

THESE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE

Discipline : Electronique

Présentée par

Stephane Lecoeuche

Titre :

Reconnaissance de caractères industriels par application
d'un système de réseaux de neurones à boucle de rétroaction

Soutenue, le 20 Novembre 1998

JURY :

M. Attila Baskurt,
M. Léon-Claude Calvez,
M. Vincent Devlaminck,
M. Salah Maouche,
M. Jean Paul Dubus,



Rapporteur
Rapporteur
Examineur
Examineur
Directeur
de thèse

A ma mère

REMERCIEMENTS

Je dédie avant tout cette thèse à ma mère qui m'a donné les moyens, le courage et la volonté de poursuivre mes études. Je regrette de ne pouvoir lui en présenter aujourd'hui l'aboutissement mais, c'est vers elle qu'à présent mes pensées sont tournées.

J'exprime toute ma gratitude à :

Monsieur Attila Baskurt, Professeur à l'université de Lyon,
Monsieur Léon-Claude Calvez, Professeur à l'université de Brest,
pour avoir accepté d'être les rapporteurs de cette thèse.

Monsieur Vincent Devlaminck, Maître de conférences HDR à l'université de Lille1
Monsieur Salah Maouche, Professeur à l'université de Lille1
pour avoir accepté d'en être les examinateurs.

Je tenais à remercier chaleureusement **Monsieur Jean-Paul Dubus**, Professeur à l'université de Lille1, directeur de thèse, pour m'avoir accueilli au sein de son équipe, pour sa patience et pour ses conseils avisés à la fois scientifiques et techniques qui m'ont permis d'aboutir à la réalisation de la thèse.

Je remercie l'**Ecole d'Ingénieurs du Pas-de-Calais** et tout particulièrement la direction de l'école en les personnes de :

Monsieur Philippe Durand
Monsieur Jacques Parissaux
Monsieur Dominique de Rancourt
Monsieur Pascal Ryckelynck,

qui m'ont donné les moyens matériels et techniques pour mener à bien ce doctorat.

Je tenais à adresser mes remerciements à **Monsieur Sylvain Lalot** pour son écoute, ses conseils fort utiles lors de la rédaction du manuscrit et le temps qu'il a su me consacrer sans compter au long de ces trois années, à **Monsieur Denis Deguillemont**, pour ses conseils et son aide et sa participation à l'élaboration de la thèse, et bien évidemment à tout le personnel de l'école en citant plus particulièrement Hélène, Gérald et Pierre-Yves pour leur participation à ce manuscrit.

Merci à Christine pour avoir bien voulu lire et corriger cet écrit pourtant peu littéraire et m'avoir accordé trois ans de patience. Merci enfin à ma famille et belle famille pour leur soutien moral.

Introduction générale 1

Première Partie :
Introduction aux systèmes de reconnaissance de caractères

INTRODUCTION	3
I.1 GENERALITES SUR LA RECONNAISSANCE DE DOCUMENTS	3
I.2 NOTIONS TYPOGRAPHIQUES	5
I.3 STRUCTURE D'UN SYSTEME DE RECONNAISSANCE DE CARACTERES	8
I.3.1 L'ACQUISITION	9
I.3.1.1 NUMÉRISATION	9
I.3.1.2 PRÉTRAITEMENT	10
I.3.2 LA RECONNAISSANCE DE DOCUMENTS	10
I.3.2.1 SEGMENTATION DE DOCUMENTS	11
I.3.2.2 RECONNAISSANCE DE CARACTÈRES	11
I.3.2.2.1 Extraction de caractéristiques	11
I.3.2.2.2 Classification	11
I.3.2.3 FUSION DE DONNÉES	12
I.3.3 LES POST TRAITEMENTS	12
I.3.3.1 LA VÉRIFICATION CONTEXTUELLE	12
I.3.3.2 L'ÉTAGE DE SORTIE	13
I.4 EVOLUTION DES TECHNIQUES EN RECONNAISSANCE DE CARACTERES	13
CONCLUSION DE LA PREMIERE PARTIE	17

Deuxième Partie :
La classification

INTRODUCTION	18
II.1 LA CLASSIFICATION UTILISEE EN RECONNAISSANCE DE CARACTERES	18
II.1.1 LE MATRIX MATCHING	18
II.1.2 CLASSIFICATION DE CARACTÉRISTIQUES	21
II.1.2.1 LES TECHNIQUES STRUCTURELLES	22
II.1.2.2 LES TECHNIQUES SYNTAXIQUES	23
II.1.2.3 LES TECHNIQUES STOCHASTIQUES	23
II.1.2.4 LES TECHNIQUES STATISTIQUES	24
II.2 LES METHODES STATISTIQUES CLASSIQUES	25
II.2.1 CLASSIFICATION BAYÉSIENNE	27
II.2.1.1 RÈGLE DE BAYES	27
II.2.1.2 RÈGLE DE BAYES À COÛT MINIMUM	28
II.2.2 MÉTHODES NON-PARAMÉTRIQUES	29
II.2.2.1 ALGORITHME DES K-PLUS PROCHES VOISINS.	30
II.2.2.2 ALGORITHME DES FENÊTRES DE PARZEN	31
II.2.3 MÉTHODES DES NUÉES DYNAMIQUES	32
II.3 LES METHODES NEURONALES	33
II.3.1 CONSTITUTION DES BASES D'ÉVALUATION	34
II.3.2 LE PERCEPTRON MULTICOUCHE OU PMC	35
II.3.3 LE RÉSEAU DE HAMMING	38
II.3.4 RÉSEAUX À FONCTIONS DE BASE RADIALE OU RBF	39
II.3.5 RÉSEAU DE NEURONES PROBABILISTES OU PNN	42
II.3.6 LES CARTES TOPOLOGIQUES AUTOORGANISATRICES OU SOM	43
II.3.7 LE LEARNING VECTOR QUANTIZATION OU LVQ	45
II.3.7.1 VECTOR QUANTIZATION (VQ)	45
II.3.7.2 LEARNING VECTOR QUANTIZATION (LVQ)	46
CONCLUSION DE LA DEUXIEME PARTIE	47

Troisième Partie :
L'extraction de caractéristiques

INTRODUCTION	50
III.1 EXTRACTION DES CARACTERISTIQUES EN OCR	52
III.1.1 GÉNÉRALITÉS SUR L'EXTRACTION DE CARACTÉRISTIQUES	52
III.1.2 EXEMPLES DE CARACTÉRISTIQUES LIÉES AUX CARACTÈRES	54
III.1.2.1 CARACTÉRISTIQUES LOCALES	54
III.1.2.1.1 Intersections avec des droites	54
III.1.2.1.2 Paramètres d'allongement	55
III.1.2.1.3 Mesure de densité spatiale	55
III.1.2.2 CARACTÉRISTIQUES GÉOMÉTRIQUES ET TOPOLOGIQUES	55
III.1.2.2.1 Détection de particularités locales	56
III.1.2.2.2. Calcul des pentes et des variations angulaires par polygonisation	56
III.1.2.2.3 Analyse de concavités, de convexités et de boucles	56
III.2 RECONNAISSANCE INVARIANTE PAR TRANSFORMATIONS GLOBALES	57
III.2.1 DÉVELOPPEMENTS EN SÉRIE	57
III.2.2 LES INVARIANTS INTÉGRAUX	58
III.2.2.1 LA TRANSFORMÉE DE FOURIER	58
III.2.2.2 LA TRANSFORMÉE DE MELLIN	58
III.2.2.3 LA CORRÉLATION CROISÉE	59
III.2.3 LES INVARIANTS ALGÈBRIQUES	59
III.3 METHODES NEURONALES ET RECONNAISSANCE INVARIANTE	61
III.3.1 RÔLE DE L' APPRENTISSAGE	61
III.3.2 COMBINAISON AVEC DES TECHNIQUES INVARIANTES CLASSIQUES	63
III.3.2.1 LE MODÈLE DE YÛCEER	63
III.3.3 LES ARCHITECTURES SPÉCIALISÉES	65
III.3.3.1 LE NEOCOGNITRON	65
III.3.3.2 LE MODÈLE DE WIDROW	67
III.3.3.3 LE HONN : HIGHER-ORDER NEURAL NETWORK	70
III.3.3.4 LE MODÈLE DE WANG	72
CONCLUSION DE LA TROISIEME PARTIE	74

Quatrième partie :
Etude d'un système de reconnaissance de caractères industriels

INTRODUCTION	76
IV.1 PRESENTATION DU PROBLEME POSE	76
IV.1.1 LA RECONNAISSANCE DE CARACTÈRES EN MILIEU INDUSTRIEL	76
IV.1.2 INTRODUCTION AU MARQUAGE INDUSTRIEL	78
IV.1.3 NOTIONS DE FONTES INDUSTRIELLES	78
IV.1.3.1 LA NORME OCR-A	79
IV.1.3.2 LA NORME OCR-B	79
IV.1.3.3 LES NORMES POUR LE CODAGE MICR	79
IV.1.4 PRÉSENTATION DU PROBLÈME	80
IV.1.5 PROPOSITION D'UN SYSTÈME	82
IV.2 LE PREMIER RESEAU	84
IV.2.1 OPÉRATIONS DE VISION : LISSAGE ET SEUILLAGE	85
IV.2.2 OPÉRATIONS DE NORMALISATIONS EN POSITION, EN TAILLE ET EN ORIENTATION	93
IV.2.2.1 INVARIANCE EN POSITION	94
IV.2.2.2 CARACTÉRISTIQUES EN TAILLE ET EN ORIENTATION	96
IV.2.2.3 MISE À L'ÉCHELLE OU NORMALISATION	99
IV.3 LE DEUXIEME RESEAU	103
IV.3.1 CONSTRUCTION DU CLASSIFIEUR	103
IV.3.1.1 DÉTERMINATION DE LA TAILLE DE LA COUCHE D'ENTRÉE	106
IV.3.2 MODIFICATIONS APPORTÉES AU RÉSEAU RBF	111
IV.3.2.1 INTRODUCTION DE LA NOTION DE FORME	111
IV.3.2.2 MODIFICATIONS APPORTÉES À LA COUCHE PATTERN	112
IV.3.2.3 MODIFICATION DE LA RÈGLE D'APPRENTISSAGE	113
IV.3.2.4 PERFORMANCES LIÉES À L'AMÉLIORATION	114
IV.4 LA PHASE DE RETOUR : LA COOPERATION	116
CONCLUSION DE LA QUATRIEME PARTIE	119

Cinquième partie :
Validations expérimentales

INTRODUCTION	120
V.1 PERFORMANCES DU SYSTEME	120
V.1.1 CONDITIONS D'EXPÉRIMENTATION	120
V.1.2 PERFORMANCES DU PREMIER RÉSEAU	120
V.1.3 EXPÉRIMENTATION DU SECOND RÉSEAU	126
V.1.4 EXPÉRIMENTATION DE LA BOUCLE DE RETOUR	129
V.1.5 PERFORMANCES GLOBALES	130
V.2 TENTATIVE DE COMPARAISON AVEC DES SYSTEMES COMMERCIALISES	131
V.3 DISCUSSION AUTOUR DE L'IMPLEMENTATION MATERIELLE	132
CONCLUSION DE LA CINQUIEME PARTIE	134

Conclusion générale	135
---------------------	-----

Bibliographie	136
---------------	-----

Annexe A	150
----------	-----

Annexe B	155
----------	-----

Annexe C	157
----------	-----

Annexe D	159
----------	-----

Introduction générale

Aujourd'hui, les besoins en traçabilité définis par les nouvelles normes ISO9000 [ISO9001], imposent aux entreprises de maîtriser parfaitement le process de fabrication. Dans cette optique, il est nécessaire de suivre précisément un article durant toute sa chaîne de fabrication. Cette traçabilité ne peut être réalisée que grâce à des solutions de marquage et d'identification. Trois principales solutions sont actuellement exploitées : les codes à barres, les data matrix et l'utilisation de caractères d'imprimerie.

Bien que l'identification automatique soit le plus souvent réalisée par l'application de systèmes de type code à barres et data matrix, la solution d'identification à l'aide de caractères d'imprimerie est a priori la plus intéressante. Les caractères ont l'avantage d'être directement compréhensibles par l'homme. Les informations contenues dans un seul marquage par caractères peuvent également être plus importantes (ex : date de fabrication, lot, référence, identifiant, regroupés lors d'un même marquage). Par comparaison, pour la même surface utilisée le codage par caractères contient six fois plus d'informations que le code à barres [AAS97].

La reconnaissance de caractères ou OCR (Optical Character Recognition) est une chose aisée dans l'environnement bureautique, mais, en milieu industriel, elle reste difficile à réaliser les difficultés étant nombreuses. Outre celles liées aux applications d'OCR bureautiques (la variété de fontes et la segmentation des caractères à reconnaître), les difficultés sont essentiellement dues à des problèmes d'acquisition et de lisibilité des caractères. En effet, les problèmes rencontrés dans la tâche de reconnaissance proviennent de la variété des marquages rencontrés, de la qualité des supports et des variations lumineuses lors de l'acquisition.

Notre travail a consisté à définir les concepts d'un système de reconnaissance qui pourrait s'appliquer au milieu industriel et disposant des caractéristiques suivantes :

- L'autoadaptation de l'acquisition aux conditions de prises d'images ainsi qu'au type de marquage et au type de support rencontrés.
- L'invariance en position, en taille et, d'une certaine manière, en orientation du caractère sélectionné. Ces propriétés sont obtenues par traitement permettant l'extraction d'un modèle invariant du caractère.
- Une reconnaissance multifonte obtenue par l'utilisation d'un classifieur de type supervisé.

Introduction générale

Etant donné les développements actuels des techniques neuronales et leurs aptitudes à résoudre des problèmes de reconnaissance de forme, nous nous sommes volontairement orientés vers les réseaux de neurones.

Dans une première partie, nous rappelons les notions essentielles liées à la reconnaissance de caractères. Après avoir rapidement présenté la reconnaissance de documents à travers son historique et ses domaines d'application, nous aboutissons à une revue de l'état de l'art des techniques utilisées. Nous décrivons l'architecture générale d'un système de reconnaissance et principalement les techniques utilisées pour les fonctions d'extraction de caractéristiques et de classification qui composent cette architecture. Nous montrons les limites actuelles de ces techniques en terme de performances et de domaines d'applications.

Dans une deuxième partie, nous introduisons les notions de classification utilisées pour la reconnaissance de caractères. Quelques techniques classiques de classification statistique sont décrites afin de permettre d'aborder les techniques neuronales. Nous présentons ensuite les principales architectures neuronales capables d'effectuer une tâche de classification.

Dans une troisième partie, nous définissons la notion de reconnaissance invariante. Nous rappelons les principales techniques d'extraction de caractéristiques invariantes connues et décrivons les techniques neuronales, les plus marquantes, applicables à cette extraction.

Dans une quatrième partie, après avoir redéfini plus précisément la problématique de notre sujet, nous présentons notre système de reconnaissance de caractères. Ce dernier est constitué d'un assemblage de deux réseaux de neurones et d'une boucle de rétroaction. Le premier réseau obtient le modèle binarisé d'un caractère en permettant une adaptation aux conditions d'acquisition (luminosité, marquage, support) et une invariance en position, en taille et en orientation. Le second réseau effectue la classification de cette représentation invariante. Une boucle de rétroaction adapte, à partir des résultats de la classification, les paramètres du premier réseau pour optimiser la reconnaissance.

Dans une cinquième partie, nous présentons les performances du système appliqué à différents types de caractères. Nous déterminons ainsi ses limites actuelles et précisons ses performances en comparaison avec celles de systèmes existants utilisables dans des conditions analogues.

Première Partie :

Introduction aux systèmes de
reconnaissance de caractères

Introduction

La reconnaissance de caractères n'est pas une préoccupation récente. En effet, diverses recherches ont été menées sur ce sujet depuis le début du siècle et ont trouvé nombre d'applications. Il est donc important, avant d'exposer et de situer notre étude, dont le but est de définir un système de reconnaissance en milieu industriel, de consacrer une partie à l'approche de cette question.

Le premier chapitre va permettre d'introduire les éléments essentiels concernant la reconnaissance de caractères et les systèmes d'OCR (Optical Character Recognition). Nous présenterons, dans le premier paragraphe, de manière générale la reconnaissance de documents à travers l'historique puis préciserons les domaines d'applications de l'OCR. Nous aborderons les thèmes de recherche actuelle en reconnaissance de caractères en présentant les limites reconnues des systèmes actuels qui sont : la reconnaissance manuscrite omniscriteur, la reconnaissance de documents bruités ou dégradés et la reconnaissance sur des supports spécifiques. Après un deuxième paragraphe énonçant très brièvement les notions liées à la typographie et aux polices et mettant en évidence les contraintes de la reconnaissance multiforme, nous énumérerons dans le troisième paragraphe les concepts utilisés pour concevoir un système de reconnaissance complet.

Nous présenterons, dans le dernier paragraphe de cette première partie l'état de l'art actuel des techniques de reconnaissance. Nous mettrons en évidence l'évolution des techniques utilisées et leurs performances en fonction de la complexité des difficultés auxquelles se sont heurtés les systèmes de reconnaissance. On aboutira aux tendances proposées pour résoudre les problèmes actuels.

I.1 Généralités sur la reconnaissance de documents

La reconnaissance d'écriture remonte au début du siècle. Le point de départ fut la nécessité de créer une machine de lecture venant en aide aux non-voyants. Le premier système permettant cette reconnaissance fut développé par Tyurin mais il fallut attendre les années 1940 pour voir apparaître la première version informatique de ce système [MAN86]. C'est en 1950 que les premières versions commerciales virent le jour [GOV90]. Cependant, bien qu'un grand nombre d'algorithmes aient été développés durant cette période, les machines de l'époque ne permirent que la réalisation de systèmes peu coûteux en temps de calcul, et basés sur la reconnaissance de caractères d'imprimerie monofont. Après un délaissement de quelques années dû à l'apparition des premiers systèmes de reconnaissance vocale, la reconnaissance de caractères a connu un engouement spectaculaire depuis ces dernières années tant dans le milieu universitaire que dans le milieu industriel.

C'est ainsi que grâce au développement de machines avec des puissances de calcul beaucoup plus importantes, des systèmes plus complexes ont pu être élaborés. Les applications se sont multipliées autour des années 1975. Une enquête a dénombré 45 lecteurs de caractères aux Etats-Unis à cette époque [SCH82]. Les recherches furent étendues à la reconnaissance multiforme puis plus récemment à une reconnaissance omniforme. En 1986, les premiers logiciels de reconnaissance multiforme apparurent et l'évolution récente des techniques a permis l'utilisation des logiciels de reconnaissance sur des caractères manuscrits [MAI93].

Actuellement des centaines de systèmes de reconnaissance sont commercialisés. Ils sont regroupés en deux catégories : les lecteurs spécifiques à un type de documents : lecture de chèques, d'enveloppes, de formulaires,... et les lecteurs polyvalents : bureautique, presse,

documentation technique, archivage [SRI92a]. Actuellement, la plupart de ces systèmes sont omnifontes.

Cette multiplication des applications de reconnaissance de caractères est liée à la volonté de limiter la quantité et le volume des documents à stocker et ainsi de faciliter leur traitement. En effet, la reconnaissance de caractères évite toutes les phases de codage manuel réalisées par des opérateurs en utilisant le clavier comme interface. Elle aura donc pour effet d'économiser du temps, de simplifier une tâche fastidieuse et de limiter la quantité de papier utilisée [LUB95]. Ainsi le premier but des sociétés qui se sont équipées en logiciels d'OCR fut d'automatiser tout traitement répétitif comme la saisie de formulaires, le traitement des chèques, le tri du courrier, etc... Le second but recherché par ces sociétés était de limiter le stockage de documents puis de faciliter leur recherche. L'OCR fut très profitable pour le fonctionnement des centres d'information tels que les bibliothèques. Actuellement, l'utilisation du réseau Internet et donc d'une communication informatisée rend encore plus nécessaire le passage de l'information papier vers l'information numérique.

Les principales applications déduites de la lecture de documents par des machines sont :

➤ L'aide à la lecture pour les non-voyants

Les systèmes de reconnaissance associés à des synthétiseurs vocaux permettent la compréhension de documents et livres pour les aveugles.

➤ La saisie automatique de document [OGI94][DES95]

La reconnaissance de caractères permet un traitement automatique de pages d'écriture. De nombreux systèmes ont été développés pour : la lecture de cartes ou plans cadastraux, la lecture des fax et l'envoi de courrier électronique.

➤ La lecture des tickets de transport aérien

Chaque place réservée nécessite trois enregistrements : un auprès de la compagnie, un auprès de l'agence de voyage et un pour le voyageur. Afin de limiter le grand nombre de billets ainsi créés et d'éviter l'attente avant embarquement, de nombreuses compagnies ont recours à un système d'identification automatique qui lit le ticket et compare les indications avec la base de données de chaque vol.

➤ La lecture de formulaires [WRI90]

De nombreuses enquêtes ou fiches de renseignements utilisent des formulaires préimprimés. L'utilisation d'un système de reconnaissance, capable de lire directement les données dans les zones réservées permet d'effectuer rapidement la saisie de ces documents.

➤ La lecture des passeports

Certaines douanes sont équipées de lecture de passeports afin d'identifier chaque voyageur. Le système permet de lire le nom, la nationalité, le numéro de passeport et aussi de contrôler directement auprès des bases de données des services d'immigration, l'autorisation de séjour.

➤ La gestion automatique des chèques bancaires ou postaux [PLA89][LET95][SIM98]

Les chèques sont automatiquement traités grâce à lecture automatique du montant en chiffres et en lettres. L'utilisation d'un système lisant les deux montants réduit les risques d'erreur. Cependant, les systèmes les plus récents ne garantissent que 70% de reconnaissance en utilisant le montant en chiffres et en lettres. Pour la sécurité, les signatures peuvent être vérifiées également.

➤ L'acheminement automatique du courrier [HUL84][AUG93][HEU94]

Le système permet la lecture puis l'acheminement des enveloppes en utilisant à la fois le code postal et l'adresse de destination. De nombreux services postaux ont développé des collaborations avec des universités pour la création d'une trieuse automatique de courrier. C'est avec la gestion automatique des chèques, l'une des applications industrielles les plus importantes. C'est aussi l'une des plus difficiles à réaliser. En effet, les données à traiter étant essentiellement du type manuscrites, le taux de réussite d'un tel système est encore actuellement très bas.

➤ La presse [ARC95]

Certains groupes de presse ont cherché à développer les systèmes de reconnaissance de caractères afin d'imprimer directement les journaux à partir des documents dactylographiés des journalistes. Dès 1965, l'éditeur américain Perry publication a utilisé un tel système.

➤ Le contrôle du processus de fabrication [GOK86][COO90][MAT92][ATH96]

Des systèmes de lecture sont utilisés dans les manufactures textiles, dans l'industrie métallurgique. Ils permettent d'informer un superviseur sur l'état de la production. On améliore ainsi la traçabilité, la gestion des stocks, le tri de colis...

Dans le cadre de la reconnaissance de caractères imprimés, les systèmes sont confrontés à la notion de fontes. En effet, l'évolution de la typographie a permis l'introduction d'un très grand nombre de polices de caractères. Les systèmes de reconnaissance sont classés en trois catégories :

On distingue ainsi :

- des systèmes monoscripteurs qui ne sont adaptés qu'à un type de fontes.
- des systèmes pluriscripteurs qui ne sont adaptés qu'à un certain nombre de fontes.
- des systèmes omniscriteurs qui sont adaptés à tous les types de fontes.

Ainsi, le développement d'un système de reconnaissance nécessite la connaissance des polices dont il doit assurer la reconnaissance.

Le processus de reconnaissance est souvent basé sur l'étude des caractéristiques propres aux caractères. Ces caractéristiques sont dépendantes de la typographie utilisée. Il est donc nécessaire de caractériser chaque caractère selon le type de fontes utilisé afin d'élaborer un système de reconnaissance.

I.2 Notions typographiques

Le terme "typographie" regroupe l'ensemble des techniques et des procédés permettant de reproduire des textes par l'assemblage de caractères en relief (définition du dictionnaire petit Robert). Le principe de la typographie est apparu au XV^{ème} siècle et est quasiment resté inchangé depuis [PHI98]. Avant 1450, les seules reproductions de livres étaient le fruit de scribes. Le premier livre imprimé fut la Bible de Gutenberg. C'est grâce à l'utilisation, par Fust et Schoeffer, des techniques développées par Gutenberg que cet ouvrage put être produit en gros volume. Le principe de la typographie consiste à utiliser des poinçons d'acier représentant l'image miroir du type (ou caractère) pour marquer une page de métal tendre. Cette page est ensuite recouverte d'encre et pressée sur le papier. En développant sa technique, Gutenberg fut amené à créer la première police de caractères appelée "blackletter". Cette unique police fut utilisée pendant près d'un siècle. Ce n'est qu'au XVII^{ème} siècle que les principales polices sont apparues.

Le nom de fonte est issu de la façon de fabriquer les poinçons d'acier qu'utilisaient les typographes dans l'imprimerie. Cette notion, synonyme de police, définit l'ensemble des caractères d'un même type (fondus ensemble). Ainsi, la police est essentiellement caractérisée par la nature des empattements utilisés et le dessin utilisé pour les caractères. Chaque caractère de la police est défini par sa hauteur, son corps (qui est défini par la hauteur plus l'espace interligne), sa largeur et sa chasse définie par la largeur et l'espace inter-caractères [BEL92] (figure I.1).

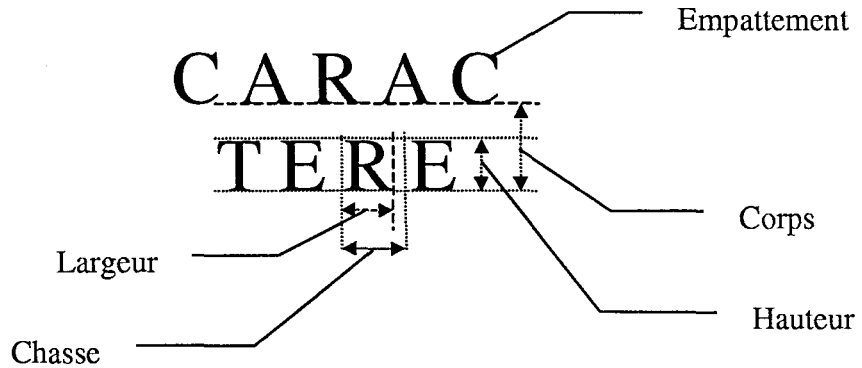


Figure I.1 : Caractéristiques d'une fonte

Les unités utilisées en typographie pour les mesures dimensionnelles sont le point Didot pour les français (1 point Didot = 0.3759 mm) et le point PICA pour les anglais (=0.35277 mm). La typographie pouvant par extension définir la mise en forme de documents, une normalisation de la taille des caractères est apparue en fonction des différentes zones du document (figure I.2).

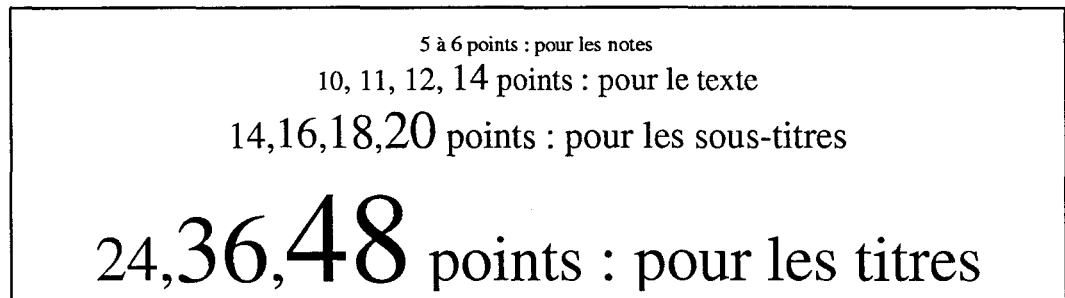


Figure I. 2: Echelle des corps en point Didot

Le dessin représente la forme et l'épaisseur du caractère. L'épaisseur, ou graisse, peut varier en fonction de la taille du caractère variant du maigre, au gras jusqu'au noir. La forme appelée aussi casse, décrit différents caractères de même type (figure I.3).

Police capitale	ABCDEF	Police capitale gras	ABCDEF
Police bas de casse	Abcdef	Police capitale gras italique	<i>ABCDEF</i>
Police petite capitale	ABCDEF	Police capitale noir	ABCDEF
Police capitale italique	<i>ABCDEF</i>	Police capitale condensée	ABCDEF

Figure I.3 : Variations du dessin

Il existe deux grandes classifications françaises des caractères d'imprimerie : la classification Thibaudau et la classification Vox [BEL92]. Ces classifications sont basées sur la différenciation des empattements.

- La classification Thibaudau réalisée en 1920 par le typographe Francis Thibaudau distingue 4 grandes familles de caractères :
 - ❖ Les Didots ayant des empattements fins et filiformes.
 - ❖ Les Elzéviens ayant des empattements triangulaires.
 - ❖ Les Egyptiennes ayant des empattements quadrangulaires.
 - ❖ Les Antiques n'ayant pas d'empattements.

Didot	ABCD	Elzevir	ABCD
Egypte	ABCD	Antique	ABCD

Figure I.4 : Classification Thibaudau

- La classification Vox, réalisée en 1950 par le typographe Maximilien Vox, distingue 9 grandes familles de caractères didones : galades, humaines, incisives, linéales, manuscrites, mécaniques, réelles et scriptes. Cette distinction est basée sur l'évolution de la typographie.

Actuellement, ces classifications ne sont plus suffisantes. En effet, l'informatique et l'apparition des traitements de texte ont permis d'augmenter le nombre et la variété des polices de caractères. De nombreux logiciels permettent à chaque utilisateur de définir sa propre police de caractères. Les polices sont maintenant définies de manière vectorielle, représentation mathématique des contours des polices : ce sont les polices vectorielles TRUE TYPE. En outre, de plus en plus, les nouveaux types de polices développés sont des polices fantaisistes ou des polices imitant l'écriture manuscrite (figure I.5).

Airbus special	ABCDEF	Maiandra	ABCDEF
Lucidia	<i>ABCDEF</i>	Goudy	ABCDEF
Flexure	ABCDEF	cmb	<i>ABCDEF</i>
Comic	ABCDEF	Harrington	<i>ABCDEF</i>

Figure I.5 : Exemples de polices True Type

Concernant les systèmes d'OCR, des comités de standardisation ont tenté de définir des normes de caractères. Ont été ainsi normalisées deux polices de caractères : OCR-A et OCR-B (figures I.6). La première a été définie en 1960 par l'ANSI (American National Standards Institute) et la seconde en 1965 par l'ECMA (European Computer Manufacturers Association). Cette normalisation a vite été abandonnée étant donnée la variété de fontes actuellement définies.

0123456789-&#
ABCDEFGHIJKLM
NOPQRSTUVWXYZ

Figure I.6a : Norme OCR-A

0123456789-&#
ABCDEFGHIJKLM
NOPQRSTUVWXYZ

figure I.6b : Norme OCR-B

De plus, un ensemble de contraintes, définies par les normes, concernant les propriétés optiques du papier (réflectance, pureté, opacité), l'encre utilisée et les qualités d'impression (couleur noire obligatoire, régularité et largeur des traits), ainsi que les dimensions des caractères ont conduit à abandonner ces deux normes pour les systèmes standard et bureautiques. Seules les caractéristiques typographiques ont été conservées et restent en vigueur pour les systèmes de marquage industriel et les systèmes de reconnaissance associés.

I.3 Structure d'un système de reconnaissance de caractères

Un système de reconnaissance doit être capable d'effectuer la lecture d'un document à partir de son image. La structure d'un système d'OCR comporte trois parties principales (figure I.7) : l'acquisition et traitement d'image, l'analyse du document puis l'interface de sortie vers l'environnement [REM92].

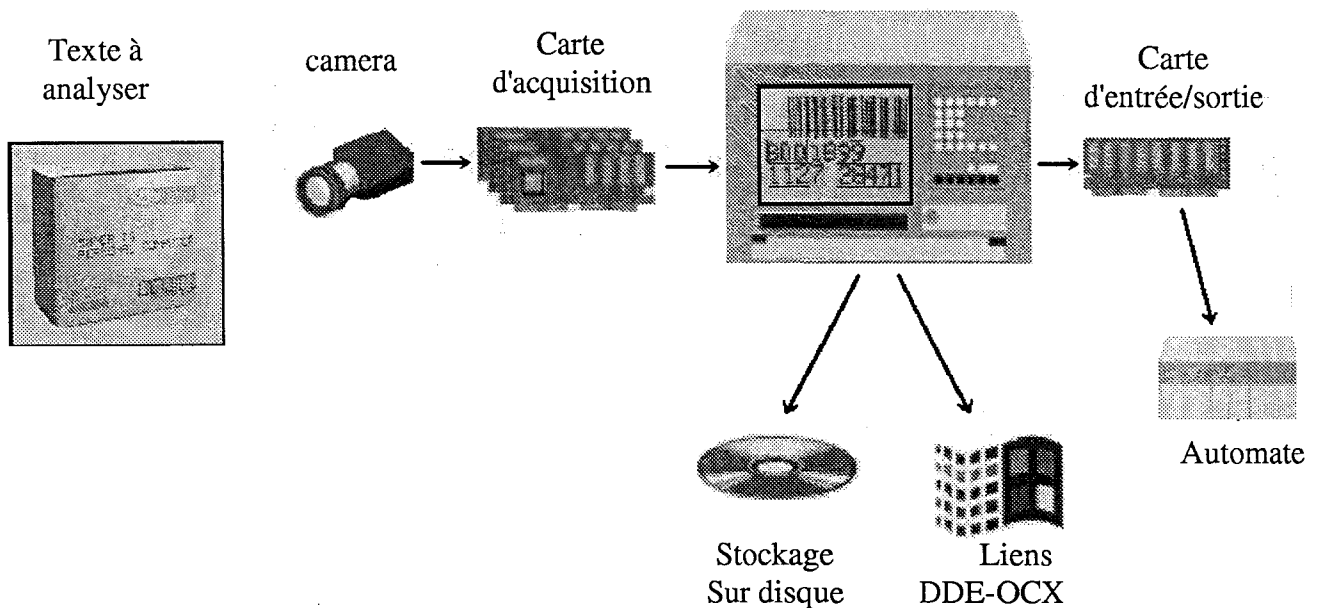


Figure I.7 : Exemple de système de reconnaissance de caractères

Chacune de ces parties peut se décomposer en plusieurs opérations [SRI92a][MAI93] (figure I.8).

La première opération consiste à effectuer l'acquisition du texte afin de disposer d'une image numérique, et il est alors nécessaire de corriger les imperfections dues à la saisie au moyen de prétraitements d'image.

L'étape de reconnaissance du document est l'étape principale. Trois opérations sont effectuées : à partir du document, on isole les caractères, on en effectue la reconnaissance, puis, si nécessaire, on ajoute un étage de fusion de données afin d'améliorer la qualité de la reconnaissance.

La dernière étape concerne les post-traitements. La vérification des résultats permet de contrôler leur qualité en s'assurant de la cohérence des noms trouvés puis des phrases. Lors de la reconstruction du document, des traitements finaux d'ordre lexical, sémantique et syntaxique permettent de détecter ces incohérences. L'étage de sortie permet de mettre en forme les résultats afin de donner lieu à leur exploitation.

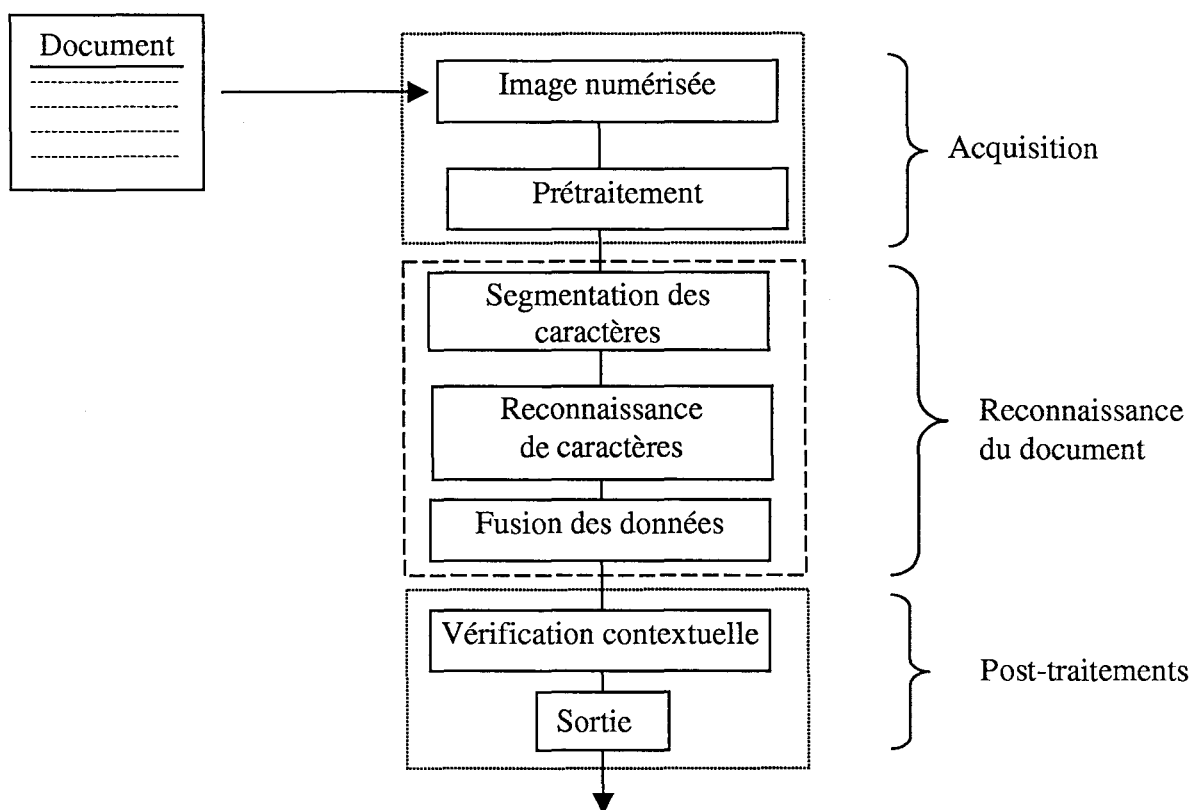


Figure I.8 : Synoptique d'un système de reconnaissance

I.3.1 L'acquisition

I.3.1.1 Numérisation

La première opération consiste à effectuer l'acquisition du document. Cette opération consiste à digitaliser le document à l'aide d'un scanner ou d'une caméra vidéo ou CCD. Il est

pour cela nécessaire d'utiliser éclairage, objectif et détecteur et d'effectuer un balayage du document.

La qualité de l'acquisition est le premier facteur essentiel conditionnant la qualité de reconnaissance. Cette qualité est fonction de l'éclairage et de la résolution utilisés pour capturer l'image du texte. Le type d'éclairage choisi doit permettre de mettre facilement en évidence le texte du document (les caractères de la trame du papier) et la résolution doit permettre de distinguer plus facilement les différentes classes de caractères.

Pour pouvoir effectuer une reconnaissance de caractères correcte, certains systèmes imposent des contraintes sur la qualité de l'impression et donc sur l'encre et sur le papier. Des normes ont été définies [ISO1831] sur la nature du papier et de l'encre. Ainsi, le papier doit réfléchir un maximum de lumière : facteur de réflectance ; la brillance doit être faible et il doit être homogène (fibre papier, contamination,...). En outre, l'encre doit permettre un très bon contraste, la densité d'encrage doit être bonne.

La résolution de l'image doit être suffisamment élevée pour détecter un maximum d'informations dans l'image. Les résolutions de 300 dpi permettent d'obtenir le meilleur compromis qualité / taille de l'image. Cependant, cette résolution peut être choisie en fonction du type de reconnaissance utilisé. Par exemple, une reconnaissance par extraction de caractéristiques (notion qui sera décrite dans la 3^{ème} partie) se basant sur des notions géométriques (présence ou absence de courbes de traits,...) peut utiliser une résolution moindre que pour une reconnaissance par " template matching" (détaillée en 2^{ème} partie) se basant sur une comparaison de toute l'image.

Dans le cas de l'écriture manuscrite, une tablette graphique peut aussi être utilisée pour la reconnaissance. Cette tablette sert d'interface entre le scripteur et le système de reconnaissance et permet d'obtenir des informations supplémentaires relatives à l'écriture : sens de déplacement, lever/poser de crayon. Dans ce cas, les systèmes sont définis comme systèmes de reconnaissance en ligne.

Pour notre étude, nous ne nous intéresserons qu'aux systèmes de reconnaissance hors ligne, les systèmes de reconnaissance en ligne étant surtout orientés vers l'interface homme-ordinateur et ayant comme but, la suppression du clavier.

I.3.1.2 Prétraitement

Lors de la saisie du document, des déformations dues à la chaîne d'acquisition peuvent intervenir. De nombreux facteurs entrent en compte : la qualité des objectifs, la chaîne de transmission et la numérisation. Les premières opérations effectuées consistent à "nettoyer" l'image initiale. Il s'agit d'éliminer les déformations induites par l'acquisition, de concentrer et de localiser la représentation du caractère.

I.3.2 La reconnaissance de documents

Cette phase est décomposée en trois opérations [BOK92]. La première, la segmentation de documents, consiste à extraire individuellement les caractères du document afin de permettre, dans une deuxième étape d'effectuer la reconnaissance de caractères. Une troisième étape basée sur la fusion de données peut être ajoutée afin d'augmenter les chances d'effectuer une reconnaissance correcte.

I.3.2.1 Segmentation de documents

Cette opération de segmentation consiste à décomposer, tout d'abord, le document en lignes puis chaque ligne en mots et enfin à segmenter chaque mot en lettres. Pour les systèmes de reconnaissance utilisant le mot comme support, cette dernière segmentation n'est pas réalisée. A l'issue de l'étape de segmentation, l'image originelle est décomposée en une multitude d'images correspondant soit aux mots, soit aux caractères.

I.3.2.2 Reconnaissance de caractères

Le processus de reconnaissance de caractères est le plus complexe. Généralement, il comporte deux phases : l'extraction de caractéristiques et la classification. L'extraction de caractéristiques permet de déterminer un vecteur dont les composantes caractérisent chaque type de caractère. La classification va permettre de déterminer la classe d'appartenance du caractère à l'aide de ce vecteur de caractéristiques.

I.3.2.2.1 Extraction de caractéristiques

Le processus d'extraction consistant à représenter un caractère par un vecteur de caractéristiques, le codage le plus élémentaire consiste à construire un vecteur constitué d'autant de composantes qu'il y a de pixels dans l'image. Leur niveau de gris définit alors le codage de ce vecteur. Afin d'éviter l'utilisation de vecteurs de taille trop importante, les techniques d'extraction de caractéristiques cherchent à définir un codage, déduit d'un ensemble de mesures, qui distingue le mieux les différents types de caractères.

Ces mesures doivent être les plus génériques possible pour ne pas dépendre des polices utilisées mais aussi suffisamment précises pour identifier chaque caractère. Les mesures sont effectuées à partir de la matrice image du caractère et représentent généralement des propriétés locales et/ou globales. Les caractéristiques obtenues tiennent compte des particularités métriques, statistiques ou topologiques. Les méthodes les plus courantes sont [GAI83][MIC84] : la détection de traits horizontaux et verticaux, le calcul d'intersection de lignes, le calcul de concavités, la détection et la localisation de boucles et les mesures liées à la dimension et à la surface du caractère.

I.3.2.2.2 Classification

La partie classification est généralement composée de deux phases : l'apprentissage des types de caractères à reconnaître et l'identification de caractères en rapport avec l'apprentissage effectué. Certaines techniques ont tenté d'utiliser un processus de reconnaissance de caractères sans apprentissage, mais aucune jusqu'à présent ne s'avère fiable [LET90] [JOD94b].

➤ L'apprentissage permet au système d'élaborer sa bibliothèque de caractéristiques. Avant de pouvoir effectuer la reconnaissance, le système doit apprendre les caractéristiques de chaque caractère et les garder en mémoire. La méthode d'apprentissage la plus triviale consiste à mémoriser différents vecteurs représentant chaque caractère présenté durant cette phase. L'apprentissage est dépendant du type d'écriture utilisé et des fontes utilisées. Ainsi, dans le cas de reconnaissance multiforme, chaque fonte doit être préalablement apprise par le système. Dans le cas d'écriture manuscrite, l'apprentissage consiste à référencer les caractéristiques obtenues pour chaque scripteur afin d'en dégager les constantes.

➤ L'apprentissage étant fait, l'identification consistera à déterminer, à partir de ces caractéristiques apprises, la classe d'appartenance du caractère présenté.

On référence trois grandes catégories d'approches utilisées pour la reconnaissance de caractères : l'approche statistique, l'approche structurelle et l'approche stochastique. Cette classification regroupe les techniques d'extraction de caractéristiques ainsi que des techniques de classification associées.

L'approche statistique ou approche globale va utiliser des mesures faites sur le caractère afin de le coder sous forme de vecteur. Ce vecteur définit le caractère dans un nouvel espace de représentation. Cette approche consiste à construire un classifieur permettant de distinguer les différentes classes à l'intérieur de cet espace de représentation.

L'approche structurelle est plus proche du fonctionnement du raisonnement humain. Elle consiste à ordonner hiérarchiquement les caractéristiques déduites des caractéristiques locales de la forme, sous forme d'un arbre de décision. L'identification s'effectuera par étapes successives.

L'approche stochastique est essentiellement utilisée pour la reconnaissance manuscrite. Elle permet de modéliser l'écriture en tenant compte des états rencontrés par le stylet. L'information temporelle et le sens du tracé vont permettre de coder l'écriture. Cependant, dans le cas de l'écriture imprimée, cette approche n'apporte que peu d'intérêt.

I.3.2.3 Fusion de données

Lorsqu'individuellement aucune des techniques choisies ne se révèle satisfaisante, une combinaison de ces techniques permet d'obtenir un meilleur résultat. Ce procédé constitue une des tendances actuelles. En effet, il permet d'utiliser la complémentarité des techniques associant ainsi leurs qualités. Cependant, la combinaison des techniques doit respecter des règles qui tiennent compte des spécificités de chacune. L'élaboration de ces règles est assumée par l'étape dite de fusion. On dénombre trois approches différentes en la matière [KIT98]:

➤ Une approche séquentielle qui consiste à appliquer successivement chaque technique en s'appuyant sur le résultat de la précédente.

➤ Une approche parallèle qui consiste à appliquer l'ensemble des techniques séparément puis à combiner les résultats dans une étape finale. Chaque technique de classification est appliquée aux mêmes données

➤ Une approche hybride qui combine approche séquentielle et approche parallèle.

I.3.3 Les post-traitements

I.3.3.1 La vérification contextuelle

Les résultats issus de la reconnaissance peuvent être améliorés à l'aide de post-traitements qui consistent à vérifier l'exactitude des résultats. Les contrôles sont effectués au niveau linguistique. Ils sont donc relatifs aux phrases et donc appliqués après reconstitution du texte. Trois niveaux de vérification sont possibles [HEU94] : la vérification syntaxique, la vérification lexicale et la vérification sémantique.

La vérification lexicale fut la première méthode utilisée. Il suffit de comparer les mots obtenus à un dictionnaire de référence. Afin d'optimiser la vérification, certains auteurs ont

développé la notion de sous-dictionnaire qui regroupe les mots par ensemble selon leur thématique, leur taille, etc... L'intérêt est bien sûr de limiter l'étendue du lexique. L'utilisation du dictionnaire permet de lever une partie des ambiguïtés possibles entre les caractères, d'éviter par exemple l'assimilation de O et 0, rn et m, 1 et l.

La vérification syntaxique se base sur l'utilisation de la grammaire, la construction de phrases. Elle consiste à détecter les séquences de mots qui n'ont pas d'usage dans la langue, par exemple : « Je m'en vais » ou « chatOn ».

La vérification sémantique est la vérification la plus évoluée, elle fait intervenir la notion de sens. Le système après avoir appris à lire, doit comprendre le sens de l'énoncé. Jusqu'alors peu de recherches ont été faites sur ce thème.

Dans le cadre de la lecture de chèques ou d'adresses postales, une vérification de redondance est faite. On vérifie la correspondance entre les chiffres (montant numérique, code postal) et les lettres (montant littéral, localité) [SIM98].

I.3.3.2 L'étape de sortie

Cette dernière étape permet de transmettre les résultats du système de reconnaissance vers un système de gestion de données. Dans le cadre d'une application bureautique, la sortie est un fichier, un tableur, une base de données. Dans le cadre du tri postal, de la lecture de chèques, le résultat est directement envoyé vers un système automatisé qui va permettre soit le routage, soit l'écriture bancaire.

Pour conclure, la structure d'un système de reconnaissance de caractères implique l'imbrication d'un grand nombre d'opérations. Au travers de chacune d'elles de nombreux algorithmes ont été développés pour la conception de systèmes OCR. Les phases les plus délicates et aussi les plus complexes étant celles liées à la reconnaissance des caractères, c'est-à-dire l'extraction de caractéristiques et la classification, de nombreuses techniques ont été développées afin d'en améliorer les performances et d'accroître la qualité des systèmes de reconnaissance [MAN86][GOV90]. Nous nous proposons donc à présent de dresser un panorama succinct de ces techniques développées au fil des années ainsi que de leurs adaptations successives.

I.4 Evolution des techniques en reconnaissance de caractères

Les premiers systèmes de reconnaissance de caractères furent fondés sur l'analyse directe de l'image. Cette approche, la plus simple, consiste à concevoir un classifieur basé directement sur la représentation rétinienne de l'image [REM85]. Cette méthode est aussi connue sous le terme de "reconnaissance analytique" ou "méthode matricielle". La technique employée est une comparaison de matrices, l'une correspondant à un modèle connu et l'autre correspondant au modèle à identifier. On applique tout d'abord l'ensemble des matrices connues sur le modèle. En analysant l'ensemble des coefficients de corrélations obtenus, on effectue alors le choix du type de caractère. Cependant, les limitations de cette technique apparaissent très vite. En effet, il est

rare d'obtenir une corrélation de 100% avec les modèles appris. En outre, le temps de traitement est très dépendant du nombre de caractères appris. Pour obtenir des temps de traitement raisonnables, le système ne peut disposer que de peu de caractères mémorisés. Mais surtout cette technique ne supporte aucune variation lumineuse et spatiale. Une amélioration de la méthode consiste à tolérer une déformation du modèle. Les techniques se regroupent sous le terme "elastic matching". Cette technique est encore couramment utilisée dans le cas de caractères parfaitement normalisés et sur des documents très simples.

Ainsi les restrictions imposées par les déformations des caractères limitent l'efficacité de ces méthodes. C'est pourquoi il s'est avéré nécessaire de développer des techniques de reconnaissance basées sur l'utilisation de caractéristiques extraites de l'image du caractère. Cette technique permet d'avoir un fonctionnement beaucoup plus rapide. Le principe est d'identifier et de distinguer chaque caractère par des caractéristiques qui lui sont propres (taille, nombre de traits, d'intersection, de boucles,...) et ainsi, plutôt que d'analyser une image complète, seule l'analyse de quelques mesures va permettre une classification. Dès les années 1980, plusieurs distinctions de ces méthodes ont été proposées [SUE80][GAI83]. Les principales catégories de méthodes rencontrées sont :

- Les transformations globales.
- Les techniques structurelles.
- Les caractéristiques extraites de la distribution des points.
- Les caractéristiques basées sur des critères topologiques et géométriques.

Parmi ces méthodes, les plus intéressantes présentent des propriétés d'invariance. Ainsi, les caractéristiques obtenues sont indépendantes de la position, de la taille et de l'orientation du caractère. Complémentaires aux développements de nouvelles techniques d'extraction de caractéristiques, de nouvelles techniques de classification ont dû être élaborées.

Faisant partie intégrante de la reconnaissance de formes, les principales techniques de classification ont été appliquées à la reconnaissance de caractères. Ces techniques d'analyse et de décision sont généralement relatives aux caractéristiques extraites.

Les premières techniques de classification sont issues de l'approche statistique qui utilise les caractéristiques globales de la forme. Cette approche a permis de mettre en œuvre des techniques simples [SHR87]. La décision bayésienne a également été appliquée à la reconnaissance de caractères. [FRE91] a proposé une technique de reconnaissance efficace à 99,7% pour des caractères de fontes simples. En 1996, une nouvelle méthode probabiliste baptisée "sn-tuples" (version probabiliste du classifieur n-tuple) [LAC96] améliore les performances en terme de taux de reconnaissance (97,6 % sur la base CEDAR de chiffres manuscrits) et temps de traitement (200 caractères par seconde).

L'approche syntaxique s'est développée à l'aide des techniques d'extraction de caractéristiques locales. En 1977, une technique de reconnaissance [CHE77] basée sur le langage PDL : Picture Description Language) a été appliquée pour les caractères de la police OCR-B. Le résultat obtenu est une reconnaissance entre 80 et 87 % sur 1280 exemples de caractères imprimés. Les méthodes de mise en correspondance avec des modèles décrivant la disposition spatiale des primitives morphologiques [MIT89] permettent d'obtenir des performances avoisinant les 88.9% de reconnaissance (10.1% de rejet et 1.0% d'erreur).

Les méthodes structurelles ont été déduites de l'analyse topologique des caractères. Elles sont regroupées sous le terme de " reconnaissance morphologique ". Belaïd [BEL92] utilise cette

approche pour la reconnaissance des caractères imprimés puis manuscrits. Il s'agit de structures de graphe : à partir de primitives très simples, on cherche à décrire le caractère. Cette approche nécessite l'utilisation d'un arbre de décision qui a été appliquée à la reconnaissance de caractères imprimés multifontes en 1985 [PAK85].

L'approche la plus tardive et la plus complexe fut l'approche stochastique basée essentiellement sur les chaînes de Markov. Un système d'analyse de cartes cadastrales a été proposé en 1991 [BOU91]. La reconnaissance de caractères imprimés multifontes a été traitée suivant cette approche en 1992 [ANI92] dans le cadre d'une reconnaissance de mots puis en 1993 cette approche a été étendue à la reconnaissance manuscrite [AVI93].

Cependant, la performance des systèmes de reconnaissance de caractères et plus généralement des formes doit beaucoup à l'apparition des réseaux de neurones artificiels. Ces derniers ayant un fonctionnement statistique se sont avérés être d'excellents classifieurs. Leur adaptation à la reconnaissance de caractères fut très aisée car ils utilisent directement les caractéristiques extraites (métriques [AUG93], topologiques [HEU94], structurelles [BUR91])

Les premières utilisations de réseaux de neurones en reconnaissance de caractères correspondent à des utilisations de réseaux de type mémoire associative. En 1988, une étude [PAW89], correspondant à un apprentissage de 391 chiffres et une évaluation sur 1173 chiffres, a comparé différents modèles de réseaux (Hopfield, perceptron, machine de Boltzman, mémoire associative,...) et a montré que la mémoire associative possède les meilleures performances avec cependant un taux de reconnaissance très moyen (74%). L'évolution des techniques neuronales a permis d'améliorer les taux de reconnaissance notamment grâce aux algorithmes d'apprentissage.

Ainsi, un réseau [STR89] basé sur l'apprentissage, par des critères statistiques, d'un ensemble de fonctions booléennes permet d'effectuer une reconnaissance à 79% avec 15% de rejet et 6% d'erreur. L'apparition de l'algorithme de rétropropagation (règle d'apprentissage très efficace) a permis d'améliorer les performances avec 92% de reconnaissance, 3% de rejet et 5.0% d'erreur [LEC89].

Aujourd'hui, de nombreuses études montrent que les techniques neuronales sont plus performantes pour la classification de caractères que les techniques statistiques classiques. Ainsi, les réseaux ont un taux d'erreur moins important de 4 fois pour des classifieurs basés sur des mesures de distance (Euclidienne, Normal) et 1,5 fois pour les techniques statistiques tel que les k-plus proches voisins [BLU94].

En marge des travaux énoncés, certains systèmes de reconnaissance ont été développés autour des techniques d'intelligence artificielle (systèmes multiexperts [SUE90]) ou d'algorithmes génétiques [MAR93]. Cependant, étant donné la complexité de la tâche de classification, ces techniques ne sont pas très adaptées et présentent des performances faibles : 93% et 83% sur des caractères manuscrits

A l'heure actuelle, la reconnaissance de caractères imprimés est devenue très banale. En effet, de nombreux logiciels de bureautique sont actuellement commercialisés avec des qualités de fonctionnement très élevées. Les applications en bureautique sont omniprésentes. Chaque scanner est commercialisé avec un logiciel d'OCR et pour une centaine de francs, on peut obtenir un logiciel ayant de très bonnes performances pour des documents imprimés sur support papier. Cet aboutissement a été rendu possible grâce aux efforts consacrés par la communauté scientifique à ce thème durant les années 80 et l'explosion de la micro-informatique a permis d'en accélérer la mise en œuvre.

De nos jours, les travaux concernant la reconnaissance de caractères sont essentiellement consacrés à la reconnaissance de caractères manuscrits. Les principaux moteurs de ces travaux sont les applications liées au tri postal et à la gestion des chèques. Mais bien que de nombreux systèmes équipent les services postaux et établissements bancaires, leur taux de reconnaissance demeure insuffisant.

En effet, l'écriture manuscrite est tellement diverse qu'elle impose un grand niveau de complexité des algorithmes de reconnaissance. La variété des scripteurs oblige à prendre en compte un grand nombre de caractéristiques pouvant varier fortement d'une écriture à l'autre. De plus, la qualité des documents manuscrits est souvent plus faible que celle des documents imprimés, la qualité du papier, la qualité de l'encre ou du crayon et la nature du support en sont responsables. La figure I.9 illustre, grâce à quelques exemples, les difficultés liées à l'écriture manuscrite.

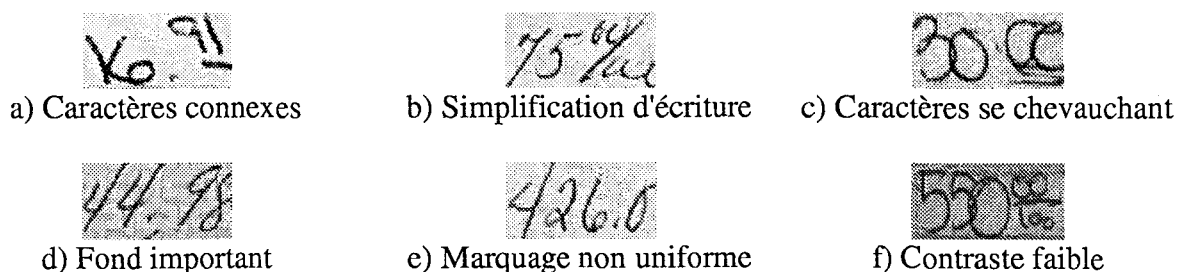


Figure I.9 : Exemples des difficultés liées à la reconnaissance de caractères manuscrits

Ainsi, parallèlement à la reconnaissance des caractères manuscrits mettant en œuvre des problèmes de classification dus à la grande variété de scripteurs, d'autres recherches visent à augmenter les performances des systèmes de reconnaissance en leur permettant la reconnaissance des documents bruités ou dégradés (exemples d), e) et f)).

La qualité du document à analyser est également la plus grosse limitation des systèmes de reconnaissance de caractères imprimés. L'amélioration des performances des systèmes d'OCR, pour permettre la reconnaissance sur des documents de mauvaise qualité, est également un axe de recherche actuelle important. Les applications visées par ces travaux sont la reconnaissance de caractères appliqués à la lecture de livres anciens [BED97], l'identification de plaques minéralogiques [BOU91], le contrôle de tickets et l'identification industrielle [ATH96].

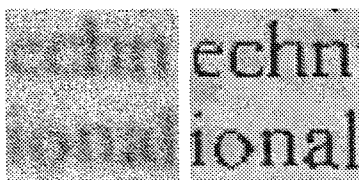


Figure I.10 : Exemples de difficultés liées à la qualité de l'impression

En complément des deux thèmes de recherche précédemment énoncés, on peut également citer le nombre important de travaux de recherche actuellement menés, concernant la reconnaissance de caractères spécifiques (arabe [TRE95][AMI98], bangla [CHA98], chinois [HIL93][LEE97],...).

Conclusion de la première partie

En dehors de l'écriture manuscrite, à l'heure actuelle, les principales sources de mauvais fonctionnement de systèmes d'OCR sont liées à la qualité de l'acquisition. Dans le cadre d'images de mauvaise qualité (bruit, luminosité), dès que les conditions d'acquisition ne sont plus optimales, le taux de reconnaissance diminue considérablement et les systèmes de reconnaissance sont mis en défaut. La résolution de ce problème passe par des algorithmes adaptés nécessitant une étape de prétraitement, d'extraction de caractéristiques plus robustes et un classifieur plus complexe. L'autre difficulté des systèmes de reconnaissance est liée à la dégradation des caractères. Dès que la qualité du document chute, le système présente un taux d'erreur important.

Notre travail a consisté à définir un système de reconnaissance de caractères pouvant s'appliquer à l'identification industrielle. Un tel système doit s'adapter à un grand nombre d'applications OCR présentant des problèmes liés à la qualité de l'acquisition.

En effet, notre but est de concevoir un système capable de s'adapter à des conditions d'acquisition variable (caméra, éclairage), d'effectuer une reconnaissance quelles que soient la position, la taille et l'orientation des caractères et enfin de s'affranchir du type de marquage et du type de support utilisés. Pour cette étude, nous nous sommes orienté vers la réalisation d'une architecture neuronale, qui, à partir de l'acquisition d'un caractère, va extraire un modèle invariant puis effectuer la classification.

Notre choix d'une architecture neuronale découle de l'évolution des réseaux de neurones appliqués à la classification notamment ceux utilisés en reconnaissance de caractères.

Actuellement, ces techniques neuronales développées pour la classification offrent des performances identiques ou supérieures aux techniques éprouvées. En effet, ces premières semblent plus adaptées à résoudre les cas complexes que ces dernières. Dans les prochains chapitres, nous définirons les concepts de classification et d'extraction de modèle invariant pour ensuite aborder la description de notre système.

Nous présenterons, dans le second chapitre, les techniques de classification utilisées en reconnaissance de caractères. Cette partie permettra, après avoir exposé les différentes approches, de définir les principales techniques statistiques classiques avant de découvrir les réseaux de neurones utilisés en classification ainsi que leur avantage. En complément de la présentation bibliographique de chacune des principales architectures, nous ferons part des résultats de notre étude pratique appliquée à la reconnaissance de caractères. Cette étude va nous permettre de déterminer la réelle aptitude des réseaux de neurones à effectuer une reconnaissance de caractères multiformes et de sélectionner le type de réseau classifieur que nous allons utiliser lors de la réalisation de la partie classification de notre système.

La troisième partie nous permettra d'introduire les méthodes développées pour la reconnaissance invariante. Après avoir introduit les méthodes d'extraction de modèle invariant et transformations invariantes classiques, nous présenterons les principales techniques neuronales développées à ce jour. Cette partie nous donnera également l'opportunité d'introduire les principales techniques d'extraction de caractéristiques utilisées pour la reconnaissance de caractères imprimés. Les notions et les concepts présentés dans cette partie nous permettront de construire la partie de notre système permettant l'extraction d'un modèle invariant.

Deuxième Partie :

La classification

Introduction

La partie reconnaissance d'un système d'OCR que nous avons décrite dans la première partie, est construite autour d'une opération de classification. A partir d'une représentation d'un caractère, le classifieur va déterminer sa classe d'appartenance. Dans le cas de la reconnaissance de caractères hors ligne, la seule information qui va être utilisée est une image du caractère. On peut dénombrer six approches pour les techniques de reconnaissance employées pour l'OCR : le "template matching", l'approche structurelle, l'approche syntaxique, l'approche stochastique, l'approche statistique et l'approche connexionniste. Chacune est caractérisée par sa façon de traiter l'information contenue dans l'image. La technique du "template matching", la plus ancienne, va utiliser directement les niveaux de gris de l'image présentée et comparer ces niveaux avec ceux d'échantillons déjà connus. C'est le principe utilisé par toutes les techniques de corrélation. Les approches structurelles, stochastiques et syntaxiques utilisent une étape de caractérisation utilisant des techniques d'analyse d'image afin d'extraire des caractéristiques et des primitives du caractère à classer. Dans ce cas, les techniques de classification sont simples. Il s'agit de déterminer si telle ou telle caractéristique est présente ou absente. L'approche statistique et l'approche connexionniste, beaucoup plus récentes, utilisent aussi des techniques d'analyse d'images. Contrairement aux approches basées sur l'analyse de forme, ces techniques d'analyse ont pour but d'extraire des caractéristiques métriques sur l'objet : hauteur, largeur, niveau de gris, moyenne.

Cette deuxième partie est consacrée à l'étude des classifieurs applicables à la reconnaissance de caractères. Plus principalement, elle a pour but de définir les systèmes les mieux adaptés à la reconnaissance de caractères en utilisant l'approche connexionniste.

Dans un premier temps, nous ferons un bref exposé des principales techniques de classification utilisées pour la reconnaissance de caractères suivant les différentes approches. Le premier paragraphe présente, le "template matching", l'une des premières techniques utilisées pour la reconnaissance de caractères puis énonce les principales techniques structurelles, syntaxiques et statistiques appliquées à l'OCR.

Nous introduirons ensuite les méthodes statistiques classiques pour pouvoir ensuite introduire l'utilisation des réseaux de neurones. Dans notre étude bibliographique des classifieurs neuronaux, nous effectuerons une comparaison de leurs performances en les appliquant à un problème de reconnaissance multiforme.

A l'issue de cette étude, nous allons pouvoir sélectionner un type d'architecture neuronale que nous utiliserons dans la réalisation de notre système de reconnaissance.

II.1 La classification utilisée en reconnaissance de caractères

Avant d'évoquer les notions liées à la classification statistique et à la présentation des techniques neuronales efficaces pour la classification, ce premier paragraphe permettra de prendre connaissance des techniques éprouvées pour la reconnaissance de caractères depuis de nombreuses années.

II.1.1 Le template matching

Cette technique de comparaison de formes [ROS82][KUN93] est encore actuellement le principal moteur de reconnaissance OCR utilisé par les automates de vision proposés dans le cadre du contrôle industriel [ROC94][SIE97]. Cette approche est la plus simple pour effectuer la reconnaissance de caractères. Elle fut à l'origine des premiers systèmes commercialisés et reste

couramment utilisée pour les problèmes simples. Le "template matching" compare l'image prise du caractère avec une librairie constituée d'un ensemble des différents types de caractères : les "templates" ou modèles de librairie. Le système compare les niveaux de gris du modèle avec ceux des différents éléments de la librairie, puis il affecte à l'image la classe du modèle qui lui correspond le mieux. La technique mathématique utilisée est la corrélation. A partir de la bibliothèque de modèles, le système de reconnaissance va corrélérer l'image présentée avec chacun de ces modèles. L'étude des coefficients de corrélation va permettre de classer les différentes lettres.

Soit I l'image N*N avec $f(x,y)$: intensité en niveau du gris du pixel x,y représentant l'image issue du capteur.

Soit l'imagette 2M*2M avec $h(x,y)$: intensité en niveau du gris du pixel x,y représentant un modèle connu du système. Le template matching donne comme résultat :

$$g(x,y) = \sum_{t_1=-M}^M \sum_{t_2=-M}^M f(x+t_1, y+t_2)h(t_1, t_2) \quad (II.1)$$

Cette fonction est calculée sur toute l'image I. L'introduction du rapport signal sur bruit permet de définir un critère de classification.

$$SNR(x_0, y_0) = \frac{g^2(x_0, y_0)}{\sum_{(x,y) \neq (x_0, y_0)} g^2(x, y)} \quad (II.2)$$

Si le rapport signal sur bruit maximal est supérieur à un seuil θ (seuil de classification), il nous donne la position x_0, y_0 de la forme reconnue. La difficulté de cette technique est de choisir correctement les seuils de décision. En effet, les coefficients de corrélation peuvent présenter des variations très faibles. Le choix est alors délicat.

Cependant, cette technique peut être améliorée par l'utilisation de la corrélation normalisée qui permet d'optimiser la distinction des différents pics de corrélation.

Cette technique s'exprime par :

$$g(x,y) = \frac{1}{\sigma_f(x,y)\sigma_h} \sum_{t_1=-M}^M \sum_{t_2=-M}^M (f(x+t_1, y+t_2) - m_f(x,y))(h(t_1, t_2) - m_h) \quad (II.3)$$

avec $\rightarrow m_f(x,y) = \frac{1}{(2M+1)^2} \sum_{t_1=-M}^M \sum_{t_2=-M}^M f(x+t_1, y+t_2)$ la moyenne de la zone de l'image observée.

$\rightarrow \sigma_f(x,y) = \left(\frac{1}{(2M+1)^2} \sum_{t_1=-M}^M \sum_{t_2=-M}^M (f(x+t_1, y+t_2) - m_f(x,y))^2 \right)^{\frac{1}{2}}$ l'écart type dans cette zone.

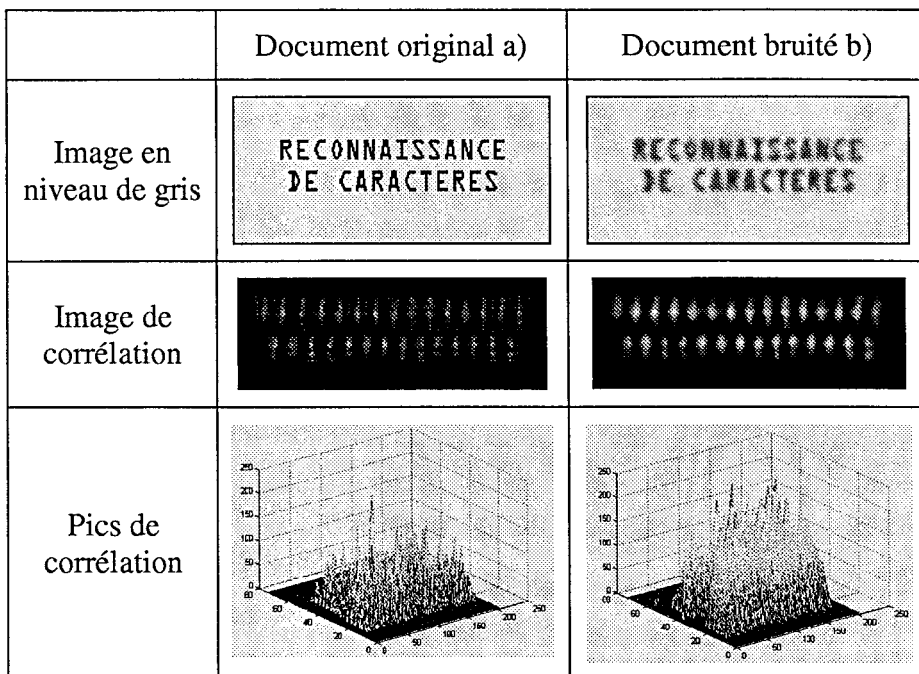
$\rightarrow m_h(x,y) = \frac{1}{(2M+1)^2} \sum_{t_1=-M}^M \sum_{t_2=-M}^M h(t_1, t_2)$ la moyenne du modèle.

et $\sigma_h = \left(\frac{1}{(2M+1)^2} \sum_{t_1=-M}^M \sum_{t_2=-M}^M (h(t_1, t_2) - m_h)^2 \right)^{\frac{1}{2}}$ l'écart type du modèle.

L'avantage des techniques de "template matching" est qu'elles ne nécessitent que peu d'informations sur les formes à reconnaître. En effet, il n'est pas nécessaire d'effectuer une analyse des formes à reconnaître afin de les décrire selon des points caractéristiques. Seule une image dont on dispose a priori va permettre la reconnaissance.

En revanche, les désavantages de cette technique sont liés aux durées de traitement et à la sensibilité aux bruits. En effet, plus le système dispose d'un grand nombre d'éléments de bibliothèque, plus le nombre de corrélations à calculer est important. Une fois ces calculs effectués, il s'agit de choisir le pic de corrélation le plus important. Cependant, l'influence du bruit sur l'image va modifier les pics de corrélation et va donc provoquer des erreurs.

L'exemple (figure II.1) illustre notre propos sur l'influence du bruit. A partir de deux images, une image originelle (de taille 200x70) et une image bruitée artificiellement, nous visualisons le résultat de la corrélation. Dans le cas de l'image claire a), le pic de corrélation maximale est beaucoup plus facilement identifiable que dans le cas de l'image bruitée b).



RA

modèle utilisé

Figure II.1 : Influence du bruit sur les pics de corrélation

Les techniques de "template matching" sont encore utilisées et certains travaux apparaissent encore afin de mieux distinguer les pics de corrélation [CAS86][RAO94]. De nouvelles techniques cherchent ainsi à mettre en application des algorithmes beaucoup plus rapides. De fait, les dernières techniques [LI97] proposent, en comparaison avec la corrélation normalisée des rapports SNR augmentés de 11 à 30 dB et des durées de traitement divisées par 20 à 50. Des techniques concernant l'approche multirésolution (pyramidale) des images ont également permis de diviser par 100 les temps de traitement [TAN81][HIR97]. Enfin, certaines techniques de corrélation n'utilisant que les contours ont également été développées [YAS97].

Face à la diversité des caractères et des polices à reconnaître, le "template matching" s'avère inadapté dans le cas d'une reconnaissance multifonte, étant donné le nombre d'éléments constituant la bibliothèque de modèles. Cette technique reste cependant utilisée dans le cas de l'OCV (Optical Character Verification) qui permet de déterminer s'il y a présence ou non de caractères, ou si le nombre de polices est très restreint.

En ce qui concerne l'OCR, il a été nécessaire d'améliorer les systèmes de reconnaissance. Des nouvelles techniques dites de classification de caractéristiques sont apparues.

II.1.2 Classification de caractéristiques

Contrairement à la méthode présentée précédemment : le template matching qui effectue une classification uniquement à partir des niveaux de gris de l'image, les techniques de classification de caractéristiques vont utiliser une représentation simplifiée de la forme (ou du caractère) pour en déduire sa classe d'appartenance. Cette représentation essentiellement vectorielle est obtenue à partir d'une première étape dite d'extraction de caractéristiques. La classification de caractéristiques n'est que la seconde étape du système de reconnaissance basé sur l'utilisation de caractéristiques. Le principe consiste à associer une technique d'analyse d'image (l'extraction de caractéristiques) et une technique de décision (la classification) simple (figure II.2). Cette technique permet ainsi d'effectuer la reconnaissance en limitant l'espace de représentation (le vecteur caractéristique) à classifier.

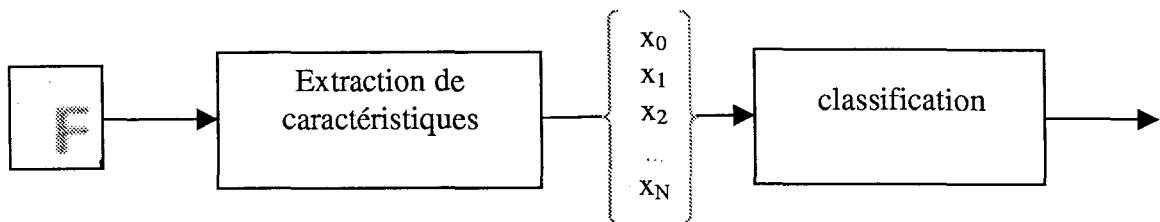


Figure II.2 : Principe de la classification de caractéristiques

Ces techniques de classification de caractéristiques sont donc étroitement liées aux techniques d'extraction. Celles-ci, selon Milgram [MIL93], sont directement à l'origine de la distinction des différentes techniques de classification.

L'auteur distingue cinq principales approches :

- L'approche structurale.
- L'approche syntaxique.
- L'approche stochastique.
- L'approche statistique.
- L'approche connexionniste.

Contrairement à l'approche globale qui est plus généraliste (reconnaissance de forme, décision, diagnostic), un grand nombre de techniques structurales ou syntaxiques ont été développées spécialement pour la reconnaissance de caractères. Historiquement, les premières techniques d'analyse structurale et syntaxique découlent des techniques de corrélation. Plutôt que d'utiliser comme dans la technique du "template matching", la représentation complète du caractère, on utilise une bibliothèque de primitives générales telles que droites, courbes, intersections. L'une des premières bibliothèques, les primitives de Coueignoux est représentée figure II.3.

Troncs	Extremités horizontales	Courbes	Traits-Points
Arcs	Extremités Courbes	Queue	Ventre

Figure II.3 : Exemples de primitives de Coueignoux

Les trois paragraphes suivants vont énoncer les différentes techniques de classification rencontrées en reconnaissance de caractères en fonction des différentes approches. Les techniques d'extraction de caractéristiques seront détaillées dans le troisième chapitre.

II.1.2.1 Les techniques structurelles

Ces techniques sont basées sur l'analyse et la caractérisation des formes [PAV77]. Appliquées à l'OCR, elles consistent à extraire les éléments qui caractérisent et distinguent les caractères comme : des traits horizontaux, verticaux, des intersections, des jonctions. Ces éléments sont également dénommés primitives. Ainsi, la représentation structurelle de la lettre "H" peut être donnée par trois primitives : 2 traits verticaux, 1 trait horizontal.

Les techniques de reconnaissance structurelle consistent à ordonner hiérarchiquement des primitives sous forme d'un arbre de décision. Après avoir extrait ces primitives de la forme observée, la reconnaissance structurelle se base sur les relations liant ces différentes primitives. Ces relations peuvent être combinatoires ou métriques. Dans l'exemple suivant (figure II.4), on peut caractériser le chiffre «2» par une bibliothèque de primitives. Ainsi, le vecteur de caractéristiques du chiffre 2 est codé par {H, C, D, G} :

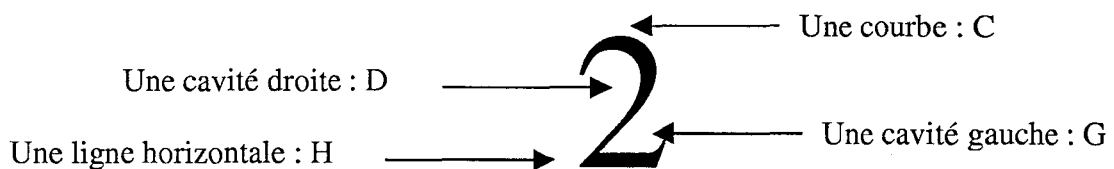


Figure II.4 : Exemples de caractéristiques structurelles

Le codage utilise les primitives de la bibliothèque mais également les relations entre ces primitives dans la forme. Ainsi, le chiffre 5 peut également être codé par {H, C, D, G}. La distinction entre les deux caractères peut s'effectuer en utilisant la disposition des primitives. Ainsi, le chiffre 2 est codé par C puis D puis G puis H : {C,D,G,H} et le chiffre 5 est codé par {H,G,D,C}.

De nombreuses méthodes de classification structurelle ont été développées à partir d'une application en reconnaissance de caractères. Les deux principes de cette classification [MIC84][MIL93] sont :

- La mise en correspondance des représentations de type graphe. On peut citer comme principales techniques : les graphes d'appariement, les arbres d'interprétation.
- Les techniques métriques qui sont les mesures de distance, les distances entre chaînes, les différentes techniques de relaxation.

L'étape de caractérisation, le choix puis l'extraction des caractéristiques, reste l'étape la plus importante dans la reconnaissance structurale. Les primitives doivent permettre de caractériser et de distinguer toutes les formes à reconnaître. Dans l'exemple présenté figure II.1, il est évident que les primitives ne sont pas suffisantes pour caractériser tous les caractères alphanumériques.

Le principal désavantage des méthodes structurales est lié à l'extraction des primitives directement de la représentation physique des formes. Ceci constitue un réel handicap car cette description résiste très mal aux transformations géométriques : translation, rotation mais surtout au bruit. De même à partir d'une représentation bruitée ou incomplète, il sera quasiment impossible de résoudre la tâche de classification.

II.1.2.2 Les techniques syntaxiques

Les techniques syntaxiques sont les moins répandues. Elles ne s'appliquent en effet qu'à des domaines précis de la reconnaissance de formes et sont très proches des méthodes structurales. Elles nécessitent l'utilisation d'un alphabet et d'une grammaire. Leur particularité est d'avoir des modèles entièrement codés par un alphabet particulier. Par exemple, en utilisant le codage de Freeman, "l'alphabet", on code le chiffre 2 {6,6,6,1,1,1,0,7,6,5} la "grammaire" du chiffre (figure II.5).

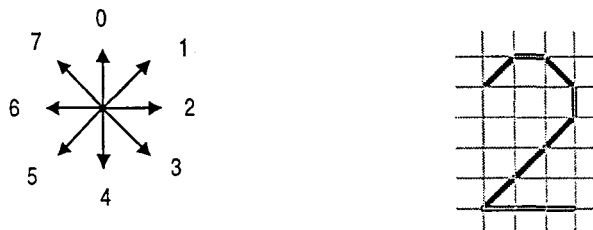


Figure II.5 : Exemple de technique syntaxique : codage de Freeman

Dans le cadre de l'approche syntaxique, les techniques de classification sont très souvent simples. Elles sont très dépendantes du choix de l'alphabet. Celui-ci est choisi afin de permettre de décrire de la façon la plus générale qui soit le caractère et ce, tout en permettant de le distinguer aisément. Les principaux algorithmes de classification pour les méthodes syntaxiques sont basés sur des techniques métriques. En fonction du choix de l'alphabet, on pourra utiliser soit des techniques simples comme la distance inter-chaînes soit des techniques plus évoluées et efficaces comme la relaxation probabilistique [FU82].

Les limitations de ces méthodes sont identiques à celles des méthodes structurales.

II.1.2.3 Les techniques stochastiques

Les techniques stochastiques sont essentiellement utilisées pour la reconnaissance de l'écriture manuscrite en ligne [BEL92]. En complément des primitives issues de la forme du caractère, on utilise les informations temporelles liées au mouvement du crayon. Ainsi, on

dispose de la séquence d'apparition des primitives. On dispose également des éléments complémentaires comme le lever de crayon, l'origine et la fin de la lettre voire l'intensité de l'appui du crayon.

Le caractère est modélisé sous forme de graphe d'états, où chaque état représente une observation et par les relations entre ces différents états, on parle alors de chaîne stochastique.

La reconnaissance stochastique consiste à rechercher dans un graphe d'états le chemin le plus court décrivant les éléments observés. La technique utilisée est essentiellement basée sur l'utilisation de chaînes de Markov [BAU72] où chaque caractère est décrit à l'aide d'une séquence de primitives et une séquence d'états. Les chaînes markoviennes cachées ne sont qu'un moyen supplémentaire d'extraction des primitives de la forme. Une étape de classification est nécessaire. La plus généralement utilisée pour l'approche stochastique est l'algorithme de Viterbi et ses variantes [RAB89].

Dans le cas de caractères imprimés, l'aspect temporel étant perdu, les techniques stochastiques n'offrent que très peu d'intérêt. Cependant, certains travaux les ont utilisées, notamment les chaînes de Markov cachées [ANI92], pour la reconnaissance hors ligne. Dans ce cas, l'aspect temporel est estimé à partir d'un suivi de contours des caractères. Les résultats de cette technique sont corrects et le principal avantage est d'obtenir des temps de traitement très courts. Cependant, cette méthode est limitée par les risques de discontinuité des contours.

II.1.2.4 Les techniques statistiques

Les techniques statistiques utilisent une représentation purement numérique des formes. Elles sont élaborées à partir des caractéristiques faites de mesures et de transformations effectuées sur la forme. C'est surtout depuis l'apparition de systèmes de traitement numériques que les méthodes statistiques ont pu se développer. En effet, l'utilisation d'une représentation numérique oblige à utiliser des systèmes performants dans le traitement de l'information. L'intérêt de ces techniques est de pouvoir construire facilement un classifieur si celui-ci possède des capacités d'apprentissage, à partir d'un ensemble de mesures faites et des connaissances a priori.

Dans le cadre de notre étude, nous nous intéresserons uniquement aux techniques statistiques pour aboutir aux techniques connexionnistes. En effet, ces techniques de classification peuvent s'appliquer à toutes les approches précédemment énoncées. Ce sont également les techniques les plus performantes pour résoudre des problèmes de classification complexes mais aussi les plus robustes : ce sont les seules qui aujourd'hui peuvent être un peu plus insensibles au bruit des modèles présentés [ATH96].

Ces techniques sont regroupées sous le terme plus général de théorie de la décision. Elles généralisent la notion de classification comme la discrimination de vecteurs (échantillons) présents dans un espace de représentation. Les dimensions de cet espace (au nombre de composantes des vecteurs) peuvent être à la fois représentées par des composantes structurelles, syntaxiques, topologiques ou métriques. Les paragraphes suivants ont pour but d'introduire les différents types de classifieurs statistiques et de détailler les principales techniques utilisées. Ces méthodes statistiques seront ensuite présentées sous leur adaptation neuronale. Cette étude sera orientée par notre volonté de définir un système de reconnaissance applicable aux caractères imprimés.

II.2 Les méthodes statistiques classiques

La théorie de la décision est définie comme l'approche statistique fondamentale pour résoudre les problèmes de classement et de classification. A partir d'une représentation de l'objet que l'on cherche premièrement à caractériser, la théorie de la décision va définir les éléments qui permettent sa reconnaissance automatique. Le but essentiel de la théorie de la décision et des procédures de classification est d'identifier les classes entre lesquelles se répartissent les vecteurs de caractéristiques appartenant à un espace de représentation.

L'étape de décision en reconnaissance de forme est basée sur l'utilisation d'un classifieur. Ce classifieur ou système de décision permet à partir d'un échantillon de déterminer sa classe d'appartenance. L'échantillon présenté au classifieur correspond à un vecteur représentant l'objet. Ce vecteur est construit à partir de mesures et d'opérations (transformations) effectuées sur l'objet (ces notions seront développées dans la troisième partie). Chacune des composantes de ce vecteur va définir une dimension de l'espace de représentation des échantillons \mathbb{E}^N . Chaque échantillon est alors représenté par un point dans cet espace. Le rôle du classifieur consiste à discriminer cet espace en régions correspondant aux classes.

Cette fonction que réalise le classifieur, est généralement obtenue à l'aide d'une phase d'apprentissage. Durant cette étape, le classifieur va définir à partir de exemples dits d'apprentissage représentatifs des individus à classer, les régions et/ou les frontières définissant chaque classe. La figure II.6 représente un exemple d'espace de classification de 5 types de fleurs en fonction de la hauteur et de la largeur de leurs pétales.

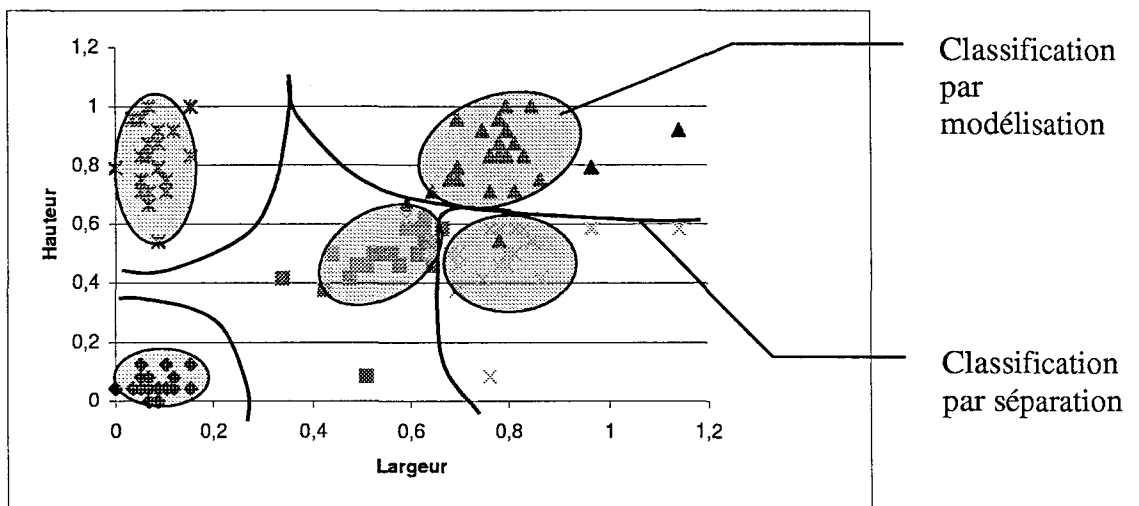


Figure II.6 : Séparation d'un espace de représentation \mathbb{E}^2

La phase d'apprentissage est la phase la plus délicate. L'apprentissage nécessite la création d'une base d'apprentissage constituée d'individus-exemples les plus judicieusement choisis. Ces exemples retenus doivent permettre de généraliser le fonctionnement du classifieur. Il est donc nécessaire de retenir les modèles permettant d'approximer au mieux toutes les classes à distinguer. Ceci impose de choisir les individus les plus proches des frontières interclasses. Leur nombre et leur localisation vont définir la précision du fonctionnement du classifieur.

Cette étape d'apprentissage peut être supervisée : on présente au classifieur un individu et on lui indique la classe d'appartenance, ou non-supervisée : on ne présente que l'individu, le classifieur organise la connaissance a priori des classes.

L'apprentissage non-supervisé est principalement utilisé lorsque l'on ne dispose pas de toutes les informations a priori : nombre et structure des classes, allure des descripteurs. Cette connaissance a priori étant inconnue, le système non-supervisé va essayer de générer des lois de classification en organisant les données présentées en phase d'apprentissage. On parle principalement de système auto-organisé.

Plus en marge, certains systèmes peuvent utiliser un apprentissage renforcé basé sur les probabilités d'appartenance des données aux classes à distinguer ainsi qu'un apprentissage semi-supervisé basé sur des indications qualitatives concernant les données [JOD94b].

La fonction classification étant construite, pour tout individu présenté, le classifieur va estimer la classe d'appartenance en comparant sa position avec les régions apprises. Cette phase de fonctionnement est appelée phase de décision. La classe d'appartenance sera estimée à partir de critères de proximité. Les critères de proximité sont nombreux et sont à l'origine des répartitions des différentes techniques de classification. Ces techniques seront présentées dans le deuxième paragraphe de cette partie.

Cependant, le seul critère de proximité entre l'échantillon et une région n'est pas suffisant pour définir correctement un classifieur. Pour un bon fonctionnement du système de reconnaissance, il faut pouvoir éviter les erreurs de classification. La notion essentielle qui doit être introduite dans chaque classifieur est le rejet. Ainsi, on peut définir des critères nécessaires à la constitution d'un classifieur.

A partir d'un individu présenté, un système de classification doit être capable de déterminer si :

- Il appartient à la classe C_i .
- Il n'est pas parfaitement conforme à la classe C_i .
- Il est à la frontière de deux classes.
- Il n'appartient à aucune des classes connues.
- Il appartient à deux ou plusieurs classes.
- Il est impossible à traiter.

A partir des estimations faites, le système de reconnaissance doit ensuite déterminer le résultat en tenant compte de la possibilité d'appartenance d'un individu aux différentes classes. Les notions essentielles de rejet peuvent être apportées. Le rejet d'ambiguïté va permettre de rejeter l'individu lorsqu'il se situe à la frontière entre deux classes ou lorsque il appartient à plusieurs classes. A l'inverse, la notion de rejet de distance agit sur un individu classé, mais éloigné de tous les prototypes définissant sa classe (figure II.7).

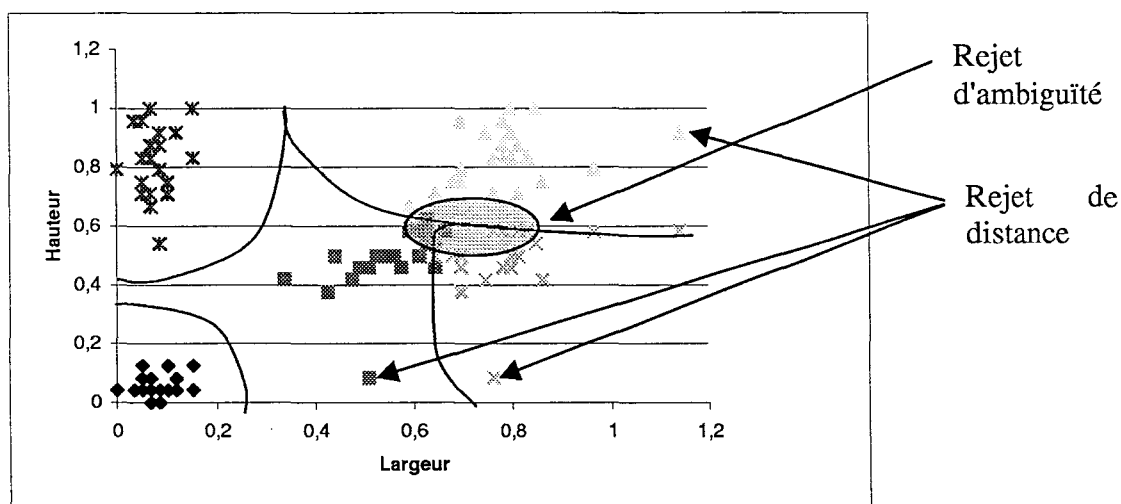


Figure II.7 : Notions de rejet

Les approches de la classification statistique sont nombreuses : distance minimum, analyse discriminante linéaire et quadratique, analyse factorielle discriminante et classification bayésienne.

Nous n'aborderons, ici que la classification bayésienne. Le point de départ est la règle de Bayes. A partir de cette règle, de nombreux algorithmes de physique statistique ont été développés.

II.2.1 Classification bayésienne

Le théorème énoncé par le mathématicien Thomas Bayes est à l'origine en 1763 de la classification bayésienne. Cette théorie [LAP51][DEM63] est à la base des principales méthodes utilisées en physique statistique. Elle permet d'introduire les notions de probabilité dans la résolution des problèmes en reconnaissance de formes. A partir de la représentation physique des formes à reconnaître, on extrait des vecteurs caractéristiques. Ces vecteurs à classer sont considérés comme des réalisations d'un vecteur aléatoire x caractérisé par une densité de probabilité $f(x)$. Connaissant a priori les probabilités des classes d'appartenance, la théorie bayésienne permet de définir a posteriori les densités de probabilité d'appartenance du vecteur x aux différentes classes. Cette théorie introduit également la notion d'erreur de décision due à un recouvrement entre les différentes densités de probabilité de chaque classe. Elle permet par une adaptation de la minimiser.

II.2.1.1 Règle de Bayes

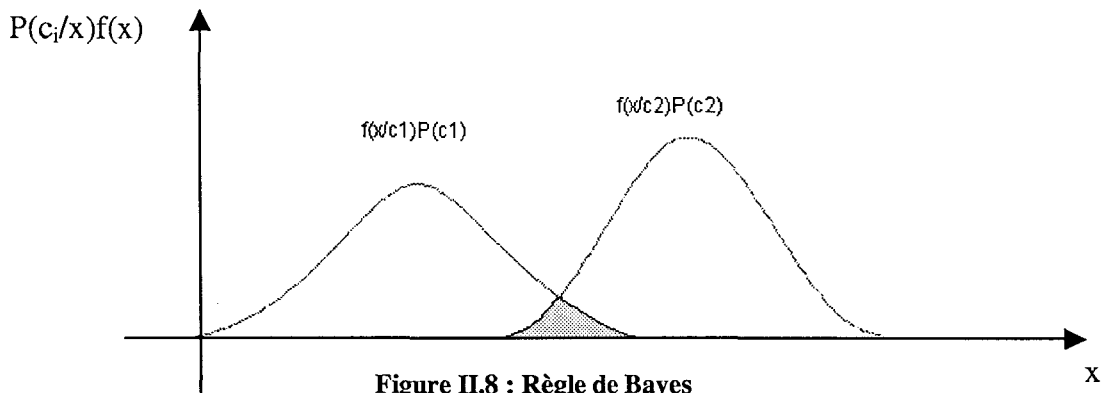
Soient $c_1, c_2, c_3, \dots, c_M$ M classes, où chaque classe est a priori caractérisée par une probabilité $P(c_i)$. Soit un élément x de \mathbb{R}^e auquel on attribue une loi de probabilité conditionnelle suivant chaque classe c_i : $f(x/c_i)$. On associe $f(x)$ sa densité de probabilité à :

$$f(x) = \sum_{i=1}^M f(x/c_i)P(c_i) \quad (\text{II.4})$$

La règle de Bayes énonce que la classe associée à l'échantillon x est celle qui maximise :

$$P(c_i/x) = \frac{f(x/c_i)P(c_i)}{f(x)} \quad (\text{II.5})$$

Cette relation permet de calculer a posteriori les probabilités $P(c_i/x)$.



En définissant la règle de cette forme, on est assuré de choisir la classe d'appartenance ayant la probabilité la plus forte. Sous cette forme simple et dans notre exemple figure II.8, la frontière entre deux classes est définie selon une droite. Ainsi, pour un échantillon x situé à droite de la frontière, sa classe d'appartenance est c_2 et si x est situé à gauche de la frontière, sa classe d'appartenance est c_1 . C'est autour de cette droite et dans une zone proche que l'on risque de faire apparaître une probabilité d'erreur (zone de recouvrement en grisé).

II.2.1.2 Règle de Bayes à coût minimum

Un critère de coût est introduit pour optimiser la procédure de décision optimale et ce, afin de minimiser l'erreur de recouvrement. La règle de Bayes est adaptée en introduisant la notion d'un coût de classification. En fonction de chaque classe d'appartenance de la forme, on associe un coût. Soit $K(i/j)$ le coût de classer x dans c_i lorsqu'il appartient à c_j . La règle de Bayes consiste à calculer le coût pour chaque classe et de choisir celle pour laquelle le coût est minimal. Ainsi, la classe i est la classe choisie si :

$$\sum_{j=1}^M K(i/j)P(c_j/x) = \text{Min} \sum_{j=1}^M K(k/j)P(c_j/x) \quad k=1,2,\dots,M \quad (\text{II.6})$$

Le critère est maintenant le risque global c'est à dire calculé sur toutes les classes. La règle de décision traditionnellement utilisée est celle fixée par les coûts dits de pénalisation symétrique. Chaque erreur est pénalisée d'un coût équivalent valant 1.

Ainsi $K(i/j) = 0$ si $i = j$; $K(i/j) = 1$ si $i \neq j$

$$\text{Le critère minimal est donc : } \sum_{j=1, j \neq i}^M K(k/j)P(c_j/x) = 1 - P(c_i/x) \quad (\text{II.7})$$

Cette technique de classification a été appliquée à la reconnaissance de caractères multifontes [BAI91]. Le taux d'erreur obtenu par cet algorithme est de 0,5%. Ceci illustre la minimisation du risque d'erreur de cette technique.

Cependant, pour utiliser la règle de Bayes, on suppose connues la probabilité associée à chaque classe et les densités conditionnelles de probabilité. En pratique, on les connaît rarement et on tente de les estimer. Cette estimation des densités conditionnelles de probabilité est la plus complexe.

Elle peut se faire selon deux approches regroupées sous les termes de : méthodes paramétriques et méthodes non-paramétriques [DUB90].

- Les méthodes paramétriques font des hypothèses sur la famille de lois à laquelle appartiennent les probabilités conditionnelles.
- Les méthodes non-paramétriques ne font aucune estimation sur leur nature.

Les méthodes dites paramétriques supposent une connaissance totale des lois de probabilité régissant les observations. A partir d'un espace de représentation E de dimension N connu, elles permettent d'attribuer une classe de sortie c_x à un vecteur x appartenant à cet espace. Pour ce faire, le vecteur x doit obéir à une loi de probabilité multidimensionnelle $f(x/c_i)$. Ainsi les méthodes paramétriques recherchent les probabilités des classes à partir d'un ensemble défini a priori : les lois de probabilités de chaque individu x . Ces méthodes nécessitent de connaître a priori toutes les lois.

Pour les méthodes non-paramétriques, les lois de probabilité de chaque classe sont totalement inconnues. Ces méthodes estiment les lois de probabilité à partir d'un critère de similitude avec les exemples présentés durant l'apprentissage. Les deux principales méthodes sont les fenêtres de Parzen et la loi des k-plus proches voisins (k-ppv).

De ces deux approches, l'une dite statistique car estimant les probabilités des classes et l'autre dite métrique car se basant sur un critère de distance, les techniques non-paramétriques sont préférées. En effet, tout en ayant une justification statistique, leur développement est beaucoup plus aisé et beaucoup moins lourd que pour les techniques paramétriques.

Ici, en dehors de la règle de Bayes, seules les méthodes non-paramétriques seront présentées.

II.2.2 Méthodes non-paramétriques

Les méthodes non-paramétriques consistent à décrire une classe par un ou plusieurs vecteurs prototypiques. Ces méthodes ne font aucune hypothèse sur la densité de probabilité à estimer. A partir d'un nombre fini d'échantillons, ces méthodes essaient d'interpoler la valeur de la densité de probabilité, soit en faisant diffuser les exemples (méthode de Parzen) soit en utilisant la métrique et une statistique locale (k-plus proches voisins).

On cherche à estimer la densité de probabilité d'un vecteur aléatoire x , à partir d'un ensemble de représentations x^i . La méthode utilisée consiste à calculer $f(x)$, densité de probabilité de x en moyennant l'ensemble des différents noyaux obtenus sur les échantillons de x^i (x_i).

A partir d'un ensemble d'exemples $\{x_1, x_2, x_3, x_4, \dots, x_q\}$ formant la base d'apprentissage, on cherche à estimer la densité de probabilité $f(x_k)$ pour tout exemple x_k . On désigne par P la probabilité qu'un exemple x appartienne à une région R avec :

$$P = \int_R f(x) dx \quad (\text{II.8})$$

Pour estimer la densité de probabilité $f(x|x \in R)$ que x appartienne à R , on peut admettre que celle-ci est équivalente à la probabilité P sur le domaine $V(R)$. Ainsi, on note :

$$f(x|x \in R) = \frac{P}{V(R)} \quad (\text{II.9})$$

Deux méthodes permettent de générer des domaines R . La première consiste à se fixer le nombre d'observations χ en fonction du nombre de prototypes et à faire croître le domaine de manière à englober χ voisins de x_k : c'est la méthode des k-plus proches voisins (k-ppv). La seconde consiste à lier la taille du domaine au nombre de prototypes. C'est la méthode du noyau [DUB90].

II.2.2.1 Algorithme des k-plus proches voisins.

L'algorithme des k-plus proches voisins (k-ppv) est l'un des algorithmes les plus simples que l'on puisse développer en reconnaissance de formes. Il s'agit d'une méthode non-paramétrique basée sur l'estimation d'une loi de probabilité. Cette méthode cadre avec les méthodes de classification bayésienne [COV67]. Elle nécessite l'utilisation d'une phase d'apprentissage de type supervisée. La règle de décision est d'affecter à x , la classe la plus représentée parmi les k-plus proches voisins. Ici, on fixe le nombre χ et on fait varier la taille du domaine de manière à former un ensemble de χ voisins.

Après avoir initialement fixé un nombre χ de voisins à atteindre, on définit un domaine de référence souvent de volume unité. L'étape suivante consiste à augmenter itérativement la taille de ce domaine jusqu'à dénombrer à l'intérieur de ce domaine le nombre initialement fixé χ de vecteurs prototypes.

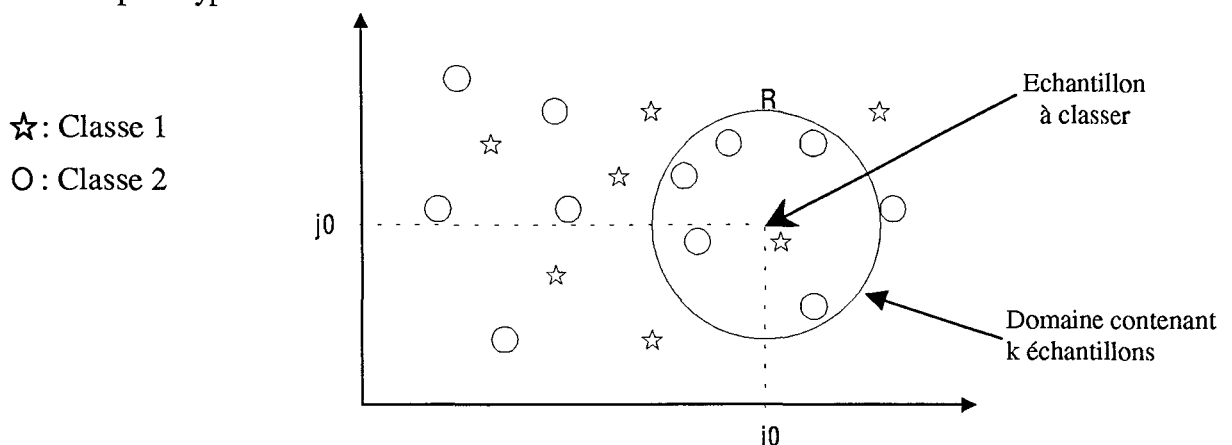


Figure II.9 : Méthode des k-ppv

Après avoir défini un domaine R circulaire de rayon unitaire et dont le centre (i_0, j_0) correspond à l'échantillon à classer, la figure II.9, décrivant un espace à deux dimensions où sont représentés des exemples d'appartenance à deux classes (étoile, cercle), montre le résultat de l'itération si l'on choisit un nombre de 6 voisins. A l'intérieur de ce domaine sont présents les 6 plus proches vecteurs prototypes dont la classe cercle est dominante. Ainsi, l'approximation de $f(x)$ avec x centre du domaine R a pour résultat la classe cercle la classe la plus représentée à l'intérieur de R .

Par extension, on définit la méthode du 1-plus proche voisin comme la méthode associant à l'échantillon testé, la classe du modèle le plus proche dans l'espace.

Pour pouvoir appliquer l'algorithme des plus proches voisins, il faut disposer au préalable d'un ensemble d'exemples k ainsi que de leur classe d'appartenance. Les résultats de cette méthode sont des résultats asymptotiques.

Cet algorithme présente l'avantage de ne faire aucune hypothèse sur les lois des probabilités conditionnelles. Il exploite la disposition de l'espace de représentation obtenu à l'issue de l'apprentissage. Cette technique reste la principale référence lors de comparaison entre différentes techniques de classification [ATH96] [CHT97].

L'un des inconvénients est le temps nécessaire à l'extraction des k éléments voisins, problème lié à l'utilisation de méthode non-paramétrique, le temps de calcul pour la recherche des plus proches voisins fut une limitation aux développements de la méthode telle qu'elle est décrite. Une solution consiste en l'utilisation de l'approche région. On effectue alors un maillage

de l'espace de représentation et on applique la méthode à chacune des nouvelles surfaces [DEL80]. Des techniques de tri peuvent être également appliquées sur les données afin de simplifier le calcul [MIC83]. Une autre technique dite des proches voisins condensés [CHT96] a été développée afin d'améliorer les temps de traitement. Elle est basée sur le prétraitement des données et la condensation [HAR68]. Cette technique consiste à sélectionner les prototypes les plus significatifs limitant ainsi la taille de la base d'apprentissage. La technique du 1-ppv est alors suffisante.

II.2.2.2 Algorithme des fenêtres de Parzen

Appliquée à une tâche de classification, cette technique, comme les k-plus proches voisins, permet d'approximer les densités de probabilités conditionnelles [DEV82]. Cependant, contrairement à la méthode précédente, la méthode des fenêtres de Parzen ne fixe aucun paramètre. Cette estimation est déduite de l'équation (II.10). Elle consiste à pondérer chaque prototype en fonction de la distance par rapport au point x_k pour un type de noyau choisi.

$$\text{Ainsi : } \hat{f}(x_k) = \frac{1}{q} \sum_{i=1}^q \frac{1}{V[R(x_k)]} \varphi(x_k - x_q) \text{ avec } \varphi(y) = \frac{1}{h} \Delta\left(\frac{\|y\|}{h}\right) : \text{noyau} \quad (\text{II.10})$$

Avec q le nombre total de vecteurs prototypiques
 V le volume du domaine R

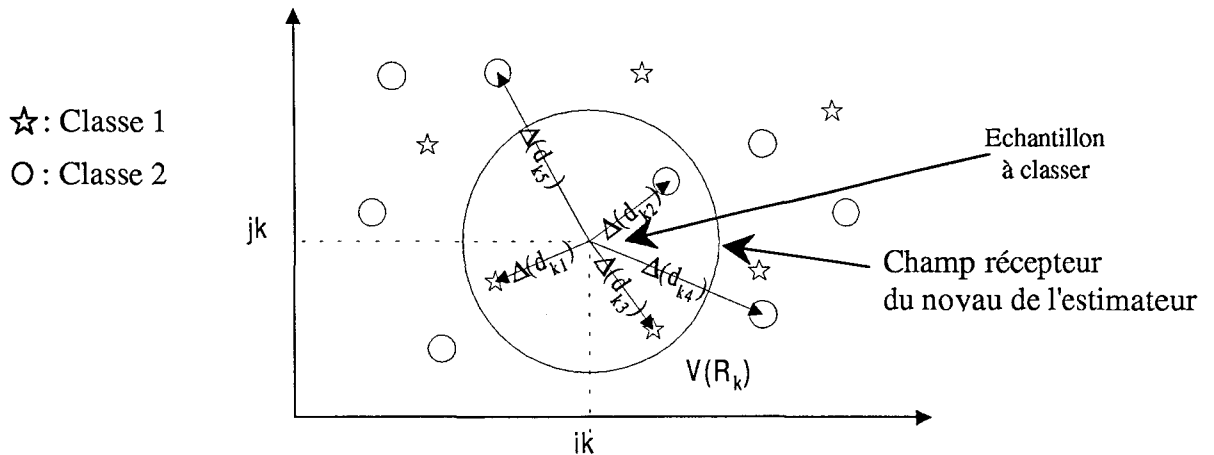


Figure II.10 : Estimateur de Parzen

La figure II.10 illustre la notion de noyau. La fonction $\Delta(d)$ est appelée noyau de l'estimateur. Elle définit une fenêtre, le champ récepteur, sur l'espace de représentation à l'intérieur de laquelle on effectue la somme pondérée des exemples en fonction de la distance les séparant du centre. Cette fonction doit satisfaire à deux conditions : positive et sommable sur l'espace valant 1. Différents types de noyaux peuvent être choisis : Cubique, Triangulaire, Normal, Exponentiel et distribution de Cauchy. Ainsi, si l'on choisit un noyau cubique et unitaire, on se rapproche de la définition des k-ppv.

Cette méthode peut être interprétée comme la superposition de M fonctions élémentaires, chacune étant centrée sur un prototype de la base d'apprentissage. En fonction du paramètre h , on définit la zone d'influence de chaque noyau sur les autres prototypes. Avec une valeur importante, on lisse le résultat. Chaque prototype intervenant dans l'estimation de $f(x_k)$. A

l'inverse s'il est trop petit, seuls les prototypes proches de x_k sont influents ce qui provoque un trop grande sensibilité à la localisation [POS87].

L'avantage de cette méthode est de ne fixer aucun paramètre lié à la base d'apprentissage. Les inconvénients se résument à la nécessité de disposer de toute la base d'apprentissage et de devoir choisir le bon compromis pour la taille du noyau.

Avant d'achever ce rapide tour d'horizon des méthodes statistiques paramétriques et non-paramétriques de classification pour en arriver à la présentation des méthodes neuronales, nous aborderons le cas d'une technique de classification automatique : la méthode des nuées dynamiques.

II.2.3 Méthodes des nuées dynamiques

Le principe des techniques de classification automatique est de permettre une partition automatique des entrées en différentes classes. L'intérêt de la classification non supervisée est d'être efficace là où l'on ne dispose que de peu, voire, pas d'informations sur le nombre de classes utilisées et la nature des répartitions entre ces classes.

Dans la bibliographie deux approches sont définies pour la classification automatique :

- L'approche hiérarchique permet l'organisation des données par l'utilisation d'arbres. Parmi les méthodes, on trouve le regroupement itératif de points [LAN67], les voisins réciproques [ROU85].

- L'approche non-hiérarchique organise les données en optimisant la fonction de décision. On trouve ainsi les méthodes des centres mobiles [DID82] et les nuées dynamiques [JAI88].

L'approche non-hiérarchique est, contrairement à l'approche hiérarchique, applicable à un problème complexe en terme de nombre de classes, de nombre de vecteurs, de dimensions des vecteurs. A l'inverse, la structure des classes peut être plus complète par l'approche hiérarchique qui peut décrire des frontières entre classes plus précisément.

La méthode présentée ici est la méthode des nuées dynamiques, l'une des principales méthodes non supervisées. Comme toute technique de classification automatique, celle-ci est basée sur le "clustering" ou coalescence des données. L'idée est de définir une mesure de similarité entre les différentes entrées x ce qui permet d'obtenir une représentation simplifiée de l'espace \mathbb{R}^e . Pour ce faire deux types d'algorithmes peuvent être utilisés : les algorithmes de clustering (algorithmes de rangements) ou les algorithmes de quantification vectorielle.

- Les premiers regroupent les données en différents ensembles créant une partition de \mathbb{R}^e de manière à optimiser un critère global de similarité.

- La quantification vectorielle a pour objectif de déterminer les erreurs entre un individu et chaque vecteur prototypique. Ainsi, l'erreur E_j entre l'individu x et le vecteur prototypique x_j^p se note :

$$E_j = \int \int_x \|x - x_j^p\|^2 f(x) dx \quad (\text{II.11})$$

avec $f(x)$ densité de probabilité de x

La méthode des nuées dynamiques est une méthode basée sur la quantification vectorielle. Elle a pour objectif de désigner sur l'espace des représentations \mathbb{R}^e une partition S qui définit une classe dans une zone de l'espace. Pour ce faire on définit un ensemble de

partitions $\{S_1, S_2, \dots, S_M\}$ représentant M classes satisfaisant à un critère global de qualité. Chaque partition est construite à partir d'un ensemble d'apprentissage supposé représentatif de chaque classe. La méthode des nuées dynamiques est une méthode d'auto-organisation qui permet d'effectuer des regroupements des éléments de \mathbb{R}^e suivant un critère de similarité. Ce critère est obtenu par une mesure de dissimilarité $\sigma(x, S)$ entre un élément x et un sous ensemble S de \mathbb{R}^e .

Il n'est pas possible d'appliquer cette méthode au cas où la classification doit s'adapter au cours du temps à de nouvelles observations. En effet, la méthode nécessite une définition a priori des différentes partitions. Pour cela on utilise plutôt l'algorithme des k -moyennes ou k -means. A partir d'un espace connu de k groupes composés d'un vecteur, cet algorithme permet d'ajouter un vecteur au groupe dont il est le plus proche.

Une autre technique de classification automatique a été développée afin de permettre d'introduire la notion de "flou" dans la définition des frontières. Cette technique dite algorithme des centres mobiles flous (ACMF) [BEZ81] calcule les degrés d'appartenance du modèle à chaque classe. La phase dite de suppression du flou permet de ne conserver que la classe dont le degré d'appartenance est le plus élevé.

Les limitations de cette technique sont constituées par les choix a priori concernant l'initialisation aléatoire des centres de classes ainsi que la méthode choisie pour le calcul de distance. Ces choix sont très déterminants dans la qualité de la classification.

En complément des méthodes paramétriques et non-paramétriques, est apparue un nouveau type, dérivé des méthodes statistiques : l'approche connexionniste. Ce nouveau type de classification est construit par la réalisation de fonctions discriminantes sans référence à la distribution statistique des probabilités. Il est principalement réalisé à l'aide de réseaux de neurones du type perceptrons multicouches ou réseaux de Hopfield. D'autres types de réseaux sont également utilisés en classification. Ceux-ci découlent de l'adaptation neuronale de méthodes classiques. Le paragraphe suivant présente les principaux réseaux utilisés en classification. Nous présentons, en annexe A, un bref rappel sur les notions de réseaux de neurones.

II.3 Les méthodes neuronales

Depuis quinze ans, la neuromimétique puis le connexionnisme ont permis la création d'un grand nombre de réseaux de neurones doués pour la classification [LEM89][MAR90][BIS95][RIP96][JAI97][LOO97]. Les principales modélisations neuronales sont adaptées aux problèmes de reconnaissance de formes. Pour pouvoir résoudre ce type de problèmes, le système employé doit être capable de traiter une grande quantité d'informations et d'en extraire une caractéristique de sortie générale. Grâce à la faculté d'apprentissage, le réseau génère une architecture spécifique à un problème donné. Le réseau, soumis à une quantité d'exemples, a mémorisé l'ensemble des classes ainsi que leurs frontières, il est ensuite capable de généraliser ses connaissances pour d'autres exemples non appris. Le réseau a donc mémorisé dans son architecture un comportement qui lui permet d'approximer une fonction de classification. L'utilisation de réseaux de neurones en la matière présente deux principaux avantages :

- Ils permettent de construire un système qui peut approximer une fonction discriminante à l'aide des modèles présentés lors de leur apprentissage.

➤ Utilisés après l'apprentissage, les réseaux de neurones sont capables de s'adapter à des exemples non rencontrés. Ainsi, cela suppose que le réseau peut reconnaître et extraire des formes à partir de vecteurs d'entrée bruités ou incomplets.

Nous allons achever cette partie dédiée à la classification, en présentant les principales architectures neuronales développées et utilisées dans le cadre des problèmes de reconnaissance et de classification.

Avant d'aborder la description des différentes architectures neuronales, nous précisons dans le prochain paragraphe les conditions de constitution de la base de données qui nous a permis d'évaluer les performances des différents réseaux pour un problème de reconnaissance de caractères.

II.3.1 Constitution des bases d'évaluation

Pour l'évaluation des classifieurs neuronaux, nous avons construit deux bases : une base d'apprentissage et une base de test. Chaque base est créée à partir d'un ensemble de exemples correspondant à des imagerie de chaque caractère selon différentes polices.

Pour construire notre base d'apprentissage, nous avons employé des fontes standard utilisées par les traitements de texte. Les documents ainsi réalisés ont été scannés puis segmentés. A ces modèles, ont été ajoutés des exemples issus de documents faxés, provenant d'une base de données disponible sur le serveur des universités du Sussex et de Brighton. Les caractères choisis pour construire les bases proviennent de documents présentant un contraste important (encre noire/papier). L'introduction de ces exemples légèrement bruités nous a permis d'éviter de ne traiter que des cas trop idéaux. Chaque exemple de la base d'apprentissage correspond à une imagerie 25*25. La représentation choisie est une représentation binaire. La figure II.11 représente quelques exemples pour la classe « C » de la base d'apprentissage.

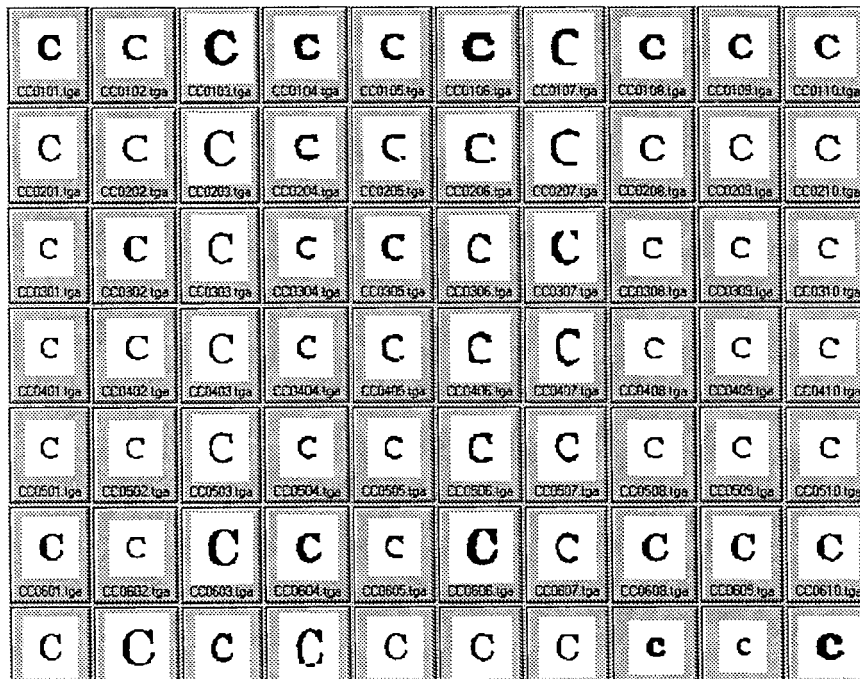


Figure II.11 : Exemples des exemples de la lettre "C" utilisés pour les bases

Des exemples de caractères ont été également utilisés pour construire la base de test qui va permettre de valider le fonctionnement et les performances du classifieur sur des exemples non appris. 36 classes sont définies par les exemples des bases : 26 lettres majuscules et 10 chiffres.

La taille et la répartition des caractères pour les bases sont données dans le tableau suivant :

	Base d'Apprentissage	Base de Test
Nombre de exemples	810	453
Modèles "Papier"	666	414
Modèles "Faxes"	144	39

Tableau II.1 : Constitution des bases d'apprentissage et de test

Les paragraphes suivants sont consacrés à la présentation des différentes architectures et des différents résultats issus de notre étude pour chaque réseau. Nous avons caractérisé les performances de chaque réseau en fonction des taux d'erreur obtenus en phase d'apprentissage et en phase de test à différents pas de l'apprentissage. Ceci reflète indirectement l'évolution du taux de reconnaissance des caractères obtenu par le réseau. Cette information suffit pour comparer les performances des différents réseaux.

II.3.2 Le Perceptron Multicouche ou PMC

Dès le début de leur développement, les réseaux de neurones artificiels ont été appliqués à la reconnaissance. Ainsi, le premier réseau, très connu, le perceptron de Rosenblatt, modélisation très simple du système visuel, fut appliqué à la reconnaissance d'objets. Ce premier modèle développé sur la base d'une structure à trois couches a été basé sur une schématisation du modèle biologique visuel. La première des couches simule la rétine, la deuxième est une couche dite d'association, la dernière est une couche de décision effectuant la reconnaissance des modèles présentés sur la couche d'entrée. Cependant, les limitations de ce premier système neuronal furent vite atteintes. Les travaux de Minsky et Papert [MIN69] ont effectivement démontré que le système n'était capable que de résoudre des problèmes linéairement séparables. C'est seulement dans les années 80, que l'algorithme de rétropropagation a permis l'explosion des PMC. Basé sur l'architecture du perceptron originel, l'adjonction de couches dites cachées plus une descente de gradient a permis de généraliser le perceptron à tous les problèmes de classification. Les perceptrons multicouches sont actuellement les réseaux les plus développés et les plus utilisés. Ces réseaux basés sur l'algorithme de rétropropagation sont des extensions des modèles historiques : Perceptron et Madaline [ROS56].

Les perceptrons multicouches sont construits sur la base d'une architecture neuronale à couches feed-forward (figure II.12). Le réseau dispose d'une couche d'entrée, de couches cachées et d'une couche de sortie. On applique sur chaque neurone de la couche d'entrée, une composante du vecteur caractéristique. Les différentes couches cachées permettent de réaliser l'opération de classification. La couche de sortie donne le résultat de cette classification. Typiquement, il y a en sortie autant de neurones qu'il y a de classes à approximer. Le neurone le plus actif de la couche de sortie représente la classe d'appartenance du vecteur d'entrée.

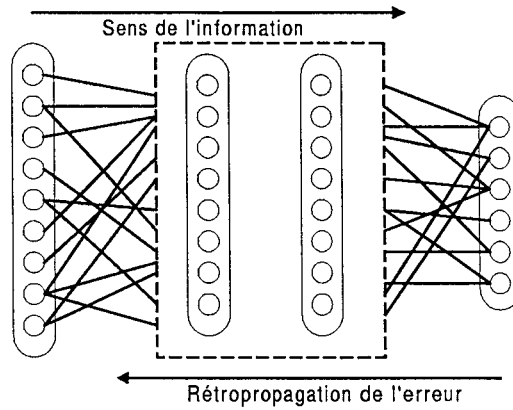


Figure II.12 : Architecture du perceptron multicouche

La méthode d'apprentissage utilisée est une méthode dite supervisée. Durant la phase d'apprentissage, l'algorithme de rétropropagation du gradient RPG [RUM86] permet d'adapter les différents poids du réseau en fonction de l'erreur produite par la couche de sortie par rapport à un vecteur de sortie désiré. C'est l'apprentissage supervisé qui permet de déterminer l'erreur produite sur la couche de sortie. Cette erreur est obtenue en comparant la réponse du réseau à un échantillon de la base d'apprentissage et la classe de cet échantillon.

$$\text{L'erreur quadratique } E \text{ obtenu en sortie s'exprime par } E = \frac{1}{2} \sum_{j=1}^N (D_j - S_j)^2$$

avec D : vecteur de sortie désirée, S sortie du réseau.

L'erreur de la couche de sortie ainsi exprimée, c'est grâce à un calcul différentiel simple [LEC87] que l'on peut exprimer l'erreur E à la sortie de chaque neurone j d'une couche i quelconque à partir des erreurs de la couche précédente $i+1$. Le résultat est :

$$E_j^i = S_j^i \cdot \sum_k w_{kj}^{i+1} \cdot E_k^{i+1} \quad (\text{II.12})$$

avec S_j^i : sortie du neurone j de la couche i
 W_{kj} : poids de connections entre le neurone j de la couche i et le neurone k de la couche $i+1$.

L'erreur ainsi exprimée, les poids peuvent être modifiés selon la règle suivante :

$$\Delta W_{jq}^i = -\alpha \cdot E_j^i \cdot S_q^{i-1} = -\alpha \cdot S_j^i \cdot \sum_k w_{kj}^{i+1} \cdot E_k^{i+1} \cdot S_q^{i-1} \quad (\text{II.13})$$

avec α coefficient d'apprentissage.

De plus, on ajoute un terme de lissage afin d'améliorer l'algorithme précédent. En effet, il est très souvent nécessaire de limiter les variations instantanées de poids. Celles-ci auraient comme conséquence une instabilité. Le terme ainsi ajouté, appelé momentum (μ), permet de conserver une certaine dynamique de la modification précédente. Ainsi, le pas d'adaptation est plus petit, donc la convergence est plus stable.

$$\Delta W_{jq}^i = -\alpha \cdot E_j^i \cdot S_q^{i-1} + \mu \cdot \Delta W_{jq}^{i-1} \quad (\text{II.14})$$

De nombreuses variantes de la règle d'apprentissage ont été proposées. L'amélioration de la convergence constitue le but principal. Il s'agit de s'assurer que le réseau lors de l'apprentissage converge vers un minimum [BAB89], d'améliorer la rapidité de la convergence [MAS92] et d'améliorer ce réseau en terme de convergence. Cependant, l'une des principales difficultés de conception d'un réseau PMC est de le choisir correctement.

L'utilisation du réseau en classification est basée sur la construction d'une fonction discriminante grâce à une phase d'apprentissage. La classification ainsi effectuée est une classification par séparation. Le réseau génère les frontières séparant les différentes classes.

Le choix du nombre de couches cachées est essentiel pour la qualité de la classification. En effet, les propriétés de séparation d'un PMC, qui permettent de déterminer le nombre de frontières de décision, sont fonction du nombre de couches cachées. La figure II.13 représente le pouvoir de séparation d'un perceptron en fonction de ce nombre.

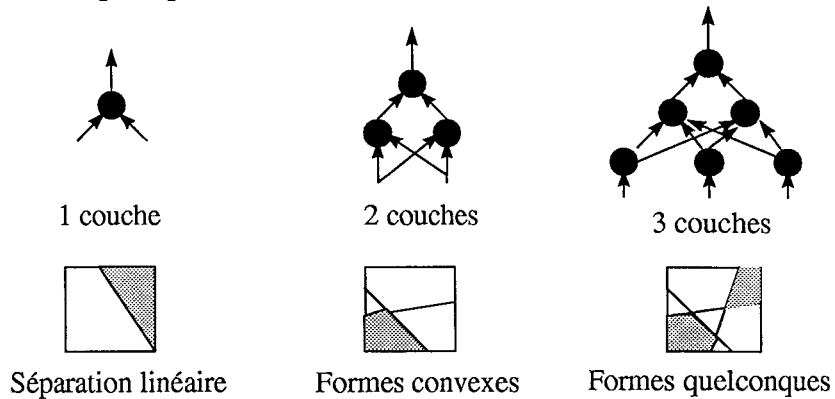


Figure II.13 Le pouvoir de séparation des PMC

La figure II.14 représente l'évolution de l'erreur sur notre base d'apprentissage et de l'erreur sur notre base de test lors de la phase d'apprentissage. Pour l'utilisation du réseau, il est nécessaire de comparer les deux erreurs afin de limiter le nombre d'apprentissage et ainsi de conserver les propriétés de généralisation qui sont nécessaires en phase de fonctionnement. Ainsi, le nombre optimal d'apprentissage, correspondant à une erreur de test minimal, est de 20000 pas.

Dans notre étude des performances du réseau PMC, nous avons également déterminé le nombre et la taille optimale des couches cachées. La figure II.15 montre l'évolution de l'erreur de la base de test pour différentes tailles d'architecture.

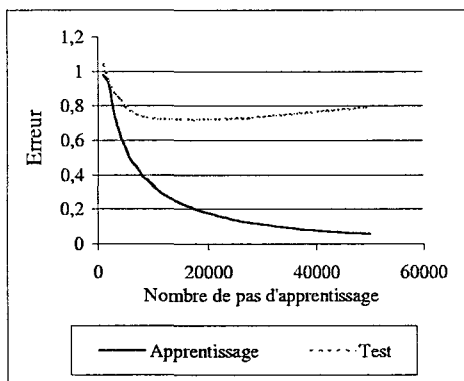


Figure II.14 : Performances du réseau PMC

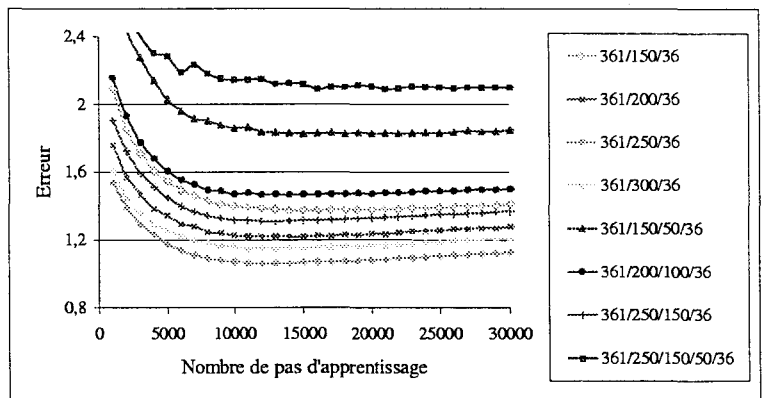


Figure II.15 : Influence du nombre de neurones cachés

L'étude montre que la performance au niveau de la classification des perceptrons est très dépendante du choix du nombre de couches ainsi que du nombre de neurones cachés. En effet, le

classifieur sera plus performant si on lui ajoute un grand nombre de couches cachées. Cependant, ceci contribue à alourdir fortement la structure du réseau, ce qui influe alors directement sur la performance concernant le temps de traitement.

Les PMC sont actuellement les réseaux les plus utilisés en classification de part leur ancienneté. Ce sont des réseaux qui restent très performants lorsqu'ils sont utilisés comme classifieurs. Cependant, ils sont limités de part la nature de leur règle d'apprentissage. Réaliser un classifieur disposant de nombreuses classes de décision demande une architecture importante de 5 couches avec un nombre de neurones sur les couches cachées proportionnel au nombre de classes. Cette architecture importante est très pénalisante pour le processus d'apprentissage puisqu'elle oblige à construire une importante base de données et que ses exemples seront soumis plusieurs fois au réseau afin qu'il puisse adapter correctement toutes ses liaisons synaptiques.

La difficulté de l'utilisation des PMC pour la classification est le choix du nombre de couches cachées et du nombre de neurones qui les composent. A part quelques règles comme celle énoncées précédemment, le choix est arbitraire ou expérimental. La principale conséquence est que le manque ou la redondance de neurones va dégrader les performances du classifieur. Des études récentes cherchent à minimiser le nombre de neurones redondants tout en préservant la qualité de la reconnaissance [JOU94].

II.3.3 Le réseau de Hamming

Le réseau de Hamming [LIP87] [PAO89] est un réseau permettant l'estimation de la classe d'appartenance d'un vecteur d'entrée en utilisant le calcul d'une Couche de sortie g. Ce classifieur est basé sur la notion de vraisemblance. Ce réseau est un réseau à compétition constitué de trois couches utilisant un apprentissage supervisé (figure II.16).

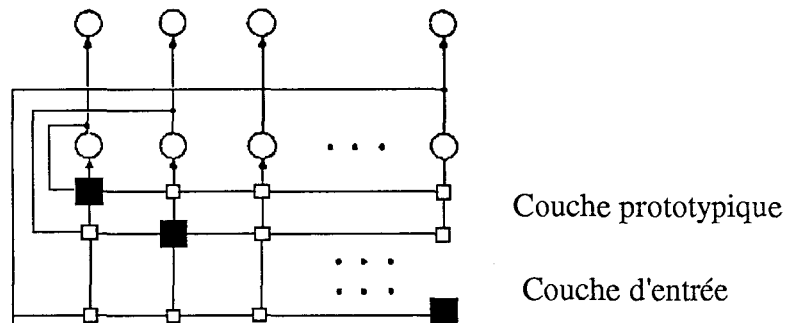


Figure II.16 : Le réseau de Hamming

La première couche est la couche d'entrée, la seconde couche est la couche prototypique (P). Chaque neurone (j) de cette couche représente un vecteur de la base d'apprentissage. Ce dernier est codé, sous forme neuronale, par les poids de connections (W_{ij}) entre le neurone et la couche d'entrée. A partir d'une couche d'entrée présentant au réseau le vecteur à classifier, cette couche de neurones va permettre de calculer la distance de Hamming séparant le vecteur d'entrée x et un prototype appris P_i . L'équation II.15 donne l'expression de cette distance dans le cas de vecteurs binaires et dont la dimension est N .

$$d_H(x, P_i) = \sum_k (x_k - P_{jk})^2 = N - \sum_k (x_k \cdot P_{jk}) \quad (II.15)$$

L'expression est réalisée sous forme neuronale par des connections entre les neurones de la couche d'entrée et ceux de la couche prototypique et par des interconnections entre les neurones de cette dernière (représentées figure II.16)

La dernière couche, dont les neurones sont interconnectés latéralement, est une couche à inhibitions latérales qui permet de ne conserver qu'un seul neurone actif, celui représentant la distance la plus faible.

Le réseau d'Hopfield est également un réseau basé sur l'obtention d'un minimum. Ce réseau d'architecture totalement connecté va minimiser l'énergie contenue dans les différents neurones. Dans le cadre d'un problème de classification, le réseau de Hamming, basé sur le minimum d'erreur se révèle plus simple tout en étant plus performant.

La figure II.17 représente l'évolution de l'erreur en phase d'apprentissage et en phase de test du réseau de Hamming sur les deux bases d'évaluation. Les courbes montrent les faibles performances de ce réseau lorsque l'espace à classer est de dimension élevée. Dans le cas d'images 25*25, la taille de la couche d'entrée est de 225 neurones ce qui est beaucoup trop important pour ce type de réseau.

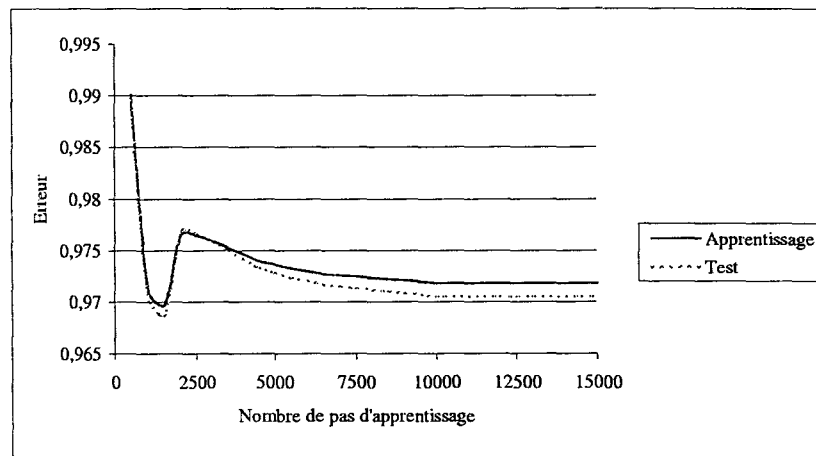


Figure II.17 : Performances du réseau de Hamming

II.3.4 Réseaux à fonctions de base radiale ou RBF

Dans le cadre de la classification, des réseaux de neurones ont été modélisés pour approximer des densités de probabilité. Ce sont les Radial Basis Function Networks ou réseaux à fonctions de base radiale. Ces réseaux sont aussi appelés classificateurs à noyaux car ils sont la modélisation neuronale des fenêtres de Parzen. Cette méthode des noyaux est une méthode d'approximation de densités de probabilité très connue en statistiques [POW87]. Contrairement au réseau PMC, le réseau RBF effectue une classification par modélisation. Le réseau va chercher à approximer les différentes régions représentant les différentes classes.

On rencontre les réseaux RBF pour la première fois dans les travaux de Broomhead et Lowe [BRO88]. La technique d'interpolation, nommée méthode d'interpolation RBF, utilisée par ce réseau lui permet d'approximer des densités de probabilité grâce à des fonctions-noyaux.

Une fonction noyau permet d'obtenir la réponse d'un ensemble de données de l'espace si celles-ci sont situées à l'intérieur du champ récepteur de la fonction. La fonction noyau est centrée sur un point de l'espace. A partir de ce point, un critère de distance permet de définir

l'influence des données sur le résultat de la fonction. Ainsi pour des données proches du centre C_i la fonction sera maximale. A l'inverse, cette fonction décroît de façon monotone avec l'éloignement des données.

La fonction la plus utilisée pour représenter une fonction-noyau est une gaussienne ψ . Elle est donnée par :

$$\psi(x) = \exp\left(-\frac{\|x - C\|^2}{\sigma^2}\right) \quad (\text{II.16})$$

Le critère de distance utilisé est une mesure de distance euclidienne. σ représente la taille du champ récepteur.

Le réseau RBF va utiliser des fonctions-noyaux centrées sur différents points de l'espace de représentation. Ces fonctions vont permettre de réagir localement à une région restreinte de cet espace. L'architecture d'un réseau RBF est constituée de trois couches (figure II.19). Une couche d'entrée une couche de sortie et la couche dite cachée constituée de neurones dont la fonction de transfert est décrite par l'équation II.16.

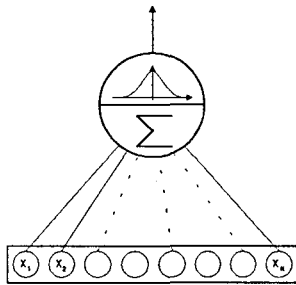


Figure II.18 : Le neurone RBF

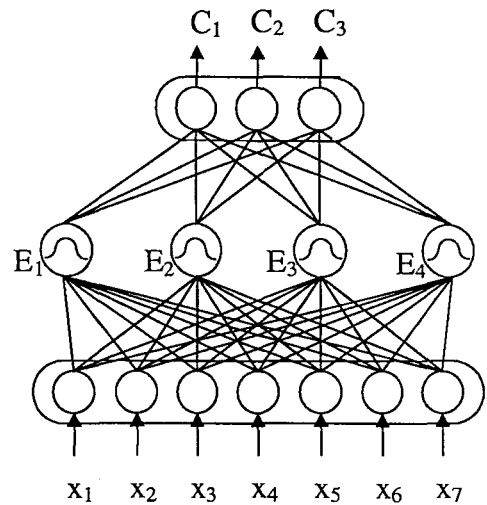


Figure II.19 : Le réseau RBF

Le centre de la fonction-noyau est symbolisé par les connexions synaptiques du neurone avec la couche d'entrée (figure II.18). Ainsi, si le vecteur présenté sur la couche d'entrée est proche du centre de la fonction, la sortie du neurone sera importante. A l'inverse, si le vecteur est éloigné, la réponse sera faible. D'une manière simple, il suffit de mémoriser dans les poids synaptiques une forme. Le neurone ainsi défini aura une réponse forte si un vecteur présenté est proche du centre caractérisé par ses poids. En choisissant un nombre suffisant de neurones RBF, chacun étant centré sur une forme à reconnaître, on réalise très facilement un classifieur.

La taille du champ récepteur est une variable interne du neurone et sa fonction d'activation est de la forme gaussienne. Le neurone ainsi obtenu est une bonne approximation d'une fonction-noyau.

Le nombre de neurones situés sur la couche cachée correspond au nombre de formes à mémoriser. Chacun d'eux permet d'estimer la densité de probabilité du vecteur d'entrée par rapport à la forme définie par leurs poids synaptiques.

La dernière couche permet de coder les résultats de cette couche cachée de telle manière que les différentes classes puissent être reconnues par des séparations linéaires. D'après les réponses obtenues par les fonctions noyaux, le traitement de la dernière couche ne consiste à

choisir que la neurone le plus actif de la couche cachée, celui-ci représentant la classe d'appartenance de la forme.

Ainsi, pour les réseaux RBF, la couche cachée calcule la distance entre l'entrée et les différents noyaux, la couche de sortie consiste à choisir la classe en fonction du neurone le plus actif dans la couche cachée.

L'apprentissage s'opère sur la couche cachée puis la couche de sortie. La première étape consiste à paramétrer les fonctions-noyaux sur les différentes régions de l'espace de représentation à caractériser puis à déterminer les connexions de la couche de sortie. Les connexions entre couche cachée et couche de sortie, vu la simplicité du traitement sont obtenues avec une technique d'apprentissage simple : règle du delta. La principale étape de l'apprentissage consiste à trouver la forme exacte des fonctions noyaux : c'est-à-dire trouver leur centre C_i et la taille du champ récepteur σ_i .

La figure II.20 représente les performances du réseau RBF. Tout comme le perceptron multicouche, les performances de ce réseau sont très dépendantes du choix de paramètres judicieux. Il s'agit de définir correctement la taille des champs récepteurs qui va permettre d'améliorer le fonctionnement en généralisation du classifieur (voir étude sur le PNN figure II.21). Le nombre de neurones à choisir sur la couche cachée est également déterminant pour la qualité du fonctionnement. Ainsi, pour une taille de la couche cachée variant de 400 à 950, le taux d'erreur sur la base de test pour 50000 pas d'apprentissage varie de 0,65 à 0,45. Cependant, ce paramètre est plus facile à choisir mais surtout nettement moins pénalisant dans le cas d'un mauvais choix par rapport au PMC. De plus, il est possible de définir le nombre de neurones cachés en fonction de la difficulté. Ainsi, pour des chiffres 0 et 1, la distinction étant facile peu de neurones cachés sont utilisés. A l'inverse, pour le 5 et le 8, il est possible de définir un plus grand nombre de neurones afin d'améliorer la distinction [BLU93].

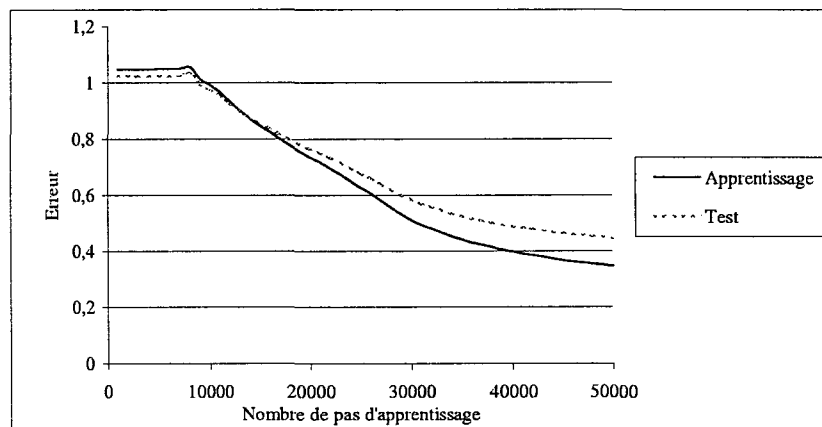


Figure II.20 : Performances du réseau RBF

Le réseau RBF est capable de calculs très puissants. Il se comporte comme le perceptron multicouche en approximateur universel. Ses performances en sont semblables voire supérieures et son apprentissage est plus rapide et plus simple. De plus, les réseaux RBF s'avèrent plus insensibles à la destruction de leurs poids.

II.3.5 Réseau de neurones probabilistes ou PNN

Depuis quelques années, l'intérêt de l'approche Bayésienne dans la modélisation neuronale se matérialise. Les premiers travaux intégrant la décision bayésienne aux systèmes neuronaux furent réalisés au début des années 90 grâce à Specht. Ce chercheur a présenté le réseau Probabilistic Neural Network (PNN) qui est l'implémentation neuronale de la méthode statistique dite classification bayésienne [SPE90]. Ce réseau fournit une solution générale pour la résolution des problèmes de classification en offrant l'accès à la statistique bayésienne. Il est basé sur l'estimation non-paramétrique des fonctions de densité de probabilité. Cette technique neuronale découle de l'estimateur de Parzen. Un critère de maximum de vraisemblance permet également ainsi d'estimer une densité de probabilité.

Le réseau PNN est un réseau à couches de type feed-forward.

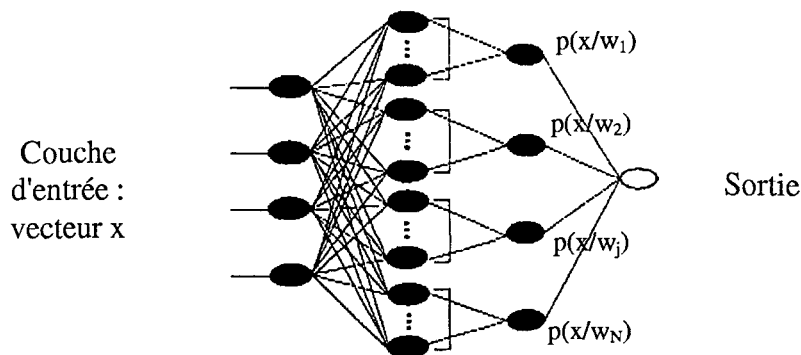


Figure II.21 : Le réseau PNN

Pour créer un réseau du type classifieur bayésien, on utilise des sous-réseaux constitués de neurones à noyaux. Les sous-réseaux utilisés sont des réseaux RBF. Ils permettent d'estimer les densités de probabilité conditionnelles du vecteur d'entrée par rapport aux différentes classes $p(x/C_i)$. Il y a donc autant de sous réseaux RBF qu'il y a de classes, et ce de manière à estimer toutes les densités de probabilité. Pour construire le classifieur bayésien, il suffit de prendre les sorties ou neurones ($p(x/H_i)$) de chaque sous réseaux RBF et d'en choisir le plus actif avec un réseau du type Winner Take All .

Contrairement aux algorithmes de rétropropagation des perceptrons multicouches, le PNN effectue un apprentissage extrêmement rapidement. Un autre point fort est la possibilité de modifier la frontière entre les différentes classes sans recommencer la période d'apprentissage.

Cependant, le PNN nécessite un grand nombre d'exemples pour une bonne généralisation. De plus, il est nécessaire d'utiliser tous les exemples de la base d'apprentissage (sous la forme d'un neurone de la couche cachée) pour estimer la densité de probabilité du vecteur d'entrée

Les performances de ce réseau sont très proches du réseau RBF. La principale différence entre ces deux réseaux concerne la phase d'apprentissage. Celle-ci est beaucoup plus rapide pour le PNN en raison de la structure de la couche cachée. Chaque neurone de cette couche correspond à un unique exemple de la base d'apprentissage, il n'a donc pas d'adaptation successive des poids entre la couche cachée et la couche d'entrée. Cependant, ceci pénalise le réseau PNN dans sa phase de généralisation. Ses performances sont alors uniquement dépendantes du choix de la taille du champ récepteur des neurones fonctions-noyaux. La figure II.22 illustre l'influence de la taille du champ-récepteur des neurones cachés (type fonctions radiales) sur les résultats du PNN.

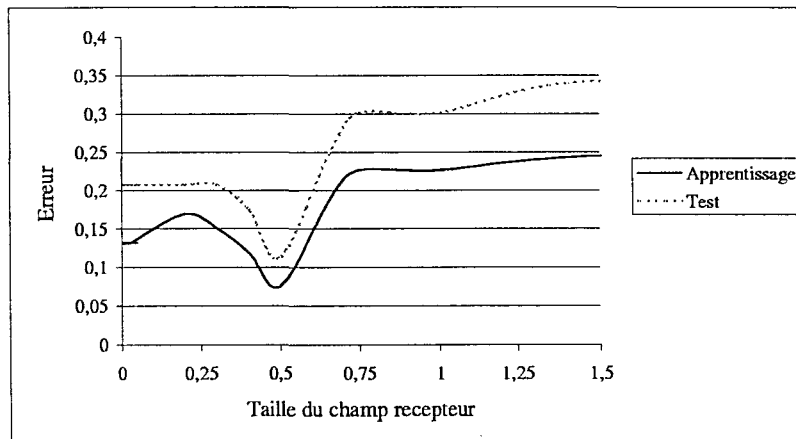


Figure II.22 : Performances du réseau PNN en fonction de la taille du champ récepteur

II.3.6 Les cartes topologiques auto-organisatrices ou CAO

Les Self-Organizing Maps (SOM ou CAO en français) ont été développées par T.Kohonen [KOH84]. Cette architecture neuronale a été réalisée en s'inspirant d'une structure proche du cortex auditif. Kohonen a pour cela introduit le concept de cartes topologiques auto-organisatrices. Grâce à des études neurobiologiques, de nombreux chercheurs ont montré les propriétés d'auto-organisation présentes dans le système nerveux des vertébrés (cortex visuel, cortex sensoriel, cortex auditif). Ainsi, les neurones du cortex auditif sont groupés selon les fréquences qui provoquent la plus grande réponse dans chaque neurone. Ils s'organisent, seuls, en réponse aux différents signaux entendus. Partant de cette constatation, des chercheurs en modélisation neuronale ont développé des algorithmes d'auto-organisation. Le modèle le plus connu est celui de Kohonen développé dans les années 1980.

En terme de classification, cette méthode s'apparente aux méthodes de clustering. Plus précisément, les cartes topologiques de Kohonen sont la modélisation neuronale de la méthode des nuées dynamiques. Le gros avantage de cette méthode, en dehors des propriétés topologiques, est d'être une méthode non supervisée, c'est à dire que la phase d'apprentissage peut s'effectuer de manière automatique sans influence extérieure.

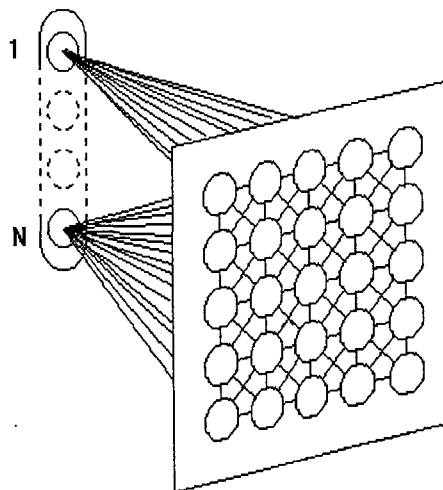


Figure II.23 : Le réseau de Kohonen

La topologie du réseau est très particulière : il est en effet formé de deux couches : une couche d'entrée et une couche de sortie représentée par une carte où tous les neurones sont connectés suivant le plan. Cette couche de sortie aussi appelée couche de Kohonen est une couche compétitive. Chacun des neurones de cette couche est connecté à tous les autres neurones de cette même couche. Les poids des connections entre ces neurones sont définis selon un critère de distance. Ce type de pondération est défini schématiquement sous le terme de chapeau mexicain (figure II.24) Ainsi, un neurone possède une action excitatrice "+" sur ses voisins proches et une action inhibitrice "-" sur les plus éloignés. Cette définition de la connexité entre neurones d'une même couche permet de définir la compétition. Ainsi, l'activité de la carte, lorsque l'on lui présente une entrée, est concentrée suivant une région. La propriété de cette carte topologique est qu'elle conserve les relations de voisinage de l'espace de représentation des échantillons.

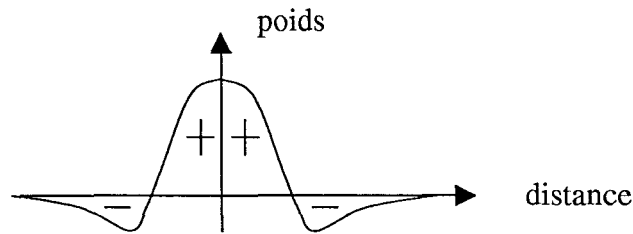


Figure II.24 : Chapeau mexicain

La figure II.25 représente les courbes d'erreur en phase d'apprentissage et phase de test obtenues sur nos bases d'évaluation.

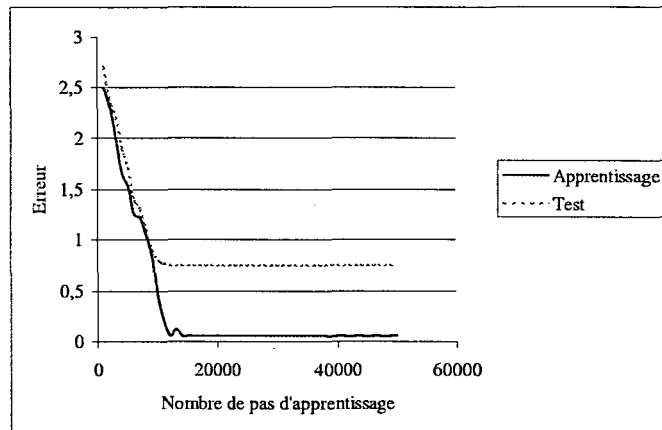


Figure II.25 : Performances du réseau CAO

Tout comme le réseau de Hamming, les performances sont limitées par la taille de la couche d'entrée. Idéalement, le réseau dispose d'une couche d'entrée beaucoup plus faible que sa couche cachée. Or chaque neurone caché correspond à un exemple de la base d'apprentissage. Il faudrait alors une base d'apprentissage trop importante pour obtenir des performances correctes.

L'utilisation de ces cartes en tant que classifieurs n'est pas optimale. En effet, les frontières inter-classes ne peuvent être correctement déterminées étant donné la structure de la couche [AUG93]. Plus récemment Kohonen a proposé des évolutions aux cartes auto-organisatrices ou topologiques. Il a introduit de nouvelles méthodes qui sont des versions supervisées des cartes SOM. Ces méthodes neuronales sont issues d'algorithmes de quantification vectorielle. Les réseaux présentés ci-dessous sont des classifieurs issus des méthodes de classification non-paramétriques.

II.3.7 Le Learning Vector Quantization ou LVQ

L'approche classique de la technique de quantification vectorielle consiste à diviser l'espace des données en plusieurs régions par un vecteur unique appelé vecteur de référence. Parmi les méthodes neuronales, la technique LVQ (Learning Vector Quantization) est la plus représentative.

Les techniques de Learning Vector Quantization figurent parmi les techniques compétitives les plus simples. Il s'agit de techniques dont le but est de déterminer la valeur optimale d'un ensemble de vecteurs prototypes. Ces vecteurs sont représentatifs des données à traiter et servent à la classification. Les vecteurs prototypes sont représentés par les poids d'entrée des neurones. L'apprentissage modifiant ces poids permet de retrouver les valeurs utiles pour ces prototypes. Le réseau et sa dynamique s'en voient extrêmement simplifiés. Le choix de la classe est effectué en comparant vecteur d'entrée et poids d'entrée des neurones

Le Learning vector quantization est un réseau de neurones décrit en 1988 par T.Kohonen [KOH88]. Ce réseau effectue une classification en estimant les frontières de décision plutôt que la distribution des classes. Le principe est d'utiliser une base d'exemples que l'on modifiera durant la phase d'apprentissage. C'est à partir d'une méthode utilisée en traitement du signal, que Kohonen s'est inspiré pour définir le LVQ [LEB89]. Cette méthode est la méthode "Vector Quantization".

II.3.7.1 Vector Quantization (VQ)

Cette méthode permet de produire une approximation de la distribution d'une classe pour une base d'exemples. Des vecteurs prototypiques sont positionnés afin d'approximer le mieux possible la répartition de données. La technique consiste à présenter au réseau tous les exemples d'un corpus d'apprentissage. Pour chaque exemple, on va modifier la base d'apprentissage en cherchant à adapter le modèle le plus proche de l'exemple. Dans l'architecture neuronale, on modélise un modèle par un neurone et le vecteur caractéristique de ce modèle est représenté par les poids de liaison entre ce neurone et les neurones de la couche d'entrée. Après avoir comparé l'exemple à tous les éléments de la base, on sélectionne celui le plus proche de l'exemple en choisissant un critère de distance.

$$\|x - w_c\| \leq \min_i (\|x - w_i\|) \quad (\text{II.17})$$

avec x : vecteur caractéristique de l'exemple

W_x : poids synaptiques du neurone x ou vecteur caractéristique du modèle x

Un fois le modèle sélectionné, on modifie les poids du neurone le plus actif de façon à rapprocher ses poids du patron qui l'a fait gagner.

La règle d'apprentissage se définit par : $\Delta w_c = \eta(t)(x - w_c)$ avec $0 < \eta(t) < 1$ pas d'apprentissage qui décroît avec le temps. La diminution du pas d'apprentissage permet d'éviter une non-convergence et donc de garantir une stabilisation des poids.

La phase d'apprentissage permet de positionner des vecteurs. Il suffit ensuite d'étiqueter ces vecteurs par le nom des classes qu'ils représentent. Pour ce faire, il suffit de choisir pour chaque neurone la classe la plus représentative des exemples de la base d'apprentissage associés à ce neurone. Nous obtenons ainsi une représentation de la distribution des classes sur tout l'espace.

Néanmoins en raison du choix d'un critère de distance euclidienne, cette méthode reste très sensible à la représentation métrique de l'espace des données. Ainsi, les performances de cette méthode seront fortement liées à la taille de la base d'apprentissage mais aussi à la dimension des vecteurs caractéristiques.

Kohonen s'est inspiré de cette méthode pour créer les réseaux de neurones "Learning Vector Quantization" qui sont des variantes supervisées du VQ et qui n'approximent pas non plus la distribution des classes mais leurs frontières [KOH90a].

II.3.7.2 Learning Vector Quantization (LVQ)

Le principe consistant à décrire une classe par un ou plusieurs vecteurs prototypiques est commun à plusieurs méthodes de classification : la méthode des Kppv, la fenêtre de Parzen, ou la technique du LVQ. Dans ces techniques, un vecteur prototypique représente un nuage de données d'une même classe. Il suffit donc pour classifier une nouvelle entrée de trouver le vecteur prototypique le plus ressemblant et de lui donner sa classe associée.

Le LVQ est basé comme le VQ sur le calcul des distances euclidiennes. A chaque exemple de la base d'apprentissage correspond un neurone et ses liaisons synaptiques avec le premier réseau correspondent aux composantes de son vecteur caractéristique. La procédure utilisée est identique à celle de VQ. On applique successivement des exemples (qui sont choisis dans la base d'apprentissage) au réseau. On sélectionne le modèle le plus proche de l'exemple puis on adapte les poids. Cependant, dans l'adaptation des poids, on introduit maintenant la notion de classe. Selon que l'exemple appartient à la classe du modèle ou non, l'adaptation ne sera pas la même.

$$\begin{aligned} \Delta w_c &= +\eta(t)(x - w_c) & (a) \\ \Delta w_c &= -\eta(t)(x - w_c) & (b) \end{aligned} \tag{II.18}$$

Ainsi lorsque la classe est identique, on applique une modification positive (a). L'exemple présenté va influencer dans sa direction le modèle du corpus. A l'inverse, si les classes sont différentes (b), l'exemple va éloigner le modèle. La fonction $\eta(t)$ reste une fonction monotone et décroissante.

Le principe du LVQ est donc de déplacer successivement les modèles de manière à faire apparaître les différentes frontières entre les différentes classes, les vecteurs s'éloignant de ces frontières. La disposition des vecteurs met en évidence des surfaces bayésiennes de décision (qui sont la définition statistique des frontières). Ces surfaces séparent les différentes classes et les probabilités d'appartenance à l'intérieur de ces surfaces sont identiques.

L'existence des différentes variantes a été motivée pour optimiser :

- soit le temps de l'apprentissage (LVQ1, OLVQ1)
- soit l'approximation des frontières (LVQ2, LVQ3)
- soit la classification en ajoutant une fonction bayésienne (Extended LVQ)

[KOH90b]

La figure II.26 représente les performances du réseau LVQ1 appliqué à notre base d'évaluation. Le réseau est doté de bonnes performances de reconnaissance. Ces principales limitations sont dues à l'apprentissage (en temps de calcul et non pas en pas d'apprentissage) et à un sur-apprentissage. L'apprentissage une fois réalisé, le réseau reste à une erreur de test assez élevée (0.5). Ce réseau est plus adapté à des problèmes simples ou les espaces de représentation sont de faible dimension et le nombre de classes est relativement peu élevé.

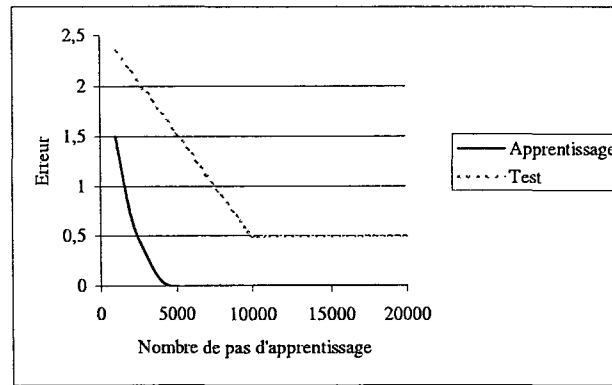


Figure II.26 : Performances du réseau LVQ

Conclusion de la deuxième partie

Ainsi la reconnaissance de caractères est un domaine de la reconnaissance de formes pour lequel de nombreuses techniques de classification ont été développées et expérimentées. La motivation a toujours été d'améliorer la qualité de la reconnaissance ainsi que les performances des systèmes. Notre étude montre que l'évolution des techniques de classification est liée à l'évolution du nombre de polices rencontrées. Après avoir commencé par des techniques de comparaison de matrice, la reconnaissance multifonte a imposé l'utilisation de techniques structurales. Celles-ci sont basées sur l'utilisation d'un classifieur estimant la classe d'appartenance à l'aide d'une représentation vectorielle du caractère. Ces techniques utilisent une étape d'extraction de caractéristiques de type structurel et/ou syntaxique. A l'issue de cette caractérisation, un classifieur détermine la classe en utilisant de critères simples de décision (distance, arbre, combinaison).

L'évolution fut poursuivie par l'introduction pour la reconnaissance de caractères de techniques de classification plus robustes. L'apparition de ces techniques dans les systèmes de reconnaissance a permis de discriminer des espaces de représentation plus importants. Ainsi, le nombre de caractéristiques extraites a pu être augmenté afin d'améliorer la qualité de la reconnaissance.

Dernièrement, l'apparition des réseaux de neurones a permis de simplifier la mise en œuvre et d'améliorer la classification dans le cas de problèmes complexes. Ces techniques neuronales ont pu remplacer certaines techniques statistiques et ont également apporté de nouvelles approches concernant l'approximation de fonctions discriminantes.

Bien qu'établies sur la base du connexionnisme et sur une étape d'apprentissage, toutes les techniques neuronales ne sont pas équivalentes et aussi performantes. Ainsi, des techniques comme le réseau à cartes auto organisatrices et le réseau LVQ se révèleront beaucoup plus efficaces dans le cas où les problèmes à traiter sont simples (dimension de l'espace de représentation et nombre de classes à distinguer). A l'inverse, afin de maîtriser correctement le processus d'apprentissage et de discriminer correctement l'espace de représentation, l'utilisation des réseaux à apprentissage supervisé sera nécessaire dans les cas complexes.

Le perceptron multicouches réalise une bonne approximation de fonction discriminante grâce à son algorithme d'apprentissage bien que son fonctionnement soit difficile à interpréter et qu'un grand nombre de paramètres influant sur la qualité de la classification soient peu significatifs à manipuler.

Les réseaux du type fonctions radiales ou probabilistes réalisent une bonne estimation de densités de probabilités tout en ayant des fonctionnements beaucoup plus maîtrisables.

Cependant, contrairement au PMC, leur capacité de généralisation est un peu moins bonne en raison du fait qu'il s'agit d'une classification par modélisation. Ils ne pourront donc pas classer un individu très éloigné des modèles appris. Dans le cadre de la reconnaissance de caractères et étant donné le nombre de classes à apprendre, ce cas reste rare et n'est donc pas pénalisant.

La figure II.27 regroupe les performances pour chacun des réseaux étudiés lorsqu'ils ont été appliqués à notre problème de reconnaissance de caractères. (Pour chaque cas, le choix de leurs caractéristiques a été effectué afin d'obtenir le meilleur taux de reconnaissance).

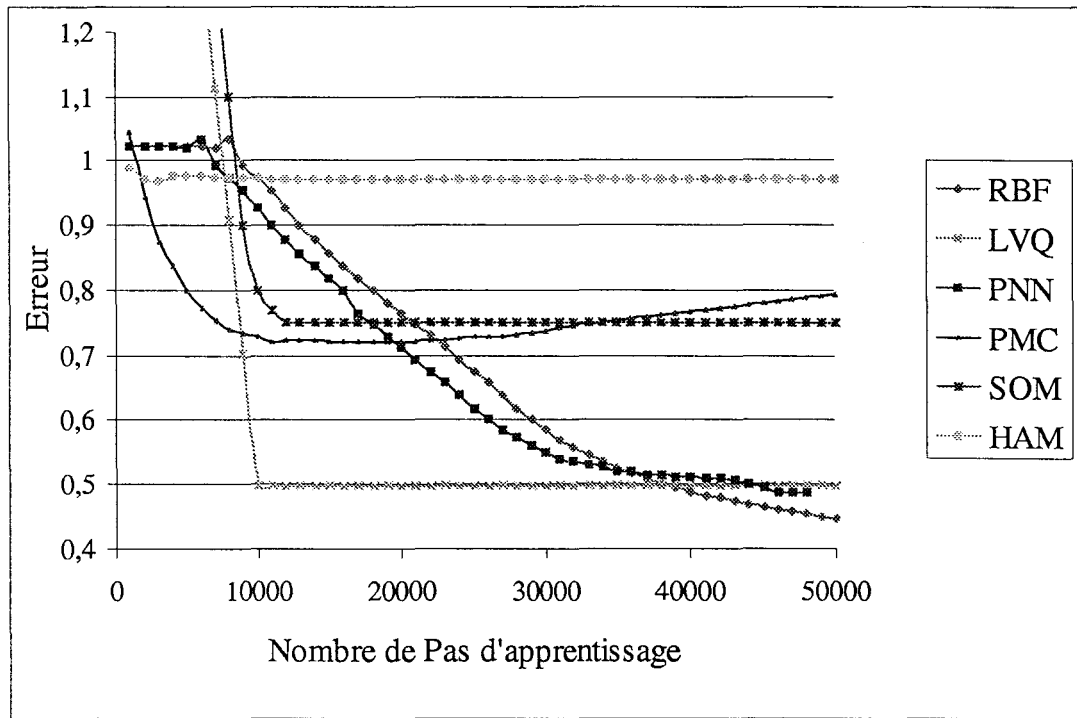


Figure II.27 : Comparatif des différents réseaux sur la base II.3.1

	Réseau PMC	Réseau HAM	Réseau RBF	Réseau PNN	Réseau SOM	Réseau LVQ
Modélisation	Séparation	Calcul de distance	Distribution gaussienne	Distribution Gaussienne	Calcul de distance	Quantification vectorielle
Complexité de l'architecture (en neurones)	1011	1312	2173	2236	1632	1523
Erreur de test minimum	0,72	0,97	0,31	0,36	0,75	0,49
Nb. de pas d'app. pour erreur mini.	15 000	10 000	80 000	70 000	11 000	10 000

Tableau II.2 : Tableau récapitulatif

En fonction des résultats représentés figure II.28 et du tableau récapitulatif II.2, des réseaux présentés, le RBF et le PNN semblent les mieux adaptés. Ceci est en accord avec de nombreux travaux comparant les techniques de classification par réseaux de neurones

Deuxième Partie : La classification

(comparaison PMC, LVQ, CAO [AUG93]; comparaison PMC, RBF, PNN [GRO93]; comparaison Hamming, PMC [BES95]; comparaison PMC, PNN [CHT97]).

Dans l'optique de la réalisation d'un système de reconnaissance qui doit avoir un temps de traitement suffisamment faible, notre étude a mis en évidence le réseau RBF. En effet, ce réseau dispose du meilleur compromis rapidité / qualité de reconnaissance par rapport au réseau PNN.

Etant donné la problématique de notre étude, nous utiliserons pour l'étage de classification de notre système un réseau de neurones du type fonctions radiales. En plus de discriminer un espace de représentation important, cette architecture nous laisse la possibilité de l'adapter plus aisément à notre problème.

Ce chapitre a montré les performances des réseaux de neurones pour la reconnaissance de caractères. Il a aussi montré la nécessité d'apprendre un grand nombre de caractères pour effectuer une reconnaissance multiforme. Dans notre étude, les caractères appris ont tous la même taille, or pour construire un système de reconnaissance global, il faut pouvoir traiter des caractères de taille variable. Pour cette raison, nous utiliserons un étage placé devant le classifieur qui va permettre de normaliser le caractère en position, en taille et en orientation. Il présentera un modèle invariant au classifieur et évitera donc l'utilisation d'une trop grande base d'apprentissage. Pour choisir la structure de cet étage, nous présenterons dans le troisième chapitre les principales techniques, permettant d'extraire des modèles invariants, qui peuvent construire cet étage.

Troisième partie :

L'extraction de
modèles invariants

Introduction

Les classifieurs présentés dans la deuxième partie ont la particularité d'être tous construits sur le même modèle : ils permettent d'attribuer la classe d'un vecteur d'entrée en fonction de son appartenance à une région (classe) définie, par expérience, dans l'espace de classification. Les premiers systèmes de reconnaissance furent basés sur l'utilisation de la représentation naturelle de l'objet (tout le signal, toute l'image). Dans ce cas, ce vecteur est constitué d'un grand nombre de composantes (identique au nombre de pixels de l'image ou au nombre d'échantillons du signal). Le nombre de modèles à utiliser pour l'apprentissage est également très grand étant donné les différentes représentations dues aux légères variations (aspect, dégradation, distorsion). La dimension de l'espace de représentation (chacune des composantes du vecteur représentant une dimension) et le grand nombre de modèles d'apprentissage, obligent à construire un classifieur très complexe et très performant.

L'objectif principal de ces techniques d'extraction dites "d'extraction de caractéristiques" est d'améliorer le fonctionnement du classifieur en simplifiant l'espace de représentation de la forme en ne sélectionnant que les informations (les dimensions) les plus discriminantes pour la distinction des différentes classes. Ces techniques permettent de construire un vecteur représentatif de la forme à classifier en effectuant des observations sur cette forme. Ce vecteur dit "vecteur caractéristique" est construit à partir d'analyse des caractéristiques locales de la forme et sur des méthodes de transformation de l'espace de représentation. Chacune de ses composantes correspond à une caractéristique particulière (singulière) de l'objet à étudier.

Dans les problèmes de classification, la principale difficulté est de reconnaître un modèle lorsque celui-ci a subi des modifications de représentation telles qu'une translation, une rotation ou une homothétie. Il est alors nécessaire de prendre en compte toutes les déformations d'un modèle dans la construction de la base d'apprentissage du classifieur et donc d'obtenir une base très importante. Ce problème montre l'avantage important d'utiliser les techniques d'extraction. En effet, celles-ci permettent d'effectuer une reconnaissance invariante ou reconnaissance PSRI (Position, Scale and Rotation Invariant) tout en limitant la base d'apprentissage du classifieur. Ces techniques extraient des caractéristiques qui sont indépendantes de la position, la taille et l'orientation de la forme. Toutes les variations de représentation de la forme sont ainsi caractérisées par un unique modèle dit modèle invariant. Cette invariance est obtenue à partir d'un ensemble de transformations et/ou de normalisations qui vont permettre de retrouver, quelle que soit la modification (ou la déformation) subie par l'objet, le même vecteur caractéristique et donc représenter la même classe d'appartenance. Ainsi, on élimine de la base d'apprentissage tous les modèles issus de variations de représentation dues au facteur d'échelle, à l'orientation et à la position (figure III.1).

De nombreuses techniques de reconnaissance invariante ont déjà été développées et sont généralement regroupées selon deux approches :

➤ La première consiste à séparer l'opération d'extraction de caractéristiques invariantes de la classification. A partir d'une image, on cherche à extraire de l'objet des caractéristiques générales et insensibles à son déplacement (translation, homothétie, rotation). Ces caractéristiques invariantes sont ensuite utilisées par le classifieur pour effectuer la tâche de classification.

➤ La seconde consiste à combiner les deux étapes en une seule. Cette méthode est la moins répandue et seuls quelques réseaux de neurones basés sur ce principe sont utilisés.

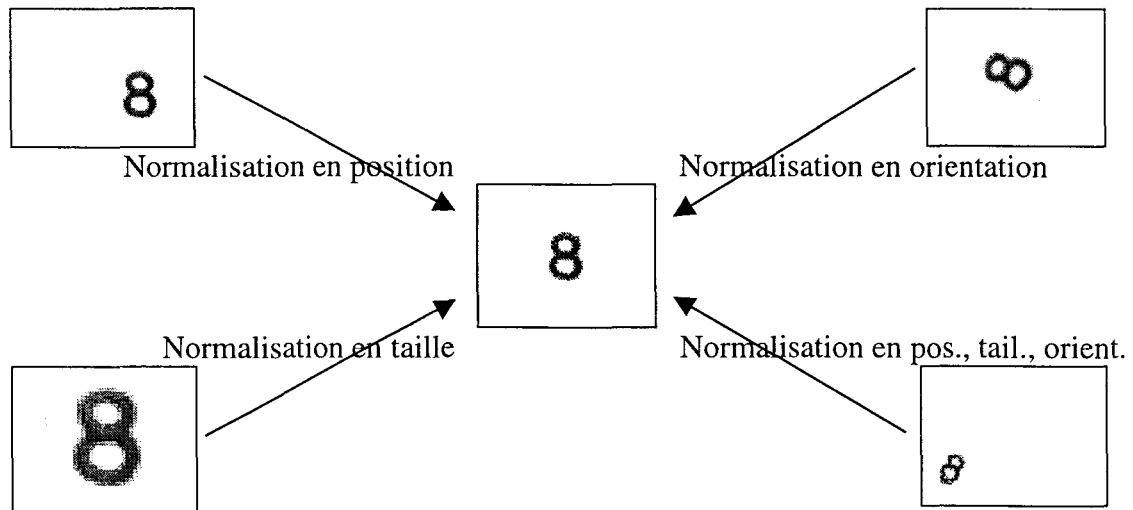


Figure III.1 : Notions de modèle invariant

La résolution du problème de l'invariance en position, en taille et en orientation est essentielle si l'on veut disposer d'un système de reconnaissance simple tout en étant robuste. Pour la constitution d'un système de reconnaissance de caractères, la notion d'invariance en taille doit être obligatoirement incorporée. Cette invariance va permettre de définir un modèle unique quelle que soit la taille de la police utilisée (cf. 1^{ère} partie).

L'invariance en position et en rotation ne pose généralement pas de problèmes pour la reconnaissance de documents scannés, les caractères sont généralement alignés. Pour ce type de documents, la difficulté est plus due à un problème de segmentation que de normalisation de caractères.

Dans notre cas, les propriétés d'invariance en rotation et en position sont plus utiles. Dans le cas de l'OCR appliqué au monde industriel, les caractères peuvent être considérés comme isolés car les indications telles que des codes sont généralement étendues. Le marquage s'effectuant parfois en mouvement, la localisation du caractère n'est pas précise. Le marquage peut également s'effectuer sur des supports circulaires (pignon, cylindre). De plus, la qualité des têtes de marquage et le déplacement des articles ne peuvent garantir le parfait alignement des caractères.

Ces différents aspects nous obligent à introduire dans notre système de reconnaissance, en plus de l'invariance en taille, des propriétés d'invariance en position et en orientation. Cette partie a pour but d'effectuer un état de l'art de la reconnaissance invariante appliquée à des classificateurs afin de présenter les solutions actuellement existantes applicables à notre problème.

Avant d'aborder directement la reconnaissance invariante, nous décrirons brièvement, les principales techniques d'extraction de caractéristiques utilisées en reconnaissance de caractères. Nous consacrerons le deuxième paragraphe à la présentation des principales techniques d'extraction de caractéristiques. Ceci nous permettra de connaître les notions de base avant d'aborder dans le dernier paragraphe l'utilisation de techniques neuronales pour la reconnaissance invariante.

A l'issue de cette partie, connaissant les atouts et les limites des techniques actuelles, nous allons pouvoir développer une nouvelle architecture neuronale capable d'effectuer une normalisation invariante.

III.1 Extraction des caractéristiques en OCR

Comme nous l'avons vu, dans la majorité des applications en OCR, il n'est pas nécessaire d'utiliser des techniques spécifiques permettant l'extraction de caractéristiques invariantes. Ce premier paragraphe présente quelques techniques classiques d'extraction utilisées en OCR. Le but est de se familiariser avec les notions générales d'extraction de caractéristiques et de donner des exemples appliqués aux caractères.

III.1.1 Généralités sur l'extraction de caractéristiques

A partir d'une représentation physique complexe (signal, image,...) d'un objet, l'étape d'extraction de caractéristiques permet d'obtenir un vecteur caractérisant cet objet. Ce vecteur est couramment obtenu à partir d'opérations de prétraitement puis d'analyses et de mesures (figure III.2). Cette étape est généralement difficile et délicate en reconnaissance de forme car il est nécessaire de sélectionner judicieusement les différentes opérations qui définissent les différentes composantes du vecteur. En effet, c'est en fonction du choix des caractéristiques et de leur nombre que les performances du classifieur vont varier. Ce choix est lié à la nature des objets à classifier.

Dans le cadre d'une reconnaissance de visage, les caractéristiques peuvent être liées à la physiologie de la personne (forme du visage, taille de la bouche, hauteur du front,...).

Dans le cadre de la reconnaissance de caractères, les caractéristiques sont liées à la présence de traits, de courbes, la hauteur, la largeur du caractère, sa graisse.

Dans le cadre de la reconnaissance de la parole, ces caractéristiques sont liées aux fréquences, à l'étalement du signal,...

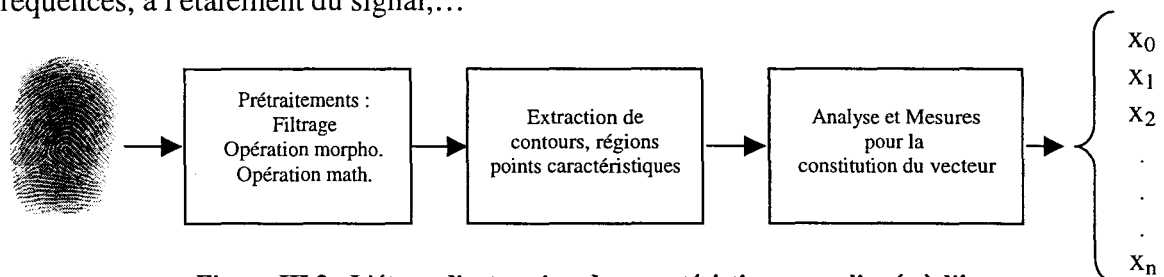
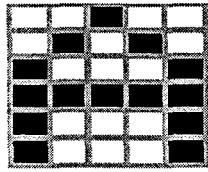


Figure III.2 : L'étape d'extraction de caractéristiques appliquée à l'image

C'est ce choix, judicieux ou non, des caractéristiques qui permettra d'améliorer le classifieur en le simplifiant et le rendant plus robuste. Le nombre de composantes (caractéristiques) du vecteur est fonction de la complexité des formes à identifier. Un grand nombre de composantes va permettre de distinguer mieux les différentes classes mais va compliquer le classifieur. A l'inverse un faible nombre va permettre d'obtenir une reconnaissance très rapide mais moins fiable dans la séparation des classes. Un très grand nombre de techniques d'extraction et de choix de caractéristiques a été mis en œuvre afin de tenter d'optimiser ce compromis. L'idéal est de choisir des caractéristiques très marquantes mais aussi les moins sensibles aux variations de représentation. Par exemple, il est beaucoup plus fiable de choisir la couleur des yeux pour distinguer différentes personnes que la taille de leurs pupilles qui varie en fonction de l'éclairage.

Initialement, les premiers algorithmes développés ont utilisé comme vecteur de caractéristiques, l'entière représentation physique de la forme (tous les pixels d'une image, toutes les fréquences dans un signal [REM85]). Le vecteur ainsi obtenu représente tout simplement le codage rétinien ou représentation naturelle. Dans le cas du traitement d'images, ce codage n'est

autre que la valeur des pixels de l'image. Ainsi, le vecteur V obtenu avec N pixels de l'image est de dimension N (figure III.3).



$$V = \{00100\ 01010\ 10001\ 11111\ 10001\ 10001\}$$

Figure III.3 : Représentation rétinienne

Ce codage extrêmement simpliste ne tolère aucune déformation ni aucun bruit. En effet, l'image d'un même caractère bruitée, translaturée ou inclinée n'aura pas le même codage. Le classifieur sera donc incapable de réaliser correctement sa tâche. Les systèmes basés sur la représentation rétinienne s'avèrent beaucoup trop lents et surtout très sensibles aux variations de représentation. Ils sont donc inefficaces pour effectuer une classification robuste.

L'introduction d'un étage de caractérisation des formes a donc permis de rendre les systèmes de classification plus efficaces aux transformations et aux dégradations. Par souci de synthèse, de nombreux auteurs ont proposé de classifier toutes les caractéristiques. De ces différents travaux, deux classifications concernant l'extraction de caractéristiques pour la reconnaissance de caractères ont été retenues par la communauté scientifique :

➤ La première a été proposée en 1979 par G. Gaillat et M. Berthod [GAI83]. Les auteurs ont tenté de regrouper les méthodes selon les différentes notions physiques utilisées. Cette classification a été remaniée en 1986 par J. Mantas [MAN86] et regroupe les différentes caractéristiques suivant sept classes :

- Comparaison globale
- Transformations globales
- Détection de particularités locales
- Recherche d'intersections avec des droites (verticales, horizontales,...)
- Analyse de concavités et des boucles
- Techniques structurelles

➤ La seconde classification fut proposée en 1980 par C.Suen, M.Berthod et S.Mori [SUE80]. Les groupes sont constitués à partir de trois types de caractéristiques :

- Caractéristiques extraites de la distribution des points
- Caractéristiques basées sur des critères topologiques et géométriques
- Transformations globales

Cette classification fut aussi révisée par divers auteurs dont V.K.Govindan et A.P.Shivaprasad en 1990 [GOV90].

Les caractéristiques tenant compte de la distribution des points sont obtenues à partir de méthodes d'analyse structurelle du caractère. Le caractère est décomposé en éléments simples ou formes primitives (points, droites, boucles, concavités,...). Ainsi, ces éléments simples et leurs relations décrivent un caractère. Ainsi le « d » est décomposé en un trait vertical et une boucle orientée à gauche. Ce sont les techniques qui permettent de mettre en œuvre les classifieurs les plus simples. Il suffit de vérifier si telle ou telle caractéristique est présente. Cependant, la difficulté reste de choisir correctement les primitives. Il faut s'assurer que deux modèles différents n'ont pas les mêmes caractéristiques. Ceci est très difficile à réaliser pratiquement dans le cas où il y a un grand nombre de classes à distinguer.

Les caractéristiques topologiques ou géométriques sont extraites des propriétés locales et globales du caractère (segment, courbe, point, boucle, épaisseur du trait). Ces caractéristiques

peuvent donc définir à la fois une propriété due à la distribution des points et une propriété topologique (les intersections, les distances,...). Des mesures très simples soit quantitatives (aire, périmètre moment), soit qualitatives (valeurs booléennes à l'aide de présence ou absence de propriétés de la forme) permettent d'obtenir ces caractéristiques. Ces techniques sont liées à l'analyse physique du caractère. Cette analyse est très efficace pour la reconnaissance de caractères d'imprimerie dans les applications bureautiques. En exagérant, un caractère va être caractérisé par le même vecteur quelles que soient sa police et sa graisse.

Les transformations qui sont extraites de l'étude globale de la forme visent à réduire la taille de l'espace à étudier mais aussi à obtenir des caractéristiques invariantes. Les principales méthodes utilisées sont les développements en série (Fourier, Walsh, Haar), les transformées de Fourier et de Hough, les moments géométriques. Ces techniques seront décrites plus précisément dans le paragraphe III.2 intitulé "Reconnaissance invariante par transformations globales".

Le paragraphe suivant présente quelques caractéristiques locales et topologiques couramment utilisées dans la reconnaissance de caractères d'imprimerie.

III.1.2 Exemples de caractéristiques liées aux caractères

Les techniques d'extraction de caractéristiques pour la reconnaissance de caractères hors-ligne sont basées sur l'analyse d'image. Nous n'aborderons pas dans cette partie le cas très précis de la reconnaissance manuscrite en ligne. Dans ce cas, des caractéristiques issues du stylet ou de la tablette (durée du trait, force, saut,..) peuvent améliorer la reconnaissance.

III.1.2.1 Caractéristiques locales

III.1.2.1.1 Intersections avec des droites [PAK85]

Les caractéristiques sont issues du nombre d'intersections entre la forme et différentes droites (figure III.4). Celles-ci peuvent être multiples : verticales, horizontales ou diagonales disposées à partir d'emplacements prédéfinis. Les caractéristiques peuvent aussi être obtenues en balayant l'image avec une seule droite. La position et le nombre des intersections sont les caractéristiques obtenues.

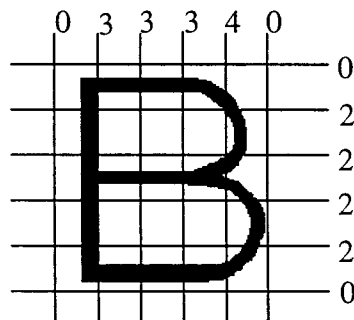


Figure III.4 : Exemple de caractérisation par intersections

III.1.2.1.2 Paramètres d'allongement

Les mesures de l'allongement d'une forme sont obtenues en projetant les pixels du caractère suivant une direction donnée. Dans le cas de projections horizontales et verticales, les profils obtenus vont permettre de caractériser la hauteur et la largeur du caractère (figure III.5).

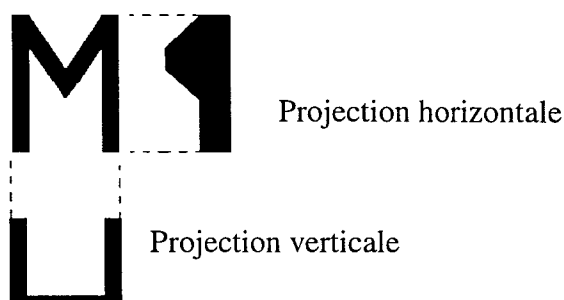


Figure III.5 : Projections horizontale et verticale

L'axe de projection est choisi sur un axe perpendiculaire à l'allongement que l'on veut caractériser. On peut ainsi caractériser aisément les allongements suivant la hauteur et la largeur.

III.1.2.1.3 Mesure de densité spatiale

On cherche à caractériser la forme suivant les différentes régions d'influence. L'image de l'objet est divisée en plusieurs zones suivant une grille (N×N). On détermine ensuite la valeur de la densité des pixels appartenant à la forme à l'intérieur de chaque zone (figure III.6). Chaque valeur est alors seuillée suivant l'importance de la densité. Chacune de ces valeurs calculées puis seuillées va définir une composante binaire du vecteur caractéristique. L'avantage de ces caractéristiques est de préserver l'allure de l'objet et de le rendre très insensible au bruit grâce au seuil. De plus, des propriétés de symétrie de la forme peuvent être aisément obtenues.

La difficulté réside dans le choix de la grille. Plus celle-ci est fine, plus le nombre de paramètres obtenus est important et plus l'extraction du vecteur caractéristique sera coûteuse en temps. A l'inverse, avec un découplage plus important, la dimension du vecteur sera plus petite mais on risque de perdre de l'information sur la forme à reconnaître.

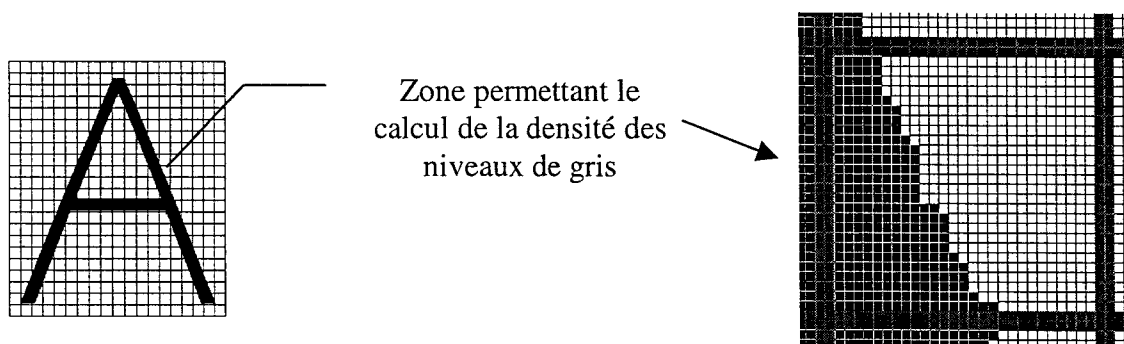


Figure III.6 : Mesures de la densité spatiale

III.1.2.2 Caractéristiques géométriques et topologiques

Ces caractéristiques sont basées sur les propriétés physiques de la forme. Les principales méthodes de caractérisation sont présentées dans les trois paragraphes suivants.

III.1.2.2.1 Détection de particularités locales

Il s'agit d'une analyse structurelle de la forme. Le but est de rechercher toutes les singularités locales qui décrivent la forme. L'analyse consiste à rechercher la présence de certaines formes élémentaires à l'aide de masques successifs de taille fixe (figure III.7). En déplaçant un masque sur la forme, on cherche à déterminer les endroits où l'on retrouve le motif du masque. Le motif à rechercher doit être le plus simple mais aussi le plus significatif possible : traits verticaux, horizontaux, courbes concaves, courbes convexes, courbes orientées, croisements, intersections «Y» et «T».

En raison de la rigidité des masques, cette technique peut s'appliquer facilement à des caractères imprimés. A l'opposé, une mise en œuvre sur de l'écriture manuscrite est difficile

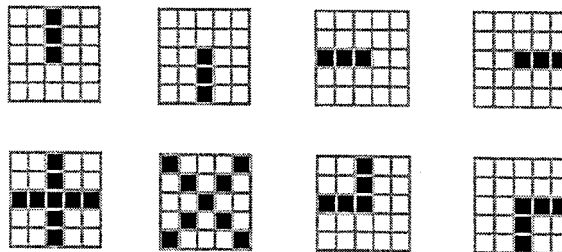


Figure III.7 : Exemples de masques utilisés

III.1.2.2.2. Calcul des pentes et des variations angulaires par polygonisation

La mise sous forme de polygones permet de réaliser une classification des primitives de type segment de droites [BEL92]. Chaque caractère est décomposé en un ensemble de segments de droite caractérisés par une taille, une orientation et un sens à partir d'une origine O.

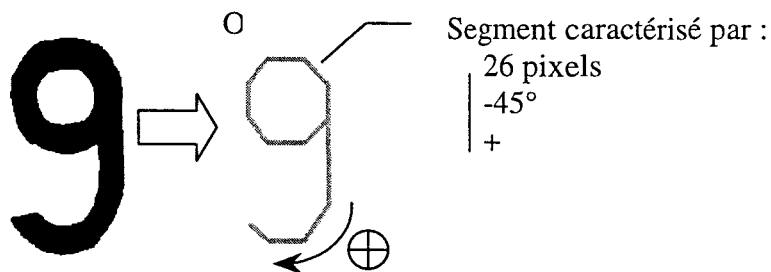


Figure III.8 : Polygonation d'un caractère

III.1.2.2.3 Analyse de concavités, de convexités et de boucles

Les caractéristiques obtenues par l'analyse de concavités et de boucles sont largement utilisées dans les algorithmes de reconnaissance. Par exemple, de nombreux critères ont été établis pour la reconnaissance de caractères manuscrits. En effet, l'avantage de ces caractéristiques est une grande insensibilité au bruit et aux déformations. Ces caractéristiques peuvent à la fois décrire des propriétés locales et globales de la forme.

Deux approches permettent l'étude des concavités. La première est de partir d'un point de l'image et de chercher à caractériser la concavité à partir de celui-ci (zones blanches). La seconde est de décrire la concavité à partir des contours de l'objet.

L'ensemble des techniques d'extraction de caractéristiques précédemment présentées a permis d'obtenir de nos jours des systèmes de reconnaissance de caractères d'imprimerie fiables et robustes. Cependant, dans le cas d'applications plus difficiles, c'est-à-dire en dehors de la bureautique, il est nécessaire d'utiliser des techniques qui puissent effectuer une reconnaissance en s'affranchissant de conditions d'acquisition très variées : bruit, position de l'objet par rapport à la caméra. Certaines caractéristiques, telles que les caractéristiques issues de la représentation statistique des points obtenues à partir de mesures quantitatives effectuées sur le caractère (distances, intersections, moments, n-tuples, lieux caractéristiques,...) ont des propriétés d'invariance. Cependant, ces mesures ne sont pas toujours fiables.

Des techniques dites de transformations globales sont plus performantes pour résoudre les problèmes liés aux variations de représentation des formes. Elles permettent de modifier l'espace de représentation de la forme en un espace présentant des propriétés d'invariance.

Le but du paragraphe suivant est de rappeler les principales techniques classiques dites "transformations invariante" qui ont été appliquées à l'OCR afin de pouvoir ensuite aborder les techniques neuronales.

III.2 Reconnaissance invariante par transformations globales

C'est principalement l'utilisation de transformations globales tels que les développements en série, les calculs intégraux, les moments algébriques qui permet de déterminer des caractéristiques invariantes grâce à un changement d'espace de représentation du codage rétinien. Toutes ont comme motivation d'obtenir une reconnaissance plus robuste, insensible aux bruits et/ou insensible à la position, à la taille et à l'orientation des formes.

III.2.1 Développements en série

Les caractéristiques obtenues sont issues d'une transformation globale de la représentation rétinienne du caractère. Cette transformation modifie donc le vecteur \mathbb{R}^N afin de changer l'espace de représentation. La transformation la plus répandue utilise les descripteurs de Fourier. Les descripteurs de Fourier ont été appliqués très tôt à la reconnaissance de caractères [BRI68]. Ils sont calculés à partir des coefficients des séries de Fourier. Ils ne se basent plus sur le vecteur rétinien mais sur les contours du caractère. Il est donc nécessaire d'effectuer un prétraitement de l'image de manière à en extraire les contours. Ainsi, pour un contour décrit par des coordonnées discrètes (x_m, y_m) de longueur Λ , les coefficients de Fourier $\alpha_{x,n}$ et $\beta_{x,n}$, selon la coordonnée x s'expriment par :

$$\alpha_{x,n} = \frac{T}{2\pi^2 n^2} \sum_q \frac{x_q}{\Lambda_q} \cos\left(n \frac{2\pi}{T} \Lambda_q\right) \quad \text{et} \quad \beta_{x,n} = \frac{T}{2\pi^2 n^2} \sum_q \frac{x_q}{\Lambda_q} \sin\left(n \frac{2\pi}{T} \Lambda_q\right) \quad (\text{III.1})$$

avec n : l'ordre du coefficient, q : point courant, Λ_q : distance entre l'origine du contour et la position courante (q) et T : périmètre.

C'est à partir de ces coefficients que l'on obtient le descripteur de Fourier F_d :

$$F_d(n) = \sqrt{\alpha_{x,n}^2 + \alpha_{y,n}^2 + \beta_{x,n}^2 + \beta_{y,n}^2} \quad (\text{III.2})$$

La figure III.9 représente, pour un exemple de caractère, le contour reconstruit à partir d'un nombre donné d'harmoniques.

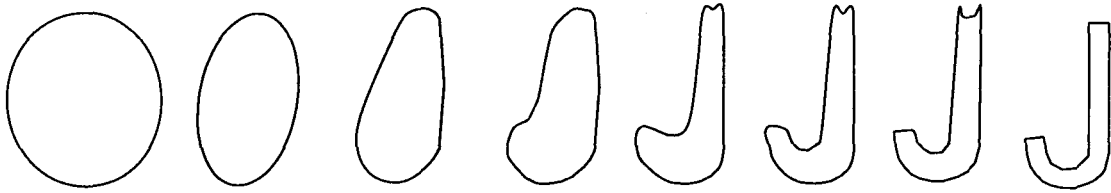


Figure III.9 : Exemples de descripteurs de Fourier d'ordre 1 à 35 pour un caractère

On construit généralement le vecteur de caractéristiques en prenant comme composante l'amplitude des différentes harmoniques.

Les descripteurs de Fourier possèdent des propriétés d'invariance en translation et en taille. Ainsi, seule l'harmonique de rang 0 (déterminant le centre de gravité) est sensible à la position du caractère. Une rotation du caractère n'influence que la phase du descripteur.

La limite à l'utilisation des descripteurs de Fourier est due au prétraitement et à l'extraction du contour. Là, le bruit peut générer des contours fantaisistes. Leur utilisation est alors inutile.

III.2.2 Les invariants intégraux

III.2.2.1 La transformée de Fourier

La transformée de Fourier est historiquement la première technique basée sur le calcul intégral qui a permis d'effectuer l'invariance en taille et en rotation. L'avantage de la transformée de Fourier est d'être très couramment utilisée dans de nombreux domaines d'application. Cette technique est aujourd'hui très facilement réalisable. Des évolutions ont permis de l'adapter aux différents domaines de représentation des signaux (transformée monodimensionnelle continue, discrète; transformée bidimensionnelle continue, discrète; transformée rapide, ...). La transformée bidimensionnelle discrète est bien évidemment la plus adaptée à la représentation rétinienne de l'image. Elle s'exprime, pour une image $f(i,j)$ de taille $N \times N$, par :

$$F(k,l) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) e^{-j2\pi \left(\frac{ki}{N} + \frac{lj}{N} \right)} \quad (\text{III.3})$$

On peut utiliser la transformée de Fourier monodimensionnelle sur chacune des coordonnées des contours plutôt qu'une transformée bidimensionnelle sur toute l'image. Ceci correspond alors aux descripteurs de Fourier.

De nombreuses études appliquant la transformée de Fourier à la reconnaissance des caractères ont montré les limites de cette technique. La principale est liée à l'invariance en rotation car elle n'est pas suffisante pour distinguer tous les caractères (ex : "6" et "9"). Cette technique doit également faire appel à des classifieurs performants car l'espace de représentation donne des représentations spectrales de caractères difficiles à discriminer.

III.2.2.2 La transformée de Mellin

La transformée de Mellin 2D est définie par :

$$z(w_1, w_2) = \int_0^{\infty} \int_0^{\infty} f(x_1, x_2) x_1^{iw_1-1} x_2^{iw_2-1} dx_1 dx_2 \quad (\text{III.4})$$

Une variante existe sous le nom de la transformée de Fourier-Mellin. Cette adaptation permet d'obtenir à la fois une invariance en translation mais aussi en rotation [SHE91].

III.2.2.3 La corrélation croisée

La technique de corrélation croisée d'une forme f et d'un modèle t s'exprime par :

$$C(g) = \int_s f(x)t(g^{-1}x)dx \quad (III.5)$$

Il s'agit ensuite de faire varier g , représentant soit l'opération de translation, d'homothétie ou de rotation tel que $C(g)$ est maximum. Dans ce cas, gt (la transformation du modèle t par la fonction g) approxime le mieux f .

La technique de la corrélation triple permet d'obtenir l'invariance en position en taille et en orientation. Cette technique est une généralisation de l'autocorrélation.

$$C_3(a,b) = \frac{1}{T^n} \int_s f(x)f(x+a)f(x+b)dx \quad (III.6)$$

avec a et b représentant des vecteurs déplacements.

Cette technique ne reste cependant envisageable que d'un point de vue théorique et demeure difficilement réalisable sur des images réelles car, comme pour le template matching, cette technique nécessite un très gros coût de calcul.

III.2.3 Les invariants algébriques

Ils permettent de conserver les propriétés d'invariance en rotation, en translation et en taille d'une forme. Les invariants sont obtenus à partir de moments calculés sur des images binaires. Les moments ont été appliqués la première fois par Hu en 1962 [HU62]. Depuis, de nombreuses recherches ont été menées sur les méthodes de calcul de ces moments. Il a été également montré que les moments invariants sont une forme spécifique des descripteurs de Fourier Mellin [SHE96].

Les principaux moments utilisés en reconnaissance invariante sont les moments géométriques, les moments centraux et les moments de Zernike.

La définition pour un moment d'ordre (p,q) est donnée par :

$$m_{pq} = \sum_{x=1}^M \sum_{y=1}^N x^p y^q f(x,y) \text{ pour } p,q = 0,1,\dots,\infty \quad (III.7)$$

avec $f(x,y)$: représentation mathématique de l'image binaire

Le moment m_{00} représente la surface du caractère et les moments d'ordre 1 permettent la détermination du centre de gravité (x_g, y_g) du caractère.

$$x_g = \frac{m_{10}}{m_{00}} \quad \text{et} \quad y_g = \frac{m_{01}}{m_{00}} \quad (III.8)$$

La définition d'un moment invariant en translation d'ordre (p,q) est la suivante.

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^M (x - x_g)^p (y - y_g)^q f(x,y) \quad (III.9)$$

Ces moments s'appellent également moments centraux.

Ainsi, les moments centraux d'ordre 2 (μ_{20} , μ_{02} , μ_{11}) permettent de mettre en évidence les axes principaux du caractère, son allongement horizontal, vertical et l'orientation générale de la forme.

Comme les descripteurs de Fourier, l'utilisation des moments permet de reconstruire une image normalisée. La figure III.10 représente pour un exemple de caractère, des images reconstruites en fonction de l'ordre des moments utilisés.

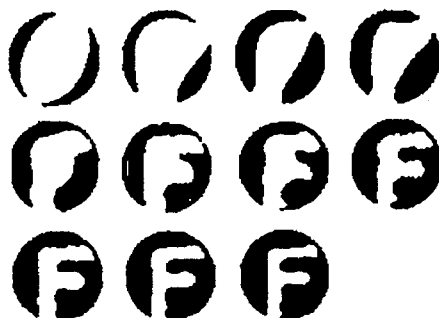


Figure III.10 : Moments algébriques d'ordre 1 à 11 appliqués à un exemple de caractère d'après [HEN96]

C'est à partir de combinaisons de ces moments centraux d'ordre 2 et 3 que Hu a défini sept moments invariants en taille, en orientation et homothétie. Ces moments invariants ont été appliqués à la reconnaissance de caractères imprimés. Les caractéristiques ainsi obtenues sont moins sensibles aux diverses déformations. L'avantage des moments réside dans la simplicité de leur calcul.

Cependant, à partir d'un certain ordre, les moments sont très sensibles au bruit ce qui oblige à n'utiliser que les moments d'ordre les plus bas. Des études actuelles cherchent à résoudre ce problème en modifiant le concept de moments d'ordre positif et entier en y incluant des moments d'ordre négatif et d'ordre décimal [GRU97].

Un autre problème est la détermination du nombre de moments à calculer. L'utilisation des moments de Zernike [KHO90a] a permis d'éliminer ce problème grâce à leur propriété d'orthogonalité qui permet une reconstruction du caractère. Ils sont basés sur un ensemble de polynômes complexes qui forment une base orthogonale à l'intérieur du cercle unité. Ces derniers semblent plus performants que les moments géométriques. Des études ont montré leur supériorité (80 à 90 % de reconnaissance avec des caractères bruités contre 40 %) [JIA91].

Nous avons limité notre étude aux invariants 2D. Certaines techniques cherchent à reconnaître l'objet quelles que soit sa position, sa taille, son orientation mais également quel que soit l'angle de vue. Celles-ci portent sur l'obtention d'invariants 3D. Les recherches sont essentiellement développées pour le suivi de cible et la vision embarquée [HEN94][SHE97].

Pour notre problème, l'obtention d'invariants 3D, beaucoup plus complexes à déterminer que les invariants 2D, n'est pas justifiée. On peut en effet supposer que l'axe de prise de vue est toujours perpendiculaire à l'objet. L'objet étant généralement inspecté lors de déplacements sur un convoyeur ou directement sur une palette.

N'ont été exposées, ici, que quelques méthodes de transformation invariante permettant d'aborder par la suite la présentation des techniques neuronales utilisées en reconnaissance invariante. Dans la littérature, de nombreuses autres techniques ont été proposées. Ainsi, les développements séries de Walsh, Haar, Hadamard, le développement de Karhunen-Loeve, la transformée de Hough ont été appliqués pour la reconnaissance de caractères. Toutes ces techniques ont l'avantage d'avoir des propriétés invariantes obtenues par modification de l'espace de représentation de la forme [ROT96][WOO96].

Ces invariants vont permettre la reconnaissance invariante soit en effectuant une normalisation d'image (les moments de Zernike) afin d'obtenir une nouvelle représentation du modèle, soit directement en les combinant afin de déterminer un vecteur de caractéristiques invariante, directement utilisés par le classifieur (les moments invariants de Hu). Des études récentes [RHO96][SHE97] tendent à montrer que la normalisation d'image est plus avantageuse, son principal avantage étant d'obtenir un modèle standard de la forme et non pas une représentation codée.

Les réseaux de neurones ont permis d'apporter de nouvelles solutions aux problèmes liés à la reconnaissance invariante. En effet, ils peuvent permettre de s'affranchir d'un grand nombre de caractéristiques grâce à certaines règles d'apprentissage, soit en permettant de définir un nouvel espace de représentation invariant, soit en effectuant directement une classification avec des règles liées à l'invariance. Ces trois aspects vont être décrits dans le paragraphe suivant. Son intérêt est de parcourir les différentes techniques d'invariance qui sont susceptibles de s'associer avec notre classifieur déterminé dans la deuxième partie. Ceci nous permettra de définir les caractéristiques de la partie invariance de notre système.

III.3 Méthodes neuronales et reconnaissance invariante

Les réseaux de neurones peuvent être utilisés selon trois approches pour la reconnaissance de formes invariantes [BAR91][GUP94][BIS95].

➤ La première, la plus simple, est basée sur l'augmentation de la base d'apprentissage d'un réseau classifieur afin d'y introduire des modèles déformés. Cette approche est basée sur les capacités d'apprentissage et les capacités de généralisation des réseaux de neurones à apprentissage supervisé [SIM93][SCH96a].

➤ La deuxième approche consiste à combiner les méthodes classiques d'invariances (utilisant des transformations mathématiques: les moments, les descripteurs de Fourier) précédemment décrites et un classifieur de type neuronal [BLU91][AUG93].

➤ La troisième concerne l'utilisation d'architectures neuronales spécialement conçues pour cette tâche [BIE87][FUK88][FUK91][SRI92b][LAD93][TAK94].

III.3.1 Rôle de l'apprentissage

L'intérêt de l'utilisation des réseaux de neurones est qu'ils s'adaptent à un nouveau type de comportement simplement à l'aide d'une phase d'apprentissage. Simplement en présentant des modèles à reconnaître et leur classe correspondante, le réseau va résoudre le problème de classification. A partir d'un réseau classifieur basé sur la représentation rétinienne des caractères, on peut réaliser simplement une reconnaissance invariante. Il suffit pour cela d'ajouter dans la base d'apprentissage des modèles présentant des variations de position, de taille ou d'orientation. Généralement, on génère ces modèles déformés à partir des modèles de la base initialement constituée en leur faisant subir de nombreuses transformations telles qu'une translation, une rotation et/ou une homothétie. Ces nouveaux modèles ainsi appris permettent de réaliser un système de reconnaissance invariante. Ceci est rendu possible en tirant profit des capacités d'apprentissage et de généralisation des réseaux de neurones.

Le désavantage de cette technique est évident. Il est nécessaire de créer une base d'apprentissage très importante voire démesurée. Cette limite impose qu'un tel réseau ne pourra traiter que des modèles présentant des variations proches de celles apprises. En effet, pour résoudre l'invariance on doit connaître a priori les déformations à reconnaître. Cette solution ne peut s'appliquer que pour un problème de classification avec un faible nombre de classes mais également avec des modèles subissant des transformations faibles.

Bien qu'actuellement, cette solution semble inabordable, l'évolution des processeurs permettra, peut-être un jour, de la réaliser facilement en disposant de puissances de calcul importantes lui permettant d'apprendre toutes les possibilités de combinaisons.

En 1992, une technique dite "tangent prop" [SIM92] a permis d'introduire une reconnaissance invariante aux faibles variations sans la nécessité d'augmenter la base d'apprentissage. Le réseau qui a servi de support à cette technique est le réseau à couches PMC. La règle d'apprentissage a été modifiée afin de permettre l'incorporation automatique de connaissances grâce à la phase d'apprentissage. L'idée est d'augmenter artificiellement la base d'apprentissage en présentant au réseau des modèles dérivés de ceux de la base initiale. Cette technique modifie l'algorithme de rétropropagation en y introduisant la notion de vecteur tangent.

Soit $M_{x,t}$: l'ensemble décrit par les modèles obtenus par une transformation t sur un modèle initial x et pour un ensemble de valeur du paramètre p caractérisant la transformation.

$$M_{x,t} = \{x' | p \in [p_1, p_2] : x' = t(x, p)\} \quad (III.10)$$

On suppose cet ensemble représenté par une courbe paramétrique $C(\alpha)$ continue dans l'espace des représentations. Pour illustrer cette notion, la figure suivante (figure III.11) schématise deux ensembles de modèles obtenus par rotation d'un caractère avec un angle variant de -5° à 5° et par une homothétie avec un rapport variant de 0,8 à 1,2.

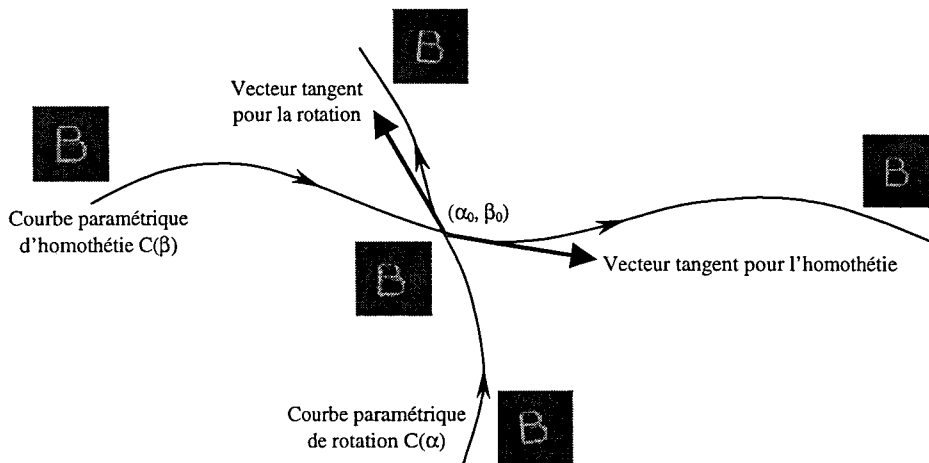


Figure III.11 : Exemples de vecteur tangent d'après [SCH96]

Le vecteur tangent τ^n est défini par la dérivée de la courbe C . Il est représenté par une tangente à la courbe au point α_0 . Ainsi, le vecteur tangent pour un vecteur x^n (n représente la dimension du vecteur) s'exprime par :

$$\tau^n = \left. \frac{\partial t(x^n, \alpha)}{\partial \alpha} \right|_{\alpha=\alpha_0} \quad (III.11)$$

La particularité de cette technique est de ne pas modifier les modèles à l'entrée du réseau mais de modifier la sortie désirée du réseau et plus précisément en modifiant le calcul de l'erreur quadratique calculé à la sortie du réseau entre la sortie réelle et la sortie désirée. Au calcul d'erreur classique (exprimé II.12) s'ajoute le terme suivant :

$$\Omega = \frac{1}{2} \sum_n \sum_k \left(\sum_i J_{ki}^n \tau_i^n \right) \quad (III.12)$$

avec J la matrice jacobienne du réseau multicouche [BIS95].

P.Simard et al. [SIM92] ont appliqué cette technique à la reconnaissance de caractères. La première expérimentation de cette technique a montré que par rapport à l'algorithme de rétropropagation du gradient les performances étaient identiques avec une base d'apprentissage environ deux fois moins importante.

Une technique également basée sur la notion de vecteur tangent, a permis l'amélioration du classifieur du type k-ppv [SIM93]. Cette amélioration a été obtenue en modifiant la distance euclidienne extrêmement sensible aux variations de représentation rétinienne par la *distance tangente* [SCH96a]. Soient deux modèles A et B, cette nouvelle distance s'exprime alors par :

$$D(A, B) = \min_{\alpha\beta} \|t(A, \alpha) - t(B, \beta)\|^2 \quad (\text{III.13})$$

avec $t(A, \alpha)$ et $t(B, \beta)$ les modèles obtenus par transformation avec α et β décrivant tous les modèles possibles.

Ce nouveau type de calcul de distance a montré de bonnes performances. Appliqué à la reconnaissance de caractères manuscrits, le taux d'erreur n'est que de 2.6%.

Récemment, la distance tangente a été utilisée comme fonction coût d'un réseau auto-associatif. Cette adaptation, [SCH96b], utilise un réseau diabolo par type de classe à distinguer. Avec un vecteur d'entrée de type codage rétinien, ce réseau peut par une simple descente de gradient apprendre les différentes invariances. Le taux de reconnaissance sur la base NIST est de 2.55%.

Cependant, les techniques liées à l'utilisation de vecteur tangent ne réalisent pas une véritable reconnaissance invariante. Elles permettent simplement d'éviter les distorsions liées à l'utilisation d'une représentation naturelle et ne peuvent effectuer une reconnaissance si les modèles présentés ont subi de fortes variations.

III.3.2 Combinaison avec des techniques invariantes classiques

La seconde approche pour réaliser une reconnaissance invariante par réseaux de neurones est tout simplement d'utiliser un classifieur basé sur une architecture neuronale et un étage de prétraitement qui va extraire des caractéristiques invariantes telles des moments géométriques, les descripteurs de Fourier. L'invariance n'est pas réalisée directement par le réseau de neurones. Simplement, on utilise ses capacités d'adaptation. En effet, on a montré, dans la 2^{ème} partie, que les classifieurs neuronaux sont utilisés comme classifieurs numériques (ils découlent de la classification statistique). Ainsi, on peut très facilement appliquer les résultats issus de calculs de moments, des coefficients de Fourier,... à l'entrée du classifieur neuronal.

De nombreux travaux ont été faits dans ce sens. La plupart d'entre eux sont essentiellement des combinaisons entre des techniques permettant le calcul de moments géométriques, centraux ou Zernike ou des coefficients de Fourier et des perceptrons multicouches [KHO88].

Nous ne détaillerons pas ces travaux car ils ne sont pas caractéristiques de la capacité des réseaux à effectuer par eux-mêmes la reconnaissance invariante. Nous ne présenterons dans ce paragraphe, qu'une approche originale (le modèle de Yüceer) qui combine classifieur neuronal et étage de normalisation par blocs. En effet, bien que cet étage de normalisation ne soit pas réalisé par un modèle neuronal, son principe, d'utiliser des successions de blocs est proche des architectures neuronales qui effectuent le traitement par succession de couches.

III.3.2.1 Le modèle de Yüceer

Le système de reconnaissance de formes présenté par Yüceer en 1993 [YUC93] fut développé pour permettre la reconnaissance invariante de formes. Ce système est capable

d'effectuer la reconnaissance de formes ayant subies des translations, rotations ou/et homothéties. Ce système est constitué de deux étages, l'un défini comme "pré-processeur forme", l'autre comme classifieur (figure III.12).

Le premier étage permet de reconstruire une représentation invariante de la forme à classifier par normalisation des formes présentées.

Le classifieur, issu de techniques neuronales, va à partir de cette représentation déterminer la classe d'appartenance.

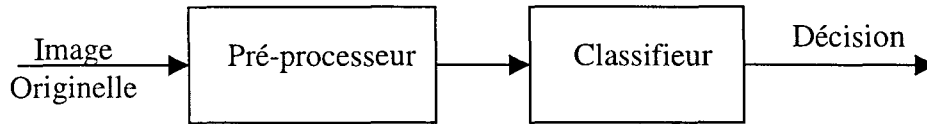


Figure III.12 : Modèle de Yüceer

Le pré-processeur permet d'obtenir une représentation stable de formes déformées ou bruitées. La principale tâche de cette étape de prétraitement est d'effectuer une invariance en position, en taille et en orientation. La modélisation du pré-processeur est obtenue, par la volonté des auteurs, en adaptant des propriétés simples. Pour ce faire trois principaux blocs ont été définis et placés en cascade (figure III.13).

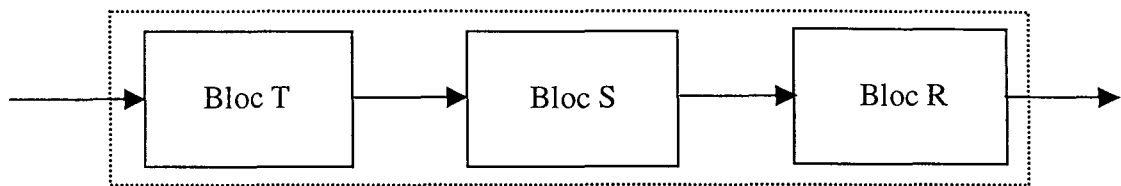


Figure III.13 : L'étage préprocesseur

Le premier bloc défini est le bloc T. Celui-ci permet de réaliser l'invariance en position en modifiant l'image originale de manière à faire coïncider le centre de gravité (x_g, y_g) de celle-ci avec le centre de l'image. Ce bloc réalise une transformation d'invariance en translation à l'aide d'une fonction géométrique simple f_T .

$$f_T(x, y) = f(x + x_g, y + y_g) \quad (\text{III.14})$$

Le second bloc, bloc S, permet d'effectuer l'invariance en taille. La fonction obtenue f_{TS} permet en appliquant un facteur d'échelle s de modifier l'image issue du bloc T. Ce facteur d'échelle est déterminé à partir de la taille de la forme et d'une taille désirée.

$$f_{TS}(x, y) = f_T(s.x, s.y) \quad (\text{III.15})$$

Le troisième et dernier bloc est le bloc R. Ce bloc utilise les propriétés de la transformation de Karhunen-Loeve [YUC92]. En effet, il s'agit de déterminer la direction de la variance maximum de la forme. L'image est orientée de manière à mettre en correspondance cette direction avec l'axe horizontal. La fonction décrivant l'invariance en rotation s'exprime par :

$$f_{TSR}(x, y) = f_{TS}(\cos(\theta).x - \sin(\theta).y, \sin(\theta).x + \cos(\theta).y) \quad (\text{III.16})$$

avec $\sin(\theta)$ et $\cos(\theta)$ définis à partir de la matrice de covariance, puis de la dérivée du vecteur propre issu de la plus grande valeur propre.

Ce sont donc ces trois blocs mis en cascade qui vont permettre d'obtenir une représentation invariante de la forme à classifier. Une variante de ce réseau est également proposée : PREP-2. Elle consiste à ne plus définir une mise à l'échelle identique dans toutes les

directions, mais adaptée en fonction de certains axes. Ceci permet de s'affranchir de systèmes de capture présentant des variations de résolution suivant les différents axes.

Le classifieur est réalisé à partir d'un réseau de neurones du type réseau à couches feedforward. L'algorithme d'apprentissage utilisé est l'algorithme de rétropropagation du gradient. L'entrée de ce réseau est une image issue de l'étage préprocesseur. La couche de sortie de ce réseau représente bien évidemment le résultat de la classification. Cette dernière couche est donc constituée d'autant de neurones qu'il y a de classes à distinguer.

Le principal avantage de ce système est qu'il permet de réaliser une reconnaissance invariante à l'aide d'un classifieur neuronal performant sans avoir à augmenter la taille de la base d'apprentissage. Les performances de ce système appliqué à la reconnaissance de caractères présentent d'importantes variations en fonction du type d'invariance obtenue : 100% pour la translation, 94% pour la mise à échelle, 89% pour l'orientation.

La principale limitation est liée à la notion de discontinuité. En effet, les auteurs précisent qu'il est nécessaire d'apporter des propriétés d'interpolation à la fonction f_{TS} . En effet, étant donné la nature de la transformation, la mise à l'échelle peut, dans le cas d'un agrandissement du modèle étudié, conduire à l'obtention d'un modèle discontinu. Dans ce cas, et étant donné la nature du classifieur, la reconnaissance est impossible sans interpoler les pixels manquants. Une autre limitation de ce système est liée à la mise en pratique de ce système. Ceci est dû au fait qu'il soit un système hybride. En effet, une partie utilise une modélisation mathématique (géométrique) et l'autre une modélisation neuronale. Dans le cadre d'une réalisation, l'obtention de la fonction invariance en orientation semble très délicate à mettre en œuvre étant donné la complexité du calcul.

III.3.3 Les architectures spécialisées

Les réseaux décrits dans ce paragraphe ont été modélisés uniquement dans le but d'effectuer la reconnaissance invariante. Quatre types de réseaux vont être décrits : le Néocognitron, le réseau de Widrow, le HONN (High Order Neural Network) et le réseau de Wang. Ce sont des architectures à couches où la couche d'entrée est une couche à deux dimensions représentant, par un codage rétinien, l'image de la forme à classifier. Ils se distinguent par la façon d'organiser et d'ajuster les connections synaptiques entre les différentes couches.

III.3.3.1 Le Neocognitron

En s'inspirant du perceptron multicouches, K. Fukushima a développé en 1975 son premier modèle de réseau appelé alors cognitron [FUK75]. Ce réseau à couches dispose d'un algorithme, développé par son auteur, d'auto-organisation. Grâce à cet algorithme, le réseau ayant alors un apprentissage non supervisé a montré des capacités de classification supérieures à celle du perceptron multicouches.

En 1982, K. Fukushima et Sei Mitake ont présenté la première version du neocognitron [FUK88]. Leurs motivations étaient de créer un réseau à apprentissage supervisé capable d'effectuer une reconnaissance de forme tout en étant tolérant aux déformations et aux déplacements. Le but du Neocognitron est de permettre une reconnaissance invariante en position, ainsi qu'une reconnaissance de modèles présentant de légères variations de taille et d'orientation. Ce réseau est basé sur le modèle du système visuel humain. Le Neocognitron est

un réseau à couches fonctionnant par extraction hiérarchique de caractéristiques, l'information y est traitée de manière hiérarchisée. Ces couches sont organisées en différents modules.

Chaque module du Néocognitron a la même structure. Il est décomposé en deux couches organisées successivement : la couche U_S puis la couche U_C . A l'intérieur de chacune de ces couches, les neurones sont regroupés telle une carte auto-organisatrice selon leurs réponses aux stimuli. Ils définissent ainsi des sous-groupes appelés plans, les plans S et les plans C où chaque plan S est associé à un plan C.

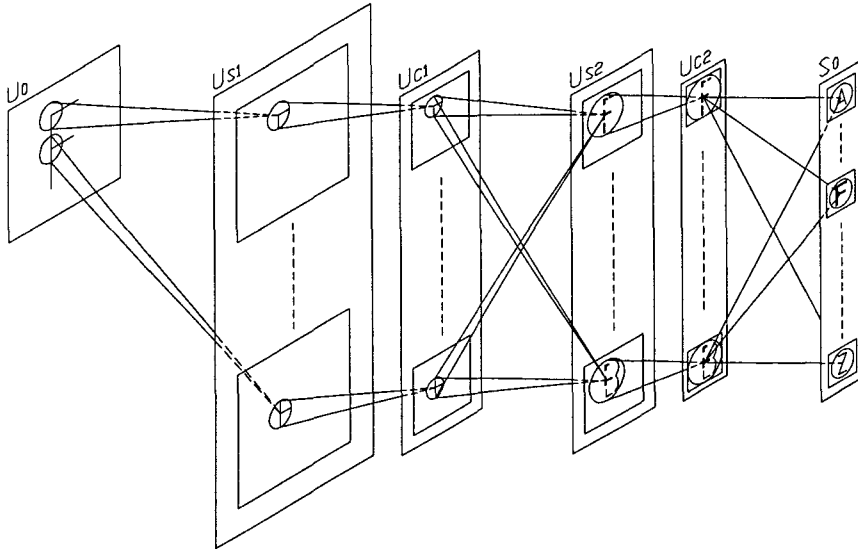


Figure III.14 : Architecture du Néocognitron

A partir d'une couche d'entrée U_0 représentant l'image du modèle à reconnaître, la première couche du premier module va mettre en évidence les caractéristiques locales présentes dans l'image. Chacun des plans S de cette première couche correspond à une caractéristique particulière à reconnaître telle qu'une ligne horizontale, une ligne verticale, une courbe ainsi tous les neurones présents dans le plan vont extraire la même caractéristique. Chacun de ces neurones appelés neurones S (Simple) est connecté suivant une région particulière de la couche U_0 . La figure III.14 montre que chaque neurone du plan a une vision limitée de la couche d'entrée. C'est une particularité du réseau Néocognitron. La zone ainsi définie est aussi appelée zone de visibilité ou fenêtre. L'ensemble des neurones constituant le plan S va permettre la couverture de toute la couche U_0 . Lorsqu'un neurone du plan S est actif, cela correspond à la présence de la caractéristique, attribuée au plan, dans la région contrôlée par le neurone S.

Les connexions entre les neurones S de ces plans et la couche d'entrée sont définies lors d'un apprentissage supervisé. Elles permettent de coder la caractéristique à extraire.

Chaque plan de la couche C représente, lui aussi, un type de caractéristique. Chaque neurone C (Complexe) de ce plan est connecté aux neurones S du plan S associé. Ces connexions, fixes quel que soit le plan concerné, décrivent une région d'intérêt sur le plan S.

Cette région correspond au champ récepteur du neurone C. L'utilité de cette couche est de concentrer l'information et de permettre une reconnaissance insensible aux modèles déformés. Ainsi, si un neurone S est actif et quelle que soit sa position dans le champ-récepteur du neurone ce dernier va être actif. On tolère ainsi une erreur sur la localisation de la caractéristiques dans la région contrôlée par le neurone S.

Tandis que les neurones des modules inférieurs permettent d'extraire les caractéristiques locales (les droites, les points, les courbes ...), ceux des modules supérieurs vont extraire les caractéristiques globales (les intersections, les origines, les angles.). Les modules sont également

organisés en couches puis en plans. Cependant, la taille de ces plans évolue. Plus le module est d'ordre élevé, plus le nombre de neurones U_C est faible et plus la taille des plans est importante.

Ce réseau effectue également la classification du modèle présenté grâce au dernier module qui n'est plus constitué que de neurones, autant de neurones qu'il y a de classes. Ainsi le neurone actif sur la couche de sortie désigne la classe d'appartenance du modèle présenté.

Le Néocognitron permet d'être insensible aux translations et au changement d'échelle des formes observées [FUK88]. En revanche, il n'est pas efficace pour des rotations. Il réalise une invariance locale, c'est-à-dire une invariance pour des transformations de faibles étendues et n'est pas adapté pour une reconnaissance PRSI. De plus, le Néocognitron a deux limitations :

- Le nombre de cellules augmente en fonction du nombre de caractéristiques à extraire et du nombre d'objets à reconnaître. Généralement, le nombre de neurones est très grand car l'invariance est obtenue à l'aide de ressources redondantes.
- Le processus d'apprentissage est très long. En effet, l'apprentissage des primitives est long et délicat car il nécessite un apprentissage spécifique par couche.

III.3.3.2 Le modèle de Widrow

Widrow s'est intéressé dès 1988 à l'utilisation des réseaux de neurones appliqués à la reconnaissance PRSI de formes [WID88]. Plus particulièrement, il définit une architecture neuronale à couches dans laquelle il a introduit des propriétés de translation gauche-droite et haut-bas mais aussi des propriétés d'invariance rotation et taille. Son réseau est une architecture simple composée de sous-réseaux Adaline. La couche d'entrée de son réseau correspond à la représentation rétinienne de l'objet. Pour définir les couches suivantes, Widrow a introduit la notion de couche "Slab". Sur cette couche de même dimension que la couche d'entrée sont disposés des neurones Adaline (AD). Ces neurones sont entièrement connectés à la couche d'entrée et leur sortie est reliée à un unique neurone MAJ (figure III.15). Sa sortie est fonction du vote entre toutes ses entrées. L'entrée majoritaire est représentée en sortie.

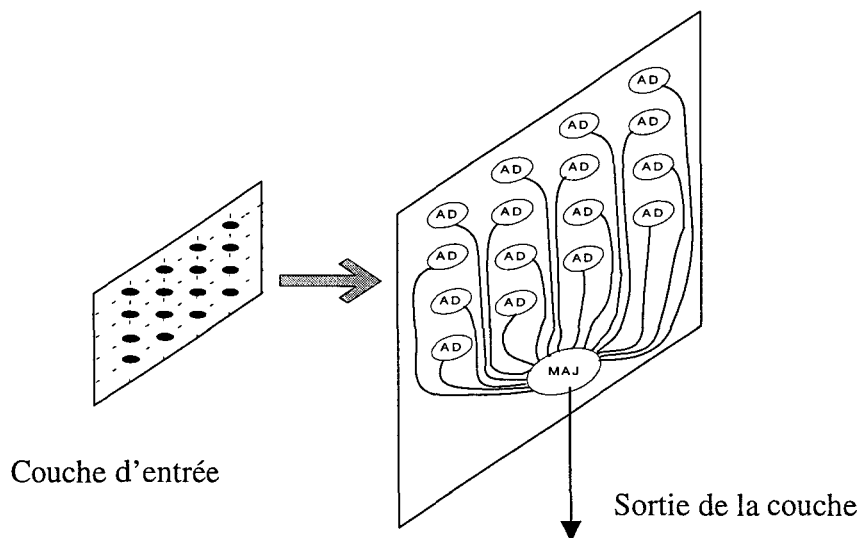


Figure III.15 : couche "Slab0"

L'invariance en translation est réalisée en utilisant plusieurs couches "Slab" et en ajustant d'une manière particulière les poids de connexions entre leurs neurones Adaline et ceux de la couche d'entrée.

Les poids de connexion des neurones Adaline vont permettre d'introduire les propriétés d'invariance. Le tableau suivant représente les poids de connexions pour les neurones de la première couche « Slab0 » représentés par la flèche grise.

Position de l'Adaline	0	1	2	3
0	W_{ij}	$T_{D1}W_{ij}$	$T_{D2}W_{ij}$	$T_{D3}W_{ij}$
1	$T_{B1}W_{ij}$	$T_{D1}T_{B1}W_{ij}$	$T_{D2}T_{B1}W_{ij}$	$T_{D3}T_{B1}W_{ij}$
2	$T_{B2}W_{ij}$	$T_{D1}T_{B2}W_{ij}$	$T_{D2}T_{B2}W_{ij}$	$T_{D3}T_{B2}W_{ij}$
3	$T_{B3}W_{ij}$	$T_{D1}T_{B3}W_{ij}$	$T_{D2}T_{B3}W_{ij}$	$T_{D3}T_{B3}W_{ij}$

Tableau III.1 : Poids synaptiques de la couche "Slab0"

A partir d'une initialisation aléatoire des connexions W_{ij} entre l'Adaline (0,0) et tous les neurones (i,j) de la couche d'entrée, les autres neurones utilisent cette référence pour leurs propres poids. En effet, ceux-ci sont obtenus en effectuant des opérations de translation T sur ces poids. Ainsi T_{Bx} représente une translation vers le bas de x pixels, T_{Dx} une translation vers la droite de x pixels.

Etant donné la nature du neurone MAJ, il est évident que sa sortie sera insensible aux variations dues aux translations haut-bas, gauche-droite.

Le nombre de couches "Slab" choisies est généralement identique au nombre de neurones de la couche d'entrée. Les connexions des autres couches "Slab" à la couche d'entrée sont définies de la même manière que pour la couche "Slab0". Le seul changement intervient dans l'initialisation de nouvelles connexions W_{ij} ². On peut alors modéliser la sortie de ce réseau comme un espace de représentation de dimension identique à l'espace d'entrée. La figure III.16 représente le réseau dans sa totalité.

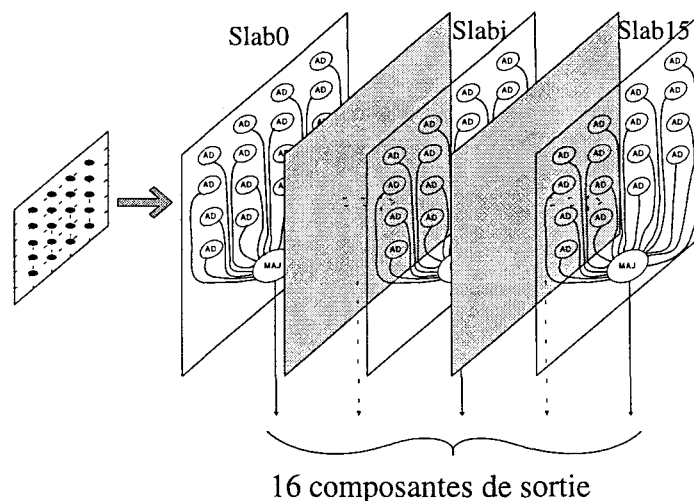


Figure III.16 : Le réseau de Widrow

Ce réseau peut être amélioré en lui incorporant, en plus des propriétés d'invariance en translation, des propriétés d'invariance en rotation et en taille. Le principe proposé est identique à celui de la translation. Ainsi, Widrow a proposé la résolution des problèmes liés à des rotations d'objets selon quatre angles 0, 90, 180, 270.

Pour ce faire, il propose d'ajouter pour chaque couche "Slab" déjà définie pour la translation, trois nouvelles couches. Ces couches transforment par une rotation les poids de connexion déjà calculés. La figure III.17 montre que l'unique couche "Slab0" est remplacée par un module de quatre couches « Slab » : Slab0c0, Slab0c90, Slab0c180, Slab0c270.

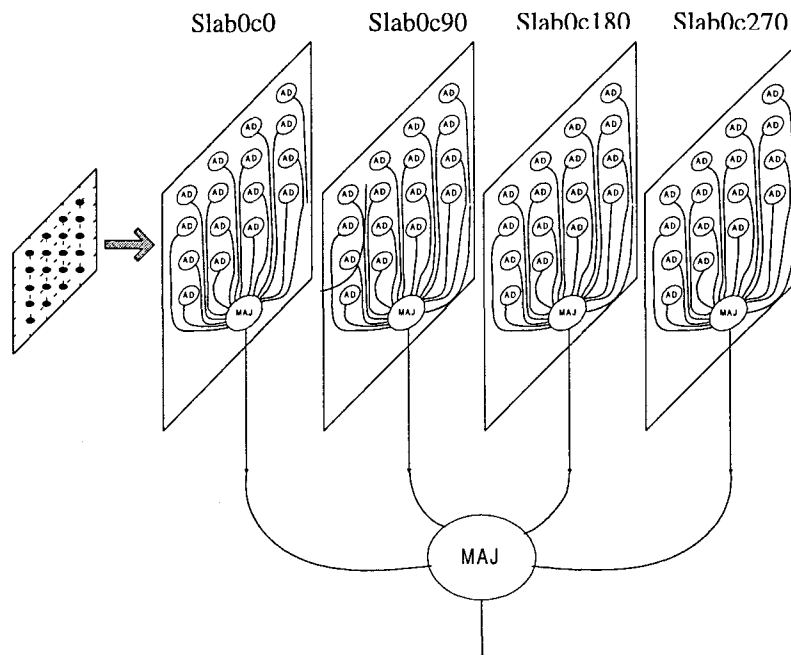


Figure III.17 : invariance en translation et en rotation

Ainsi pour la couche Slab0c90, les poids de connexion des neurones Adaline (tableau III.2) sont fonction des transformations de translation mais aussi de la rotation de 90 degrés.

Position de l'Adaline	0	1	2	3
0	$R_{A90}W_{ij}$	$R_{A90}T_{D1}W_{ij}$	$R_{A90}T_{D2}W_{ij}$	$R_{A90}.....$
1	$R_{A90}T_{B1}W_{ij}$	$R_{A90}T_{D1}T_{B1}W_{ij}$	$R_{A90}T_{D2}T_{B1}W_{ij}$	$R_{A90}.....$
2	$R_{A90}T_{B2}W_{ij}$	$R_{A90}T_{D1}T_{B2}W_{ij}$	$R_{A90}T_{D2}T_{B2}W_{ij}$	$R_{A90}.....$
3	$R_{A90}T_{B3}W_{ij}$	$R_{A90}T_{D1}T_{B3}W_{ij}$	$R_{A90}T_{D2}T_{B3}W_{ij}$	$R_{A90}.....$

Tableau III.2 : Poids synaptiques de la couche "Slab0c90"

Les couches Slab0c180 et Slab0c270 sont décrites de la même manière, seul l'angle de la rotation est changé.

Sans aborder dans le détail l'invariance en taille, on peut dire que cette invariance est obtenue selon le même principe que l'invariance en rotation. Il suffit de définir de nouvelles couches basées sur celles existantes. Pour chacune de ces nouvelles couches, on ajoute un terme de transformation qui est une homothétie. Le centre de celle-ci est le centre de l'espace rétinien et le rapport serait fonction de la couche concernée. Ainsi, pour définir une invariance en taille pour des rapports 1/2, 1, 2, 3, il est nécessaire d'ajouter 3 nouvelles couches par couche déjà existante.

Bien qu'il puisse réaliser des opérations d'invariance simplement, ce réseau est limité. En effet, la conception d'un réseau disposant d'une invariance en rotation ainsi que d'une invariance en taille suffisamment précises nécessite une architecture extrêmement importante. De plus, la structure proposée par Widrow ne peut effectuer une reconnaissance invariante en rotation que pour des modèles présentés en entrée ayant subi des rotations d'angle multiple de 90°.

Ainsi, pour une imagette 5*5, garantir une invariance à la fois en translation, en taille (4 rapports) et en orientation (4 angles) oblige à construire un réseau constitué de :

- 1 couche d'entrée de 25 neurones
- 25 couches de 25 neurones pour la translation
- 3* 25 couches de 25 neurones pour l'invariance en orientation
- 3 * (4*25) couches de 25 neurones pour l'invariance en taille

soit 7500 neurones Adaline +25 neurones MAJ pour toutes les invariances.

La taille de cette architecture semble donc totalement inadaptée à une réalisation matérielle.

De plus, pour construire un système de reconnaissance invariante à partir de ce modèle, il est nécessaire d'y ajouter un réseau, le réseau « descrambler ». Celui-ci va, à partir des résultats des différentes couches "Slab", permettre d'obtenir en sortie le modèle présenté à l'invariance, dans une position, une taille et une orientation normalisée. Enfin, pour effectuer la classification, un second réseau reste nécessaire.

III.3.3.3 Le HONN : Higher-Order Neural Network

Le réseau HONN [SPI92] permet de réaliser une reconnaissance invariante en position, en taille et en orientation. L'architecture et les connexions de ce réseau lui permettent d'effectuer cette invariance. Sa particularité est d'effectuer cette invariance sans apprentissage.

Le réseau HONN est un réseau à couches feedforward. Sa structure est celle d'un réseau dit sigma-pi. Le réseau sigma-pi se distingue du réseau sigma (réseau classique à couches) par la nature des connexions entre couche d'entrée et deuxième couche. Contrairement au réseau sigma où toutes les connexions intercouches sont variables, les connexions pour le réseau sigma-pi entre couche d'entrée et deuxième couche sont fixes.

La sortie d'un neurone i d'un réseau d'ordre 4 s'exprime par :

$$y_i = \xi \left(\sum_j w_{ij} x_j + \sum_k \sum_j w_{ijk} x_j x_k + \sum_l \sum_k \sum_j w_{ijkl} x_j x_k x_l + \sum_m \sum_l \sum_k \sum_j w_{ijklm} x_j x_k x_l x_m \right) \quad (\text{III.17})$$

avec ξ : fonction seuillage; x_j : activité x du neurone j de la couche précédente;
 w_{ji} : poids d'interconnexion entre y_i et x_j .

Le réseau utilisé pour la reconnaissance invariante PRSI est un réseau strictement du troisième ordre [SPI92]. L'architecture simplifiée de ce type de réseau, disposant de 4 neurones d'entrée I et d'un neurone de sortie O, est décrite figure III.18. Chaque neurone d'entrée est associé à deux autres neurones de la couche d'entrée afin de définir un triplet. La sortie du réseau HONN est fonction de ces triplets ainsi que de leur pondération synaptique w_{ijkl} .

L'activité du neurone de sortie s'exprime par :

$$O_i = \xi \left(\sum_j \sum_k \sum_l w_{ijkl} I_j I_k I_l \right) \quad (\text{III.18})$$

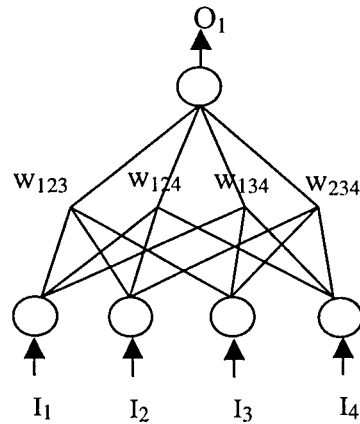


Figure III.18 : Exemple de réseau HONN du troisième ordre

Pour la reconnaissance de formes, ce réseau est appliqué à des images de contour. L'invariance en position, taille et orientation est obtenue à l'aide des triplets. Chacun représente un triangle dans l'image (figure III.19). Les poids de connexion permettent de mémoriser les angles (α, β, γ) formés par ce triangle. Ces angles sont organisés dans le sens des aiguilles d'une montre en partant toujours de l'angle le plus faible. Ainsi, la sortie du neurone HONN sera identique pour des triplets (triangles) géométriquement similaires. Le principe ainsi utilisé dans cette architecture est la triple corrélation.



Figure III.19 : Notion de triplet

La principale limitation du réseau HONN est liée au nombre de combinaisons de triplets possibles. Plus l'image est importante, plus le nombre de triplets est élevé. Ainsi, une image de 9×9 comporte 85 320 combinaisons de triplets. Une image 128×128 comporte $7.3 \cdot 10^{11}$ combinaisons! De plus, le nombre de connexions rend l'implémentation parallèle tout à fait improbable. Des techniques ont été proposées afin d'éviter cette limitation. La plus simple consiste à définir des zones dans lesquelles les triplets peuvent se situer. Ainsi, la technique dite de « coarse coding » permet de limiter de manière importante le nombre de connexions en définissant des régions de recherche des triplets [SPI93].

Une autre possibilité de développement des HONN est d'utiliser une approche multirésolution en complément. Différents travaux ont combiné au réseau HONN cette approche : la transformée ondelette, l'approche pyramidale "gaussian pyramid" et la représentation quadtree [KAN92][LIU97]. Cette solution permet d'améliorer la phase d'apprentissage de manière à la rendre plus rapide mais également de diminuer la taille de l'architecture du système.

III.3.3.4 Le modèle de Wang

Ce modèle [WAN96], créé en 1996, est décomposé en deux parties : extraction de caractéristiques invariantes et classification. Il permet d'effectuer une reconnaissance quelle que soit la position la taille et l'orientation des formes présentées. Les principes neuronaux développés par les auteurs pour réaliser les invariances sont inspirés des moments géométriques. Nous ne décrivons ici que la partie permettant l'obtention des invariances.

L'invariance en position et en orientation est réalisée par la première partie. Celle-ci est constituée de trois couches de neurones à deux dimensions. la couche PNL (Position Normalisation Layer) effectue une normalisation en position du caractère. La couche RNL (Rotation Normalisation Layer) effectue une normalisation en orientation et la dernière couche FEL (Feature Extraction Layer) va extraire les caractéristiques utilisées pour la classification.

L'invariance en taille est réalisée grâce à la seconde partie le classifieur : un réseau à couches.

Chacune des couches PNL et RNL est constituée de neurones interconnectés. Deux types de neurones y sont rencontrés : le neurone "information" noté N et le neurone "contrôle" noté C. Ils sont tous les deux à sortie binaire. Leur fonction d'activation f_a est une fonction binaire à seuil avec $f_a(x)=1$ si $x \geq 0$ et $f_a(x)=0$ si $x < 0$.

La structure de la première couche de normalisation en position PNL est rectangulaire. Les neurones y sont organisés suivant des coordonnées cartésiennes avec comme centre de système de coordonnées le centre de la couche. Cette couche est constituée de $X_p * Y_p$ neurones noté $N_{PNL}(i,j)$.

Cette couche PNL est aussi la couche d'entrée du système. L'image binaire du modèle y est appliquée. L'activité de chaque neurone $N_{PNL}(i,j)$ de cette couche correspond au niveau de gris du pixel (i,j) de l'image. Cette couche est une couche à compétition. Chaque neurone y est interconnecté et après stabilisation de la couche, l'activité du neurone $N_{PNL}(i,j)$ s'exprime par :

$$N_{PNL}^{t+1}(i,j) = f_a \left(\begin{aligned} &C_x^+ N_{PNL}^t(i+1,j) + C_x^- N_{PNL}^t(i-1,j) + C_y^+ N_{PNL}^t(i,j+1) \\ &+ C_y^- N_{PNL}^t(i,j-1) + (1 - C_x^+ - C_x^- - C_y^+ - C_y^-) N_{PNL}^t(i,j) \end{aligned} \right) \quad (III.19)$$

avec f_a : fonction d'activation du neurone.

Sont connectés à chaque neurone dit d'information $N_{pnl}(i,j)$ quatre neurones on-off C_x^+ , C_x^- , C_y^+ , C_y^- , appelés neurones de contrôle. L'invariance en position est obtenue en combinant localement les activités des neurones N_{PNL} et celles de ces neurones. L'activité des neurones C va permettre de caractériser la position de la forme dans l'image. Elle est calculée à partir des moments géométriques d'ordre 1. L'activité de ces neurones s'exprime par :

$$\begin{aligned} C_x^+ &= f_a(m_a(x_a - 0,5)) & C_y^+ &= f_a(m_a(y_a - 0,5) - W(C_x^+ + C_x^-)) \\ C_x^- &= f_a(-m_a(x_a + 0,5)) & C_y^- &= f_a(-m_a(y_a + 0,5) - W(C_x^+ + C_x^-)) \end{aligned} \quad (III.20)$$

avec

$$\begin{aligned} x_a &= \frac{1}{m_a} \sum_i \sum_j i N_{PNL}(i,j) & y_a &= \frac{1}{m_a} \sum_i \sum_j j N_{PNL}(i,j) \\ \text{et } m_a &= \sum_i \sum_j N_{PNL}(i,j) & & \text{avec } i, j \text{ décrivant la couche PNL} \end{aligned}$$

et avec W poids de connexion entre les neurones de contrôle suffisamment grand pour que les neurones de contrôle C_x soient nuls quand les neurones C_y sont actifs et vice versa.

La normalisation en position est obtenue itérativement. Le modèle présenté en entrée va subir successivement des décalages jusqu'à ce que x_a et y_a soient nuls.

En effet, supposant un modèle dont le centre de gravité est $\{x,0\}$. Soit $x>0$, seul le neurone C_x^+ va être actif (valant 1), ce qui impose que :

$$N_{PNL}^{t+1}(i, j) = f_a(N_{PNL}^t(i+1, j)) \quad (III.21)$$

donc le modèle va subir un décalage vers la gauche. Une fois la stabilité obtenue, la forme est représentée au centre de la couche PNL.

La structure de la couche RNL effectuant l'invariance en rotation est un peu plus particulière. Constituée de $Xr \times Yr$ neurones, elle est organisée suivant Xr cercles concentriques constitués de Yr partitions (figure III.20). Chaque neurone de cette couche est caractérisé par un système de coordonnées polaire où r représente le rayon et a l'angle.

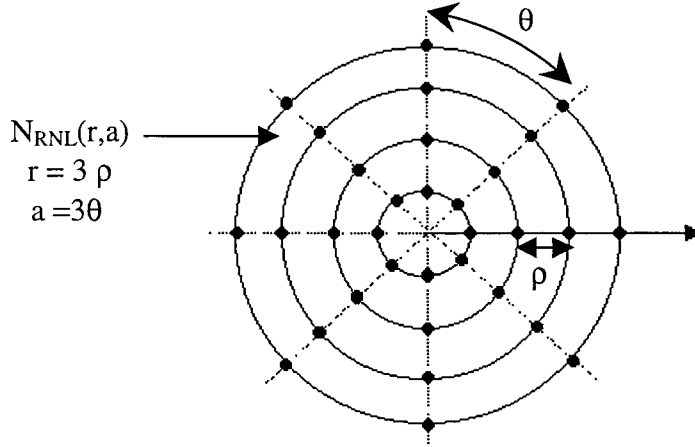


Figure III. 20 : Représentation de la couche RNL

Chaque neurone de cette couche reçoit les informations issues de la couche PNL via un changement de repère qui va transformer le repère des coordonnées cartésiennes en coordonnées polaires. Chaque neurone $N_{RNL}(r,a)$ de cette couche est interconnecté avec ses voisins et après avoir obtenu la stabilité de la couche, son activité s'exprime par :

$$N_{RNL}^{t+1}(r, a) = f_a(C_{cl}N_{RNL}^t(r, a-1) + C_{cc}N_{RNL}^t(r, a+1) + (1-C_{cl} + C_{cc})N_{RNL}^t(r, a)) \quad (III.22)$$

Le calcul des poids C_{cl} et C_{cc} correspond à trouver l'orientation de la forme. Celle-ci est définie par une différence entre deux points caractéristiques (A,B) de la forme. Le premier est obtenu en effectuant un calcul d'un moment géométrique du second ordre sur l'ensemble de la couche RNL. Le point B caractérise la forme selon l'expression d'un moment géométrique du troisième ordre. Ainsi, on définit un vecteur O qui caractérise l'orientation de la forme.

Pour le point A:

$$x_a = \frac{1}{R_1} \sum_r \sum_a (r)^2 \cos(a) N_{RNL}(r, a)$$

$$y_a = \frac{1}{R_1} \sum_r \sum_a (r)^2 \sin(a) N_{RNL}(r, a)$$

Pour le point B:

$$x_b = \frac{1}{R_2} \sum_r \sum_a (r)^3 \cos(a) N_{RNL}(r, a)$$

$$y_b = \frac{1}{R_2} \sum_r \sum_a (r)^3 \sin(a) N_{RNL}(r, a)$$

Le vecteur O :

$$x_o = \sum_r \sum_a \left(\frac{(r)^3}{R_2} - \frac{(r)^2}{R_1} \right) \cos(a) N_{RNL}(r, a)$$

$$y_o = \sum_r \sum_a \left(\frac{(r)^3}{R_2} - \frac{(r)^2}{R_1} \right) \sin(a) N_{RNL}(r, a) \quad (III.23)$$

Ce vecteur $O \{x_0, y_0\}$ va permettre le calcul des deux neurones de contrôle. Avec le même principe que celui utilisé sur la couche PNL, les poids C_{cl} et C_{cc} permettent d'effectuer itérativement la normalisation en orientation.

Le modèle décrit par Wang, est un système de reconnaissance à part entière. Ainsi s'ajoute aux couches effectuant les invariances en position et en orientation une couche permettant d'extraire un vecteur de caractéristiques. Cette couche FEL-A est constituée de deux sous-couches. La première est équivalente à la couche RNL et représente les neurones suivant des coordonnées polaires $N_{FEL-A}(r, a)$. La seconde couche FEL-B est une couche à 1 dimension. L'activité de chaque neurone $N_{FEL-B}(a_1)$ est obtenue en sommant l'activité des neurones de coordonnées (r, a_1) avec r décrivant toute la couche FEL-A. Cette représentation permet de mettre en évidence la taille du modèle et ainsi d'aborder facilement la normalisation en taille effectuée par le classifieur.

Le modèle de classifieur proposé est un réseau feedforward à couches. Sans entrer dans le détail de sa modélisation, l'auteur réalise la normalisation en taille grâce à la définition d'un nouveau type de neurone MSN (Modified Sigmoid Neuron). Ce neurone grâce à deux types de poids l'un excitateur les $a(i)$ l'autre inhibiteur les $b(i)$ qui sont déterminés lors d'un apprentissage supervisé, possède la même réponse O pour deux formes présentant des échelles différentes. Elle s'exprime par :

$$O = f_a \left((\xi + 1) \frac{a'(i) \cdot E_{FEL-B}}{b(i) \|E_{FEL-B}\|} - \xi \right) \quad (III.24)$$

avec ξ : sélectivité

E_{FEL-B} : ensemble $\{ N_{FEL-B}(0); \dots ; N_{FEL-B}(Nb) \}$ avec Nb nombre de neurones

La principale limitation de cette architecture est le temps de traitement. En effet, la normalisation s'effectue itérativement. Il convient donc d'effectuer un grand nombre de boucles afin de normaliser l'image originelle. De plus, il est impossible de garantir le temps de traitement. En effet, le nombre de cycles dépend directement de la position, la taille et de l'orientation de l'objet.

Conclusion de la troisième partie

La reconnaissance de caractères a été l'un des thèmes de recherche qui a permis la mise au point d'un très grand nombre de techniques utilisées en reconnaissance de formes. Les principales évolutions ont été liées à la création de nouvelles techniques d'extraction de caractéristiques. En effet, les performances des systèmes de reconnaissance de caractères d'imprimerie sont essentiellement liées aux développements de ces techniques. L'utilisation de primitives a permis d'utiliser des algorithmes de classification très simples, très rapides et peu gourmands en ressources. En dehors des applications classiques, il est apparu nécessaire d'introduire des notions d'invariance en OCR. Ceci a été rendu possible par l'utilisation de techniques de transformations invariantes telles que la translation, la rotation ou l'agrandissement.

La reconnaissance invariante a pu être réalisée suivant deux approches :

- L'extraction et l'utilisation de caractéristiques invariantes
- La normalisation, qui possède l'avantage d'obtenir un modèle invariant de la forme à étudier.

C'est cette dernière approche qui a retenu notre attention dans le cas de l'association avec l'étape classifieur sélectionnée dans la 2^{ème} partie car elle a l'avantage d'effectuer directement une normalisation d'image.

Concernant notre volonté d'obtenir une reconnaissance de caractères en utilisant des techniques neuronales, l'étude de cette troisième partie nous a montré que de l'association entre des classifieurs neuronaux et étapes d'extraction de caractéristiques invariantes, seules les architectures spécifiques peuvent apporter le meilleur compromis en qualité des résultats et complexité de mise en oeuvre. Cependant, les architectures neuronales qui permettent d'effectuer une invariance en position en taille et en orientation par normalisation ne sont actuellement pas beaucoup développées. Seuls quelques réseaux ont été développés dans cette optique [BIS95]. Cependant, ces architectures restent très limitées et leur réalisation très difficile. En effet, les normalisations sont généralement effectuées itérativement. Pour des images de grande taille et où les formes peuvent subir des variations importantes, cette technique n'est pas applicable. De plus, il est impossible de garantir un temps de traitement pour chaque modèle. Celui-ci va dépendre du nombre d'itérations à effectuer pour obtenir le modèle invariant.

Etant donné notre souhait de développer un système de reconnaissance invariante de caractères, nous avons été amenés à développer une nouvelle architecture neuronale. Ce réseau a été réalisé en adaptant certains concepts présentés dans cette partie notamment la normalisation par blocs successifs (Yüccer et Wang). Ce souhait de développer une nouvelle architecture est également dû au fait que pour des conditions d'acquisition très variées (luminosité, marquage, position, taille et orientation), aucune solution complète n'existe, il est nécessaire d'associer une étape de prétraitement et l'étape de normalisation. Cette architecture originale va être décrite très précisément dans la 4^{ème} partie.

Les quatrième et cinquième parties sont consacrées à l'étude de notre système de reconnaissance. En effet, après avoir déterminé dans la deuxième partie, une architecture neuronale adaptée pour la classification de caractères et après avoir défini les concepts d'une nouvelle architecture utilisable pour la reconnaissance invariante et la normalisation, nous pouvons aborder la présentation détaillée de notre système.

Quatrième Partie :

Etude d'un
système de reconnaissance
de caractères industriels

Introduction

Après avoir présenté, dans la première partie, certaines limites auxquelles se heurte actuellement la reconnaissance de caractères imprimés, nous consacrerons la quatrième partie à la proposition et à l'étude d'un système de reconnaissance de caractères permettant d'améliorer les performances pour certains de ces cas limites. Ces limites sont essentiellement dues aux variations de luminosité, aux variétés de support et de teinte, aux variations et aux dégradations de marquage et aux problèmes d'invariance lors de la prise de vue.

Le thème de la reconnaissance de caractères en milieu industriel permet de réunir l'ensemble des principaux problèmes énoncés précédemment. Nous avons cherché à construire un système de reconnaissance basé sur des notions d'invariance aux conditions lumineuses, à la nature des supports et des marquages utilisés ainsi qu'à la position, à la taille et à l'orientation des caractères lors de la prise de vue. De plus, l'utilisation de caractères en milieu industriel nous contraint, en complément des polices couramment utilisées, à tenir compte de polices spécifiques. Notre système a été élaboré de la manière la plus générale possible afin de permettre une application en reconnaissance de forme quelconque. Sa modélisation est également basée sur l'évolution rapide des techniques de l'informatique industrielle et des chaînes de traitement parallèle. En effet, dans le cadre de l'image, l'utilisation des algorithmes parallèles est beaucoup plus cohérente en terme d'organisation du traitement et des temps de calcul.

Dans le premier paragraphe de cette quatrième partie, nous présenterons les principales difficultés liées à une reconnaissance de caractères en milieu industriel.

IV.1 Présentation du problème posé

IV.1.1 La reconnaissance de caractères en milieu industriel

Le terme "caractères industriels" désigne les caractères principalement utilisés pour le marquage des articles lors de leur fabrication. Ils sont utilisés pour référencer, par exemple, le type de production, la date de traitement au niveau de différents lieux de production, les matières premières utilisées,... Ils sont très importants pour suivre la production et assurer la traçabilité des produits. Ils permettent également de référencer et de stocker les produits finis, d'identifier automatiquement les produits industriels, de contrôler leur qualité ainsi que leur conformité, et enfin ils permettent de trier les productions (figure IV.1).

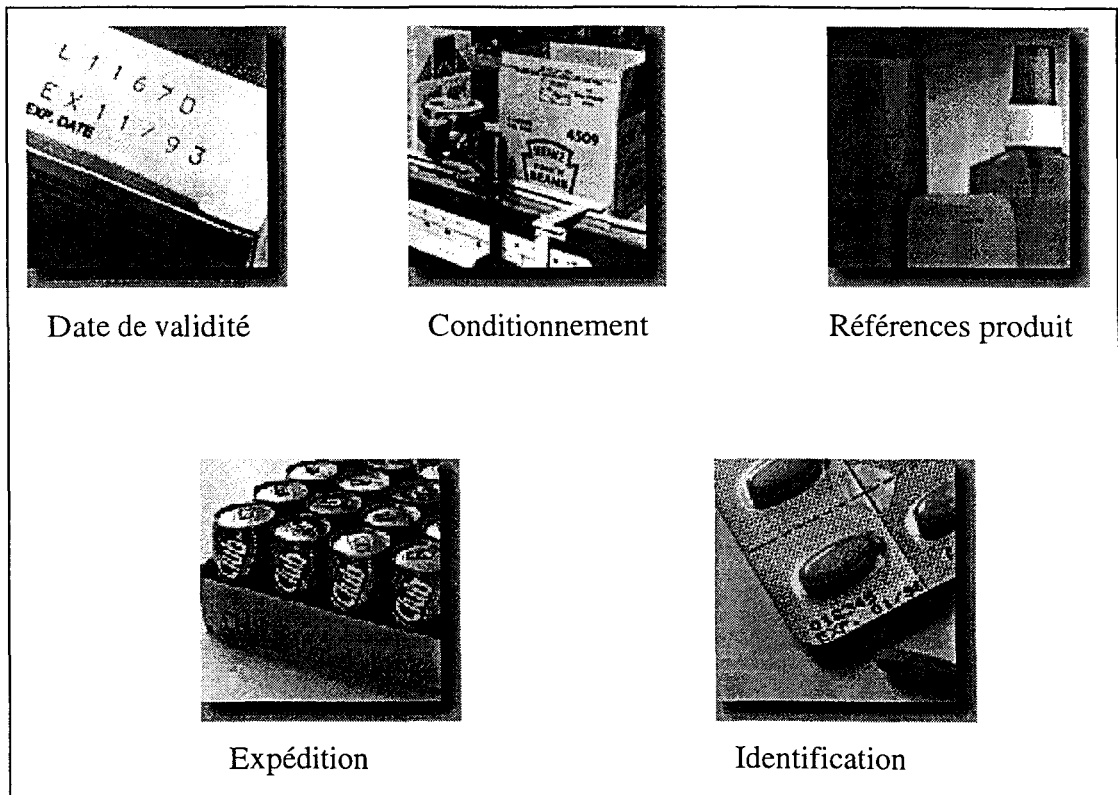


Figure IV.1 : Exemples d'application marquage

Lors d'un process de fabrication, il est nécessaire d'identifier, à de nombreuses reprises, l'objet en cours de production. Il est donc intéressant de disposer de systèmes pouvant reconnaître les caractères automatiquement à différents stades de la production. Bien que de nombreux logiciels de reconnaissance de caractères existent pour les applications bureautiques (plus d'une quarantaine), peu de systèmes de reconnaissance de caractères ont été créés pour les applications industrielles. Les principaux systèmes utilisés dans les industries manufacturières peuvent se regrouper selon deux approches :

Les systèmes configurables : le CVIM-OCR Pak de Rockwell
le acuReader/OCR II de Cognex

Les systèmes programmables : Logiciel Xcaliper d'Optimas Corporation.
Logiciel Prophecy d'Imaging Technology.
Logiciel NeuroCheck de DATA Translation.
Librairie Mil de Matrox.

Ceux-ci sont principalement basés sur les techniques de "template matching" ou sur une classification à base de réseaux de neurones.

La première société française s'étant illustrée dans la recherche et le développement pour la reconnaissance de caractères en milieu industriel est l'entreprise AI Vision systèmes, on peut lui attribuer l'implantation de plus d'une centaine de systèmes.

Généralement le bon fonctionnement de ces systèmes est lié à leur performance mais aussi au type de système de marquage utilisé. Faire le choix d'une solution de reconnaissance de caractères en milieu industriel revient, actuellement, à associer correctement un système de

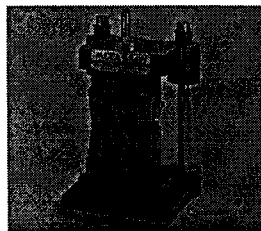
reconnaissance et une solution de marquage. Dans le cadre du développement d'un système de reconnaissance de caractères appliqué au milieu industriel, il nous a été nécessaire de découvrir les caractéristiques des différents marquages pour essayer de définir une solution de reconnaissance indépendante de la technique de marquage utilisée.

IV.1.2 Introduction au marquage industriel

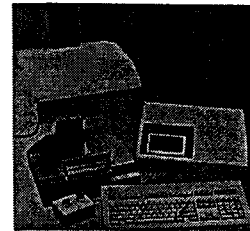
Pour permettre l'identification automatique des produits, il est nécessaire d'y apposer des étiquettes ou de les marquer directement. Le marquage est donc la première étape nécessaire à une gestion automatisée de la production. Les techniques de marquage sont très diverses. Elles sont adaptées en fonction des besoins de l'identification et du type d'articles à traiter. Le marquage industriel a recours à des techniques informatiques (imprimante matricielle, jet d'encre, laser), à des techniques de gravage (micropercussion, estampage, gravure laser, gravure chimique) ou à des techniques d'imprimerie (sérigraphie, offset, tampographie) (figure IV.2).



Offset



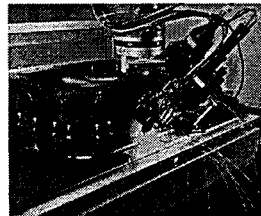
Estampage



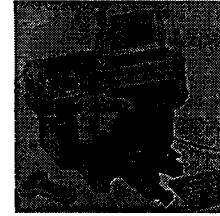
MicroPercussion



Jet d'encre



Laser



Machine à graver

Figure IV.2 : Exemples de techniques de marquage

Contrairement au nombre et à la variété de marquages utilisés, les polices de caractères utilisées sont bien souvent sommaires, les caractères industriels étant généralement d'une typographie très simple (polices standards type courier ou serif). Certaines polices ont été spécialement développées pour le marquage industriel afin d'améliorer la lisibilité dans les conditions difficiles (support, marquage, environnement,...). Ces fontes dites industrielles définissent des caractères très distincts permettant ainsi de limiter certaines sources d'ambiguïté entre deux types de caractères (entre le C et le G, le O et le D...).

IV.1.3 Notions de fontes industrielles

Trois principales normes ont été spécialement définies pour les techniques de marquage industriel. Nous les présentons ici brièvement.

IV.1.3.1 La norme OCR-A

Cette norme a été établie en 1960 par l'ANSI (American National Standards Institute) sous le standard ANSI X3.17-81. Elle est également reconnue sous le standard ISO [ISO1073-1]. Les caractères qui la définissent ont été choisis de manière à supprimer toute ambiguïté entre eux évitant ainsi des confusions possibles entre le "D", le "O", le "Q" et le chiffre "0".

```

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890 ! " # % & / ( ) = ? \ { } [ ]
+ * # ' < > , ; . : - ¡ ¢ ¥ £
    
```

Figure IV.3 : Norme OCR -A

IV.1.3.2 La norme OCR-B

La norme OCR-B que l'on peut également connaître sous le nom de [ISO1073-2] ou ECMA-11 (European Computer Manufacturers Association), est reconnue par l'ANSI sous le standard ANSI X3.49-75.

```

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
1234567890 ! " $ % & / ( ) = ? \ { } [ ]
< > , ; - : - _ Ä Ö Ü ä ö ü ß + * # '
    
```

Figure IV.4 : Norme OCR-B

IV.1.3.3 Les normes pour le codage MICR

Le codage MICR (Magnetic-Ink Character Reading) [ISO2033] utilise des nombres qui peuvent être lus sans ambiguïté par une tête magnétique. Cette norme est essentiellement utilisée pour les applications de reconnaissance de caractères dans l'industrie bancaire, plus particulièrement pour le traitement automatique des chèques. Les polices de caractères utilisées pour le codage MICR sont E-13B, CMC-7 (figures IV.5 et IV.6), OCR-A et OCR-B.

1 2 3 4 5 6 7 8 9 0

Figure IV. 5 : E-13B



Figure IV.6 : CMC-7

La police E-13B, développée en 1958, est essentiellement utilisée aux USA, Canada, Australie. La norme CMC-7 est une norme française, développée en 1964 par BULL. Elle est utilisée dans la quasi-totalité des pays européens exceptés les pays britanniques [SRI97].

L'encre utilisée pour ces caractères est une encre magnétique. Ainsi, un système de lecture magnétique peut identifier chaque type de caractère par codage.

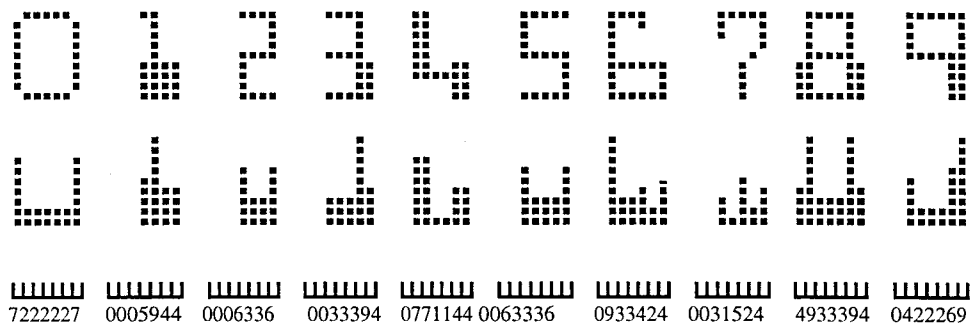


Figure IV.7 : Codage MICR

L'exemple de la figure IV.7 représente un codage MICR obtenu à partir de la police E-13-B. A partir d'une image 9x7 pixels, on représente chaque nombre par un code de 7 chiffres. Chacun de ces chiffres représente la somme de pixels noirs suivant chacune des 7 colonnes.

Actuellement, en dehors de certains moyens de marquage très spécifiques (gravage, microburinage, estampage), la variété des polices utilisées n'est plus limitée. Ainsi, aux polices spécifiques décrites précédemment, peuvent s'ajouter un très grand nombre de polices couramment employées en bureautique (figure IV.8).



Figure IV.8 : Quelques fontes industrielles

IV.1.4 Présentation du problème

Le premier chapitre nous a montré que pour les applications bureautiques la reconnaissance de caractères est aujourd'hui aisée à réaliser. Pour les applications industrielles, cela reste cependant difficile. Les raisons sont nombreuses.

La première raison est évidemment liée à la nature de l'acquisition. L'identification industrielle est, à l'exception des lecteurs à barres, essentiellement réalisée à l'aide de caméras CCD. Ce type d'acquisition est par opposition au scanner utilisé en bureautique, d'une résolution faible et très sensible aux variations d'éclairage extérieur. Les images ainsi obtenues sont de qualité moindre. Ce type d'acquisition pose des problèmes de localisation des objets et donc des caractères lors de la prise de vue.

Quatrième Partie : Etude d'un système de reconnaissance de caractères industriels

Une seconde raison est liée à la qualité du contraste entre les caractères et le fond. Contrairement à un support papier et à un marquage par encre pour les applications bureautiques, les associations support/marquage rencontrées en milieu industriel n'offrent pas toujours de bon contrastes (figure IV.9). Cela signifie que le système doit se révéler capable de s'adapter aux caractéristiques lumineuses qu'il soit placé sur une chaîne de fabrication de tôles, dans une papeterie ou dans une tannerie...

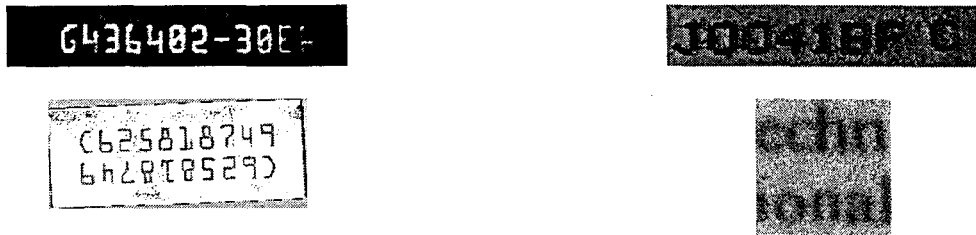


Figure IV.9 : Exemples de variations lumineuses

En dehors de problèmes de luminosité, les supports peuvent également s'avérer être des sources de problèmes. En effet, dans de nombreuses applications, la lisibilité des caractères est perturbée par un fond non uniforme. La figure IV.10 représente quelques exemples de supports où la reconnaissance de caractères est difficile à réaliser.



Figure IV.10 : Exemples de supports difficiles

Concernant la phase de reconnaissance, les polices rencontrées sont variées ce qui impose comme pour les applications bureautiques d'effectuer une reconnaissance multifonte. Cependant, une difficulté est spécifique aux techniques de marquage en milieu industriel. Celles-ci créent des modèles particuliers (caractère représenté sous forme de pointillés en production de canettes de fer, sous forme de trous dans l'industrie du cuir, etc.). La figure IV.11 représente des exemples de caractères utilisés pour le marquage industriel.



Figure IV.11 : Quelques polices spécifiques

Ainsi pour résumer, les problèmes rencontrés dans le cadre de la reconnaissance de caractères en milieu industriel peuvent être regroupés suivant quatre catégories.

a) Les problèmes liés à l'acquisition

- Variations lumineuses lors de l'acquisition.
- Non-uniformité de l'éclairage.
- Position de la caméra par rapport à l'objet (distance, angle de vue)
- Variation de position de l'objet dans la scène à observer

b) Les problèmes liés à la nature du support

- Mauvaise qualité du support (réflexion, bruit).
- Informations parasites (trame du papier, coup sur le support, informations vues en transparence).
- Support pré-imprimé (chèques)
- Opacité faible (Verre)

c) Les problèmes liés à la nature du marquage

- Segmentation des caractères
- Erreurs de marquage.
- Variations de la qualité du marquage dans le temps
- Qualité de l'encre (réflectance)

d) Les problèmes liés à la variété des polices

- Reconnaissance multifonte
- Polices industrielles spécifiques (OCR-A, OCR-B, MICR,...)
- Caractères fonction de la police mais également du marquage utilisé (tirets, points)

IV.1.5 Proposition d'un système

L'objectif de notre travail consiste à définir un nouveau système de reconnaissance qui puisse résoudre certains problèmes de la reconnaissance de caractères industriels. Notre but est de définir un système robuste aux variations lumineuses. Etant donné la nature des contrôles à effectuer, il est également nécessaire d'introduire une reconnaissance invariante qui permette au système d'effectuer sa tâche de reconnaissance quel que soit le positionnement du texte et quelle que soit sa taille.

Nous avons voulu créer un système simple qui puisse dans le cadre d'une réalisation pratique avoir des temps de traitement extrêmement courts afin d'être compatible avec des cadences de production très élevées dans l'hypothèse d'une réalisation.

Comme l'indique notre étude bibliographique des parties 2 et 3, nous nous sommes orientés vers l'utilisation des techniques neuronales utilisées en reconnaissance de formes pour construire notre système. Nous avons cherché à développer un système autoadaptatif basé sur des algorithmes courants et fiables, mais en les adaptant sous forme d'architecture neuronale de manière à permettre une implémentation parallèle. Nous avons éliminé de l'étude du système de

reconnaissance, les parties concernant la segmentation et le post-traitement. Des techniques déjà éprouvées [BEL92][KUN93][HEU94][DEF95] peuvent parfaitement s'articuler autour de notre système de reconnaissance de caractères décrit dans ce chapitre.

Nous avons cherché à introduire dans notre système de nouveaux concepts qui permettent de réaliser l'adaptation du système aux variations des conditions d'acquisition. Cette adaptation repose principalement sur la coopération entre un étage qui va effectuer l'extraction du caractère de l'image et un étage de classification. Le système a été construit à l'aide de deux réseaux de neurones et d'une boucle de retour permettant d'assurer l'autoadaptabilité du système (figure IV.12).

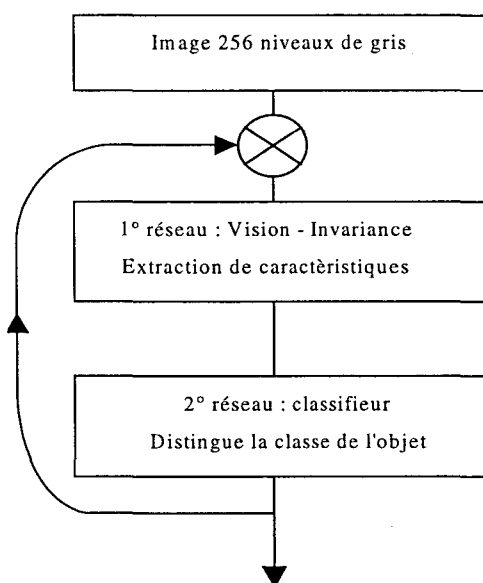


Figure IV.12 : Architecture générale

Le premier réseau, spécialisé pour la vision extrait à partir d'une image, la représentation du caractère. Nous avons dû développer le premier étage de vision de manière à faciliter l'adaptation aux variations lumineuses. En plus de cette propriété d'adaptation, ce premier étage est doté de propriétés d'invariance en position, en taille et en orientation. Ainsi, quelles que soient la position et la taille du caractère, quelle que soit l'orientation du support, quels que soient le type de support et le type de marquage utilisés, cet étage doit être capable d'extraire un modèle binaire et invariant du caractère.

Le second réseau, spécialisé dans la classification reconnaît la classe du modèle présenté par le premier réseau. Cet étage a été conçu à partir d'un réseau classifieur du type Radial Basis Functions. Nous avons bâti ce classifieur de manière à effectuer grâce à un processus d'apprentissage, une reconnaissance multiforme et multimarquage. La principale évolution de ce classifieur sera d'introduire la notion de forme. Cette notion permettra de limiter la taille de l'architecture tout en permettant la reconnaissance de caractères très spécifiques au marquage industriel.

La dernière technique mise au point pour ce système est la boucle de coopération. Cette coopération, rendue possible par l'ajout de quelques neurones, va exploiter les résultats de l'étage de vision et de l'étage de classification pour venir modifier des paramètres de la partie vision du premier réseau et ainsi permettre l'adaptation automatique au type de support utilisé et au type de marquage.

Les trois paragraphes suivants vont décrire les concepts utilisés dans chacun des deux étages du système et la boucle de retour.

Le premier des paragraphes, consacré à la description de la partie vision et invariance, mettra en évidence les concepts développés pour assurer une reconnaissance de caractères peu sensible aux types de support et de marquage.

Le deuxième paragraphe présentera une étude faite sur le réseau classifieur. Après avoir déterminé, dans la deuxième partie, le réseau du type Radial Basis Functions comme étant l'architecture la plus adaptée en fonction des performances de classification, de la facilité d'implémentation et des temps de traitement, nous présenterons, ici, les modifications apportées afin d'améliorer le fonctionnement du classifieur en fonction des résultats du premier réseau et de l'utilisation de fontes et de marquage industriels.

Enfin, le dernier paragraphe détaillera la structure et le fonctionnement de la boucle de retour qui, en fonction des résultats de classification, modifie les paramètres du premier réseau.

IV.2 Le premier réseau

Le rôle du premier réseau est d'extraire l'information du caractère issu d'une image. L'architecture de ce réseau est une architecture à couches composées de neurones formels classiques. Chaque neurone est totalement connecté aux neurones de la couche précédente. A partir d'une couche d'entrée I constituée de $m \cdot n$ neurones représentant les pixels de la caméra, le réseau traite l'information selon trois opérations successives. Il effectue premièrement des opérations simples telles que le lissage et le seuillage puis des opérations plus complexes lui permettant d'obtenir en sortie un modèle normalisé en position, en rotation et en taille (figure IV.13).

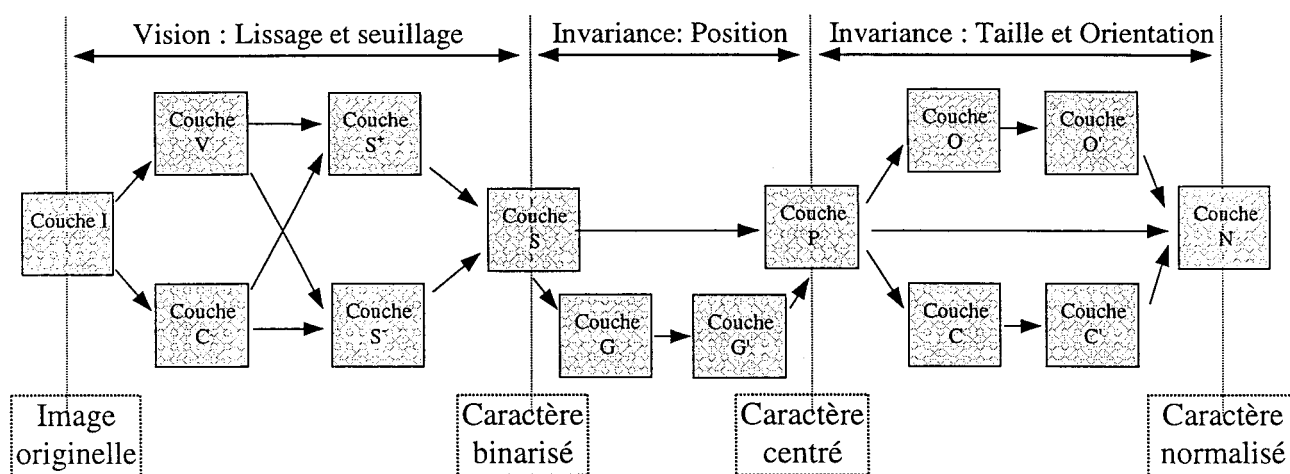


Figure IV.13 : Architecture du premier réseau

Sa première tâche est d'effectuer une binarisation de l'image afin de n'extraire que l'information du caractère. A partir de la couche d'entrée, la première partie du réseau, (constituée de 5 couches : V, C, S*, S, S) effectue cette opération de vision. Cette opération est plus complexe qu'une simple opération de seuillage. En effet, le système doit être capable de reconnaître un objet sombre sur un fond blanc (ex : caractère imprimé sur du métal) ou un objet clair sur un fond sombre (ex : caractère gravé sur du cuir).

Etant donné que notre système de reconnaissance doit être appliqué à un environnement industriel, une seconde tâche a été attribuée à notre réseau afin de permettre au système de s'affranchir de la connaissance a priori de la position du caractère à reconnaître. La seconde

partie réalisée par 3 couches : G, G', P consiste donc à modifier le modèle présenté en un modèle normalisé en position donc centré dans l'espace de représentation.

La dernière partie consiste à normaliser le caractère en taille et en orientation de manière à obtenir une reconnaissance quelles que soient l'orientation et la taille du modèle présenté à l'entrée grâce aux traitements réalisés par 5 couches : C, C', O, O', N.

Les différentes fonctionnalités du réseau sont présentées dans deux paragraphes. Le premier présente les opérations de lissage et de seuillage constituant la partie vision. Le second traite des opérations de normalisation en position, en taille et en orientation. Elle consiste à obtenir une représentation invariante du caractère.

IV.2.1 Opérations de Vision : Lissage et Seuillage

La première tâche effectuée par ce réseau est une tâche de vision. Elle permet d'extraire le caractère sous forme d'image binarisée. Les systèmes actuels de reconnaissance de caractères sont essentiellement basés sur la binarisation de l'image acquise pour ensuite effectuer la segmentation et la reconnaissance des caractères. L'algorithme généralement utilisé pour la binarisation est le seuillage global [WHI83][MOR92]. Dans le cas d'images compliquées, ce seuillage n'est plus suffisant. Une première solution est de proposer un système qui puisse effectuer un seuillage adaptatif [KAM93][TRI95] : pour différentes régions de l'image une valeur de seuil est choisie. Une approche récente [WU97] consiste à utiliser l'analyse de texture. La binarisation et la segmentation sont réalisées en distinguant la texture des caractères par rapport au fond.

Les études les plus récentes ont montré que la segmentation des caractères est plus efficacement réalisée à partir d'images en niveau de gris [COS89][GOR93][CHO95]. Notre système de reconnaissance de caractères est basé sur une utilisation au préalable de ces techniques. Ainsi, à l'entrée du système ici décrit nous considérons que les images présentées sont des images en niveau de gris de caractères isolés.

L'utilité de l'étude d'une partie vision est d'avoir permis de développer une nouvelle technique d'extraction de modèle binarisé d'un objet quelles que soient les caractéristiques lumineuses de l'objet et du fond. Cette technique a également la particularité de permettre l'adaptation de la qualité de l'extraction du modèle en fonction des résultats de sa classification.

L'opération réalisée par la partie vision de notre système est inspirée de la notion du filtrage centre-voisinage. Cette notion est issue d'une modélisation biologique : les champs récepteurs centre-voisinage. Les cellules ganglionnaires de type champ récepteur sont situées sur la rétine et sont de deux natures : les cellules on-center off-surround et les cellules off-center on-surround (figure IV.14).

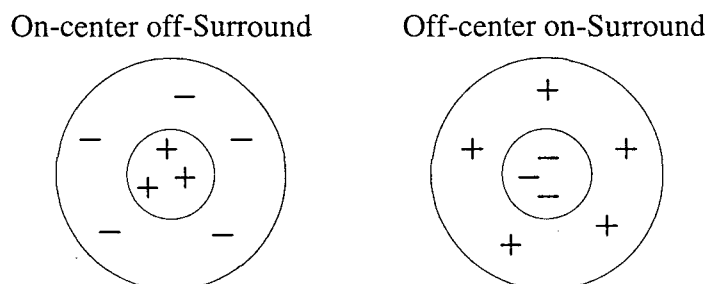


Figure IV.14 : Cellules Centre -Voisinage

Pour les cellules de type On-center Off surround, la zone centrale est excitée « + » par la lumière tandis que la zone éloignée est inhibée « - ». La comparaison des potentiels moyens sur chacune de ces deux zones définit le niveau de déclenchement du signal de sortie de la cellule. Ainsi, si la zone centrale reçoit plus de luminosité que la zone voisinage, la cellule sera active. A l'inverse, son potentiel de sortie sera nul. La cellule Off-center On-surround a le comportement opposé.

La taille des champs récepteurs de la cellule est caractéristique de la qualité de la détection réalisée. Si le champ est important, les cellules distinguent les caractéristiques grossières de la forme. A l'opposé, un champ étroit permet de distinguer les détails. Les figures IV.15 mettent en évidence ces propriétés.

La figure IV.15a montre que seul le champ large va émettre un signal d'excitation. En effet, dans le cas du champ étroit, la zone d'influence Centre et la zone d'influence Voisinage ne "voit" que la bande blanche. La différence entre ces deux zones étant nulle, le neurone à champ étroit est inactif. A l'inverse, le champ large met en évidence l'alternance bande blanche/bande noire.

En revanche, dans la figure IV.15(b), c'est le champ étroit qui va permettre de délivrer un signal d'excitation. La justification est identique. Dans le cas du champ large, la zone centre "voit" moitié bande blanche, moitié bande noire tout comme la zone voisinage. N'ayant pas de différence de potentiel entre les deux zones, le neurone reste donc inactif.



Figure IV.15 : Influence de la taille du champ récepteur

La première partie de notre réseau va exploiter cette notion de centre-voisinage. En effet, les deux régions d'une cellule centre-voisinage se comportent comme deux filtres passe-bas qui, appliqués sur l'image, lissent les potentiels situés dans une région donnée. Centrés sur le même pixel, un filtre effectue un lissage des pixels proches du centre et un autre effectue un lissage avec les pixels situés dans le voisinage. Nous avons facilement modélisé l'influence d'un filtre sur un potentiel par un coefficient de filtrage (CF) fonction de la distance (D) entre le potentiel traité et le centre du filtre. Deux fonctions CF^C et CF^V sont définies pour une cellule : l'une représentant l'influence du filtre Centre, l'autre l'influence du filtre Voisinage (figure IV.16). Ces fonctions sont définies de manière à ce que le nombre de pixels influencés par chacune des deux régions soit identique. Ces deux fonctions sont également définies à partir d'un paramètre σ . Il caractérise la taille d'influence du filtre centre-voisinage par analogie avec la taille du champ-recepteur de la cellule centre-voisinage.

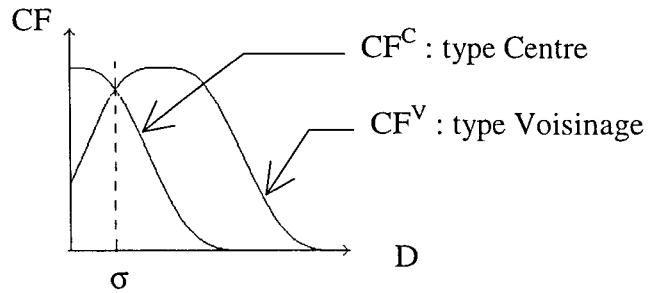


Figure IV.16 : Influence de la cellule centre-voisinage

Appliquées sur l'image, ces deux fonctions vont définir deux régions de lissage distinctes. C'est en comparant les résultats des deux filtres, que l'on pourra réaliser aisément la binarisation. Nous avons choisi cette modélisation pour la partie vision afin d'obtenir une détection du caractère quelle que soit la différence de niveaux de gris avec le fond. En effet, le choix d'appliquer ces deux filtres puis d'effectuer la différence permet de s'affranchir facilement du niveau de gris du fond de l'image par rapport à l'objet.

Les opérations précédemment présentées ont été réalisées suivant une modélisation neuronale. En plus de la couche d'entrée, quatre couches de neurones ont été définies. Les deux premières C (Centre) et V (voisinage) réalisent des opérations de lissage et les deux suivantes S⁺ et S⁻ réalisent le seuillage.

La première couche utilisée, de dimension m×n, est la couche d'entrée du système notée I. Cette couche permet de présenter l'image initiale au réseau. Sur celle-ci, chaque neurone de coordonnées (i,j) est représenté par le pixel de mêmes coordonnées.

Les couches de neurones C et V sont définies pour effectuer les deux opérations de lissage de l'image de départ suivant les deux régions particulières. Ainsi, l'activité d'un neurone (i,j) de la couche C représente le résultat du lissage d'un filtre Centre centré sur (i,j). La connexion pondérale entre un neurone (i,j) de la couche C et un neurone (p,q) de la couche I représente le coefficient du filtre suivant CF^C(D) avec D : distance séparant le neurone (p,q) du neurone (i,j). Ainsi, les connexions entre le neurone (i,j) de la couche C et les neurones de la couche I définissent la région de lissage Centre (figure IV.17).

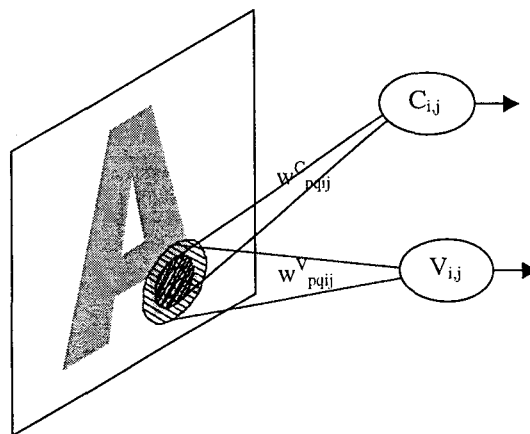


Figure IV.17 : Schéma des connexions entre la couche d'entrée et les neurones des couches C et V

L'activité C_{ij} du neurone i,j situé sur la couche C est définie par :

$$C_{ij} = \sum_{p=0}^{m-1} \sum_{q=0}^{n-1} w_{pqij}^C \cdot I_{pq} = \text{(par simplification de notation)} \quad \sum_{pq} w_{pqij}^C \cdot I_{pq} \quad (\text{IV.1})$$

Le poids w^C décrit les coefficients de lissage. Il permet de définir la zone d'influence "Centre". Le lissage est obtenu en moyennant l'activité des neurones à l'intérieur de cette région de lissage.

Ce poids décrit en (IV.2), fonction de la distance $D_{pqij} = \sqrt{(p-i)^2 + (q-j)^2}$, définit une zone où seuls les neurones (p,q) proches du neurone (i,j) ont une influence suffisante (figure IV.18).

$$w_{pqij}^C = \exp\left(-\left(\frac{\sqrt{(p-i)^2 + (q-j)^2}}{\sigma}\right)^\varepsilon\right) \quad (IV.2)$$

Le coefficient ε définit l'allure de la variation du poids en fonction de la distance. Une valeur faible définit une variation importante, à l'inverse, une valeur importante définit une variation faible. Pour nos essais, nous avons fixé cette valeur à 3.

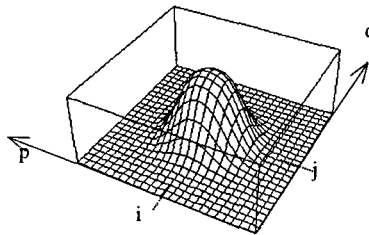


Figure IV.18 : Représentation du poids w_{pqij}^C

La couche V est construite sur le même principe mais définit le filtrage Voisinage. L'activité V_{ij} d'un neurone de la couche V est définie par le lissage de l'activité de chaque neurone à l'intérieur d'une région particulière définie grâce aux poids w_{pqij}^V .

$$V_{ij} = \sum_{pq} w_{pqij}^V \cdot I_{pq} \quad (IV.3)$$

Les poids w^V (eq IV.4) de la couche V définissent le lissage avec des neurones éloignés de celui traité. La figure IV.19 représente le voisinage défini par ces poids.

$$w_{pqij}^V = \exp\left(-\left(\left|\frac{\sqrt{(p-i)^2 + (q-j)^2}}{\sigma} - 1\right|\right)^\varepsilon\right) \quad (IV.4)$$

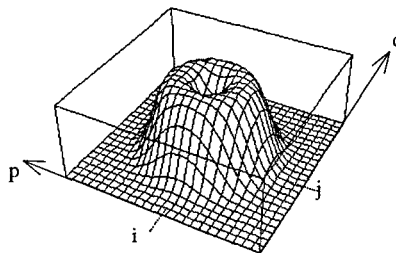


Figure IV.19 : Représentation du poids w_{pqij}^V

Les expressions (IV.2) et (IV.4) des poids montrent que les deux voisinages sont définis à l'aide d'un paramètre σ . Celui-ci détermine la taille de la zone d'influence d'une cellule centre-voisinage (figure IV.16).

Les figures IV.20 suivantes représentent les potentiels issus des couches C et V pour un exemple présenté sur la couche I.

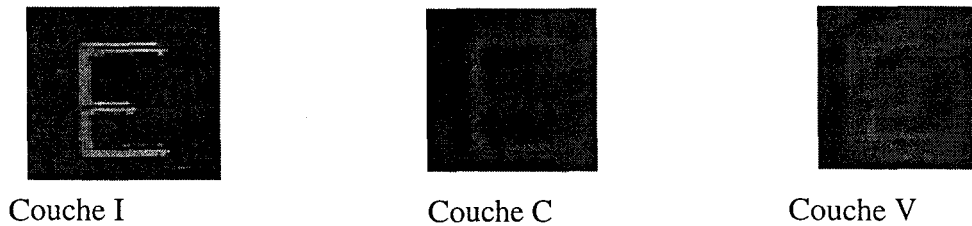


Figure IV.20: Exemples de couche C et V

Après les couches C et V, les deux couches suivantes S^+ et S^- réalisent la binarisation par seuillage. La couche S^+ effectue la soustraction de l'activité de la couche C à la couche V tandis que la couche S^- effectue la soustraction de l'activité de la couche V à la couche C. L'activité du neurone (i,j) de la couche S^+ et celle du neurone (i,j) sur la couche S^- sont définies par :

$$S_{ij}^+ = \xi(C_{ij} - V_{ij}) \quad (\text{IV.5})$$

$$\text{et} \quad S_{ij}^- = \xi'(V_{ij} - C_{ij}) \quad (\text{IV.6})$$

Le seuillage est réalisé grâce à la fonction d'activation $\xi(x)$ des neurones définie (figure IV.21) comme un échelon fonction de l'activité.

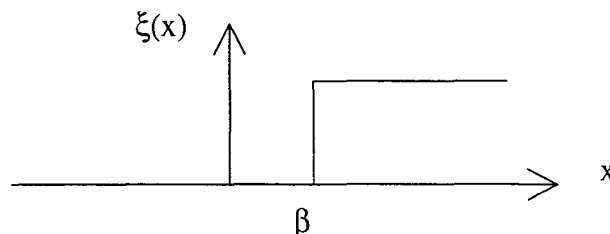


Figure IV.21 : fonction seuillage

La fonction $\xi'(x)$ est identique à $\xi(x)$ seule la valeur du seuil (β') est différente. Le fait de définir un seuil β' dont la valeur diffère de celle de β peut permettre de compenser le fait que les deux régions de lissage (Centre et Voisinage) n'aient pas une influence strictement identique sur les pixels de l'image. L'expérience montre qu'il est généralement préférable de choisir une valeur du seuil β' légèrement inférieure à celle de β .

Le fait d'avoir défini ces quatre couches nous permet de réaliser aisément la binarisation du caractère. Grâce aux régions de lissage définies par les poids w_{pqij}^C et w_{pqij}^V et aux fonctions de seuillage, nous remarquons, que des deux couches S^+ et S^- , la couche ayant une activité neuronale la plus faible correspond à la représentation binaire du caractère. En effet, puisque le caractère a une représentation localisée par rapport au fond, mettre en évidence le caractère par rapport au fond apporte moins d'information que l'inverse. Ainsi, si la couche S^+ est la couche ayant le moins d'activité, le seuillage a permis de mettre en évidence un objet sombre par rapport à un fond clair (exemple du "C" dans la figure IV.22). La couche S^- permet de mettre en évidence l'inverse : un objet clair par rapport à un fond sombre (exemple du "S" dans la figure IV.22).


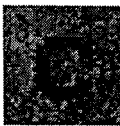










Couche I	 ($\sigma=4$)		 ($\sigma=7$)	
Couches C et V				
Couches S ⁺ et S ⁻				
Couche S				

Figure IV.22 : Rôle des couches S⁺ et S⁻

Pour la suite du traitement, des deux couches S⁺ et S⁻, la couche ayant l'activité totale la plus faible et donc représentant le caractère binarisé, est définie comme couche S.

Les propriétés mises en évidence dans cette partie Vision permettent de distinguer facilement le caractère du support. Une phase de test concernant le fonctionnement de la partie vision a mis en évidence l'avantage de regrouper le lissage et le seuillage au niveau des quatre premières couches. Le fait d'avoir défini les couches C et V réalisant deux régions de lissage avec un seul paramètre significatif σ puis les couches S⁺ et S⁻ avec le seul paramètre β nous a permis d'obtenir une adaptation avec seulement deux paramètres à régler (la valeur de β' a été fixée à 90% de la valeur de β). Le tableau suivant représente les résultats obtenus à l'issue de la partie vision pour un modèle donné mais pour différentes valeurs de seuil et de voisinage pour un modèle représenté figure IV.23.



Figure IV.23 : Image de départ (40 pixels * 36 pixels)















































































$\beta \backslash \sigma$	1	2	3	4	5	6	7	8	9	10	11	12	13
1													
2													
3													
4													
5													
6													

Tableau IV.1 : Influences de σ et β sur la qualité de la binarisation

Le tableau précédent montre l'influence des deux valeurs sur la binarisation.

On remarque que le paramètre σ agit sur la qualité de la définition de l'image résultante. En effet, plus la valeur est importante, plus les régions de lissage sont importantes et plus on perd de détails. A l'inverse, plus la région de lissage est faible, plus on met en évidence les détails : les contours mais aussi le bruit.

La valeur de seuil β ne doit également pas être trop faible pour ne pas augmenter le bruit. Ce seuil va agir sur la mise en relief de l'objet par rapport au fond. Si cette valeur est élevée, seuls les pixels présentant le maximum de différences entre la région centre et la région voisinage seront présents. Ils représentent généralement le squelette de l'objet.

De ces deux paramètres, on remarque que σ agit le plus nettement sur le résultat. En effet, la moindre variation de sa valeur entraîne des modifications très marquantes sur l'image résultante. A l'inverse, l'influence d'une variation de la valeur du seuil agit légèrement.

Les courbes (figures IV.24 et IV.25) quantifient ces résultats en déterminant la qualité de la binarisation en fonction des variations du seuil β et du voisinage σ . La qualité est caractérisée par un taux de correspondance (IV.7) entre le modèle binaire obtenu et le modèle optimum. Le critère simple choisi est l'erreur entre la surface (S) obtenue en modifiant les paramètres (σ, β) et la surface (R) choisie comme référence.

$$TC = 1 - \frac{|R - S|}{R} \quad (IV.7)$$

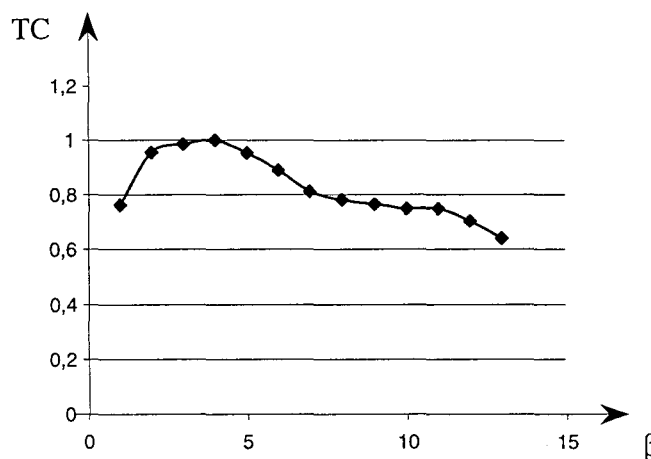


Figure IV.24 : Taux de correspondance en fonction du seuil β pour $\sigma = 4$

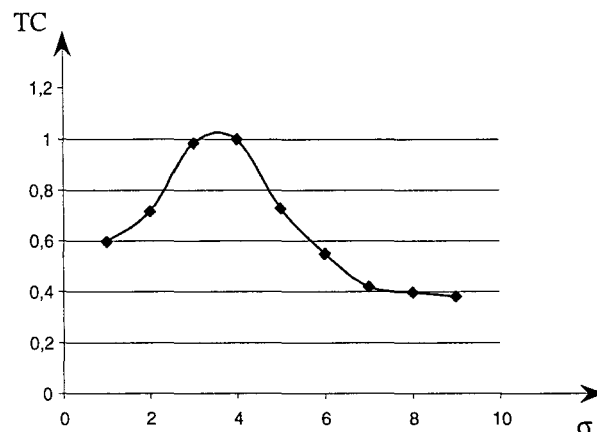


Figure IV.25 : Taux de correspondance en fonction du voisinage σ pour $\beta = 3$

Dans le souhait d'obtenir une adaptation à tout type de marquage et à tout type de support, il faut pouvoir distinguer des objets de couleur proche du fond bien que ce dernier puisse être bruité. Notre modélisation nous permet cette adaptation en ne modifiant que la taille σ . En effet, la variation du seuil β agit peu sur la qualité de la binarisation (figure IV.24). En revanche, c'est en variant faiblement la taille du voisinage σ que l'on peut améliorer le taux de correspondance (figure IV.25). Nous avons montré que cette valeur est la plus influente sur la qualité de la binarisation lors de variations.

La valeur du seuil β doit être fixée la plus faible possible afin de permettre la distinction entre les régions centre-voisinage tout en évitant de mettre en évidence du bruit. Le tableau IV.2 montre qu'en fixant le seuil β à 3 et en variant le paramètre σ on obtient une représentation binaire correcte du caractère à distinguer. Les limites sont bien évidemment obtenues lorsque la différence des niveaux de gris entre l'objet observé et le fond est proche de la valeur du seuil β . En dehors de cette limite, quelle que soit l'image présentée, c'est en effectuant un choix correct de la valeur de σ que l'on obtiendra une représentation du caractère binarisé.

Image de départ	Niveaux de gris	$\sigma=1$	$\sigma=2$	$\sigma=3$	$\sigma=4$	$\sigma=5$	$\sigma=6$	$\sigma=7$	$\sigma=8$
	Fond : 250 Objet : 10								
	Fond : 208 Objet : 32								
	Fond : 192 Objet : 48								
	Fond : 176 Objet : 64								
	Fond : 160 Objet : 80								
	Fond : 144 Objet : 96								
	Fond : 128 Objet : 112								
	Fond : 128 Objet : 123								

Tableau IV.2 : Influence du paramètre σ sur différents niveaux de gris pour l'objet et le fond

Un dernier facteur intervenant sur la qualité de la binarisation est la taille du caractère à extraire. Nous avons réalisé une étude qui a permis de montrer que, quelle que soit la taille du caractère à extraire, seule une modification du paramètre σ permet d'obtenir une image résultat correcte. Le tableau IV.3 résume cette étude. Il montre, pour différentes tailles d'images, la valeur du paramètre σ optimum. Ces images ont été obtenues en zoomant ou réduisant une image de référence. Ainsi, ce tableau caractérise surtout l'influence des traits des caractères (la graisse) sur la valeur de σ .

Taille des caractères	L :15 pixels H : 14 pixels	L :30 pixels H : 28 pixels	L : 60 pixels H : 56 pixels	L : 120 pixels H : 114 pixels	L : 300 pixels H : 280 pixels
Valeur de σ	1	3	7	14	25

Tableau IV.3 : Influence de la taille du caractère

En conclusion, la variation du seul paramètre σ permettra l'adaptation du système de reconnaissance au type de marquage ou/et de supports et ce quelle que soit la taille du caractère.

A partir de ce modèle binarisé, on cherche à obtenir une représentation normalisée afin de pouvoir s'affranchir des variations concernant la localisation et la taille du caractère lors de la procédure d'acquisition.

IV.2.2 Opérations de normalisations en position, en taille et en orientation

Afin d'éviter l'utilisation d'architectures neuronales trop complexes et pour certaines difficilement réalisables, nous avons créé un système connexionniste permettant d'effectuer une reconnaissance invariante. L'invariance va consister à repositionner le caractère en fonction de son centre de gravité puis à modifier sa représentation selon une taille et une orientation désirées. Les techniques utilisées ont été inspirées des techniques présentées dans la troisième partie. Nous avons conservé une architecture à blocs [YUC93]. Le premier effectue une normalisation en position, le second la normalisation en taille et en orientation. Partant du fait que les caractères à reconnaître sont issus de fontes parfaitement normalisées, c'est en adaptant des algorithmes simples sous forme d'architecture neuronale que l'on a pu construire ces différents blocs. Le premier constitué de 3 couches (G,G' et P) va permettre d'obtenir une invariance en position (figure IV.13). Il s'agit de translater le caractère pour le positionner au centre de l'image et donc au niveau neuronal de translater l'activité neuronale de la couche S vers le centre de la couche P. Le second bloc, constitué de 5 couches (O,O',T,T' et N) va effectuer l'invariance en taille et en orientation. Les principes mis en œuvre dans les deux blocs sont détaillés dans les deux paragraphes suivants.

Le réseau qui est développé pour effectuer l'invariance en position, en taille et en orientation n'est pas spécifique à la reconnaissance de caractères. Nous avons cherché à développer un système capable d'effectuer une reconnaissance invariante de formes. Pour illustrer le fonctionnement de notre réseau, nous utiliserons comme image test le support représenté figure IV.26. La figure IV.27 représente l'image binarisée que l'on pourrait obtenir à la sortie de la couche S. C'est à partir de cette image que le réseau va effectuer les différentes invariances.

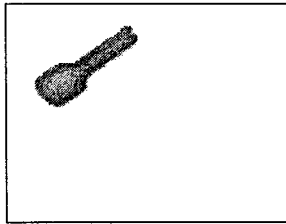


Figure IV.26 : Objet test

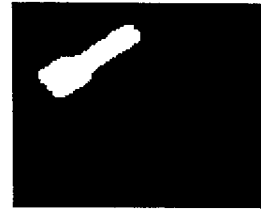


Figure IV.27: Image binarisée

IV.2.2.1 Invariance en position

La deuxième partie du réseau permet de normaliser le caractère en position. Le principe consiste à extraire en premier lieu le centre de gravité du caractère puis de pratiquer l'opération de translation entre ce point et le centre de la couche.

Notre choix d'effectuer l'invariance par centre de gravité plutôt que par centre géométrique est lié à l'obtention d'une représentation plus discriminante pour le classifieur.

Dans le cas d'une invariance par centre géométrique, la représentation obtenue après normalisation en taille utilise la totalité de la couche de sortie. En revanche, l'utilisation du centre de gravité n'utilise pas toute la résolution possible sur la couche de sortie mais permet d'obtenir une représentation plus discriminante. La figure IV.28 représente deux exemples de différences entre centre de gravité et centre géométrique. Dans le cas d'une normalisation d'un caractère "T" par utilisation de l'information du centre de gravité ou celle du centre géométrique, la différence de représentation permet dans le cas du centre de gravité d'avoir une représentation plus distincte par rapport à la lettre "I".

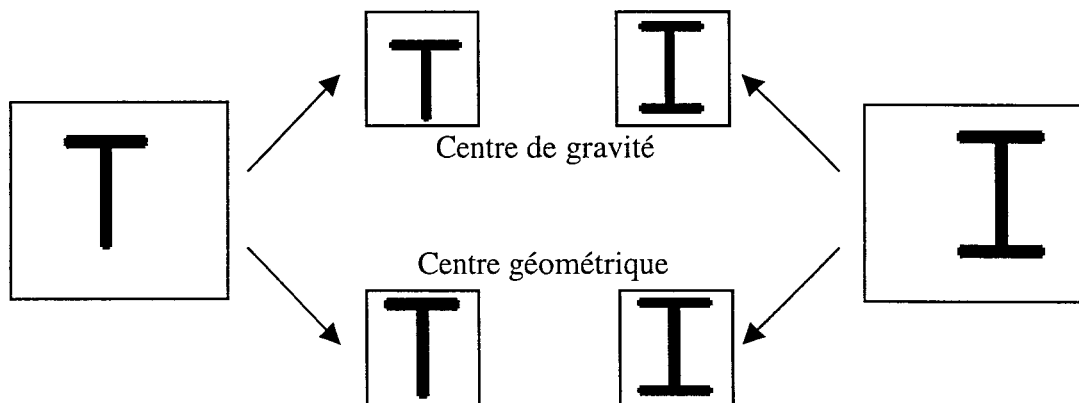


Figure IV.28 : Différences entre centre de gravité et centre géométrique

Pour effectuer l'invariance en position à partir de la couche S, 3 couches de neurones sont utilisées : les couches G et G' et la couche P (figure IV.13). On cherche tout d'abord à extraire l'information concernant la position du caractère. Cela consiste à localiser un neurone qui coïncide avec le centre de gravité de l'activité neuronale de la couche S. Pour cela, on utilise les deux couches G et G'. Pour chaque neurone (i,j) de la couche G, on détermine son activité en sommant l'activité de chaque neurone (p,q) de la couche précédente pondérée par la distance les séparant. On définit cette activité G_{ij} par :

$$G_{ij} = \sum_{pq} D_{pqij} \cdot S_{pq} \quad (IV.8)$$

avec D_{pqij} : distance entre le neurone (p,q) du neurone (i,j) telle que $D_{pqij}^2 = (p - i)^2 + (q - j)^2$

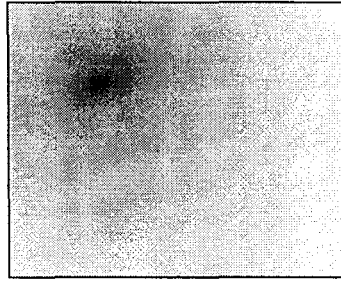


Figure IV.29 : Représentation de la couche G

On remarque (figure IV.29) que le neurone de la couche G d'activité minimale (en noir sur la figure IV.29) correspond au centre de gravité de la forme. On cherche à le mémoriser grâce à la couche G' en utilisant le principe de l'interaction du type "Loser Take All", ne conservant que le neurone correspondant au potentiel le plus faible. En effet, l'interaction est basée sur la compétition entre neurones d'une même couche. Après un temps de convergence, tous les neurones de la couche à l'exception d'un neurone, sont d'activité nulle. Ce neurone est, avant la convergence, le neurone d'activité la plus faible. Dans notre cas, le neurone d'activité la plus faible de la couche G est le seul neurone d'activité G'_{ij} égale à 1 sur cette couche après convergence.

La couche suivante sert à effectuer l'invariance en position. Cette transformation permet de faire coïncider le centre de gravité du caractère et le centre de l'espace de représentation (le centre de la couche). Une nouvelle couche P de neurones effectue la normalisation en position en utilisant le résultat de la couche S ainsi que celui de la couche G'. Nous définissons l'activité P_{ij} d'un neurone (i,j) de cette couche P par :

$$P_{ij} = \sum_{pq} w_{pqij}^P \cdot S_{pq} \quad (IV.9)$$

L'utilisation du poids w_{pqij}^P dans l'équation (IV.9) permet de définir une translation de l'activité des neurones. Pour cela, nous définissons le poids par :

$$w_{pqij}^P = \sum_{l=1}^m \sum_{k=1}^n G'_{kl} \cdot \exp[-\kappa \cdot (p - k - i)] \cdot \exp[-\kappa \cdot (q - l - j)] \quad (IV.10)$$

avec κ choisi très grand par expérience.

Le caractère a maintenant une position normalisée dans ce nouvel espace de représentation qui est la couche P. Son centre de gravité correspond au centre de cette couche.



Figure IV.30 : Image après invariance en position

Après avoir effectué cette invariance en position, la dernière partie de ce premier réseau va introduire les notions d'invariance en taille et en orientation. La normalisation en taille s'avère essentielle pour la reconnaissance de caractères. Ainsi, pour chaque police il n'est nécessaire d'apprendre que la forme de caractères constituant une fonte. De fait, quelle que soit la taille choisie pour la police, le modèle à reconnaître n'est fonction que de la fonte. L'invariance en orientation est plus discutable. En effet, certaines lettres ne peuvent garantir une invariance en orientation correcte. Ainsi, pour certaines polices, on ne peut distinguer la lettre "w" du "m", "d" du "p", "q" du "b",... Cette invariance dans le cadre de caractères isolés ne peut être garantie à 180 degrés près.

Les principes ici développés sont simples. Ils consistent à déterminer la taille de la forme à étudier puis son orientation globale. Une fois les caractéristiques concernant la taille et l'orientation du caractère connues, la dernière étape consiste à modifier la couche P, modèle invariant en position, de manière à faire correspondre la taille du caractère à une taille désirée et l'orientation du caractère avec l'axe vertical.

IV.2.2.2 Caractéristiques en taille et en orientation

Dans les deux cas, deux couches supplémentaires sont nécessaires T et T' pour la taille et O et O' pour l'orientation. Contrairement aux précédentes, elles sont à une seule dimension. Les couches T' et O' permettent de mémoriser la taille et l'orientation du caractère selon le même principe que la couche G'.

Pour extraire la taille du caractère, on définit une première couche T, constituée d'un nombre (Nb) de neurones identique à la moitié de la taille de la couche de neurones P. Cette nouvelle couche permet de déterminer la quantité d'activité neuronale contenue dans différents voisinages. Ces voisinages sont centrés au milieu de la couche P et leur taille est définie par l'indice r de chaque neurone de la couche T. Nous définissons l'activité T_r de ses neurones par :

$$T_r = \varphi \left(\sum_{pq} w_{pqr}^T \cdot P_{pq} \right) \text{ avec } \varphi(x) = \exp\left(\frac{x}{\lambda}\right) - 1 \quad (\text{IV.11})$$

avec $w_{pqr}^T = 1$ si $p^2 + q^2 < r^2$ et = 0 ailleurs.

L'activité des neurones (r) de la couche T est fonction de l'activité neuronale des neurones (p,q) de la couche P contenue dans une région définie par leur indice.

Le terme $\sum_{pq} w_{pqj}^T \cdot P_{pq}$ représente la sommation de l'activité des neurones (p,q) à l'intérieur d'une région définie par le paramètre r (En dehors de cette région le poids w_{pqr}^T est nul) (figure IV.31).

Le caractère exponentiel de $\varphi(x)$ et le choix de la valeur λ permettent d'obtenir une différence importante d'activité entre deux neurones même si les voisinages qu'ils définissent contiennent une activité neuronale proche. Ainsi, à l'issue du traitement effectué par la couche T, le neurone T_r le plus actif détermine grâce à son indice la taille du caractère (ou de l'objet) à normaliser.

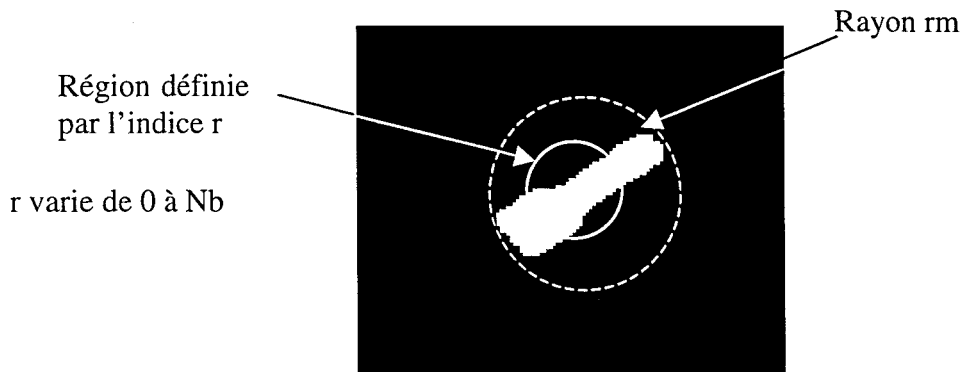


Figure IV.31 : Détermination de l'activité T_r

La figure IV.32 représente l'activité des neurones de la couche sous forme d'image. Plus l'activité est forte, plus le niveau de gris est élevé.

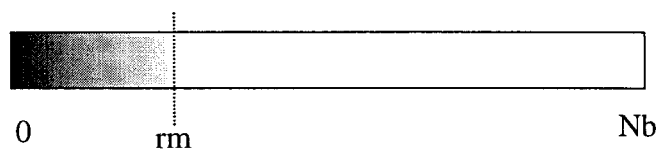


Figure IV.32 : Représentation de la Couche T

Ensuite, à l'aide de la couche T' , on obtient un seul neurone d'indice (rm) dont l'activité (T'_{rm}) est non nulle. Son indice nous permet de mémoriser la taille de l'objet rencontré. Pour déterminer le seul neurone actif, la couche utilise une interaction du type "Winner Take All" en tenant compte du voisinage associé à chaque neurone de la couche T. On mémorise alors le neurone d'activité T_r la plus forte avec le voisinage r le plus faible possible.

Après avoir défini la taille du caractère, on cherche à extraire son orientation générale. Pour cela, nous définissons deux couches de neurones O et O'. Le principe utilisé pour la détermination de l'orientation générale du caractère est la projection de l'activité neuronale de la couche P sur différentes orientations. Pour cela, la couche O est constituée de 180 neurones correspondant aux différentes orientations d'angle (a) de 0 à 180 degrés. On détermine l'activité O_a de chaque neurone de la couche O, en sommant l'activité neuronale de chaque neurone (p,q) de la couche P pondérée par un poids w_{pqa}^O .

$$O_a = \sum_{pq} w_{pqa}^O \cdot P_{pq} \quad (IV.12)$$

Le poids w_{pqa}^O est fonction de la distance D séparant le neurone (p,q) du centre de la couche. Il symbolise la projection perpendiculaire de l'activité (P_{pq}) sur la droite dont l'orientation (a) est définie par l'indice du neurone (figure IV.33).

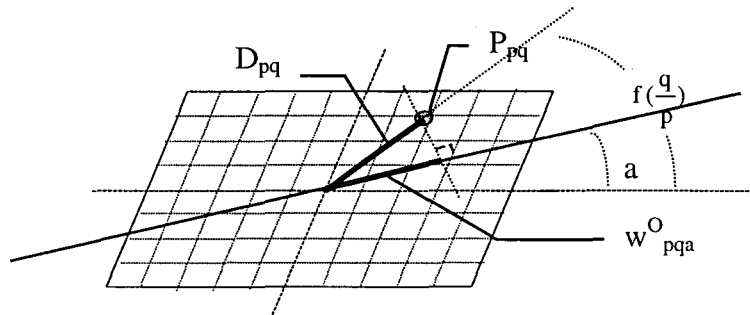


Figure IV.33 : Détermination du poids w_{pqa}^0

Le poids w_{pqa}^0 s'exprime donc simplement :

$$w_{pqa}^0 = D_{pq} \cdot \cos \left[u \left(f \left(\frac{q}{p} \right) - a \right) \right] \quad (\text{IV.13})$$

Avec $u(x) = \pi - |x|$ si $\frac{\pi}{2} \leq x \leq \frac{3\pi}{2}$;
 $= |x|$ ailleurs.

et $f\left(\frac{q}{p}\right) = \arctan\left[\frac{q}{p}\right] \quad p > 0, q > 0$

$$= \pi + \arctan\left[\frac{q}{p}\right] \quad p < 0, \forall q$$

$$= 2\pi + \arctan\left[\frac{q}{p}\right] \quad x > 0, q < 0$$

Il s'agit ensuite de comparer l'activité des différents neurones de la couche O. On remarque ainsi que le neurone ayant l'activité la plus forte permet de définir l'orientation principale du caractère. Ce neurone est mémorisé grâce à la couche O' par une interaction du type « Winner Take All ».

Cependant, la formulation précédente n'est pas suffisante. Par exemple, dans certains cas particuliers comme celui représenté figure IV.34 (illustrant la couche P où seuls les neurones de deux rangées verticales sont actifs : les deux traits verticaux), même en ayant un rapport hauteur-largeur largement supérieur à 1 ($L > H$), il n'est pas garanti de trouver le neurone O_{90} (décrivant une orientation verticale) comme étant le plus actif.

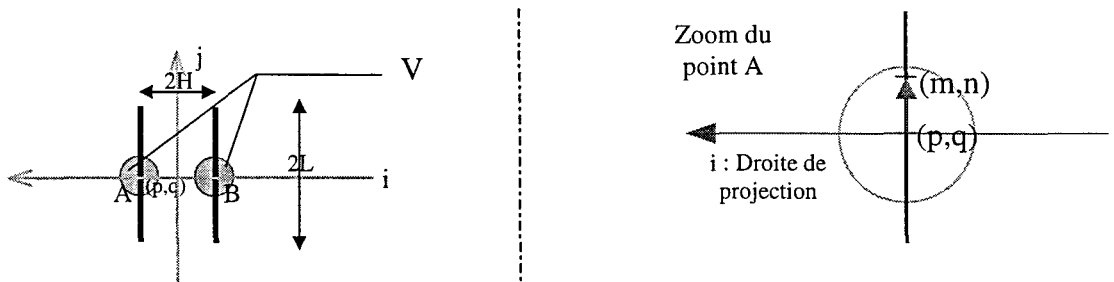


Figure IV.34 : Cas particulier pour la couche O

Si $(L+1) < 2H$, le neurone le plus actif de la couche O est le neurone O_0 (ou O_{180}). En effet, le calcul de l'activité du neurone O_a pour un angle $a=0$ consiste à sommer l'activité de tous les

neurones projetée sur l'axe horizontal. Or, pour ce calcul, l'influence des neurones des voisinages V, caractérisant la barre verticale, est trop importante. Bien que décrivant une orientation perpendiculaire à celle calculée, ces neurones vont contribuer au calcul. Cette erreur doit être résolue en négligeant leur activité. Ils ne doivent donc pas intervenir dans le calcul O_0 selon deux critères. Ils sont trop proches de la droite de projection et les neurones actifs décrivent entre eux une orientation perpendiculaire à celle calculée.

Il est donc nécessaire de faire intervenir un facteur fonction de l'orientation que décrit le neurone traité par rapport à l'angle (a). Ce facteur est calculé grâce aux neurones voisins. Dans le voisinage V centré en (p,q), on pondère l'activité de chaque neurone (m,n) en fonction de l'orientation décrite par ((p,q):(m,n)) par rapport à l'angle (a). En sommant sur tout le voisinage, on pondère ainsi l'activité du neurone (p,q) en fonction de l'orientation qu'il définit avec ses voisins par rapport à celle calculée (a). Ce facteur assure un renforcement si l'orientation générale à l'intérieur du voisinage V est parallèle à l'orientation (a) étudiée, et un amoindrissement pour une direction perpendiculaire. Nous modifions alors l'équation (IV.13) par :

$$w_{pq}^o = D_{pq} \cdot \cos \left[u \left(f \left(\frac{q}{p} \right) - a \right) \right] \cdot \Lambda \cdot \sum_{mn} P_{mn} \sin \left[f \left(\frac{q-n}{p-m} \right) - a \right] \quad (IV.14)$$

avec Λ coefficient de normalisation.

La figure IV.35 représente la couche O obtenue à partir du modèle représenté figure IV.31. L'activité des neurones O_a est représentée par une valeur de niveau de gris.

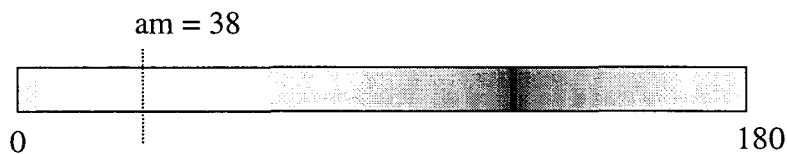


Figure IV.35 : représentation de la couche O

Après le traitement de cette couche O, une deuxième couche O' du type "Winner Take All" permet de conserver le neurone (O'_{am}) d'activité la plus forte. L'orientation globale (am) de la forme est alors mémorisée.

Concernant le résultat, nous avons remarqué que pour certains caractères symétriques, ayant un rapport hauteur-largeur supérieur à 1, deux neurones O_a ont une activité forte : pour un angle (a) faible mais aussi pour un angle ($\pi-a$). Il est alors impossible de déterminer sûrement l'orientation. Il faut donc appliquer dans ce cas les deux modèles issus de deux orientations possibles à l'entrée du second réseau.

IV.2.2.3 Mise à l'échelle ou normalisation

Après avoir mémorisé la caractéristique taille dans la couche T' et la caractéristique orientation dans la couche O', on peut effectuer une normalisation du caractère en taille et en orientation grâce à une unique couche. L'invariance en orientation consiste à faire coïncider l'orientation générale du caractère avec l'axe vertical. L'invariance en taille consiste à réduire ou agrandir le caractère de manière à normaliser sa taille à celle de la couche de sortie

$(2 \times \text{Dim} + 1) \times (2 \times \text{Dim} + 1)$. Ces deux normalisations sont réalisées à l'aide d'une seule couche. Cette couche appelée couche N est entièrement connectée aux neurones de la couche P, de la couche T' et de la couche O'. L'activité N_{ij} du neurone de coordonnées (i, j) (avec $-\text{Dim} \leq i$ et $j \leq \text{Dim}$) est définie par :

$$N_{ij} = \sum_{pq} w_{pq}^N \cdot P_{pq} \quad (\text{IV.15})$$

$$w_{pq}^N = \sum_a \sum_r O'_a T'_r \cdot \Psi \left[\left(\sin \left[f \left(\frac{q}{p} \right) - f \left(\frac{j}{i} \right) - a \right] \right)^n \right] \cdot \exp \left[- 2 \left(\frac{D_{pq}}{\frac{r}{\text{Dim}} \cdot \sqrt{i^2 + j^2}} - 1 \right)^2 \right] \quad (\text{IV.16})$$

$$\text{avec } \Psi[x] = \frac{1}{2}(x + |x|)$$

Le calcul du poids consiste à déplacer l'activité neuronale de la couche P suivant une rotation et une homothétie (figure IV.36). La nouvelle taille (Dim) est choisie en fonction de la taille désirée du modèle de sortie et la nouvelle orientation est verticale. Le centre de la couche définit le point invariant de la transformation. Le terme $\exp[\dots]$ permet de définir l'homothétie avec l'aide du rapport taille du caractère/ taille désirée. Le terme $\Psi[\dots]$ assure la réalisation de la rotation grâce à la couche O' permettant de déterminer l'angle de rotation.

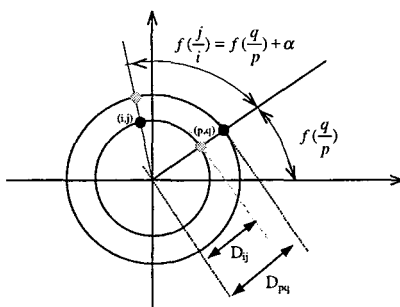


Figure IV.36 : Normalisation en taille et en orientation.

L'avantage de regrouper les deux invariances rotation-taille au sein d'une même étape est d'économiser du temps de traitement. Cependant, ceci n'est vrai que pour des images de petite taille. Pour des images importantes, il est beaucoup plus avantageux d'effectuer en premier lieu l'invariance en taille de manière à utiliser un espace de représentation plus restreint pour la détermination de la caractéristique orientation.

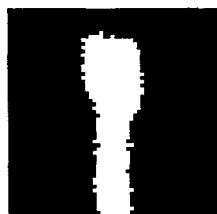


Figure IV.37 : Image normalisée

La figure IV.37 représente l'image obtenue à la sortie de ce premier réseau à partir de l'image de la figure IV.26. Il nous permet d'extraire à partir de l'image d'un caractère un modèle utilisable par un classifieur.

Nous avons effectué une phase de validation des étapes de normalisation à partir d'images de synthèse. Ces images, représentant un même objet ayant subi des translations et/ou des rotations et/ou des homothéties, ont été appliquées à notre réseau. Quelques résultats sont représentés dans le tableau IV.4. La phase de validité a montré un bon fonctionnement de notre réseau. Cependant de légères variations apparaissent sur différentes images. Ceci est principalement dû à la nature discrète des images. En effet, le positionnement du centre de gravité, le calcul de la taille ou le calcul de l'orientation ne sont pas toujours identiques. Ces mesures sont faites au pixel près.

Image initiale	Modèle invariant en position	Modèle invariant	Image initiale	Modèle invariant en position	Modèle invariant

Tableau IV.4 : Tests de l'invariance

Avant de développer la partie classification de notre système, il a été nécessaire d'utiliser ce premier réseau afin de déterminer les modèles invariants correspondant aux différents caractères et ainsi de définir la base d'apprentissage qui sera utilisée par le classifieur. En effet, le premier réseau utilise l'orientation générale du caractère pour le normaliser en orientation. Ainsi à la sortie du premier réseau, le caractère est représenté de sorte que son orientation générale coïncide avec l'axe vertical. Ceci ne correspond pas forcément avec l'orientation habituellement utilisée pour l'écriture. Ainsi, la figure IV.38 montre le caractère "L" tel que nous l'écrivons ainsi que son modèle invariant.



Figure IV.38 : Un exemple de caractère et de son modèle invariant

Le tableau IV.5 représente les modèles invariants obtenus pour les polices de caractères OCR-A et OCR-B. Ces modèles ainsi que ceux de nombreuses polices de caractères vont permettre de construire la base d'apprentissage du second réseau et d'obtenir une reconnaissance multiforme.

Norme OCR-B				Norme OCR-A Extended			
Caractère initial	Modèle invariant	Caractère Initial	Modèle Invariant	Caractère initial	Modèle invariant	Caractère initial	Modèle invariant
A		N		A		N	
B		O		B		O	
C		P		C		P	
D		Q		D		Q	
E		R		E		R	
F		S		F		S	
G		T		G		T	
H		U		H		U	
I		V		I		V	
J		W		J		W	
K		X		K		X	
L		Y		L		Y	
M		Z		M		Z	

Tableau IV.5 : Modèles invariants pour les polices OCR-A et OCR-B

Suite à ce premier réseau, un étage dit classifieur va utiliser le modèle invariant pour déterminer la classe d'appartenance du caractère présenté à l'entrée du système. Connaissant a priori les classes à distinguer et une grande partie de polices de caractères à reconnaître, une phase d'apprentissage supervisé va permettre de construire le classifieur.

C'est également à l'aide de l'étude des performances du classifieur que nous allons déterminer la valeur du paramètre Dim de l'équation (IV.16)

IV.3 Le deuxième réseau

Notre étude bibliographique de la deuxième partie a démontré que les techniques neuronales nous permettaient de construire des classifieurs performants et qu'elles constituaient actuellement les techniques les plus employées pour tenter d'améliorer la classification dans le cas de données complexes [JOD94a][HER94][BIS95][LOO96][RIP96]. C'est donc vers l'approche neuromimétique que nous avons orientée la construction du classifieur.

Notre démarche a consisté à évaluer les performances des principaux réseaux de neurones utilisés en classification afin de déterminer une architecture utilisable pour construire l'étage de classification de notre système. Les réseaux testés sont le perceptron multicouche (PMC), le réseau Radial Basis Function (RBF), le réseau Probabilistic Neural Network (PNN), les cartes auto-organisatrices (CAO), le Learning Vector Quantization (LVQ) et le réseau de Hamming. Nous les avons appliqués à un problème de reconnaissance de caractères. A l'issue de cette étude, dont les résultats ont été présentés dans la deuxième partie, notre choix s'est porté sur le réseau RBF. A partir de cette évaluation faite sur des cas quasi-idéaux (document papier ou fax), il est nécessaire d'optimiser le réseau initialement choisi afin de permettre la classification de caractères au sein de notre système de reconnaissance de caractères industriels. Une nouvelle base d'apprentissage a été définie en tenant compte tout d'abord des modèles invariants issus du premier réseau, puis des modèles liés aux spécificités de notre application.

La première partie de l'étude du réseau classifieur concerne sa construction. Il s'agit de déterminer la taille de l'architecture du réseau RBF et plus précisément de déterminer le nombre optimum de neurones de la couche d'entrée avec le compromis entre la qualité de reconnaissance et la taille d'architecture.

Il est également nécessaire d'améliorer le réseau RBF afin de l'adapter aux problèmes de reconnaissance de caractères industriels. Nous avons modifié le processus d'apprentissage et la structure afin de permettre la reconnaissance de caractères multifontes dont certaines fontes spécifiques liées à l'utilisation de systèmes de marquage industriel.

IV.3.1 Construction du classifieur

Nous avons construit le deuxième réseau en utilisant une modélisation proche du réseau RBF. Notre réseau est constitué de quatre couches (figure IV.39).

La couche d'entrée (M) est la couche de sortie (N) du premier réseau. Par analogie avec une image, cette couche peut être considérée comme bidimensionnelle. L'activité d'un neurone de coordonnée l (tel que $l = i + j * \text{largeur de l'imagette}$) de la couche M correspond au niveau binaire du pixel (i,j) de l'imagette du caractère normalisé.

La couche de sortie (S) permet de connaître la classe d'appartenance du modèle présenté à l'entrée. Le nombre de neurones constituant cette couche est identique au nombre de classes à distinguer. Ainsi, le seul neurone actif de cette couche détermine la classe.

Deux couches cachées (P et Q) sont utilisées. La première couche, la couche P, utilise des neurones de type fonction de base radiale pour estimer la densité de probabilité du modèle par rapport aux modèles appris. La seconde couche cachée, la couche Q, détermine la densité de probabilité d'appartenance du modèle aux différentes classes.

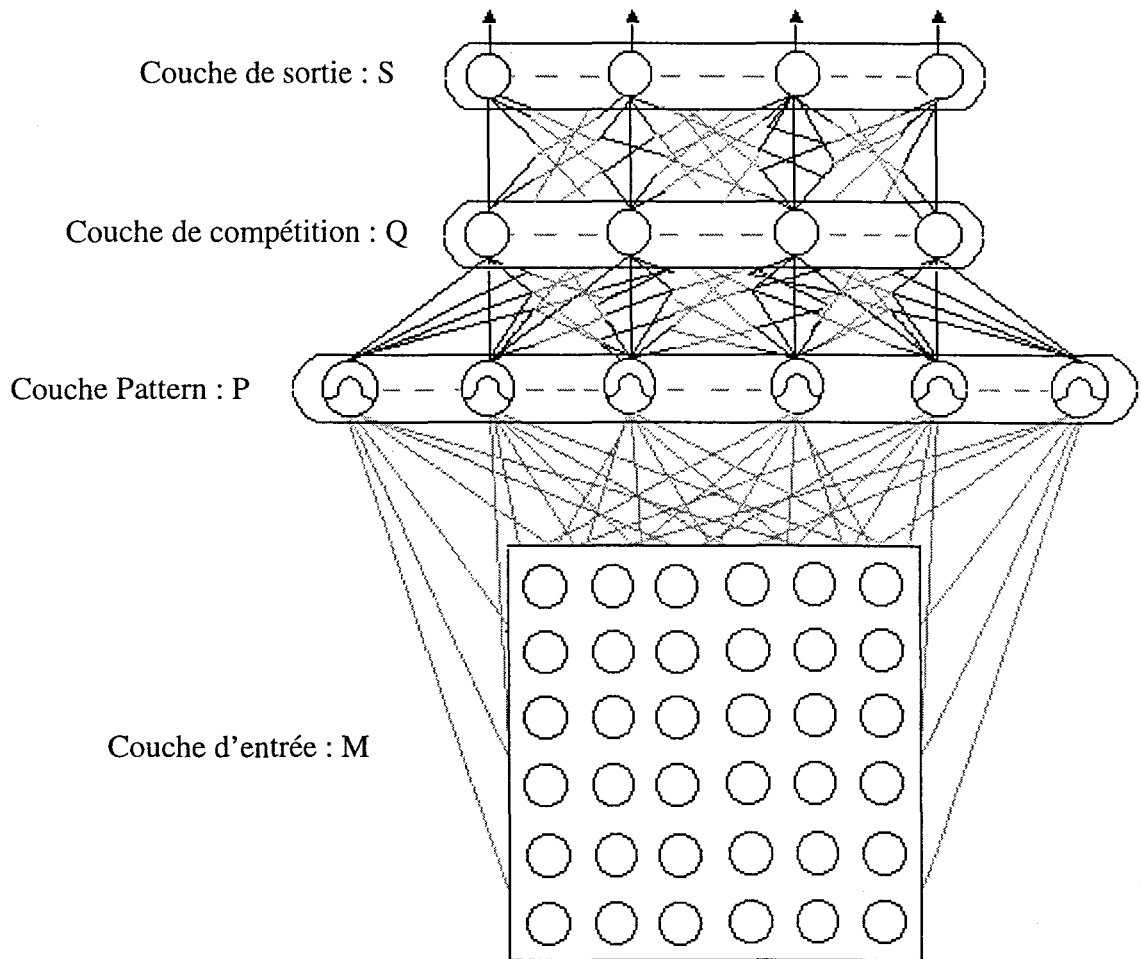


Figure IV.39 : Structure du second réseau

L'apprentissage du réseau est fait de manière supervisée. D'une manière simple, il consiste à associer à chaque modèle M (vecteur $M: \{m_0, m_1, m_2, m_3, \dots, m_n\}$) de la base d'apprentissage un neurone (k) de type fonction-noyau de la couche P ou aussi appelé neurone prototype. Schématiquement, l'apprentissage modifie les poids de connexion w_i^k de ce neurone k avec la couche d'entrée par les composants du vecteur d'apprentissage. Ainsi $w_i^k = m_i$

En phase de reconnaissance, la sortie des neurones de la couche P définit la fonction de densité conditionnelle $p(X/M^k)$ telle que le vecteur d'entrée X appartienne au modèle défini par le neurone (k). La fonction de transfert des neurones de la couche P est une gaussienne utilisant comme variable la distance entre le vecteur d'entrée ($X: \{x_0, x_1, x_2, x_3, \dots, x_N\}$) et le vecteur des poids de connexion $W: \{w_0^k, w_1^k, w_2^k, w_3^k, \dots, w_N^k\}$ entre le neurone caché (k) et la couche d'entrée. La figure IV.40 rappelle le fonctionnement d'un neurone (k) de la couche cachée P (voir 2^{ème} partie §II.3.4 (20)). Ainsi, si le vecteur X est identique au vecteur M appris et mémorisé dans les liaisons synaptiques W , la sortie du neurone k est maximale. La fonction de transfert va permettre d'estimer la densité à partir de l'erreur entre le modèle présenté et le modèle appris.

L'algorithme d'apprentissage est très simple mais impose un nombre de neurones sur la couche P équivalent au nombre de modèles à apprendre. Il doit donc être nécessaire d'inclure dans la base d'apprentissage tous les modèles des caractères issus du premier réseau et suivant pour des polices différentes afin d'obtenir une reconnaissance multifonte.

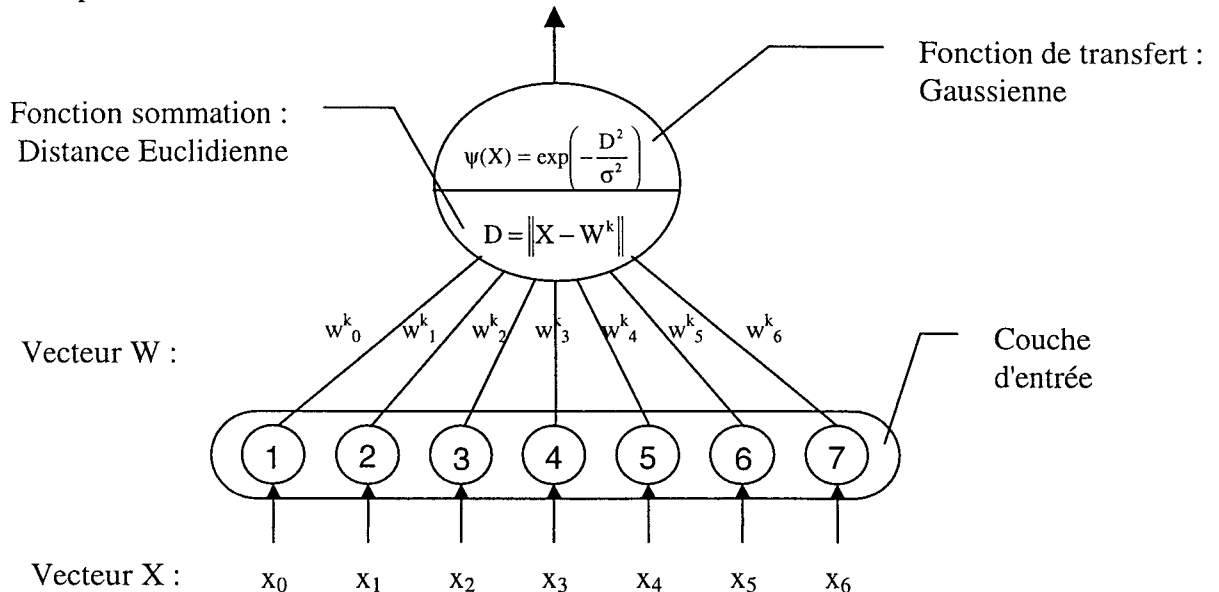


Figure IV.40 : Fonctionnement d'un neurone de type fonction-noyau

La procédure d'apprentissage permet également de déterminer les poids entre la couche P et la couche Q. La couche Q comporte, comme pour la couche de sortie, autant de neurones qu'il y a de classes. L'activité Q_i de chacun des neurones de la couche Q est égale à la somme de l'activité des neurones de la couche précédente définissant un modèle rattaché à la classe définie par le neurone (i).

Par exemple, le neurone P_{52} correspondant à un "R" de la police «Courrier » et le neurone P_{126} , correspondant à un "R" de la police OCR-A sont reliés au même neurone Q_{18} (avec 18 l'indice correspondant au caractère R). Suivant une modélisation statistique, la sortie de chaque neurone (i) de la couche Q définit une fonction de densité de probabilité $p(X/C_i)$ telle que X le vecteur d'entrée du classifieur appartienne à la classe C_i . Grâce à cette propriété, on affirme obtenir une modélisation neuronale de classifieur bayésien.

La dernière couche S est rattachée à la précédente suivant une interaction du type « Winner Take All ». Cette couche est donc constituée d'un seul neurone actif correspondant au neurone le plus actif de la couche Q.

Etant donné le cadre de notre application, nous avons volontairement limité le nombre de classes à 36 neurones. En effet, l'essentiel des caractères rencontrés dans les applications industrielles sont des caractères en capitales et/ou numériques. Ce choix impose un nombre de 36 neurones pour la couche de sortie S et également pour la couche Q. Ainsi, les neurones S_1 à S_{26} représentent les lettres de A à Z, les neurones S_{27} à S_{36} représentent les chiffres de 0 à 9.

La figure IV.41 représente un exemple d'architecture utilisée pour les tests sous l'environnement Nworks [NEU96].

➤ La couche M, constituée de 529 neurones permet de recueillir l'imagette du caractère.

- Les neurones de la couche cachée P permettent de comparer le modèle avec ceux appris. Ainsi, un neurone ayant une activité importante correspond à un modèle appris proche du J.
- La sortie des neurones de la couche Q montre que la plus forte densité de probabilité correspond à la lettre J, le neurone "J" a une activité importante.
- La couche de sortie S donne le résultat de la classification : seul le neurone "J" est actif.

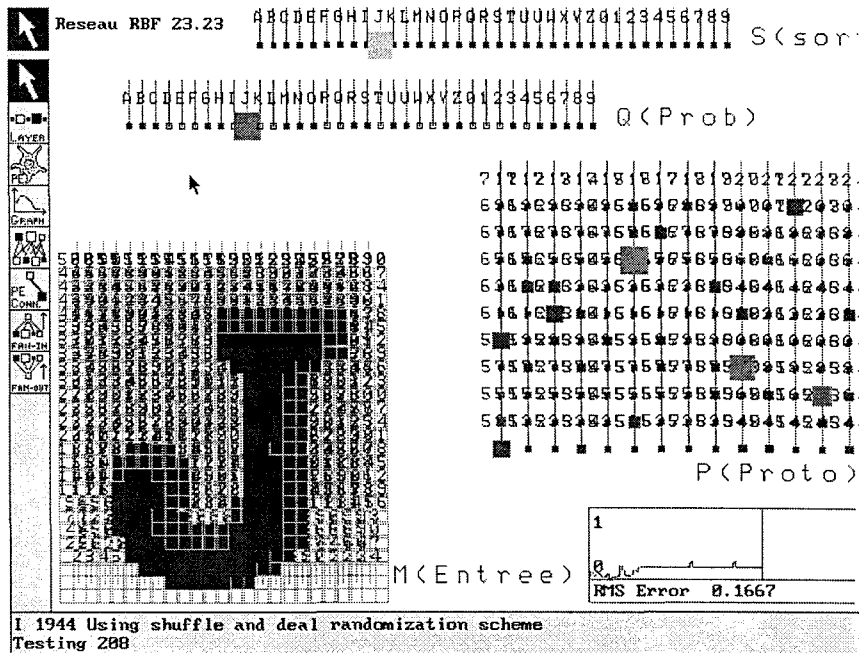


Figure IV.41 : Exemple d'architecture complète pour une couche d'entrée 23*23

Concernant la taille de la couche d'entrée, celle-ci doit être fixée en respectant un compromis entre la qualité de la reconnaissance et la taille du réseau. En effet, chaque neurone de cette couche définit une composante du vecteur à classifier. Il est nécessaire de choisir le vecteur le plus petit possible de manière à définir l'architecture de réseau la plus réduite possible pour obtenir des temps de traitement courts. Cependant, il s'agit de choisir une taille minimale permettant de conserver suffisamment d'informations pour distinguer correctement toutes les classes.

IV.3.1.1 Détermination de la taille de la couche d'entrée

Cette étude a pour but de déterminer la taille optimale de la couche d'entrée du réseau classifieur. Grâce à l'utilisation du premier réseau, le modèle présenté à l'entrée du classifieur est invariant en taille. Le problème de classification ne concerne donc que la distinction des différentes classes pour un vecteur d'entrée de dimension fixée. Cette dimension influe sur la qualité de la reconnaissance. Si la résolution du modèle présenté à l'entrée est faible, le classifieur n'a pas assez d'informations pour distinguer correctement les différentes classes. A l'inverse, si la résolution est importante, le classifieur peut définir très précisément les différentes classes mais nécessitera un temps de traitement important. C'est à partir de l'étude des performances de quinze différents réseaux dont la taille d'entrée varie de 7x7 à 35x35, que nous allons déterminer le meilleur compromis.

La couche d'entrée du réseau classifieur étant la couche de sortie du premier réseau, nous avons pu facilement construire à partir de deux bases initiales (l'une d'apprentissage, l'autre de test) les bases d'apprentissage et de test correspondant à 15 réseaux étudiés. Pour cela, nous avons fait varier la taille du modèle obtenu à la sortie de la couche T (le paramètre Dim dans l'équation (IV.16)) de 7 à 35 avec un pas de 2. On a ainsi construit les 15 bases qui n'ont comme différence que la résolution des modèles qui les composent.

Afin de limiter les sources de mauvais fonctionnement du classifieur et d'en évaluer le plus correctement possible les performances, les paramètres σ et β de la partie vision du premier réseau ont été fixés manuellement en choisissant les valeurs les plus adaptées en fonction des images présentées. La partie effectuant l'invariance en orientation a également été inutilisée. Pour cette première étude, les modèles ne sont donc pas normalisés en orientation.

Les modèles qui ont constitué les deux bases initiales sont ceux qui ont permis l'évaluation des réseaux classifieurs dans la deuxième partie. Ont été également ajoutés des modèles plus spécifiques au marquage industriel (Annexe B).

L'évaluation des performances des quinze réseaux a consisté à étudier lors de la phase d'apprentissage, l'évolution de l'erreur sur la base d'apprentissage et de l'erreur sur la base de test en fonction du pas d'apprentissage afin de comparer les performances en terme de qualité de reconnaissance. L'erreur correspond à un calcul de distance entre la sortie réelle des neurones de la couche Q et la sortie désirée (tous les neurones à 0 sauf le neurone correspondant à la classe du modèle présenté).

La figure IV.42 représente l'évolution de l'erreur sur la base d'apprentissage en fonction du pas d'apprentissage et la figure IV.43 représente l'évolution de l'erreur sur la base de test en fonction du pas d'apprentissage (voir pages suivantes). Bien que les erreurs continuent à décroître, nous avons volontairement limité l'apprentissage à 100000 pas. En effet, dès que l'erreur approche le niveau de 0.4, les erreurs de classification des caractères n'évoluent quasiment plus. C'est à dire, bien que la sortie réelle du neurone correspondant à la classe du modèle ne soit pas à 1, il est le plus actif de la couche Q et donc désigne la bonne classe. Ce niveau d'erreur relativement élevé est dû à un calcul d'erreur quadratique sur un vecteur de taille importante (les 36 neurones de sortie du réseau).

L'analyse de la figure IV.42 montre que l'erreur est évidemment liée à la taille du réseau. Plus la taille de la couche d'entrée augmente, plus pour un pas donné, le taux d'erreur baisse.

Les mesures faites sur la base d'apprentissage montrent que les performances des différents réseaux sont comparables exceptée la plus faible dimension 7x7 où la résolution est trop faible pour pouvoir reconnaître parfaitement tous les modèles de la base. Les courbes concernant les réseaux 19x19 à 35x35 sont les plus rapprochées (écart de 2 % dans la zone 90000 à 100000).

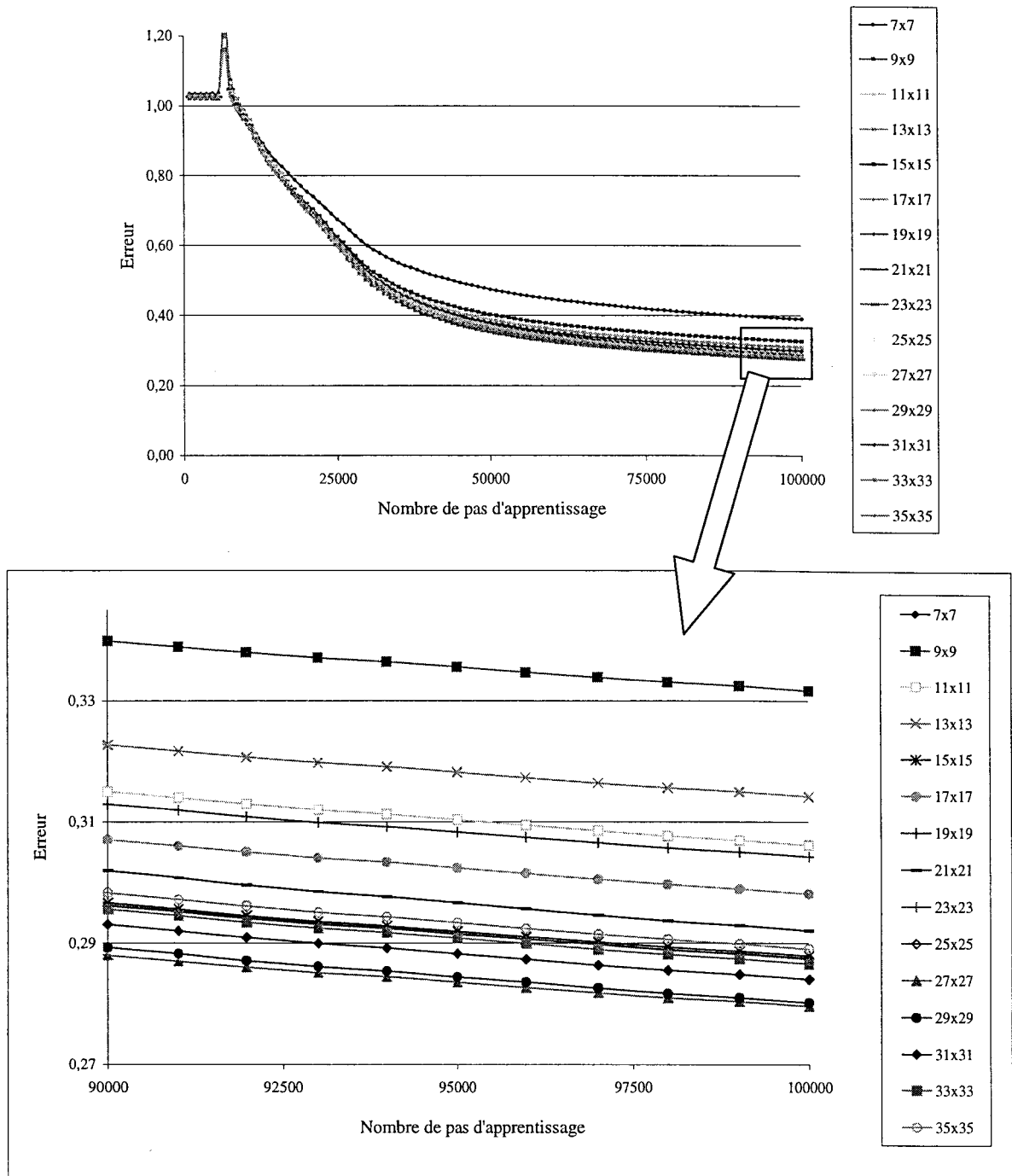


Figure IV.42 : Evolution de l'erreur d'apprentissage en fonction du nombre de pas d'apprentissage pour différentes tailles de couches d'entrée

Les mesures faites sur la base de test sont plus intéressantes et reflètent mieux les performances du classifieur. Contrairement à l'étude sur la base d'apprentissage, on met en évidence (figure IV.43) l'influence de la taille sur le taux de reconnaissance en fonction du nombre de pas d'apprentissage. Ainsi, on remarque qu'il faut présenter près de 100000 modèles pour que le réseau 9x9 ait un taux d'erreur de 0,4 sur les modèles de la base de test. A l'inverse, le réseau 29x29 atteint ce taux pour 52000 pas d'apprentissage.

La figure IV.43 (Zoom) permet également de mettre en évidence que les réseaux 27x27 et 29x29 sont les deux tailles optimales pour un nombre de pas d'apprentissage raisonnable (80000).

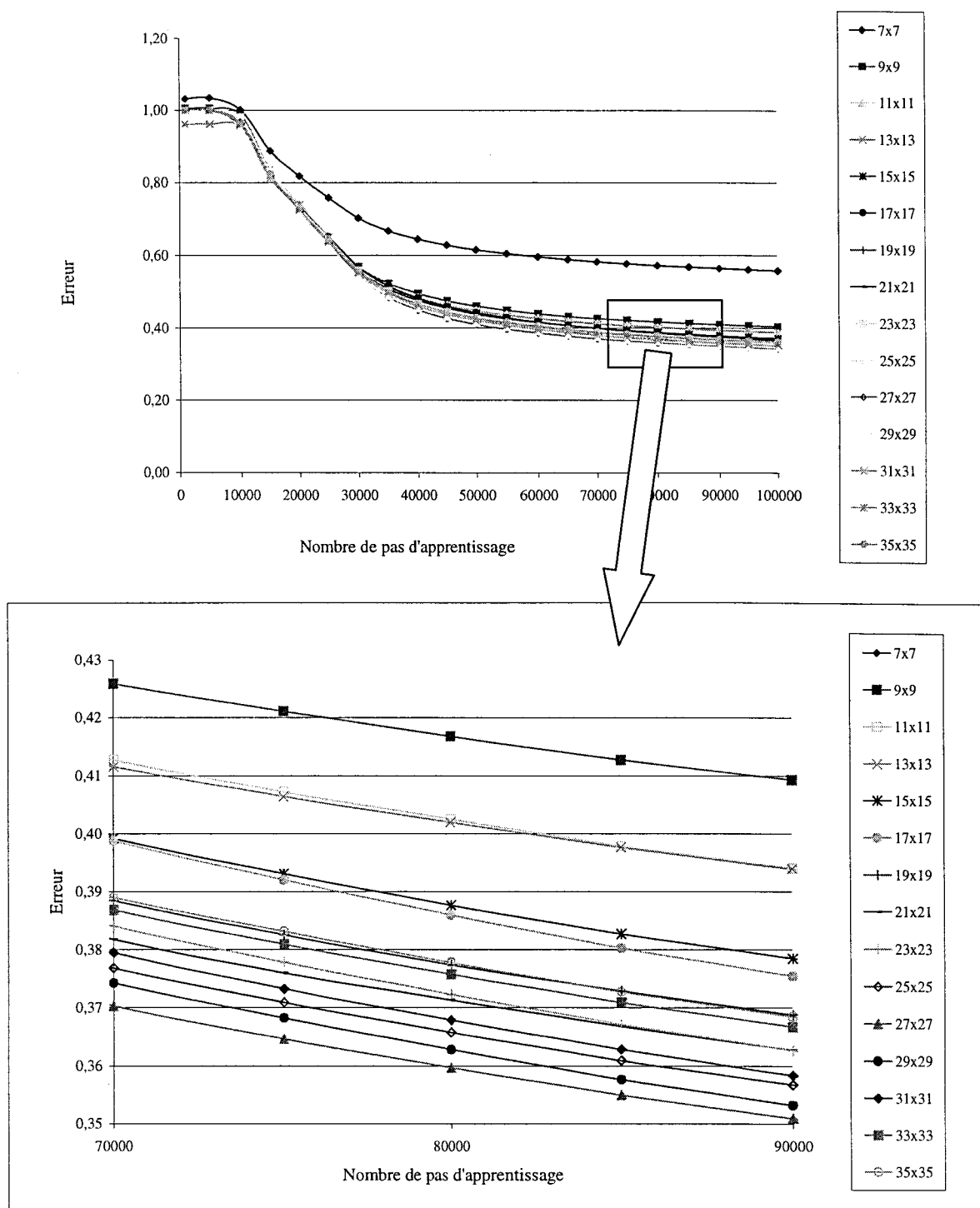


Figure IV.43 : Evolution de l'erreur de test en fonction du nombre de pas d'apprentissage pour différentes tailles de couches d'entrée

Les courbes précédentes ont permis de caractériser les différents réseaux en terme de qualité de reconnaissance et d'efficacité de l'apprentissage en fonction du nombre de pas d'apprentissage.

Une nouvelle représentation des résultats est nécessaire afin de caractériser l'influence de la taille de la couche d'entrée sur le temps de traitement du réseau.

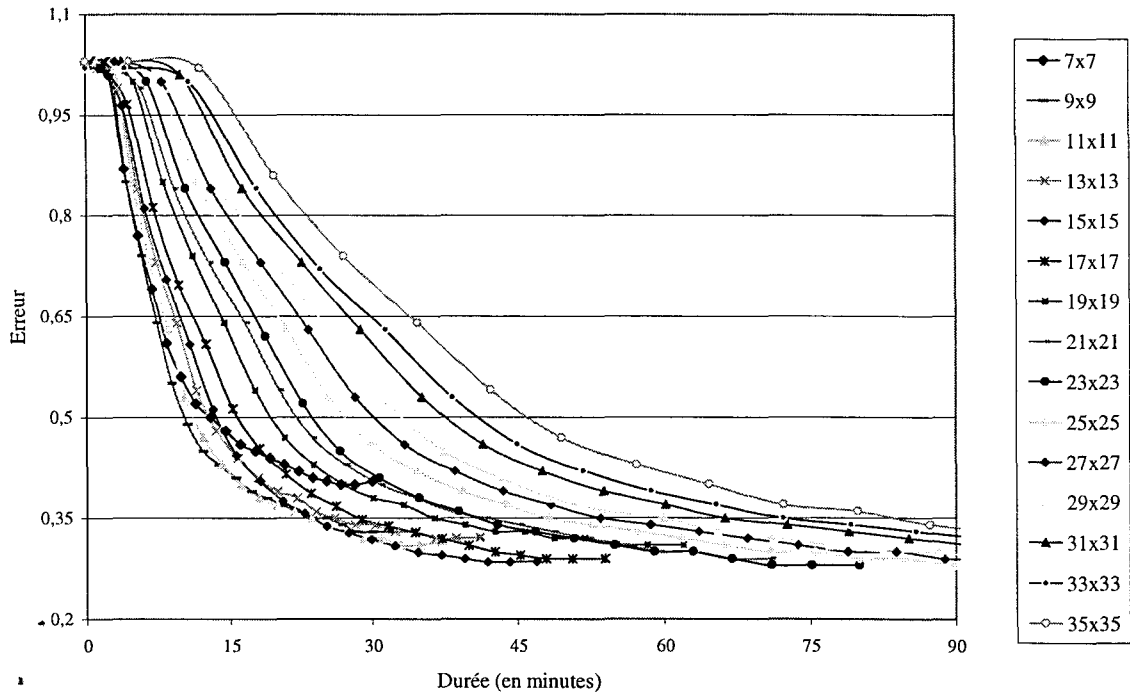


Figure IV.44 : Evolution de l'erreur de test en fonction de la durée d'apprentissage pour différentes tailles de couches d'entrée

La figure IV.44 caractérise l'évolution de l'erreur sur la base de test en fonction de la durée de l'apprentissage. Nous avons volontairement limité l'étude à 90 minutes. Les résultats présentés mettent en évidence l'influence de la taille sur la durée d'apprentissage. Ainsi, le réseau 35x35 nécessite, pour obtenir une erreur valant 0,4, une durée d'apprentissage près de 3,75 fois plus importante que le réseau 9x9 (64,72 min contre 17,13 min).

En prenant pour durée de référence, la durée d'apprentissage du réseau 23x23 nécessaire pour obtenir une erreur valant 0,4 et pour erreur de référence, l'erreur du réseau 23x23 à 60 minutes, on peut construire le tableau IV.6.

Taille de la couche d'entrée	15	19	23	27	35
Durée d'apprentissage pour obtenir un taux d'erreur de 0.4	0,53	0,84	1 (ref.)	1,44	2
Erreur de test pour 60 minutes d'apprentissage	1,13	1	1 (ref.)	1,3	1,15

Tableau IV.6 : Influence de la taille des modèles invariants

Le fait de déterminer la durée d'apprentissage pour une erreur de test donnée donne une indication du temps de traitement du réseau en phase de fonctionnement. En effet, en dehors des cas limites (< 11 et >29), le taux d'erreur n'est fonction que du nombre de pas d'apprentissage. Ainsi, le nombre d'apprentissage est identique, la durée n'est fonction que temps de traitement du réseau.

En fonction du compromis qualité de classification et temps de traitement, nous avons fixé à 23x23 la taille de la couche d'entrée du réseau classifieur.

Outre le choix de la taille de la couche d'entrée, le choix de la taille de la base d'apprentissage doit être judicieux. En effet, la taille de la seconde couche du réseau est directement liée à celle-ci. Ainsi, la base d'apprentissage doit être suffisante pour contenir un nombre de modèles garantissant un taux de reconnaissance correct mais sans pour autant augmenter de façon trop importante la taille de la couche P.

Nous avons pu voir que c'est l'invariance en taille qui va permettre d'adapter la taille du caractère à celle de la couche d'entrée du classifieur (fixée à 23x23). Cependant, cette normalisation en taille et l'utilisation des polices de caractères industriels oblige la modification de la structure traditionnelle du réseau RBF.

IV.3.2 Modifications apportées au réseau RBF

IV.3.2.1 Introduction de la notion de Forme

La variété des marquages rencontrée oblige à prendre en compte des polices particulières telles que les polices à points du jet d'encre ou d'estampage (figure IV.45)

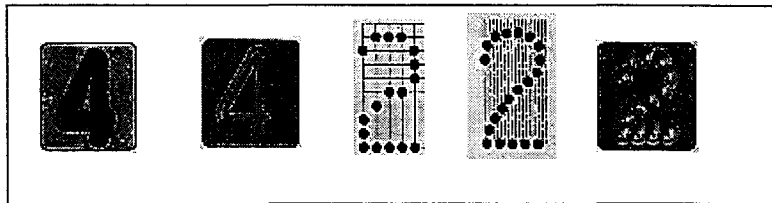


Figure IV.45 : Exemples de caractères spécifiques au marquage industriel

Les problèmes causés par ces polices en discontinuité sont semblables à ceux rencontrés à la sortie du premier réseau lorsque la normalisation en taille va créer un modèle invariant discontinu. En effet, dans les rares cas où la taille désirée est fortement supérieure à la taille réelle du caractère, la normalisation en taille n'interpole pas la totalité du caractère (figure IV.46).




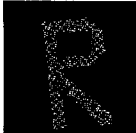
			
Image 21*21	Image 41*41	Image 100*100	Image 150*150
Caractère initial	Modèle invariant	Caractère initial	Modèle invariant

Figure IV.46 : Exemples de modèles invariants discontinus

Un autre problème est lié au traitement du premier réseau : des caractères apparemment identiques peuvent mener à des modèles invariants légèrement différents (tableau IV.7). Ceci est dû au fait que les pixels ont des coordonnées discrètes. Par exemple, le centre de gravité de la couche X n'est pas exactement le centre du caractère mais le neurone le plus proche. Ainsi d'un modèle à l'autre, le centre peut varier de 4 neurones.

La première solution pour permettre la reconnaissance des problèmes précédemment énoncés est d'augmenter exagérément la base d'apprentissage du classifieur. Cette solution est peu réaliste et remet en cause notre volonté de limiter cette base en incorporant dans le système de reconnaissance un étage permettant d'effectuer une invariance.

La seconde solution consiste à augmenter la taille du champ-recepteur (σ) de la fonction d'activation des neurones de la couche cachée, rappelée ci-dessous.

$$\varphi(X) = \exp\left(-\frac{\|X - C\|^2}{\sigma^2}\right) \quad (IV.17)$$

Cette possibilité a priori suffisante entraîne une chute des performances du classifieur. En effet, le risque d'erreur peut être augmenté. Le tableau suivant donne un contre-exemple à l'augmentation de cette taille.





	Modèle appris	Modèle n°1	Modèle n°2	Modèle 3
Image				
Distance Euclidienne	0	0,06	0,11	0,12
φ pour $\sigma = 0.5$	1 (R)	0,78	0,64	0,61
φ pour $\sigma = 1$	1 (R)	0,94 (R)	0,89	0,88
φ pour $\sigma = 1.05$	1 (R)	0,95 (R)	0,90 (R)	0,89

Tableau IV.7 : Influences de la taille du champ récepteur

R=Reconnu comme la lettre C (limite fixé à $\varphi = 0.9$)

Pour résoudre le problème posé par l'utilisation de polices dites industrielles et les incertitudes du premier réseau, nous avons modifié le réseau classifieur RBF en y incorporant la notion de forme. Ainsi grâce à cette notion, il suffit d'apprendre une unique forme regroupant des modèles très proches. La figure IV.47 montre, d'une manière exagérée, que la forme (représentée en gris sur la figure) regroupe différents modèles invariants proches (représentés en noir sur la figure). La forme apprise, la phase de test consiste à déterminer si le modèle est inclus ou non à la forme



Figure IV.47 : Illustrations de la notion de forme

Outre l'avantage de permettre la reconnaissance de fontes particulières, l'utilisation des formes va également permettre de limiter le nombre de neurones sur la couche cachée en regroupant les exemples d'apprentissage et ainsi améliorer le temps de traitement du réseau.

IV.3.2.2 Modifications apportées à la couche Pattern

Afin d'introduire la notion de forme, il est nécessaire de modifier le réseau RBF en changeant la fonction sommation et son processus d'apprentissage. Les différentes formes sont

mémorisées grâce aux poids des connections entre les neurones de la couche M et ceux de la couche P. Ainsi, il y a autant de neurones sur la couche P que de formes. La fonction d'activité des neurones définissant les fonctions-noyaux de la couche P reste de forme gaussienne, seule l'expression du centre de chaque fonction noyau est modifiée. La forme F_i mémorisée grâce aux poids W_i permet de définir les coordonnées du centre de la fonction X_i . Il suffit de pondérer les coordonnées du vecteur x avec le vecteur poids W_i de manière à exprimer $X_i = W_i \cdot X$.

La fonction de transfert d'un neurone caché (i) de la couche P s'exprime par :

$$\varphi_i = \Omega(\|X - W_i \cdot X\|) = \exp\left(-\frac{\|X - W_i \cdot X\|^2}{\phi_i^2}\right) \quad (\text{IV.18})$$

avec $\|X - W_i \cdot X\|$ la fonction sommation du neurone (i) (IV.19)

IV.3.2.3 Modification de la règle d'apprentissage

La phase d'apprentissage permet de mémoriser les différentes formes en adaptant les poids de connections w_k entre la couche M et la couche P. Ainsi, on adapte la phase d'apprentissage en définissant un critère de proximité $\min 1$ entre un nouveau modèle (A) à apprendre et un neurone définissant un prototype ou forme (F_k) appartenant à la classe du vecteur d'entrée.

La procédure d'apprentissage est décrite à la figure IV.48.

- 1) Initialisation aléatoire des poids w_i pour tous les neurones i de la couche Q
 - 2) Tant Que Présentation d'un modèle d'apprentissage X
 - 3) Choix du neurone k de distance minimale (d_k) entre X et le centre de fonction-noyau
 - 4) si $d_k < \min 1$ et si Classe de la forme $F_k =$ Classe de X
 $w_k^{t+1} = w_k^t + X$; retour en 2)
 - 5) si $d_k < \min 1$
 $w_k^{t+1} = w_k^t + \alpha(w_k^t - X)$; retour en 2)
 - 6) $w_k^{t+1} = X$; retour en 2)
- avec α facteur d'apprentissage

Figure IV.48 : Procédure d'apprentissage prenant en compte la notion de forme

Le fait d'avoir modifié la fonction de transfert des neurones cachés pour y introduire la notion de forme, oblige à modifier par rapport à l'architecture RBF classique les connections entre la couche Q et la couche de sortie décrivant la classe d'appartenance. L'exemple d'un vecteur nul présenté à l'entrée de notre classifieur illustre ce propos. Dans le cas d'un vecteur nul, la nouvelle fonction de sommation (IV.19) est nulle quelle que soit la forme (c'est à dire quelle que soit la valeur des poids). Ainsi, la sortie de tous les neurones de la couche cachée est maximale et vaut 1! Il est donc nécessaire de modifier l'interaction « Winner Take All » entre la couche Q et la couche S par une compétition entre les neurones de la couche Q du type inhibition latérale.

IV.3.2.4 Performances liées à l'amélioration

Afin de déterminer les performances de notre réseau classifieur, nous l'avons comparé à l'architecture RBF classique et au réseau PMC. Bien que la technique développée ici soit initiée pour la reconnaissance de caractères particuliers (figures IV.44 et IV.45), nous avons voulu comparer ses performances sur le problème de classification avec les bases de caractères utilisées dans l'étude de la deuxième partie. Ceci va nous permettre de valider les performances du réseau comme un classifieur plus général.

Nous avons fixé une comparaison en fixant :

- la taille de la couche d'entrée à 19x19.
- le réseau PMC à 1 seule couche cachée (cf §II.3.2)

Cette comparaison porte sur le taux d'erreur TE^{APP} sur la base d'apprentissage pour un nombre de pas d'apprentissage de 40000 et sur le taux d'erreur TE^{TES} sur la base de test pour un même nombre de pas d'apprentissage et est résumée dans le tableau IV.8. Les bases d'apprentissage et de test sont les mêmes que celles utilisées au §IV3.1.1.

Type	Notre réseau	Réseau RBF	Réseau PMC
Nombre de modèles de la base d'apprentissage	1512	1512	1512
Nombre de neurones sur les couches cachées	468	1512	350
Taille des champs récepteurs	0,25	0,5	Néant
TE^{APP}	0,523	0,468	0,278
TE^{TES}	0,557	0,507	0,696
Temps d'apprentissage	1	0,97	5,41
Temps de traitement	1	3,21	0,92

Tableau IV.8 : Comparaisons de performances de notre réseau classifieur

Comme nous l'avons souhaité le principal avantage de notre réseau, outre la capacité à classifier des caractères suivant des polices particulières, est d'obtenir de bonnes performances en classification tout en réduisant la taille de l'architecture. Notre modèle possède ainsi des performances proches du réseau RBF en terme de qualité de reconnaissance mais avec une architecture d'une taille proche du réseau PMC.

Ainsi, cette technique permet d'éliminer les inconvénients du réseau RBF par rapport au PMC, à savoir le trop grand nombre de neurones cachés et le temps de traitement plus long, et ce tout en conservant ses avantages qui sont un taux d'erreur correct, des temps d'apprentissage plus courts, l'absence de coefficients momentum, l'absence de minimum locaux lors de l'apprentissage.

Les tableaux IV.9 et IV.10 (page suivante) détaillent les performances en fonctionnement de notre classifieur obtenu après apprentissage de 100000 pas. Les modèles utilisés pour les tests de fonctionnement sont ceux de la base d'apprentissage et la base de test précédemment utilisées.

	nbe	nbr	Te (%)	Tr (%)
A	2	156	1,26582	98,734
B	3	155	0,66964	98,101
C	9	149	2,00893	94,304
D	6	152	1,33929	96,203
E	4	154	0,89286	97,468
F	6	152	1,33929	96,203
G	7	151	1,5625	95,570
H	3	155	0,66964	98,101
I	4	154	0,89286	97,468
J	5	153	1,11607	96,835
K	2	156	0,44643	98,734
L	2	156	0,44643	98,734
M	4	154	0,89286	97,468
N	3	155	0,66964	98,101
O	5	153	1,11607	96,835
P	5	153	1,11607	96,835
Q	9	149	2,00893	94,304
R	4	154	0,89286	97,468
S	3	155	0,66964	98,101
T	1	157	0,22321	99,367
U	4	154	0,89286	97,468
V	5	153	1,11607	96,835
W	3	155	0,66964	98,101
X	4	154	0,89286	97,468
Y	4	154	0,89286	97,468
Z	3	155	0,66964	98,101
	110	3998	0,976	97,322

Tableau IV.9 : Taux de reconnaissance

Avec Nbe : nombre d'erreurs
 Nbr : nombre de caractères reconnus
 Te : Taux d'erreur en pourcentage
 Tr : Taux de reconnaissance en pourcentage

Le tableau IV.9 illustre les taux de reconnaissance de notre réseau classifieur en fonction des différentes lettres de l'alphabet. Pour chaque classe, 158 modèles différents ont été présentés. Les performances du réseau sont assez élevées, le taux de reconnaissance est de près de 97,5% avec des modèles issus de plus d'une soixantaine de polices différentes. L'analyse de la matrice de confusion (tableau IV.10) montre que le classifieur ne génère que peu d'erreurs de "non-sens" (ex : un modèle de lettre X est reconnu comme un A). Ces erreurs peuvent être interprétées comme humaines et essentiellement dues à la représentation rétinienne des caractères. L'utilisation d'un post-traitement permettra très facilement de corriger ce type d'erreurs.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	nbe
A	-	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2
B	0	-	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	3
C	0	0	-	0	0	0	2	0	0	0	0	0	0	0	6	0	1	0	0	0	0	0	0	0	0	0	9
D	0	0	1	-	0	0	1	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0	0	6
E	0	0	0	0	-	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
F	0	0	0	0	3	-	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	6
G	0	0	0	1	0	0	-	0	0	0	0	0	0	1	2	0	3	0	0	0	0	0	0	0	0	0	7
H	2	0	0	0	0	0	0	-	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3
I	0	0	0	0	0	0	0	0	-	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4
J	0	0	0	0	0	0	0	0	3	-	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	5
K	0	0	0	0	0	0	0	0	1	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
L	0	0	0	0	0	0	0	0	1	0	0	-	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2
M	0	0	0	0	0	0	0	1	0	0	0	0	-	2	0	0	0	0	0	0	0	0	0	1	0	0	4
N	0	0	0	0	0	0	0	1	0	0	0	0	1	-	0	0	0	0	0	0	0	0	0	1	0	0	3
O	0	0	1	2	0	0	0	0	0	0	0	0	0	0	-	0	2	0	0	0	0	0	0	0	0	0	5
P	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-	2	2	0	0	0	0	0	0	0	0	5
Q	0	0	0	0	0	0	2	0	0	0	0	0	0	0	6	0	-	0	0	0	1	0	0	0	0	0	9
R	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	-	0	0	0	0	0	0	0	0	0	4
S	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	3
T	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	1
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	4	0	0	0	0	0	4
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	-	0	0	1	0	0	5
W	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0	-	0	0	0	3
X	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-	2	0	4
Y	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	2	0	0	-	0	4
Z	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	3
nbe	4	2	2	4	4	4	7	3	9	3	2	2	1	4	17	3	10	5	1	0	9	6	3	1	3	1	110

Tableau IV.10 : Matrice de confusion

Etant spécialement conçu pour cela, notre réseau doit également permettre d'obtenir les meilleurs taux de reconnaissance pour des caractères de type industriel, cette étude sera réalisée dans la dernière partie.

Le dernier élément de notre système est la boucle de rétroaction. Cette boucle va permettre au premier réseau de modifier l'opération de vision si le classifieur n'a pu reconnaître la classe du caractère.

IV.4 La phase de retour : la coopération

La qualité de reconnaissance du système est principalement liée à l'opération de vision. Le développement d'une coopération entre les deux réseaux réalisant l'un, l'extraction de modèle invariant; l'autre, la classification permet d'améliorer l'opération de vision. Le résultat de l'étape de coopération agit directement sur les opérations de lissage, déterminante pour la qualité de la binarisation. Grâce à la définition des poids de lissage (figures IV.18 et IV.19), le seul paramètre essentiel à modifier est la taille de la zone d'influence déterminant les poids de lissage (σ). La modification à apporter est introduite par une boucle qui permet un retour des informations de l'étage de reconnaissance vers l'étage vision. Cette boucle est constituée de trois neurones (figure IV.49). Le neurone η_Q permet de traiter l'information de classification à la sortie de la couche Q. Le neurone η_M récupère l'information présente à l'entrée du classifieur. Enfin, le neurone η_S utilise l'information de ces deux neurones pour assurer l'adaptation de la partie vision.

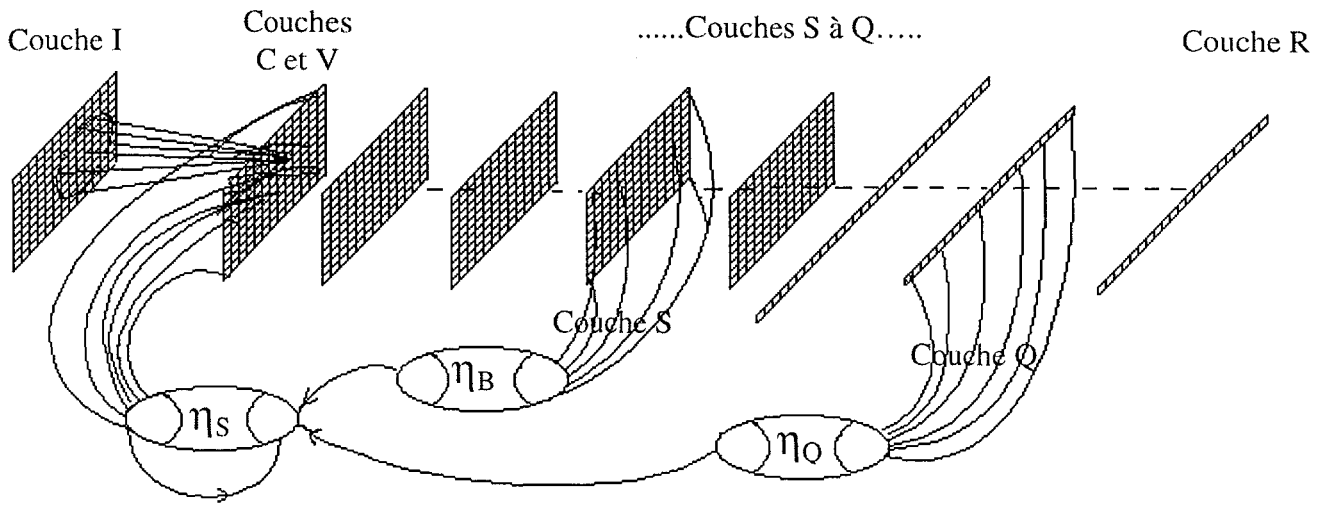


Figure IV.49 : Structure de la boucle de retour

La sortie du neurone η_S est reliée à tous les neurones de la couche C et de la couche V. Son niveau d'activité remplace la valeur σ initialement fixée. Ainsi, les équations des poids de lissage décrits en (IV.2 et IV.4) deviennent :

$$w_{pqij}^c = \exp\left(-\left(\frac{\sqrt{(p-i)^2 + (q-j)^2}}{\eta_S}\right)^e\right) \quad w_{pqij}^v = \exp\left(-\left(\left|\frac{\sqrt{(p-i)^2 + (q-j)^2}}{\eta_S} - 1\right|\right)^e\right) \quad (\text{IV.20})$$

L'activité du neurone (η_S) est déterminée à partir des deux autres neurones qui lui sont interconnectés. Le premier neurone (η_Q) permet d'estimer à partir de la couche Q s'il est nécessaire de modifier le traitement de la partie vision. Cette estimation est déterminée en fonction des taux de reconnaissance obtenus par la couche Q. Le neurone (η_Q) est un neurone à sortie réelle. La sortie est maximale (valant 1) si tous les neurones de la couche Q ont la même activité (aucun modèle n'a pu être mis en évidence). La sortie est nulle si un modèle est reconnu.

L'activité du neurone (η_Q) s'exprime par :

$$A_{\eta_Q} = \exp\left(-\left(\sum_{k=1}^N \Psi(Q_k - \bar{Q})\right)^2\right) \quad \text{avec} \quad \bar{Q} = \frac{1}{N} \sum_{k=1}^N Q_k \quad (\text{IV.21})$$

avec N : nombre de neurones sur la couche Q et ψ décrit en (IV.16)

Le second neurone η_B détermine le sens de l'évolution de σ en fonction de l'activité présente à la sortie de la couche S. La sortie de ce neurone est réelle. Elle vaut 1 lorsque l'activité neuronale de la couche S est nulle et vaut -1 lorsqu'elle est maximale. L'activité nominale est déterminée en fonction de la taille du caractère déterminée grâce à la couche C'.

L'activité du neurone (η_B) s'exprime par :

$$A_{\eta_B} = \tanh\left[\sum_r C'_r \cdot r^2 \cdot v - \sum_k S_k\right] \quad (\text{IV.22})$$

avec v coefficient correcteur définissant le rapport entre la surface du caractère et sa taille.

L'activité du dernier neurone (η_S) est déterminée à partir du résultat de la sortie du neurone (η_Q), du neurone (η_B) et de sa sortie à l'état précédent (η_S)^{t-1}.

$$A_{\eta_S}^{t+1} = A_{\eta_S}^t + \Delta A_{\eta_S}^{t+1} \quad (IV.23)$$

$$\text{avec } \Delta A_{\eta_S}^{t+1} = \gamma \cdot A_{\eta_B} \cdot A_{\eta_Q} + \delta \cdot \Delta A_{\eta_S}^t \quad (IV.24)$$

Le coefficient γ de l'équation IV.24 est le coefficient correcteur définissant le pas d'adaptation. Il est aussi nécessaire d'introduire un terme tenant compte de la variation à l'instant précédent coefficientée par δ ce qui permet de conserver une dynamique de l'adaptation.

Ainsi, en connectant la sortie du neurone η_S à chaque neurone de la couche C et de la couche V, la taille du voisinage de lissage est facilement modifiée en remplaçant la valeur σ par l'activité $S_{\eta_S i}$ dans les équations définissant le calcul des poids de lissage.

L'adaptation du système de reconnaissance s'effectue à l'aide de plusieurs rebouclages qui permettent au système de converger vers la solution. Si le taux de reconnaissance diverge lors de plusieurs boucles, le modèle est alors déclaré non reconnu. La principale difficulté de l'étape de coopération est d'effectuer une adaptation rapide. Ce sont donc les paramètres définis lors de cette étape qui doivent permettre une convergence rapide. La rapidité de l'adaptation est liée à l'influence du facteur γ tandis que le paramètre δ permet d'augmenter les chances de convergence.

Le tableau suivant met en évidence l'évolution de l'adaptation pour différentes valeurs des paramètres γ et δ . Dans ces exemples, nous partons de σ valant 7 pour tendre vers 2.76 (solution optimale).

Nbre d' itérations	a) $\gamma=1 ; \delta=0$	b) $\gamma=1 ; \delta=1$	c) $\gamma=1 ; \delta=2$	d) $\gamma=1 ; \delta=0.5$	e) $\gamma=2 ; \delta=0$	f) $\gamma=1 ; \delta=0.5$	g) $\gamma=3 ; \delta=0.2$
0	7	7	7	7	7	7	7
1	6,37	6,37	6,37	6,37	5,75	5,75	5
2	5,78	5,15	4,53	5,47	4,67	4,05	3
3	5,25	3,43	0,41	4,48	3,7	2,45	2,5
4	4,74	1,56	0	3,5	2,97	2,53	3,2
5	4,27	1	0	2,65	2,68	3,36	2,6
6	3,9	4	0	2,37	2,4	3,46	2,9
7	3,6	1,5	0	2,39	2,5	2,7	2,8
8	3,4	3,7	0	2,55	2,7	2,5	2,75
9	3,26	2	0	2,75	2,9	2,65	2,79
10	3,2	3,2	0	2,6	2,68	3,1	2,76
11	3,15	2,7	0	2,59	2,6	2,9	2,77
12	3,12	3	0	2,8	2,5	2,7	2,76
13	3,1	2,9	0	2,7	2,6	2,8	2,765

Tableau IV.11 : Influences des paramètres γ et δ sur l'évolution de σ

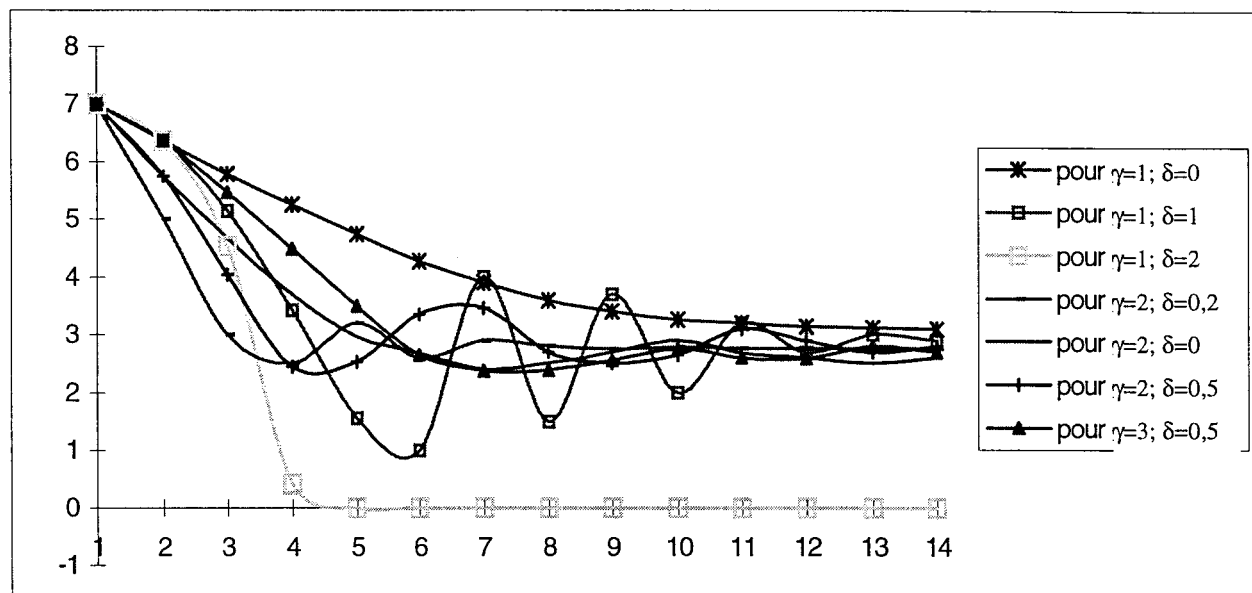


Figure IV.50 : Evolution de σ en fonction du nombre d'itérations

Le tableau IV.11 et la figure IV.50 montrent l'influence de gamma et delta sur la qualité de la convergence. Dans le cas c) une convergence trop brutale rend impossible l'adaptation du voisinage σ . En effet, pour pouvoir reconnaître le modèle, il faut utiliser une valeur du voisinage comprise entre 2 et 3. De même, dans le cas b) des oscillations trop importantes autour de ces valeurs, rendent impossible la reconnaissance. A l'inverse dans le cas a) la convergence est trop lente ce qui implique un grand nombre d'itérations pour reconnaître le modèle. Suite à cette étude, nous avons fixé $\gamma=2$ et $\delta=0,2$.

Conclusion de la quatrième partie

En choisissant comme thème, la reconnaissance de caractères en milieu industriel, nous avons cherché à développer un nouveau système en proposant des concepts originaux permettant de résoudre certains problèmes posés par ce type d'application.

Basé sur la technique classique d'associer un étage d'extraction de caractéristiques et un étage classifieur, nous avons proposé un système basé sur l'utilisation de deux réseaux de neurones mais également sur un bouclage entre ces deux réseaux qui va permettre d'améliorer les capacités de reconnaissance du système.

Pour construire notre système, nous avons développé deux nouvelles architectures neuronales en y incluant de nouveaux concepts permettant d'améliorer la reconnaissance de caractères pour l'environnement industriel.

La première architecture neuronale permet l'extraction d'un modèle binarisé invariant à partir d'une image de caractère.

Un étage de vision a été développé pour permettre la binarisation quels que soient le support et le marquage et les conditions lumineuses rencontrés. Ainsi, nous avons modélisé la notion de filtrage centre/voisinage sous forme des couches neuronales.

Etant donné la nature des articles à identifier et le contexte de la prise d'image, nous avons également introduit, dans l'architecture du premier réseau, des notions d'invariance en

position, en taille et en orientation. Dans le but de réalisation, l'ensemble de notre système utilise une description neuronale. Pour cela, nous avons été amenés à modéliser une nouvelle architecture permettant de réaliser ces invariances.

De nouvelles techniques ont été également développées au niveau du second réseau, le classifieur. En partant d'une architecture classique, le réseau RBF, nous avons adapté la fonction de transfert des neurones de la couche cachée afin d'introduire la notion de forme. Cette notion nous a permis de simplifier l'architecture en limitant la taille de la couche cachée mais également d'améliorer ses performances de classification, en permettant la reconnaissance de modèles particuliers.

Enfin, le dernier concept développé est une boucle de rétroaction entre le résultat du classifieur et l'étage de vision. Elle permet d'assurer l'adaptabilité automatique du système au type de marquage et au type de support. Le paramètre σ déterminant dans la qualité de la binarisation de l'image initiale, est directement modifié par cette boucle.

Suite à la description des concepts mis en place pour la réalisation, nous consacrerons la dernière partie de notre manuscrit à la présentation des performances de notre système appliqué à un ensemble d'images de caractères réelles.

Cinquième Partie :

Validations expérimentales

Introduction

Cette cinquième et dernière partie présente les résultats expérimentaux de notre système. Il s'agit de mettre en évidence ses avantages et d'évaluer l'amélioration des performances qu'il apporte pour la reconnaissance de caractères industriels.

Le premier paragraphe est consacré à l'analyse des caractéristiques du système. Nous montrons en premier lieu les performances de la partie vision et de la partie invariance du premier réseau en appliquant un ensemble d'exemples de caractères qui présentent des types de marquage et des types de support variés. Nous étudions ensuite le fonctionnement du second réseau qui assure la classification. Nous vérifions l'obtention d'une reconnaissance de caractères multiforme en tenant compte des spécificités des fontes utilisées en marquage industriel. Enfin, nous démontrons les capacités d'autoadaptation de notre système grâce à sa boucle de rétroaction.

Dans le deuxième paragraphe nous présentons les performances globales de notre système en le comparant avec des systèmes commerciaux existants.

Enfin, nous examinons, dans le dernier paragraphe, les possibilités d'implémentation de notre architecture. Nous tenterons d'évaluer les possibilités et les fonctions de coûts de réalisation faites à partir d'outils actuellement disponibles.

V.1 Performances du système

V.1.1 Conditions d'expérimentation

La constitution d'une base de caractères de type industriel a été la première difficulté rencontrée pour notre étude. En effet, contrairement à l'écriture manuscrite pour laquelle de nombreux serveurs Internet donnent accès à des bases standardisées (généralement issues des postes ou des banques), seule une base standardisée de caractères imprimés, la base NIST Special Database 8 : Machine print database [NIST98], est disponible. Celle-ci aurait pu servir à analyser les performances de notre système en comparaison avec d'autres. Cependant, cette base s'avère peu adaptée car elle ne contient que des caractères ayant des fontes classiques : Courier, Helvetica, New Century Schoolbook, Optima, Palatino, et Times Roman et d'autre part, elle reste trop onéreuse (Special Database 8 - \$1 895.00)

Pour permettre la constitution de notre base d'essais, nous nous sommes adressés aux sociétés proposant des systèmes de reconnaissance d'OCR en milieu industriel (Edixia, Allen Bradley, COGNEX) et vers les sociétés de marquage industriel (SIC, MecaGrav, Image,...). Certaines images de test (applications d'OCR spécifiques) ont été également récupérées sur différents sites Internet.

Nous reproduisons, en annexe C et D, les éléments fondamentaux de cette base d'essais.

V.1.2 Performances du premier réseau

Pour la phase expérimentale du premier réseau, il est intéressant d'observer son fonctionnement au niveau de l'extraction du modèle binarisé grâce à la partie vision mais également au niveau de l'invariance grâce à l'étape de normalisation.

L'étude de la partie vision a pour but de mettre en évidence ses performances en comparaison avec la technique du seuillage global. Le tableau V.1 montre à partir de différents cas (marquage, support, légères variations lumineuses) les résultats obtenus avec un seuillage global et avec l'application de la technique de filtrage centre/voisinage. Les résultats présentés ici, sont donnés avec la valeur de seuillage (S) et la valeur de la taille des filtres (σ) permettant d'obtenir un résultat optimal.







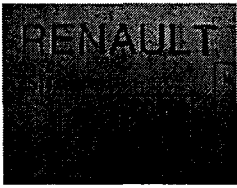
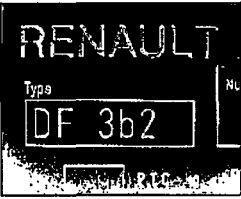
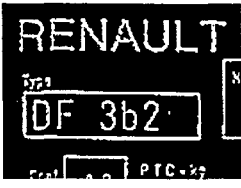



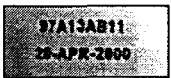


	Image initiale	Image binarisée	Image à l'issue de la partie vision
(A)			
		S = 37	$\sigma = 4$
(B)			
		S = 59	$\sigma = 3$
(C)			
		S = 100	$\sigma = 3$
(D)			
		S = 88	$\sigma = 2$
(E)			
		S = 110	$\sigma = 2$

Tableau V.1 : Résultats de la partie vision

Ce premier tableau de résultats met en évidence les capacités de l'étage constitué par les couches C,V et S, à extraire des modèles binarisés de caractères pour des images présentant des variations de support et de marquage. Ainsi, pour des caractères plus clairs ou plus sombres que le fond, l'étage de vision (la sortie de la couche S) donne des résultats de binarisation corrects (exemple (A) et (B)). Dans le cas du seuillage global, il a été nécessaire de modifier le type de seuillage (noir→blanc ou blanc→noir).

Les exemples (C) à (E), où le fond n'a pas un niveau de gris uniforme, montrent l'avantage de la technique du filtrage. En effet, dans le cas de faibles variations de luminosité extérieure, l'étage de vision est capable d'effectuer correctement la binarisation sur toute l'image.

L'utilisation des filtres de lissage centre/voisinage offre également l'avantage d'une insensibilité aux légères dégradations. Les exemples (F) à (I) représentés dans le tableau V.2 montre les capacités de l'étage de vision à supprimer les effets d'une légère dégradation du marquage. Cette propriété est essentiellement obtenue grâce aux opérations de lissage. Dans certains cas, comme l'exemple (I), cette propriété permet de simplifier la reconnaissance des caractères présentant des polices de marquage particulières. En effet, la difficulté d'interpréter un caractère comme une suite de points (cas du microburinage) est supprimée par lissage de l'image.




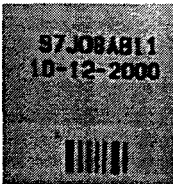






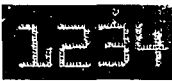

	Image initiale	Image binarisée	Image à l'issue de la partie vision
(F)			
		S = 78	$\sigma = 3$
(G)			
		S = 100	$\sigma = 2$
(H)			
		S = 158	$\sigma = 4$
(I)			
		S = 93	$\sigma = 7$

Tableau V.2 : Insensibilité aux légères dégradations

Dans les exemples présentés dans les tableaux précédents, nous remarquons que la valeur de σ est liée à la qualité du support (plus le support est bruité, plus σ doit être grand) mais également à la taille du caractère (plus la taille est grande, plus σ est grand et inversement). Dans le cas d'un support bruité ayant des caractères de faible taille, le compromis entre extraction du modèle bruité et insensibilité aux bruits est difficile à réaliser. Les exemples (K),(L) et (M) dans le tableau V.3 montrent les limites de notre système dans le cas d'image bruitée et lorsque les caractères sont de petite taille.

Cependant, dans la majorité des cas, lorsque la taille des caractères est plus importante, l'utilisation de l'étage de vision peut être affranchie de l'utilisation d'un étage de prétraitement. Seuls certains supports extrêmement bruités vont nécessiter des opérations d'analyse plus complexes telle que l'analyse de texture. Quelques cas limites pour notre système sont représentés dans le tableau V.4.






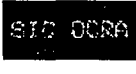


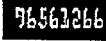

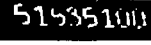
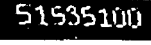
(J)			
		S = 73	$\sigma = 7$
(K)			
		S = 68	$\sigma = 2$
(L)			
		S = 52	$\sigma = 2$
(M)			
		S = 156	$\sigma = 3$

Tableau V.3 : Difficultés liées à la taille des caractères et au bruit de l'image







(N)			
		S = 105	$\sigma = 8$
(O)			
		S = 96	$\sigma = 8$

Tableau V.4 : Cas limites

La deuxième partie de notre validation concerne le rôle de l'invariance dans la reconnaissance de caractères industriels.

L'invariance en position par normalisation suivant le centre de gravité possède, outre l'avantage évident de permettre de recadrer le caractère pour l'opération de normalisation en taille et en orientation, possède l'avantage d'obtenir une représentation plus discriminante pour le classifieur.

Alors que l'invariance en taille est indispensable pour la reconnaissance de caractères afin d'être insensible aux différentes tailles de polices, l'invariance en orientation n'est motivée que par une utilisation du système dans les conditions liées à l'identification industrielle. Plus particulièrement, les articles circulaires (disques LASER, couronnes, roues dentées) utilisent un marquage également circulaire. Les systèmes actuels effectuent la reconnaissance grâce à une opération de déroulé qui "redresse" l'image circulaire en une image rectangulaire. Notre opération de normalisation en orientation permet alors l'économie de ce prétraitement.

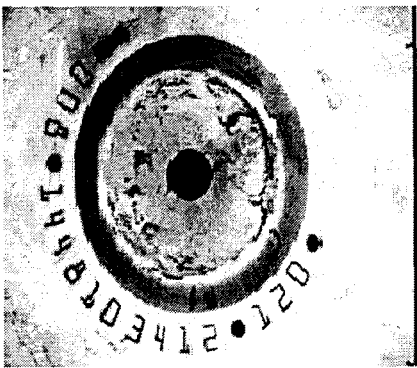
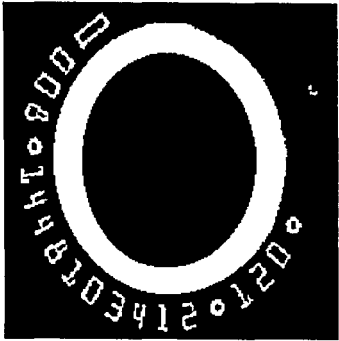

	
Image initiale	Image binarisée
Chaîne normalisée à l'issue de l'étape de normalisation en orientation	

Figure V.1 : Normalisation en orientation

La figure V.1 donne un exemple sur lequel nous avons l'invariance en orientation. Cependant, cette invariance doit être utilisée et analysée avec prudence car la normalisation doit être limitée à une rotation de 180° pour limiter les ambiguïtés de distinction de caractère. A titre d'exemple, pour une rotation de 180°, le modèle représentant un "b" aura une représentation proche de celle du "q".

La normalisation en orientation comme la normalisation en position permet d'améliorer dans certains cas, la discrimination des caractères. En revanche, pour certains autres cas, elle est très pénalisante et nécessite l'utilisation d'un étage de vérification en post-traitement. La figure V.2 montre que le caractère "N" et le caractère "Z" ont un modèle invariant très proche.





B	P	N	Z
			
A) Augmentation de la discrimination		B) Diminution de la discrimination	

Figure V.2 : Influence de la normalisation en orientation sur la classification

L'utilisation de l'invariance en orientation constitue également un inconvénient sur certains modèles de caractères ayant subis un type de dégradation particulière. La figure V.3 illustre deux exemples de représentation ne permettant pas une normalisation correcte. Les modèles invariants obtenus ne correspondent alors à aucun modèle de la base d'apprentissage.





Image initiale	Modèle invariant	Image initiale	Modèle invariant
			
(A)		(B)	

Figure V.3 : Exemples donnant un modèle invariant incorrect

Après avoir montré les capacités du premier réseau à extraire des modèles binarisés et invariants pour un nombre important d'images de caractères, l'expérimentation du second réseau a pour but de déterminer les performances en terme de qualité de reconnaissance

V.1.3 Expérimentation du second réseau

Cette étude expérimentale a pour but de valider le réseau classifieur. Son rôle consiste à reconnaître les caractères quelle que soit la police industrielle utilisée. Cette étude nous permet de valider les capacités du réseau à effectuer une reconnaissance multifontes incluant les polices standard et les polices utilisées dans les systèmes de marquage industriel.

Les modèles constituant la base d'apprentissage du réseau sont des images de caractères imprimés par imprimante Laser sur papier. Les polices employées sont des polices TrueType identiques à celles couramment employées dans les systèmes de marquage industriel. Ceci nous garantit la réalisation d'un ensemble d'apprentissage complet présentant un nombre important de caractères de taille et de fontes variées.

L'apprentissage a été réalisé pour environ 5000 caractères. Nous présentons ici les résultats obtenus par le classifieur sur la base de tests présentée en Annexe C. Cette base regroupe un ensemble varié (2912 modèles) de caractères (lettres capitales) selon différentes polices.

Ces modèles de la base de tests sont présentés au réseau classifieur via l'entrée du premier réseau. Le modèle présenté à l'entrée du second réseau est donc le modèle invariant du caractère initial.

Le tableau V.5 donne les performances du réseau en terme de taux d'erreur (TE), taux de rejet (TJ) et taux de reconnaissance (TR) pour chacune des différentes classes.

Le taux d'erreur de lecture est relativement faible (moins de 2%). Cependant, le taux de rejet (classe non reconnue) important, contribue à diminuer le taux de reconnaissance.

Cinquième Partie : Validations expérimentales

	NbE	RF	TE (%)	TJ (%)	TR (%)
A	9	0	0,000	8,088	91,912
B	12	1	0,893	9,961	89,147
C	13	6	5,357	6,194	88,449
D	16	3	2,679	11,504	85,817
E	16	0	0,000	14,230	85,770
F	12	3	2,679	7,767	89,554
G	13	1	0,893	10,679	88,428
H	13	0	0,000	11,618	88,382
I	11	3	2,679	7,143	90,179
J	10	1	0,893	7,767	91,340
K	13	0	0,000	11,760	88,241
L	7	1	0,893	5,304	93,803
M	9	5	4,464	3,592	91,944
N	7	3	2,679	3,304	94,017
O	19	6	5,357	11,732	82,911
P	10	2	1,786	7,491	90,723
Q	14	2	1,786	10,875	87,339
R	14	1	0,893	11,793	87,314
S	3	0	0,000	2,414	97,586
T	11	0	0,000	9,473	90,527
U	10	3	2,679	6,383	90,939
V	16	3	2,679	11,720	85,602
W	14	1	0,893	11,209	87,898
X	6	0	0,000	5,102	94,898
Y	9	0	0,000	8,039	91,961
Z	10	4	3,571	5,632	90,797
Total et Moyenne	296	49	1,683	8,491	89,826

Tableau V.5 : Performances de la classification

Le tableau V.6 permet d'analyser le faible taux de reconnaissance. Il résume les performances globales du réseau avec comme comparaison les performances obtenues pour des modèles uniquement normalisés en position et en taille.

	Modèle invariant en Position et en taille	Modèle invariant en position, en taille et en orientation
Taux de reconnaissance	96,1 %	89,8 %
Taux de Rejet	2,6 %	8,5%
Taux d'erreur	1,3 %	1,7 %

Tableau V.6 : Taux de reconnaissance et d'erreur

Comme indiqué précédemment, l'invariance en orientation, en dehors de certains cas très spécifiques, est inutile et est une source de dégradation des performances du classifieur. En effet,

Cinquième Partie : Validations expérimentales

sans compter sur l'erreur qui peut être faite au niveau de la normalisation en orientation lorsque le caractère est dégradé (figure V.4), le modèle obtenu peut être très proche d'un modèle d'une autre classe (Cas B figure V.3).

La matrice de confusion, présentée tableau V.7, met en évidence les erreurs faites par le réseau classifieur en indiquant, pour chaque classe et pour 112 modèles présentés, le nombre d'erreurs faites et le résultat de, elle indique les erreurs faites pour chaque classe.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	NbE
A	-	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	3	0	0	1	0	2	0	0	1	9
B	0	-	1	3	1	1	0	2	0	0	0	0	0	0	0	1	0	2	1	0	0	0	0	0	0	0	12
C	0	0	-	0	0	0	4	0	0	0	0	0	0	0	4	0	3	0	0	0	0	1	0	0	0	0	13
D	0	4	2	-	1	0	0	0	0	1	0	0	0	0	7	0	0	0	0	0	1	0	0	0	0	0	16
E	0	1	0	0	-	4	0	2	0	2	2	1	0	0	0	0	0	2	0	0	0	0	0	0	0	1	16
F	0	2	0	0	8	-	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	12
G	0	0	3	1	0	0	-	0	0	0	0	0	0	0	2	0	1	0	0	6	0	0	0	0	0	0	13
H	0	0	0	0	3	2	0	-	2	0	0	0	0	0	0	0	2	0	0	0	0	2	0	2	0	0	13
I	0	0	0	0	0	1	0	2	-	2	0	2	0	0	0	1	0	0	0	3	0	0	0	0	0	0	11
J	0	1	0	0	0	0	0	0	2	-	0	3	0	0	0	0	0	0	0	0	2	0	0	0	0	1	10
K	0	0	0	0	1	0	0	1	1	0	-	0	3	0	0	0	1	4	0	0	0	0	0	2	0	0	13
L	0	0	0	1	1	1	0	0	0	0	1	-	0	1	0	0	0	2	0	0	0	0	0	0	0	0	7
M	0	0	0	0	0	0	0	0	0	1	0	0	-	0	0	0	0	0	0	0	0	0	7	0	0	1	9
N	0	0	0	0	0	1	2	0	0	0	0	0	1	-	0	0	0	0	0	0	0	0	0	0	0	3	7
O	0	0	4	5	0	0	3	0	0	0	0	0	0	0	-	0	4	0	1	0	0	0	2	0	0	0	19
P	1	3	0	0	0	2	0	0	0	0	1	0	0	0	0	-	0	2	0	0	1	0	0	0	0	0	10
Q	0	0	2	2	0	0	1	0	1	0	0	0	0	0	5	2	-	0	0	0	1	0	0	0	0	0	14
R	2	1	0	0	2	0	0	0	0	0	2	0	0	1	0	3	1	-	0	0	0	0	1	1	0	0	14
S	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	-	0	0	0	0	0	0	0	3
T	0	0	0	0	1	1	0	2	1	0	1	1	0	0	0	0	0	0	0	-	0	0	1	2	0	0	11
U	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	-	5	1	0	0	0	10
V	0	0	0	0	0	0	0	0	0	0	1	1	3	2	0	0	0	1	0	0	4	-	2	0	1	0	16
W	0	0	0	0	0	0	0	0	0	0	0	0	6	0	1	0	0	2	0	0	0	4	-	0	0	0	14
X	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	-	2	0	6
Y	0	0	0	0	0	1	0	1	2	0	0	1	0	1	0	0	0	0	0	0	0	0	0	3	-	0	9
Z	0	1	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	1	0	0	1	1	0	1	0	-	10
NbE	5	13	13	13	18	14	11	11	10	7	9	9	15	11	20	9	13	20	2	10	12	13	16	12	3	7	296

Tableau V.7 : Matrice de confusion

Bien que le nombre de modèles (26x112) qui constituent la base de test soit faible, l'analyse de la matrice de confusion permet de caractériser le comportement du classifieur. Après avoir mis en évidence les faiblesses de l'invariance en orientation (5 N reconnus pour Z) (3 A pour R) (3 Z pour N), le comportement du classifieur s'avère cohérent et donne des résultats interprétables. Le fonctionnement du classifieur est proche du fonctionnement humain. Ainsi, il est évident que la représentation rétinienne des caractères rend possible les erreurs de reconnaissance. Un grand nombre d'erreurs est lié aux confusions entre des caractères ambigus (le O, le D, le G).

Dans le cas où le modèle présenté est difficile à reconnaître, le classifieur ne peut, tout comme l'œil humain, garantir le résultat, d'où un taux de rejet important. Cependant, son fonctionnement du type classification par modélisation permet d'approcher le résultat. Ainsi, pour un I présenté à l'entrée, les neurones de sortie les plus actifs représentent des classes proches du résultat (I,l,1,j). Ainsi, pour effectuer le bon choix, un étage de vérification sémantique (en applications industrielles, il s'agit généralement d'une vérification de la lettre clé) suffit pour lui permettre de choisir entre l'une de ces classes tout comme l'homme l'aurait fait.

V.1.4 Expérimentation de la boucle de retour

Le rôle de cette boucle, définie grâce au neurone η_s , est d'assurer l'autoadaptation du système aux caractéristiques lumineuses. Nous savons que seul le paramètre σ est important pour la binarisation (§IV.2.1). Les deux exemples de la figure V.4 montrent l'évolution de la sortie de la couche S en fonction de σ .

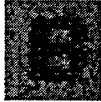









Couche I	Couche S $\sigma=2$	Couche S $\sigma=4$	Couche S $\sigma=6$	Couche S $\sigma=8$
				
				

Figure V.4 : Influence de σ sur la couche S

L'étude de la quatrième partie a montré que la boucle de rétroaction entre la sortie du classifieur et l'étage vision du premier réseau permet d'adapter correctement la valeur de σ .

Les neurones η_s , η_Q , η_B sont utilisés pour modifier cette valeur. A la fin de chaque cycle (vision-invariances-reconnaissance), leur sortie est modifiée en fonction de la qualité de la reconnaissance, en fonction de la taille du caractère et en fonction de l'activité neuronale de S.

Pour expérimenter cette étape, nous avons appliqué à l'entrée du système bouclé une image 120x120 en fixant le paramètre de départ σ à la valeur 3. Le tableau V.8 représente les évolutions des différentes couches lors de plusieurs cycles.





Cycle	σ	Couche B	\bar{q}	η_Q	η_B	$\Delta\eta_s$	η_s
1	3		0,5	0,9	1	0,72	3,72
2	3,72		0,3	0,7	1	0,7	4,6
3	4,6		0,2	0,3	1	0,3	4,9
4	4,9		0,1	0,05	0,6	0,03	4,93

Tableau V.8 : Influence de la rétroaction

Après trois cycles, le système reconnaît le caractère. Seul le neurone correspondant à la classe E est actif sur la couche de sortie du second réseau. Ceci se traduit, en particulier par une valeur de sortie du neurone η_Q faible (0.3). Le dernier cycle montre la convergence rapide vers un état stable. Dans ce cas, le neurone η_Q a une sortie quasi nulle. L'évolution du neurone η_B de

par la nature de la fonction \tanh se résume simplement à une sortie valant +1 ou -1. Ceci suffit à permettre l'adaptation de la valeur du neurone η_S . C'est seulement dans le cas précis où la somme de l'activité S_k tend vers celle estimée par la taille r (terme 1 dans l'équation (IV.22)) que la sortie $S\eta_B$ tend vers 0.

V.1.5 Performances globales

Les performances globales ont été évaluées à partir d'un ensemble d'images de caractères industriels. Il s'agit de mesurer les capacités d'adaptation du système à différents types de marquage et à différents types de support. La figure V.5 donne un exemple d'images appliquées et l'annexe D regroupe une partie des images utilisées.





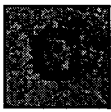



a)	Bois Imprimé		$\sigma=2$	
b)	Plastique Jet d'encre		$\sigma=5$	
c)	Acier Ciselé		$\sigma=7$	
d)	Aluminium Gravé		$\sigma=5$	

Figure V.5 : Exemples de marquage

Bien que l'évaluation n'a pu être effectuée sur une très grande base de données étant donné les difficultés rencontrées lors de la constitution, le tableau V.9 représente les tendances actuellement obtenues par notre système. Nous avons cherché à caractériser le système en comparant le pourcentage de réussite de reconnaissance en fonction du paramètre netteté défini comme l'écart entre les niveaux de gris du caractère et le fond mais également par sa capacité à s'adapter rapidement aux différentes caractéristiques lumineuses d'un exemple à l'autre.

Netteté	Support-marquage	taux de réussite	nombre de cycle d'adaptation
1	Papier-encre	100	1
0.85	Bois-imprimé	99	1
0.75	Résine-encre	99	2
0.6	Acier-gravé	99	2
0.5	Aluminium-gravé	98	2
0.4	Plastique-gravé	97	3
0.3	Verre-gravé	80	3
0.1	Surface luisante	5	4

Tableau V.9 : Performances globales

V.2 Tentative de comparaison avec des systèmes commercialisés

Le système développé n'est pas un système de reconnaissance de caractères complet au sens de la première partie de ce manuscrit. En effet, nous n'avons développé que la partie "reconnaissance" du système. En plus de l'indispensable étage de segmentation, actuellement, la quasi-totalité des systèmes de reconnaissance dispose d'outils de post-traitements permettant de vérifier la cohérence des résultats. Etant donné la nature variée des différents systèmes, la comparaison des performances réelles entre les différents systèmes reste difficile à estimer. N'ayant pu utiliser les systèmes sur les mêmes types d'application et surtout sur une même base de test, nous ne pouvons donner des indications que par rapport aux différentes caractéristiques des systèmes.

Pour pouvoir comparer notre système, nous avons utilisé 3 systèmes d'OCR exploités dans le monde industriel.

- Un automate de vision : la CVIM2 OCR-PAK d'ALLEN BRADLEY
- Un logiciel programmable : la librairie MIL-OCR de MATROX
- Un logiciel configurable : le logiciel Prophétie d'IMAGING TECHNOLOGIES

Le tableau V.10 résume les caractéristiques générales des différents systèmes.

	CVIM2	MIL	Prophety	Notre système
Type	Corrélation	Corrélation normalisée	Réseau de neurones	Réseau de neurones
Invariance	Non	Non	Par apprentissage	Oui
Sensibilité aux bruits	Non	Oui	Oui	Non
Apprentissage	Limité (capacité mémoire)	Limité Temps de traitement	Oui	Oui
Avantages	Temps de traitement	Simplicité d'utilisation	Apprentissage Invariance	Nombre de classes Adaptation lumineuse
Désavantages	Capacité apprentissage	Capacité apprentissage	Sensibilité aux bruits	Réalisation matérielle

Tableau V.10 : Caractéristiques générales

V.3 Discussion autour de l'implémentation matérielle

Il est également très difficile de comparer les performances des systèmes en terme de temps de traitement. Actuellement, les résultats présentés dans la partie expérimentale sont issus d'une simulation du système. Le système présenté bien que modélisé sous forme parallèle a été décrit en langage C et simulé sur un processeur séquentiel. Ceci nous a permis de valider les performances de la reconnaissance afin de garantir le fonctionnement dans le cas d'une réalisation parallèle.

Pour estimer le temps de traitement réel de notre système, nous n'avons pu que comparer notre architecture à des systèmes actuellement réalisés [BLA89]. Les performances réelles ont été simplement estimées sous forme de coût en opérations réalisées par chaque étage.

Le tableau V.11 détermine la durée des opérations effectuées par chaque couche à chaque étage avec :

- NbI : pixels présents sur l'image de départ.
- NbN : pixels sur le modèle normalisé.
- M : coût d'une multiplication (ou division).
- A : coût d'une addition (ou soustraction).
- $\mathcal{C}(X)$: coût fonction d'activation X.

Couches C & V	$NbI(NbI.M+(NbI-1)A)$
Couches S ⁺ & S ⁻	$NbI.(A+\mathcal{C}(\xi))$
Couche G & G'	$NbI(NbI.M+(NbI-1)A)$
Couche P	$NbI.(NbI.M+(NbI-1)A)$
Couche T & T'	$\sqrt{NbI}.(NbI.M+\mathcal{C}(\varphi)+(NbI-1)A)$
Couche O & O'	$180.(NbI.M+(NbI-1)A)$
Couche N	$NbN.(NbI.M+(NbI-1)A)$
Total	$[5NbI+NbN+2.\sqrt{NbI}+360].(NbI.M)+2.NbI.\mathcal{C}(\xi)+2.\sqrt{NbI}.\mathcal{C}(\varphi)+[2NbI+(5NbI+NbN+360+2.\sqrt{NbI}).(NbI-1)].A$

Tableau V.11 : Coût de traitement

Deux réseaux proches des nôtres ont été récemment réalisés par deux équipes de chercheurs. La première équipe [TRO93] a implémenté le neocognitron. La seconde [ESIEE] a implémenté un réseau appliqué à la reconnaissance de caractères basé sur les réseaux RBF. Longtemps évités à cause de l'implémentation matérielle de la gaussienne, les réseaux RBF ont été implémentés matériellement en 1994 par la société IBM.

Les performances des deux réseaux sont réunies dans le tableau V.12.

	RBF -ESIEE	NEOCO
Nom	IBM ZISC036	NLX110
Type	Carte PC sur bus ISA	Carte PC
Nombre de neurones total	576 neurones	1048576 neurones
Couche d'entrée	32*32	128*128
Taux de reconnaissance	96 %	-
Temps de traitement	64 μ s	300 millions de mac/s

Tableau V.12 : Performances des réseaux de neurones matériels : RBF et Neocognitron

Cinquième Partie : Validations expérimentales

L'architecture du Néocognitron est une architecture, d'un point de vue type d'opérations (addition/multiplication) et connectiques entre les différentes couches, proche de notre architecture.

Ainsi, en fonction des performances de la carte NEOCO (tableau V.12), et du nombre d'opérations par notre premier réseau (tableau V.11), nous pouvons estimer les temps de traitement, pour un caractère 64x64 présenté à l'entrée.

De même en connaissant les temps de traitement de la carte IBM pour un réseau RBF disposant d'une couche d'entrée de taille 24x24, nous pouvons estimer le temps de traitement de notre réseau classifieur.

Le tableau V.13 regroupe les estimations de temps de traitement pour le premier réseau, le second réseau et la totalité du système pour une image initiale de 64x64.

		Notre modèle	Cartes existantes
Premier réseau	Nombre de neurones	33 832	1048576
	Nombre de connections	1.98 E ⁸	2.01 E ¹⁰
	Nombre d'opérations	2.2 E ⁴	3.02 E ⁶
	Temps de traitement	730 μs	105 ms
Second reseau	Nombre de neurones	776	1076
	Temps de traitement	21μs	64μs
Totalité	Nombre de neurones	34 608	1049652
	Temps de traitement	761 μs	105 ms

Tableau V.13 : Evaluation du temps de traitement

Ainsi, se basant sur une adaptation moyenne de 3 boucles de rétroaction (tableau V.10), nous pouvons estimer le temps de traitement moyen pour une image à 1,7 ms.

Cependant, la durée moyenne de traitement doit être plus petite. En effet, les essais de rebouclage ont été effectués sur des caractères isolés. Dans la réalité, les caractères sont présents sous forme de chaîne. Ainsi, seul le traitement du premier caractère pourra être long. En effet, une fois la valeur du paramètre σ fixée correctement, on peut supposer que celle-ci ne variera que faiblement pour les caractères voisins.

Conclusion de la cinquième partie

Bien qu'il soit très difficile d'évaluer les performances globales de notre système et de le comparer avec les systèmes existants, les premiers essais montrent que les concepts développés pour notre système (les deux réseaux de neurones et la boucle de rétroaction) permettent d'améliorer la reconnaissance en milieu industriel par rapport aux techniques existantes.

La technique du centre/voisinage se révèle plus efficace pour la binarisation d'image que la technique du seuillage global. Ses avantages sont nets pour des images présentant des contrastes faibles, des variations de luminosité et pour des images présentant des marquages ou des supports légèrement dégradés.

L'invariance en orientation permet d'envisager la reconnaissance de caractères pour certains cas très spécifiques de marquage circulaire sans ajout d'une étape de prétraitement du type déroulé.

Les performances du classifieur sont également intéressantes. Ainsi, même avec un taux de 90% pour une base de caractères difficiles (images bruitées, caractères orientés, polices industrielles) ses performances sont supérieures à celles des systèmes actuellement commercialisés. En effet, grâce à l'utilisation de la notion de forme, la phase d'apprentissage est plus simple et le nombre de polices reconnues est beaucoup plus important.

Enfin, les premiers essais du système avec la boucle de rétroaction montrent ses capacités à s'autoadapter à un grand nombre de variétés de support et de marquage ainsi qu'aux variations de luminosité. Le système est alors capable de s'appliquer à un grand nombre d'applications OCR rencontrées dans le milieu industriel.

Les performances globales actuellement obtenues sont encourageantes. Cependant, afin d'envisager la réalisation complète d'un tel système, il est nécessaire de poursuivre la phase de validation et sûrement d'améliorer les propositions précédemment faites.

Conclusion Générale

Conclusion générale

Bien que la reconnaissance de caractères soit devenue très courante dans le monde informatique, pour les applications en milieu industriel et plus précisément de l'identification automatique, les systèmes d'OCR restent rares étant donné la nature et la qualité des solutions de marquage et des conditions d'acquisition.

Notre contribution est, après avoir précisé certains problèmes liés à l'OCR en milieu industriel, de proposer un système de reconnaissance mettant en oeuvre quelques solutions permettant d'en améliorer les performances.

Notre étude s'est orientée dès le début vers les réseaux de neurones. En effet, leur application à la classification et à la reconnaissance invariante a permis d'améliorer les performances des systèmes de reconnaissance de caractères.

En nous basant sur nos besoins et sur les possibilités qu'ils offrent, nous avons développé une nouvelle architecture de reconnaissance de caractères originale, basée sur l'utilisation de deux réseaux de neurones et d'une boucle de rétroaction, qui apporte une solution à une partie du problème de la reconnaissance de caractères industriels.

Le premier réseau, basé sur une nouvelle architecture neuronale, permet d'extraire un même modèle pour un caractère qui a subi, dans l'image, des déformations telles qu'une translation, une rotation et une homothétie. En plus de cette propriété d'invariance, cette architecture dispose d'un étage de vision qui permet au système de s'adapter à un certain nombre de variations lumineuses qu'elles soient dues au support ou au marquage ou aux conditions extérieures.

Le second réseau effectue la classification d'un modèle extrait par le premier réseau quelle que soit la police utilisée. Une modification apportée au réseau RBF a été proposée afin de permettre et de simplifier la reconnaissance de caractères ayant des polices spécifiques. Nous avons introduit la notion de forme qui facilite la reconnaissance des caractères représentés sous forme discontinue et qui limite le nombre de neurones sur la couche cachée.

Le dernier concept développé est une boucle de coopération entre le classifieur et le premier réseau. Celle-ci va adapter automatiquement le système aux caractéristiques lumineuses rencontrées.

Nous avons montré que notre système est capable de résoudre un certain nombre de problèmes liés à la reconnaissance de ces caractères en milieu industriel. Les essais réalisés mettent en évidence les possibilités supplémentaires qu'offrent notre système par rapport aux solutions existantes qui sont, l'adaptation automatique aux caractéristiques lumineuses, la possibilité d'effectuer la reconnaissance de caractères pour une grande variété de supports et de marquages et la reconnaissance multiforme pour un grand nombre de polices utilisées en marquage industriel.

D'un point de vue applicatif, notre système devra s'incorporer dans une chaîne de reconnaissance complète. En effet, il se limite actuellement à la seule reconnaissance de caractères isolés. Il est donc nécessaire d'y ajouter une étape de segmentation qui extrait, à partir d'une image capturée en niveau de gris, des caractères isolés et une étape de post-traitement qui vérifie le résultat de la reconnaissance. Cette dernière étape reste très facile à réaliser étant donné la présence, dans de nombreux cas d'applications industrielles, de code de redondance.

Des futurs travaux consisteraient à faire progresser l'insensibilité de la reconnaissance aux supports bruités qui demeure une importante limite à un système de reconnaissance globale.

Bibliographie

BIBLIOGRAPHIE

- [AAS97] K.Aas, L.Eikvil, "Decoding bar codes from human-readable characters", *Pattern Recognition Letters*, vol 18, pp 1519-1527, 1997.
- [AMI98] A.Amin, "Off-line arabic character recognition : The state of the art", *Pattern recognition*, vol 31, n°5, pp 517-530, 1998.
- [ANI92] J.Anigbogu, *Reconnaissance de textes imprimés multiformes à l'aide de modèles stochastiques et métriques*, Thèse de doctorat, Nancy, 1992.
- [ARC95] Archimag, "Les tendances'96 : La GED sous toutes ses formes", n°89, pp 18-23, 1995.
- [ATH96] D.Athamena, *Inspection, détection et reconnaissance de caractères en milieu industriel*, thèse de doctorat, INSA, Lyon, 1996.
- [AUG93] J.M.Auger, *Méthodes neuronales pour la reconnaissance de caractères manuscrits : comparaison et coopération*, Thèse de doctorat, Paris, 1993.
- [AVI93] M.Avila, C.Olivier, T.Paquet & Y.Lecoutier, "Procédure de reconnaissance de l'écriture manuscrite, basée sur des chaînes de Markov cachées et appliqués à un vocabulaire limité.", 14^{ème} colloque GRETSI, pp 803-806, Juan-les-pins, 1993.
- [BAB89] N.Baba, "A new approach for finding the global minimum of error function of neural networks", *Neural Networks*, vol.2, pp 367-373, 1989.
- [BAR91] E.Barnard & D.Cassasent, "Invariance and neural networks", *IEEE Transactions on Neural Networks*, vol.3 n°2, pp 498-508, 1991.
- [BAS95] K.Basulaim, *Application des réseaux de neuronaux en visionique : traitement d'images et classification*, Thèse de doctorat, LeHavre, 1995.
- [BAU72] L.Baum, "An inequality and associated maximization technique for probabilistic functions of Markov process", *Inequalities*, Vol.3, pp1-8, 1972.
- [BEA90] J.P.Asselin de Beauville, "Reconnaissance de chiffres manuscrits saisis sans contrainte.", *BIGRE* n°68, pp 89-101, LeHavre, Mai 90.
- [BED97] L.Bedini, A.Bozzi & A.Tonazzini, "Digital techniques for character recognition in old documents", *ERCIM news* n°28, 3 pages, 1997.
- [BEL91] S.Belkasim, M.Shridar & A.Ahamadi, "Pattern recognition with moments invariants : a comparative study and new results", *Pattern Recognition*, vol.24, pp 1117-1138, 1991.
- [BEL92] A.&Y.Belaid, *Reconnaissance des formes : Méthodes et applications*, InterEditions, Paris, 1992.

Bibliographie

- [BEZ81] J.Bezdek, C.Coray, R.Gunderson & J.Watson, "Detection and characterization of cluster substructure", *SIAM Journal of Applied Mathematics*, 40(2), pp 358-372, 1981.
- [BIE87] E.Biennenstock & C.Von der Malsburg, "A neural network for invariant pattern recognition", *Europhysics Letters*, 4, pp 121-126, 1987.
- [BIS95] C.Bishop, *Neural networks for pattern recognition*, Oxford University Press, 1995.
- [BLA89] F.Blayo, "Tour d'horizon sur les implémentations de réseaux de neurones.", *Réseaux connexionnistes en informatique : méthodes et applications*, 3^{ème} université d'été, université de Lyon I, 1989.
- [BLU93] J.Blue, "Computational experience with radial basis function networks", Computing and applied mathematics laboratory, NISTIR 5168, National institute of standards and technology, 12 pages, Gaithersburg, 1993.
- [BLU94] J.Blue, G.Candela, P.Grother, R.Chellappa & C.Wilson, "Evaluation of pattern classifiers for fingerprint and Ocr applications", *Pattern Recognition*, Vol.27(4), pp 485-501, 1994.
- [BOK92] M.Bokser, "Omnidocument Technologies", *Proceedings of The IEEE*, vol.80, no.7, pp 1066-1078, July, 1992
- [BOU91] J-L.Bourdon, *Parallélisme et cognition appliqués à la reconnaissance de caractères*, Thèse de doctorat, Université de Rouen, 1991.
- [BOU92] J.Bourrely, *Reconnaissance de formes et approximations de fonctions par réseaux neuromimétiques et par méthodes statistiques : approche théorique et expérimentale*, Thèse de doctorat, Paris, 1992.
- [BRI68] E.Brill, *Character recognition via Fourier descriptors*, Wescon, session 25, Los Angeles, 1968.
- [BRO88] D.Broomhead & D.Lowe, "Multivariable functional interpolation and adaptative networks", *Complex systems*, vol 2, n°3, pp 321-355, 1988.
- [BUR91] G.Burel, *Réseaux de neurones en traitement d'images : Des méthodes théoriques aux applications industrielles*, Thèse de Doctorat, Brest, 1991.
- [CAS86] D.Casasent & W.Chang, "Correlation synthetic discriminant functions", *Applied Optics*, vol.25, n°14, pp 2343-2350, 1986.
- [CHA98] B.Chaudhuri & U.Pal, "A complete printed bangla OCR system", *Pattern recognition*, vol.31, n°5, pp 531-549, 1998.
- [CHE77] A.Chehikian, *Conception et réalisation d'un automate de reconnaissance de caractères alphanumériques*, Thèse d'Etat, Grenoble, 1977.

Bibliographie

- [CHO95] S.Choy, C.Clifford & W.Siu, "New Single-pass algorithm for parallel thinning.", C.V.I.U.Vol.62, N°1, pp 69-77, 1995.
- [CHT97] Y.Chtioui, *Reconnaissance automatique des semences par vision artificielle basée sur des approches statistiques et connexionnistes*, Thèse de doctorat, Université de Nantes, 1997.
- [COH94] T.Cohignac, *Reconnaissance de formes planes*, Thèse de doctorat, Université Paris IX, 1994.
- [COO90] P.B.Coogan & J.E.Campbell, "Machine vision systems are 'lookin' good", IC&C, 1990, vol 63, n°11, pp 76-79, 1990.
- [COS89] M.Coster & J.L.Chermant, "Précis d'analyse d'images.", Presses du CNRS, 1989.
- [DAV 89] E.Davalo & P.Naïm, *Des réseaux de neurones*. Eyrolles 1989.
- [DAV93] E.Davalo & P.Naïm, *Réseaux de neurones*. Eyrolles 1993.
- [DEF95] O.Deforges, *Segmentation robuste d'images de documents par une approche multirésolution. Conception et validation d'une architecture parallèle dédiée*, Thèse de doctorat, Nantes, 1995.
- [DEL80] C.Delannoy, "Un algorithme rapide de recherche des plus proches voisins", RAIRO informatique, 14(3), pp 275-286, 1980.
- [DEM63] W.Deming, *Two Papers by Bayes*, New York, Hafner, 1963.
- [DES95] P.Deseilligny, H.Le Men & G.Stamon, "Lecture automatique des écritures sur cartes scannées", L'écrit et le document, Traitement du signal, vol 12, Numéro spécial "Analyse de documents", n°6, pp 639-652, 1995.
- [DEV92] C.Devouge, *Quelques aspects mathématiques de l'autoorganisation neuronale et des perceptrons multicouches*, Thèse de doctorat, Paris, 1992.
- [DID82] E.Diday, J.Lemaire, J.Pouget & F.Testu, *Éléments d'analyse de données*, Bordas, Paris, 1982.
- [DUB90] B.Dubuisson, *Diagnostic et reconnaissance des formes*, Hermès, Paris, 1990.
- [FRE91] P.W.Frey & D.J.Slate, "Letter recognition using holland-style adaptive Classifiers", Machine Learning, vol 6, pp 161-182, 1991.
- [FU82] K.S.Fu, *Syntactic pattern recognition and applications*, Prentice-Hall 1982.
- [FUK75] K.Fukushima, "Cognitron : A self-organizing multilayered neural network", Biol Cybernetics, vol 20, pp 121-136, 1975

Bibliographie

- [FUK88] K.Fukushima, "Neocognitron : A hierarchical network capable of visual pattern recognition", *Neural Networks*, Vol 1, pp 119-130, 1988.
- [FUK92] M.Fukumi, S.Omatu, F.Takeda & T.Kosaka, "Rotation-invariant neural pattern recognition system with application to coin recognition", *IEEE Trans.Neural Networks*, Vol 3, pp 272-279, 1992.
- [FUK97] M.Fukumi, S.Omatu & Y.Nishikawa, "Rotation-invariant neural pattern recognition system estimating a rotation angle", *IEEE Trans.On Neural Networks*, vol.8, n°3, pp 568-581, 1997.
- [GAI83] G.Gaillat, *Méthodes syntaxiques de la reconnaissance de formes*, ENSTA, Paris, 1983.
- [GAU 92] P.Gaussier, *Simulation d' un système visuel comprenant plusieurs aires corticales : application à l' analyse d' une scène*, Thèse de doctorat , Paris XI, nov 1992.
- [GIL90] K.Gill, "OCR aids robotic assembly", *Sensor review*, vol 10, n°4, pp 170-173, 1990.
- [GOK86] D.Gokana, *Contribution à la reconnaissance automatique de caractères manuscrits. Application à la lecture optique de caractères sur supports mobiles*, Thèse de doctorat, Orsay,1986.
- [GOR93] L.O'Gorman, "The document spectrum for page layout analysis", *IEEE Trans Pattern Analysis and Machine Intelligence*, No15(11), pp 1162-1173, 1993.
- [GOV90] V.Govindan & A.P.Shivaprasad, "Character Recognition - A Review.", *Pattern Recognition*, vol 23, n°7, pp 671-683, 1990.
- [GRO93] P.J.Grother & G.T.Candela, "Comparaison of handprinted digit classifiers", National Institute of Standards and technology, Image Recognition Group, 1993.
- [GRU97] M.Gruber & K.Y.Hsu, "Moment-based image normalization with high noise-tolerance", *IEEE trans on PAMI*, vol 19, n°2, pp 136-140, 1997.
- [GUP94] M.Gupta & G.Knopf; *Neuro-visions systems*, IEEE Press, 1994
- [HAR68] P.Hart, "The condensed nearest neighbor rule", *IEEE Trans.On Information Theory*, vol IT-14, pp 515-516, 1968.
- [HEB 49] D.O.Hebb, *The organization of behavior*. New York : Wiley, 1949.
- [HEN94] H.Henocq, *Localisation et reconnaissance de formes tridimensionnelles en vision et robotique*, Thèse de doctorat, Université de Bretagne Occidentale, 1994.
- [HEN96] J.L.Henry, *Reconnaissance et contexte : Une approche coopérative pour la lecture de textes imprimés*, Thèse de doctorat, Lyon, 1996.

Bibliographie

- [HER94] S.Herault & C.Jutten, *Réseaux neuronaux et traitement du signal*, Collection Hermès, 1994.
- [HEU94] L.Heutte, *Reconnaissance de caractères manuscrits : Application à la lecture automatique des chèques et des enveloppes postales*, Thèse de doctorat , Rouen, 1994.
- [HIL93] T.H.Hildebrandt & W.Liu, "Optical recognition of handwritten chinese characters : Advances since 1980", *Pattern recognition*, vol 26, n°2, pp 205-225, 1993.
- [HIR97] M.Hirooka, K.Sumi, M.Hashimoto, H.Okuda & S.Kuroda, "Hierarchical distributed template matching", in *Proc Machine Vision Applications*, SPIE Proceedings, pp 176-183, Feb 97.
- [HOP 85] J.J.Hopfield, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, No52, pp 141-152, 1985.
- [HU62] M.Hu, "Visual pattern recognition by moments invariants", *IRE Trans.Inform. Theory IT*, vol.8, pp 179-187, 1962.
- [HUB 88] D.Hubel, *Le cerveau : le cerveau*. Pour la science, diffusion Belin, 1988.
- [HUL84] J.Hull, G.Krishnan, P.Palumbo & S.Srihari, "Optical character recognition techniques in mail sorting : A review of algorithms" *Advanced Character recognition projet*, USPS, Technical report n°214, 1984. Cx Available from Internet: <URL: <http://www.cedar.buffalo.edu/Publications/techReps/Ocrtech84/paper84.html> >
- [ISO1073-1] ISO 1073-1:1976, Alphanumeric character sets for optical recognition—Part 1 : Character set OCR-A – Shapes and dimensions of the printed image.
- [ISO1073-2] ISO 1073-2:1976, Alphanumeric character sets for optical recognition—Part 2 : Character set OCR-B – Shapes and dimensions of the printed image.
- [ISO1831] ISO 1831:1980, Printing specifications for optical character recognition
- [ISO2033] ISO 2033:1983, Traitement de l'information : Codage des jeux de caractères pour reconnaissance automatisée (MICR et ROC).
- [ISO9001] NF EN ISO9001, Modèle pour l'assurance de la qualité en conception, développement, production, installation et prestations associées, 1994.
- [JIA91] X.F.Jiang & H.Bunke, "Simple and fast computation of moments", *Pattern Recognition*, vol 24, n°8, pp 801-806, 1991.
- [JAI88] A.K.Jain & R.C.Dubes, *Algorithms for clustering data*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

Bibliographie

- [JAI97] A.K.Jain & J.Mao, "Guest editorial special issue on artificial neural networks and statistical pattern recognition", *IEEE trans on Neural Networks*, Vol 8, n°1, pp 1-4, 1997.
- [JOD94a] J.F.Jodouin, *Les réseaux de neurones*, Collection Hermes, 1994.
- [JOD94a] J.F.Jodouin, *Les réseaux neuromimétiques*, Collection Hermes, 1994.
- [JOU94] C.Jou, S.You & L.Chang, "Analysis of hidden nodes for multi-layer perceptron neural networks", *Pattern recognition*, vol 27, N°6, pp 859-864, 1994.
- [KAM93] M.Kamel & A.Zhao, "Extraction of binary character/graphics images for grayscale documents images", *Computer Vision Graphics and Image Processing*, No55(3), pp 1162-1173, 1993.
- [KAN92] T.Kanaoka, R.Chellappa, M.Yoshitaka & S.Tomita, "A high-order neural network for distortion invariant pattern recognition", *Pattern Recognition* (13), pp 837-841, 1992.
- [KHO88] A.Khotanzad and J.-H.Lu, "Distortion Invariant Character Recognition by a Multi-Layer Perceptron and Back-Propagation Learning", *Proc.IEEE Int. Conf.Neural Networks*, pp.I625-I632, 1988.
- [KHO90] A.Khotanzad, "Classification of invariant representations using neural network.", *IEEE Transaction on Acoustic Speech and Signal Processing*, Vol 38, n°6, pp 1028-1038, Juin 1990.
- [KIM95] R.Kimmel, D.Shaked, N.Kiryati & A.Bruckstein, "Skeletonization via distance maps and levels sets.", *C.V.I.U.* Vol 62, N°3, pp 382- 391, November 1995.
- [KIT98] J.Kittler, M.Hatef R.Duin & J.Matas, "On combining classifiers", *IEEE on pattern analysis and machine intelligence*, vol 20, n°3, pp 226-239, 1998.
- [KOH 84] T.Kohonen, *Self organization and associate memory*, Springer Verlag, Berlin, 1984.
- [KOH 88] T.Kohonen, "The neural phonetic typewriter," *Computer*, pp11-22, march 1988.
- [KOH90a] T.Kohonen, "The self-organizing map", *Proceedings of the IEEE*, vol 78, n°9, pp 1464-1480, 1990.
- [KOH90b] T.Kohonen, *Statistical pattern recognition*, In *Advance neural computer*, Elsevier Science, 1990.
- [KUN93] M.Kunt, *Traitement numérique des images*, Presses polytechniques et universitaires romandes, Lausanne, 1993.

Bibliographie

- [KWO93] H.Kwon, B.Kim, D.Cho & H.Hwang, "Scale and rotation invariant pattern recognition using complex-log mapping and augmented second order neural network", *Electronics Lett.*, Vol.29(7), pp 620-621, 1993.
- [LAC96] S.Lucas & A.Amiri, "Statistical syntactic methods for high performances OCR", Department of Electronic Systems Engineering, University of Essex, 27 pages, 1996.
- [LAD93] M.Lades , "Distortion Invariant Object recognition in the dynamic link architecture", *IEEE trans .computers*, Vol.42 (3), pp 300-311, 1993.
- [LAN67] G.Lance & N.William, "A general theory of classificatory sorting strategies : Hierarchical systems", *Computation Journal*, 9, 1967.
- [LAP51] P.Laplace, *A Philosophical Essay on Probabilities*, unabridged and unaltered reprint of Truscott and Emory translation, Dover Publications, Inc., New York, 1951.
- [LEB 89] E.Le Bail & A.Mitchie, "Quantification vectorielle par le réseau neuronal de Kohonen," *Traitement du signal*, vol 6, No 6, pp 529-539, 1989.
- [LEC 87] Y.Lecun, *Modèles connexionnistes de l' apprentissage*. Thèse de doctorat. Université de Paris IV, 1987.
- [LEC89] Y Lecun, "Handwritten Digit Recognition : Applications of neural networks chips and automatic learning.", *IEEE Communications Magazine*, pp1523-1535, November 1989.
- [LEE97] S.W.Lee, *Special issue : Oriental character recognition*, pattern recognition, vol 30, n°8, pp 1253-1254, 1997.
- [LEM 89] H.Lemberg, "les réseaux de neurones artificiels," *INFOPC* No 71, pp 20-28, 1989.
- [LET90] A.Leturq, *La reconnaissance des caractères manuscrits sans apprentissage*, Thèse de Doctorat, Paris VI, 1990.
- [LET95] E.Lethellier, M.Leroux & M.Gilloux, "Traitement des montants numériques de chèques postaux. Une méthode de segmentation basée sur la reconnaissance", *L'écrit et le document, Traitement du signal*, Vol 12 n°6 pp 551-559, 1995.
- [LI97] B.Li, D.Zhao, R.Villabos & S.Cabrera, "Robust-statistic-based template matching", in *Proc Machine Vision Applications*, SPIE Proceedings, pp 176-183, Feb 97.
- [LIU97] C.L Liu, J.H.Kim & R.W.Dai, "Multiresolution locally expanded HONN for handwritten numeral recognition", *Pattern recognition letters*, vol 18, pp 1019-1025, 1997.
- [LIP 87] R.Lippman, "An introduction to computing with neural nets," *IEEE ASSP mag*, pp4-23, apr.1987.

Bibliographie

- [LUB95] M.Lubkov, "L'OCR arrive à point", Archimag, n°87, pp 40-43, 1995.
- [LOO97] C.Looney, *Pattern recognition using neural networks : Theory and algorithms for engineers and scientists*, Oxford University Press, 1997.
- [MAI93] J.P.Maillet, "Les techniques de reconnaissance de caractères.", Archimag, n° 66, pp 33-39, 1993.
- [MAN86] J.Mantas, "An overview of character recognition methodologies", Pattern Recognition, Vol 19, n°6, pp 425-430,1986.
- [MAR 90] A.Marren, C.Harston & R.Pap, *Handbook of neural computing applications*.Academic press Inc, San Diego C.A., 1990.
- [MAR93] A.Marcelli, B.Bontempi, G.Parrella & D.Russi, "Using genetics algorithms for on-line recognition", Rapport de recherche n°67, Université de Naples "Frederico II" 52 pages, 1993.
- [MAS92] M.H.Masson, *Réseaux de neurones et reconnaissances de formes*, Thèse de Doctorat, U.T.C.Compiègne, 1992.
- [MAT92] K.Matsuzaki, H.Suzuki, H.Ohnari, M.Ohno & M.Isobe, "Neural network based industrial character recognition system and application for laser marking characters", Journal of Transactions of the institute of Electrical Engineers of japan, vol 112-D, n°6, pp 561-568, 1992.
- [MCC 43] W.S.McCulloch & W.Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of mathematical biophysics*, vol 5, pp.115-133, 1943.
- [MEA87] C.Mead, "Silicon models of neural computation.", 1st IJCNN, Vol 1, pp 91-109, San Diego, 1987.
- [MEN88] M.Menon & K.Heinemann, "Classification of patterns using a self-organizing neural network", Neural Networks, Vol 1, pp 201-215, 1988.
- [MIC84] L.Miclet, *Méthodes structurelles de la reconnaissance de formes*, Eyrolles, Paris, 1984.
- [MIL93] M.Milgram, *Reconnaissance de formes : Méthodes numériques et connexionnistes*, Armand Colin Editeur, Paris, 1993.
- [MIN 69] M. Minsky & S. Papert, *Perceptrons : An introduction to computational geometry*.(book), the M.I.T Press, Cambridge, 1969.
- [MIT89] B.Mitchell & A.Gillies, "A model based computer vision system for recognizing handwritten ZIP codes.", Machine Vision and Applications, Vol 2, n°4, pp 231-243, 1989.

Bibliographie

- [MOR92] S.Mori and C.Y.Suen and K.Yamamoto, "Historical Review of OCR Research and Development".Proceedings of The IEEE, vol.80, no.7, pp 1029-1058, July, 1992.
- [NAT 93] R.Natowicz, "Segmentation d'images par cartes auto-organisantes de Kohonen," *14^{ème} Colloque du GRETSI*, pp 703-706, 1993.
- [NEU96] NeuralWare, *Reference guide for NeuralWorks explorer*, NeuralWare Inc, 1996.
- [NIS95] H.Nishida, "Structural feature extraction using multiples bases.", C.V.I.U.Vol 62, N°1, pp 78-89, July 1995.
- [OHT92] M.Ohta, Y.Anza & S.Yoneda, "A neural network with a mechanism for detecting and zooming", *International Journal Electronique*, N° 73, pp 447-459, 1992.
- [OGI94] J.M.Ogier, *Contribution à l'analyse automatique de documents cartographiques : Interprétation de données cadastrales*, Thèse de doctorat, Rouen, 1994.
- [PAK85] K.Pakker & A.Chehikian, "Reconnaissance de caractères multipolices par analyse structurelle hiérarchique", Actes du 5^{ème} congrés AFCET : Reconnaissance des formes et intelligence artificielle, Grenoble, pp 817-828, 1985.
- [PAO89] Y.Pao, *Adaptative pattern recognition and neural networks*, Addison-Wesley Pub.Co.Inc.Reading : Massachusetts, 1989.
- [PAV77] T.Pavlidis, *Structural pattern recognition : an introduction*, Springer Verlag, 1977.
- [PAW89] T.Pawlicki, "Neural network models and their applications to handwritten digit recognition", IEEE Communications Magazine, November 1989.
- [PHI98] T.Phinner, "A brief history of type", 10 pages, 1998, Available from Internet <URL:<http://www.redsun.com/typography/abriefkistoryoftype>>
- [PLA89] R.Plamondon & G.Lorette, "Automatic signature verification and writer identification : The state of the art", *Pattern recognition*, Vol 32, n°2, pp 107-131, 1989.
- [POS87] J.G.Postaire, *De l'image à la décision : analyse des images numériques et théorie de la décision*, Dunod, Paris, 1987.
- [POW87] M.J.Powell, "Radial basis functions for multivariable interpolation : A review", In *Algorithms for Approximation of Functions and Data*, M. J. C and C. M. G., Eds., Oxford University Press, pp 143-167, 1987.
- [RAB89] L.R.Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition", *Proceedings of IEEE*, Vol 77, n°2, pp 257-286, 1989.

Bibliographie

- [RAO94] R.Rao & J.Ben-Arie, "Multiple template matching using the expansion filter", IEEE trans on.Circuits and systems for video technology, vol.3, n°3, pp 490-503, 1994.
- [REM92] C.Remy, "Lecture optique de caractères", Techniques de l'ingénieur, H1405, vol H1, 10 pages, 1992.
- [ROC94] Rockwell, *CVIM OCR2-Pak User guide*, 1994.
- [ROS59] F.Rosenblatt, *Principles of neurodynamics*. Spartan books, New York, 1959.
- [ROS82] A.Rosenfeld & A.Kak, *Digital pictures processing*, Academic Press, San diego, 1982.
- [ROT96] I.Rothe, H.Süsse & K.Voss, "The method of normalization to determine invariants", IEEE Trans On PAMI, Vol.18, n°4, pp 366-376, 1996.
- [ROU85] M.Roux, *Algorithmes de classifications*, Masson, Paris, 1985.
- [RUM86] D.Rumelhart, G.Hinton & R.J.Williams, "Learning internal representations by error propagation", in Rumelhart&McClelland Editions Parallel distributed processing – Explorations in microstructure of cognition.Cambridge, MA : MIT Press, 1986.
- [RYB91] I.Rybak, A.Golovan, V.Gusakova, N.Shevtova & L.Podladchikova, "A neural network system for active visual perception and recognition", *Neural Networks world*, Vol.4, pp 245-250, 1991.
- [SAC92] E.Sackinger, B.Boser, J.Bromley, Y.Lecun & L.Jackell, "Application of the ANNA neural network chip to high speed character recognition.", IEEE trans on neural network, Vol.3, n°9, pp 498-505, Mai 92.
- [SCH82] J.Schurman, "Reading machines", 6th Int J Conf.On Pattern Recognition, Munich, pp 1032-1044, 1982.
- [SCH96a] H.Schwenk & M.Milgram, "Constraint tangent distance for on-line character recognition", Int Conf.on Pattern Recognition, pp 515-519, 1996.
- [SCH96b] H.Schwenk, *Amélioration des classifieurs neuronaux par incorporation de connaissances explicites : Application à la reconnaissance de caractères manuscrits*, thèse de doctorat, Université Pierre et Marie Curie, 1996.
- [SEN92] A.W.Senior, "Off-line handwriting recognition : A review and experiments", Cambridge University Engineering Department, F-INFENG, TR105, 23 pages, 1992.
- [SHE96] Y.Sheng & J.Duvernoy, "Circular Fourier radial Mellin descriptors for pattern recognition", J.Optical Society America, Optics and Image Sci, vol 3, pp 885-888, 1996.

Bibliographie

- [SHE97] D.Shen & H.S.Horace, "Generalized affine invariant image normalization", IEEE Trans.on PAMI, vol 19, n°5, pp 431-440, 1997.
- [SHR87] M.Shridhar & A.Badreldin, "Context-directed segmentation algorithm for hand written numeral strings.", Image and Vision Computing, vol 5, n°1, pp 3-9, February 1987.
- [SIE96] Siemens, *Simatic S5 : Videomat IV Manual*, 1996.
- [SIM92] P.Simard, B.Victorri, Y.Lecun & J.Denker, "Tangent prop – a formalism for specifying selected invariances in an daptative network", Advances in neural information processing systems, vol.4, pp 50-58, CA:Morgan Kaufman, 1992.
- [SIM93] P.Simard, Y.Lecun & J.Denker, "Efficient pattern recognition using a new transformation distance", Advances in neural information processing systems, Vol.5, pp 50-58, CA:Morgan Kaufman, 1993.
- [SIM98] J.C.Simon, "La reconnaissance des formes à l'épreuve des faits", LaRecherche, n°312, pp 58-62, 1998.
- [SHE97] D.Shen & H.Horace, "Generalized affine invariant image normalisation", IEEE Trans On.PAMI, Vol.19, n°5, pp 431-440, 1997.
- [SPI92] L.Spirkovska & M.B.Ried, "Robust position, scale and rotation invariant object recognition using high-order neural networks", Pattern recognition, vol.25, n°9, pp 975-985, 1992.
- [SPI93] L.Spirkovska & M.B.Ried, "Coarse-coded high-order neural networks for PRSI Object recognition", IEEE trans. on Neural networks, Vol.4, n°2, pp 276-283, 1993.
- [SPE90] D.F.Specht "Probabilistic neural networks", Neural Networks, Vol.3, n°1, pp 109-118, 1990.
- [SRI92a] S.N.Srihari, "High-performance reading machines", Proceedings of the IEEE, vol 80, n°7, pp 1120-1132, 1992.
- [SRI92b] N.Sriniva & M.Jouaneh, "A neural network for invariant pattern recognition", IEEE trans.Signal Process.Vol.40, pp 1595-1599, 1992.
- [STE 88] C.Stevens, *Le cerveau : le neurone*.Pour la science, diffusion Belin, 1988.
- [STR89] L.Stringa, "Efficient classification of totally unconstrained handwritten numerals with a trainable neural network.", Pattern Recognition Letters, vol 10 ,pp 273-280, October 1989.
- [SUE80] C.Suen, M.Berthod & S.Mori, "Automatic recognition of handprinted characters : The state of art", IEEE Computer Vision and Pattern Recognition, vol 2, n°4, pp 469-487, 1980.

Bibliographie

- [SUE90] C.Suen, C.Nadal, T.Mai & L.Lam, "Recognition of handwritten numerals based on the concept of multiexperts", Proc.of the international Workshop on frontiers in handwriting recognition, CENPARMI, Montreal, April 6-7, pp 131-144, 1990.
- [TAK94] M.Takano, T.Kanaoka, J.Skrzypek and S.Tomita, "A note on a higher-order neural network for distortion invariant pattern recognition", Pattern Recognition Lett., Vol 15(6), pp 631-635, 1994.
- [TAN81] S.L.Tanimoto, "Template matching in pyramids", Computer graphics and image processing, vol 16, pp 356-369, 1981.
- [TRE95] J.Trenkle, A.Gillies, E.Erlandson & S.Schlosser, "Arabic Character Recognition", Symposium on Document Image Understanding Technology (SDIUT), 24-25 October, Bowie, pp 191-195, 1995.
- [TRI95] Ø.Trier & T.Taxt, "Evaluation of Binarization Methods for Document Images", IEEE Trans On Pattern Analysis And Machine Intelligence, Vol.17, No3, pp 312-315, 1995.
- [TRO93] A.Trotin, *Un réseau de neurones pour la classification d'images : De l'algorithme à l'implantation*, Thèse de doctorat, Ecole National supérieure des Télécommunications, Paris, 1993.
- [WAN93] D.Wang, "Pattern recognition : Neural Networks in perspective", IEEE Expert, pp 52-60, 1993.
- [WAN96] S.S.Wang & W.G.Lin, "A new self-organizing neural model for invariant pattern recognition", Pattern Recognition, Vol.29, n°4, pp 677-687, 1996.
- [WEN94] E.Wendling, *Informatique avancée : méthodes de résolution de problèmes complexes*, Editions ADIT, Strasbourg, 1994.
- [WHI83] J.White & G.Rohrer, "Image thresholding for optical character recognition and other applications requiring character image extraction," IBM J.Reseach and developpement, vol.27, No 4, pp 400-411, July 1983.
- [WID88] B.Windrow, R.Winter, R.Baxter, "Layered neural nets for pattern recognition", IEEE trans.On Acoust. SpeechSignal Processing, vol 36, n°7, pp 1109-1118, 1988.
- [WOO96] J.Wood, "Invariant pattern recognition : a review", *Pattern Recognition*, Vol 29 n°1, pp 1-17,1996.
- [WRI90] P.T.Wright, "On-line recognition of handwriting", GEC Journal of research, vol 8 n°1, pp 42-48, 1990.
- [WU95] V.Wu & R.Manmatha, "Extracting text from greyscale images", UM-CS-995-88, UMass, Amherst, Oct, 1995.

Bibliographie

- [WU97] V.Wu, R.Manmatha & E.Riseman."Finding Text In Images", Intl conf.on Digital Libraries (DL'97), pp 1-10, 1997.
- [YAS97] M.Yasuda, K.Yamamoto & H.Yamada, "Effect of the perturbed correlation method for optical character recognition", Pattern recognition, Vol 30, n°8, pp 1315-1320, 1997.
- [YIN92] H.Yin, *L'étude de réseau neuronal d'apprentissage non supervisé et ses applications aux reconnaissances de formes*, Thèse de Doctorat, U.T.C.Compiègne, 1992.
- [YUC93] C.Yuceer & K.Oflazer, "A rotation, scalling and translation invariant pattern classification system", Pattern Recognition, Vol 26(5), pp 687-710, 1993.

BIBLIOGRAPHIE CORRESPONDANTE AU TRAVAIL DE THESE

S.Lecoeuche, D.Deguillemont, JP Dubus, "Autoadaptive neural networks applied to industrial characters recognition", Applications of Digital Image Processing XIX, SPIE96 International Symposium, Denver, 4-9 Aout, pp. 571-582, 1996.

S.Lecoeuche, D.Deguillemont, JP Dubus, "Neural system applied on an invariant industrial character recognition", Applications of Artificial Neural Networks in Image Processing II, Electronic Imaging 97, San José, 8-14 Fevrier, pp. 88-99, 1997.

S.Lecoeuche, D.Deguillemont, JP Dubus, "Design of a new character recognition system for industrial applications", GRWS98, Munich, 21-25 Septembre, pp (A paraître dans LNCS des éditions Springer) 1998.

Annexe A :

Notions sur
les réseaux de neurones

Depuis ces dernières années, la reconnaissance de forme a subi un regain d'intérêt grâce à l'apparition de nouvelles méthodes d'intelligence artificielle telles que la logique floue, les réseaux de neurones et les algorithmes génétiques. Toutes ces méthodes ont leurs origines dans les recherches biologiques. Tandis que la logique floue est particulièrement adaptée à l'aide à la décision, les réseaux de neurones sont parfaitement appropriés à la reconnaissance de forme. Leur structure, leur processus d'apprentissage leur ont permis d'obtenir de très bons résultats dans des opérations de classification. L'utilisation des réseaux neuronaux rendent l'interprétation plus aisée. En effet, ces derniers tentent de simuler les mécanismes du cerveau en reproduisant ses structures de base. Ils possèdent des caractéristiques identiques à ce dernier : facultés d'auto-organisation, d'association, d'apprentissage. Les modèles neuromimétiques ont cherché à copier les mécanismes biologiques ou naturels, (Le réseau d'Hopfield, inspiré de la théorie de verres de spin ou le réseau de Kohonen inspiré de l'étude biologique du cortex auditif). Les modèles connexionnistes plus nombreux et plus puissants ont cherché à approximer des fonctions permettant de réaliser la classification, les perceptrons multicouches étant les plus connus d'entre eux.

Généralités sur les réseaux de neurones

C'est grâce à la découverte de deux cybernéticiens que l'élaboration de réseaux de neurones fut possible. En 1943, Mc CULLOCH et PITTS [MCC43], se basant sur une étude biologique ont modélisé le neurone, créant ainsi le premier neurone formel.

A.1 Les Fondements Biologiques

Le neurone biologique est une cellule vivante qui, recevant des informations, engendre un influx nerveux [STE 88].

Sa structure peut être décomposée en trois parties:

- un corps cellulaire assurant le métabolisme et les fonctions du neurone.
- des prolongements centripètes ou dendrites qui transmettent au corps cellulaires les informations périphériques qu'ils recueillent.
- un prolongement centrifuge ou axone qui transmet l'influx nerveux né du corps cellulaire lorsque l'excitation issue des dendrites est supérieur à un seuil.

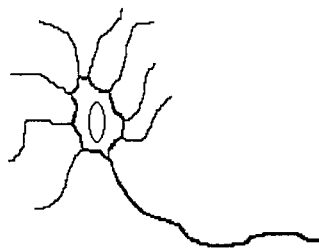


Figure I. A.11 : Le neurone biologique

Le regroupement et la succession de nombreux neurones en circuits permettent le fonctionnement nerveux. L'articulation entre les neurones se fait par de nombreuses synapses. Celles-ci comportent un bouton synaptique sur la ramification terminale d'un axone et un récepteur synaptique sur la dendrite ou le corps cellulaire d'un autre neurone. Bien que le nombre et la

position des neurones soient fixes, l'évolution est rendue possible grâce à la modification des liaisons inter-neurales. Celles-ci sont modifiées en fonction des événements rencontrés par le réseau [HUB88].

A.2 Les réseaux de neurones artificiels

Le réseau est caractérisé par

- le type des neurones utilisés
- sa topologie
- son évolution ou méthode d'apprentissage

A.2.1 Le neurone formel

Le neurone formel imite le plus parfaitement possible la structure du neurone biologique. Il se compose d'un vecteur d'entrée de composante e_i , d'une fonction sommation E déterminant l'excitation totale reçue par le neurone, d'une fonction d'activation A traduisant l'activité du neurone et enfin d'une fonction de sortie S déterminant la valeur de l'influx [MCC43].

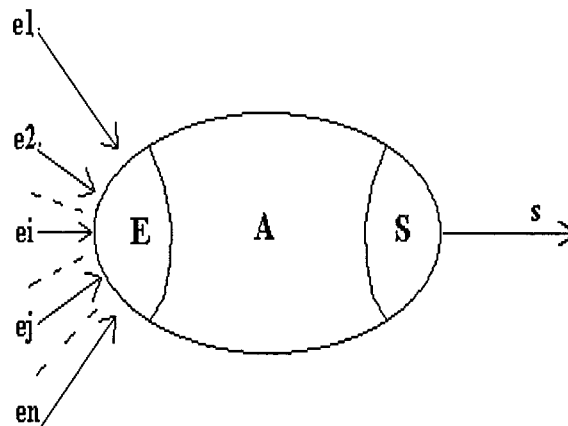


Figure I. A.12 : Le neurone formel

Tous les neurones utilisés dans les réseaux possèdent cette architecture. Ce sont les fonctions utilisées pour E , A , S qui définissent le neurone.

Ainsi : → e_i peut être binaire ou réelle

→ $E = h(e_1, e_2, e_3, \dots, e_n)$ où $h()$ est une fonction sommation du type :

- linéaire
- affine
- polynomiale
- ...

→ $A = f(E)$ où $f()$ est une fonction de la forme :

- binaire à seuil
- linéaire à seuil
- sigmoïde
- stochastique

→ $S = g(A)$ avec g , dans la plupart des cas, qui est la fonction identité tel que $S = A$

A.2.2 La topologie du réseau

Le réseau est caractérisé par l'ensemble des interconnexions de neurones. Les structures ainsi obtenues décrivent deux types de réseaux :

- Les réseaux à couches
- Les réseaux totalement connectés

A.2.2.1 Les réseaux à couches

Ce type de réseau présente une couche d'entrée recevant les informations extérieures et une couche de sortie renvoyant les résultats du traitement. Entre les deux, se situent plusieurs couches cachées qui effectuent ce traitement. Les neurones utilisés sont de même type dans une même couche. Cependant, ceux de couches différentes peuvent être de types différents.

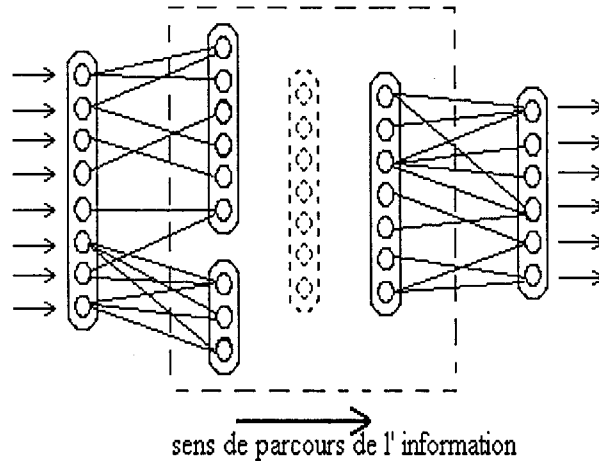


Figure I. 13: Le réseau à couches

L'information à traiter est appliquée au réseau sous la forme d'un vecteur d'entrée puis propagée vers la couche de sortie suivant le même sens de déplacement. Le résultat apparaît lui aussi sous forme de vecteur. Les connexions entre chaque neurone de couches différentes peuvent être totales ou partielles. Ainsi un neurone peut être connecté soit à certains, soit à tous ceux de la couche suivante. Cependant, les neurones d'une même couche ne sont pas interconnectés. [DEV92]

Ce type de réseau est aussi appelé *réseau feedward*.

A.2.2.2 Les réseaux totalement connectés

Ce type de réseaux est constitué de neurones jouant à la fois le rôle d'entrée et de sortie.

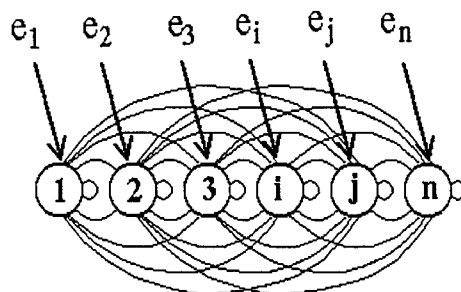


Figure I. A.14 : Le réseau totalement connecté

L'information est appliquée sur chaque neurone, sous forme de vecteur. Ceux-ci communiquent avec tous les autres et peuvent aussi communiquer avec eux-mêmes. C'est l'état en sortie de ces neurones qui déterminent les composantes du vecteur de sortie.

Ce type de réseau est aussi appelé *réseau feed-back*.

A.2.3 L'evolution du reseau

A chaque connexion de deux neurones, on associe un poids caractérisant leur interaction. Cela correspond biologiquement aux liaisons synaptiques.

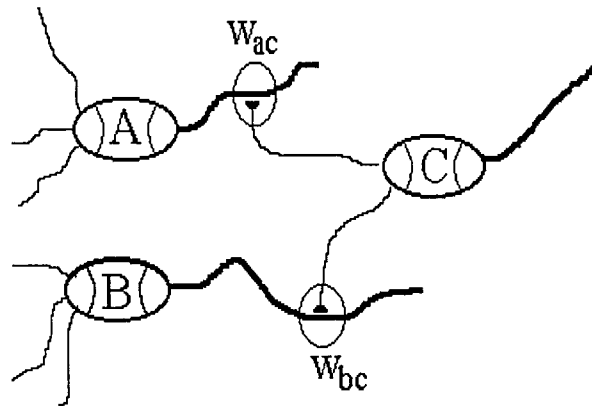


Figure I. 15: Les liaisons synaptiques

La connaissance d'un réseau est mémorisée dans son architecture et dans la valeur de ses poids. Son apprentissage se fait, comme pour le cerveau, par le biais d'une adaptation des liaisons synaptiques en fonction du type de problème à traiter. L'apprentissage d'un réseau consiste à modifier les connexions. En général, apprendre pour un réseau consiste à ôter les connexions existantes ; développer de nouvelles connexions ; modifier la force des connexions.

Pour ce faire, plusieurs règles d'apprentissage ont été établies. Ces règles peuvent se regrouper en deux classes: l'apprentissage compétitif et l'apprentissage associatif.

La première règle d'apprentissage a été déduite de la règle de Hebb [Hebb49]. Bien que celle-ci fut élaborée pour les réseaux biologiques; les réseaux de neurones formels doivent également s'y conformer. Cette règle décrit *la nécessité d'activité conjointe des éléments pré et post synaptiques*.

Elle peut être formulée de la façon suivante : $\Delta W_{ij} = a X_i X_j$

avec X_i : l'activité du neurone i pré synaptique. .

X_j : l'activité du neurone j post synaptique.

W_{ij} : le poids de la connexion

- Ainsi :
- si deux neurones sont actifs en même temps, leur liaison est renforcée. (a)
 - si les deux neurones sont au repos, leur poids reste inchangé. (b)
 - si la cellule excitatrice ou la cellule réceptive est activée et l'autre non le poids prend une valeur plus faible. (c)

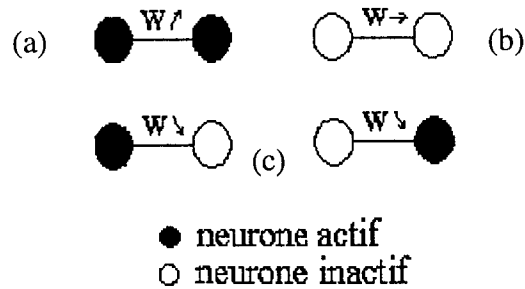


Figure I. A.16 : La règle de Hebb

La règle de Hebb fut la première et continue à être la principale base des règles d'apprentissage de réseaux de neurones. A partir de cette première règle, d'autres règles plus complexes ont été développées. Ainsi, afin de permettre à des groupes de neurones d'une même couche de s'auto-organiser, on ajoute des liaisons inhibitrices qui permettent de n'avoir qu'un neurone actif dans le groupe [LIP87]. Cette interaction du type *Winner take all* est presque toujours associée à la règle de Hebb.

Contrairement au cerveau, où l'apprentissage se fait toujours de manière autonome, certains réseaux nécessitent un opérateur extérieur. On parle alors d'apprentissage supervisé. L'apprentissage d'un réseau peut être réalisé de différentes façons [Moutier92].

A.2.3.1 L'apprentissage supervisé

On soumet au réseau un modèle en entrée et l'on compare alors l'erreur obtenue entre la sortie obtenue et la sortie désirée. L'apprentissage supervisé consiste à ajuster les poids de chaque connexion, à l'aide d'un algorithme itératif, jusqu'à ce que l'erreur soit nulle ou minimale. Le réseau doit être stable lorsque l'apprentissage est réalisé. Il faut donc s'assurer que l'erreur minimale entraîne la stabilité du réseau.

A.2.3.1 L'apprentissage non supervisé

On ne soumet pas le réseau à une influence extérieure. Il s'organise lui-même en développant une sélectivité aux caractéristiques des vecteurs d'entrée.

A.2.3.1 L'apprentissage par stockage de l'information

Dans ce cas, le réseau ne traite pas une information, il joue le rôle de mémoire distribuée et associative, les poids synaptiques contiennent les informations à stocker.

Annexe B :

Quelques Exemples
Base d'apprentissage
Base de test
§IV.3.1.1

Annexe B : Quelques exemples de caractères
pour les bases d'apprentissage et de test (§IV.3.1.1)

A	A	A	A	A	A	A	A	A	A	A	A
A.tga	Aa.tga	AC0201.tga	AC0202.tga	AC0210.tga	AC0402.tga	AC0405.tga	AC0701.tga	AC0708.tga	AC0709.tga	acora.tga	acorb.tga
B	B	B	B	B	B	B	B	B	B	B	B
b.tga	Bar.tga	BC0201.tga	BC0202.tga	BC0210.tga	BC0402.tga	BC0405.tga	BC0701.tga	BC0708.tga	BC0709.tga	bocra.tga	boorb.tga
C	C	C	C	C	C	C	C	C	C	C	C
c.tga	car.tga	CC0201.tga	CC0202.tga	CC0210.tga	CC0402.tga	CC0405.tga	CC0701.tga	CC0708.tga	CC0709.tga	ccora.tga	ccorb.tga
D	D	D	D	D	D	D	D	D	D	D	D
d.tga	Dar.tga	DC0201.tga	DC0202.tga	DC0210.tga	DC0402.tga	DC0405.tga	DC0701.tga	DC0708.tga	DC0709.tga	dcora.tga	dcorb.tga
E	E	E	E	E	E	E	E	E	E	E	E
e.tga	Ear.tga	EC0201.tga	EC0202.tga	EC0210.tga	EC0402.tga	EC0405.tga	EC0701.tga	EC0708.tga	EC0709.tga	ecora.tga	ecorb.tga
F	F	F	F	F	F	F	F	F	F	F	F
f.tga	Far.tga	FC0201.tga	FC0202.tga	FC0210.tga	FC0402.tga	FC0405.tga	FC0701.tga	FC0708.tga	FC0709.tga	fcora.tga	fcorb.tga
G	G	G	G	G	G	G	G	G	G	G	G
g.tga	Gar.tga	GC0201.tga	GC0202.tga	GC0210.tga	GC0402.tga	GC0405.tga	GC0701.tga	GC0708.tga	GC0709.tga	gocra.tga	gocrb.tga
H	H	H	H	H	H	H	H	H	H	H	H
h.tga	Har.tga	HC0201.tga	HC0202.tga	HC0210.tga	HC0402.tga	HC0405.tga	HC0701.tga	HC0708.tga	HC0709.tga	hocra.tga	hocrb.tga
I	I	I	I	I	I	I	I	I	I	I	I
i.tga	Iar.tga	IC0201.tga	IC0202.tga	IC0210.tga	IC0402.tga	IC0405.tga	IC0701.tga	IC0708.tga	IC0709.tga	icora.tga	icorb.tga
J	J	J	J	J	J	J	J	J	J	J	J
j.tga	Jar.tga	JC0201.tga	JC0202.tga	JC0210.tga	JC0402.tga	JC0405.tga	JC0701.tga	JC0708.tga	JC0709.tga	jcora.tga	jcorb.tga
K	K	K	K	K	K	K	K	K	K	K	K
k.tga	Kar.tga	KC0201.tga	KC0202.tga	KC0210.tga	KC0402.tga	KC0405.tga	KC0701.tga	KC0708.tga	KC0709.tga	kocra.tga	koorb.tga
L	L	L	L	L	L	L	L	L	L	L	L
l.tga	Lar.tga	LC0201.tga	LC0202.tga	LC0210.tga	LC0402.tga	LC0405.tga	LC0701.tga	LC0708.tga	LC0709.tga	locra.tga	loorb.tga
M	M	M	M	M	M	M	M	M	M	M	M
m.tga	Mar.tga	MC0201.tga	MC0202.tga	MC0210.tga	MC0402.tga	MC0405.tga	MC0701.tga	MC0708.tga	MC0709.tga	mcora.tga	mcorb.tga
N	N	N	N	N	N	N	N	N	N	N	N
n.tga	Nar.tga	NC0201.tga	NC0202.tga	NC0210.tga	NC0402.tga	NC0405.tga	NC0701.tga	NC0708.tga	NC0709.tga	ncora.tga	ncorb.tga
O	O	O	O	O	O	O	O	O	O	O	O
o.tga	Oar.tga	OC0201.tga	OC0202.tga	OC0210.tga	OC0402.tga	OC0405.tga	OC0701.tga	OC0708.tga	OC0709.tga	ocora.tga	ocorb.tga
P	P	P	P	P	P	P	P	P	P	P	P
p.tga	Par.tga	PC0201.tga	PC0202.tga	PC0210.tga	PC0402.tga	PC0405.tga	PC0701.tga	PC0708.tga	PC0709.tga	pcora.tga	pcorb.tga

Annexe B : Quelques exemples de caractères
pour les bases d'apprentissage et de test (§IV.3.1.1)

Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
q.tga	Qa.tga	QC0201.tga	QC0202.tga	QC0210.tga	QC0402.tga	QC0405.tga	QC0701.tga	QC0708.tga	QC0709.tga	Qcra.tga	Qcra.tga
R	R	R	R	R	R	R	R	R	R	R	R
r.tga	Ra.tga	RC0201.tga	RC0202.tga	RC0210.tga	RC0402.tga	RC0405.tga	RC0701.tga	RC0708.tga	RC0709.tga	Rcra.tga	Rcra.tga
S	S	S	S	S	S	S	S	S	S	S	S
s.tga	Sa.tga	SC0201.tga	SC0202.tga	SC0210.tga	SC0402.tga	SC0405.tga	SC0701.tga	SC0708.tga	SC0709.tga	Scra.tga	Scra.tga
T	T	T	T	T	T	T	T	T	T	T	T
t.tga	Ta.tga	TC0201.tga	TC0202.tga	TC0210.tga	TC0402.tga	TC0405.tga	TC0701.tga	TC0708.tga	TC0709.tga	Tcra.tga	Tcra.tga
U	U	U	U	U	U	U	U	U	U	U	U
u.tga	Ua.tga	UC0201.tga	UC0202.tga	UC0210.tga	UC0402.tga	UC0405.tga	UC0701.tga	UC0708.tga	UC0709.tga	Ucra.tga	Ucra.tga
V	V	V	V	V	V	V	V	V	V	V	V
v.tga	Va.tga	VC0201.tga	VC0202.tga	VC0210.tga	VC0402.tga	VC0405.tga	VC0701.tga	VC0708.tga	VC0709.tga	Vcra.tga	Vcra.tga
W	W	W	W	W	W	W	W	W	W	W	W
w.tga	Wa.tga	WC0201.tga	WC0202.tga	WC0210.tga	WC0402.tga	WC0405.tga	WC0701.tga	WC0708.tga	WC0709.tga	Wcra.tga	Wcra.tga
X	X	X	X	X	X	X	X	X	X	X	X
x.tga	Xa.tga	XC0201.tga	XC0202.tga	XC0210.tga	XC0402.tga	XC0405.tga	XC0701.tga	XC0708.tga	XC0709.tga	Xcra.tga	Xcra.tga
Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
y.tga	Ya.tga	YC0201.tga	YC0202.tga	YC0210.tga	YC0402.tga	YC0405.tga	YC0701.tga	YC0708.tga	YC0709.tga	Ycra.tga	Ycra.tga
Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
z.tga	Za.tga	ZC0201.tga	ZC0202.tga	ZC0210.tga	ZC0402.tga	ZC0405.tga	ZC0701.tga	ZC0708.tga	ZC0709.tga	Zcra.tga	Zcra.tga

Annexe C :

Quelques Exemples
Base de test
§V.1.3.

Annexe C : Exemples de polices TrueTypes utilisées
pour les bases d'apprentissage et de test (§V.1.3.1.)

Abadi MT Condensed	ABCDEFGHIJKLMN OPQRSTUVWXYZ
AdLib BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Airbus Special	ABCDEFGHIJKLMN OPQRSTUVWXYZ
<i>Aldine401 BT</i>	<i>ABCDEFGHIJKLMN OPQRSTUVWXYZ</i>
Aldine721 BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Aldine721 Lt BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Americana BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
AmeriGarmnd BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
<i>Amerigo BT</i>	<i>ABCDEFGHIJKLMN OPQRSTUVWXYZ</i>
Amerigo Md BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Architecture	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Arial	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Arial Black	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Arial Narrow	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BANKGOTHIC LT BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BANKGOTHIC Md BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BauerBodni Blk BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BauerBodni BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BAUERBODNI TITL	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bauhous	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bauhaus Hv BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bauhaus Lt BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bauhaus Md BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BellCent Add BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BELLCENT BdList BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BellCent NamNum BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Belwe Bd BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
<i>Benguiat</i>	<i>ABCDEFGHIJKLMN OPQRSTUVWXYZ</i>
Benguiat Bk BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Benguiat-Bold	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BOBIDLYhold	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bodoni	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bodoni BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bodoni-Bold	ABCDEFGHIJKLMN OPQRSTUVWXYZ
<i>Bodoni-Normal-Italic</i>	<i>ABCDEFGHIJKLMN OPQRSTUVWXYZ</i>
Bodoni-Poster	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bodony	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BOINK LET	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Book Antiqua	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bookman	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bookman Old Style	ABCDEFGHIJKLMN OPQRSTUVWXYZ
BookmanITC Lt BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Bookplate	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Borzoi	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Boulder	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Brassfield	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Carmina Lt BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Carmina Md BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
CastleT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
CastleTlig	ABCDEFGHIJKLMN OPQRSTUVWXYZ
CastleUlt	ABCDEFGHIJKLMN OPQRSTUVWXYZ
CenturyOldst BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
CG Times	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Cooper Md BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
COPPERPLATE GOTH	ABCDEFGHIJKLMN OPQRSTUVWXYZ
COPPERPLATE GOTH	ABCDEFGHIJKLMN OPQRSTUVWXYZ
COPPRPLGOTH BdCh BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
COPPRPLGOTH Hv BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Courier	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Courier New	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Courier10 BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Eras Bk BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Flareserif821 BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Flareserif821 Lt BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Folio Md BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ
Folio XBd BT	ABCDEFGHIJKLMN OPQRSTUVWXYZ

Annexe C : Exemples de polices TrueTypes utilisées
pour les bases d'apprentissage et de test (§V.1.3.1.)

Galliard BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Garamond	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
GarmdITC Bk BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
GarmdITC BkCn BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Geometr212 Bk BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Geometr706 BkCn BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Geometr706 Md BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
GeoSlab703 Lt BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Humnst777 Bk BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Humnst777 BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
JuniusModern	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
KEYFUNCH	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Latin725 Md BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
LinePrinter	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
MACHINE BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Modern20 BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Modern880 BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
<i>NewBskell BT</i>	<i>A B C D E F G H I J K L M N O P Q R S T U V W X Y Z</i>
News701 BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
<i>News706 BT</i>	<i>A B C D E F G H I J K L M N O P Q R S T U V W X Y Z</i>
<i>NewsGoth BT</i>	<i>A B C D E F G H I J K L M N O P Q R S T U V W X Y Z</i>
NewsGoth Cn BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
NewsGoth Dm BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
NewsGoth XcN BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Newtext Rg BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Novarese Bk BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
OCR A Extended	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
OCR-B-10 BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Orator10 BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Orbit-B BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Raleigh Lt BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Raleigh Md BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Rockwell	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Rockwell Condensed	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
SerpentineDBol	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Square 721 Bk	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Square721 BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Square721 Cn BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Square721 Dm	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Square721 Ex	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
SquareSlab711 Bd BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
STENCIL BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Stymie Lt BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Stymie Md BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Tahoma	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Times New Roman	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Transit521 BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Transit551 Md BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Univers	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Verdana	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Village	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Zurich Cn BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Zurich Lt BT	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Annexe D :

Quelques Exemples
de Caractères Industriels

Annexe D : Quelques exemples de caractères industriels



Annexe D : Quelques exemples de caractères industriels



THESE pour obtenir le grade de **docteur de l'Université de LILLE I**; Discipline : Electronique

Soutenue le 20 Novembre 1998 par **Stephane Lecoeuche**

TITRE : *Reconnaissance de caractères industriels par application d'un système de réseaux de neurones à boucle de rétroaction*

JURY : Mrs. **Attila Baskurt**, **Léon-Claude Calvez**, Rapporteurs;
Mrs. **Vincent Devlaminck**, **M. Salah Maouche**, Examineurs;
M. Jean Paul Dubus, Directeur de thèse;

RESUME :

La reconnaissance de caractères en milieu industriel est confrontée aux problèmes de la variété des supports utilisés (métal, bois, papier,..) et du type d'écriture (jet d'encre, gravure, ..). L'étude présentée porte sur la conception d'un système de reconnaissance de caractères qui possède les propriétés d'autoadaptation nécessaires à la résolution de ces problèmes.

Le système proposé est élaboré à partir de deux réseaux de neurones et d'une boucle de rétroaction. Le premier réseau permet d'extraire un modèle binarisé invariant quels que soient le matériau et le type de marquage et quelles que soient la position, la taille et l'orientation du caractère. Il est construit suivant une architecture à couches structurées décrivant une opération de vision et une opération de normalisation. Un second réseau supervisé du type RBF réalise la classification de ce modèle. Ce réseau a été conçu pour permettre et faciliter la reconnaissance multifont en prenant compte de la variété et de la spécificité des polices utilisées en marquage industriel. En fonction de la qualité de reconnaissance, le système peut grâce à la boucle de rétroaction, apporter au premier réseau l'information nécessaire à l'amélioration de la vision des caractères.

La première partie est une introduction à la reconnaissance de caractères et aux systèmes d'OCR, nous présentons ensuite en deuxième partie une bibliographie sur les réseaux de neurones utilisés en classification et en troisième partie quelques réseaux de neurones utilisés en reconnaissance invariante. Dans une quatrième partie, nous présentons l'architecture de notre système et détaillons les concepts originaux tels que la notion de centre/voisinage développée dans l'étape de vision, l'architecture de normalisation et la notion de forme introduite dans la phase d'apprentissage du réseau RBF. La cinquième partie est consacrée à l'analyse des performances du système appliqué à des cas réels.

MOTS CLES :

Reconnaissance de Formes; Reconnaissance de Caractères; Réseaux de Neurones;
Classification; Invariance; Autoadaptation

ABSTRACT :

Character recognition in an industrial environment is a complex problem. This is due to the variety of supports (metal, wood, paper, ...) and/or to the different types of writing (inkjet, engraving, ...). This study presents the conception of a self adaptive recognition system that is necessary to solve a part of this problem.

This system is based on two neural networks and a feedback loop. The first network is a connectionist network that consists of layers that ensure vision and normalization operations. Whatever the support and the type of writing, and whatever the position, the size and the orientation of the character, it extracts an invariant binaryzed model. This model is classified by the second network which is a supervised RBF network. This network has been designed to allow and make easy the multifont recognition, taking into account the variety and specificity of the industrial fonts. Depending on the quality of the recognition, the second network passes on, through the feedback loop, information to the first network that needs it to improve the vision of the character.

The first part of the study is an introduction to character recognition and to OCR systems. The second part is a bibliography that presents neural networks that are used for classification. The third part presents some neural networks that are used for invariant recognition. In the fourth part, we present the global architecture of our system and we give a detailed description of the new concepts we have introduced such as the center/neighborhood notion (introduced in the vision layer), the normalization architecture, and the notion of shape that we have developed for the learning stage of the second network. The last part is dedicated to the analysis of the performances of the system applied to actual characters.

KEYWORDS :

Pattern Recognition; Character Recognition; Neural Networks
Classification; Invariance; Autoadaptation