) \dots\dots\dots (12)$$

Bibliographie

- [1] Alexandre Aupetit, "Les réseaux de neurones artificiels", 1997.
- [2] J-P.Cocquerez, "Analyse d'image : filtrage et segmentation", 1995.
- [3] E.Davalo, P.Naïm, "Des réseaux de neurones", 1993.
- [4] J.Hérault, C.Jutten, "Réseaux neuronaux et traitement du signal", 1994.
- [5] M.Kunt , " Traitement d'images", 1993.
- [6] C.Remy, " L'intelligence artificielle", Dunod 1994 .
- [7] C.Touzet, "Les Réseaux de neurones, introduction au connexionisme cours, exercice, TP", 1992
- [8] Y.Yeddou, "Etude de synthèse sur les réseaux de neurones et leurs application", Magistère en automatique à l'école national polytechnique, 1998
- [9] Salim MELAZ et Mohamed SOUNA.
Conception et réalisation d'un système de reconnaissance de visage par la transformée de cosinus discrète.
Thèse d'ingénieur d'état, Institut d'informatique.
USTHB, 1996.
- [10] Jean-Paul Haton Reconnaissance des formes, Méthodes et applications
- [11] Contribution à la réalisation d'un système de reconnaissance des caractère arabes imprimés multipolices.
Thèse de magistère INI octobre 94 : Amer Benhouhou.
- [12] Reconnaissance des formes, Etapes et décision en utilisent une étape structurelle. PFE 95 : Zerouati Faycal & Zerkini Benyhia.
- [13] J-J Toumazet Traitement de l'image sur un micro-ordinateur : France 87.
- [14] A-Blaid & K-Tonibre Analyse de document, de l'image à la sémantique.
: colloque national sur l'écrit et le document. Nancy 92.
- [15] J-J TOUMAZET Traitement de l'image par l'exemple : France 90.
- [16] Reconnaissance d'un système OCR d'un texte arabe multitable multifonte
PFE 99: BAALI Abdenour & BOUNOUH Djamel.