République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université HADJLAKHDAR -BATNA-
Faculté des Sciences de l'Ingénieur
Département Génie Industriel

Mémoire présenté
Pour l'obtention du diplôme de
Magistère en Génie Industriel
Option : Génie Industriel et Productique

Par
DERDOUR Khedidja
Ingénieur d'état en informatique

Intitulé

Reconnaissance de formes du chiffre arabe imprimé : Application au code à barre d'un produit

Dr. N. Mouss	MC A	Université de Batna	Présidente
Dr. H. Mouss	MC A	Université de Batna	Promoteur
R. Bensaadi	MA B	Université de Batna	Coencadreur
Dr. O. Kazar	MC A	Université de Biskra	Examinateur
Dr. N. Benoudjit	MC A	Université de Batna	Examinateur
S. Abdelhamid	MA A	Université de Batna	Invité

2008/2009

Remerciements

Je Remercie le laboratoire pour avoir mis à notre disposition le matériel nécessaire.

J'adresse en premier lieu mes sincères remerciements et à exprimer toute ma gratitude à mes encadreurs Mme H. Mouss et R. Bensaadi pour leur disponibilité leur idées et leur conseils

Je remercie la présidente pour avoir accepter de juger ce travail, et les membres du jury qui ont pris de leurs temps pour lire, juger ce travail

Je remercie Mr. J. P Rennard pour l'accueil de mes emails et ses conseils, et à Mr N. Benoudjit pour ses orientations.

Enfin, remercier est une tâche ingrate puisqu'elle conduit inévitablement à des oublis. Je vais donc remercier tous ceux qui de près ou de loin ont contribué par un mot, un geste, une attention à ce travail et qui ne sont pas cité.

TABLE DES MATIERES

LISTE DES FIGURES

CHAPITRE 1 : RECONNAISSANCE DE FORMES

CHAPITRE 2 : LES PARAMETRES CARACTERISTIQUE

CHAPITRE 3 : LA CLASSIFICATION

CHAPITRE 4 : CODE A BARRES

CHAPITRE 5 : L'APPLICATION

LISTE DES TABLES

CHAPITRE 2 : LES PARAMETRES CARACTERISTIQUE

CHAPITRE 4 : CODE A BARRES

CHAPITRE 5 : L'APPLICATION

INTRODUCTION GENERALE

La vision par ordinateur est un thème à la frontière entre la robotique, l'informatique et les neurosciences. Dans le cadre de l'informatique, ce sont essentiellement les aspects « interprétation du signal d'image » où la reconnaissance d'objets par leurs images qui attire jusqu'à ce jour l'intérêt de la recherche et s'implique dans des applications diverses : sécurité, diagnostic, météo, aide à la navigation robotique…etc. Dans les chaines de fabrication, le robot apprend à trier, souder, visser assembler, en bref à remplacer l'homme dans la réalisation de ses opérations avec une précision nettement meilleure.

Le domaine de Reconnaissance de Formes (RdF) est devenu aujourd'hui l'un des grands domaines dans lesquels travaillent de plus en plus de chercheurs. Le but est de trouver des algorithmes pouvant résoudre sur ordinateur les problèmes de RdF qui sont intuitivement résolus par l'homme. Les ordinateurs séquentiels sont très performants pour les calculs, ils peuvent exécuter des opérations complexes plus vite que le cerveau humain. En RdF, le cerveau est capable d'analyser une scène très complexe instantanément, alors que le meilleur des algorithmes de RdF ne pourrait reconnaître que des formes relativement simples.

Sans nul doute, l'homme est le plus parfait des systèmes RdF qui existent. La diversité des tâches de reconnaissance que nous pouvons accomplir sur des formes à grande variabilité est restera toujours impressionnante. Nous distinguons sans effort, par exemple le visage d'une femme de celui d'un homme, un linge sale d'un linge propre, etc.

Les progrès scientifiques et techniques nous permettent aujourd'hui d'essayer d'imiter certaines de ces facultés à l'aide des machines. Sans aller jusqu'à l'idée absurde de remplacer complètement l'homme par une machine, il y a malgré tout, beaucoup de cas particuliers où les travaux de reconnaissance simples effectués par l'homme peuvent être confiés à une machine. Les motivations de cette substitution se situent sur trois plans principaux qui sont ceux de l'efficacité, du social et de l'économie [1].

A titre d'exemple, un être humain perdra beaucoup de temps à séparer dans un ensemble de plusieurs milliers de chiffres provenant de mesures, ceux qui correspondant à une anomalie de ceux qui sont corrects. Un autre exemple de cette inefficacité s'observera dans la recherche d'un mot donné dans un long texte. Une machine peut effectuer ces travaux beaucoup plus rapidement que l'homme.

Sur le plan social, l'homme est souvent confronté à des tâches monotones, ennuyeuses, répétitives tel que l'ouvrier qui, à longueur de journée, doit placer les mêmes vis dans les mêmes trous, ou dangereuses tel que la réparation d'une centrale nucléaire. Une machine ou un robot peuvent effectuer des tâches similaires.

Enfin sur le plan économique, il peut être moins onéreux, par exemple, d'automatiser le contrôle de qualité sur une chaîne de production ou le tri sur une chaîne de montage.

Le but de la reconnaissance de l'écriture est de transformer un texte écrit en une représentation compréhensible par une machine et facilement reproductible par un logiciel de traitement de texte. Cette tâche n'est pas triviale car les mots possèdent une infinité de représentations. Chaque personne produit une écriture qui lui est propre, de plus il existe de nombreuses polices de caractères pour l'imprimer avec de nombreux styles (gras, italique, souligné, ombré etc.) et des mises en page différentes et complexes. Suivant le type d'écriture qu'un système doit reconnaître (manuscrit, cursif ou imprimé), les opérations à effectuer et les résultats peuvent varier notablement.

Ce travail s'intéresse principalement à la reconnaissance de séquences de chiffres imprimés.

La reconnaissance d'écritures alphabétiques imprimées est entrée depuis quelques années dans une phase d'applications industrielles et il existe des logiciels de reconnaissance de caractères (que l'on trouve souvent sous le nom de 'Optical Character Recognition'.-OCR-).

Il n'est cependant pas rare de trouver des réseaux affichant après apprentissage des taux de reconnaissance de plus de 99%, bien qu'ils ne soient pas forcément adaptés à une utilisation industrielle (qui impose des contraintes de temps, de traitement de données et d'équipements informatiques).

La reconnaissance de chiffres arabes est considérée comme la plus simple à réaliser. Le vocabulaire est très restreint (10 chiffres), les taux avoisinent actuellement les 99.98% sur des documents imprimés de bonne qualité, par les logiciels OCR.

Dans le cadre de ce travail nous proposons un système de reconnaissance du chiffre arabe imprimé ; dédié à la lecture automatique des codes à barres type EAN (European Article Numbering). Il s'agit donc de développer une méthode capable d'attribuer automatiquement à un chiffre une étiquette qui indique sa nature (zéro, un, deux, etc.). Ce travail entre dans le cadre d'un projet de recherche CNEPRU « Exploration du potentiel des techniques de l'intelligence artificielle, la vision par

ordinateur dans un contexte de coordination multi-robot- », identifié sous le code J0201320090043, qui fait appel aux techniques de la vision par ordinateur appliquée en productique, en particulier la reconnaissance d'objets en mouvement en vue de subir des traitements industriels particuliers.

Pour réaliser ce travail on a divisé ce mémoire à 5 chapitres :

Le premier chapitre de ce travail contient une description générale du problème de reconnaissance de formes, avec une présentation de différentes étapes d'un système de reconnaissance de l'écriture.

Le deuxième chapitre est consacré à l'extraction des paramètres caractéristiques à savoir les cavités, le zonage, le profil, les histogrammes, et les moments invariants de Hu et de Zernike

Le troisième chapitre présente les différentes méthodes de classification utilisée, la méthode de classification KPPV (K plus proche voisins), KPPV floue et les réseaux de neurones

Le quatrième chapitre étudie en détail le code à barre de type EAN le plus utilisé sur le marché.

Le dernier chapitre présente notre application de reconnaissance de chiffres arabes d'un code à barre. Nous suivons les étapes d'un système de RdF, de l'acquisition des images à l'extraction des attributs, nous décrivons ensuite les classifieurs utilisés.

Afin de construire cette application, nous avons utilisé le langage de programmation Matlab qui peut être vu comme un langage de programmation de haut niveau, permettant de programmer de manière très rapide différents calculs scientifiques, et notamment ceux qui font appel à la manipulation de matrices.

Enfin, une conclusion générale, nous résumons la démarche de notre travail. Et nous proposons quelques perspectives.

CHAPITRE 1

RECONNAISSANCE DE FORMES

CHAPITRE 1

RECONNAISSANCE DE FORMES

La reconnaissance de formes (RdF) -en anglais pattern recognition- ou reconnaissance de motifs est un sous-domaine de l'apprentissage automatique. Elle consiste en une automatisation de tâches de perception artificielle réalisées usuellement par le cerveau et le système sensoriel humain, exemple : reconnaître un caractère, un son, un signal, un objet dans une image numérique. Une forme est une représentation simplifiée du monde extérieur définie sous une forme acceptable par l'ordinateur (par exemple un vecteur de réels, ou bien un mot d'un langage donné, ...)

Dans ce chapitre, nous présentons le système de RdF qui a pour objectif de prédire la classe d'appartenance d'une forme inconnue. Pour y parvenir, trois étapes sont nécessaires. La première consiste à faire des prétraitements et à isoler la forme à reconnaître, la deuxième c'est de Calculer les paramètres caractéristiques sur la forme à reconnaître, la troisième est la classification où la décision est prête en utilisant une base d'exemples prédéfini –apprentissage-.

1.1 Introduction

La reconnaissance de forme (RdF) est historiquement un chapitre de l'intelligence artificielle. Elle consiste à attribuer automatiquement une étiquette à une forme présentée dans une image.

En effet, cette discipline se trouve au croisement de beaucoup d'autres dont les principales sont : les Statistiques, la Linguistique, la Recherche Opérationnelle, la Théorie des Communications, l'Informatique, la Biologie, l'Optique, l'Electricité. Nous pouvons citer certains de ces domaines d'application :

Reconnaissance des formes sur signaux temporels

- Signal de parole : reconnaissance de la parole (qu'a t on dit ?), reconnaissance du locuteur (qui a parlé ?)
- Signaux biomédicaux : électrocardiogramme, ...
- Surveillance d'instruments, diagnostic de panne.

Reconnaissance des formes dans les images numériques

- Lecture automatique de caractères
- Reconnaissance d'empreintes digitales
- Radiographies
- Analyse de scènes, interprétation d'images, « computer vision »

Pour la reconnaissance automatique de caractères, l'objet est de convertir des images qui sont compréhensibles par l'homme, en un code interprétable par un ordinateur. On considère donc souvent la reconnaissance des formes comme un problème de classification, c'est-à-dire l'affectation de chaque donnée prévisible à la catégorie pertinente.

Cependant dans ce chapitre, nous nous intéressons à la reconnaissance de l'écriture, où nous la présentons avec les différentes approches, et les différentes étapes à suivre.

1.2 Reconnaissance de l'écriture

1.2.1 Définition

La reconnaissance de l'écriture est mieux connue sous le nom d'OCR (Optical Character Recognition), du fait de l'emploi de procédés d'acquisition optiques [5].

L'OCR connaît plusieurs applications pratiques dans différents domaines d'activité parmi lesquels on peut citer :

- *Les banques et les assura*nces pour l'authentification de chèques bancaires (correspondance entre montants et libellé d'une part, et correspondance entre l'identité du signataire et sa signature, d'autre part), et la vérification de clauses de contrats pour les assurances.
- *La poste pour la lecture* des adresses et le tri automatique du courrier.
- *Les télécommunications* pour l'échange de fichiers informatiques à distance.
- *La police et la sécurité* pour la reconnaissance de numéros minéralogiques pour le contrôle routier, l'authentification et l'identification de manuscrits et l'identification du scripteur.
- *Les affaires et l'industrie* pour la gestion des stocks et la reconnaissance de documents techniques.
- *La bureautique* pour l'indexation et l'archivage automatique de documents, et pour la publication électronique.
- *L'administration* pour la reconnaissance de plans cartographiques et la lecture automatique de documents administratifs.

1.2.2 Reconnaissance de l'imprimé ou du manuscrit

L'approche n'est pas la même selon qu'il s'agisse de reconnaître un imprimé ou un manuscrit. Dans le cas de l'imprimé, les caractères sont bien alignés et souvent bien séparés verticalement, ce qui simplifie la phase de lecture, bien que certaines fontes présentent parfois des accolements qu'il faut défaire. De plus, le graphisme des caractères est conforme à un style calligraphique (fonte) qui constitue un modèle pour l'identification. Dans le cas du manuscrit, les caractères sont souvent ligaturés et leur graphisme est inégalement proportionné. Cela nécessite l'emploi de techniques de délimitation très spécifiques et souvent des connaissances contextuelles pour guider la lecture.

1.2.3 Reconnaissance monofonte, multifonte ou omnifonte

Les notions que nous présentons ici concernent uniquement les textes imprimés. Un système est dit *monofonte* s'il ne traite qu'une fonte à la fois, c'est-à-dire que le système ne connaît l'alphabet que dans une seule fonte. L'apprentissage y est simple puisque l'alphabet représenté est réduit. Un système est dit *multifonte* s'il est capable

de reconnaître un mélange de quelques fontes parmi un ensemble de fontes préalablement apprises. Dans ce cas, le prétraitement doit réduire les écarts entre les caractères (taille, épaisseur et inclinaison), l'apprentissage doit gérer les ambiguïtés dues aux éventuelles ressemblances de caractères des différentes fontes, et la reconnaissance doit identifier les subtiles différences entre ces caractères. Enfin, un système est dit omnifonte s'il est capable de reconnaître toute fonte sans l'avoir absolument apprise, ce qui relève actuellement du domaine de la recherche.

1.2.4 Système avec ou sans apprentissage

Un système avec apprentissage comporte un module d'introduction de modèles de caractères de référence. Des échantillons significatifs de l'écriture (imprimés ou manuscrits) en nombre suffisant (des dizaines, voire des centaines d'échantillons par caractère) sont saisis en mode manuel ou automatique. Dans ce premier cas, l'utilisateur indique au système l'identité de chaque échantillon permettant au système d'organiser ses classes en fonction du vocabulaire étudié. Pour le manuscrit, l'idéal serait d'apprendre les caractères directement à partir d'un texte, mais cela pose des problèmes évidents de segmentation. Dans le second cas, les échantillons sont regroupés automatiquement à partir d'analyse morphologique, sans connaissance de leur classe.

Dans un système sans apprentissage, une base de connaissance est intégrée au système et des algorithmes d'analyse particuliers sont utilisés.

1.3 Les Approches de la Reconnaissance de Formes

Il n'y a pas une théorie unifiée de RdF cependant, il existe deux approches principales: l'approche basée sur la théorie statistique de la décision et l'approche structurelle.

1.3.1 Approche statistique

Une approche classique en RdF est fondée sur l'étude statistique des mesures que l'on a effectuées sur les objets à reconnaitre. L'étude de leur répartition dans un espace métrique et la caractérisation statistique des classes permet de prendre une décision de reconnaissance du type « plus forte probabilité d'appartenance à une classe ». Ces méthodes s'appuient en général sur des hypothèses concernant la description statistique des familles d'objets analogues dans l'espace de représentation [6].

Dans cette approche, on supposera donc que les mesures faites sur une forme peuvent s'exprimer sous la forme d'un vecteur $X=(x_1,\ldots\ldots\ldots\ldots,x_n)$ de l'espace R^n. On dispose d'un ensemble d'apprentissage, c'est-à-dire d'un jeu de tels vecteurs dont on connaît en plus la classe d'appartenance. Le problème peut ainsi se résumer sommairement : Etant donné un vecteur inconnu, obtenu par mesures sur une forme, trouver la classe à laquelle on doit l'affecter ? Autrement dit, reconnaître la forme inconnue en fonction de l'apprentissage effectué [6].

L'approche statistique ou vectorielle consiste à représenter les formes par des vecteurs de caractéristiques puis à construire des classifieurs (paramétrique, non paramétrique, ou neuronale) opérant sur ces vecteurs [4].

1.3.2 Approche structurelle

Si l'approche statistique permet de se placer dans un cadre mathématique solide et général, elle présente néanmoins le défaut d'oublier la nature des mesures qui sont faites sur les formes et de les traiter de façon abstraite. On conçoit cependant qu'il est plus simple et plus riche d'utiliser des paramètres descriptifs liés à la nature même des formes étudiées. Si l'on possède une technique de suivi de contour (ce qu'il est facile d'imaginer dans les images à deux niveaux de gris) et un détecteur de segment, on voit que tous les triangles par exemple peuvent se décrire par une caractéristique du type : « la frontière est composée d'un segment horizontal de longueur l, suivi d'un segment oblique de longueur à peu près l, suivi d'un segment oblique de longueur à peu près l, qui se termine au point de départ de premier» [6].

L'intérêt d'une telle description est d'être universelle pour tous les triangles équilatéraux, indépendamment de leur taille et de leur position dans l'image. Elle décrit ces figures, d'une part en définissant des formes élémentaires (segments de droites dans la frontière) et, d'autre part, par l'assemblage de ces formes élémentaires pour constituer la figure totale. On dit qu'une telle description est de type structurel, puisqu'elle s'attache plutôt à définir les caractéristiques intrinsèques de la forme qu'à donner sa description métrique (qui ne se révèle pas riche dans un tel exemple) [6].

Dans le cas d'une description structurelle, il faudra utiliser des formalismes plus complexes. Les formes sont décomposées en primitives simples (c'est-à-dire en composantes élémentaires –non décomposables- d'une forme. Elles sont mises en évidence à la phase d'extraction. A titre d'exemple : un segment de droite, une boucle ... dans une figure géométrique) qui peuvent être des graphèmes ou même les

pixels de l'image. Elles sont ensuite représentées par un objet complexe, composé de primitives, comme une chaîne ou un graphe. Le processus de reconnaissance grammatical, stochastique ou graphique est propre à la représentation utilisée.

La représentation structurée peut sembler plus naturelle en RdF. le traitement structuré des formes peut se répartir en deux grandes familles [4] :

- Syntaxique : car elle tente de s'appuyer sur les travaux de la linguistique formelle. Les formes sont alors codées par des « mots » utilisant un alphabet dont chaque terme représente un élément de la forme à décrire. Définir une classe de formes revient à établir des règles syntaxiques caractérisant les mots acceptables. Le choix de l'alphabet et des règles syntaxiques est souvent fort délicat, mais des méthodes pour déterminer ces règles ont été proposées.
- Structurelle : tout en organisant la représentation des formes à l'aide de primitives de description et de relations entre ces primitives, elle ne s'appuie pas sur les grammaires formelles mais sur des techniques combinatoires (isomorphismes de graphes, arbres d'interprétation, ...) ou numériques (métrique sur les mots, relaxation,…).

1.4 Schéma général de la reconnaissance de formes

La reconnaissance automatique de caractères, au sens large du terme, est une discipline vieille. Elle pose la question de l'intelligence humaine et des possibilités de l'intelligence artificielle la LAD (Lecture Automatique de Document) et plus généralement la RdF sont des domaines de recherche actifs depuis la fin des années soixante [10].

La RdF consiste à attribuer automatiquement une étiquette à une forme présente dans une image. D'une autre manière, la reconnaissance automatique de caractères a pour objet de convertir des images qui sont compréhensibles par l'homme, en un code interprétable par un ordinateur.

Une illustration du processus de RdF est présentée sur la figure 1.1.

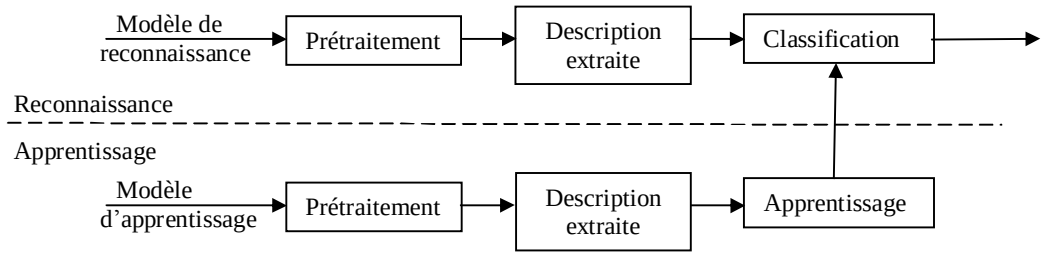


Figure 1.1. Schéma global d'un processus de reconnaissance de caractère

1.4.1 Prétraitement

Un caractère est représenté dans l'ordinateur par une matrice de grandes dimensions. Afin de réduire la dimension de représentation, et, ce faisant, de rendre plus facile la conception du système de classification, une étape de prétraitement est nécessaire. Son objectif est de faciliter la caractérisation de la forme (caractère, chiffre, mot) ou de l'entité à reconnaître soit en nettoyant la forme (élimination du bruit) ou en réduisant la quantité d'information à traiter pour ne garder que les informations les plus significatives.

Le prétraitement est basé sur le traitement d'images qui désigne un ensemble de méthodes dont l'objectif est de transformer des images (pour en améliorer l'apparence ou pour les coder de manière plus compacte en vue d'une transmission), soit d'en extraire de l'information (par exemple lorsqu'il s'agit de reconnaître automatiquement l'écriture). Il s'agit donc d'un domaine très vaste, qui trouve de plus en plus d'applications.

a) Techniques de prétraitement

Les techniques de prétraitement sont différentes suivant que les caractères sont donnés sous forme d'image ou de suite de segments. Dans notre application nous restons dans la première catégorie. A cause de la forte granularité de l'échantillonnage et des divers problèmes d'éclairage et de saisie, l'image du caractère peut subir des manquements ou des empâtements. Il convient de corriger si possible ces problèmes avant toute étape d'analyse. Les techniques de prétraitement sont les suivantes :

- ***Lissage :*** L'image du caractère peut être entachée de bruit dû aux artefacts de l'acquisition et souvent à la qualité du document, conduisant soit à des

absences de points (trous) soit à des empâtements ou des excroissances et donc à une surcharge de points. Les techniques de lissage permettent de résoudre ces problèmes.

- ***Normalisation de la taille :*** La taille d'un caractère peut varier d'une écriture à l'autre, ce qui peut causer une instabilité des paramètres. Une technique naturelle de prétraitement consiste à ramener les caractères à la même taille. L'algorithme de normalisation que nous avons utilisé est très simple et il est présenté dans le chapitre 3 de ce mémoire.
- ***Amincissement :*** Le but de l'amincissement d'un caractère est de simplifier l'image du caractère en une image plus facile à traiter en la réduisant par exemple à une dimension, c'est-à-dire que l'épaisseur du caractère est réduite à un pixel.
- ***Redressement :*** Il est souhaitable que l'image soit parfaitement droite. Il faut donc détecter l'angle de déviation globale (en anglais skew), puis effectuer une rotation inverse de la valeur trouvée. Cette technique permet de détecter l'italique [7] .

b) Formats d'image en mémoire

Une image peut être représentée de différentes manières au niveau informatique. Le logiciel Matlab supporte quatre type d'images : images binaires, images d'intensités, images couleur RGB (en anglais : Red, Green, Blue), images couleurs indexées.

- ***Images binaires :*** Une image binaire est une matrice rectangulaire dont les éléments valent 0 (noir) ou 1 (blanc).
- ***Images d'Intensités :*** Une image d'intensité est une matrice dans laquelle chaque élément est un réel compris entre 0 (noir) et 1 (blanc). On parle aussi d'image en niveaux de gris,
- ***Images couleur RGB :*** Pour représenter la couleur d'un pixel il faut donner trois nombres, qui correspondent au dosage de 3 couleurs de base : Rouge, Vert et Bleu (en anglais, RGB : Red, Green, Blue). On peut ainsi représenter une image couleur par trois matrices dont chacune correspond à une couleur de base.
- ***Images couleur* indexées *:*** La représentation indexée ne permet de représenter

qu'un nombre limité de couleurs. Ces couleurs sont mémorisées dans une table de couleurs (colormap) qui est une matrice *n* x 3 (où *n* est le nombre de couleurs). L'image est alors une matrice contenant des nombres entiers compris entre 1 et *n*, chaque entier jouant le rôle d'un index relatif à la table de couleurs

c) La morphologie mathématique

Un opérateur de morphologie mathématique reçoit une image en entrée et fournit une image en sortie. On peut le définir par l'intermédiaire d'un élément structurant, qui est une forme. On peut le voir comme un masque. Il y a deux opérations morphologiques de base, l'érosion et la dilatation [2] :

L'érosion : Soit B un élément structurant de dimension impaire et B_x cet élément centré en un pixel x. L'érosion de X par B consiste à poser en chaque pixel x de l'image, la question : « B_x est-il contenu entièrement dans X? ». L'ensemble des positions x correspondant à une réponse positive forme le nouvel ensemble Y, appelé érodé de X par B. Cet ensemble satisfait l'équation :

$$Y = X \ominus B = \{x | B_x \subseteq X\} \qquad \textbf{(1-1)}$$

La dilatation : L'opération de dilatation se définit de manière analogue à l'érosion. En prenant le même élément structurant centré en x, B_x, la question posée pour chaque point x de l'image est : « B_x touche-t-il l'ensemble X? ». C'est à dire, existe-t'il une intersection non vide entre B_x et X? L'ensemble des points x de l'image correspondant aux réponses positives forme le nouvel ensemble X des dilatés de X.

$$Y = X \oplus B = \{x | B_x \cap X \neq \emptyset\} \qquad \textbf{(1-2)}$$

A partir des deux opérations de base, on peut définir d'autres plus complexes, telles que l'ouverture et la fermeture :

L'ouverture : L'ouverture est l'application de l'opérateur érosion puis de l'opérateur dilatation avec le même élément structurant. Y, l'ouverture de X par B, est notée :

$$Y = X \circ B = (X \ominus B) \oplus B \qquad \textbf{(1-3)}$$

En général, l'ensemble de départ n'est pas conservé car une partie de la forme éliminée par l'érosion ne peut être recréée par une dilatation. L'ensemble Y est plus régulier (moins de détails au niveau du contour) que l'ensemble initial X. En termes « géographiques » ou morphologiques, l'ouverture adoucit les contours, coupe les isthmes étroits, supprime les petites îles et les caps étroits.

La fermeture : La fermeture est l'opération « inverse » de l'ouverture. Elle consiste en une dilatation suivie d'une érosion (toujours en gardant le même élément structurant) :

$$Y = X \cdot B = (X \oplus B) \ominus B \qquad \textbf{(1-4)}$$

Un ensemble fermé est également moins riche en détails que l'ensemble initial. La transformation par fermeture bouche les canaux étroits, supprime les petits lacs et les golfes étroits.

Supposons, par exemple, que l'on effectue, ces quatre opérations, avec pour élément structurant un carré (matrice 3x3) (figure1.2.)[8] :

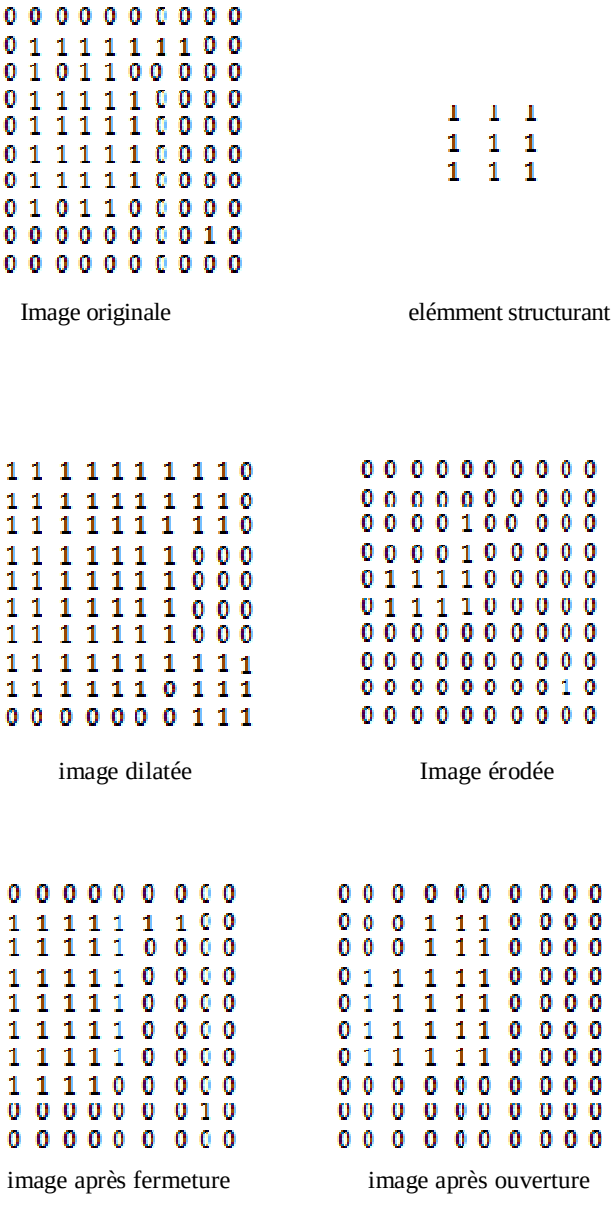


Figure 1.2. Principe des opérations morphologique

L'utilisation de ces opérations morphologique peuvent être utiliser pour l'extraction du contour. Le contour interne d'un objet s'obtient par une érosion d'un objet A (suivant l'élément B) suivie d'une différence : $A - (A \ominus B)$. Le contour externe s'obtient par une dilatation suivie d'une différence : $A - (A \oplus B)$

d) Les filtres

Le filtrage est une opération fondamentale en traitement d'images. Il permet d'améliorer la perception de certains détails, de réduire le bruit, de compenser certains défauts du capteur, etc.

Les contours constituent une information essentielle pour certaines applications de traitement d'images. Les contours correspondent en général à des changements brusques de propriétés physiques ou géométriques de la scène. En particulier, les contours d'un objet permettent en général de caractériser sa forme.

La détection de contours peut être réalisée grâce à des filtres dont les coefficients ont été soigneusement choisis. Ces filtres comportent deux masques de convolution, un pour la détection des contours verticaux et un autre pour la détection des contours horizontaux. On obtient après convolution une image des gradients. Cette image pourra par la suite être modifiée dans une phase de seuillage pour ne contenir que des pixels à 1 ou 0 correspondant respectivement à point de contour ou à un non point de contour.

Nous aborderons trois jeux de filtres particuliers, les plus utilisés : les filtres de Prewitt, Roberts, et Sobel.

- ***Filtres de Prewitt* :** Les filtres de Prewitt sont les suivants :

  Filtre horizontal : $h=\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$,

  Filtre vertical : $v=\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$

- ***Filtres de Sobel* :** Les filtres de Sobel sont les suivants :

  Filtre horizontal : $h=\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Filtre vertical : $v=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}$

- ***Filtres de Robert :*** Les filtres de Roberts sont les suivants :

  Filtre diagonal : $a=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$

  Filtre anti-diagonal : $b=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}$

e) Segmentation

Fondamentalement, la segmentation est un processus qui consiste à découper une image en régions connexes présentant une homogénéité selon un certain critère, comme par exemple la couleur. L'union de ces régions doit redonner l'image initiale.
La segmentation est une étape importante pour l'extraction des informations qualitatives de l'image.

On regroupe généralement les algorithmes de segmentation en trois grandes classes :

- ***Segmentation basée sur les pixels :*** Elle travaille sur des histogrammes de l'image. Par seuillage, clustering ou, l'algorithme construit des classes de couleurs qui sont ensuite projetées sur l'image.
- ***Segmentation basée sur les régions :*** Elle correspond aux algorithmes d'accroissement ou de découpage de région. Dans l'accroissement de région on part d'un ensemble de petites régions uniformes dans l'image (de la taille d'un ou de quelques pixels) et on regroupe les régions adjacentes de même couleur jusqu'à ce qu'aucun regroupement ne soit plus possible. Pour le découpage de région, on part de l'image entière que l'on va subdiviser récursivement en plus petites régions tant que ces régions ne seront pas suffisamment homogènes. Les algorithmes dit split and merge : Sont un mélange de ces deux approches.
- ***Segmentation basée sur les contours :*** Elle s'intéresse aux contours des objets dans l'image.

f) Localisation

Lorsque l'image n'est pas trop complexe, on peut réaliser une localisation en comptant les pixels blancs dans le sens des lignes, puis, pour chaque ligne de chiffres, en comptant les pixels blancs dans le sens des colonnes. La comparaison du résultat du

comptage avec an seuil permet de détecter les débuts et fins de chiffres.

1.4.2 Extraction des caractéristique et primitives

L'extraction de caractéristiques ou primitives, en anglais, « feature extraction » transforme le bitmap d'origine, caractère ou forme modèle, en une description numérique ou symbolique dans un espace abstrait, selon un formalisme prédéfini [10]. La description sera évidemment beaucoup plus simple pour une reconnaissance de bas niveau de type global que pour une reconnaissance symbolique (structurel). Dans le premier cas, par exemple, on pourra se contenter de détecter les intersections du caractère avec une grille fixe, dans le deuxième on pourra décrire avec un véritable langage, les segments, la concavité et les boucles du caractère analysé. Dans le premier cas l'image est transformée en un vecteur de valeurs numériques, dans le deuxième en une suite de symboles, obtenues par une analyse de caractère en formes élémentaires [7].

Les attributs à extraire doivent être discriminantes le plus possible et répondent aux exigences suivants [1]:

> Une faible variance intra classe, une grande variance inter classe, une faible nombre d'attributs et l'indépendances de la translation, de la rotation et le facteur d'échelle.

Il existe un très grand nombre de descriptions possibles pour une forme suivant l'information à modéliser et les invariances à exprimer. La plus courante, considère d'une part les descripteurs (attributs) basés sur la frontière (descripteurs externes) et d'autre part ceux basés sur la forme elle-même (descripteurs globaux ou internes).

a) Descripteurs globaux

Ils sont basés sur la forme elle-même, l'image I (matrice binaire) d'une forme peut être décrite par une application $f : (x,y) \rightarrow \{0,1\}$ où $(x,y) \in I$ sont les coordonnées des pixels de l'image. Si le pixel (x, y) est en dehors de la forme $f(x,y)=0$ et $f(x,y)=1$ s'il appartient à la forme. L'ensemble des points de la forme est noté $F=\{(x,y) \in I \mid f(x,y)=1\}$ et $I(n \times m)$ est le support de l'image avec m le nombre de lignes et n le nombre de colonnes[10].

Les moments de Hu, Les moments de Zernike sont des descripteurs globaux puisqu'ils prennent en compte l'organisation intérieure de la forme du caractère et un petit nombre de ces moments sert pour décrire un caractère. Pour ces raisons, nous

nous sommes intéressés à ce genre d'attributs.

b) Descripteurs externes [10]

Ils sont basés sur la frontière. Comme pour une forme, l'image I d'un contour peut être vue comme une application :

$g(x, y) \rightarrow \{0, 1\}$ où $(x, y) \epsilon$ I sont les coordonnées des pixels de l'image. Si le pixel (x, y) est en dehors du contour $g(x, y) = 0$ et $g(x, y) = 1$ s'il appartient au contour. comme : l'extraction de contour, les descripteurs de fourrier

1.4.3 L'apprentissage

Le rôle du module d'apprentissage consiste à caractériser chaque classe par les paramètres définissant la forme (c'est la détermination des espaces d'appartenance des individus).

Supposons que des formes soient définies par des vecteurs de R2 (donc deux paramètres réels) et supposons que nous disposons d'échantillons de deux classes d'objets. On peut alors représenter ces échantillons dans l'espace R^2 des paramètres (figure 1.3). L'apprentissage peut alors par exemple consister à trouver automatiquement une courbe séparant les échantillons de chacune des deux classes. C'est la recherche de cette courbe qui va être l'apprentissage des deux classes :

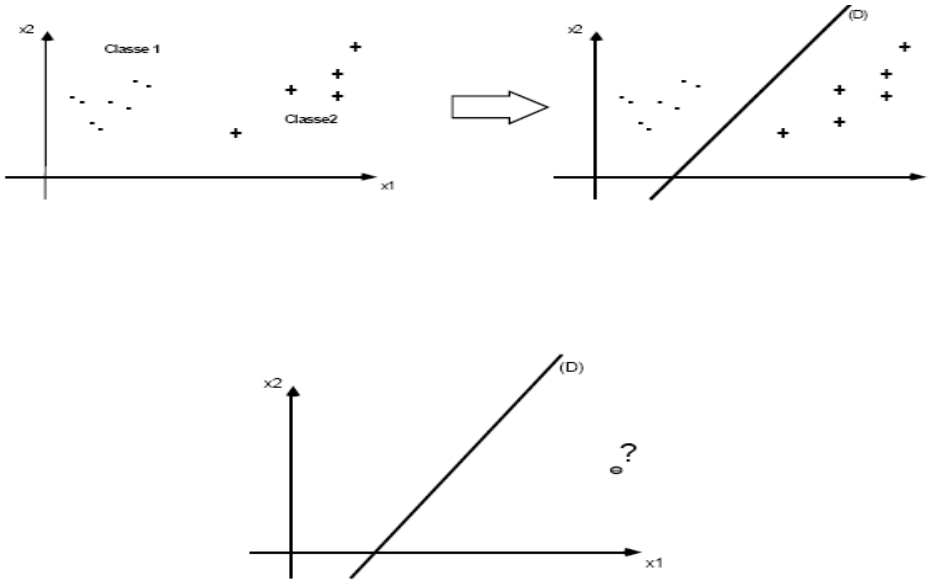


Figure 1.3. Exemple d'apprentissage et de règle de décision induite

a) Les procédés d'apprentissage

Les procédés d'apprentissage sont différents suivant qu'il s'agisse de caractères imprimés ou manuscrits, et suivant qu'il s'agisse de reconnaître une ou plusieurs fontes. Nous donnons dans la suite quelques exemples de procédés d'apprentissage [5].

- ***L'apprentissage manuel***

Il est réalisé lors d'une étape préliminaire à la reconnaissance, en introduisant un grand nombre d'échantillons de référence. L'utilisateur joue le rôle de professeur pour indiquer le nom de chaque échantillon. Le choix des caractères de référence est fait à la main en fonction de l'application. Le nombre d'échantillons peut varier de quelques unités à une dizaine par caractères. Ce type d'apprentissage est appelé apprentissage supervisé.

- ***L'apprentissage automatique***

Ce deuxième type d'apprentissage consiste à doter le système d'un mécanisme automatique lui permettant de trouver les classes de référence avec une assistance minimale. Des échantillons sont introduits en grand nombre par l'utilisateur sans indication de leur classe (apprentissage non supervisé). Ce type d'apprentissage est intéressant car il permet de renseigner l'utilisateur sur les ambiguïtés entre les caractères afin d'agir en conséquence, par exemple, en ajoutant des échantillons pour renforcer la représentative d'une classe.

- ***L'apprentissage continu***

Afin de pouvoir s'adapter à un nouveau texte, le système peut compléter ou affiner sa base d'apprentissage en rajoutant des échantillons extraits du forme étudié. L'apprentissage se fait ainsi de manière continue.

1.4.4 Classification

Classer un ensemble d'objets, c'est attribuer à chacun une classe ou « catégorie » parmi plusieurs classes définies à l'avance. Cette tâche est appelée « classification » ou « discrimination ». Un algorithme qui réalise automatiquement une classification est appelé classifieur [18][10][3].

Lorsque les données sont étiquetées, la reconnaissance consiste à décider de l'étiquette à attribuer à tout nouvel individu en fonction de sa position dans l'espace des représentations.

Réaliser la classification d'une base d'observations (base d'apprentissage),

consiste à trouver les frontières des classes qui la compose. Le terme de classification regroupe en fait trois cas de figure qui impliquent des stratégies différentes. Ces classificateurs peuvent être séparés en trois catégories distinctes : les classificateurs paramétriques, les classificateurs non paramétriques et les classificateurs dits « neuronaux »,

a) Les classificateurs paramétriques

La classification est basée sur une paramétrisation des lois de distribution. Pour déterminer la segmentation idéale de l'espace des attributs, il est souhaitable de connaître la distribution statistique des valeurs prises par un attribut, pour une classe donnée (on fait appel à une famille paramétrable de densité ou de surfaces et on optimise les paramètres pour minimiser la probabilité d'erreur). On applique alors la théorie statistique de décision (Bayes, vraisemblance maximum, etc.). la forme analytique des distributions les plus courantes dépend d'un certain nombre de paramètres (comme la moyenne, l'écart type de la distribution gaussienne). La classification est d'abord calculée analytiquement en fonction de ces paramètres. On estime ensuite la valeur de ces paramètres en fonction des données disponibles pour les substituer dans le résultat analytique.

b) Les classificateurs non-paramétriques

Si en revanche, aucune des techniques de paramétrisation d'une distribution inconnue n'est applicable, on utilise alors les données pour estimer la forme de la distribution. On développe un algorithme qui converge vers la densité (estimer les densités de probabilité des classes sans effectuer, ou presque, d'hypothèse sur leurs lois) ou la surface idéale qu'elle soit (moyennant certaine hypothèse)

c) Les classificateurs neuronaux

Intègrent des fonctions discriminantes à la suite d'un apprentissage par l'exemple. C'est à dire une phase d'apprentissage automatique à partir d'une base d'apprentissage est nécessaire. Ensuite les performances du classifieur sont évaluées sur une base de test pour vérifier qu'il est capable de généraliser les connaissances apprises à des formes différentes de celles utilisées pour l'apprentissage. C'est la phase de généralisation.

1.5 Conclusion

Nous avons présenté dans ce chapitre des généralités sur le domaine de la RdF, les techniques pouvant être utilisées, les problèmes à résoudre et les différentes étapes

dans un système de RdF, a savoir, prétraitement, extraction des primitives, classification (figure 1.1).

En ce sens, les chapitres suivants seront consacrés à la présentation des étapes de système de RdF. Comme l'étape de prétraitement est bien expliquée dans ce chapitre, le chapitre suivant sera consacré essentiellement aux paramètres caractéristiques utilisés.

CHAPITRE 2

LES PARAMETRES CARACTERISTIQUES

CHAPITRE 2

LES PARAMETRES CARACTERISTIQUES

L'extraction de paramètres caractéristiques -en anglais, « feature extraction »- est une étape de grande importance. Si elle est mal conçue, il sera difficile, voire impossible, d'effectuer une reconnaissance efficace.

Dans ce chapitre nous allons caractériser la forme des chiffres par des paramètres que nous voyons pertinents pour leur forme. Pour cela nous allons détecter sur le chiffre des zones caractéristiques : différents types de cavités, les histogrammes de projection, le zonage, le profil, les moments géométriques, et les moments Zernike.

2.1 Introduction

Pour la reconnaissance de chiffres imprimés, les primitives recherchées doivent présenter une certaine invariance géométrique d'une part, et une certaine invariance statistique d'autre part. La première signifie une tolérance vis-à-vis des opérateurs de translation, de rotation, et de changement d'échelle, tandis que la seconde signifie une tolérance vis-à-vis du bruit inévitablement présent sur chaque caractère [3].

Les attributs à extraire sont discriminantes le plus possible. Dans ce chapitre nous allons présenter quelques paramètres qui répondent aux exigences, pas nécessairement tous en même temps de l'invariance géométrique mentionnées dans le premier chapitre & 1.4.2 : les cavités, les histogrammes de projection, le zonage, le profil, les moments géométriques et les moments Zernike

2.2 Détection des cavités [2]

Nous définissons 5 types de cavités : Est, Ouest, Nord, Sud, et Centrale. Cela suppose la définition de 4 directions cardinales dans l'image : Ouest vers la gauche, Nord vers le haut, etc. [2].

Par exemple, un point de l'image appartient à une cavité Ouest si est seulement si les conditions suivantes sont simultanément vérifiées :

- Le point n'appartient pas au tracé.
- A partir, de ce point, en se déplaçant en ligne droite vers l'Ouest, on ne croise pas le tracé.
- A partir de ce point, en se déplaçant en ligne droite vers le Nord, le Sud ou l'Est, on croise le tracé.

Par analogie, on peut facilement en déduire les définitions des cavités Nord, Sud, Est. Par contre un point de l'image appartient à une cavité centrale si et seulement si les conditions suivantes sont simultanément vérifiées :

- Le point n'appartient pas au tracé.
- A partir de ce point, en se déplaçant en ligne droite vers l'Ouest, le Nord, le Sud, ou l'Est, on croise le tracé.

La figure 2.1 montre les cavités détectées sur les chiffres. Chaque type de cavité est représenté par une couleur (Est : rouge, Ouest : vert, Sud : bleu, Centrale : violet).

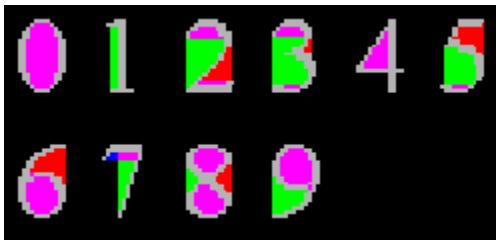


Figure 2.1. Les différentes cavités

Les cavités peuvent être détectées en faisant des intersections de dilatations par des demi-droites, Appelons E la demi-droite qui s'étend vers l'Est, O celle qui, s étend vers l'Ouest, N vers le Nord et S vers le Sud.

Si on dilate l'image d'un chiffre avec la demi-droite E, on obtient une image sur laquelle les points blancs sont : soit des points du tracé ; soit des points à partir desquels, en se déplaçant vers l'Est, on rencontre le tracé.

Il s'ensuit que l'image CO des cavités Ouest est obtenue ainsi :

$$CO = (I \oplus E) and \overline{(I \oplus W)} and (I \oplus N) and (I \oplus S) and \bar{I} \qquad \textbf{(2-1)}$$

Où ‘and’ désigne le ET logique, pixel par pixel, entre deux images binaires, $(I \oplus E)$ désigne la dilatation de l’image I par la demi-droite E, et la barre supérieur désigne la négation, c'est-à-dire l’inversion des 0 et des 1. Cette équation ne fait que traduire la définition des cavités ouest.

A partir de l'image d'un chiffre, le programme calculait les images CE, CS, CO, CN, et CC, représentant respectivement les cavités Est, Sud, Ouest, Nord, et Centrales.

Dans notre travail, nous allons utiliser comme paramètres caractéristiques, les surfaces relatives des différents types de cavités. Par exemple, notons s_{ce} la surface relative des cavités Est et S_{CE} leur surface absolue (en nombre de pixels). On a alors :

$$s_{ce} = \frac{S_{CE}}{S_{CE}+S_{CW}+S_{CS}+S_{CN}+S_{CC}+1} \qquad \textbf{(2-2)}$$

Une définition similaire est utilisée pour les surfaces relatives des autres cavités. L'ajout de ‘1’ au dénominateur a en général une influence négligeable sur le résultat. Son objectif est uniquement d'éviter une division par zéro dans le cas d'une forme qui ne comprendrait aucune cavité.

2.3 Les histogrammes de projection

Les histogrammes de projection sont depuis très longtemps utilisés dans le domaine de la reconnaissance de formes. Leur principe est de sommer le nombre de pixels noirs de chacune des lignes (respectivement des colonnes) de l'image binaire de la forme (figure 2.2).

Les projections des histogrammes ont été introduites en 1956 par Glauberman, cette technique est principalement utilisée pour la segmentation des caractères, des mots et de lignes de texte, ou pour détecter si une image d'un texte numérisé est tournée.

La projection horizontale (respectivement verticale) est le nombre de pixel de chaque ligne (respectivement colonne). Elles peuvent être indépendantes de l'échelle en divisant par le nombre total de pixels imprimés en caractère d'image.

Toutefois, les projections d'histogrammes sont très sensibles à la rotation et, dans une certaine mesure, la variabilité dans le style d'écriture. En outre, des informations importantes sur le caractère forme semble être perdu [9].

Figure 2.2. Histogramme horizontale et verticale du chiffre 5

2.4 Zoning (densités)

C'est la densité de pixels noirs calculés dans différentes zones (zoning) de l'image du chiffre. Pour obtenir ces mesures, on "découpe" horizontalement et verticalement le rectangle englobant le chiffre en zones de taille égale; Le nombre de pixels noirs dans chaque zone forme alors les composantes du vecteur de caractéristiques. En découpant par exemple l'image en « n » zones verticales (d'égale largeur) et « m » zones horizontales (d'égale hauteur), on obtient un vecteur à nxm composantes.(notre cas 2x2) [9]

Les densités dans chaque zone devront être normalisées (en les divisant par

exemple par la surface de la zone-moyenne-) puisque les chiffres ne sont pas tous de même taille.

2.5 Profils

Les caractéristiques à extraire des chiffres sont leurs profils gauche et droite. Pour obtenir ces mesures, on détermine sur un certain nombre de lignes, en général, réparties uniformément sur la hauteur du chiffre, la distance entre le bord gauche (respectivement. droite) du chiffre et le premier pixel noir rencontré sur cette ligne; L'ensemble de ces distances définit un profil gauche (respectivement. droit) du chiffre.

Les profils gauche et droit doivent être normalisés (en les divisant par exemple par la largeur de l'image du chiffre traité) puisque les chiffres ne sont pas tous de même taille.

2.6 Les moments géométriques

Les moments sont souvent utilisés en physique pour décrire la répartition des masses dans un corps. En associant le niveau de gris d'un point de l'image à la masse d'un corps en chaque point, on peut reprendre le même formalisme pour décrire la répartition des niveaux de gris dans un objet. Dans le cas d'un objet binaire où g(k,l) prend la valeur « 1 » à l'intérieur (si le point appartient à l'objet) et « 0 » à l'extérieur, les différents moments fournissent des informations importantes concernant l'arrangement spatial des points de l'objet [1].

Donc nous pouvons définir, comme en mécanique et en statistique, une double suite de coefficients appelés les moments géométriques. Le principe d'utilisation des moments est de sélectionner un sous-ensemble de M_{ij} qui contienne suffisamment d'information pour caractériser l'image de façon unique, pour une application donnée.

Pour une image numérisée et discrète (codage rétinien) de taille K*L, alors g(k,l) est la fonction décrivant le contenu de l'image en chacun de ses points de coordonnée k,l ; i et j étant deux entiers naturels quelconques :

M_{ij} est la définition du moment géométrique d'ordre (i+j) avec la répétition |i - j| [10].

$$M_{ij} = \sum_{k=1}^{K} \sum_{l=1}^{L} (k)^i (l)^j g(k,l) \qquad \textbf{(2-3)}$$

où g(k,l) est le niveau de gris de l'image défini. $(k)^i (l)^j$ est la base

L'esemble complet de moments d'ordre n :{ M_{ij}, i+j<=n } est (n+1)(n+2)/2, où

les moments d'ordre n représentent une propriété caractéristique :

- **Ordre 0 :** Représente le poids total de l'image M_{00} (surface de l'objet c. à. d nombre de pixels)[8].
- **Ordre 1 :** Permet de déterminer les coordonnées du centre de gravité de l'image (x_g,y_g) [8] :

$$x_g = \frac{M_{10}}{M_{00}} \qquad \textbf{(2-4)}$$

$$y_g = \frac{M_{01}}{M_{00}} \qquad \textbf{(2-5)}$$

- **Ordre 2 :** Donne unre représentation de la distribution des pixels d'un objet autour de son centre de gravité. Il permet de calculer les axes principaux de l'objet, son orientation,sa taille, son rayon de giration. Si seul les moments jusqu'à l'ordre 2 utilisés, une reconstruction de l'image originale amènera à une ellipse [8].
- **Ordre 3 :** Dissymétrie (des projections).
- **Ordre 4 :** Applatissement (des projections)**.**

Les ensembles $\{m_{i0}\}$ et $\{m_{0j}\}$ peuvent être considérés comme les moments de la projection de l'image suivant l'axe des x et l'axe des y.

Afin de rendre les moments M_{ij} invariants par translation, on les définit en choisissant (x_g,y_g) comme origine. On parle alors de moments centrés μ_{ij} dont la définition formelle est donné par l'équation :

$$\mu_{ij} - \sum_{k=1}^{K}\sum_{l=1}^{L}(k - x_g)^i (l - y_g)^j g(k,l) \qquad \textbf{(2-6)}$$

Les moments centrés peuvent se calculer au moyen de la relation précédente ou en fonction des moments non centrés par la relation [1] [8] :

$$\mu_{ij} = \sum_{r=0}^{i}\sum_{s=0}^{j} C_r^i C_s^j (-x_g)^{i-r} (-y_g)^{j-s} M_{rs} \qquad \textbf{(2-7)}$$

Avec :

$$C_b^a = \frac{a!}{b!(a-b)!} \qquad \textbf{(2-8)}$$

Il est assez facile d'exprimer les moments centrés en termes de moments géométriques. Pour les neuf premiers, nous avons [11] :

$$\mu_{00} = M_{00} \quad \textit{(2-9)}$$

$$\mu_{10} = \mu_{01} = 0 \quad \textit{(2-10)}$$

$$\mu_{11} = M_{11} - (M_{00}.x_g.y_g) \quad \textit{(2-11)}$$

$$\mu_{20} = M_{20} - (M_{00}.x_g^2) \quad \textit{(2-12)}$$

$$\mu_{02} = M_{02} - (M_{00}.y_g^2) \quad \textit{(2-13)}$$

$$\mu_{21} = M_{21} - (M_{20}.y_g) - (2\,M_{11}.x_g) - (2\,M_{00}.x_g^2 y_g) \quad \textit{(2-14)}$$

$$\mu_{12} = M_{12} - (M_{02}.x_g) - (2\,M_{11}.y_g) - (2\,M_{00}.y_g^2 x_g) \quad \textit{(2-15)}$$

$$\mu_{30} = M_{30} - (3\,M_{20}.x_g) + (2\,M_{00}x_g^3) \quad \textit{(2-16)}$$

$$\mu_{03} = M_{03} - (3\,M_{02}.y_g) + (2\,M_{00}y_g^3) \quad \textbf{(2-17)}$$

En divisant chaque moment μ_{ij} par $(\mu_{00})^{(i+j+2)/2}$ en construit les moments normalisés η_{ij} qui sont invariants par homothétie (changement d'échelle, grossissement)

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{((i+j+2)/2}} \quad \textbf{(2-18)}$$

Les 7 moments de Hu sont définis A partir des moments normalisés η_{ij} d'ordre 2 et 3, qui sont invariants en translation, échelle et rotation [10][12]:

$$\emptyset_1 = \eta_{20} + \eta_{02} \quad \textbf{(2-19)}$$

$$\emptyset_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad \textbf{(2-20)}$$

$$\emptyset_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad \textbf{(2-21)}$$

$$\emptyset_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 \quad \textbf{(2-22)}$$

$$\emptyset_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03}) + (\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad \textbf{(2-23)}$$

$$\emptyset_6 = (\eta_{20} + \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) + 4\eta_{11}(\eta_{03} + \eta_{12})\ (\eta_{21} + \eta_{03}) \quad \textbf{(2-24)}$$

$$\emptyset_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad \textbf{(2-25)}$$

Les six premiers moments sont invariants : Aux translations et Aux changements, problématique quand il s'agit de reconnaître des images « miroirs ». C'est pourquoi

M. K. Hu exploite aussi dans son système le moment sept qui n'est pas invariant aux réflexions. Ce dernier change de signe lorsqu'une telle transformation est appliquée à l'image et permet donc de détecter celle-ci.

Nous prenons comme exemple les chiffres 5 et 3, et nous leur faisons subir quelques translations, grossissements.

La table2.1. montre que les moments Hu sont invariants à la translation.

La table2.2. montre qu'il sont invariant aux changements d'échelle (utilisant les chiffres de la figure 2.3).

La table2.3. montre qu'ils sont invariants à la rotation.

Table 2.1. Les moments Hu pour le chiffre 5 avec différents translation.

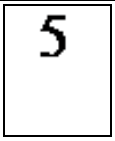
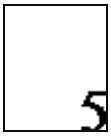
		
Φ1	0.4776	0.4776
Φ2	0.0750	0.0750
Φ3	0.0027	0.0027
Φ4	0.0016	0.0016
Φ5	0.0000	0.0000
Φ6	0.0001	0.0001
Φ7	0.0000	0.0000

Table 2.2 Les moments Hu pour le chiffre 5 avec différentes taille.

				
Φ1	0.2963	0.3516	0.4160	0.4776
Φ2	0.0228	0.0340	0.0530	0.0750
Φ3	0.0009	0.0013	0.0012	0.0027
Φ4	0.0004	0.0021	0.0018	0.0016
Φ5	0.0000	0.0000	0.0000	0.0000
Φ6	0.0000	0.0004	0.0003	0.0001
Φ7	0.0000	-0.0000	0.0000	0.0000

Table 2.3. Les moments Hu pour le chiffre 3 avec différentes rotations.

	0° 	**11°** 	**45°** 	**90°**	**120°** 	**-37°** 	**-70°** 
Φ1	0.3712	0.2628	0.2586	0.2984	0.2513	0.2677	0.2733
Φ2	0.0424	0.0175	0.0168	0.0283	0.0164	0.0164	0.0178
Φ3	0.0036	0.0007	0.0006	0.0012	0.0008	0.0006	0.0009
Φ4	0.0001	0.0001	0.0004	0.0004	0.0004	0.0003	0.0005
Φ5	-0.0000	-0.0000	0.0000	-0.0000	-0.0000	-0.0000	-0.0000
Φ6	-0.0003	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000
Φ7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

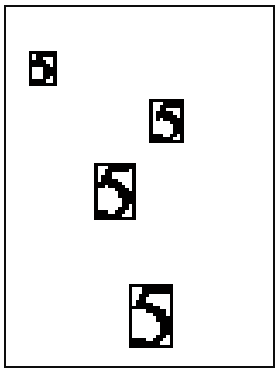


Figure 2.3. Le chiffre 5 avec différentes taille

2.7 Les moments de Zernike

Les moments géométriques fournissent un codage compact et facile à calculer, doté de bonnes propriétés d'invariance théorique. En fait, ce codage reste grossier si on conserve peu de termes et on n'a pas de méthodes permettant de déterminer le nombre idéal de moments pour un problème donné. Ce dernier inconvénient est surmonté par les moments de Zernike qui possèdent une propriété d'orthogonalité permettant la reconstruction (en général approchée) de la forme à partir des moments [4].

L'intérêt du calcul des moments de Zernike repose sur l'invariance par translation, par changement d'échelle et par rotation. Ils constituent un espace

vectoriel dans lequel l'image de la forme est projetée. Cela permet de s'affranchir du problème de normalisation rencontré dans le cas des moments de Hu, mais aussi d'aboutir à une description plus précise des formes, par utilisation d'ordres plus élevés, en maintenant les propriétés d'invariance.

Les polynômes de Zernike furent proposés en 1934 par Zernike. Dérivés de ces polynômes, les moments ont été utilisés par de nombreux auteurs en reconnaissance de caractères. Plusieurs études montrent également la supériorité de ces descriptions par rapport à d'autres approches [10], car elles réduisent le bruit, les redondances et peuvent être reconstruite.

2.7.1 Formule des moments de Zernike

Les moments Zernike A_{nm} sont défini comme des polynômes complexes qui forment un ensemble orthogonal complet du disque (cercle) unité $x^2+y^2<=1$. Ils correspondent à la projection de la forme f(x, y) sur une base ZP de fonctions orthogonales $V_{nm}(x, y)$ et se défini par [10] :

$$ZP = \{V_{nm}(x,y) | \, x^2 + y^2 \leq 1\} \qquad \textbf{(2-26)}$$

Où le polynôme complexe V_{nm} d'ordre n et de répétition m est défini tel que n - |m| soit pair et |m|<=n

$$V_{nm}(x,y) = V_{nm}(\rho,\vartheta) = R_{nm}(\rho)\exp(jm\theta) \qquad \textbf{(2-27)}$$

n : nombre entier positif ou nul

m : nombre entier positif ou négatif, tel que n-|m| pair et |m| <= n

ρ :longueur du vecteur à partir de l'origine au pixel (x,y)

θ : angle entre le vecteur ρ et l'axe x dans le sens inverse de l'aiguille d'une montre

$$V_{nm}(x,y) = R_{nm}(x,y)\exp(jm\arctan(y/x)) \qquad \textbf{(2-28)}$$

Le rapport polynomial est défini comme suit [10] :

$$R_{nm}(x,y) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!}(x^2+y^2)^{(n-2s)/2} \qquad \textbf{(2-29)}$$

Soit $\rho=(x^2+y^2)^{1/2}$ la longueur du vecteur à partir de l'origine du pixel (x,y) et θ=a tan2(x/y) l'angle entre l'axe x et ce vecteur. $R_{nm}(\rho)$, la représentation en coordonnées polaires ($x=\rho\cos\theta$, $y=\rho\sin\theta$) de $R_{nm}(x,y)$ est un polynôme de

degré n [10] :

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s!\left(\frac{n-|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!} (\rho)^{n-2s} \qquad \textbf{(2-30)}$$

Les moments de Zernike sont les projections de la fonction d'image f(x,y) dans la base orthogonale de fonction $V_{nm}(x,y)$. Le moment Zernike, d'ordre n et de répétition m, est un nombre complexe défini par :

$$A_{nm} = \frac{n+1}{\pi} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y)[V_{nm}(x,y)]^* \qquad \textbf{(2-31)}$$

où $x^2+y^2<=1$ et * définit le complexe conjugué

L'ordre des moments possède de grande influence sur la conservation de l'information angulaire. Plus l'ordre est élevé et plus les variations angulaires décrites sont fines.

Les moments Zernike sont bien connus pour être invariants à la rotation. Une normalisation d'image est nécessaire pour rendre ces moments invariants à la translation et au facteur d'échelle [10].

2.7.2 La relation entre les moments géométriques et Zernike

Les moments de Zernike sont liées aux moments μ_{ij}, une fois ces moments sont calculés nous avons également les moments Zernike, bien que, dans l'applications pratiques, il est beaucoup plus simple pour calculer les moments Zernike directement qu'à partir de leur définition [8][11][24].

$$A_{nl} = [(n+1)/\pi] \sum_{k=l}^{n} \sum_{j=0}^{q} \sum_{m=0}^{l} (-i)^m \binom{q}{j} \binom{l}{m} B_{nlk}\ \eta_{k-2j-l+m,2j+l-m} \qquad \textbf{(2-32)}$$

avec : q=(k-l)/2, (l-k) et (n-k) pairs

$$B_{nlk} = \frac{(-1)^{(n-k)/2}[(n+k)/2]!}{[(n-k)/2]![(l+k)/2]![(k-l)/2]!} \qquad \textbf{(2-33)}$$

$\binom{j}{s}$ Étant le nombre de combinaisons de s élément parmi j

A titre d'illustration, nous proposons ci-après l'expression de quelques moments issus de l'application précédente :

$$A_{00} = \frac{\eta_{00}}{\pi} = 1/\pi \qquad \textbf{(2-34)}$$

$$A_{11} = A_{1,-1} = 0 \qquad \textbf{(2-35)}$$

$$A_{22} = 3(\eta_{02} - \eta_{20} - 2i\eta_{11})/\pi \qquad \textbf{(2-36)}$$

$$A_{20} = 3(2\eta_{20} - 2\eta_{02} - 1)/\pi \quad \textbf{(2-37)}$$

$$A_{33} = 4[\eta_{03} - 3\eta_{21} + i(\eta_{30} - 3\eta_{12})]/\pi \quad \textbf{(2-38)}$$

$$A_{31} = 12[\eta_{03} + \eta_{21} - i(\eta_{30} + \eta_{12})]/\pi \quad \textbf{(2-39)}$$

$$A_{44} = 5[\eta_{40} - 6\eta_{22} + \eta_{04} + 4i(\eta_{31} - \eta_{13})]/\pi \quad \textbf{(2-40)}$$

$$A_{42} = 5\{4(\eta_{04} - \eta_{40}) + 3(\eta_{20} - \eta_{02}) - 2i[4(\eta_{31} + \eta_{13}) - 3\eta_{11}]\}/\pi \quad \textbf{(2-41)}$$

$$A_{40} = 5[6(\eta_{40} + 2\eta_{22} + \eta_{04}) - 6(\eta_{20} - \eta_{02}) + 1]/\pi \quad \textbf{(2-42)}$$

Les moments Zernike invariants jusqu'à l'ordre 12, sont dans l'annexe (A) :

Nous prenons les mêmes exemples (images) précédentes pour démontrer l'invariance à la translation, échelle, et la rotation (table 2.4., table 2.5., table 2.6.)

Table 2.4. Les moments Zernike pour différentes translations.

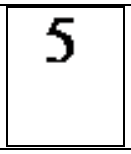
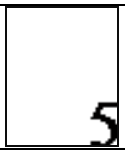
		
S1	0.0427	0.0427
S2	0.2616	0.2616
S5	0.4807	0.4807
S11	1.8937	1.8937
S23	0.2970	0.2970
S35	38.0280	38.0280
S49	4.0349	4.0349
S45	11.4349	11.4349

Table 2.5. Les moment Zernike pour différentes tailles.

				
S5	0.1772	0.4396	0.4198	0.4807
S13	0.3011	0.3956	0.4685	0.6256
S22	0.1151	0.5376	0.5489	1.0468
S34	0.0443	0.1012	0.2131	0.3404
S46	0.3775	0.9997	1.8378	3.7585
S47	0.0429	0.0886	0.2152	0.3637

Table 2.6 : Les moments Zernike pour le chiffre 3 avec différentes rotations.

	0° 	**11°** 	**45°** 	**90°**	**120°** 	**-37°** 	**-70°** 
S7	0.0662	0.0370	0.0271	0.0391	0.0147	0.0349	0.0318
S19	0.9817	0.1821	0.3698	0.4556	0.3251	0.1706	0.3238
S25	0.7284	0.4252	0.3649	0.3308	0.2930	0.3493	0.2112
S37	0.6819	0.1827	0.1353	0.2594	0.1052	0.1504	0.1495
S45	0.5789	0.4333	0.3313	0.1860	0.0766	0.0720	0.0307
S47	0.0742	0.0149	0.0072	0.0106	0.0059	0.0099	0.0058

2.8 Conclusion

Les caractéristiques choisies pour décrire ou quantifier les caractères sont très nombreuses. Dans ce chapitre, nous avons cité quelques unes parmi les plus représentatives : les cavités, l'histogramme, le zonage, le profil, les moments Hu et les moments Zernike.

Les cavités sont invariantes par rapport au changement d'échelle puisque nous avons utilisé la surface des cavités par rapport à leur surface globale (normalisation), la même chose pour l'histogramme, le zonage et le profil; mais sont sensible à la rotation. Par contre les moments Hu et les moments Zernike sont invariant à la translation, au grossissement, et à la rotation.

Pour réaliser une reconnaissance automatique de chiffres, il est nécessaire après le calcule des paramètres discriminants sur un chiffre, de fournir ces paramètres à un classifieur, c'est-à-dire Il faut noter que les paramètres caractéristiques ne sont pas totalement indépendantes des méthodes de décision et classement, qui sont décrites dans le chapitre suivant.

CHAPITRE 3

LA CLASSIFICATION

CHAPITRE 3

LA CLASSIFICATION

La classification est l'étape de décision qui réalise véritablement la reconnaissance : choix de la classe dont la représentation ou le modèle est le plus proche.

Il existe de nombreuses méthodes de classification. Dans ce chapitre nous allons présenter le classifieur le plus simple KPPV et son variante extension KPPV floue et les réseaux de neurones qui constituent actuellement un des outils les plus efficaces pour le traitement des problèmes de classification (dans notre cas il s'agit de classifier des chiffres imprimés)

3.1 Introduction

Il existe de nombreuses méthodes de classification :

- K plus proches voisins et ses variantes
- Arbres de Décisions
- Naïve Bayes (ou encore Simple Bayes)
- Réseaux de Neurones
- Machines à Support de Vecteurs (ou SVM)
- Programmation Génétique

Dans ce chapitre, nous allons présenter premièrement la méthode des KPPV « k plus proches voisins » (Plus connues en anglais sous le nom K-nearest neighbor (K-NN), ou encore Memory Based Reasoning) [1]. Il s'agit d'une méthode de classement non paramétrique très utilisés. Nous présentons, ensuite une de ses variantes la méthode, KPPV floue. Finalement nous présentons les Réseaux de Neurones qui sont considérés comme un outil privilégié pour la reconnaissance de caractères manuscrits ou imprimés dans des polices d'écriture variées.

3.2 K plus proches voisins

K plus proches voisins est une méthode de raisonnement à partir de cas. Elle part de l'idée de prendre des décisions en recherchant un ou des cas similaires déjà résolus en mémoire. Contrairement aux autres méthodes de classification (Arbres de Décision, Réseaux de Neurones, Algorithmes Génétiques) [26].

C'est l'une des méthodes les plus simples d'apprentissage automatique supervisé (Apprentissage ultra simple - Pas d'apprentissage simplement stockage de données d'apprentissage-). C'est une méthode qui est basé sur un algorithme de la famille des algorithmes dits « paresseux » à l'inverse de beaucoup d'autres méthodes d'apprentissage automatique (les réseaux de neurones artificiels, les méthodes à noyaux, etc.).

3.2.1 Principe de classifieur KPPV

Etant donnée une base d'apprentissage d'images, pour prédire la classe d'un nouveau cas, le classifieur KPPV cherche les K plus proches voisins de ce nouveau cas et prédit la réponse la plus fréquente de ces K plus proches voisins. La méthode utilise donc deux paramètres : le nombre K et la fonction de similarité pour comparer le nouveau cas aux cas déjà classés. Le principe est donné par :

1. choix d'un entier k (souvent k = nombre d'attributs + 1, ou k=√n).
2. calcul des distances (ex: distance euclidienne, c'est la distance la plus populaire).
3. retenir les k observations pour lesquelles ces distances sont les plus petites (les k plus proches voisins d'un cas de référence –quand on parle de voisin cela implique la notion de distance ou de dissimilarité-).
4. compter les nombres de fois où ces k observations apparaissent dans chacune des classes (déterminer les classes correspondantes).
5. Choisir la classe la plus représentée.

Ces étapes sont résumées comme suit [25] :

Pour une forme inconnue x à classer, nous allons examiner la distance de x à tous les échantillons (qui définissent toutes les classes), puis nous sélectionnons les K plus proches échantillons et nous affectons x à la classe majoritaire parmi ces K échantillons.

Dans l'exemple de la figure 3.1. et pour cinq voisins on classerait x dans la classe w_1:

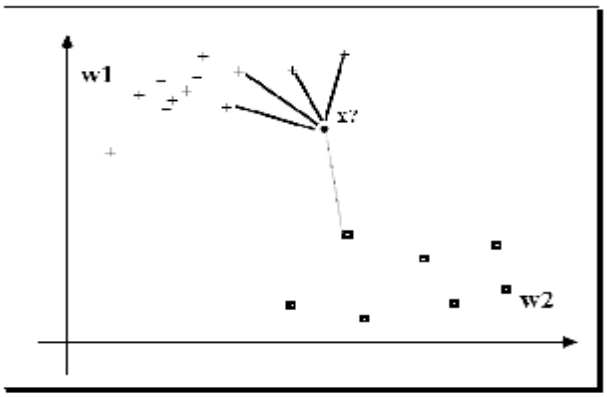


Figure 3.1. Méthode de K_plus proches voisins

3.2.2 Choix du nombre K

Le choix du paramètre K dans la règle des KPPV a une influence directe sur le style du classifieur ainsi déterminé. Une faible valeur de K va donner un classifieur de bonne résolution (définir des frontières compliquées entre classes) mais très sensible au bruit sur les échantillons et sur le vecteur à classer. Une valeur grande de K aura un comportement inverse, lissant les frontière mais peu sensible au bruit [4].

3.2.3 Définition de la distance

Le choix de la distance est primordial au bon fonctionnement de la méthode. Quoique les distances les plus simples permettent d'obtenir des résultats satisfaisants [26].

Une distance doit avoir quatre propriétés pour tous les vecteurs a, b et c [13] :

- D (a, b) ≥ 0 (non-négativité)
- D (a, b) = 0 si et seulement si a = b (réflexivité)
- D (a, b) = D (b, a) (symétrie)
- D (a, b) + D (b, c) ≥ D (a, c). (triangle inégalité)

On peut cependant noter qu'un point « a » peut avoir un plus proche voisin « b » tandis que « b » possède de nombreux voisins plus proches que « a ».

Le choix de la distance se fait en fonction des connaissances préalables du problème. Il est possible de choisir la distance en faisant varier cette distance et, pour chacun des choix, estimer l'erreur réelle. On choisit alors la distance donnant la meilleure erreur réelle estimée. Plusieurs types pour les distances :

- distance Euclidienne.
- distance “city block“ ou ‘Manhattan’ (somme des valeurs absolues)
- distance de Tchebycheff.
- distance de Mahanalobis
- .....

Dans notre cas nous avons utilisé la distance Euclidienne (la distance la plus populaire), sa formule en « d » dimensions est la suivante :

$$D(a,b) = \left(\sum_{k=1}^{d} (a_k - b_k)^2\right)^{1/2} \qquad \textbf{(3-1)}$$

3.2.4 Sélection de la classe

La méthode la plus simple est de rechercher le cas le plus proche et de prendre la même décision. C'est la méthode 1-PPV (1-NN) du plus proche voisin. Si cette méthode peut fournir de bons résultats sur des problèmes simples pour lesquels les points sont bien répartis en groupes denses de même classe, en règle générale, il faut considérer un nombre de voisins plus important pour obtenir de bons résultats [26].

Une première façon de combiner les k classes des k voisins les plus proches est le vote majoritaire. Elle consiste simplement à prendre la classe majoritaire.

Une seconde façon est le vote majoritaire pondéré. Chaque vote, c'est-à-dire

chaque classe d'un des k voisins sélectionnés, est pondéré. Le poids est inversement proportionnel à la distance entre le cas à classer et les autres k plus proches voisin classes (Une telle pondération s'appelle un noyau).

Dans les deux cas précédents, il est possible de définir une confiance dans la classe attribuée égale au rapport entre les votes gagnants et le total des votes.

3.2.5 Exemple d'application

Considérons l'exemple de la figure 3.2.a. où l'on doit classer le nouveau cas x_q. Si on choisit K = 1, x_q sera classé « + ». Si K = 5, le même x_q sera classé « - ». On voit donc que le choix de K est très important dans le résultat final [25]

Sur la figure 3.2.b. on a représenté les exemples par des points. Chaque surface autour d'un exemple montre les positions possibles de nouveaux cas à classer où le résultat de la classification sera la classe de l'exemple si K=1 (on prend simplement le point le plus proche), On obtient un diagramme de **Voronoi**.

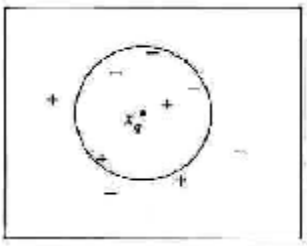


Figure 3.2.a. K_PPV

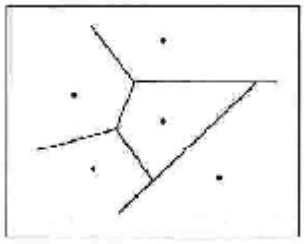


Figure 3.2.b. Voronoi

En cas d'égalité, la bonne solution consiste à tirer au hasard la classe à attribuer au point ambigu ou de pondérer les "votes" de chaque exemple par sa distance au point.

3.2.6 Critiques de la méthode KPPV

KPPV est un algorithme (& 3.2.1) de reconnaissance des formes qui a prouvé son efficacité face au traitement de données textuelles et les expériences montrent qu'il résiste bien aux données bruitées. Par contre, ils requièrent de nombreux exemples. Il a l'avantage d'être extrêmement simple et de donner en général de bons résultats.

On a souvent tendance à penser que les méthodes simples sont peu performantes. Ce n'est pas toujours le cas. Ainsi, nous pouvons démontrer que la méthode de PPV est asymptotiquement la meilleure méthode possible en termes de taux de bonne classification. Cela signifie que lorsque la base d'apprentissage est très

grande, nous ne pouvons pas faire mieux que cette méthode. Mais son inconvénient majeur reste le fait que le temps de calcul devient extrêmement important lorsque la base d'apprentissage contient beaucoup d'exemples [2].

En effet à chaque nouvelle classification, il est nécessaire de parcourir l'ensemble de la base d'apprentissage. Cet algorithme est donc coûteux en terme de temps mais offre, en général, des résultats parfaitement satisfaisants.

3.3 K plus proches voisins floue

Le principe de la méthode KPPV floue est similaire à la règle des KPPV classiques. Il se base sur une étape de recherche des k points voisins les plus proches du prototype à étudier. On examine parmi les K voisins retournés, le taux de mélange des classes. Le KPPV flou permet en plus, d'attribuer suivants les distances aux classes des k prototypes voisins, des degrés d'appartenance à ces classes.

Le point à classer x_i se voit attribuer un coefficient d'appartenance μ_{ji} à chaque classe j. Celui-ci est fonction des distances et des coefficients d'appartenance de ses k plus proches voisins. Ces coefficients doivent vérifier l'appartenance à l'intervalle [0,1] de μ_{ji}, pour tous les i et tous les j.

$$\sum_{j=1}^{c} u_{ji} = 1 \quad \text{pour tous les i}, \qquad 0 < \sum_{i=1}^{N} u_{ji} \leq N \quad \text{pour tous les j}$$

Les coefficients d'appartenance d'un nouveau point Xi à la classe j est donnée par :

$$u_{ji} = \frac{\sum_{l=1}^{k} u_{jl} (|x_i - x_l|)^{\frac{m-1}{2}}}{\sum_{l=1}^{k} \left(\frac{1}{(|x_i - x_l|)^{\frac{m-1}{2}}} \right)} \qquad \textbf{(3-2)}$$

où μ_{jl} est le coefficient d'appartenance à la classe C_j,

μ_{jl} =1 si appartient à les K plus proches voisin μ_{jl}=0 si non, de lième observation, parmi les k plus proches voisins de xi.

La variable m quant à elle, détermine l'importance de la contribution de la distance dans le calcul de la fonction d'appartenance (contrôle l'efficacité de l'ampleur de la distance). C'est le paramètre de fuzzification :

si m croit, la contribution des voisins est davantage pondérée et la notion de distance perd de son importance, si m tend vers l'unité, la contribution des voisins les plus proches sera favorisée, ainsi la notion de distance prend de l'importance, si m vaut 2, la contribution de chaque voisin est pondérée par l'inverse de la distance

respective, au carré, qui sépare une observation de l'observation à classer [20][21][22].

3.4 Les Réseaux de Neurones

Il n'y a pas de définition universellement acceptée de Réseau de Neurones (RdN). On considère généralement qu'un RdN est constitué d'un grand ensemble d'unités (ou neurones), ayant chacune une petite mémoire locale. Ces unités sont reliées par des canaux de communication (les **connexions**), qui transportent des données numériques. Les unités peuvent uniquement agir sur leurs données locales et sur les entrées qu'elles reçoivent par leurs connexions.

La plupart des RdN ont une certaine capacité d'apprentissage. Cela signifie qu'ils apprennent à partir d'exemples, de même que les enfants apprennent à lire, on lui présente des exemples de lettres et de chiffres, écrits avec des écritures et des fontes différentes. À la fin de l'apprentissage, on attend de lui qu'il ait une capacité de généralisation à partir des exemples qui lui ont été présentés, sans qu'il ne soit jamais nécessaire de lui fournir une description analytique et discursive de la forme et de la topologie des chiffres et des lettres [18].

3.4.1 Modélisation d'un neurone

La figure 3.3. montre la correspondance entre un neurone biologique et 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 (synapse). Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals [27].

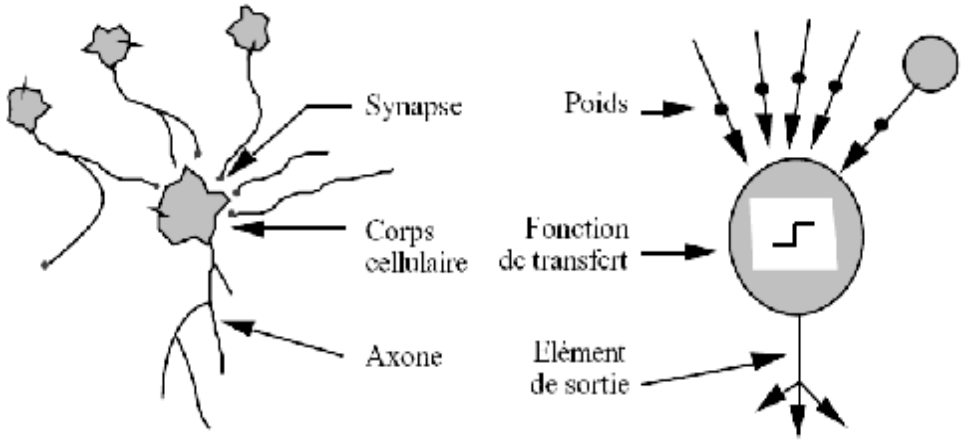


Figure 3.3. Mise en correspondance neurone biologique / neurone artificiel

On distingue deux phases pour le comportement d'un neurone. La première est habituellement le calcul de la somme pondérée des entrées (x) selon l'expression exprimé sur la figure 3.4.

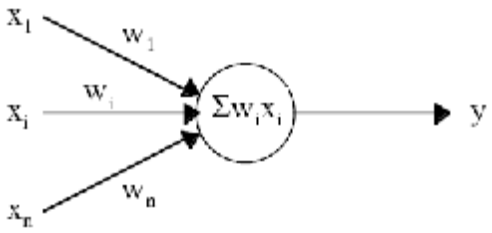


Figure 3.4. Un neurone artificiel

A partir de cette valeur, une fonction de transfert (fonction d'activation) calcule la valeur de l'état du neurone. Différentes fonctions de transfert, les trois les plus utilisés sont les fonctions «seuil» (en anglais «hardlimit»), «linéaire» et «sigmoïde». la plupart des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprisent dans l'intervalle [0, +1] ou [-1, +1].

3.4.2 Architectures de Réseaux de Neurones

On entend par architecture, la manière dont les neurones sont connectés. On distingue deux grands types : les RdN non bouclés et les RdN bouclés.

3.4.2.1 Les réseaux de neurones non bouclés

Les réseaux non bouclés, ou statiques (indépendants du temps), dans lesquels l'information circule des entrées vers les sorties, sans bouclage -retour en arrière- (Figure 3.5.); sont des systèmes statiques, utilisés principalement pour effectuer des tâches de classification, d'approximation de fonction non linéaire, de classification ou

de modélisation de processus statiques non linéaires [15][16][17].

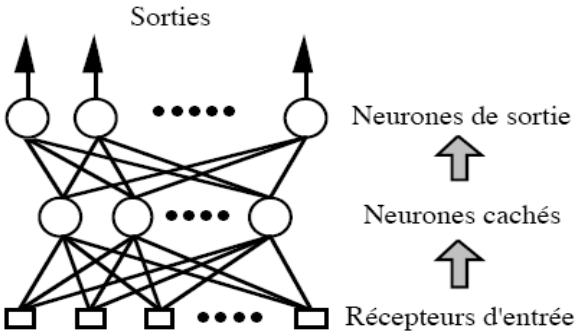


Figure 3.5. Réseau de neurones non bouclé

3.4.2.2 Les Réseaux de Neurones bouclés

Un réseau est bouclé (récurrents), ou dynamique, si son graphe possède au moins un cycle (ou des boucles qui ramènent aux entrées) (Figure 3.6.).

Il s'agit d'un registre à décalage qui permet d'introduire un retard (en anglais delay) dans une donnée que l'on veut acheminer dans un réseau [19]

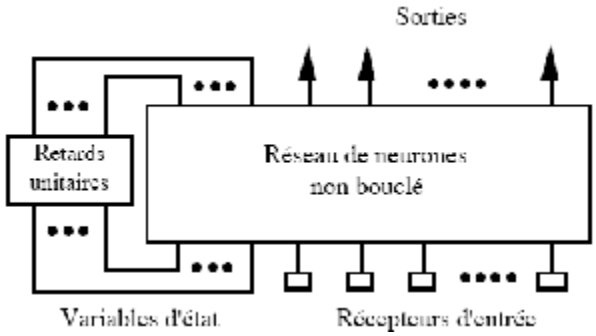


Figure 3.6. Réseau de neurones bouclé

3.4.3 Propriété fondamentale des Réseaux de Neurones

La propriété fondamentale des RdN remarquable qui est à l'origine de leur intérêt pratique dans des domaines très divers est « l'approximation parcimonieuse ». Cette expression traduit deux propriétés distinctes. D'une part, les RdN sont des approximateurs universels, et, d'autre part, une approximation à l'aide de RdN nécessite, en général, moins de paramètres ajustables (les poids des connexions) que les approximateurs usuels. Plus précisément, le nombre de poids varie linéairement avec le nombre de variables de la fonction à approcher, alors qu'il varie exponentiellement (beaucoup plus rapidement) avec la dimension de l'espace des

entrées dans le cas de la plupart des autres approximateurs usuels.

En vertu de cette propriété « l'approximation parcimonieuse », le comportement de tout système statique peut être approché par un RdN non bouclé, et celui de tout système dynamique par un réseau bouclé.

La propriété d'approximation parcimonieuse RdN d'excellents outils pour la résolution des problèmes (il faut s'assurer d'avoir bien posé le problème) de modélisation et de classification.

3.4.4 Apprentissage et généralisation

L'apprentissage est l'étape essentielle dans un RdN. Lors de l'apprentissage (modification des poids des connexions), on cherche à minimiser une erreur, sur la base d'apprentissage, entre la sortie fournie et la sortie désirée. On espère ainsi minimiser une erreur sur un ensemble de généralisation, c'est à dire un ensemble de données inconnues du réseau. On peut arrêter l'apprentissage :

- quand l'erreur entre la sortie du réseau et la sortie désirée ne diminue plus ou diminue d'une façon non significative,
- après un certain nombre d'itérations fixé à priori,
- pour la classification lorsque le réseau classe correctement toutes les données.

Le problème de ces méthodes est le risque de sur-apprentissage ou d'apprentissage « par cœur », quị ne permet pas une bonne généralisation. Il est donc préférable de tester les capacités du réseau pendant la phase d'apprentissage et d'interrompre celle-ci lorsque l'erreur en généralisation se dégrade.

L'apprentissage ne donne aucune garantie de trouver le minimum global où le nombre d'exemples d'apprentissage joue un rôle fondamental dans l'existence des minima locaux (Figure 3.7.) :

- si le nombre d'exemples est suffisant (beaucoup d'exemples par rapport au nombre de paramètres), le problème des minima locaux ne se pose pratiquement pas. Il suffit, d'effectuer quelques apprentissages avec des initialisations différentes des paramètres ;
- si le nombre d'exemples est insuffisant non seulement des minima locaux apparaissent, mais, de surcroît, le minimum global de la fonction de coût ne correspond pas forcément aux valeurs des paramètres recherchées ; il est donc inutile dans ce cas de mettre en œuvre des algorithmes coûteux pour chercher le

minimum global.

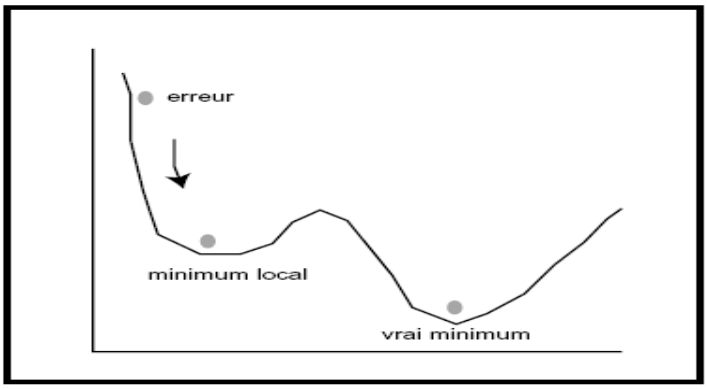


Figure 3.7. Minimum local et vrai minimum

Différentes techniques ont été mise au point pour résoudre ce problème :

Plus classiquement, l'ajout d'unités cachées, en élargissant la dimension du problème, permet généralement de réduire le nombre de mimima locaux, mais ceci n'est valide que dans certaines limites. En effet, au-delà d'un certain seuil, l'accroissement du nombre d'unités cachées engendre la formation de nouveaux minima locaux. Ensuite, l'utilisation d'un trop grand nombre d'unités cachées, en ajoutant des contraintes non pertinentes, entraîne un risque de surapprentissage (over-fitting), où on obtient une adéquation parfaite avec les données d'apprentissage, et une mauvaise adéquation avec la fonction génératrice que l'on cherche à apprendre, donc très mauvaise généralisation.

Il existe deux grandes familles d'apprentissage, l'apprentissage supervisé et le non supervisé.

3.4.4.1 Apprentissage supervisé

On parle d'apprentissage supervisé lorsqu'on dispose d'un ensemble de données dont on connaît la classe d'appartenance a priori (c'est le cas de notre problème). La base d'apprentissage est donc constituée de données et de la réponse que l'on désire. On calcul généralement une erreur entre la réponse du réseau et la réponse désirée. On modifie les paramètres du réseau de façon à réduire cette erreur.

Après l'apprentissage, le réseau est testé en lui donnant seulement les valeurs d'entrée, et en regardant si le résultat obtenu est proche du résultat désiré [27].

3.4.4.2 Apprentissage non supervisé

On parle d'apprentissage non supervisé lorsqu'on on dispose d'une base de données sans connaître leur classe d'appartenance. Il s'agit de découvrir la structure sous-jacente aux données sans imposer aucun modèle.

L'apprentissage non supervisé implique la fourniture à un réseau autonome d'une quantité suffisante d'exemples contenant des corrélations (redondance), telles que celui-ci en dégage des régularités automatiquement. Ces réseaux sont souvent appelés « auto-organisateur », ou encore « à apprentissage compétitif » [23].

3.4.5 Quelques réseaux célèbres

Il y a de très nombreuses sortes de RdN actuellement. Nous somme intéressé dans le cadre de ce travail par le perceptron et le réseau multicouche.

3.4.5.1 Le perceptron

C'est un des premiers réseaux de neurones, conçu en 1958 par Rosenblatt. Il est linéaire et monocouche. Il est inspiré du système visuel. La première couche (d'entrée) représente la rétine. Les neurones de la couche suivante sont les cellules d'association, et la couche finale les cellules de décision (figure 3.8.). Les sorties des neurones ne peuvent prendre que deux états (-1 et 1 ou 0 et 1).

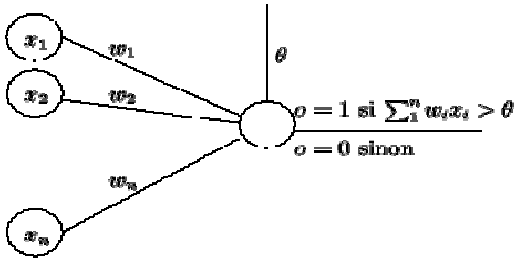


Figure 3.8. Le perceptron avec seuil

Si la sortie du réseau est égale à la sortie désirée, le poids de la connexion n'est pas modifié. Dans le cas contraire le poids est modifié[27].

a) La règle d'apprentissage du Perceptron

L'apprentissage consiste à déterminer les poids w_i pour que les exemples soient bien classés. L'algorithme commence avec des poids pris au hasard et on soumet les exemples en entrée du perceptron. On modifie à chaque fois les poids pour que le perceptron classe bien cet exemple :

1. Initialisation des poids et du seuil θ à des valeurs (petites) choisies au hasard.
2. Présentation d'une entrée, e = (x_1, ..., x_n) de la base d'apprentissage.
3. Calcul de la sortie obtenue S pour cette entrée :

$$S = \sum(w_i \, . \, x_i) - \theta \qquad \textbf{(3-3)}$$

o = signe (S) (si S > 0 alors o = +1 sinon S≤0 alors o = -1), θ : seuil.

4. Si la sortie o du Perceptron est différente de la sortie désirée d pour cet exemple d'entrée E alors modification des poids :

$$w_{ij}(t+1) = w_{ij}(t) + \eta.(d_j - o_j).e_i \qquad \textbf{(3-4)}$$

où : $d_j = +1$ si **e** est de la classe 1, $d_j = -1$ si **e** est de la classe 2 et $(d_j - x_j)$ est une estimation de l'erreur.

$W_{ij}(t)$: poids de la connexion entre les neurones i et j à l'instant t.

η : pas (taux,vitesse) d'apprentissage. Situé entre 0 et 1, on fait le souvent varier au cours de l'apprentissage. Partant d'une valeur élevée (typiquement environ 0,8), on le réduit progressivement quand on approche de la solution.

d_j : sortie désire pour l'entrée courante.

o_j : sortie obtenue pour l'entrée courante.

e_i : signal transmis par le neurone i.

5. Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement (i.e. modification des poids), retour à l'étape 2.

Si l'ensemble d'apprentissage est linéairement séparable, l'algorithme du perceptron trouve l'hyperplan séparateur en un nombre fini d'itérations.

b) Evaluation du perceptron

Malgré la simplicité apparente de ce système, il fut possible de montrer son efficacité dans certaines tâches simples [23], En 1969, Papert et Minsky ont démontré les limites du perceptron classique, L'un des principaux rencontrés est que le perceptron ne peut pas résoudre des problèmes de classification non linéaires comme par exemple la fonction logique XOR (ou exclusif).

De nouvelles formes d'apprentissage ont permis d'élargir les capacités du perceptron (comme le recodage des entrées ou les réseaux à couches caché), ouvrant un champ nouveau aux réseaux *feedforward* [14].

3.4.5.2 Les perceptrons multicouches

Les perceptrons multicouches (PMC) sont Apparus en 1985, aujourd'hui les modèles les plus employés. Plusieurs couches de traitement leurs permettent de réaliser des associations non linéaires entre l'entrée et la sortie [28].

Ce type de réseau est dans la famille générale des réseaux à «propagation vers l'avant», c'est-à-dire qu'en mode normal d'utilisation, l'information se propage dans un sens unique, des entrées vers les sorties sans aucune rétroaction. Son apprentissage

est de type supervisé, par correction des erreurs. Dans ce cas uniquement, le signal d'erreur est «rétropropagé» vers les entrées pour mettre à jour les poids des neurones. Le PMC est un des RdN les plus utilisés pour des problèmes d'approximation, de classification et de prédiction [19].

a) Architecture d'un réseau multicouche

L'architecture du PMC se compose d'au moins trois couches de neurones. Chaque neurone de la première couche est relié à la suivante par une connectivité totale et ce, jusqu'à la couche de sortie (figure 3.9.). La première couche est celle d'entrée. La dernière est la couche de sortie, tandis que les couches intermédiaires sont des couches cachées. La fonction d'activation des neurones des couches cachées et des neurones de la couche de sortie est la fonction sigmoïde. Cette différence par rapport au perceptron, est essentielle à l'entraînement du PMC et permet le traitement de problèmes non linéaires.

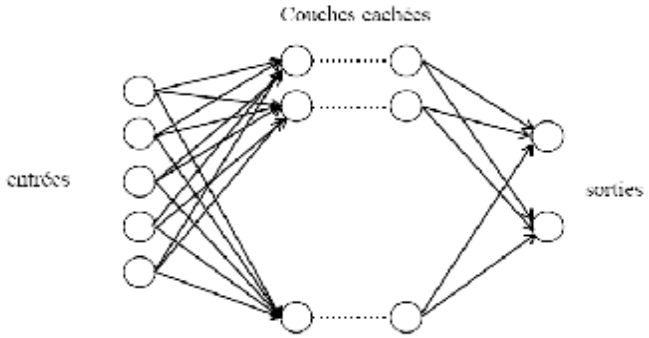


Figure 3.9. Le perceptron multicouche

b) Algorithme d'apprentissage de «Backpropagation » (rétropropagation)

La r*étropropagation* 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) [26].

L'algorithme de la rétropropagation de gradient, bien que très simple à implanter, nécessite un certain savoir-faire pour une utilisation efficace. En effet, la convergence de l'algorithme n'est pas prouvée et de multiples variables sont à ajuster précisément en fonction du problème traité. Parmi ces variables à fixer, les paramètres apparaissant dans les différentes équations, la sélection des exemples pour l'apprentissage et le test, la structure du réseau (nombre de couches, taille de la couche cachée), la configuration initiale des poids, le nombre d'itérations

d'apprentissage [27].

Le résumé de l'algorithme de rétro-propagation est ainsi défini:

$x= (x_1,x_2,\ldots\ldots\ldots\ldots,x_n)$, vecteur d'entrée

$o=(o_1,o_2,\ldots\ldots\ldots\ldots,o_m)$, vecteur de sortie désiré

$y=(y_1,y_2,\ldots\ldots\ldots\ldots,y_m)$ vecteur de sortie obtenu (réel)

1. Appliquer un vecteur d'entrée aux nœuds d'entrées puis initialiser les poids du réseau ;
2. Exécuter l'échantillon d'apprentissage à travers le réseau ;
3. Calculer les termes d'erreur de signal de la couche de sortie et les couches cachées en utilisant respectivement les équations suivante :

$$es = (o \quad x) f'(y) \qquad \textbf{(3-5)}$$

$$ec = f'(y) \sum_1^m es.w \qquad \textbf{(3-6)}$$

es : erreur de la couche de sortie, ec : erreur de la couche cachée

x : vecteur d'entré de la couche de sortie (signal de la couche cachée)

f : fonction sigmoïde, tel que : $f(x) - \frac{1}{1+e^{-x}}$

4. Mise à jour les poids de la couche de sortie et couches cachées en utilisant respectivement les équations suivantes :

$$w(t+1) = w(t) + \eta.es.x \qquad \textbf{(3-7)}$$

$$w(t+1) = w(t) + \eta.ec.x \qquad \textbf{(3-8)}$$

5. Répéter ce processus jusqu'à ce que l'erreur E devienne acceptable (aller à **2**)

$$E = \frac{1}{2} \cdot \sum_{k=1}^{m} (o_k - x_k)^2$$

$$= \frac{1}{2} \cdot \sum_{k=1}^{m} (o_k - f(y_k))^2$$

$$= \frac{1}{2} \cdot \sum_{k=1}^{m} \left(o_k - f\left(\sum_{j=0}^{n} w_j x_j\right)\right)^2 \qquad \textbf{(3-9)}$$

c) Momentum

La convergence du réseau par rétro-propagation est un problème crucial qui requiert de nombreuses itérations. Pour pallier à ce problème, un paramètre est souvent rajouté pour accélérer la convergence. Ce paramètre est appelé « le

momentum ». c'est un terme d'inertie dont le rôle est de filtrer les oscillations dans la trajectoire de la descente du gradient .donc Le momentum est un moyen efficace pour accélérer l'apprentissage et aussi pour pouvoir sortir des minimums locaux [19].

La règle de mise à jour des poids devient alors :

$$w(t+1) = w(t) + \eta.e.x + \alpha\big(w(t) - w(t-1)\big) \qquad (3-10)$$

Où 0 <= α <1 s'appelle le momentum.,e : l'erreur,

Le terme du momentum produit deux effets distincts selon la situation. Premièrement, lorsque la trajectoire du gradient a tendance à osciller, il contribue à la stabiliser en ralentissant les changements de direction. Par exemple, avec α = 0.8, cela correspond d'emblée à ajouter 80% du changement précédent au changement courant. Deuxièmement, lorsque le gradient courant pointe dans la même direction que le gradient précédent, le terme d'inertie contribue à augmenter l'ampleur du pas dans

d) Evaluation du réseau multicouche

Le succès de la rétropropagation témoigne de son efficacité. Mais l'algorithme le plus utilisé, il n'en souffre pas moins d'importantes limitations [14]:

- complexité algorithmique: l'algorithme de rétropropagation est connu pour sa lourdeur et la lenteur de sa convergence dès lors que le problème devient un peu conséquent.
- manque de bases analytiques: on ne dispose par exemple d'aucun outil fiable de dimensionnement du réseau plus particulièrement en ce qui concerne le nombre et la taille des couches cachées.
- Minima locaux, nombre d'unités cachées et surapprentissage [14]

Si le réseau est bien conçu, les RdN offrent des performances équivalentes à celles des meilleurs classifieurs en termes de taux de reconnaissance, mais ils se distinguent de ceux-ci par une plus grande facilité d'implantation sous forme de circuits spécialisés très rapides. Néanmoins, il faut souligner que l'on doit bien se garder d'utiliser systématiquement les RdN pour tout problème de classification. Il faut d'abord évaluer la difficulté du problème à traiter, et n'utiliser les RdN que lorsque c'est réellement nécessaire [26][15][16].

3.5 Conclusion

Lorsque l'on cherche à résoudre un problème de classification, on a affaire à des formes susceptibles d'appartenir à des catégories, ou classes différentes. Un classifieur

est capable d'attribuer une classe à une forme inconnue qui lui est présentée.

Nous avons présenté dans ce chapitre les classifcateurs qui sont très souvent employés dans la reconnaissance des caractères, a savoir, le classifieur KPPV, KPPV flou, et les RdN.

Les performances d'un classifieur dépendent essentiellement de la représentation des formes utilisées, c'est-à-dire du prétraitement effectué sur les données brutes, dans notre cas les chiffres d’un code à barres. Ils font l’objet du chapitre suivant.

CHAPITRE 4

CODE A BARRES

CHAPITRE 4

CODE A BARRES

L'identification automatique est un ensemble de techniques, comprenant le code-barres, la reconnaissance optique de caractères, la reconnaissance de formes "vision", l'entrée vocale, les étiquettes radio, les cartes magnétiques, etc.

*Dans ce chapitre nous présentons en général les codes à barres. Une présentation détaillée de code le plus répandu dans le monde, le code à barre EAN (**E**uropean **A**rticle **N**umbering) est ainsi faits.*

4.1 Introduction

L'utilisation des codes barres s'est développée de façon considérable au cours des dernières années. Depuis l'adoption de la norme UPC (**U**niversal **P**roduct **C**ode) par le commerce de détail à la fin des années 70. Les codes barres font parti de notre quotidien, il est évidemment un outil d'usage universel. Utilisé partout, de la chaîne de production jusqu'à la vente, il permet le suivi et l'identification de tous les produits courants ou non.

Cependant, Ce chapitre est consacré à les présentés d'une manière générale, et d'une manière détaillé les codes à barres de type EAN.

4.2 Code à barre

Les codes barres offrent une méthode rapide, facile et précise pour saisir des données. Grace à ce codage tous types de caractères peuvent être codés

4.2.1 Définition du code barre

Un code à barres (ou symbole) est la représentation graphique d'un code numérique ou alphanumérique, permettant sa lecture automatique par un matériel approprié. C'est une série de lignes verticales de largeur variable (appelées barres) et d'espaces. L'ensemble des barres et des espaces est appelé "éléments". Il existe différentes combinaisons de barres et d'espaces représentant différents caractères.

4.2.2 Avantages du code à barres

La lecture automatique autorise l'interconnexion des systèmes d'information et leur mise à jour en temps réel. Par exemple, les codes barres que l'on trouve sur les produits alimentaires ne contiennent pas le prix ou la description de l'article. Au lieu de cela, le code barres comporte un « code produit ». Lorsque le code est lu par un lecteur de code barres puis transmis à l'ordinateur, celui-ci recherche le fichier enregistré sur le disque et associé à ce code produit. Dans le fichier figure le prix, le nom du distributeur, la quantité disponible, la description du produit, etc. L'ordinateur effectue une lecture de prix en lisant le code barres puis il crée un registre des articles et additionne le prix au sous-total des produits achetés (il soustrait également la quantité du stock disponible).

Dans la grande distribution, le code à barres permet entre autres :

- d'entrer rapidement un produit en stock,
- d'en connaître l'origine,
- d'en faciliter le réapprovisionnement,
- d'automatiser la sortie des stocks et d'obtenir un inventaire simplifié permanent,
- d'optimiser les temps de traitement aux caisses.
- etc…

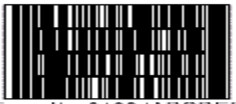
Dans une usine de fabrication, le code à barres permet également :

- de connaître l'origine des matières premières,
- d'avoir en continu des informations actualisées sur l'état de la production,
- d'aiguiller automatiquement un produit en fonction de ses caractéristiques, de ses défauts,
- d'identifier sa production,
- d'identifier les unités logistiques transmises à ses clients.

4.2.3 Les différents types de codes à barres

Il existe une multitude de types de code barre, sur une ou deux dimensions, pour le codage numérique ou alphanumérique (table 4.1.). Certains sont uniquement numériques (UPC, EAN, 2 parmi 5 entrelacé). D'autres ont une longueur fixe (l'UPC-A est formé de 12 chiffres, l'UPC-E est formé de 6 chiffres, l'EAN-13 est formé de 13 chiffres et l'EAN-8 est formé de 8 chiffres). Certains codes barres sont formés de caractères alphanumériques (Code 93, Code 128 et Code 39). Il existe un code barres permettant d'encoder les 128 caractères ASCII (Code 128) [29].

Table 4.1. Différents types de code à barres

Codes linéaires	
Code 3 de 9 (code 39)	
Code EAN / UPC	
Code 128 (EAN 128)	
Code 2 de 5 entrelacé	
Codabar /Monarch	
Code 11	
Codes Bidimensionnels	
Code PDF417 (Bidimensionnel / empilé)	 (Encodé : 01234ABCDEF)
Code Datamatrix	 (Encodé : 01234ABCDEF)
Codes empilés	
Code 16 K	
Codablock	 (Encodé : 01234ABCDEF)

4.3 Principales Applications

La liste ci-dessous donne une idée, non-exhaustive, des principales applications des codes barres :

- ***Suivi de production :*** Les lots de fabrication, les ordres de fabrications, ou les matériels en cours de production eux mêmes, sont identifiés par un code barre. La lecture de ce code, associée au poste de travail, heure, quantité produite, identification de l'ouvrier, etc. permet de connaître la situation à chaque instant, ainsi que les temps de production, consommations de matières premières, etc.
- ***Gestion de stocks :*** Les produits identifiés sont saisis lors de l'entrée et de la sortie du stock. De plus, leur position physique peut être connue en associant à l'identification produits un repérage par code-barres de l'adresse magasin.
- ***Saisie des règlements fournisseurs :*** Les livraisons fournisseurs sont identifiées par code barre à leur arrivée et peuvent ainsi être suivies lors de leur contrôle et transmission aux services comptables. La même identification servira à identifier les règlements.
- ***Analyse médicale :*** Les échantillons sont identifiés par code-barres dès leur prélèvement ou leur arrivée dans le laboratoire. Cette identification est lue par les matériels d'analyse automatique et les opérateurs.
- ***Tri automatique :*** Les colis ou objets à trier sont repérés par un code-barres lisible au vol et à distance. L'information lue au passage de l'objet sur un convoyeur sert à un automate programmable pour éjecter le colis sur la bonne sortie du convoyeur.
- ***Gestion de bibliothèque/discothèque/bandothèque ou magasin outil :*** Le code-barres identifie aussi bien les livres, bandes, etc. arrivant ou sortant que le badge d'identification de l'emprunteur. Il est donc possible de connaître à chaque moment qui détient quoi et depuis quand, ainsi que de connaître les références les plus demandées et les habitudes de chaque emprunteur.
- ***Gestion de cantine et restaurant d'entreprise :*** Le badge code-barres, économique et indestructible permet d'identifier les personnes et éventuellement de saisir rapidement les plats consommés sur un « menu » code-barres [30].

4.3.1 Utilisations typiques des certains codes à barres

Pour simplifier le choix d'une symbologie, il faut tenir compte de leurs cas d'emploi les plus fréquents :

- ***Code 39 :*** Toutes applications industrielles et autres, demandant un code alphanumérique et ne posant pas de gros problème de place disponible.

- ***Code 128 ou 93 :*** Toutes applications industrielles et autres demandant un code alphanumérique "full alpha" et/ou une grande densité. La meilleure densité en numérique est obtenue en code 128, la meilleure densité en alphanumérique est obtenue en code 93. Le Code 128 constitue souvent un très bon choix, à la fois par ses bonnes caractéristiques et par la diffusion importante qu'il commence à rencontrer, ce qui assure à l'utilisateur de trouver un plus large choix de matériels compatibles que le Code 93.
- ***Code 2 Parmi 5 Entrelace :*** Toute application industrielle et autres se suffisant d'un code numérique, en longueur fixe ou avec clé de contrôle.
- ***Code Monarch (Codabar) :*** Analyse médicale, transfusion sanguine, photographie.
- ***Code EAN (Gencod) :*** Grande distribution (Supermarchés, etc.), et Presse.
- ***Code 16 k ou 49 :*** Toute application industrielle ou autre demandant une très grande densité d'informations, où la lecture peut se faire exclusivement par laser, et ou une lecture omnidirectionnelle n'est pas nécessaire.

Le code 16K est plus simple à imprimer et à décoder (encodage similaire au code 128), ce qui assure l'utilisateur de trouver un choix vaste de lecteurs et d'imprimantes. Le code 49, plus performant en termes de fiabilité et de densité, demande des décodeurs et des imprimantes munis de microprocesseurs puissants et disposant d'une importante mémoire [30].

4.3.2 Niveau de fiabilité

Une symbologie code barres est d'autant plus fiable que la probabilité d'une erreur de substitution est faible. En effet, en matière de codes barres, l'important est d'être sur qu'une lecture faite est exacte. Si à la première tentative de lecture, aucune lecture ne peut être faite, il est en général possible de réessayer, voire, dans le cas des lecteurs lasers et des caméras, de faire plusieurs centaines de tentatives de lecture par seconde. Par contre, lorsqu'une lecture a eu lieu, on doit pouvoir être sûr que cette lecture est exacte. Cette caractéristique est l'un des avantages déterminant de la technique des codes barres. Si on peut considérer que tous les codes barres sont largement plus fiables que n'importe quelle entrée clavier, néanmoins tous les codes barres n'ont pas la même fiabilité [30].

Les principaux codes barres peuvent être classés par ordre de fiabilité croissante :

- EAN 13
- 2 parmi 5 entrelacé
- Code 128
- Code 93
- Code Monarch
- Code 39
- Code 16 K
- Code 49

Ce classement est fait en tenant compte, pour chaque symbologie, de son mode d'utilisation le plus usuel, c'est-à-dire :

- EAN 13: clé de contrôle obligatoire.
- 2 parmi 5 entrelacé: utilisation soit en longueur fixe, soit avec clé de contrôle. La fiabilité de ce code peut être améliorée en entourant le symbole d'un cadre noir, et/ou en utilisant à la fois la longueur fixe et la clé de contrôle. Il est tout à fait déconseillé d'utiliser le code 2 parmi 5 entrelacé en longueur variable sans clé de contrôle.
- Codes 128, 93, 16K, 49 : utilisés également avec une clé de contrôle.
- Code 39 sans clé de contrôle. Une clé de contrôle améliore bien entendu encore la fiabilité.

4.4 Les lecteurs de code à barres

Les codes-barres sont destinés à une lecture automatisée par un capteur électronique « le lecteur de code-barres » (voir annexe C). Il existe trois types de lecteurs de codes barres, les lecteurs fixes, les lecteurs portables à transmission par lots et ceux à fréquence radio [29].

Les lecteurs fixes restent attachés à leur ordinateur hôte ou à leur terminal et ils transmettent un article de données à la fois pendant la lecture des données [29].

Les lecteurs portables fonctionnent avec des piles et stockent les données en mémoire pour les transmettre ultérieurement par lots à l'ordinateur hôte. Des lecteurs portables plus perfectionnés permettent d'opérer également en mode non-portable, ce qui évite la nécessité d'avoir un lecteur fixe séparé [29].

Les lecteurs portables à fréquence radio (RF) fonctionnent avec des piles et transmettent les données en temps réel. Plus important, l'hôte peut communiquer à

l'opérateur des instructions lui indiquant ce qu'il doit faire dans le contexte (en fonction de la progression du travail).

Pour l'essentiel, un lecteur de code barres est formé d'un décodeur et d'un scanner (un câble est également requis pour connecter le décodeur à l'ordinateur ou au terminal). La fonction de base d'un scanner est de lire un code barres et de produire un signal électrique [29].

La première fonction est généralement assurée par le capteur optoélectronique, associé à un module logiciel au niveau du décodeur; Les autres fonctions sont réalisées par le décodeur. L'organe de saisie est un capteur optoélectronique dont le rôle est de convertir les séquences de blanc et de noirs en signaux électroniques par balayage du code. Le balayage peut être, Manuel (crayon lecteur), Automatique à partir d'un faisceau LASER, Automatique par déplacement du code devant le lecteur ou Electronique par analyse des signaux lumineux reçus par une barrette CCD.

4.4.1 Le crayon lecteur

C'est le plus simple des appareils de lecture. Le crayon effectue une lecture par passage manuel sur le code à barres. Réalisée au contact, la qualité de la lecture varie selon l'inclinaison du crayon et dépend énormément de la constance avec laquelle le code est parcouru.

Son petit prix en fait un modèle adapté aux faibles besoins de lecture, éventuellement un outil de contrôle de l'information codée.

4.4.2 Le lecteur CCD

La douchette CCD (Charge-Coupled Device, ou détecteurs à couplage de charge) autorise une lecture automatique du code à barres, nul besoin de parcourir le code et chacune de ses barres. La distance de lecture est variable (jusqu'à 20 cm) et dépend des réglages apportés au lecteur. Plus la distance est importante et plus l'éclairage du code est faible.

Les modèles CCD ne comportent pas d'élément mécanique et présentent généralement une grande robustesse.

Plus économique que les modèles lasers, les lecteurs CCD sont capables de lire tous les codes à barres linéaires. Certains modèles sont capables de lire les codes empilés (PDF417).

4.4.3 Le lecteur laser

Construit autour d'un jeu de miroirs, le lecteur laser utilise un seul rayon lumineux généré par une diode laser. La source lumineuse est dense et précise, et autorise une lecture rapprochée ou distante de plusieurs mètres ainsi qu'une lecture au vol, sur des objets ou documents en mouvement. Tout comme les lecteurs CCD, le lecteur laser réalise une lecture automatique du code. Nul besoin de parcourir le code dans sa longueur, un miroir motorisé le fait en réfléchissant le rayon laser de part et d'autre du code donnant ainsi l'illusion optique d'un trait continu. Certains lecteurs réalisent également ce balayage sur la hauteur du code (Raster en Anglais ou multitrames), d'autres encore démultiplient le balayage dans le but d'obtenir une grille sous laquelle le code peut être placé sous un degré d'orientation quelconque (omnidirectionnel).

4.5 EAN (European Article Numbering)

Appelé aussi Gencode, c'est un code à barres qui peut être lu de façon omnidirectionnelle. Il est utilisé pour tous les biens qui sont lus par un scanner aux caisses des commerces de détail.

Ce code utilisé sur tous les articles de consommation courante, sans lequel la grande distribution ne serait peut-être pas aussi grande.

Le code UPC a été inventé dans les années 70 par George J. Laurer, ingénieur chez IBM et adopté en Mai 1973. George J. Laurer, après avoir obtenu son diplôme à l'Université du Maryland en 1951, travailla chez IBM. En 1969 il se vit attribué la tâche titanesque de créer un code et son symbole d'identification de tous les produits distribués en magasins par le Uniform Grocery Product Code Council [31].

Sa solution l'Universal Product Code changea radicalement le monde de la distribution. Il a ensuite amélioré le code en lui ajoutant un 13ème caractère créant ainsi le code **EAN [31]**.

4.5.1 EAN et UPC

L'EAN est composé de 8 ou 13 chiffres représentés sous forme de séquences de barres noires et blanches formant un code à barres.

Le code **EAN-13** utilisé dans le monde entier et le code **UPC-A** (**U**niversal **P**roduct **C**ode) utilisé aux USA sont quasiment identiques. La seule différence étant que sur un UPC seuls 12 chiffres apparaissent au lieu de 13 chiffres sur l'EAN.

Un UPC est un EAN dont les 2 premiers caractères à gauche du code sont des zéros. C'est en réalité un code EAN13 dont le premier chiffre serait zéro et dont la présentation serait légèrement différente. (Figure 4.1.).

Il existe plusieurs versions de codes UPC et les plus communes sont le code UPC à 12 caractères de type A et celui à 8 caractères de type E (code UPC-A est un sous-ensemble du code EAN13).

Figure 4.1. EAN et UPC

Ces deux codes barre sont identiques, on a rajouté un zéro devant le code UPC-A pour obtenir le code EAN13 mais le motif des barres est strictement identique [32].

4.5.2 Les éléments du code EAN

La figure 4.2. montre les indications que comportent le code EAN

Figure 4.2. Les éléments du code EAN

- **Le préfixe EAN**

Le préfixe contient 2 ou 3 chiffres. Il est le code du pays (code système) qui a délivré le numéro de participant. Pour l'Algérie ce code est le 613. La liste des codes pour les autres pays se trouve en annexe B.

Les préfixes 977, 978 et 979 indiquent qu'il s'agit d'un livre ou d'une publication comportant un numéro ISBN (international standard book number) ou ISSN (international standard serial number, en français 'numéro international normalisé des publications en série) [31][32].

- **Le numéro de participant (identificateur de société)**

Il est délivré par l'organisation EAN du pays concerné contient 4 ou 5 chiffres. En Algérie cette organisation est Algérie – EAN.

- **Le Numéro d'Article (identificateur d'article)**

Le numéro d'article du producteur de l'objet ainsi codé sur 5 chiffres

- **Le Chiffre de Contrôle**

Le dernier chiffre d'un code EAN 13 est toujours une clé de contrôle (*check digit*), calculée à partir des 7 (EAN-8) ou 12 (EAN-13) premiers chiffres qui composent le code (somme de produits modulo 10) :

- Soit **x**, la somme des chiffres pairs et **y** la somme des chiffres impairs.
- Calculons **z = 3*x +y**.
- Soit m le nombre divisible par 10 immédiatement supérieur à z, la somme de contrôle est **m - z** (d'une autre manière diviser le nombre obtenu par 10 et soustraire le reste du nombre 10. Le résultat est le Check Digit)

- **Exemples d'application**

  - Soit le code à barres 978020113447#, quel est le check digit?

x =9 + 8 + 2 +1+3 + 4 = 27

y = 7 + 0 + 0 + 1 + 4 + 7 = 19

z = 3 * 19 + 27 = 84

m = 90

Somme de contrôle = 90 - 84 = **6** donc EAN13 ---> 9 780201 13447**6**

  - Soit le code 7 6 1 2 3 4 5 6 7 8 9 1 quel est le check digit ?

D'une manière plus simple le code 761234567891x se décompose comme :

7_1_3_5_7_9 → somme : 32

_6_2_4_6_8_1 → somme : 27

3*27+32 = 113 dont le chiffre des unités est **3**.

10-**3**= 7 dont le chiffre des unités est **7**.

7 est donc la « clé » du code-barres EAN 13 : 7612345678917

4.5.3 Composition du code EAN

4.5.3.1 Le système d'encodage du code EAN 13

Figure 4.3. EAN 13

Chaque caractère (chiffre) d'un code EAN 13 est composé de 7 modules (largeur d'une barre fine 0.33 mm en standard est appelée module), qui composent 2 barres et 2 espaces. Aucune barre ou espace ne comporte plus de 4 modules. Les seules exceptions à cette règle sont les caractères de début et de fin (3 modules chacun) et le séparateur central (long de 5 éléments).

Les barres verticales noires et blanches utilisées sont caractérisées par une certaine largeur, qui est toujours multiple d'une certaine largeur élémentaire (nommée module).

Tous les caractères de la partie gauche du code EAN commencent toujours par un 0 (espace) alors que ceux de la partie droite commencent toujours par un 1 (barre).

La marque (le caractère) de début de même que la marque de fin est codé : « 101 », et la marque de séparation centrale placée au milieu après le 7 ème chiffre est : « 01010 ».

La longueur totale est toujours de 12 * 7 + 2 * 3 + 5 = 95 modules; il y a toujours 30 barres au total. Les barres des séparateurs sont plus longues de 5 modules que les autres barres [31][32].

Le second caractère du préfixe et les cinq caractères du numéro de participant et les cinq caractères du numéro d'article codés comme décrit ci-après. :

- Les caractères situés à gauche du Séparateur Central du symbole EAN 13 utilisent deux jeux de codification nommés Set A et Set B. Ceux situés à droite utilisent le jeu de codification nommé Set C.
- La première colonne de codage des chiffres des éléments A et B est toujours vide. Celle des chiffres des éléments C est toujours pleine.
- La dernière colonne de codage des chiffres des éléments A et B est toujours plein. Celle des chiffres des éléments C est toujours vide.

- Le premier chiffre du Préfixe n'est pas codé mais il détermine l'alternance des Set A et B à utiliser pour le codage des 6 caractères (chiffres) situés à gauche du séparateur central.

a) Table de codage des caractères EAN 13

La table 4.2. indique comment codifier chaque caractère d'un EAN 13 selon qu'il se trouve à gauche ou à droite du Séparateur Central.

Table 4.2. Table de codage de code EAN 13

chiffre	Partie gauche		Partie droite
	Set A (Odd Parity)	**Set B (Even Parity)**	**Set C**
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Observations

- Le codage des Set A et C est similaire, les 0 du Set A devenant des 1 en Set C.
- Le codage des Set B et C est symétrique
- Pour obtenir le Set B à partir du Set A, il faut :
  - Changer tous les 1 en 0 et tous les 0 en 1 du Set A.
  - Lire le résultat de droite à gauche. On obtient le Set B.

b) Table de parité du code EAN

La table 4.3. indique le Set avec lequel chaque caractère de la partie gauche de l'EAN doit être codé. Selon le premier chiffre du Préfixe, les chiffres de gauche utilisant le Set A ou B.

Table 4.3. Table de parité du code EAN

Chiffre 1	Set à utiliser					
	Chiffre 2	Caractères du Numéro de Participant				
		Chiffre 3	Chiffre 4	Chiffre 5	Chiffre 6	Chiffre 7
0 (UPC-A)	A	A	A	A	A	A
1	A	A	B	A	B	B
2	A	A	B	B	A	B
3	A	A	B	B	B	A
4	A	B	A	A	B	B
5	A	B	B	A	A	B
6	A	B	B	B	A	A
7	A	B	A	B	A	B
8	A	B	A	B	B	A
9	A	B	B	A	B	A

c) Exemple de composition

Cet exemple illustre la composition du code EAN-13 de la figure 4.3. qui représente la valeur: 7612345678900.

- Préfixe 76,
- Numéro de Participant 12345,
- Numéro d'Article 67890
- Check Digit 0.
- Le premier caractère du préfixe est 7.

En consultant la table de parité pour la valeur 7, on trouve que le second caractère du Préfixe et le Numéro de Participant utilisent les Sets A B A B A B.

Le Code EAN est ainsi composé en mettant simplement bout à bout les chaînes de zéro et de un, Soit:

10101011110110011001001101000010100011011100101010101000010001001001000111010011100101110010101

4.5.3.2 Le système d'encodage du code EAN 8

Il ressemble beaucoup au code EAN 13. Il comporte 7 chiffres et une somme de contrôle calculée exactement de la même manière que pour le code EAN13. Les délimiteurs gauche, central et droit sont les mêmes. Les 4 premiers chiffres sont

construits avec le Set A et les 4 derniers avec le Set C. (Le Set B n'est utilisé que pour l'EAN-13).

Figure 4.4. Code EAN 8

a) Structure du code EAN-8

Le caractère de début et fin codé 101. Les 2 ou 3 caractères du Préfixe codés comme décrit ci-après. Le premier ou deux premiers caractères de participant suivi d'un Séparateur Central codé 01010. ensuite les 2 ou 3 derniers caractères de la référence article. Le dernier chiffre est le Check Digit, codé comme décrit ci-après.

b) Table de codage des caractères EAN-8

Les caractères à gauche du séparateur central utilisant le Set A. Les caractères à droite du séparateur central utilisant le Set C (table 4.4.).

Table 4.4. Table de codage des caractères EAN-8

Digit	Partie gauche	Partie droite
	Set A (Odd Parity)	**Set C**
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

c) Exemple de composition

La composition du code EAN-8 qui représente la valeur: 76123450

- Préfixe 76,
- Référence Article 12345,
- Check Digit 0.

Le Code EAN-8 est ainsi composé en mettant simplement bout à bout les chaînes de zéro et de un. Soit :

10101110110101111001100100101101010100001010011101001110111001010 1

4.6 Conclusion

Dans ce chapitre nous avons présenté une étude détaillée des codes barres EAN, puisque c'est le code le plus utilisé pour les produits.

Le chapitre suivant ne constitue qu'une présentation de la démarche suivie pour construire une petite application qui à comme objectif de reconnaitre les chiffre d'un code à barres.

CHAPITRE 5

APPLICATION

CHAPITRE 5

APPLICATION

Dans le présent chapitre nous allons présenter notre application développée sous MATLAB, qui permet de reconnaitre les chiffres arabes imprimés, en passant par les différentes étapes d'un système de RdF depuis les prétraitements (localisation / extraction des formes à reconnaître) jusqu'à la reconnaissance (classement) par différentes classifieurs : K_PPV, K_PPV flou et un réseau de neurones utilisant l'algorithme de rétropropagation du gradient.

Les performances du système développé sont mises en exegre à travers une application sur les code à barres, afin de faire une lecture automatique.

5.1 Introduction

Dans ce chapitre notre objectif est de développer, sous MATLAB, un système complet de reconnaissance de chiffres arabes imprimés. La figure 5.1. présente le schéma fonctionnel du système.

Ce schéma, résume la conception d'un système de classification, qu'il soit un réseau de neurones artificiels ou une autre méthode de classification, qui nécessite au début l'acquisition des données, puis le prétraitement (localisation) et le choix des attributs caractérisant les données.

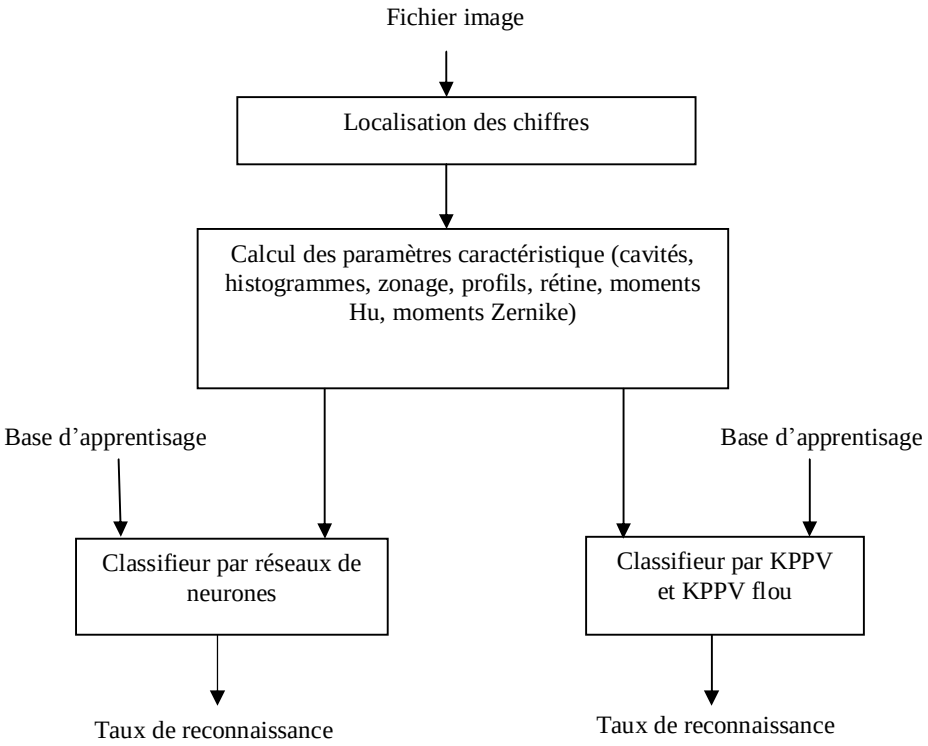


Figure 5.1. Les étapes de la reconnaissance

5.2 Acquisition des données (fichier image)

L'acquisition des données permet de convertir les données de leurs formes physiques (son, image, etc.) en une forme acceptable par l'ordinateur.

Dans l'étape d'acquisition, on doit acquérir une base de données représentant les différentes formes (différentes tailles et polices) des chiffres arabes imprimés. L'acquisition de cette base se fait par l'une des méthodes suivantes :

- Acquisition des chiffres par un logiciel graphique (Paintbrush) sous forme d'images binaires (c'est la méthode utilisé dans notre cas).
- Acquisition des chiffres à l'aide d'un Scanner (pour des applications en mode statique ou off-line).
- Acquisition des chiffres par une caméra vidéo (pour des applications en temps réel).

Nous avons utilisé la première méthode parce que les deux autres ont besoin beaucoup de prétraitement (peut être un sujet)

On dispose de 10 fichiers images binaire (voir annexe D), chaque image contenant 455 chiffres, la figure 5.2 présente le fichier image du chiffre 5. Une fois les chiffres extraits serviront de base d'apprentissage pour les classifieurs à développer.

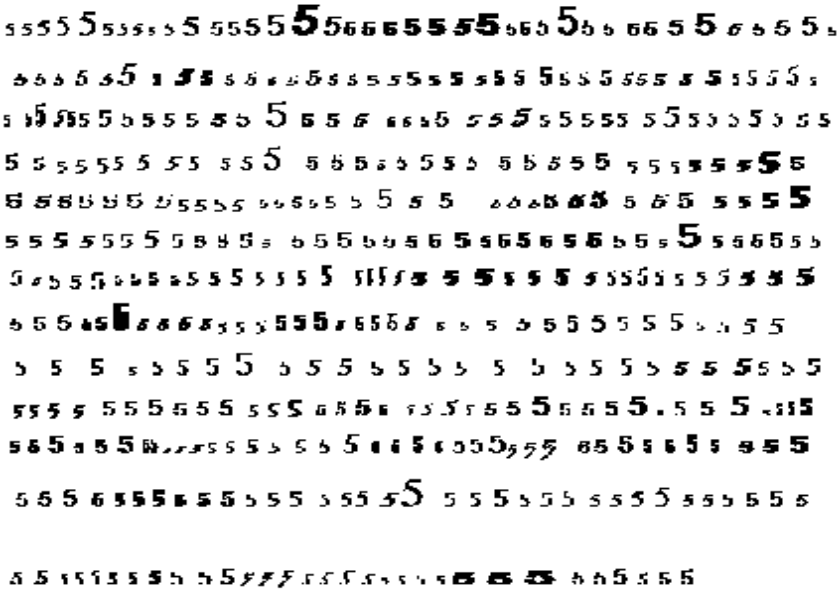


Figure 5.2 Echantillon de chiffre 5

Les performances du système développé sont testées sur des fichiers images différents, où chaque image contient 100 chiffres (voir annexe D).

5.3 Localisation :

Pour reconnaître une séquence de chiffres, on doit d'abord extraire chaque chiffre de la séquence pour le classifier ensuite.

Puisque les chiffres d'un code à barres ne sont jamais accolés les uns aux autres, l'extraction des chiffres de la séquence est relativement simple. La méthode consiste à détecter d'abord le début et la fin de chaque ligne de chiffres puis, pour chaque ligne détectée, à l'extraire et à détecter les colonnes de début et de fin de chaque chiffre.

Cela se réalise en comptant les pixels blancs dans le sens de lignes, puis, pour chaque ligne de chiffres, en comptant les pixels blancs dans les sens des colonnes. La comparaison du résultat du comptage avec un seuil permet de détecter les débuts et fins de chiffres, et les limités par des rectangles comme cela est illustré dans la figure 5.3., puis sauvegarder dans un fichier les coordonnées de ces rectangles englobant (coordonnées du coin haut gauche et du coin bas droit).

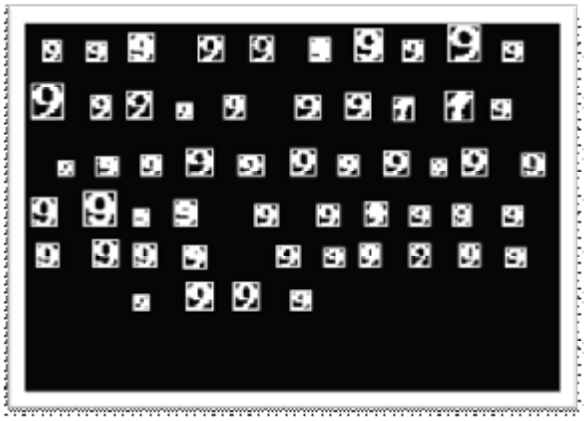


Figure 5.3. Image après localisation

5.4 Le choix des paramètres caractéristiques

Le choix d'un vecteur d'attributs pour caractériser un objet peut se révéler délicat. En effet, il faut faire un compromis entre la dimension du vecteur et le contenu des informations. Un vecteur de taille trop petite conduirait à de mauvaises performances du classifieur.

Afin de caractériser les chiffres, nous avons opté pour les primitives suivantes : les cavités, le zonage, les profils et les histogrammes de projection qui ne répondent pas au quatrième exigence (Indépendances de la rotation), les moments géométrique et les moments Zernike, qui ont l'objet du chapitre 2.

- **Cavités :** Nous allons utiliser comme paramètres caractéristiques les surfaces relatives des différents types de cavités par rapport à la surface globale du rectangle englobant le chiffre (normalisation): Ouest, Est, Sud, Nord et Central
- **Zonage :** Extraire les densités par zoning, nous avons choisi $n=m=2$ (4 zones, le chiffre est divisée en deux zones horizontale et chaque zone est divisé en deux autres zones verticales). Les densités dans chaque zone devront être normalisées en les divisant par la surface de la zone, puisque les chiffres ne sont pas tous de même taille.
- **Profils :** Extraire les profils gauche et droit pour obtenir un vecteur

caractéristique qui contient 6 profils gauche (2 au début, 2 au milieu, 2 a la fin) et 6 profils droit (2 au début, 2 au milieu, 2 a la fin). Les profils gauche et droit sont normalisés en les divisant par exemple par la largeur de l'image du chiffre traité puisque les chiffres ne sont pas tous de même taille.

- **Histogramme de projection (horizontale et verticale) :** Pour l'histogramme, après la localisation on obtient des images de taille différente. Donc le nombre d'attributs est différent d'un chiffre à l'autre. Pour dépasser ce problème nous avons normalisé l'image pour obtenir une taille unique pour tous les chiffres, par 'utilisation de la fonction 'imresize' de Matlab
- **Moment Hu :** Ceux sont 7 moments de Hu qui sont invariants en translation, échelle et rotation (chapitre 2)
- **Moment Zernike :** Nous prenons 47 moments Zernike parce que les deux premiers sont constants.
- **Rétine :** Nous prenons les pixels de la matrice (image) comme des entrées, Cette méthode est très utilisés dans certaines approches neuronales. Mais ne tolère absolument pas les transformation géométrique (rotation, translation, échelle).

5.5 Classification

Pour rendre le programme de reconnaissance plus fonctionnel, il faudrait également séparer la phase d'apprentissage de la phase de reconnaissance. En effet, surtout l'apprentissage pour un RdN peut durer plusieurs heures, donc on ne veut effectuer l'apprentissage qu'une seule fois. Il faut donc avoir la possibilité de sauver les poids dans un fichier lors de l'apprentissage et pouvoir les charger lors de la phase de reconnaissance.

5.4.1 Création des bases d'apprentissage et de test

L'apprentissage consiste à donner au classifieur plusieurs exemples de chiffres imprimées et à lui indiquer à chaque fois de quel type de chiffre il s'agit. Tous ces exemples sont placés dans ce que l'on appelle une base d'apprentissage.

Un exemple est formé d'un vecteur d'entrée dont les composantes sont les valeurs des paramètres caractéristiques et d'un vecteur de sortie qui code la classe de l'exemple. Pour le développement du programme, notre base d'apprentissage est formée de quatre matrices:

- Matrice contenant les vecteurs d'entrée.

- Matrice contenant les vecteurs de sortie.

- Matrice contenant les vecteurs positions (chaque vecteur contient quatre composantes qui déterminent le rectangle englobant le chiffre. Les deux cordonnées du point en haut à gauche suivies des deux coordonnées du point en bas à droite).

5.4.2 Classifieur K_ppv et K_ppv flou

Table 5.1. Taux de reconnaissance par KPPV

Attributs	K	Taux de reconnaissance
Cavités	1	82,60 %
	6	79,20 %
	11	76,40 %
	16	76,20 %
	21	74,80 %
Zonage	1	83,30 %
	6	75,40 %
	11	72,90 %
	16	71,10 %
	21	71,30 %
	27	71,60 %
Profil	1	93,60 %
	6	93,40 %
	11	92,20 %
	16	90,90 %
	21	89,90 %
Histogramme	1	97,40 %
	5	95,40 %
	6	95,20 %
	11	94,20 %
	16	92,80 %
	21	92,00 %
Rétine	1	99,10 %
	6	98,40 %
	11	98,10 %
	16	97,30 %
	21	96,10 %
Moments Hu	1	75,10 %
	6	61,50 %
	11	59,60 %
	16	56,70 %
	21	55,00 %
	26	55,10 %
Moments Zernike	1	92,90 %
	48	78,10%
	95	74,10 %
	142	69,70 %
	189	66,70 %

Selon Les résultats obtenus dans la table 5.1., nous voyons que les meilleurs taux de reconnaissance sont toujours pour K=1, quelque soit les paramètres caractéristiques. La table 5.1. montre aussi que les meilleurs résultats sont ceux de la rétine (99.10 %), mais puisque ce paramètre est sensible au rotation; on peut dire que les meilleurs sont les résultats des moments Zernike (92.90 %).

Table 5.2. Taux de reconnaissance par KPPV flou

Attributs	K	KPPV	KPPV flou		
			M=1	M=2	M=3
Cavités	3	79,40 %	64,70 %	64,10 %	59,10 %
	8	78,90 %	74,70 %	73,10 %	54,40 %
	13	75,50 %	78,70%	75,30 %	49,10 %
	18	75,50 %	77,40 %	73,30 %	39,10 %
	23	74,60 %	78,10 %	71,30 %	34,70 %
Zonage	3	78,80 %	62,70 %	60,40 %	54,40 %
	8	75,30 %	76,20 %	72,00 %	45,80 %
	13	71,50 %	75,60 %	71,60 %	34,50 %
	18	70,90 %	75,60 %	69,70 %	28,00 %
	23	71,40 %	75,30 %	68,70 %	24,70 %
Profil	3	92,50 %	78,10 %	77,00 %	75,30 %
	8	93,20 %	88,50 %	86,10 %	76,80 %
	13	90,90 %	91,10 %	88,00 %	71,00 %
	18	90,90 %	91,30 %	87,20 %	65,50 %
	23	90,00 %	90,80 %	86,60 %	62,00 %
Histogramme	3	96,30 %	91,40 %	89,80 %	85,20 %
	8	94,70 %	95,40 %	90,40 %	74,70 %
	13	93,90 %	94,10 %	88,60 %	66,70 %
	18	92,40 %	93,10 %	86,50 %	59,40 %
	23	91,50 %	93,10 %	85,00 %	50,60 %
Rétine	3	98,80 %	94,10 %	93,50 %	91,80 %
	8	98,10 %	98,40 %	95,10 %	88,70 %
	13	97,70 %	98,50 %	93,90 %	82,80 %
	18	97,00 %	98,30 %	92,60 %	78,60 %
	23	96,70 %	97,70 %	92 00 %	75,30 %
Moments Hu	3	70,00 %	45,50 %	46,20 %	39,30 %
	8	60,20 %	62,10 %	62,50 %	29,40 %
	13	59,10 %	66,30 %	65,00 %	20,00 %
	18	55,90 %	67,40 %	65,70 %	13,40 %
	23	56,00 %	69,40 %	66,40 %	09,70 %
Moments Zernike	3	91,80 %	83,90 %	83,50 %	78,90 %
	8	89,90 %	90,90 %	84,50 %	53,70 %
	13	88,00 %	90,60 %	80,80 %	39,70 %
	18	83,80 %	89,10 %	78,70 %	31,30 %
	23	80,60 %	87,10 %	77,00 %	25,70 %

D'après la table 5.2. nous constatons que les meilleurs résultats de la méthode KPPV floue concernent la valeur de fuzzification m=1, et même par rapport à la méthode KPPV quand la valeur de K ni trop petite ni grande.

Pour rendre la comparaison facile entre les deux méthodes KPPV et KPPV floue, nous présentons les résultats sous des graphes dans la figure 5.4. jusqu'à la figure 5.10

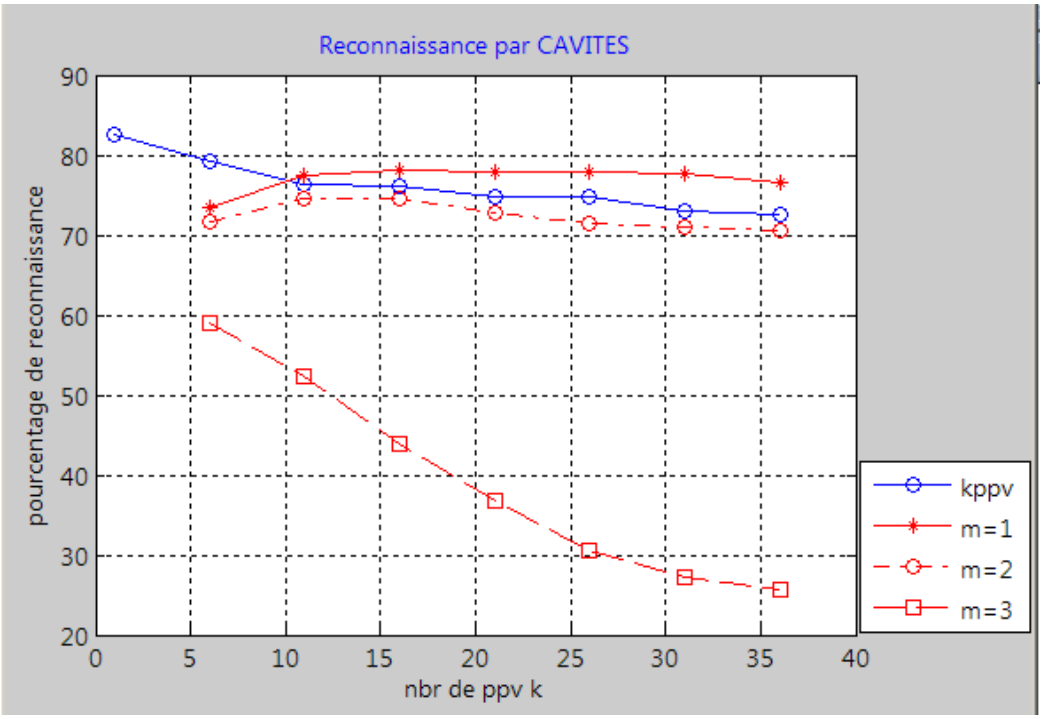


Figure 5.4. Le tau de reconnaissance par Cavités

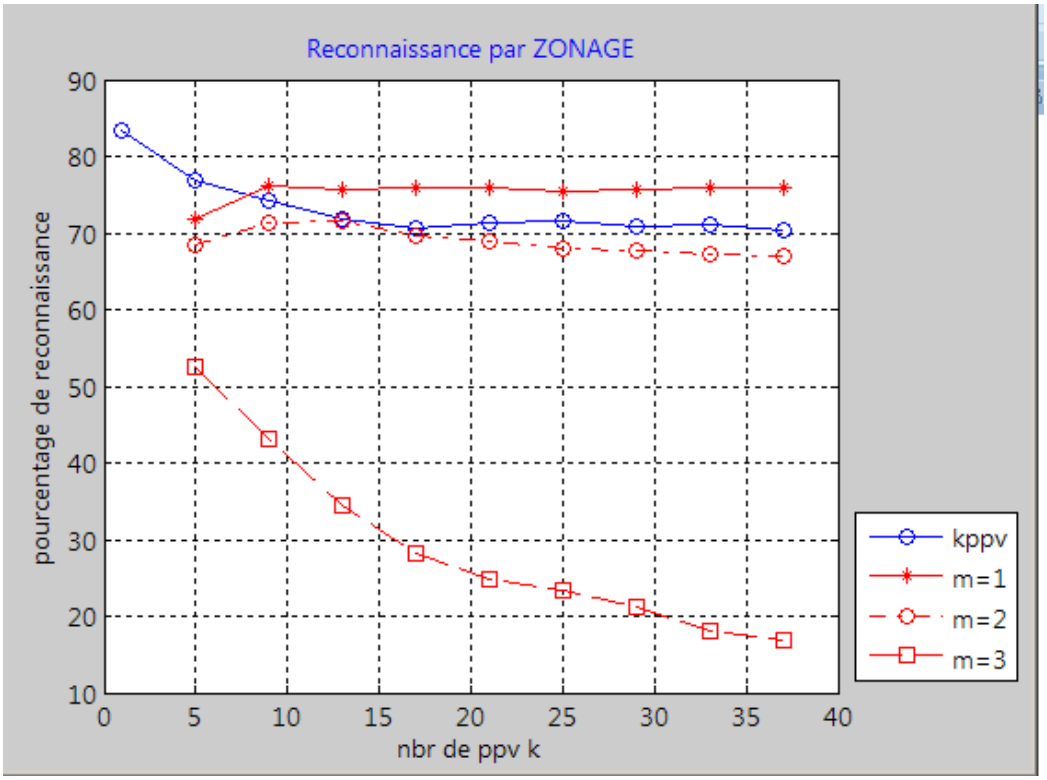


Figure 5.5. Le tau de reconnaissance par Zonage

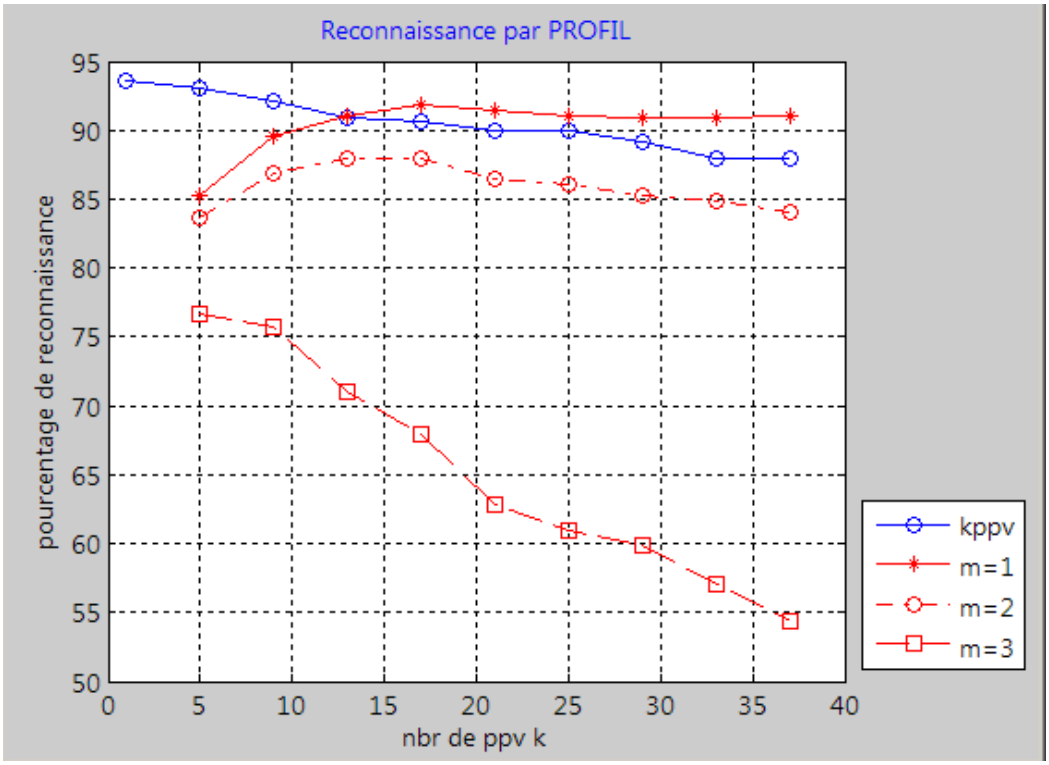


Figure 5.6. Le tau de reconnaissance par Profils

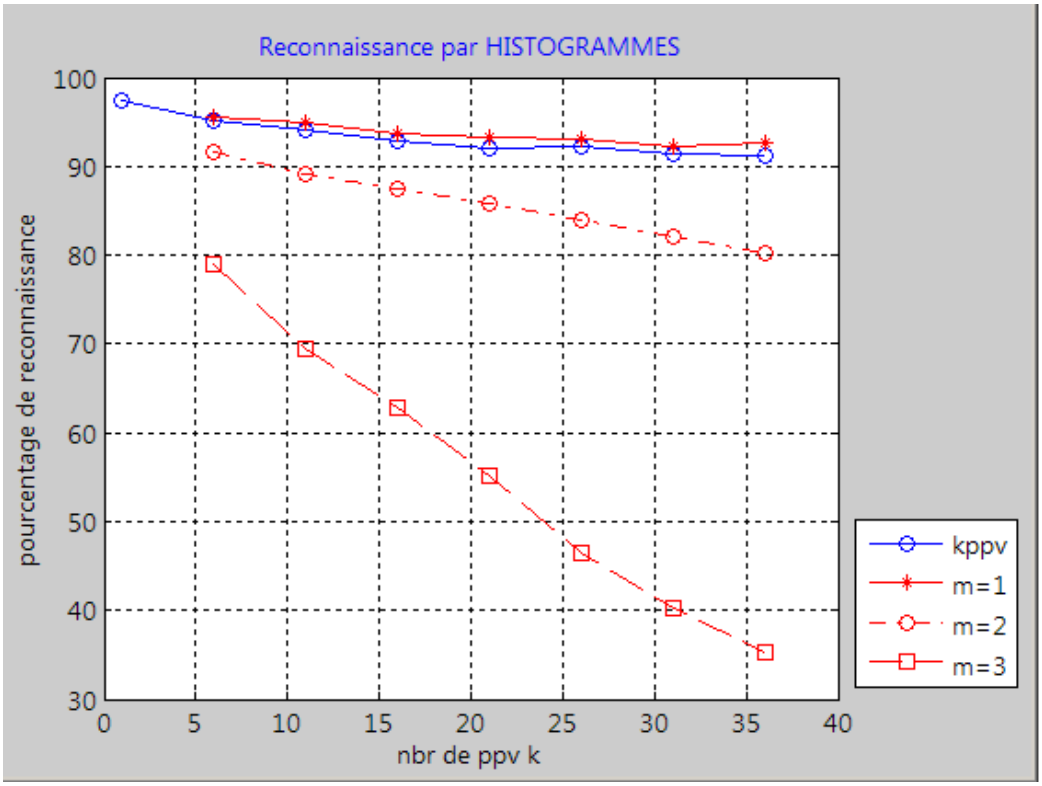


Figure 5.7. Le tau de reconnaissance par Histogrammes

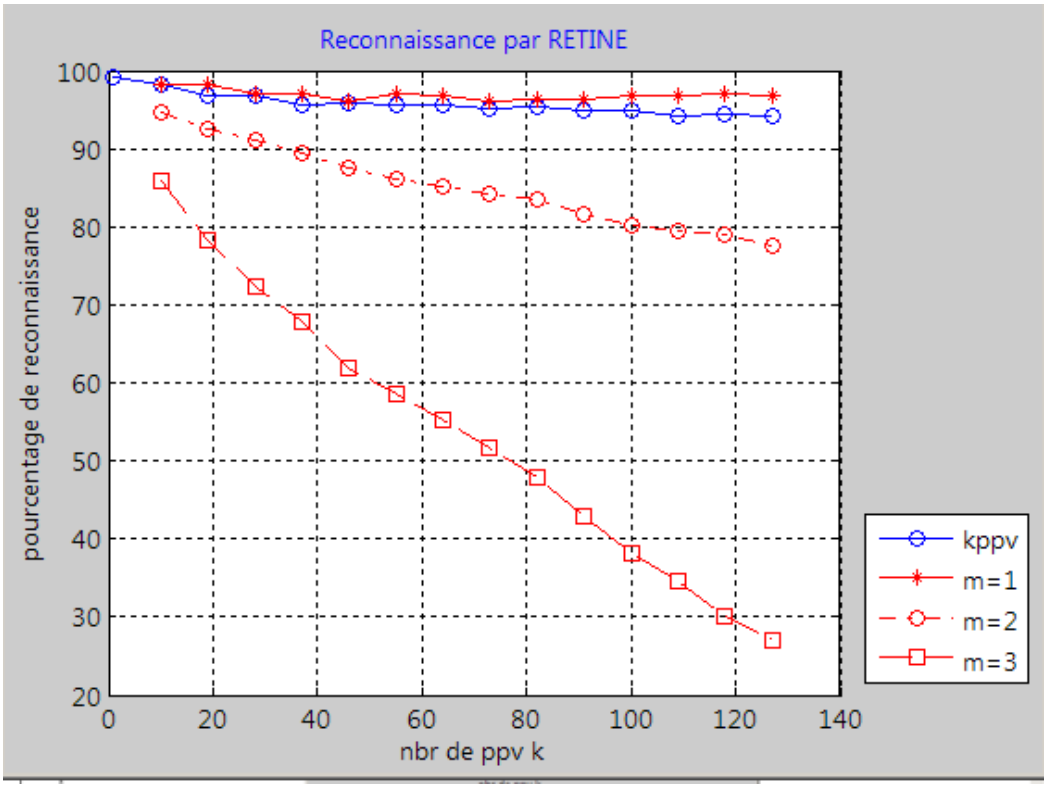


Figure 5.8. Le tau de reconnaissance par Rétine

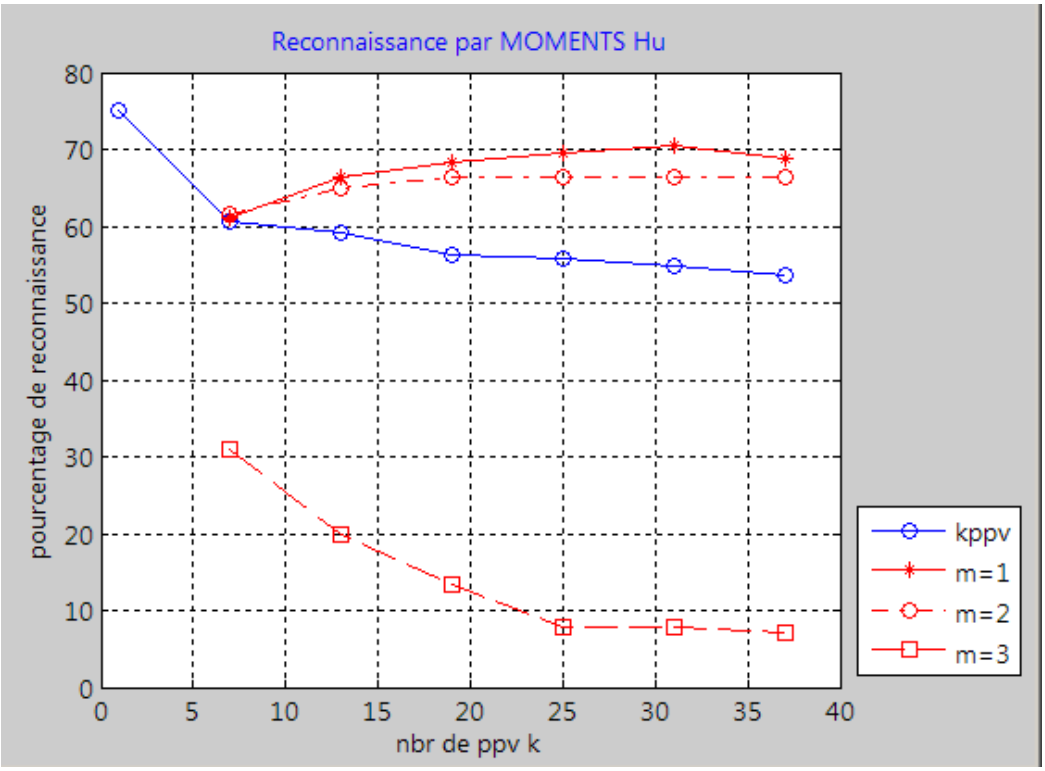


Figure 5.9. Le tau de reconnaissance par les moments Hu

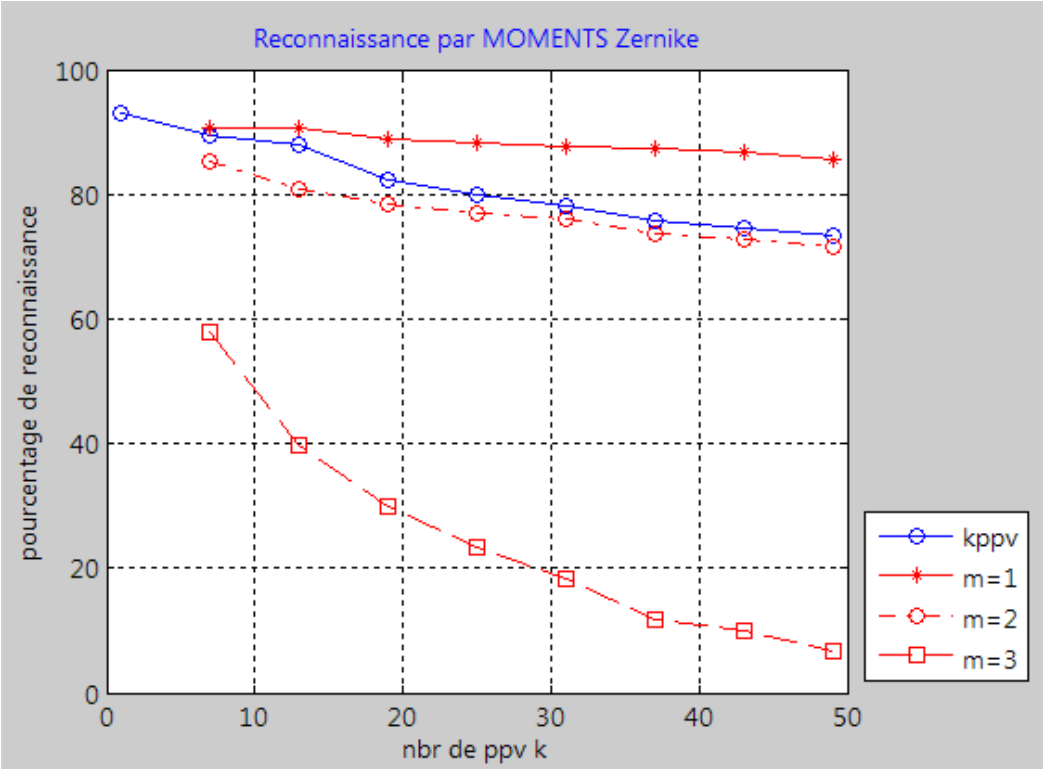


Figure 5.10. Le tau de reconnaissance par les moments Zernike

5.4.3 Le Réseau de Neurones

Nous avons utilisé un perceptron simple, et un perceptron multicouche en tenant compte des considérations pratiques suivantes :

- Les poids du réseau doivent être initialisés à de petites valeurs aléatoires.
- la valeur du taux d'apprentissage « η » a un effet significatif sur les performances du réseau. Si ce taux est petit l'algorithme converge lentement, par contre s'il est grand l'algorithme risque de générer des oscillations. Nous avons assuré un changement dynamique de pas d'apprentissage et de momentum de 0 à 1, pour assurer la convergence de l'algorithme vers une solution optimale.

- Il n'existe pas de règles permettant de déterminer le nombre de couches cachées dans un réseau donné ni le nombre de neurones dans chacune d'elles. Avec une couche cachée, il est capable, avec un nombre suffisant de neurones, d'approximer toute fonction continue. Un nombre plus important permet donc de mieux coller aux données présentées mais diminue la capacité de généralisation du réseau. il n'existe pas de règle générale mais des règles empiriques. La taille de la couche cachée doit être Soit égale à celle de la couche d'entrée, Soit égale à 75% de celle-ci, Soit égale à la racine carrée du produit des nombres dans la couche d'entrée et de sortie.

Nous avons utilisée un RdN d'une seule couche cachée en changeant le nombre de neurones.

- Théoriquement, l'algorithme ne doit se terminer que lorsque le minimum de l'erreur commise par le réseau est atteint, correspondant à un gradient nul, ce qui n'est jamais rencontré en pratique. C'est pourquoi un seuil est fixé à priori ou un nombre d'itérations afin d'arrêter l'apprentissage.
- Nous avons considéré la fonction « sigmoïde » pour le passage de la couche d'entrée à la couche cachée et de cette dernière à la couche de sortie.

Table 5.3. Taux de reconnaissance réseaux de neurones

attributs	perceptron	PMC
Cavités	36.40 %	72.43 %
Zonage	31.10 %	65.38 %
Profil	73.60 %	74.12 %
Histogramme	82.50 %	85.24 %
Moments Hu	19.80 %	71.09 %
Moments Zernike	69.50 %	72.03 %
Rétine	98.10 %	69.97 %

Au vu des résultats de la table 5.3., nous pouvons affirmer que le RdN fonctionne correctement puisqu'il apprend bien les divers exemples qu'on lui présente. Néanmoins nous voyons que les résultats obtenus à partir d'images non apprises au préalable sont légèrement moins bons. Cela peut être dû à diverses causes :

- Soit le nombre de neurones de la couche cachée ainsi que la valeur du pas ne sont pas satisfaisants et nous devrions faire des tests pour trouver les valeurs optimales.
- Soit les moments Hu et de Zernike ne permettent pas d'avoir des résultats optimaux sur le type d'image que l'on utilise.
- Nombre d'itération est peut être insuffisant.

5.6 Optimisation

Les cavités caractérisent bien le chiffre (KPPV : 82.60%, RdN : 72.43 %). Cependant, il peut y avoir des ambiguïtés dans certains cas. La même chose pour le zonage (KPPV : 83.30%, RdN : 65.38 %) et le profil (KPPV : 93.60 %, RdN : 74.12 %) et pour les histogrammes de projections (KPPV : 97.40%, RdN : 85.24 %), etc.

A ce stade, nous avons proposé une méthode qui consiste à améliorer les performance de la méthode en ajoutant des caractéristiques supplémentaires c. à. d de faire une combinaison de caractéristiques de différents type (zonage, profil, nombre de cavités etc.).

Après plusieurs essais, nous somme arrivé à trouver que les meilleures

combinaisons, c. à. d celles qui donnent le meilleur taux de reconnaissance, sont : « cavités, nombre de cavités, zonage, profil», « cavités, nombre de cavités, zonage, profil, moment Hu », « Rétine, moment Zernike », comme le montre la table 5.4. et les figures 5.11 jusqu'à 5.13

La deuxième et la troisième combinaison contenant les moments Hu et les moments Zernike, permet de prendre en considération l'invariance à l'échelle, translation, rotation.

Table 5.4. Taux de reconnaissance par K_ppv et K_ppv flou

Attributs	K	KPPV	KPPV flou			Percept-ron	PMC
			M=1	M=2	M=3		
Cavités+Zoning+ Profils + Nombre Cavités	**1**	98,40 %	/	/	/	95,80 %	73,70 %
	6	97,80 %	97,60 %	95,80 %	90,40 %		
	11	96,40 %	97,80 %	93,70 %	83,70 %		
	16	96,20 %	97,00 %	93,10 %	82,00 %		
	21	96,10 %	97,00 %	92,20 %	78,80 %		
	26	96,20 %	97,10 %	91,10 %	74,90 %		
Cavités+Zoning+ Nombre Cavités + profils + moments Hu	**1**	98,40 %	/	/	/	96,40 %	79,10 %
	6	97,50 %	98,00 %	95,90 %	90,70 %		
	11	96,40 %	67,60 %	93,70 %	84,00 %		
	16	96,30 %	97,00 %	93,10 %	82,20 %		
	21	96,10 %	97,10 %	91,90 %	78,30 %		
	26	96,20 %	97,00 %	90,80 %	74,70 %		
Rétine + Moments Zernike	**1**	96,80 %	/	/	/	94,00 %	67,87 %
	6	94,50 %	93,80 %	91,10 %	83,80 %		
	11	92,20 %	94,40 %	90,60 %	76,40 %		
	16	91,40 %	94,20 %	89,10 %	71,30 %		
	21	90,90 %	94,60 %	86,70 %	67,10 %		
	26	91,20 %	94,50 %	86,00 %	64,40 %		

Afin de rendre les résultats représentatifs, nous avons testé le projet par des chiffres, de 0 à 9 (image des chiffres, créé manuellement, pour ne pas alourdir le chapitre voir annexe E). Finalement par une image qui contient des chiffres extrait d'un code à barres.

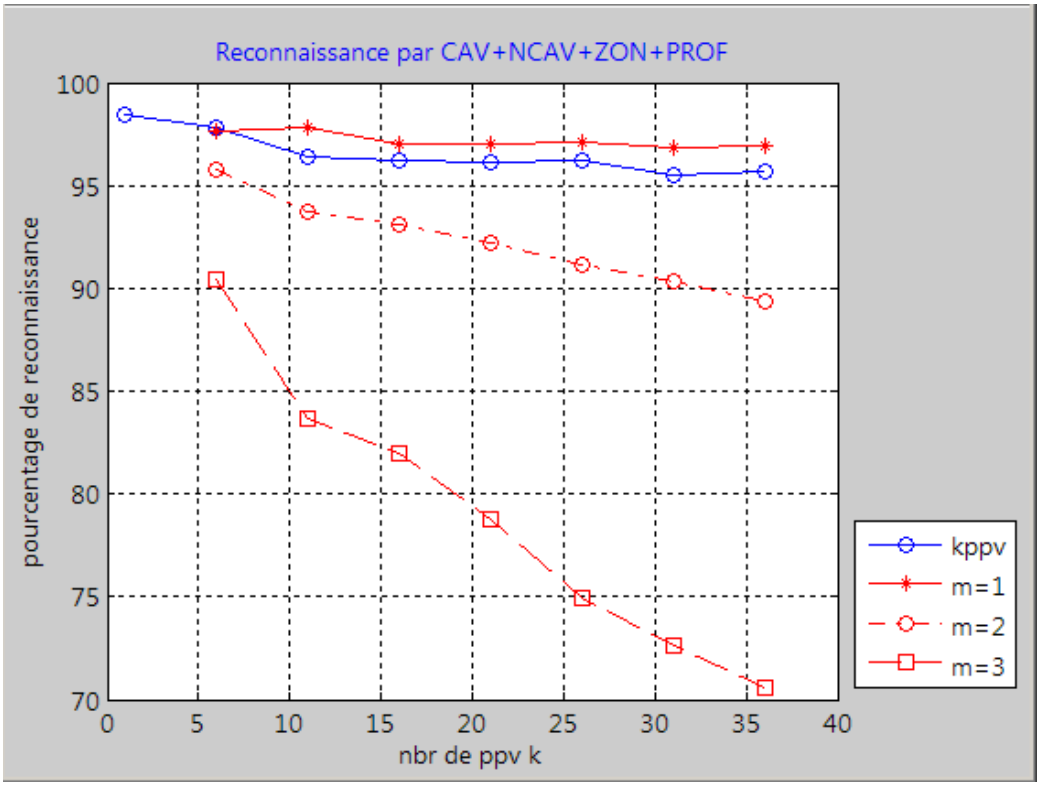


Figure 5.11. Le tau de reconnaissance par la première combinaison

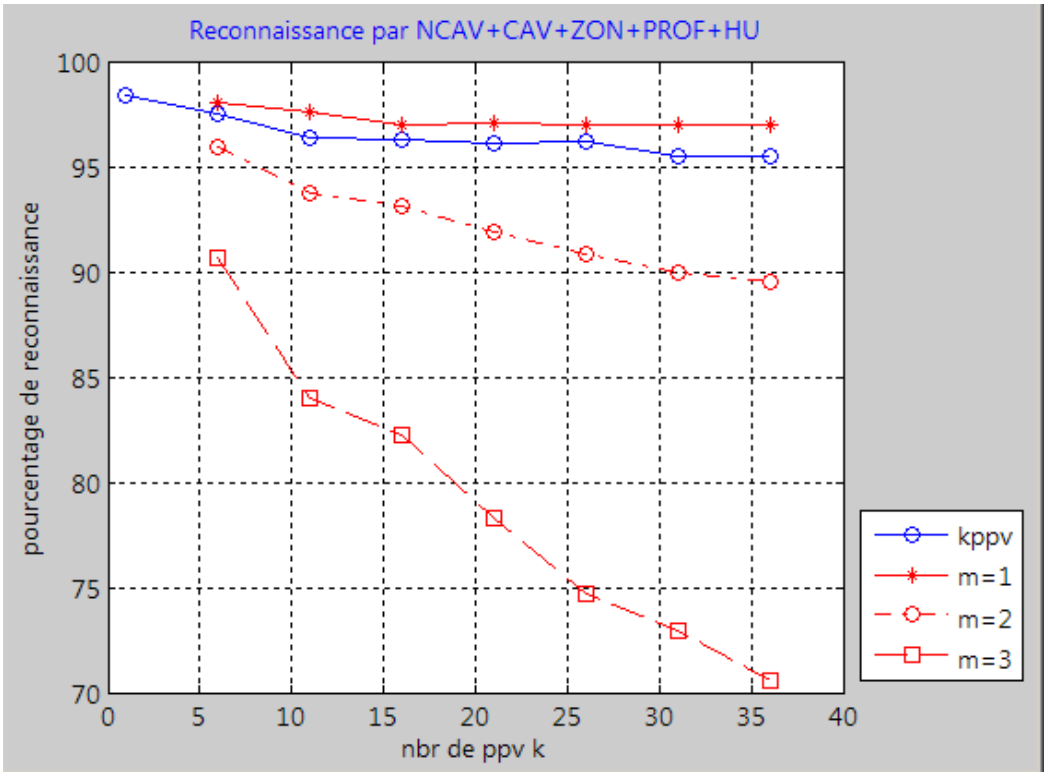


Figure 5.12. Le tau de reconnaissance par la deuxième combinaison

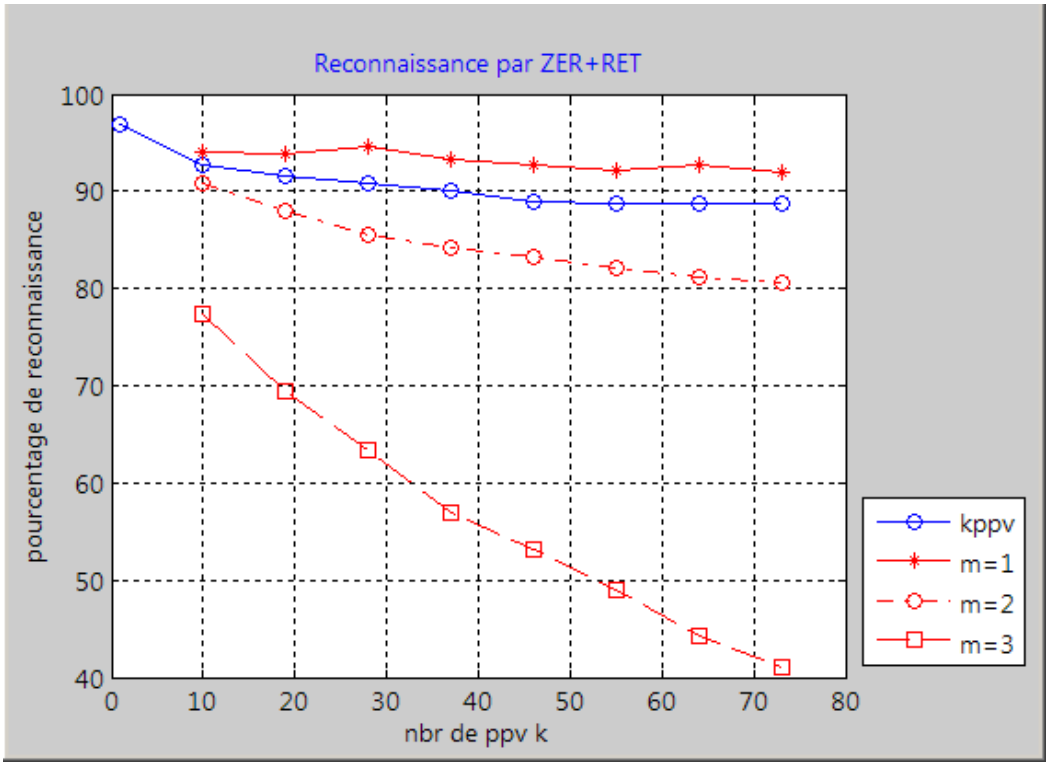


Figure 5.13. Le tau de reconnaissance par la troisième combinaison

5.7 Application sur un code à barre

Le diagramme de notre application est le suivant (figure 5.14)

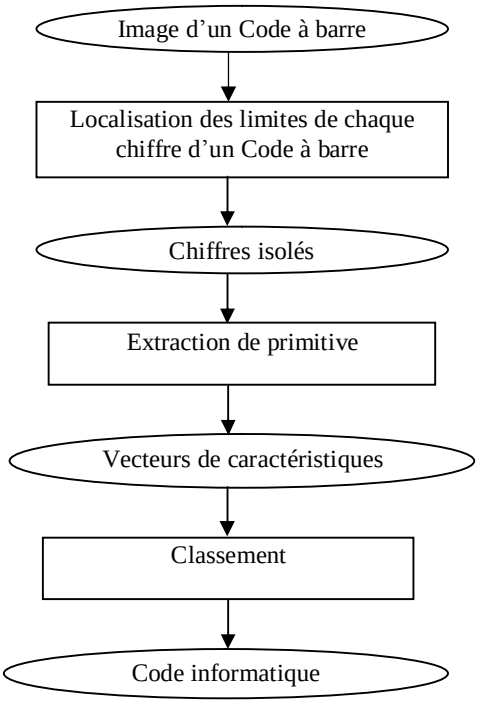


Figure 5.14. Structure d'un système de reconnaissance automatique des chiffres d'un code à barre

Nous avons scanner un code à barre d'un produit agroalimentaire (le thé) représenté sur la figure 5.15.

Figure 5.15. Code à barre

Après élimination des informations inutiles et la localisation, nous avons obtenu l'image représentée sur la figure 5.16.:

Figure 5.16. Localisation des chiffres code à barre

En utilisant les quatre classifieurs et toutes les types de paramètres caractéristique, nous obtenons les résultats suivants (table 5.5.) sachant que la séquence réel des chiffre est : « 3 4 1 9 0 4 7 0 0 3 5 2 8 » , et « ? » chiffre inconnu

Table 5.5. Résultat de reconnaissance des chiffres d'un code à barre

	KPPV	**KPPV Flou**	**Perceptron**	**PMC**
Cavités	900900100355 8	980900100355 8	30?900300322 8	/
Zonage	4410027003428	4410427003478	0490009009900	/
Profils	3419447003528	3419447003528	4440447003424	/
Histogramme	3419047003528	3419047003528	3408547 0?3525	/
Moments Hu	2772277222223	2772277225223	2272227222222	/
Moments Zernike	3773077003558	5773877803558	3773177113521	/
Rétine	3419047003528	3419047003528	8418047888588	/
Cavités+Zoning+ Profils + Nombre Cavités	3410047003528	3419047003528	8888888888888	/
Cavités+Zoning+ Nombre Cavités + profils + moments Hu	3410047003528	3419047003528	3419?47003528	/
Moments Zernike + Rétine	3419042003578	3419047003528	3419?47003528	/

5.8 Conclusion

D'après les résultats, nous pouvons conclure que notre système donne de bonnes performances dans le cas de KPPV ; un taux de reconnaissance de 98.40 %.

Nous estimons même que c'est une bonne performance dans le cas de RdN (perceptron, PMC), nous obtenons un taux de 96.40% pour le perceptron et 79.10% pour le PMC, il faut signaler peut être que notre base d'apprentissage est petite par rapport à ce qu'elle devrait être (de l'ordre de 5000 exemples au lieu de 455) surtout pour les moments Zernike.

CONCLUSION GENERALE

Nous avons proposé dans ce travail un système de lecture automatique des chiffres d'un code à barre. Il s'articule essentiellement autour de deux parties. Une partie d'apprentissage et une de reconnaissance.

Dans la première partie nous avons effectué l'acquisition des donnée, la définition des attributs caractérisant les données (les paramètres caractéristique vus comme les plus pertinents : les cavités, le zonage, les profils, l'histogramme, les moments Hu et les moments Zernike, la rétine, et des combinaisons de ces paramètres), et la construction de la base d'apprentissage. A la fin de cette phase nous obtenons une matrice qui contient les données et les attributs les caractérisant.

La deuxième partie, nous avons utilisé différentes classifieurs pour la reconnaissance du chiffre arabe : les RdN (perceptron simple, PMC), le KPPV, et KPPV flou. Où les RdN sont massivement parallèles, et que chaque neurone peut être simulé électroniquement, ce qui peut réaliser une implantation électronique du système réalisé. Le classifieur est l'élément adaptatif du système de reconnaissance. Pour fonctionner, il doit, au préalable, avoir été réglé par apprentissage. L'apprentissage consiste à donner au classifieur des exemples de chiffres, en lui indiquant, à chaque fois, de quel chiffre il s'agit (un ensemble de données étiquetées que l'on nomme base d'entraînement c'est à dire apprendre la description d'objets a partir d'une base d'exemples)

Nous avons fournis à notre système une image de code à barre. La première étape consiste à localiser et à isoler les chiffres qui composent le code à barre, la seconde étape consiste à effectuer un certain nombre de mesures qui vont permettre de caractériser la forme que nous transformerons ensuite en un vecteur. Dans la troisième étape, ce vecteur est présenté à l'entré d'un classifieur, qui nous fournira en sortie la classe du chiffre

Le taux de reconnaissance obtenu (98.40 %) est très satisfaisant, mais comme dans tout logiciel, des améliorations restent toujours à apporter. Faisons ainsi une liste non exhaustive des dites améliorations susceptibles d'être apporter ultérieurement à ce projet.

L'optimisation des résultats peut être proposée par le prétraitement (surtout le redressement), l'utilisation d'un autre logiciel graphique, et par combinaison de plusieurs classifieurs exploitant ainsi les avantages de chacun par rapport aux autres.

Signalons enfin l'émergence d'une nouvelle classe de systèmes d'apprentissage basés sur les algorithmes génétiques, ainsi que les méthodes hybrides.

REFERENCES BIBLIOGRAPHIQUES

Bibliographie

[1] M. Kunt, « *Reconnaissance des formes et analyse de scènes* », collection électricité, traitement de l'information : volume 3, presses polytechniques et universitaires romandes,2000.

[2] G. Burel, « *Introduction au traitement d'images, simulation sous MATLAB* », Edition lavoisier, 2001.

[3] B. Gosselin, « *Application de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits* », Faculté Polytechnique de Mons, présentée pour l'obtention du grade de Docteur en Sciences Appliquées, 1995-1996.

[4] M. Milgram, « *Reconnaissance des formes méthodes numériques et connexionnistes* », Edition Armand colin, Paris, 1993.

[5] W. AlFalou, « Reconnaissance de caractères manuscrits par réseau de neurones », DEA Modélisation et ingénierie du logiciel scientifique Université de Reims – Université de Rennes – IRISA Ecole Polytechnique Fédérale de Lausanne, 1998.

[6] L. Miclet, « *Méthodes structurelles pour la reconnaissance des formes* », Editions Eyrolles, 1984.

[7] P. Lefevre, « Reconnaissance de l'imprimé », Technique de l'ingénieur, H 1 348.

[8] M. Teague, « Image analysis via the general theory of moments », Optical Society of America, Vol. 70, 1980, pp. 921-930.

[9] O. D. Trier,A. K. Jain, T. Taxt, « Feature extraction methods for character recognition -a survey», Pattern Recognition, Vol. 29, 1996, pp. 641-662.

[10] D. Arrivault, «Apport des Graphes dans la Reconnaissance Non-Contrainte de Caractères Manuscrits Anciens », Thèse pour l'obtention du Grade de Docteur de l'université de Poitiers, 2006.

[11] M. K. Hu, « Visuel pattern recognition by moment invariants », IRE Transactions On Information Theory, 1962, pp. 179-187.

[12] F. Ghorbel, S. Derrode, R. Mezhoud, T. Bannour, S. Dhahbi, « Image reconstruction from a complete set of similarity invariants extracted from complex moments », 2004.

[13] R. O. Duda, P. E. Hart, D. G. Stork, « *Pattern classification* », Second edition, 1992.

[14] J. P Renard, «réseaux neuronaux », Editions Vuibert, Paris, 2006.

[15] G. Dreyfus, «les réseaux de neurones », Mécanique Industrielle et Matériaux n°51, école Supérieure de Physique et de Chimie Industrielles de la Ville de Paris (ESPCI), Laboratoire d'électronique, 1998.

[16] I. Rivals, L. Personnaz, G. Dreyfus, « Modélisation, classification et commande par réseaux de neurones : principes fondamentaux, méthodologie de conception et illustrations industrielles », Ecole supérieure de physique et de chimie industrielles de la ville de paris laboratoire d'électronique, 1995.

[17] I. Rivals, « Les réseaux de neurones formels pour le pilotage de robots mobiles », Laboratoire d'Électronique de l'ESPCI (École Supérieure de Physique et de Chimie Industrielles).

[18] G. Dreyfus, «Réseaux de neurones, méthodes et applications », Eyrolles, 2004.

[19] M. Parizeau, «Réseaux de neurones », université Laval, 2004.

[20] J. M. Keller, M. R. Gray, J. A. Givens, « A Fuzzy K-Nearest Neighbor algorithm », IEEE transaction on systems, man, and cybernetics, vol.SMC-15, 1985, pp 580-585.

[21] R. Bondugula, O. Duzlevski, D. Xu « Profiles and fuzzy k-nearest neighbor algorithm For protein secondary structure prediction », Digital biology laboratory, department of computer science, university of missouri-columbia Columbia, Mo 65211, USA.

[22] F. Afsar, M. U. Akram, M. Arif, J. Khurshid, « A Pruned Fuzzy k-Nearest Neighbor

Classifier with Application to Electrocardiogram Based Cardiac Arrhytmia Recognition », Department of Computer & Information Sciences, Pakistan Institute of Engineering & Applied Sciences (PIEAS), P.O. Nilore, Islamabad, Pakistan.

[23] M. Weinfeld, « Réseaux de neurones », techniques de l'ingénieur, H 1990

[24] N. Benoudjit, « Reconnaissance de caractères manuscrits par la méthode des moments », Université Catholique de Louvain, Département d'électricité, laboratoire d'électronique et de microélectronique, travail présenté en vue de l'obtention du diplôme d'étude approfondies en science appliquées, 1999 - 2000.

[25] docs.happycoders.org/orgadoc/artifical_intelligence/classification_documents/classification.pdf

[26] http://www.grappa.univ-lille3.fr/polys/fouille/sortie005.html#toc13

[27] www.seas.upenn.edu/~timothee/papers/vision_system.pdf

[28] www.univ-provence.fr/gsite/Local/umr_6149/umr/page_perso/Touzet/Les_reseaux_de_neurones_artificiels.pdf

[29] www.codesbarres.com/Manuals/frprimer.pdf

[30] www.tracenews.info/IMG/pdf/livre_a.macaigne.pdf

[31] http://www.gomaro.ch/codeean.htm

[32] http://grandzebu.net/index.php?page=/informatique/codbar/ean13.htm

ANNEXES

Annexe A

Les moments Zernike invariants jusqu'à l'ordre 12

$2^{ème}$ ordre

$S_1=|A_{20}|$

$S_2=|A_{22}|$

$3^{ème}$ ordre

$S_3=|A_{31}|$

$S_4=|A_{33}|$

$4^{ème}$ ordre

$S_5=|A_{40}|$

$S_6=|A_{42}|$

$S_7=|A_{44}|$

$5^{ème}$ ordre

$S_8=|A_{51}|$

$S_9=|A_{53}|$

$S_{10}=|A_{55}|$

$6^{ème}$ ordre

$S_{11}=|A_{60}|$

$S_{12}=|A_{62}|$

$S_{13}=|A_{64}|$

$S_{14}=|A_{66}|$

$7^{ème}$ ordre

$S_{15}=|A_{71}|$

$S_{16}=|A_{73}|$

$S_{17}=|A_{75}|$

$S_{18}=|A_{77}|$

$8^{ème}$ ordre

$S_{19}=|A_{80}|$

$S_{20}=|A_{82}|$

$S_{21}=|A_{84}|$

$S_{22}=|A_{86}|$

$S_{23}=|A_{88}|$

$9^{ème}$ ordre

$S_{24}=|A_{91}|$

$S_{25}=|A_{93}|$

$S_{26}=|A_{95}|$

$S_{27}=|A_{97}|$

$S_{28}=|A_{99}|$

$10^{ème}$ ordre

$S_{29}=|A_{10\ 0}|$

$S_{30}=|A_{10\ 2}|$

$S_{31}=|A_{10\ 4}|$

$S_{32}=|A_{10\ 6}|$

$S_{33}=|A_{10\ 8}|$

$S_{34}=|A_{10\ 10}|$

$11^{ème}$ ordre

$S_{35}=|A_{11\ 1}|$

$S_{36}=|A_{11\ 3}|$

$S_{37}=|A_{11\ 5}|$

$S_{38}=|A_{11\ 7}|$

$S_{39}=|A_{11\ 9}|$

$S_{40}=|A_{11\ 11}|$

$12^{ème}$ ordre

$S_{41}=|A_{12\ 0}|$

$S_{42}=|A_{12\ 2}|$

$S_{43}=|A_{12\ 4}|$

$S_{44}=|A_{12\ 6}|$

$S_{45}=|A_{12\ 8}|$

$S_{46}=|A_{12\ 10}|$

$S_{47}=|A_{12\ 12}|$

Annexe B

La liste des codes pour les autres pays

Préfixe	Pays et nom de l'organisation
00 à 13	Etats unis et canada UCC
20 à 29	Codification interne en magasin (in-store
30 à 37	France GENCOD-EAN
378 et 379	Codification interne (national) de Presse de
380	Bulgarie CCI of Bulgaria
383	Slovénie SANA
385	Croatie CRO-EAN
387	Boznie-Herzégovine EAN-BIH
400 à 440	Allemagne CCG
45 et 49	Japon DCC (Distribution Code Center)
460 à 469	Russie UNISCAN
471	Taïwan CAN
474	Estonie EAN
475	Lettonie EAN
476	Azerbaijan EAN
477	Lituanie EAN
478	Ouzbékistan EAN
479	Sri Lanka EAN
480	Philippines PANC
481	Biélorussie EAN
482	Ukraine EAN
484	Moldavie EAN
485	Arménie EAN
486	Géorgie EAN
487	Kazakhstan EAN
489	Hong Kong HKANA
50	U.K. GS1
520	Grèce EAN
528	Liban EAN
529	Chypre EAN
531	Macédoine EAN-MAC
535	Malte MANA

539	Irlande GS1
54	Belgique & Luxembourg EAN
560	Portugal GS1
569	Islande EAN
57	Danemark EAN
590	Pologne EAN
594	Roumanie EAN
599	Hongrie HAPMH
600 et 601	Afrique du Sud EAN
609	Île Maurice EAN
611	Maroc EAN
613	Algérie EAN
619	Tunisie Tunicode
621	Syrie EAN
622	Égypte EAN
626	Iran EAN
628	Arabie Saoudite EAN
64	Finlande GS1
690 à 692	Chine ANCC (Article Numbering Centre of
70	Norvège EAN
729	Israël IBA (Israeli Barcode Association)
73	Suède EAN
740	Guatemala EAN
741	Salvador EAN E1
742	Honduras ICCC
743	Nicaragua EAN
744	Costa Rica EAN
745	Panama
746	République Dominicaine EAN
750	Mexique AMECE
759	Venezuela EAN
76	Suisse GS1
770	Colombie IAC
773	Uruguay CUNA
775	Pérou APC-EAN Peru

777	Bolivie EAN
779	Argentine CODIGO-EAN
780	Chili EAN
784	Paraguay EAN
786	Équateur ECOP
789	Brésil EAN
80 à 83	Italie INDICOD
84	Espagne AECOC
850	Cuba CCRC
858	Slovaquie EAN
859	République tchèque EAN
860	Yugoslavia YANA
869	Turquie UCCT
87	Nederland EAN
880	Corée du Sud EAN Korea
885	Thaïlande (Thai Article Numbering Council)
888	Singapour SANC
890	Inde EAN
893	Vietnam EAN
899	Indonésie EAN
90 à 91	Autriche EAN
93	Australie EAN
94	Nouvelle-Zélande EAN
955	Malaisie (Malaysian Article Numbering Council)
977	Publications sérielles avec numérotation ISSN
978 et 979	livres avec numérotation ISBN
980	Reçus de remboursement (refund receipts)
99	Coupons

Annexe C

Les lecteurs de code à barres

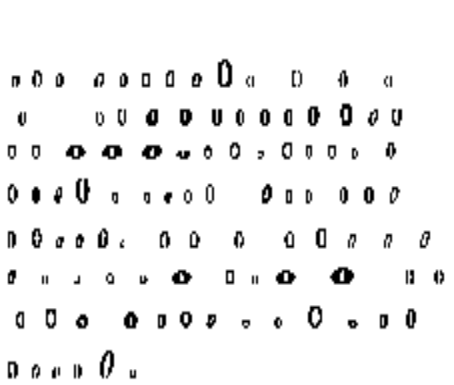
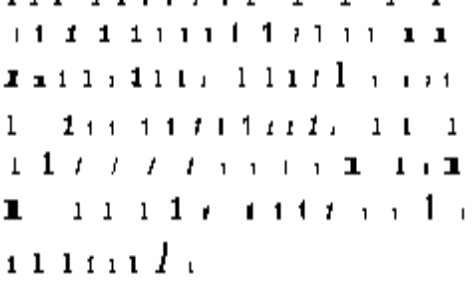
- **Crayon lecteur**

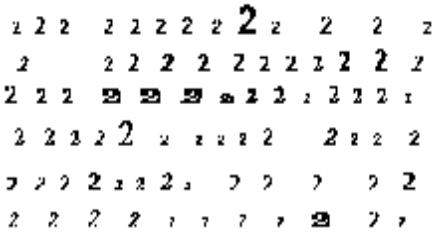
- **Lecteur CCD**

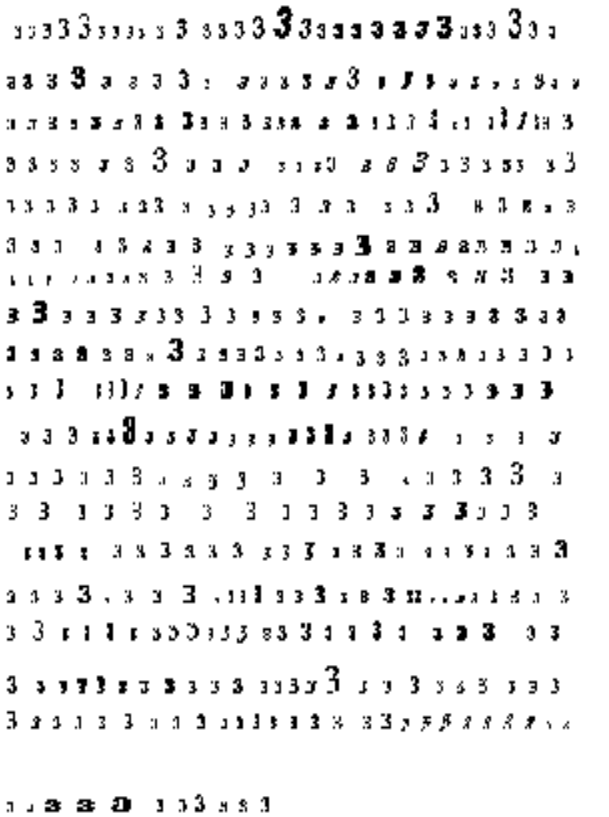
- **Lecteur laser**

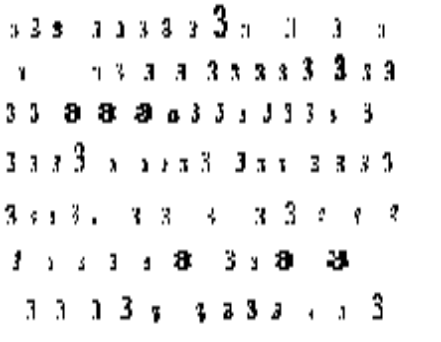
Annexe D

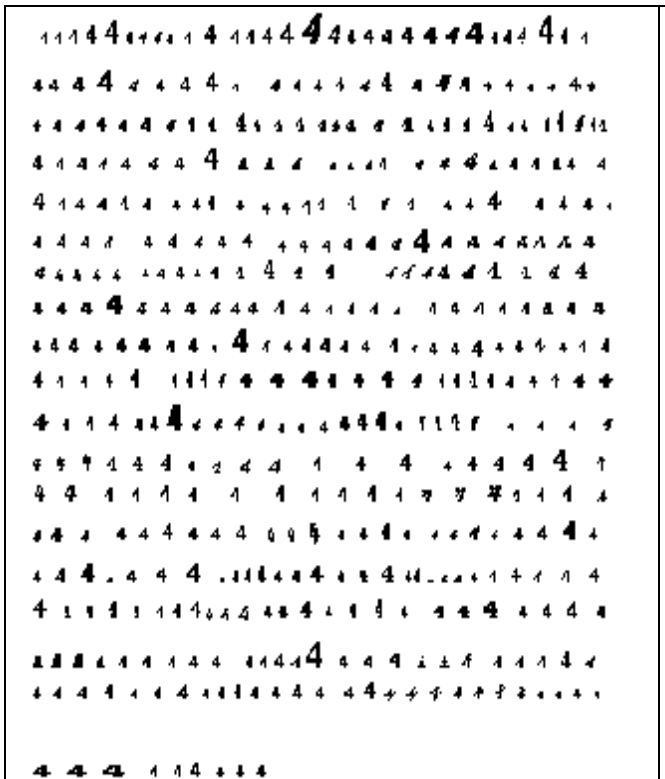
La base d’apprentissage et de test

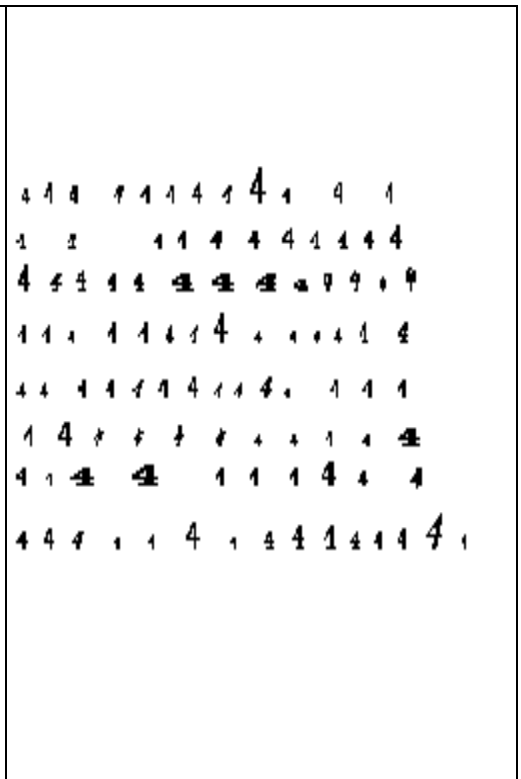
Image d’apprentissage	Image de test
	
	

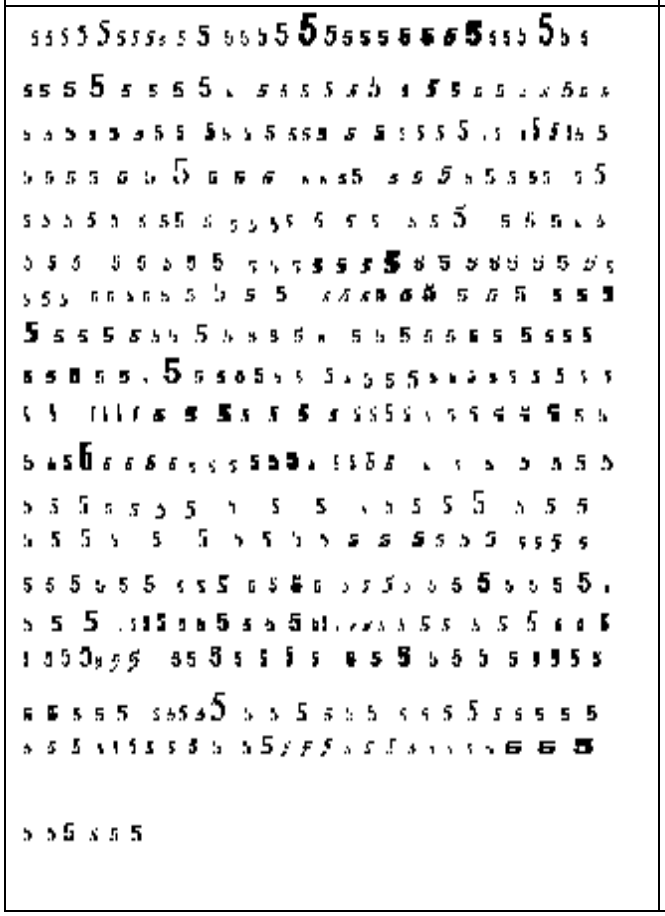


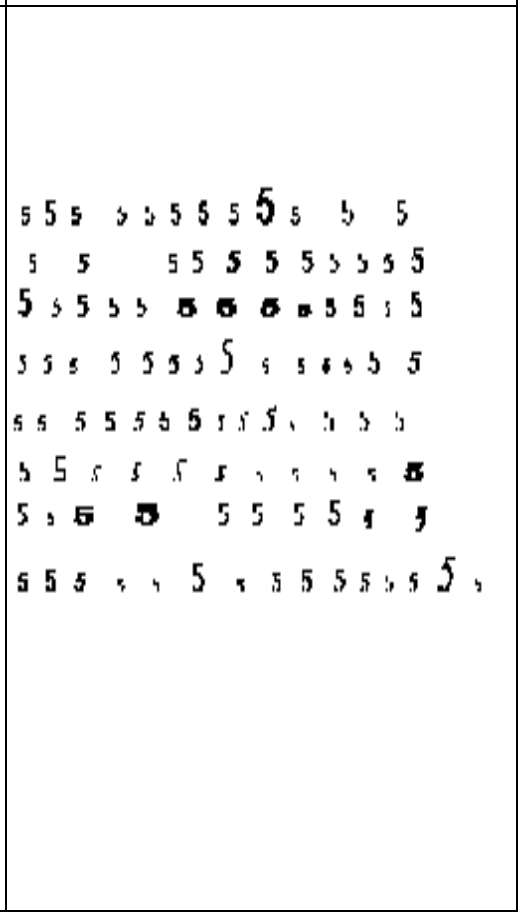


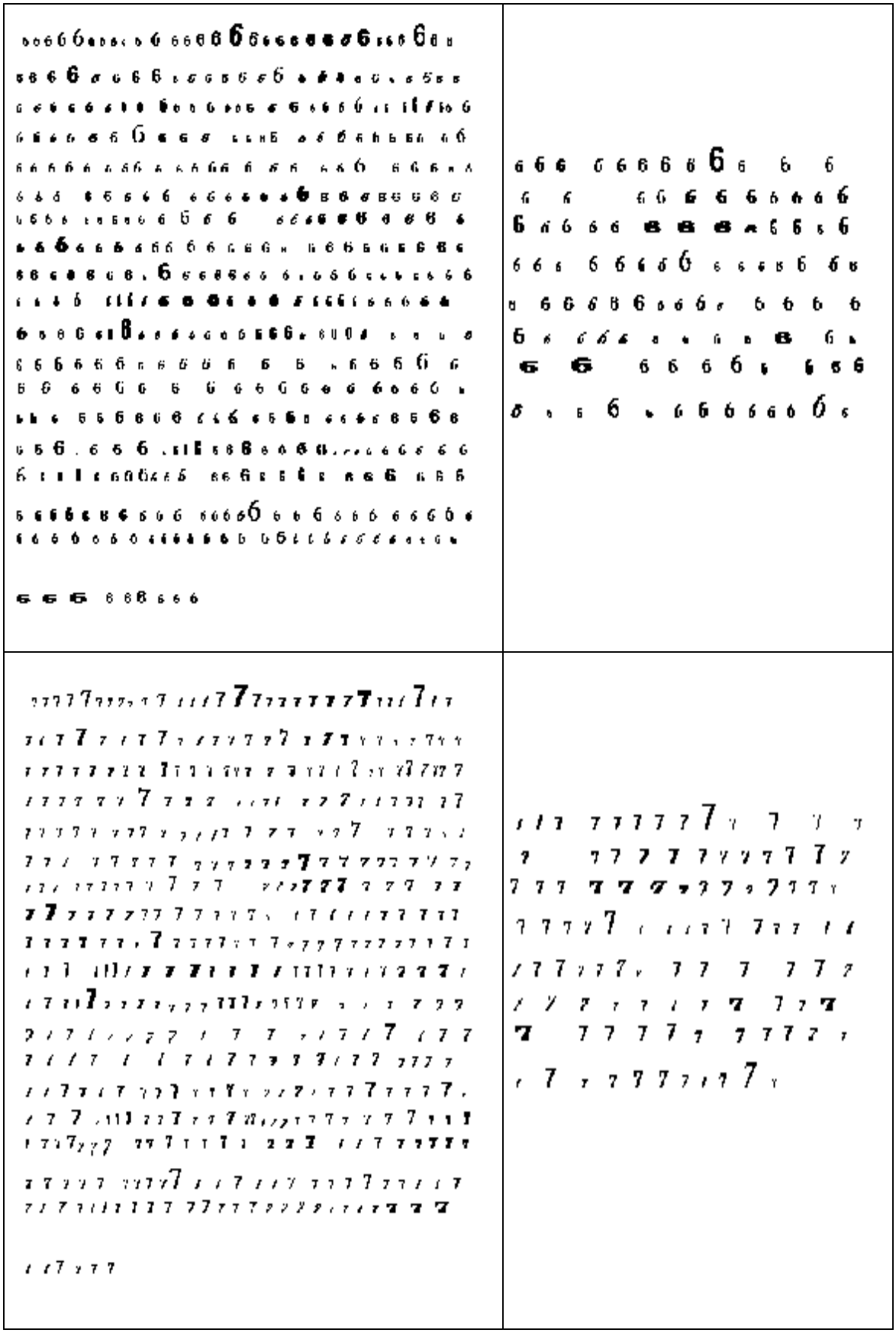


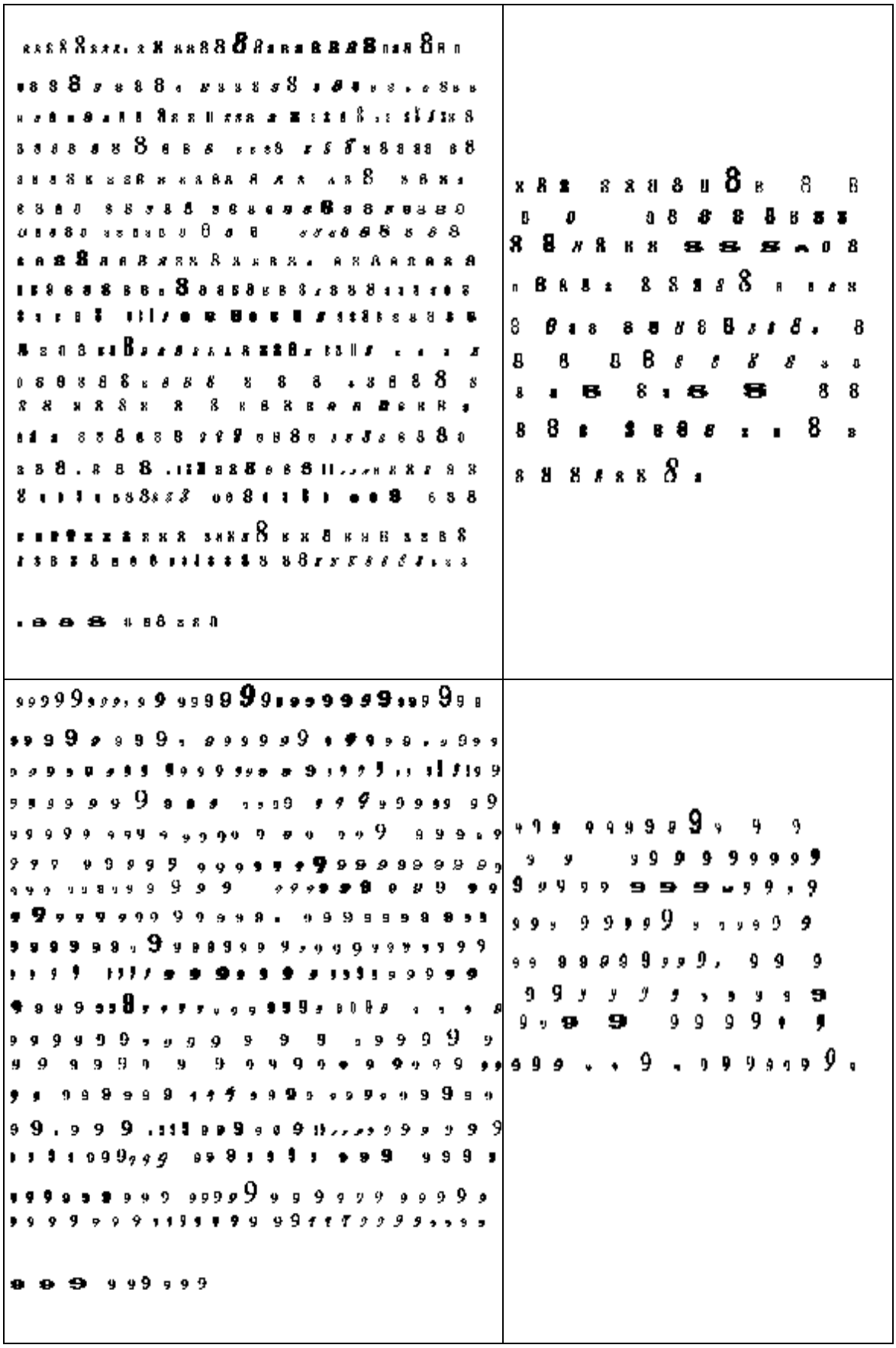










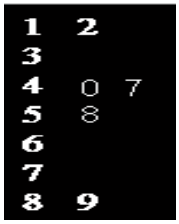


Annexe E

1. Application sur des images de validation

a) Image 1

- Nous avons créé manuellement l'image suivante par le logiciel Paintbrush.



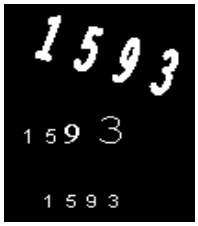
- L'image Après localisation

- La table suivante présente les résultats de reconnaissance

	K_ppv	K_ppv Flou	Perceptron	PMC
Cavités	123001206989	133400206989	333003286383	/
Zonage	123457951789	123457971789	009009900900	/
Profils	123407386789	123407586789	483404386789	/
Histogramme	12340755 6789	12340755 6789	133407556789	/
Moments Hu	133427226728	123427329726	222227222222	/
Moments Zernike	123417516789	123417316489	113117111?19	/
Rétine	123407586789	123437586789	138407886789	/
Cavités+Zoning+ Profils + Nombre Cavités	123407586789	123407386789	883888388888	/
Cavités+Zoning+ Nombre Cavités + profils + moments Hu	123407586789	123407386789	833883388889	/
Moments Zernike + Rétine	123477576789	123477576789	123437536789	/

b) Image 2

- Nous avons créé manuellement l'image suivante par le logiciel Paintbrush.



- L'image Après localisation

- La table suivante présente les résultats de reconnaissance

	K_ppv	K_ppv Flou	Perceptron	PMC
Cavités	224345934233	224345934233	523322332233	/
Zonage	121375777543	321375757545	000099999909	/
Profils	151415931593	151315931503	434445934533	/
Histogramme	559315921503	559315921503	35?305050595	/
Moments Hu	111772277232	113772277232	122272277222	/
Moments Zernike	175177777751	517772777722	?1?177?77117	/
Rétine	353315931593	353315931593	888883938383	/
Cavités+Zoning+ Profils + Nombre Cavités	455215931593	455215931593	888883938383	/
Cavités+Zoning+ Nombre Cavités + profils + moments Hu	455315931593	455215931593	838382938283	/
Moments Zernike + Rétine	159725971593	159725971593	759??595959 3	/

2. Application sur un code à barre

Nous avons scanner un code à barre d'un livre représenté sur la figure suivante.

Après élimination des informations inutile et la localisation nous avons obtenu l'image suivante :

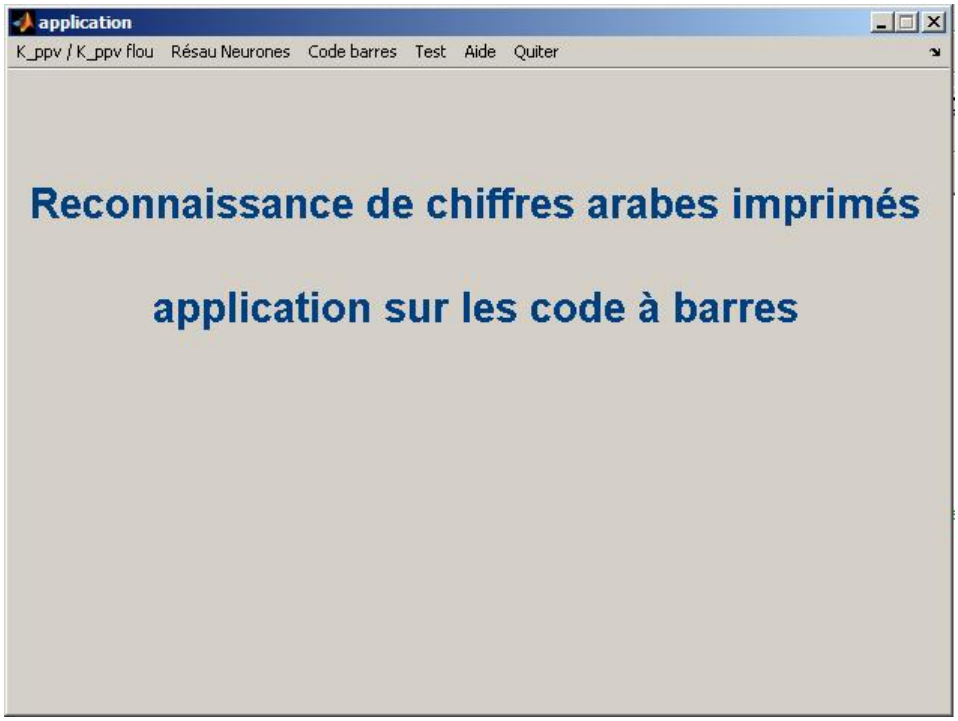
En utilisant les quatre classifieurs et tous les types de paramètres caractéristiques nous obtenons les résultats suivantes sachant que la séquence réelle des chiffres est :

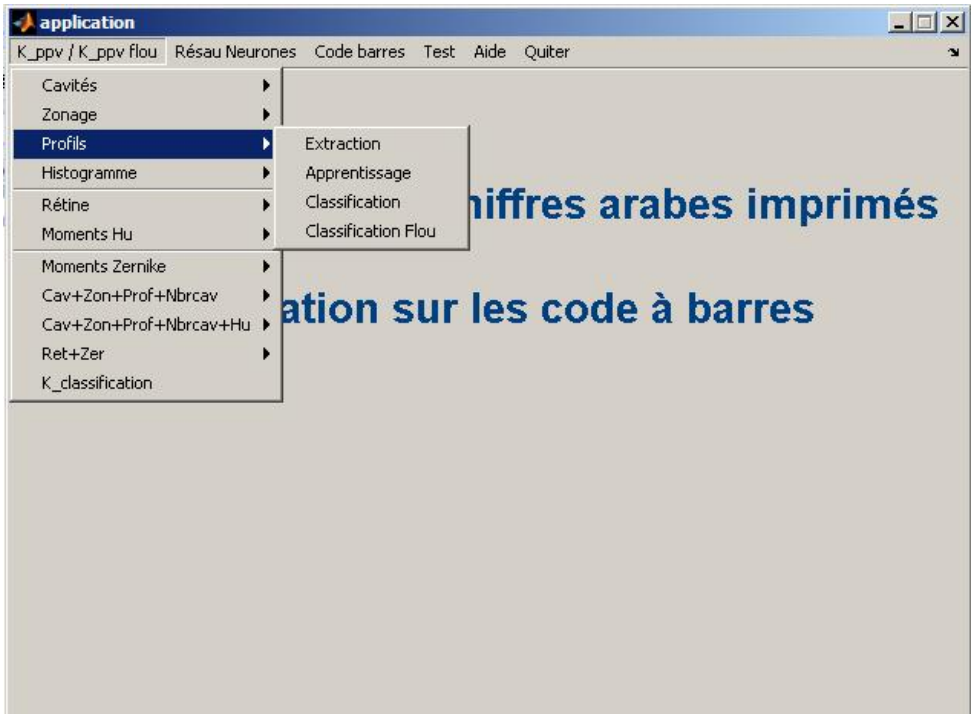
« 9 7 8 2 7 4 3 4 1 2 5 9 3 »

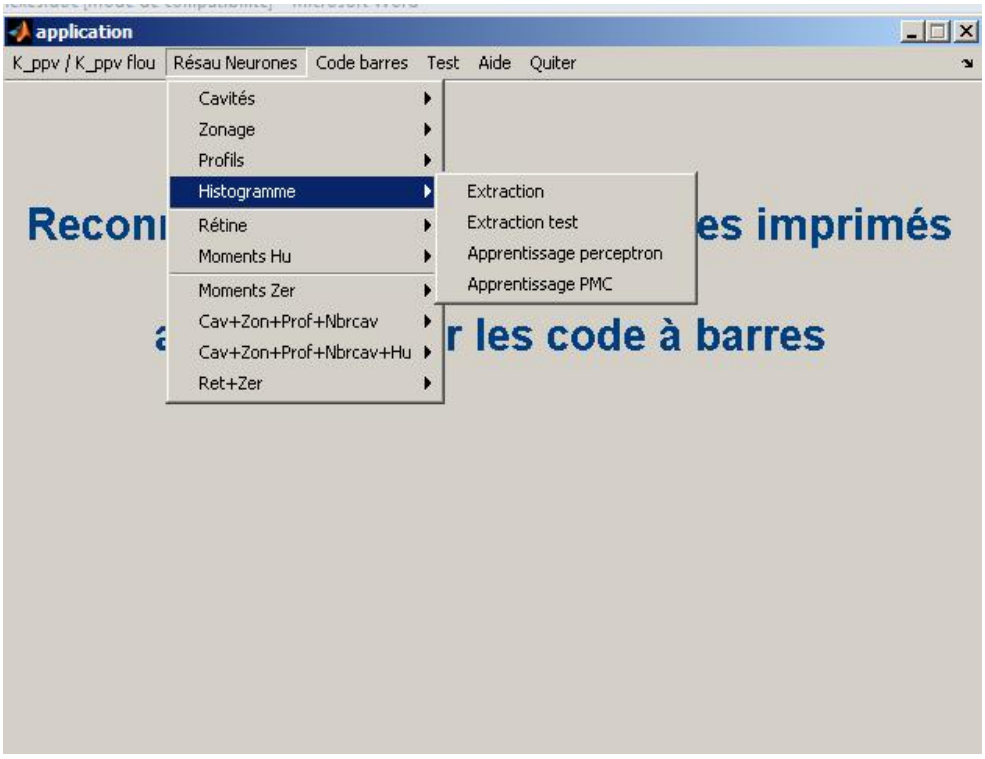
	K_ppv	**K_ppv Flou**	**Perceptron**	**PMC**
Cavités	9181163642293	9185163645593	30?9003003228	/
Zonage	3718517118554	3512554114554	0490009009900	/
Profils	4182745412745	4132745412795	4440447003424	/
Histogramme	9702713112513	9702713112573	3408547 0?3525	/
Moments Hu	3722775772232	3752775772232	2272227222222	/
Moments Zernike	7781777171767	7785777171765	3773177113521	/
Rétine	9782743412593	9782743412593	8418047888588	/
Cavités+Zoning+ Profils + Nombre Cavités	9182743412393	9182745412393	8888888888888	/
Cavités+Zoning+ Nombre Cavités + profils + moments Hu	9182743412393	9182745412393	3888883883328	/
Moments Zernike + Rétine	9682743412593	9682743412593	3419?47003528	/

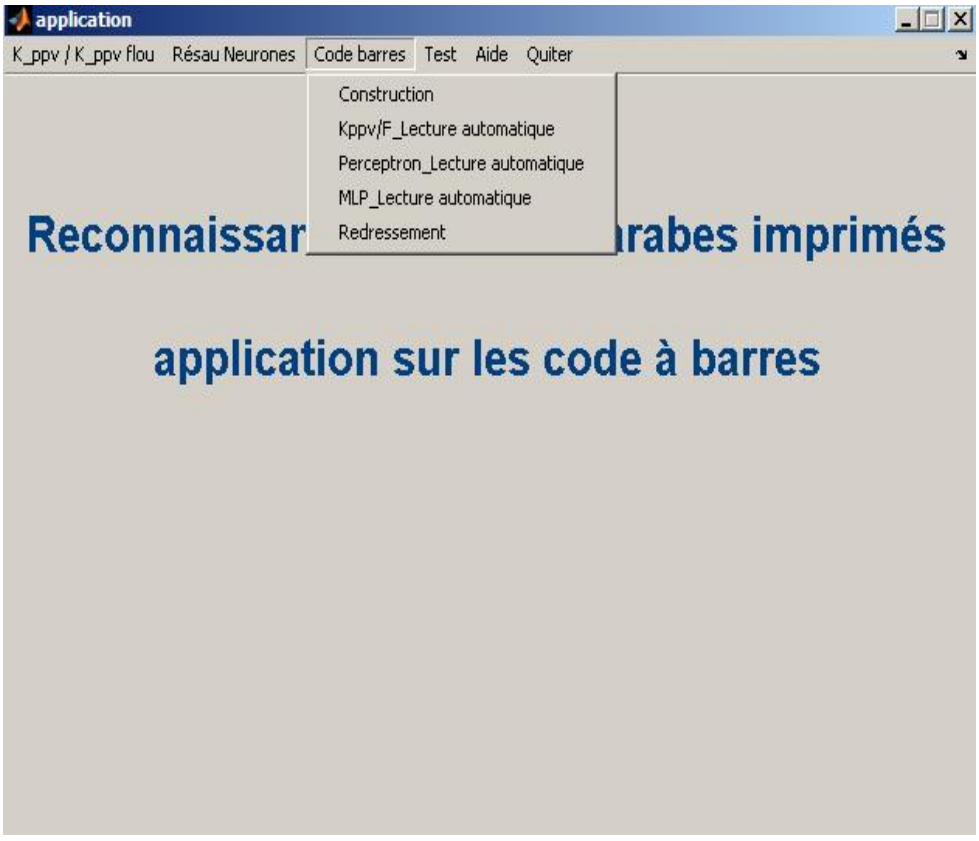
Annexe F

Interface graphique de l'application

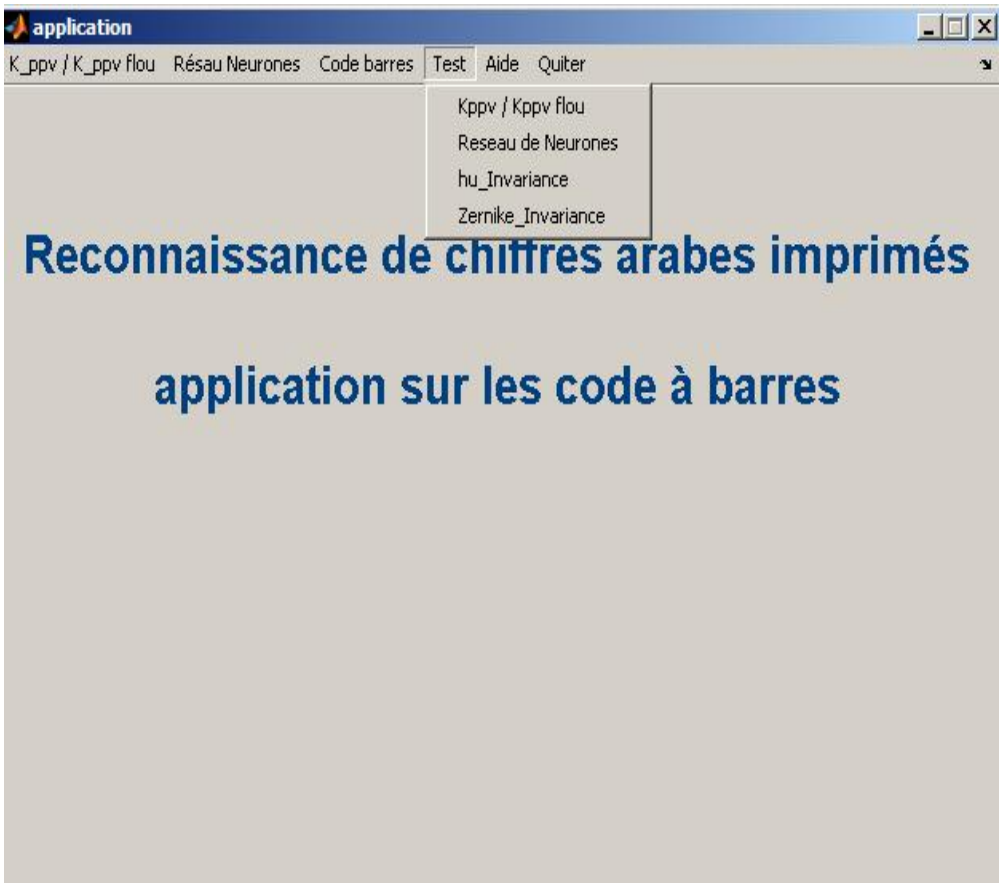


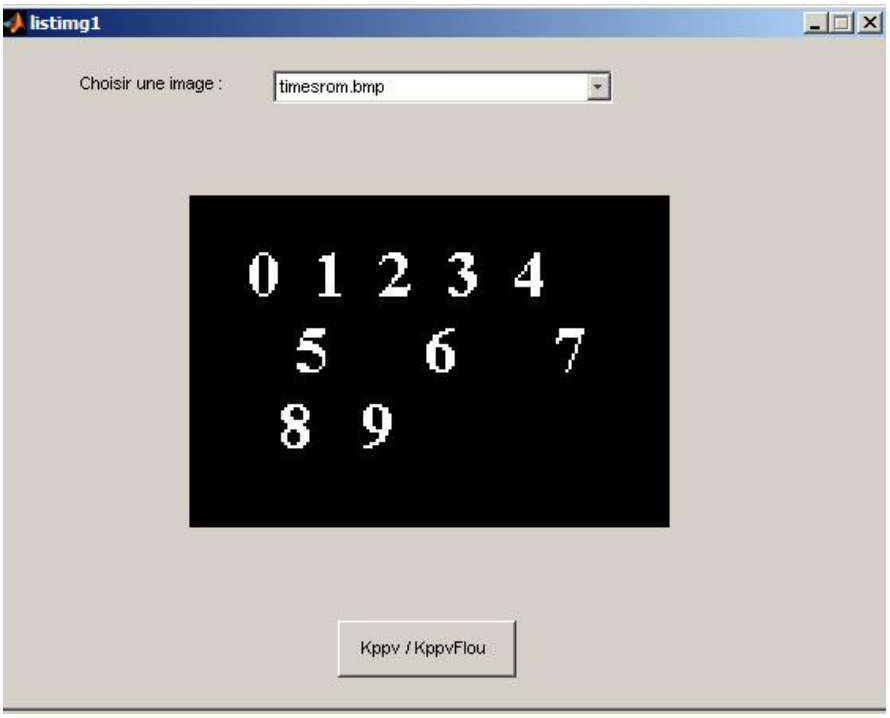


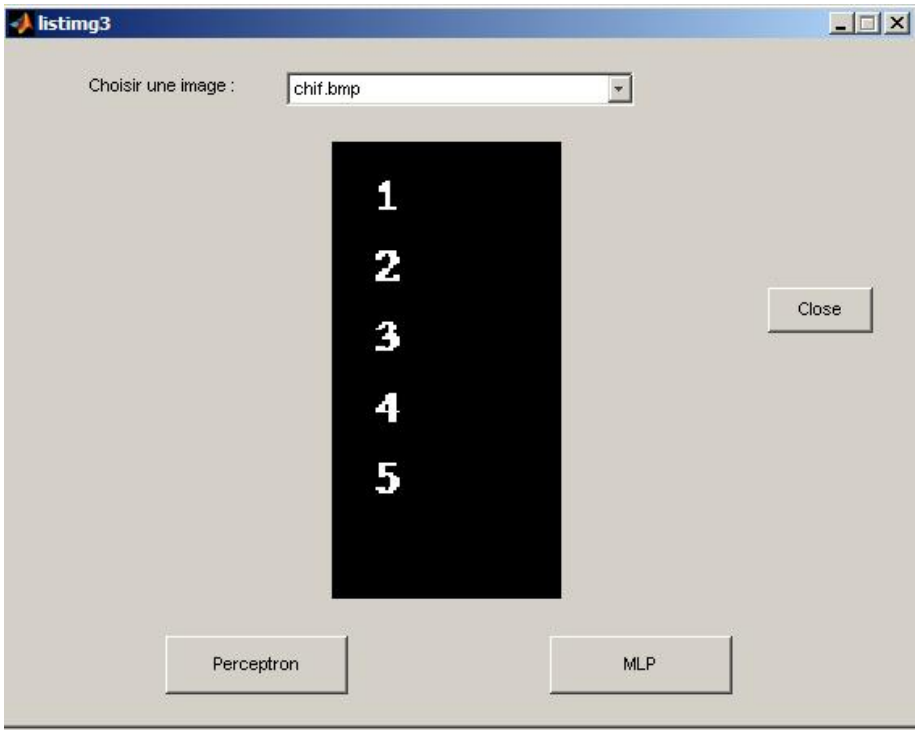

application
K_ppv / K_ppv flou
Résau Neurones
Code barres
Test
Aide
Quiter
Cavités
Zonage
Profils
Histogramme
Rétine
Moments Hu
Moments Zer
Cav+Zon+Prof+Nbrcav
Cav+Zon+Prof+Nbrcav+Hu
Ret+Zer
Extraction
Extraction test
Apprentissage perceptron
Apprentissage PMC
Recon
es imprimés
r les code à barres

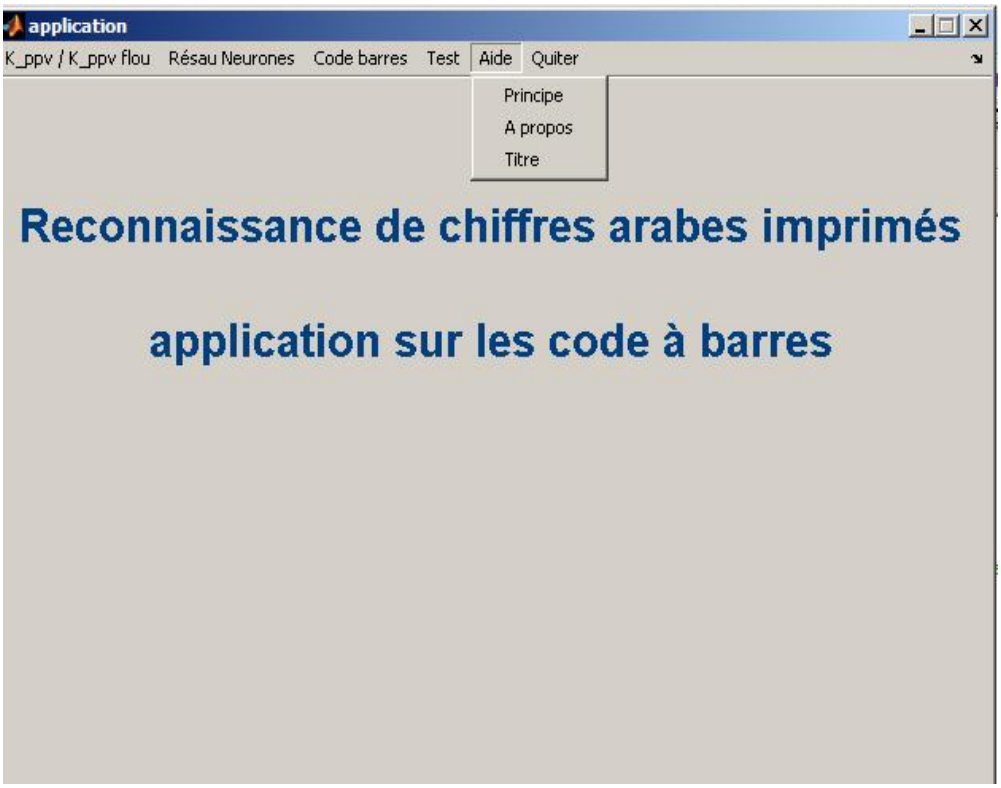

application
K_ppv / K_ppv flou
Résau Neurones
Code barres
Test
Aide
Quiter
Construction
Kppv/F_Lecture automatique
Perceptron_Lecture automatique
MLP_Lecture automatique
Redressement
Reconnaissar
rabes imprimés
application sur les code à barres

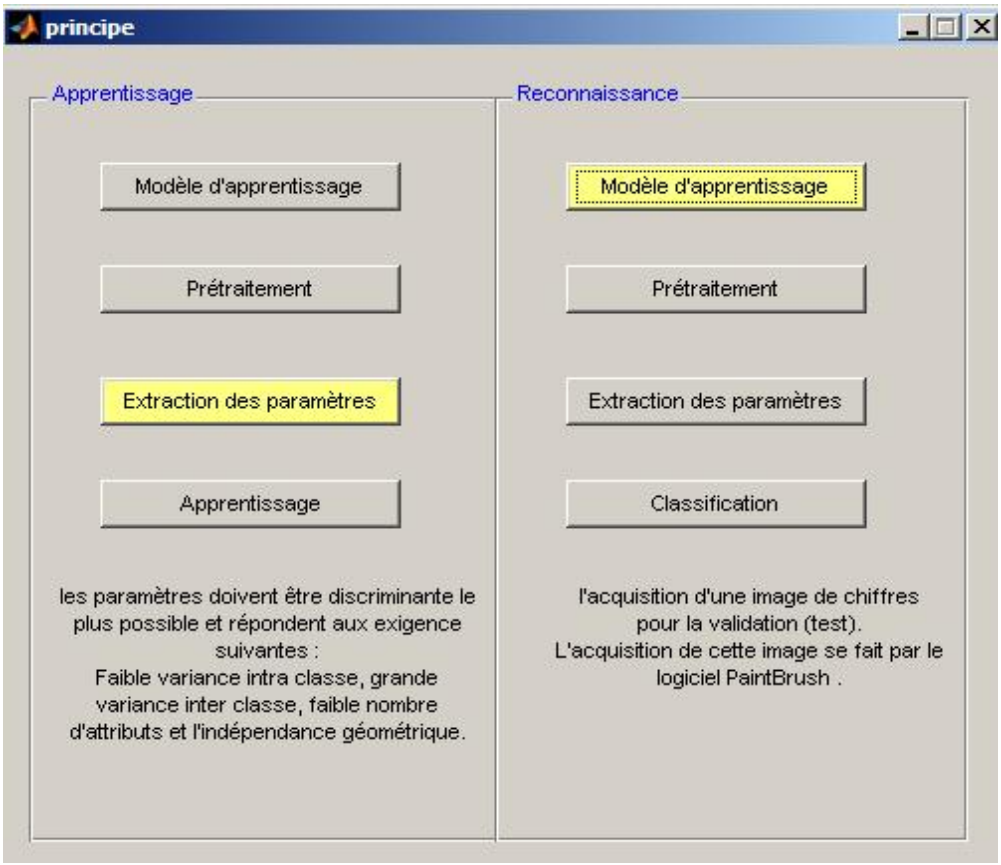

graph_cod
entrer le code EAN S.V.P , sans le code digit :
chek digit
close

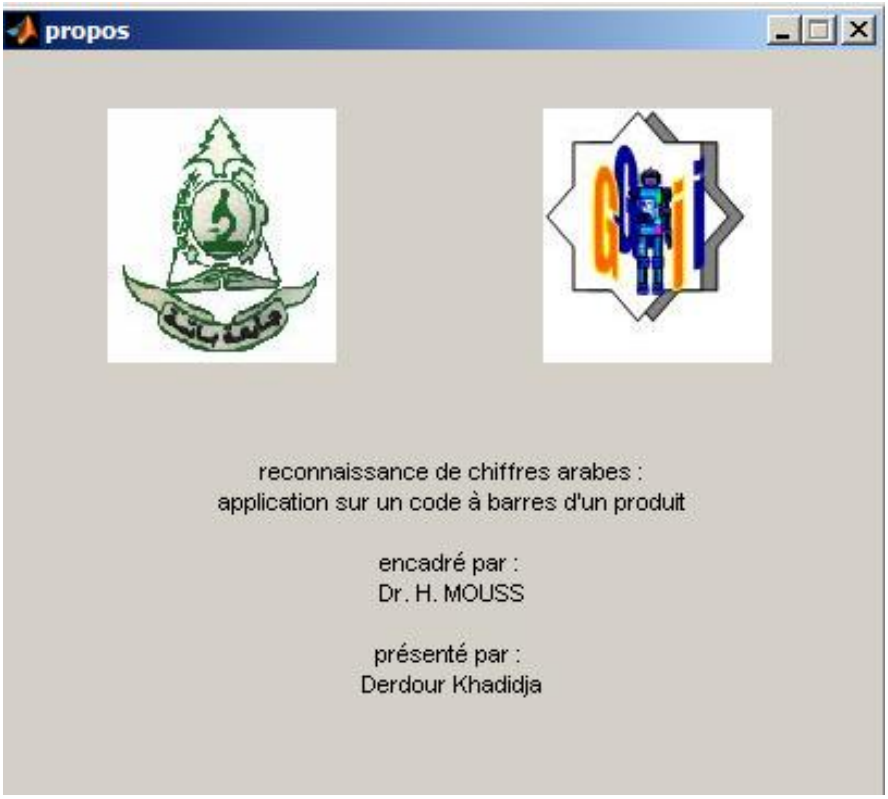

application
K_ppv / K_ppv flou
Résau Neurones
Code barres
Test
Aide
Quiter
Kppv / Kppv flou
Reseau de Neurones
hu_Invariance
Zernike_Invariance
Reconnaissance de chiffres arabes imprimés
application sur les code à barres


listimg1
Choisir une image :
timesrom.bmp
0 1 2 3 4
5 6 7
8 9
Kppv / KppvFlou


listimg3
Choisir une image :
chif.bmp
1
2
3
4
5
Close
Perceptron
MLP


application
K_ppv / K_ppv flou
Résau Neurones
Code barres
Test
Aide
Quiter
Principe
A propos
Titre
Reconnaissance de chiffres arabes imprimés
application sur les code à barres


principe
Apprentissage
Modèle d'apprentissage
Prétraitement
Extraction des paramètres
Apprentissage
les paramètres doivent être discriminante le plus possible et répondent aux exigence suivantes :
Faible variance intra classe, grande variance inter classe, faible nombre d'attributs et l'indépendance géométrique.
Reconnaissance
Modèle d'apprentissage
Prétraitement
Extraction des paramètres
Classification
l'acquisition d'une image de chiffres pour la validation (test).
L'acquisition de cette image se fait par le logiciel PaintBrush .


propos
reconnaissance de chiffres arabes :
application sur un code à barres d'un produit
encadré par :
Dr. H. MOUSS
présenté par :
Derdour Khadidja