

République algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Mohamed Khider Biskra
Faculté des sciences et sciences de l'ingénieur
Département d'informatique
Ecole doctorale de l'est

Mémoire présenté Pour l'obtention du diplôme de

Magister en informatique

Option

INTELLIGENCE ARTIFICIELLE

par

KAMEL Mohamed

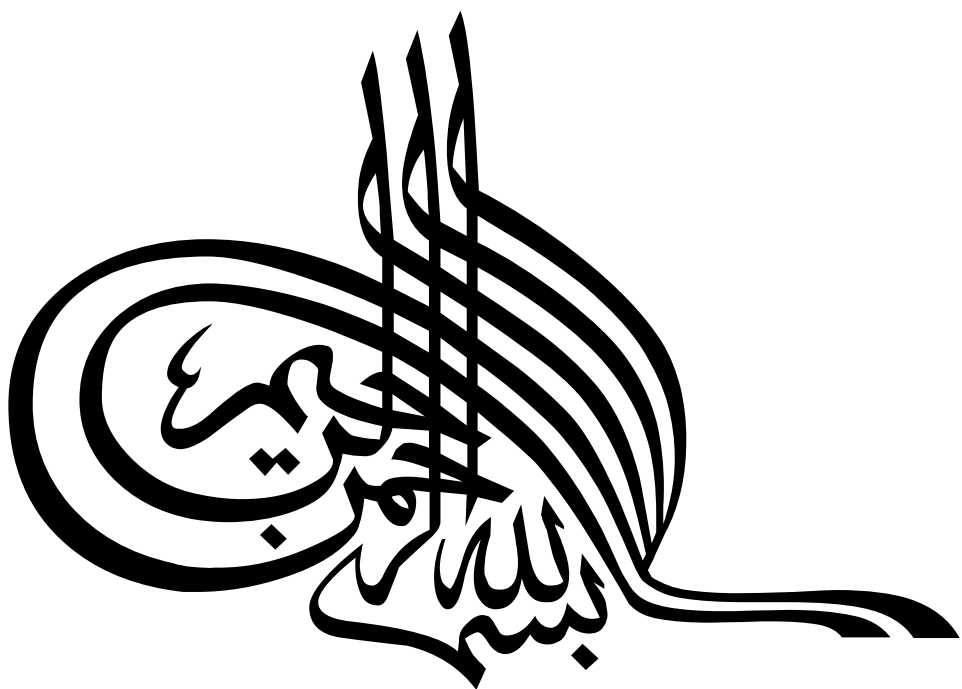
Thème

**RECONNAISSANCE DE FORMES
APPLIQUEE A L'ECRITURE ARABE MANUSCRITE
PAR DES MULTICLASSIFIEURS**

Devant le jury composé de :

Président :	Pr BOUFAIDA Mahmoud	Professeur, Université de Constantine
Rapporteur :	Dr FARAH Nadir	Maître de conférences, Université de Annaba
Examineurs :	Dr ZAROUR Nacereddine	Professeur, Université de Constantine
	Dr KAZAR Okba	Maître de conférences, Université de Biskra

Soutenu le : 06/06/2010



RESUME :

Le présent travail porte sur une étude concernant le domaine de reconnaissance de formes appliqué sur l'écriture arabe manuscrite par des multiclassifieurs, D'abord il s'agit de faire une étude générale sur la reconnaissance de formes, puis de faire une étude bibliographique sur les systèmes existants et les différentes recherches effectuées sur ce domaine, ensuite de faire une étude sur les caractéristiques morphologiques et structurelles de l'écriture Arabe, puis étudier les systèmes de classification couramment utilisés, ainsi que des concepts de bases des combinaisons parallèles des classifieurs. Pour enfin proposer un système multiclassifieur de reconnaissance de mots arabes dans un lexique défini.

MOTS-CLES : Reconnaissance de mots arabes, écriture manuscrite, extraction de caractéristiques, Multiclassifieurs, RNA, KPPV.

REMERCIEMENTS

Je tiens à remercier l'initiateur de ce projet pour toute sa sympathie, sa disponibilité, et son soutien, je veux bien sûr parler du Mr FARAH Nadir.

Je remercie ma famille et particulièrement mes chers parents, que dieu les protège, pour leur amour, leurs encouragements et leurs sacrifices, ainsi que mes amis.

Je joins à ces remerciements le président du jury Mr BOUFAIDA Mahmoud et les deux examinateurs Mr KAZAR Okba et Mr ZAROUR Nacereddine pour avoir accepté de faire partie du jury et pour leur lecture minutieuse et avisée des nombreuses pages qui suivent.

TABLE DES MATIERES

TABLE DES MATIERES

Introduction générale	2
Chapitre I : La reconnaissance de formes	
1. Introduction	5
2. La reconnaissance d'Objets	5
3. Similarité des formes et les taches de RF	8
3.1. Taches de Classification	8
3.2. Taches de Régression	12
3.3. Taches de Description	13
4. Classes, Formes et Caractéristiques	16
4.1. Les Classes	17
4.2. Les Formes	17
4.3. Les Caractéristiques	18
5. Les Approches de la RF	19
5.1. Groupement de donnees (Data Clustering)	20
5.2. Classification statistique (RF statistique)	21
5.3. Les Réseaux neuronaux	21
5.4. Reconnaissance de formes Structurale	22
6. Projet de RF	23
6.1. Taches du Projet	23
6.2. Apprentissage et Test	25
6.3. Les logiciels de RF	26
7. Conclusion	27
Chapitre II : La reconnaissance d'écriture arabe	
1. Introduction	29
2. Les phases d'un système de reconnaissance d'écriture	29
2.1. Acquisition d'image	29
2.1.1. Acquisition hors ligne	29
2.1.1. Acquisition en ligne	29
2.2. Prétraitements et Normalisations	30
2.2.1. Seuillage	30
2.2.1.1. Seuillage global	30
2.2.1.2. Seuillage adaptatif	31
2.2.2. Squelette	31
2.2.3. Normalisations	34

2.2.3.1.	Correction de l'inclinaison des lignes	34
2.2.3.2.	Correction de l'inclinaison des lettres.....	35
2.2.3.3.	Lissage du contour	35
2.2.3.4.	La ligne de base	36
2.3.	Segmentation.....	36
2.3.1.	Segmentation de la page.....	37
2.3.2.	Segmentation d'un bloc de texte en lignes.....	37
2.3.3.	Segmentation des lignes en mots	37
2.3.4.	Segmentation des mots.....	37
2.4.	Extraction de caractéristiques	38
2.4.1.	Les primitives perceptuelles	38
2.4.2.	Les primitives directionnelles.....	38
2.4.3.	Les primitives topologiques	38
2.5.	Reconnaissance	39
2.6.	Post-Traitements	39
3.	La reconnaissance de l'écriture Arabe manuscrite.....	39
3.1.	Problématique	39
3.1.1.	Présentation de la langue arabe.....	39
3.1.2.	Difficultés inhérentes à la reconnaissance de l'écriture arabe.....	40
3.1.2.1.	Alphabet	40
3.1.2.2.	Signes diacritiques	40
3.1.2.3.	Ascendants et descendants	43
3.1.2.4.	Une ou plusieurs composantes connexes par mot	43
3.1.2.5.	Ligatures verticales	43
3.2.	Etude bibliographique	44
3.2.1.	Principales bases de données existantes.....	44
3.2.1.1.	Kharma/Ahmed/Ward 1999	45
3.2.1.2.	AHDB	45
3.2.1.3.	CENPARMI	45
3.2.1.4.	IFN/ENIT	46
3.2.1.5.	CEDARABIC	46
3.2.1.6.	Applied Media Analysis : Arabic-Handwriting-1.0	47
3.2.2.	Différentes approches et systèmes existants.....	47
3.2.2.1.	Système d'Amin et al. 2000 (Approche globale)	47
3.2.2.2.	Système de Al Ohali, 2002 (Approche pseudo-analytique).....	48
3.2.2.3.	Système de Pechwitz et al. 2003 (Approche analytique).....	48
3.2.2.4.	Système de Khorsheed, 2003 (Approche analytique).....	49

3.2.2.5. Système de Farah et al. 2005 (Approche globale).....	49
4. Conclusion	50
Chapitre III : Les systèmes de classification	
1. Introduction.....	52
2. Définition formelle d'un classifieur	52
2.1. Classifieur de type classe.....	53
2.2. Classifieur de type rang	53
2.3. Classifieur de type mesure.....	53
3. Les Réseaux de Neurones.....	54
3.1. Introduction.....	54
3.2. Le Neurone Simple.....	54
3.2.1. Le neurone de Mac Culloch-Pitts	54
3.2.2. Le Perceptron.....	55
3.3. Interprétation mathématique	55
3.4. Types et Architectures des RNA.....	57
3.4.1. Le Perceptron Multicouches.....	57
3.4.1.1. La Propagation.....	58
3.4.2. Fonctions à bases radiales	59
3.5. L'Apprentissage	59
3.5.1. La loi d'apprentissage de Hebb [HEB49]	60
3.5.2. La règle d'apprentissage du Perceptron.....	60
3.5.3. Retro-propagation du gradient.....	61
3.6. Avantages et inconvénients des réseaux neuronaux	61
3.6.1. Avantages de l'approche neuronale.....	61
3.6.2. Inconvénients de l'approche neuronale	63
4. K Plus Proches Voisins.....	64
4.1. Introduction.....	64
4.2. Formalisation mathématique.....	64
4.3. Algorithme	65
5. Modèles de Markov Cachés.....	65
5.1. Introduction.....	65
5.2. Définitions	66
5.2.1. Chaîne de Markov.....	66
5.2.2. Chaîne stationnaire.....	66
5.3. Un Modèle de Markov Caché	67
5.4. Interprétation Mathématique.....	67
5.5. Les Types des HMMs.....	67

5.5.1. HMM ergodique	68
5.5.2. HMM gauche-droite.....	68
6. Conclusion	69

Chapitre IV : La combinaison des systèmes de classification

1. Introduction.....	71
2. Stratégies de combinaison	71
2.1. Approche séquentielle	71
2.2. Approche parallèle	72
2.3. Approche hybride.....	73
3. Taxonomies des méthodes de combinaison.....	75
3.1. Combinaison de classifieurs de type classe	75
3.1.1. Vote avec seuil.....	75
3.1.2. Majorité sans conflit.....	75
3.1.3. Majorité notoire	75
3.1.4. Vote avec pondération.....	76
3.2. Combinaison de classifieurs de type rang	76
3.2.1. Borda Count et ses variantes	76
3.2.2. Meilleur rang	77
3.2.3. Somme pondérée.....	77
3.2.4. Intersection	78
3.2.5. Union.....	79
3.3. Combinaison de classifieurs de type mesure	79
3.3.1. Méthodes fixes.....	79
3.3.1.1. La règle maximum	80
3.3.1.2. La règle minimum.....	80
3.3.1.3. La règle médiane.....	80
3.3.1.4. La règle produit.....	80
3.3.1.5. La règle linéaire	80
3.3.2. Les règles pondérées	81
3.3.3. Les réseaux de neurones.....	81
4. Conclusion	82

Chapitre V : Travail proposé et conclusion

1. Introduction.....	84
2. Base de mots et outils de développement.....	84
2.1. Base de Mots.....	84
2.2. Outils	85
2.2.1. Turbo Delphi 2006.....	85

2.2.2. Mathworks MATLAB.....	85
3. Les phases du système proposé.....	86
3.1. Acquisition D'image	86
3.2. Prétraitement	86
3.2.1. Seuillage	86
3.2.2. Squelettisation.....	87
3.2.3. Détection de la ligne de base	87
3.3. Segmentation.....	88
3.3.1. Segmentation en pages, lignes, mots et lettres	88
3.3.2. Segmentation en graphèmes	88
3.4. Extraction de caractéristiques	89
3.4.1. Nombres des ascendants et des descendants	89
3.4.2. Nombres des points hauts et des points bas	89
3.4.3. Nombres des points doubles hauts et des points doubles bas.....	89
3.4.4. Nombres des points triples hauts	89
3.4.5. Nombres des boucles	90
3.4.6. Nombres des composants connexes.....	90
3.4.7. Nombres des graphèmes.....	90
3.5. Apprentissage et test.....	90
3.5.1. Le premier classifieur.....	91
3.5.2. Le deuxième Classifieur.....	91
3.5.3. Le troisième classifieur	92
3.6. Combinaison	92
4. Conclusion	94
Conclusion générale	96
Bibliographie	97
Liste des tableaux.....	102
Liste des figures	103

INTRODUCTION GÉNÉRALE

INTRODUCTION GÉNÉRALE

Les techniques liées au traitement de l'information connaissent actuellement un développement très actif en liaison avec l'informatique et présentent un potentiel de plus en plus important dans le domaine de l'interface homme machine. L'écriture restera l'un des grands fondements des civilisations et le mode par excellence de conservation et de transmission du savoir. La reconnaissance de l'écriture est un domaine vaste qui constitue un sous ensemble des systèmes de la reconnaissance des formes (RF). Ces systèmes sont la première étape d'un processus de compréhension de notre univers dans le cadre global de la communication homme machine. La reconnaissance de l'écriture manuscrite ou imprimée reste encore un sujet de recherche et d'expérimentation, le problème n'est pas encore entièrement résolu bien que l'on sache atteindre des taux assez élevés dans certaines applications pour lesquelles soit le vocabulaire est limité, soit la fonte est unique ou en nombre restreint.

Il existe cependant plusieurs domaines pour lesquels la reconnaissance de l'écriture est appliquée avec un certain succès : le tri automatique du courrier, le traitement automatique de dossiers administratifs, des formulaires d'enquêtes, ou encore l'enregistrement des chèques bancaires. La reconnaissance de l'écriture manuscrite est beaucoup plus complexe que celle de l'écriture imprimée due à son extrême variabilité: variabilité des formes, des espacements entre mots et caractères, fluctuation des lignes.

Les travaux de recherche en reconnaissance de l'écriture arabe manuscrite, bien que moins avancés que pour d'autres langues, deviennent plus intensifs qu'avant.

Ce mémoire va s'intéresser à la combinaison des classifieurs pour la reconnaissance des mots arabes manuscrits, où l'objectif est de proposer un système de reconnaissance de l'écriture arabe manuscrite hors-ligne hybride à base de Réseaux de Neurones de type Perceptron Multi Couches et de l'algorithme K-Plus Proches Voisins.

Après cette introduction, dans le premier chapitre, nous présenterons rapidement les notions de bases de la reconnaissance de formes ainsi que les grandes étapes qui composent une chaîne de reconnaissance d'une forme.

Dans le deuxième chapitre, nous nous intéresserons plus particulièrement à l'écriture arabe et aux travaux effectués dans le domaine de la reconnaissance automatique de l'écriture arabe manuscrite.

Dans le troisième chapitre, nous présenterons un ensemble de classifieurs couramment utilisés en reconnaissance de l'écriture.

Dans le quatrième chapitre, nous étudierons les systèmes multi-classifieurs et les stratégies de combinaison, et plus particulièrement les stratégies de combinaison parallèles des classifieurs.

Dans le cinquième chapitre, nous présenterons notre travail sur la reconnaissance de l'écriture arabe manuscrite, L'originalité de notre travail consiste à construire deux classifieurs neuronaux et un autre basé sur l'algorithme K-Plus Proches Voisins, avant d'utiliser une stratégie de combinaison pour combiner les trois.

CHAPITRE I

LA RECONNAISSANCE DE FORMES

Dans ce chapitre:

- ✓ La Reconnaissance d'objets.
- ✓ Classes, Formes et Caractéristiques.
- ✓ Similarité des Formes et les Taches de RF.
- ✓ Les Approches de la RF.
- ✓ Projet de RF.

1. INTRODUCTION

La reconnaissance de formes comporte actuellement un champ très large des méthodes soutenant le développement de nombreuses applications dans beaucoup de différents secteurs d'activités. Les méthodes et techniques de reconnaissance de formes se trouvent généralement dans le milieu de la simulation « intelligente » des tâches, qui a certainement infiltré notre vie quotidienne [MAR01]. La fabrication aidée par les robots, les systèmes de diagnostic médical, la reconnaissance automatique des documents ou caractères (OCR), la prévision des variables économiques, l'exploration des ressources terrestres, et l'analyse des données satellitaires sont juste quelques exemples des champs d'activités où cette tendance s'applique.

Alors dans ce chapitre nous allons présenter l'ensemble des généralités sur le domaine de reconnaissance de formes. Pour commencer, nous introduisons la reconnaissance d'objets, poursuivons par présenter les classes, formes et les caractéristiques puis les approches de RF, puis Similarité des Formes et les Taches de RF et finalement les problèmes pratiques importants.

2. LA RECONNAISSANCE D'OBJETS

La reconnaissance des objets est une tâche effectuée quotidiennement par les êtres vivants et est inhérente à leur aptitude et à la nécessité de traiter l'environnement. Il est interprété dans la plupart des circonstances (la navigation vers les sources de nourriture, de la migration, l'identification des prédateurs, la détermination des camarades, etc.) avec une efficacité remarquable. La reconnaissance des objets est considérée ici dans un sens large cognitif et peut consister en une tâche très simple, comme quand un micro-organisme fuit d'un environnement avec un inadéquate pH, ou se référer à des tâches exigeantes des qualités non triviales d'inférence, d'un terme de la description et interprétation, par Exemple quand un homme doit récupérer une paire de ciseaux à partir de deuxième tiroir d'une armoire et compter à partir de bas.

Le développement des méthodes capables de simuler la variété des formes de la reconnaissance d'objets a évolué en même temps que la nécessité de construire des systèmes automatisés « intelligents », la tendance principale de la technologie d'aujourd'hui dans l'industrie et dans d'autres domaines d'activité aussi. Dans ces systèmes, les objets sont représentés d'une manière adéquate pour le type de traitement qu'ils sont soumis. De telles

représentations sont appelées *formes*. Dans ce qui suit nous allons utiliser les mots *objet* et *forme* avec la même signification.

La Reconnaissance de Formes (RF) est une discipline scientifique portant sur les méthodes de description et de classification des objets. Depuis le début du temps de l'informatique, la conception et l'implémentation des algorithmes émulant la capacité humaine pour décrire et classifier les objets ont été trouvées les plus intrigantes et difficiles des tâches. La reconnaissance de formes est donc une région fertile de la recherche, avec plusieurs liens vers de nombreuses autres disciplines, associant des professionnels à partir de plusieurs domaines [MAR01].

Selon [HAT90] :

« La reconnaissance de formes s'intéresse à la conception et à la réalisation de systèmes (matériels et logiciels) capables de percevoir, et dans une certaine mesure, d'interpréter des signaux captés dans le monde physique ».

Les applications des systèmes et des techniques de la reconnaissance de formes sont nombreuses et couvrent un champ d'activités large. Nous énumérons seulement quelques exemples se référant à quelques activités professionnelles:

Agriculture :

- Analyse de récolte
- Évaluation des sols

Astronomie :

- L'analyse des images de télescopique
- Spectroscopie automatisée

Biologie:

- Cytologie automatisée
- Propriétés de chromosomes
- Les études génétiques

L'administration civile:

- Analyse et contrôle du trafic
- Evaluation de la croissance urbaine

Économie:

- Prévisions des stocks
- Analyse de la performance d'entreprise

Ingénierie:

- Détection des pannes dans les produits manufacturés
- Reconnaissance de caractères
- Reconnaissance de parole
- Systèmes de navigation automatiques
- Analyse de pollution

Géologie:

- Classification des roches
- Estimation des ressources minières
- L'analyse des ressources géographiques en utilisant des images satellites
- Analyse sismique

Médecine:

- Analyse des électrocardiogrammes
- Analyse des électroencéphalogrammes
- L'analyse des images médicales

Militaire:

- Analyse de la photographie aérienne
- La détection et la classification des signaux radar et sonar,
- Reconnaissance automatique de cible

Sécurité:

- L'identification des empreintes digitales
- Surveillance et systèmes d'alarme

Comme on peut le déduire de ces exemples, les formes à analyser et reconnaître peuvent être des signaux (ex. des signaux électro cardiographiques), des images (ex. des photos aériennes) ou des tableaux de valeurs (ex. Les taux d'échange des stocks).

Pour une excellente référence sur la reconnaissance de formes voir le livre de Duda [DUD01].

3. SIMILARITE DES FORMES ET LES TACHES DE RF

Selon [MAR01], la *Similarité* est une notion fondamentale dans la reconnaissance de formes, indépendamment de quelle approche nous allons suivre. Nous reconnaissons que deux objets sont similaires s'ils ont des valeurs des attributs communs similaires. Souvent, la similarité est liée à un sens plus abstrait, non pas entre des objets, mais entre un objet et un *concept* cible. Par exemple, nous reconnaissons un objet comme une pomme car il correspond, dans ses caractéristiques, à l'image, le concept ou *le prototype* idéalisée, nous avons peut-être d'une pomme, c'est-à-dire, l'objet est similaire à ce concept et dissimilaire aux autres, par exemple dissimilaire avec une orange.

L'appréciation sur la similarité des formes est fortement liée à la tâche de reconnaissance des formes proposée comme décrit dans ce qui suit.

3.1. TACHES DE CLASSIFICATION

Quand évaluons la similarité entre des objets on recourt aux caractéristiques ou aux attributs qui sont de nature discriminante. Imaginons que nous allons concevoir un système de discrimination des pommes vertes à partir des oranges. La Figure I.1 illustre les représentations possibles des prototypes «pomme verte» et «orange». Dans cette tâche de discrimination, nous pouvons utiliser comme caractéristiques discriminantes et évidentes la couleur et la forme, représentées dans une manière appropriée.

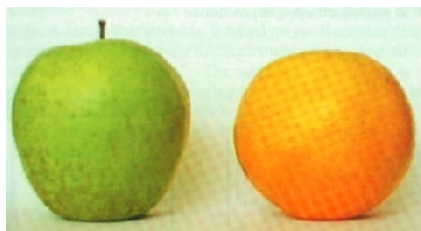


Figure I.1 : Représentations possibles des prototypes «pomme verte» et «orange».



Figure I.2 : Exemples de «pomme rouge» et «orange vert» à être caractérisés par les caractéristiques forme et couleur.

Afin d'obtenir une représentation numérique de la caractéristique « couleur », on peut commencer par diviser l'image des objets en sous couleurs rouge-vert-bleu. Ensuite on peut, par exemple, sélectionner une région centrale d'intérêt dans l'image et calculer, pour cette région, le rapport des emplacements maximaux de l'histogramme pour les sous couleurs rouge et vert (habituellement $[0, 255]$; 0 =pas de couleur, 255 =couleur pleine). La Figure I.3 montre l'image de niveau de gris correspondant à la sous couleur « vert » de la pomme et l'histogramme de l'intensité de la lumière pour une région rectangulaire d'intérêt. Le maximum de l'histogramme correspond à 186, Cela signifie que la valeur d'intensité de « vert » la plus fréquente est 186. Pour la couleur rouge, nous permettrait d'obtenir la valeur 150. Le rapport de ces valeurs est 1.24 révélant la prédominance de la couleur « vert » par rapport à la couleur rouge.

Afin d'obtenir une représentation numérique pour la caractéristique de la forme, nous pouvons, par exemple, mesurer la distance, Absent du sommet, jusqu'à la largeur maximale de l'objet et diviser cette par la hauteur, c'est-à-dire, calculer le rapport x/h , avec x , h illustré à la Figure I.3a, Dans ce cas, $x/h = 0,37$. Notez que nous supposons que les objets sont dans une position verticale standard.

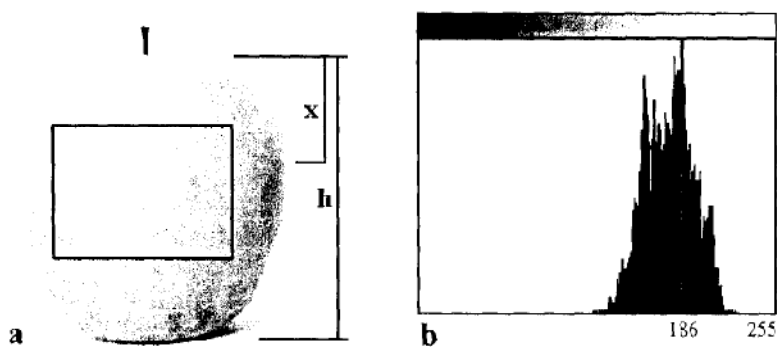


Figure I.3 : (a) Image grise de la composante verte de l'image pomme, (b) Histogramme de l'intensité de la lumière pour la région rectangulaire de l'intérêt montré à (a).

Si nous avons fait un choix judicieux de prototypes nous attendons que les représentations des échantillons des pommes vertes et des oranges correspondent aux grappes de points autour les prototypes dans l'espace bidimensionnel des caractéristiques, comme montré dans la Figure I.4a par les courbes représentant les frontières des grappes. Aussi, si nous avons fait un bon choix de caractéristiques, il est prévu que les groupes mentionnés sont raisonnablement séparés, ce qui permet la discrimination des deux classes de fruits.

La tâche de RF d'assigner un objet à une classe est dite une tâche de *classification*. D'un point de vue mathématique, il est pratique dans la classification de représenter une forme par un vecteur, qui est en 2 dimensions dans le cas présent:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \equiv \begin{bmatrix} \text{couleur} \\ \text{forme} \end{bmatrix}$$

Pour le prototype pomme verte, nous avons donc:

$$x_{\text{pomme verte}} = \begin{bmatrix} 1.24 \\ 0.37 \end{bmatrix}$$

Les points correspondant aux vecteurs des caractéristiques des prototypes sont représentés par un carré et un cercle, respectivement pour la pomme verte et l'orange, dans la Figure I.4.

Considérons une machine conçue pour séparer les pommes vertes des oranges en utilisant les caractéristiques décrites. Un morceau de fruit est présenté à la machine, ses caractéristiques sont calculées et correspondent à le point x (Figure I.4a) dans le plan couleur-forme. La machine, par l'utilisation des valeurs des caractéristiques comme entrées, doit alors décider s'il s'agit d'une pomme verte ou d'une orange. Une décision raisonnable est basée sur la distance euclidienne entre le point x et le prototype, c'est-à-dire, pour la machine, la similarité est une distance et dans ce cas elle déciderait « pomme verte ». La sortie de la machine est dans ce cas est l'une des deux valeurs, par exemple, 0 correspondant à la pomme verte et de 1 correspondant aux oranges. Cette machine s'appelle un classificateur.

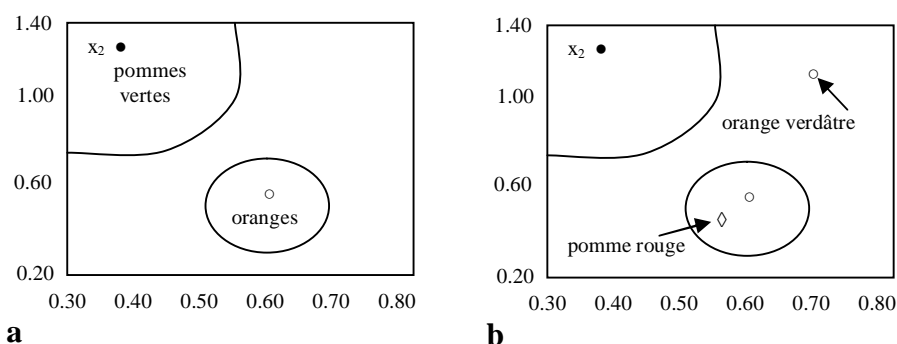


Figure I.4 : (a) des pommes vertes et des oranges dans l'espace des caractéristiques, (b) Une pomme rouge qui « ressemble » à une orange et un problème d'orange verdâtre.

Imaginons que notre classificateur reçoit comme entrées les caractéristiques de la pomme rouge et l'orange verte présentées dans la Figure I.2. Les vecteurs des caractéristiques correspondent aux points illustrés à la Figure I.4b. La pomme rouge est mal classée dans la

catégorie orange car elle est beaucoup plus proche du prototype « orange » que du prototype « pomme verte ». Ce n'est pas une surprise car, après tout, le classificateur est utilisé pour un objet clairement en dehors de son champ d'application. Concernant l'orange verdâtre, son vecteur des caractéristiques est presque à des distances égales des deux prototypes, et sa classification forme une problématique. Si nous utilisons, au lieu de la distance euclidienne, une autre mesure de distance qui donne un poids aux déviations verticales, supérieur que celle donnée aux déviations horizontales, l'orange verdâtre serait également mal classée.

Généralement dans la pratique les systèmes de classification de formes ne sont pas parfaits et nous pouvons nous attendre des erreurs dues à plusieurs causes:

- Les caractéristiques utilisées sont inadéquates ou insuffisantes. Par exemple, la classification de l'orange verdâtre serait probablement améliorée à l'aide d'une caractéristique supplémentaire de texture mesurant le degré de rugosité de surface.
- Les échantillons des formes utilisées pour la conception du classificateur ne sont pas suffisamment représentatifs. Par exemple, si nos intentions sont de discriminer les pommes des oranges nous devrions inclure dans les échantillons des pommes une variété représentative des pommes, y compris les rouges aussi.
- Le classificateur n'est pas assez efficace pour séparer les classes. Par exemple, une mesure inefficace des distances ou des prototypes inadéquats sont utilisés.
- Il y a un chevauchement des classes qu'aucun classificateur ne peut résoudre.

Concernant le choix initial des caractéristiques, il faut noter qu'il est plus un art qu'une science [MAR01], et comme dans tout art, il est amélioré par l'expérimentation et la pratique. En outre le choix approprié des caractéristiques et les mesures de similarité, il y a aussi d'autres aspects responsables du haut degré de précision de la classification chez les humains. Des aspects, tels que l'utilisation des informations contextuelles et des structures de connaissances avancées, s'appartiennent principalement au domaine de l'intelligence artificielle et ne seront pas traités dans ce travail. Même la reconnaissance humaine des objets n'est pas toujours parfaite et les informations contextuelles risquent de classer une orange verdâtre comme un citron si elle réside dans un panier de citrons.

3.2. TACHES DE REGRESSION

Nous considérons maintenant un autre type de tâche, directement liés au processus d'inférence cognitif. Nous observons un tel processus lorsque les animaux commencent une migration fondée sur les changements climatiques et les changements physiologiques internes de leurs cycles biologiques. Dans la vie quotidienne, l'inférence est un outil important car elle oriente l'optimisation de la décision. Des exemples bien connus sont, par exemple, de garder la bonne distance d'un véhicule quand conduire sur une route, la prévision des conditions météorologiques, la prévision des retenues d'une entreprise fondés sur des variables économiques.

Prenons un exemple qui consiste à prévoir la valeur d'action boursière d'une entreprise A dans le marché, basé sur l'information passé sur: les valeurs des actions de l'entreprise A et d'autres entreprises, le taux d'échange des argents, Les revenus. Dans cette situation, nous désirons prédire la valeur d'une variable basée sur une séquence des valeurs passées du même et d'autres variables, dans une situation d'une seule journée Figure I.5, ces variables sont: r_A , r_B , r_C , Taux d'échange Euro – Dollars, Les revenus de l'entreprise.

Comme on peut apprécier cette tâche de prédiction des séries du temps est un exemple d'une classe des tâches très large s'appelle en mathématiques une fonction d'approximation ou une tâche de régression. Un système qui produit la solution de régression, généralement, faire des prévisions (cercles noirs dans la Figure I.5), un peu dévié de la valeur vraie (courbe). La différence entre la valeur prédite et la valeur réelle (aussi appelée valeur cible) constitue une erreur de prédiction. Et l'objectif ici est une solution avec des valeurs prédites similaires à la cible, c'est-à-dire, avec des petites erreurs.

Comme une réalité, les tâches de régression peuvent également être exprimées sous la forme des tâches de classification. Nous pouvons diviser le domaine (r_A) à des intervalles suffisamment petits et interpréter la solution de régression comme solution de classification, où une classification correcte correspond à une valeur prédite qui tombe dans le bon intervalle (*classe*). Dans ce sens, on peut visualiser la séquence des valeurs comme un vecteur de caractéristiques, [r_A , r_B , r_C , Taux d'échange Euro – Dollars, Les revenus de l'entreprise pendant 6 mois], et encore nous exprimons la similarité en termes de distance, on parle maintenant des valeurs prédites et cibles (classifications). Notez qu'une régression grossière peut être: prédire si $r_A(t)$ est plus grande ou non que la valeur précédente $r_A(t - 1)$. Ceci est

équivalent à un problème de classification de 2-classes avec la fonction $\text{signe}(r_A(t) - r_A(t - 1))$.

Parfois les tâches de régression sont également effectuées dans le cadre d'une classification. Par exemple, dans la reconnaissance de tissu vivant, un facteur de mérite est souvent utilisé par les médecins, en fonction de plusieurs caractéristiques telles que la couleur, la texture, la réflexion de la lumière et la densité des vaisseaux sanguins. Un système de reconnaissance automatique des tissus tente alors à régresser le facteur mérite évalué par l'expert humain, avant d'établir une classification des tissus.

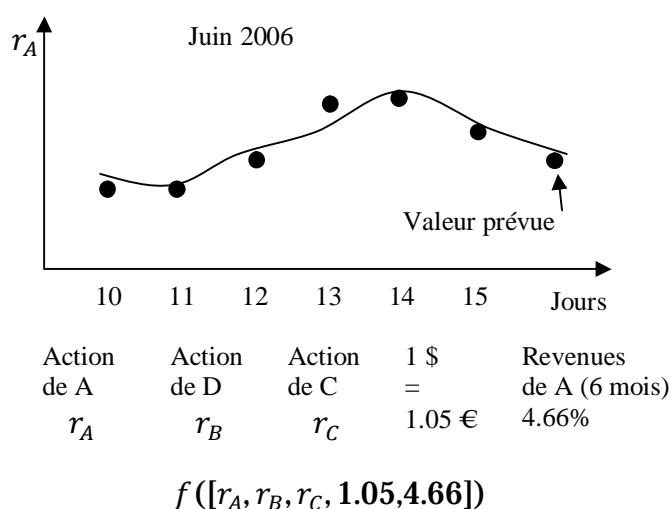


Figure I.5 : Prévisions des actions boursières d'un jour à venir. r_A, r_B, r_C sont des actions des trois entreprises. Approximation fonctionnel (cercles noirs) de la valeur vraie de r (courbe), est indiqué pour le 16 juin selon les valeurs des actions, le changement Euro - Dollar, les revenus de l'entreprise A pendant 6 mois.

3.3. TACHES DE DESCRIPTION

Dans les deux tâches de classification et de régression la similarité est une distance, et donc évaluée comme une quantité numérique. Un autre type de similarité est liée à la structure de la caractéristique des objets. Supposons que nous sommes présentés avec des tracés du rythme cardiaque d'un homme pendant une certaine période de temps. Ces tracés enregistrent la fréquence instantanée des frappes du cœur d'homme (entre 50 et 200 BPM) et sont utilisées par les obstétriciens pour évaluer le bien-être. Un tel dépistage est illustré dans la Figure I.6.

Ces tracés montrent des hauts et des bas par rapport à un certain niveau de référence correspondant au rythme de base du cœur (environ 150 BPM, Figure I.6a). Certains de ces hauts et bas sont les particularités du rythme cardiaque pour être interprété par l'obstétricien.

D'autres, comme les coups verticaux descendants dans la Figure I.6, sont des artefacts introduits par le matériel de mesure. Ces artefacts ou pics doivent être enlevés. La question est: quand une onde vers le haut ou vers le bas est un pic?

Afin de répondre à cette question, on peut commencer par décrire chaque traçage comme ordre des segments reliant les frappes successifs de cœur comme montré dans la Figure I.6b. Ces segments peuvent être alors classifiés dans les oligoéléments ou les primitifs énumérés dans le Tableau I.1.

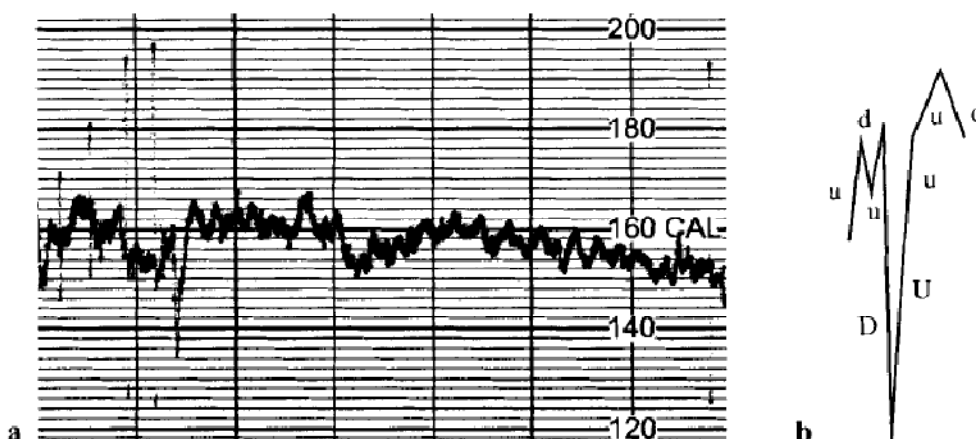


Figure I.6 : (a) Le traçage de la fréquence cardiaque avec la balance verticale par BPM. (b) un détail de la première onde de haut en bas est montrée avec ses primitifs.

Nom du primitif	Symbole	Description
Horizontal	h	Un segment de valeur constante
Pente vers le haut	u	Un segment ascendant avec la pente $< \Delta$
Pente vers le bas	d	Un segment descendant avec la pente $> -\Delta$
Pente fortement vers le haut	U	Un segment ascendant avec la pente $\geq \Delta$
Pente fortement vers le bas	D	Un segment descendant avec la pente $\leq -\Delta$

Δ est une valeur minimum de pente spécifique à l'avance

Tableau I.1 : Les primitifs des tracés de fréquence cardiaque.

A partir de ces éléments on peut décrire une transition en tant que n'importe quel séquence qui se composant d'un sous séquence des primitifs U suivis d'un sous séquence des primitifs D ou vice-versa, avec au moins un U et un D et aucuns autres primitifs entre eux.

Les Figure I.7a et Figure I.7b montrent des exemples des transitions et des non-transitions selon cette règle.

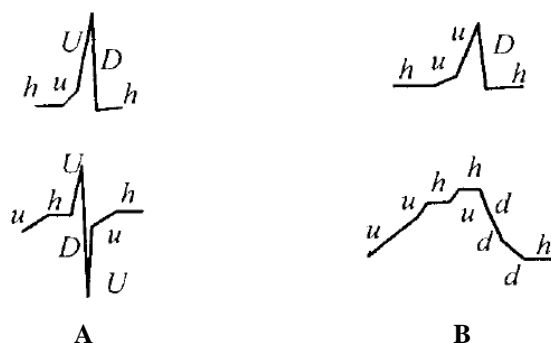


Figure I.7 : primitif d'onde pour le signal de la fréquence cardiaque :
(a) Transitions ; (b) Non-transitions.

Les non-transitions peuvent après être classifiées comme accélérations, décélérations ou d'autres types d'onde. La règle pour l'accélération peut être n'importe quelle séquence ascendante commençant par au moins un primitif u avec aucun d entre eux, en se terminant avec au moins un primitif de d avec aucun u entre eux. Un exemple est montré au bas de la Figure I.7b.

Dans cette tâche de description la similarité des objets (transitions, accélérations, décélérations, etc., dans cet exemple) est évaluée au moyen d'une *règle structurelle*. Deux objets sont similaires s'ils se conforment à la même règle. Par conséquent toutes les transitions sont semblables, toutes les accélérations sont semblables, et ainsi de suite. Noter en particulier que la transition inférieure de la Figure I.7a est, dans ce sens, plus semblable à la transition supérieure que l'onde supérieure de la Figure I.7b, bien qu'appliquant une mesure de distance aux valeurs des amplitudes de signal, en utilisant la première pic comme référence d'alignement de temps, mènerait certainement à un résultat différent.

La règle structurelle est appliquée ici à la séquence codée des primitifs, sous forme de chaîne de caractères des primitifs, afin de voir si la règle s'applique. Par exemple, une machine conçue pour décrire des tracés de fréquence cardiaque coderait le segment représenté sur la Figure I.6b en tant que « uduDUuud », identifiant de ce fait la présence d'une transition.

4. CLASSES, FORMES ET CARACTERISTIQUES

Dans les exemples de reconnaissance de formes présentés jusqu'à présent une correspondance assez simple existe entre les formes et les classes. Souvent la situation n'est pas celle simple. Considérons un cardiologue ayant l'intention de diagnostiquer une maladie de cœur basée sur l'interprétation des signaux électro cardiographiques acquis en plaçant des électrodes sur le coffre du patient. La Figure I.8 présente quatre ECG, chacun correspond à un état physiologique distinct : N - normale ; LVH - hypertrophie de ventricule gauche ; RVH - hypertrophie de ventricule droit ; MI - infarctus du myocarde.

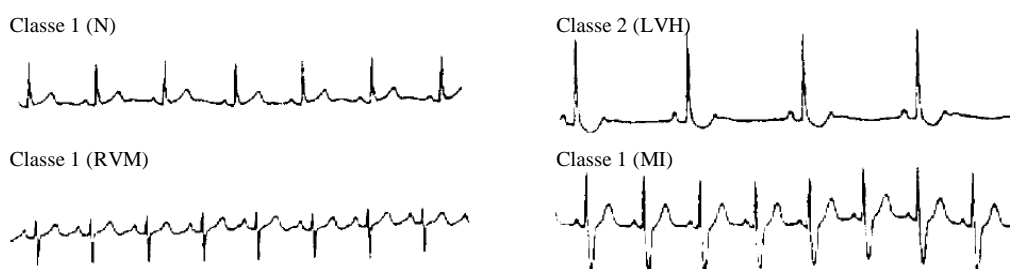


Figure I.8 : ECG de 4 classes agnostiques

La situation peut être résumée comme suite :

- Il y a un ensemble de *classes* (états) dans le quel peut être trouvé une certaine entité étudiée.
- un certain ensemble de *formes* (représentations) (signaux, images, etc.), correspondant à chaque classe (état).
- À partir de chaque forme nous pouvons extraire l'information qui la caractérise, les *caractéristiques*.

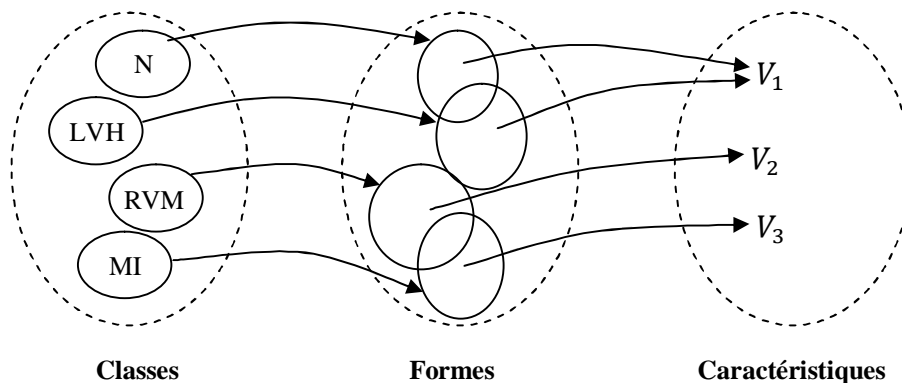


Figure I.9 : Les espaces de reconnaissance de formes pour la classification des formes (maladies de cœur dans cette situation) en utilisant caractéristiques d'ECG.

Un système de reconnaissance de formes émulant les capacités d'un humain (cardiologue), une fois présenté avec un vecteur de caractéristique, devrait déduire la maladie de cœur (Classe correspondante) du vecteur de caractéristiques. Le problème est que, comme nous voyons de la Figure I.9, il y a des chevauchements ennuyants : la même caractéristique peut être obtenue à partir de deux classes (N) et (LVH), la même forme peut être obtenue à partir de deux classes (MI) et (RVH). Le premier type de chevauchement peut être corrigé en utilisant des caractéristiques additionnelles, le deuxième type de chevauchement est intrinsèque à la méthode lui-même, en fait, les meilleurs experts en matière d'électrocardiographie ont une limite supérieure pour leur performance. Par conséquent, un système de reconnaissance de formes a fréquemment une erreur différente de zéro de performance, indépendamment de quel approche utilisée, et habituellement quelqu'un est satisfait s'il est équivalent avec ce que les experts humains peuvent atteindre.

Récapitulation de quelques notions :

4.1. LES CLASSES

Les classes sont les états de « nature » ou des *catégories* d'objets associés à des concepts ou prototypes. Dans ce qui suit, nous supposons c classes dénotés $\omega_i \in \Omega$, ($i = 1, \dots, c$), où Ω est l'ensemble de toutes les classes, connu sous le nom *espace d'interprétations*. L'espace d'interprétation a des propriétés des concepts telles que les unions, les intersections et les arbres hiérarchiques de classes.

4.2. LES FORMES

Les formes sont des représentations « physiques » des objets. Habituellement signaux, images ou tables simples des valeurs. Souvent on réfère aux formes avec *objets*, *cas* ou *échantillons*.

Dans ce qui suit nous allons utiliser la lettre n pour indiquer tout le nombre de formes disponibles afin de concevoir un système de RF, on appelle cette ensemble des formes *l'ensemble d'apprentissage*.

4.3. LES CARACTERISTIQUES

Les caractéristiques sont des mesures, des attributs ou des primitifs dérivés des formes, qui peuvent être utiles pour leur caractérisation.

Nous avons mentionné précédemment que le choix initial des caractéristiques adéquates est souvent plus un art qu'une science. Pour des raisons de simplicité on va utiliser seulement un nombre limité de caractéristiques. Fréquemment il y a des connaissances précédentes qui guident ce choix. Dans le cas de l'ECG, un traçage de 10s échantillonné à 500 Hz va donner 5000 échantillons de signal. Cependant il va être un choix catastrophique pour utiliser ces 5000 échantillons de signal comme caractéristiques! Heureusement il y a la connaissance médicale précédente qui nous guide pour choisir un ensemble de caractéristiques tout réduit. Le même type de problème apparaît quand nous voulons classifier des images numérisées. Pour une image à niveau de gris de 256x256 Pixel, nous avons un ensemble de 65536 valeurs (intensités de la lumière). Il est inconcevable d'utiliser ces valeurs comme des caractéristiques dans un système de RF! Cependant, fréquemment un ensemble réduit de mesures sur l'image est suffisant comme un vecteur de caractéristiques.

Le Tableau I.2 présente une liste de types communs de caractéristiques utilisés pour l'identification de signal et d'image.

Caractéristiques de signal	Caractéristiques d'image
<ul style="list-style-type: none"> - Amplitudes des ondes et durées. - Mesures d'histogramme. - Moments d'ondes (par exemple écart type). - Morphologie d'ondes (par exemple symétrie). - Passages à zéro. - Pics d'auto corrélation. - Pics spectrales (transformée de Fourier). 	<ul style="list-style-type: none"> - Taille de région. - couleur de région. - Intensité de lumière moyenne de région. - Moments d'image. - Mesures d'histogramme. - Pics spectrales (transformée de Fourier). - Caractéristiques topologiques (par exemple connectivité de région). - Caractéristiques morphologiques mathématiques.

Tableau I.2 : Types communs de caractéristiques de signal et d'image.

Dans ce qui suit on suppose que d est le nombre des caractéristiques ou primitifs d'une forme. Dans les problèmes de classification ou de régression nous considérons que les caractéristiques sont représentées par des nombres, une forme est donc représentée par un *vecteur de caractéristiques*:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \in X \subset R^d$$

Où X est le domaine à d dimensions des vecteurs de caractéristiques.

5. LES APPROCHES DE LA RF

Il y a une multiplicité d'approches de RF et aucun consensus clair sur comment les catégoriser. L'objectif d'un système de RF est d'accomplir une correspondance entre l'espace de représentation et l'espace d'interprétation. Une telle correspondance, qui est une classification, une régression ou une description, s'appelle également une *hypothèse*.

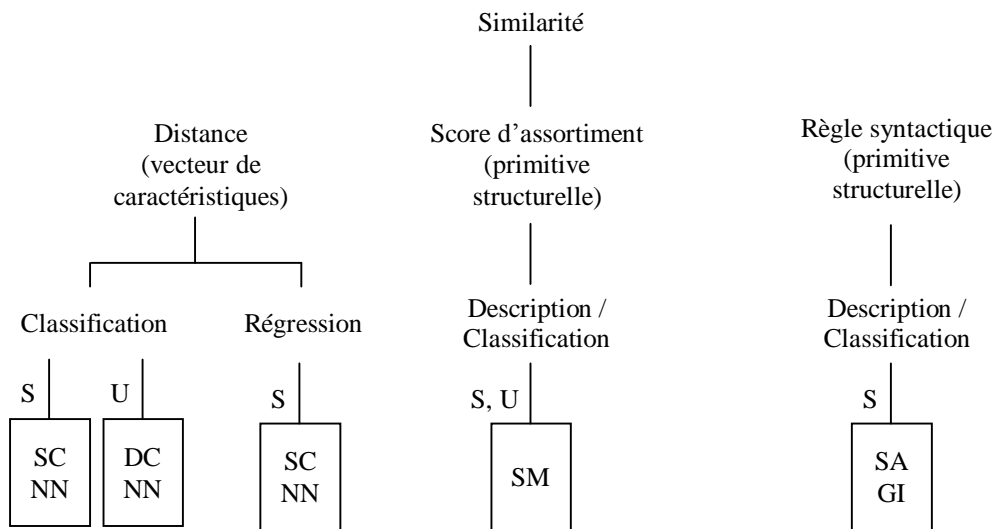


Figure I.10 : Les approches de RF - S : supervisé, U : non supervisé, SC : classification statistique, NN : réseaux neuronaux, DC : groupement de données, SM : classification structurelle, SA : analyse syntaxique, GI : inférence grammaticale [MAR01].

Il y a deux manières distinctes que telles hypothèses peuvent être obtenues :

Supervisée, guidée par les concepts (*concept-driven*) ou hypothèses inductives : trouver dans l'espace de représentation une hypothèse correspondante à la structure de l'espace d'interprétation. C'est l'approche des exemples précédents, où un ensemble de formes est donné et on suppose une solution.

Non supervisée, guidée par les données (*data-driven*) ou hypothèses déductives : trouver une structure dans l'espace d'interprétation correspondant à la structure dans l'espace de représentation. L'approche non supervisée tente de trouver une hypothèse utile basée seulement sur les relations de similarité dans l'espace de représentation.

L'hypothèse est dérivée par l'utilisation des méthodes d'apprentissage qui peuvent être statistiques, approximative (minimisation d'erreur) ou structurelle.

Tenant compte de comment l'hypothèse est dérivée et la similarité de forme est mesurée, nous pouvons établir la catégorisation hiérarchique montrée dans Figure I.10 [MAR01].

Nous procédons décrire brièvement les caractéristiques principales et le champ d'application de ces approches.

5.1. GROUPEMENT DE DONNEES (DATA CLUSTERING)

L'objectif de groupement de données est d'organiser des données (formes) en groupes significatifs ou utiles par l'utilisation d'un certain type de mesure de similarité. Le groupement de données n'utilise aucune information préalable de classe. C'est donc une méthode non supervisée de classification, les solutions sont arrivées d'une manière guidée par les données (*data-driven*), c.-à-d., ne dépend sur aucun superviseur ou maître.

Le groupement de données est utile quand on veut extraire une certaine signification à partir d'une pile d'informations non classifiée ou dans une phase exploratoire de recherche de reconnaissance des formes pour évaluer des similarités de données internes.

Exemple du groupement de données : Soit une table contenant des rendements de récolte par hectare pour plusieurs sorts de sol l'objectif est de grouper ces sorts dans des groupes significatifs.

5.2. CLASSIFICATION STATISTIQUE (RF STATISTIQUE)

La classification statistique est une approche classique de la RF établie depuis longtemps, dont il les mathématiciens établirent un grand nombre des méthodes et des formules. Elle est essentiellement basée sur l'utilisation des modèles probabilistes pour les distributions de vecteur de caractéristiques dans les classes afin de dériver des fonctions de classification. L'évaluation de ces distributions est basée sur un ensemble des formes d'apprentissage dont la classification est connue à l'avance (par exemple assigné par des experts humains). C'est donc une méthode supervisée de reconnaissance des formes, dans la manière au quel les solutions sont arrivées est guidée par les concepts (*concept-driven*). Si le classificateur est efficacement conçu il va se comporter bien sur de nouvelles formes.

Il y a des variantes de l'approche de classification statistique, qui dépendent sur le paramétrage, le modèle de distribution est utilisé ou pas.

L'approche de classification statistique est adéquate quand les formes sont distribuées dans l'espace de caractéristiques entre les différentes classes, selon des topologies simples et de préférence avec des distributions probabilistes connues.

Exemple de système de classification statistique : Une machine est donnée une tâche de séparation des taquets dans plusieurs catégories selon le type de défauts qu'ils présentent. Pour ce but, les défauts sont caractérisés par plusieurs caractéristiques. La machine utilise un classificateur statistique basé sur ces caractéristiques afin de réaliser la séparation.

5.3. LES RESEAUX NEURONAUX

Les réseaux de neurones sont inspirés à partir de la physiologie de l'organisation du cerveau. Ils sont structurés comme un ensemble d'unités identiques interconnectées connues sous le nom de *neurones*. Les interconnexions sont employées pour envoyer des signaux d'un neurone à des autres, d'une manière améliorée ou inhibée. Cette amélioration ou inhibition est obtenue en ajustant les *poids* de connexion.

Les réseaux neuronaux peuvent effectuer des tâches de classification et de régression avec ou sans supervision. Ils accomplissent ceci par des méthodes appropriées de réglage de poids, en espérant que les sorties du réseau convergent aux valeurs cibles justes.

Au contraire à la classification statistique, les réseaux neuronaux ont l'avantage d'être des machines non restreintes par des modèles (*Model-free*), ils se comportent en tant que des machines d'approximations universelles, capable de l'ajustement sur n'importe quelle sortie ou topologie désirée des classes dans l'espace de caractéristiques. Un inconvénient des réseaux neuronaux comparés à la classification statistique est que ses calculs mathématiques sont plus complexes et pour quelques décisions importantes le concepteur a souvent un peu de guidage théorique, et doit compter sur l'heuristique *essaie-et-erreur*. Un autre inconvénient, qui peut être important dans quelques circonstances, est que pratiquement aucune information sémantique n'est fournie par un réseau neuronal. Afin d'apprécier ce dernier point, imaginons qu'un médecin effectue une tâche de diagnostic aidé par un réseau neuronal et un classificateur statistique, alimentés avec les mêmes valeurs d'entrée (symptômes) et produisent la réponse correcte, peut-être contraire à la connaissance ou l'intuition du médecin. Dans le cas du classificateur statistique le médecin est probablement capable de percevoir la façon dont la sortie a été atteinte, à l'aide des modèles de distributions. Dans le cas du réseau neuronal cette perception est habituellement impossible.

Les réseaux neuronaux sont préférables sur les approches statistiques classiques, particulièrement quand la taille d'ensemble d'apprentissage est petite comparée à la dimensionnalité du problème à résoudre.

Des exemples des applications de réseau neuronal : L'approximation des fonctions, classification.

5.4. RECONNAISSANCE DE FORMES STRUCTURELLE

La reconnaissance de formes structurelle est l'approche suivie à chaque fois qu'on est besoin de prendre en compte l'ensemble de relations reliant les parties de l'objet à reconnaître. Parfois la reconnaissance prend la forme d'assortiment structurelle « structural matching », quand on doit évaluer à quel point un objet ou des parties de l'objet inconnues s'associe à un prototype. Un score d'assortiment est alors calculé à ce but, qui n'a pas nécessairement les propriétés habituelles d'une mesure de distance.

Un type particulier de la reconnaissance de formes structurelle, connu sous le nom « reconnaissance de formes syntaxique », peut être suivi quand on réussit à formaliser des règles pour décrire les relations entre les parties de l'objet. Le but de la machine de reconnaissance est alors de vérifier si un ordre des primitifs d'une forme se conforme à un certain ensemble de règles, connues sous le nom « règles syntaxiques » ou « grammaire ». À ce but un analyseur syntaxique est construit et l'ordre des primitifs est entré dans le.

L'analyse structurelle est tout à fait distinctive des autres approches. Elle fonctionne avec l'information symbolique, souvent avec des chaînes de caractères. Elle est parfois employée à un niveau plus haut que les autres méthodes, par exemple dans l'interprétation d'image, après avoir segmenté une image dans des primitifs par l'utilisation d'une approche statistique ou neuronal, la structure ou la relation lient ces primitifs peut être élucidée par l'utilisation d'une approche structurelle.

Quelques approches structurelles peuvent être implémentées par l'utilisation des réseaux neuronaux [MAR01].

Exemple d'analyse structurelle : Etant donné des images contiennent des écritures manuscrites, l'analyseur qui décrit les ascendant et les descendant est un analyseur structurelle.

6. PROJET DE RF

6.1. TACHES DU PROJET

Les systèmes de reconnaissance de formes, indépendant de l'approche suivie pour les concevoir, ont des unités facultatives spécifiques comme montré dans la Figure I.11. Quelques systèmes n'ont pas de l'unité de post-traitement et/ou de prétraitement.

Les unités d'un système de reconnaissance de formes et les tâches du projet correspondantes sont :

1. *Acquisition de formes*, qui peut prendre plusieurs formes : acquisition de signal ou d'image, collecte de données.

2. *Prétraitement et Segmentation*, en cas d'acquisition d'image, généralement l'image n'est pas prête à l'extraction des caractéristique et des étapes de prétraitement et segmentation sont nécessaires, dans le prétraitement on parle de seuillage, de squelettisation, etc.

3. *Extraction des caractéristiques* sous forme de mesures, extraction des primitifs, etc.
4. *Prétraitement des caractéristiques*, dans certains cas les valeurs de caractéristiques ne sont pas directement introduites dans le classificateur ou le descripteur. Par exemple dans les réseaux neuronaux il est habituel de normaliser les caractéristiques d'une certaine manière.
5. L'unité de *classification, régression ou de description* : est l'unité noyau du système de reconnaissance de formes.
6. *Post-traitement* : Parfois la sortie obtenue à partir de l'unité noyau de reconnaissance de formes ne peut pas être directement utilisée. Il peut être qu'elle besoin, par exemple, d'une certaine opération de décodage. Ceci s'appelle un post-traitement.

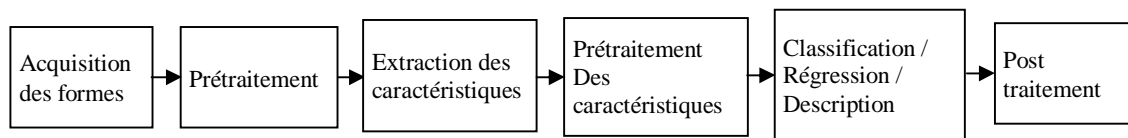


Figure I.11 : Un système de reconnaissance de formes avec ses unités. Quelques systèmes n'ont pas de l'unité de prétraitement et/ou de post-traitement.

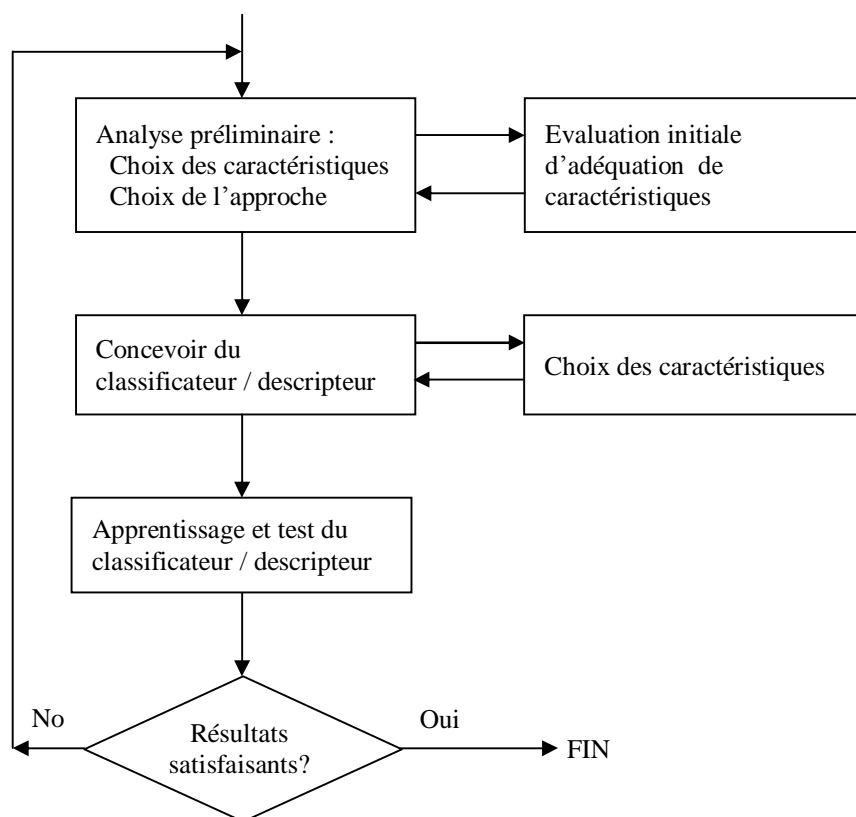


Figure I.12 : Les phases d'un projet de reconnaissance de formes. Note : l'évaluation de caractéristique à deux phases distinctes.

Bien que ces tâches soient principalement organisées séquentiellement, comme montré dans la Figure I.11, quelques boucles de contre-réaction (feedback) peuvent être présentes, au moins pendant la phase de conception, puisqu'il y a une interdépendance des solutions adoptées au niveau de chaque unité. Par exemple, le type d'acquisition de forme utilisé peut influencer le choix des caractéristiques, et donc des autres unités aussi. Autres influences sont plus subtiles : par exemple, le type de prétraitement appliqué sur les caractéristiques entrées à un réseau neuronal peut influencer la performance globale d'une manière dont il est difficile d'apprendre.

Un projet de reconnaissance de formes doit considérer toutes les tâches mentionnées et évolue dans une manière schématique à travers les phases représentées sur la Figure I.12 [MAR01].

6.2. APPRENTISSAGE ET TEST

Comme mentionné dans la section précédente le développement d'une application de reconnaissance de formes commence par l'évaluation du type de caractéristiques à utiliser et de l'approche adéquate de reconnaissance de formes pour le problème à résoudre. À ce but un ensemble initial de formes est habituellement disponible. Dans les approches supervisées cet

ensemble initial représenté par n vecteurs d -dimensionnels de caractéristiques ou n chaînes établis avec d primitifs, est utilisé pour développer le noyau de reconnaissance de formes. Il constitue l'ensemble d'apprentissage.

La performance d'un système de reconnaissance de formes est habituellement évaluée en termes de taux d'erreur pour chaque classe et un taux d'erreur globale. Quand cette évaluation de performance est basée sur les formes de l'ensemble d'apprentissage nous obtenons, en moyenne, des vues optimistes. Afin d'obtenir de meilleures évaluations d'une performance de système de reconnaissance de formes il est indispensable pour l'évaluer en utilisant un ensemble indépendant de formes, c.-à-d., formes non utilisées dans son conception ou son apprentissage. Cet ensemble indépendant de formes s'appelle un *ensemble de test*. Les évaluations de performance de système de reconnaissance de formes à l'aide d'ensemble de test nous donnent une idée d'à quel point le système est capable de généraliser ses capacités de reconnaissance sur des nouvelles formes.

Pour des systèmes de classification et de régression le degré de confiance qu'on peut avoir sur des évaluations de performance de système de reconnaissance de formes, aussi bien sur ses possibilités de généralisation, dépend fortement du rapport n/d , le rapport de dimensionnalité.

6.3. LES LOGICIELS DE RF

Il y a beaucoup de produits pour développer les applications de reconnaissance de formes, qui peuvent guider la conception d'un système de reconnaissance de formes des premières étapes jusqu'à l'évaluation finale. Une seule recherche par l'Internet révélera plusieurs de ces produits et outils, gratuits ou payants. Beaucoup de ces produits sont spécialisés, par exemple dans le secteur de réseaux neuronaux. D'une façon générale, les types suivants de logiciels peuvent être trouvés [MAR01]:

- Bibliothèques d'outil (par exemple dans C).
- Outils fonctionnant sous d'autres logiciels (par exemple Microsoft Excel ou le Matlab de Math Works).
- Produits pour la conception des applications de RF en utilisant une méthode spécifique.
- Produits pour la conception des applications de RF en utilisant une panoplie de différentes méthodes.

7. CONCLUSION

Jusqu'à ici nous avons obtenu un ensemble de notions de bases sur le domaine de reconnaissance de formes, nous pouvons poursuivre vers un domaine plus spécialisé, vers la reconnaissance de l'écriture arabe dans le deuxième chapitre.

CHAPITRE II

LA RECONNAISSANCE D'ÉCRITURE ARABE

Dans ce chapitre:

- ✓ Les phases d'un système de reconnaissance d'écriture.
- ✓ La reconnaissance de l'écriture Arabe manuscrite.

1. INTRODUCTION

Dans ce chapitre nous allons présenter les grandes étapes qui composent une chaîne de reconnaissance de l'écriture en générale, puis un état de l'art plus spécifique de la reconnaissance de l'écriture arabe manuscrite.

Nous aborderons la question des prétraitements et normalisations, la segmentation, l'extraction de caractéristiques, et nous passerons également au post-traitement. Enfin, nous terminerons ce survol avec une étude bibliographique sur les bases de données et les approches existantes de la reconnaissance de l'écriture arabe manuscrite.

2. LES PHASES D'UN SYSTEME DE RECONNAISSANCE D'ECRITURE

Un système de reconnaissance fait appel généralement aux étapes suivantes: Acquisition, prétraitements, segmentation, extraction des caractéristiques, classification, suivis éventuellement d'une phase de post-traitement.

2.1. ACQUISITION D'IMAGE

L'acquisition hors ligne et l'acquisition en ligne ce sont deux modes différents d'un système de reconnaissance des mots, ayant chacun ses outils propres d'acquisition et ses algorithmes correspondants de reconnaissance.

2.1.1. Acquisition hors ligne

Dans le cas hors ligne, il s'agit de reconnaître des textes manuscrits à partir de documents écrits au préalable. L'image du texte écrit est numérisée à l'aide d'un scanner, les informations recueillies se présentent sous la forme d'une image discrète constituée d'un ensemble de pixels Figure II.1a.

2.1.1. Acquisition en ligne

Dans le cas en ligne, il s'agit de reconnaître l'écriture au fur et à mesure de son tracé. Le texte est saisi avec un stylo et une tablette à numériser, les informations recueillies sont constituées par une suite ordonnée de points (définis par leurs coordonnées) échantillonnés à cadence fixe Figure II.1b.

La reconnaissance hors ligne ne peut pas a priori s'appuyer sur l'information temporelle du tracé qui est perdue, mais elle peut tenir compte de l'épaisseur du tracé (les pleins et les déliés). La reconnaissance en ligne peut disposer de l'information temporelle (vitesse, accélération, levés de stylo, retours en arrière, barres de t, points diacritiques), mais d'aucune information sur l'épaisseur du tracé si on ne dispose pas d'un signal de pression de la pointe du stylet sur le support.

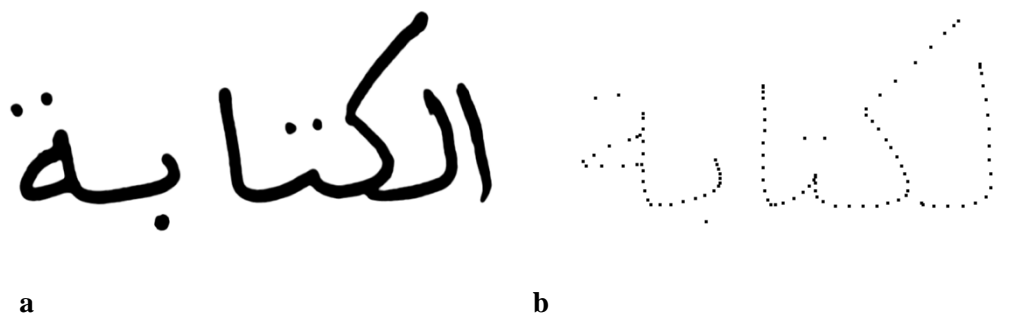


Figure II.1 : (a) Écriture hors ligne (ensemble de pixels), (b) Écriture en ligne (ensemble de coordonnées).

2.2. PRETRAITEMENTS ET NORMALISATIONS

Le prétraitement consiste à préparer les données issues du capteur à la phase suivante. Il s'agit essentiellement de réduire le bruit superposé aux données et essayer de ne garder que l'information significative de la forme représentée. Le bruit peut être dû aux conditions d'acquisition (éclairage, mise incorrecte du document, ...) ou encore à la qualité du document d'origine.

2.2.1. Seuillage

On distingue en général deux approches : les approches à seuillage global, et les approches à seuillage adaptatif [MEN08].

Pour une évaluation de différentes méthodes de binarisation, on pourra consulter le travail [TRI95].

2.2.1.1. Seuillage global

Le seuillage global consiste à prendre un seuil ajustable, mais identique pour toute l'image. Chaque pixel de l'image est comparé à ce seuil et prend la valeur blanc ou noir selon qu'il est supérieur ou inférieur. Cette classification ne dépend alors que du niveau de gris du pixel considéré.

Cette méthode convient pour les documents simples et de bonne qualité. Néanmoins, elle n'est plus applicable lorsque la qualité d'impression du texte n'est pas constante dans toute la page, des caractères peuvent être partiellement perdus. Des problèmes surviendront également si le fond est bruité ou non homogène, dans ce cas des taches parasites peuvent apparaître.

Les histogrammes de la répartition des niveaux de gris des pixels de ces images contiennent deux pics nets : l'un pour la forme, l'autre pour le fond. Un seuil global peut aisément être sélectionné entre les deux pour classer les pixels selon qu'ils appartiennent aux formes ou au fond.

2.2.1.2. Seuillage adaptatif

Dans les documents pour lesquels l'intensité du fond et l'intensité de la forme peuvent varier au sein du document, un seuillage global est inadapté. Il devient nécessaire de choisir le seuil de binarisation de manière locale. On calcule un seuil de binarisation pour chaque pixel de l'image, en fonction de son voisinage.

2.2.2. Squelette

La squelettisation est une opération qui permet de passer d'une image à sa représentation en "fil de fer". Le squelette a un pixel d'épaisseur. C'est une manière de représenter l'information indépendamment de l'épaisseur initiale de l'écriture.

Il permet d'extraire des caractéristiques importantes, comme les intersections et le nombre de tracés, leurs positions relatives. Il est également possible de renormaliser l'épaisseur de l'écriture à partir du squelette.

Il n'existe pas de définition unique du squelette. Le squelette doit seulement remplir trois conditions:

- Il doit être aussi fin que possible (typiquement, 1 pixel d'épaisseur).
- Il doit respecter la connexité.
- Il doit être centré dans la forme qu'il représente.

A chaque composante connexe de la forme, le squelette correspondant ne doit être composé que d'une seule composante connexe d'un pixel d'épaisseur incluse dans la première (voir Figure I.1).

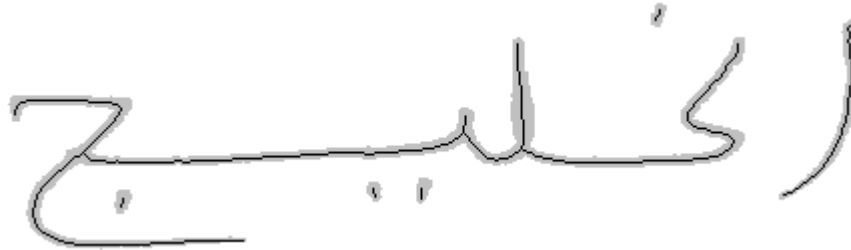


Figure II.2 : Squelettisation du mot 'الخليج'.

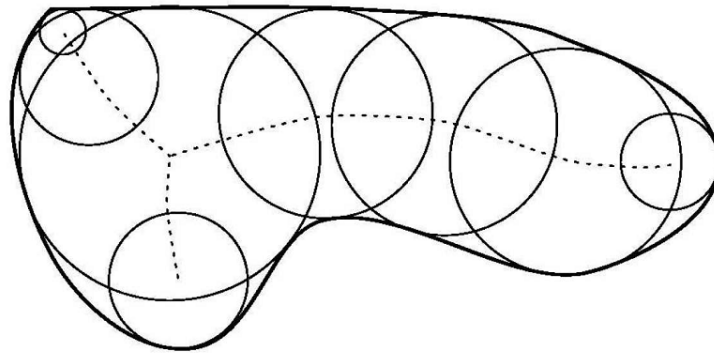


Figure II.3 : Axe médian d'une forme continue (trait pointillé) dont seul le contour est représenté [THI94].

Un autre moyen d'obtenir le squelette est de procéder par affinages successifs : la forme est "épluchée" (peeling) de manière itérative, en maintenant valide le critère de connexité.

L'algorithme de Hilditch [HIL69] exploite ce principe.

On définit le voisinage du pixel P_1 comme :

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

On définit également:

- $A(P_1)$: Nombre de transitions 0 vers 1 dans la séquence $P_2, P_3, P_4, P_5, P_6,$

P_7, P_8, P_9

- $B(P_1)$: Nombre de pixels noirs dans le voisinage de P_1 (P_1 étant exclu).

$$B(P_1) = \sum_{i=2}^9 P_i, \text{ avec } \begin{cases} P_i = 1 & \text{si le pixel est noir} \\ P_i = 0 & \text{si le pixel est blanc} \end{cases}$$

L'image est parcourue de multiples fois, et un pixel noir est marqué comme effaçable s'il respecte les quatre conditions suivantes:

$$2 \leq B(P_1) \leq 6$$

$$A(P_1) = 1$$

$$P_2 \cdot P_4 \cdot P_8 = 0 \text{ ou } A(P_2) \neq 1$$

$$P_2 \cdot P_4 \cdot P_6 = 0 \text{ ou } A(P_4) \neq 1$$

- Condition1 : $2 \leq B(P_1) \leq 6$. La première inégalité s'assure que P_1 n'est ni un point isolé, ni une extrémité. La deuxième inégalité s'assure que P_1 est bien un point du contour de la forme.
- Condition2 : $A(P_1) = 1$. Cette règle maintient le critère de connectivité.
- Condition3 : $P_2 \cdot P_4 \cdot P_8 = 0$ ou $A(P_2) \neq 1$. Cette règle s'assure que les lignes verticales de 2 pixels de large ne sont pas totalement érodées.
- Condition4 : $P_2 \cdot P_4 \cdot P_6 = 0$ ou $A(P_4) \neq 1$. Cette règle s'assure que les lignes horizontales de 2 pixels de large ne sont pas totalement érodées.

Tous les points marqués comme effaçables sont ensuite effacés, et on réitère l'opération sur la nouvelle image, jusqu'à ce que plus aucun point ne soit effaçable.

De manière générale, les méthodes d'obtention du squelette par érosions successives fonctionnent selon le principe suivant :

1: repeat

2: Appliquer masques et règles pour déterminer les pixels noirs effaçables

3: Changer tous les pixels marqués comme effaçables de noir vers blanc

4: until Aucun nouveau pixel n'a été effacé

2.2.3. Normalisations

Les normalisations ont pour objectif de rendre l'écriture la plus indépendante possible du scripteur.

Nous présenterons ici trois techniques de normalisation :

2.2.3.1. Correction de l'inclinaison des lignes

La correction de l'inclinaison des lignes de texte (également appelée correction de "skew"), consiste à redresser horizontalement les lignes d'écriture obliques.

Plusieurs méthodes sont disponibles. Les deux plus populaires sont la transformée de Hough (appliquée sur les centre de gravité des composantes connexes), et les histogrammes de projection.

La méthode de l'histogramme consiste à parcourir l'image selon des directions d proches de l'horizontale, et à compter le nombre de pixels noirs selon ces directions pour chaque ligne [MEN08] (voir Figure II.4).

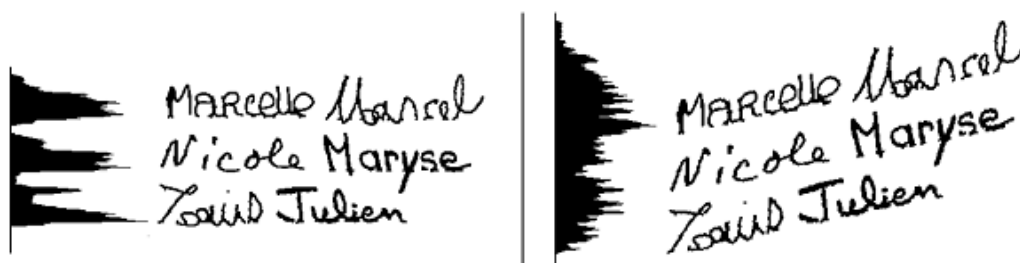


Figure II.4 : Correction de l'inclinaison des lignes à l'aide d'histogrammes de projection horizontale. Les pics les plus prononcés correspondent à la rotation correcte [CAR03].

L'angle du document (ou de la ligne), θ , est celui qui correspond à l'histogramme d'entropie maximale.

Pour corriger cette inclinaison, il suffit d'appliquer une rotation de l'image d'angle θ [MEN08]:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

2.2.3.2. Correction de l'inclinaison des lettres

Certains scribes écrivent leurs lettres de façon inclinée par rapport à l'axe vertical. Cette inclinaison de l'écriture est également appelée « slant ». Les lettres peuvent être inclinées vers la droite ou vers la gauche. Pour la même raison que dans le paragraphe précédent, il convient de corriger cette inclinaison de l'écriture pour la rendre la plus indépendante possible des spécificités d'écriture du scribe.

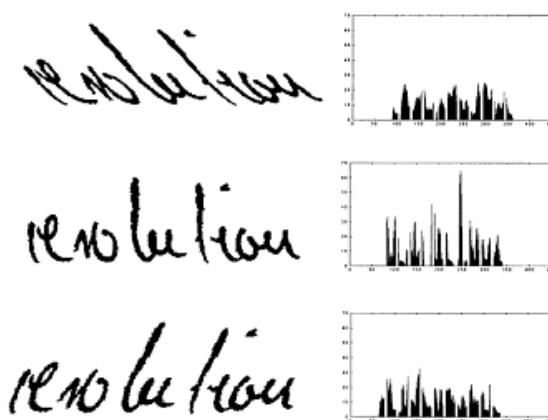


Figure II.5: Inclinaison de l'écriture, et histogrammes de projection verticale correspondants.

2.2.3.3. Lissage du contour

Dans certains cas, la numérisation, binarisation ou les prétraitements peuvent introduire des bruits dans l'image, qui se traduisent en particulier par la présence d'irrégularités le long des contours des lettres. Ces bruits peuvent dégrader les performances de reconnaissance.

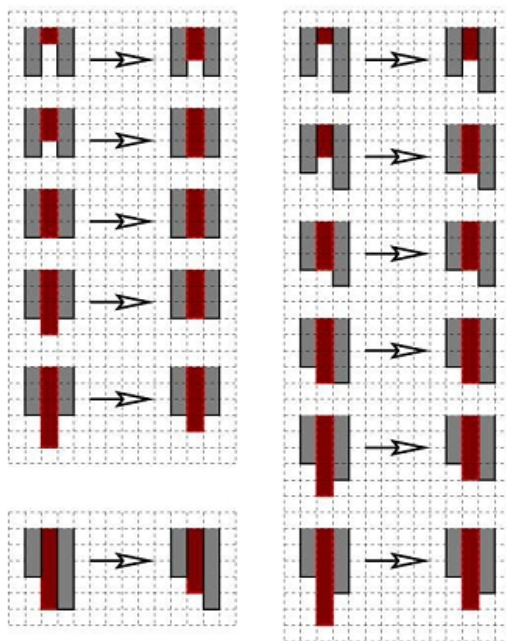


Figure II.6 : Masques qui permettent de lisser le contour des formes. Les masques complémentaires s'obtiennent par rotation [MEN08].

2.2.3.4. La ligne de base

Les lignes de base encadrent la bande des minuscules et délimitent les zones contenant les ascendants et descendants. Ces lignes sont importantes en reconnaissance de l'écriture. Cette information est utilisée pour détecter ascendants et descendants, et également pour normaliser les primitives, les rendant ainsi moins dépendantes de la hauteur de l'écriture.



Figure II.7 : Histogramme de projection horizontale et bande de base.

2.3. SEGMENTATION

Dans cette phase les différentes parties logiques d'une image sont extraites. A partir d'une image acquise il y'a d'abord séparation des blocs de texte et des blocs graphiques, puis à partir d'un bloc de texte il y'a extraction des lignes, ensuite à partir de ces lignes sont extraits les mots puis les caractères ou graphèmes [BAD95].

2.3.1. Segmentation de la page

Cette étape permet de localiser dans chaque page, les zones d'information conformément à leur apparence physique. Elle est associée généralement à l'étiquetage logique qui consiste à déterminer la nature du media représenté dans chaque zone (texte, graphique, photographie etc.).

Cette classification permet ensuite d'orienter la reconnaissance vers des systèmes spécialisés dans l'analyse de chaque type de media [BEL97].

2.3.2. Segmentation d'un bloc de texte en lignes

Cette étape consiste à séparer les différentes lignes du texte pour en extraire les mots puis les caractères composants les mots. La plupart des études proposées dans ce domaine s'appuient sur une décomposition de l'image en composantes connexes [BEN99b].

D'autres par contre utilisent des techniques s'appuyant en grande partie sur les histogrammes de la projection horizontale [BAD95]. Et certains auteurs optent pour des méthodes spécialisées telle que celle utilisée par Bennisri et Al dans [BEN99b] pour la segmentation en lignes de l'écriture arabe manuscrite.

2.3.3. Segmentation des lignes en mots

La segmentation en mots est réalisée en déterminant l'histogramme des projections verticales des lignes pour détecter les espaces entre les mots et pouvoir les séparer. Cependant cette technique peut ne pas être efficace dans certains cas où les mots se chevauchent (cas par exemple de l'écriture arabe). Dans ce cas d'autres techniques sont utilisées telles que : le suivi du contour, détermination des squelettes ou la détermination des composantes connexes...

2.3.4. Segmentation des mots

La segmentation des caractères est une opération qui tente de décomposer une image de séquence de caractères (mot) en sous-images de symboles individuels. C'est l'un des processus de décision dans un système de reconnaissance optique de caractères. Son but est de décider si un motif isolé d'une image (caractère ou autre entité identifiable du mot) est correct ou non [CAS96].

2.4. EXTRACTION DE CARACTERISTIQUES

Dans [OLI02], J. J. Oliveira et al présentent un certain nombre de primitives couramment utilisées en reconnaissance de l'écriture. Ils distinguent trois catégories:

- Les primitives perceptuelles
- Les primitives directionnelles
- Les primitives topologiques

Les auteurs indiquent que les primitives perceptuelles sont celles qui permettent d'obtenir les meilleurs résultats, et leur combinaison avec les autres est utile pour améliorer la reconnaissance.

La ligne de base est utile pour extraire et pour normaliser ces caractéristiques.

2.4.1. Les primitives perceptuelles

Dans la liste des primitives perceptuelles usuelles, on retrouve : la position des ascendants et descendants, leur hauteur, la position et la taille des boucles, des primitives extraites sur l'enveloppe convexe du mot, le nombre de transitions noir/blanc dans la zone centrale, ...

2.4.2. Les primitives directionnelles

Ces primitives sont généralement basées sur le squelette, ces types de primitives donnent la direction générale du tracé, longueurs et angles.

2.4.3. Les primitives topologiques

Ce type de caractéristiques est basé sur des densités de pixels. On peut par exemple projeter des images de tailles différentes (les graphèmes) dans une matrice de taille fixe. Les caractéristiques extraites sont les valeurs des cellules de cette matrice.

Dans ce type de primitives, on compte également les profils et histogrammes. Pour maintenir un vecteur de taille fixe, on divise l'image en un nombre fixe de bandes horizontales et verticales. Les caractéristiques sont les moyennes des valeurs sur ces bandes.

2.5. RECONNAISSANCE

Ou classification, C'est l'étape principale. On va aborder cette étape au chapitre suivant où nous présenterons un ensemble de classificateurs couramment utilisés en reconnaissance de l'écriture.

2.6. POST-TRAITEMENTS

Des vérifications contextuelles classiques telles que la recherche dans un dictionnaire, les probabilités d'occurrence de bi-gramme et de trigramme..., sont appliquées dans les différents travaux qui prévoient un post-traitement.

La méthode du dictionnaire est traditionnellement simplifiée pour accélérer la recherche et réduire la complexité du calcul : le dictionnaire est construit à partir de mots réduits à leurs racines, les suffixes et les préfixes sont éliminés. Cependant des modèles sont élaborés afin de spécifier la relation racine-suffixe-préfixe [BEN99a].

3. LA RECONNAISSANCE DE L'ECRITURE ARABE MANUSCRITE

3.1. PROBLEMATIQUE

3.1.1. Présentation de la langue arabe

L'arabe littéral, arabe moderne unifié ou encore classique est le nom que l'on donne à une variante de la langue arabe, utilisée comme langue officielle dans tous les pays arabes, et comme langue commune entre pays arabes. Elle est également employée dans la plupart des écrits et, à l'oral, dans les situations officielles ou formelles (discours religieux, politiques, journaux télévisés).

L'arabe littéral se distingue ainsi de l'arabe dialectal, qui est la langue vernaculaire parlée au quotidien et ce depuis l'expansion de l'islam. Cette variété de la langue recouvre plusieurs dialectes locaux pouvant varier assez fortement d'un pays à l'autre. Dans tous les pays arabes, un dialecte national composé par plusieurs dialectes locaux est parlé. Aucun d'entre ces dialectes n'est identique complètement à l'arabe classique ou littéraire.

Notons également que le farsi (persan), utilisé principalement en Iran et en Afghanistan, partage un grand nombre de points communs avec l'écriture arabe. Cet aspect est hors du cadre de cette thèse : nous nous restreignons ici à l'étude de l'écriture arabe manuscrite sur des

données en provenance de scripteurs tunisiens. Sauf mention contraire, les données utilisées proviennent de la base IFN/ENIT [PEC02], une base de noms de villes tunisiennes.

3.1.2. Difficultés inhérentes à la reconnaissance de l'écriture arabe

3.1.2.1. Alphabet

L'alphabet arabe comporte 28 lettres (voir Tableau II.1) La forme des lettres dépend de leur position dans le mot. Certaines lettres prennent jusqu'à 4 formes différentes : par exemple (ع ع ع ع) ou (ه ه ه ه).

Mais pour la plupart des lettres, les formes début/milieu et fin/isolé sont identiques à la ligature près. La présence d'une ligature avec la lettre précédente ou avec la lettre suivante ne modifie pas la forme de la lettre de manière significative (pas plus que dans l'écriture manuscrite cursive latine).

En arabe, les ligatures se situent toujours au niveau de la ligne d'écriture, c'est-à-dire qu'il n'existe pas de lettre à liaison haute comme le 'o' ou le 'v' en alphabet latin, il existe toutefois des ligatures verticales.

3.1.2.2. Signes diacritiques

Le terme « signe diacritique » peut porter à confusion : dans certains travaux, seules les voyelles arabes sont appelées diacritiques. Dans d'autres travaux, en revanche, tous les signes secondaires sont appelés diacritiques, qu'il s'agisse des voyelles, des points ou des autres signes (chaddah, hamzah, ...).

C'est cette deuxième terminologie que nous employons ici : un signe diacritique est une composante secondaire d'une lettre, qui vient la compléter ou en modifier le sens. Dans la suite de cette thèse, les « signes diacritiques » désigneront à la fois points, voyelles et autres signes secondaires.

3.1.2.2.1. Des points nécessaires pour différencier les lettres

Dans l'alphabet arabe, 15 lettres parmi les 28 possèdent un ou plusieurs points. Ces signes diacritiques sont situés soit au-dessus, soit en dessous de la forme à laquelle ils sont associés, mais jamais les deux à la fois.

N°	A la fin	Au milieu	Au début	Isolée
1	ى	ا	ا	ا
2	ب	ب	ب	ب
3	ت	ت	ت	ت
4	ث	ث	ث	ث
5	ج	ج	ج	ج
6	ح	ح	ح	ح
7	خ	خ	خ	خ
8	د	د	د	د
9	ذ	ذ	ذ	ذ
10	ر	ر	ر	ر
11	ز	ز	ز	ز
12	س	س	س	س
13	ش	ش	ش	ش
14	ص	ص	ص	ص
15	ض	ض	ض	ض
16	ط	ط	ط	ط
17	ظ	ظ	ظ	ظ
18	ع	ع	ع	ع
19	غ	غ	غ	غ
20	ف	ف	ف	ف
21	ق	ق	ق	ق
22	ك	ك	ك	ك
23	ل	ل	ل	ل
24	م	م	م	م
25	ن	ن	ن	ن
26	ه	ه	ه	ه
27	و	و	و	و
28	ي	ي	ي	ي

Tableau II.1 : L'alphabet arabe

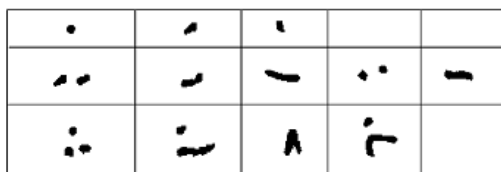


Figure II.8 : Points en arabe : un, deux ou trois points.

La Figure II.8 illustre la variabilité des styles d'écriture des points ou groupes de points en écriture arabe manuscrite. Un groupe de deux points peut ainsi s'écrire sous forme d'une seule, ou de deux composantes connexes. On remarque la très forte similarité entre deux points reliés par un trait, et une voyelle de type « A » ou « I » dont les exemples sont donnés par Figure II.9.

Un groupe de trois points peut donner lieu à une, deux ou trois composantes connexes, en fonction du style d'écriture.

3.1.2.2.2. Les voyelles

En arabe, les voyelles ne sont pas des lettres, mais des signes diacritiques associés aux lettres sur lesquelles ils s'appliquent (voir Figure II.9).

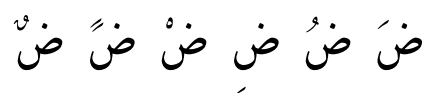


Figure II.9 : Voyelles en arabe.

En général on ne représente pas les voyelles.

L'absence de voyelles peut toutefois être source de confusions, un mot peut avoir plusieurs voyellations possibles et par conséquent plusieurs catégories grammaticales. Dans certains cas, une phrase peut donc avoir deux voyellations différentes, ce qui nous donne deux structures syntaxiques possibles. Par exemple, dans le cas :

يخشى الأستاذ الطلبة

Qui peut se voyeller des deux manières suivantes :

L'enseignant craint les étudiants.

يخشى الأستاذ الطلبة

Les étudiants craignent l'enseignant.

يخشى الأستاذ الطلبة

Les voyelles peuvent parfois être mentionnées sur certaines lettres pour lever l'ambiguïté et faciliter la lecture. Mais en général, les scripteurs les omettent purement et simplement, et c'est au lecteur qu'est réservé le soin d'interpréter correctement le sens de la phrase en fonction du contexte.

3.1.2.3. Ascendants et descendants

Comme dans l'écriture latine, l'écriture arabe contient des ascendants et des descendants (voir Figure II.10). En arabe, les descendants peuvent se prolonger horizontalement sous la bande de base, ce qui introduit une superposition verticale entre la lettre qui comprend le descendant et la lettre suivante.



Figure II.10 : Les ascendants et descendants sont entourés.

3.1.2.4. Une ou plusieurs composantes connexes par mot

6 lettres ne sont pas liées à leur successeur : ا, د, ذ, ر, ز, و. Ces lettres introduisent donc une coupure dans le mot. Un pseudo-mot est une unité connexe regroupant une ou plusieurs lettres sous forme d'une séquence. Un mot peut être composé d'un ou plusieurs pseudo-mots (voir Figure II.11).



Figure II.11 : Un mot peut être composé de plusieurs composantes connexes (pseudo-mots).

En manuscrit, l'espacement entre les différents pseudo-mots d'un même mot n'est pas forcément systématiquement supérieur à l'espacement entre deux mots différents, ce qui pose parfois des problèmes de segmentation.

3.1.2.5. Ligatures verticales

En écriture arabe, il n'y a pas de liaisons hautes comme le « v » ou le « o » en latin : les ligatures se situent au niveau de la ligne support de l'écriture (ligne de base). En revanche, les

scripteurs sont libres de constituer certains groupes de deux ou trois lettres liées verticalement en début de pseudo-mot. Ce sont les ligatures verticales (Figure II.12 et Figure II.13). En général très complexes à segmenter, nous choisissons de les reconnaître telles quelles.



Figure II.12 : لا sous forme d'une ou de deux composantes connexes



Figure II.13 : Ligatures verticales et inversion de l'ordre du tracé.

3.1.2.5.1. Ligatures verticales connectées ou non

Par définition, les éléments qui constituent une ligature verticale sont connectés. Ils appartiennent donc tous au même pseudo-mot. Toutefois, en fonction du style d'écriture et en raison des phénomènes de levés de plume, il est possible qu'une ligature verticale soit composée de plusieurs composantes connexes (voir Figure II.12).

3.1.2.5.2. Recouvrements partiels

La forme de certaines ligatures, combinée au style d'écriture du scripteur (en particulier lorsque le mot est écrit rapidement), peuvent parfois donner des résultats fâcheux, comme des recouvrements partiels, voire même une inversion de l'ordre de lecture (voir Figure II.13). Des situations très problématiques qui heureusement restent rares.

3.2. ETUDE BIBLIOGRAPHIQUE

3.2.1. Principales bases de données existantes

Le domaine de la reconnaissance automatique de l'écriture arabe manuscrite a ainsi péché pendant de nombreuses années de l'absence d'une base de données de référence, qui permette des comparaisons objectives entre les différents systèmes. Des bases plus conséquentes mais

payantes comme la base Kharma/Ahmed/Ward ou la base de chèques du CENPARMI sont apparues. Mais c'est la base IFN/ENIT, gratuite pour la recherche académique, qui s'est imposée comme la base de données de référence pour la comparaison des performances des systèmes de reconnaissance de l'écriture arabe manuscrite.

3.2.1.1. Kharma/Ahmed/Ward 1999

En 1999, [KHA99] proposent une base de données, obtenue à partir des contributions de 500 étudiants :

- 37000 mots arabes
- 10000 chiffres, à la fois arabes et indiens
- 2500 signatures
- 500 phrases

Les images sont disponibles à la fois en niveaux de gris et sous forme binarisées. Cette base est disponible pour les chercheurs Canadiens, mais son accès est soumis à restrictions pour les chercheurs du reste du monde.

3.2.1.2. AHDB

En 2002, Almaadeed et al. présentent une base de données collectées à l'aide d'une centaine de scripteurs [MAD02].

Chacun des scripteurs était invité à écrire :

- Chacun des mots du vocabulaire des montants numériques.
- Trois montants numériques écrits en toutes lettres.
- Quelques lignes de texte libre.

Les 20 mots les plus utilisés par tous les scripteurs sont annotés manuellement.

3.2.1.3. CENPARMI

En 2003, Y. Al-Ohali et al. ont achevé la réalisation d'une base de données pour la reconnaissance de chèques arabes manuscrits [ALO03].

Cette base se compose de 7000 images de chèques saoudiens issus de la pratique bancaire, scannés en niveaux de gris avec une résolution de 300 dpi.

Cet ensemble d'images a permis de mettre au point plusieurs bases de données :

- Une base de 1547 montants littéraux de chèques.
- Une base de 1547 montants numériques de chèques.
- Une base de 23325 pseudo-mots.
- Une base de 9865 chiffres indiens isolés.

Il serait également possible de mettre au point une base de données de dates à partir de ces images.

3.2.1.4. IFN/ENIT

M. Pechwitz et al. introduisent la base IFN/ENIT en 2002 [PEC02]. Il s'agit d'une base de données d'images de noms de villes tunisiennes. Outre la séquence de lettres, sont également annotées la forme que prend chacune des lettres au sein du mot, la présence des signes diacritiques secondaires, et une approximation de la ligne de base.

411 scripteurs ont été mis à contribution, pour collecter 26459 noms de villes (dans un lexique de 937 villes) et plus de 210000 caractères.

Dans cette base, l'annotation est réalisée de telle sorte qu'une séquence de lettres contienne également l'information de la forme que prend chacune des lettres au sein du mot (voir figure 3.11).

Cette base, par sa taille et sa disponibilité (gratuite pour des travaux académiques) s'est imposée comme la base de référence des travaux en reconnaissance de l'écriture arabe manuscrite. Elle sert de support à l'organisation des compétitions internationales dans le domaine de la reconnaissance de l'écriture arabe manuscrite : ICDAR Arabic Handwritten Competition, 2005 et 2007.

3.2.1.5. CEDARABIC

La base de données CEDARABIC [GRE06] comprend 10 scripteurs différents. Chacun d'entre eux a écrit 10 pages complètes différentes de texte manuscrit. Chaque page comprend

entre 150 et 200 mots, soit un total d'environ 20000 mots. Les documents sont scannés en 300 dpi. Une annotation complète des documents est faite manuellement : segmentation en mots, mais aussi séquence des lettres arabes, ainsi que la prononciation et la traduction en anglais.

Cette base de données est donc particulièrement adaptée pour des travaux sur la segmentation en mots et pour la recherche de mots clés dans une page d'écriture manuscrite en langue arabe.

3.2.1.6. Applied Media Analysis : Arabic-Handwriting-1.0

Cette base de données, qui a été rendue publique en septembre 2007, n'a pour l'instant fait l'objet d'aucune publication scientifique de la part de ses auteurs.

Ses principales caractéristiques sont :

- 5000 pages manuscrites
- Plusieurs types de documents : diagrammes, mémos, formulaires, listes (comprenant des chiffres indiens et arabes), poèmes
- Différents types de stylos : crayons à papier, marqueurs larges ou fins, pointes fines, stylos à bille. Ecriture en noir ou en couleur.
- Images disponibles en binaire ou en niveaux de gris
- Annotations XML : mots ou pseudo-mots, format UTF-8

Son coût est de 500\$ pour une recherche académique, 1500\$ pour l'industrie.

3.2.2. Différentes approches et systèmes existants

3.2.2.1. Système d'Amin et al. 2000 (Approche globale)

Le système proposé en 2000 par [AMI00] effectue la reconnaissance globale de mots arabes imprimés multi-fontes.

Pour l'auteur, le sujet reste ouvert et de nombreuses voies d'amélioration sont possibles. L'évolution des performances des systèmes de reconnaissance de l'écriture arabe au cours des 10 dernières années montre qu'il avait raison.

3.2.2.2. Système de Al Ohali, 2002 (Approche pseudo-analytique)

Dans [ALO02], Al Ohali propose un système de reconnaissance des sous-mots composant les mots du vocabulaire des montants de chèques issus de la base d'images, en utilisant une combinaison séquentielle entre un classifieur global neuronal et un classifieur analytique markovien. Les sous-mots reconnus sont concaténés pour reconstituer les mots, et, par la suite, les montants littéraux.

Testé sur un lexique de 67 sous-mots, le classifieur neuronal effectue une réduction lexicale avec un taux de réussite de 99.04%. Sa combinaison avec le classifieur markovien a augmenté le taux de reconnaissance de ce dernier de 3,68% puisqu'il est passé de 69,85 à 73,53%.

Ce système de reconnaissance de sous-mots s'intègre dans un système de reconnaissance de montants littéraux de chèques. Les mots sont reconstruits par une concaténation de sous-mots consécutifs (4, au maximum, dans ce vocabulaire) et une procédure de recherche dans un dictionnaire. Comme certains sous-mots sont communs à plusieurs mots, la séquence de sous-mots n'est pas forcément unique. Une analyse syntaxique, basée sur la grammaire des montants, permet de vérifier la validité des propositions de montants obtenues.

3.2.2.3. Système de Pechwitz et al. 2003 (Approche analytique)

[PEC03] proposent un système de reconnaissance de mots manuscrits basé sur une segmentation implicite et une modélisation markovienne des caractères. L'image du mot manuscrit passe d'abord par une étape de squelettisation, puis une normalisation horizontale pour obtenir une largeur moyenne constante de caractères et une normalisation verticale pour uniformiser la hauteur des ascendants et descendants.

L'extraction des caractéristiques est directement basée sur les valeurs des pixels. Une fenêtre rectangulaire est glissée du droite à gauche à travers de l'image de niveau de gris normalisée pour générer le vecteur de caractéristique. La Figure II.14 donne un exemple d'extraction de caractéristiques par une fenêtre glissante.

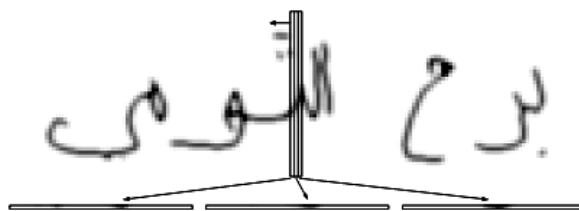


Figure II.14 : Extraction de caractéristique avec une fenêtre glissante de trois colonnes.

La base de données IFN/ENIT des mots manuscrits a été utilisée pour tester le système de reconnaissance. La base de mots a été divisée sur 4 groupes, 3 pour l'apprentissage et une pour le test. Le système a atteint 84% de reconnaissance avec l'estimation de la ligne de base basée squelette.

Pour l'auteur le sujet reste ouverte, et invite les chercheurs de tester sont systèmes avec la base de mots arabes manuscrits IFN/ENIT.

3.2.2.4. Système de Khorsheed, 2003 (Approche analytique)

Dans le système proposé par Khorsheed [KHO03], le mot manuscrit passe par une étape de squelettisation, puis le graphe du squelette est décomposé en une suite de segments après une étape d'approximation linéaire (voir Figure II.15).

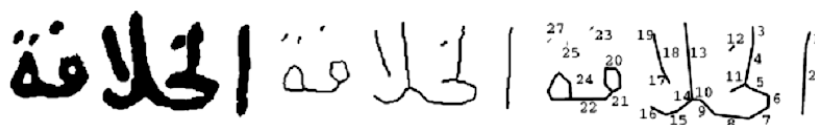


Figure II.15 : Mot originale squelette approximation linéaire.

Le taux de reconnaissance dans ce travail a atteint 72%, après l'ajout d'un poste-traitement de vérification orthographique le taux a été amélioré pour atteindre 87%.

3.2.2.5. Système de Farah et al. 2005 (Approche globale)

Cette proposition s'intègre dans le cadre des travaux de recherche effectués au LRI de l'université d'Annaba. Dans [FAR05], les auteurs proposent de combiner diverses sources d'informations pour reconnaître les 48 mots manuscrits appartenant au vocabulaire des montants littéraux. L'étape de reconnaissance est effectuée par une combinaison parallèle de trois types de classifieurs (réseau neuronal de type perceptron multicouches, K plus proches voisins, K plus proches voisins flou) utilisant les caractéristiques globales perceptuelles des mots (nombre de sous-mots, ascendants, descendants boucles et points diacritiques)

(Figure II.16). Le contexte grammatical des montants littéraux est utilisé pour prendre une décision finale sur les mots candidats obtenus.

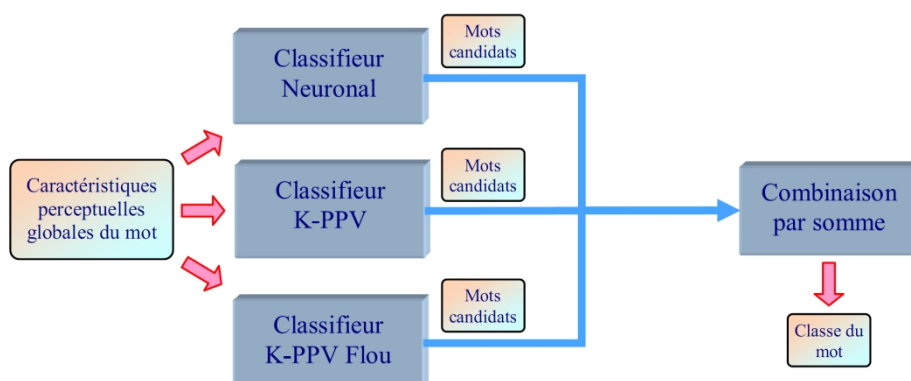


Figure II.16 : Combinaison de classificateurs dans [FAR05].

Une base de 4800 images de mots, représentant les 48 mots du lexique écrits par 100 scripteurs différents, a été utilisée. 1200 ont servi à l'apprentissage des trois classificateurs et le reste a été réservé aux tests. Les taux de reconnaissance obtenus par la combinaison des classificateurs sont de l'ordre de 94%. Une étape de post-traitement syntaxique, basée sur la grammaire des montants littéraux, a permis d'améliorer les résultats de près de 2% par rapport à la combinaison et de 4% par rapport au taux moyen de reconnaissance individuelle des classificateurs.

4. CONCLUSION

Nous avons présenté une vue globale des notions de bases des systèmes de reconnaissance d'écriture ainsi qu'une vue globale sur les systèmes de reconnaissance d'écriture arabe et quelques bases de mots existantes, on peut passer maintenant au troisième chapitre où nous étudierons les systèmes de classifications.

CHAPITRE III

LES SYSTÈMES DE CLASSIFICATION

Dans ce chapitre:

- ✓ Définition formelle d'un classifieur.
- ✓ Les Réseaux de Neurones.
- ✓ K Plus Proches Voisins.
- ✓ Modèles de Markov Cachés.

1. INTRODUCTION

Ce chapitre sera consacré à une présentation de quelques propriétés des réseaux de neurones qui ont été utilisées dans ce mémoire, puis un rappel de l'algorithme de k plus proches voisins qui a été aussi utilisé, enfin une brève description des Modèles de Markov Cachées à cause de son large utilisation dans le domaine de la reconnaissance de l'écriture arabe manuscrite.

2. DEFINITION FORMELLE D'UN CLASSIFIEUR

Définir un problème de classification, c'est d'abord se choisir une taxonomie c'est-à-dire une répartition des formes à reconnaître dans un ensemble de classes $\omega_i, i \in [1, \dots, c]$. Dans le cas le plus général, on peut associer à la forme à reconnaître x un vecteur de degré d'appartenance $D(x)$ tel que :

$$D(x) = \begin{bmatrix} D_1(x) \\ D_2(x) \\ \vdots \\ D_N(x) \end{bmatrix}$$

Avec $D_i(x) = D\{x \in \omega_i\}$ Dans ce cas, x peut appartenir à plusieurs classes si $D_i(x) \neq 0$. Toutefois, dans la majorité des problèmes de classification, on a affaire à une classification exclusive dans laquelle une forme ne peut appartenir qu'à une seule classe. On a alors $D_i(x) = \delta_{i,j}$ tel que :

$$\delta_{i,j} = \begin{cases} 1 & \text{si } j = i \\ 0 & \text{si } j \neq i \end{cases}$$

ω_i est alors souvent appelée « la vraie classe »:

$$D(x) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Selon Xu [XUL92] La réponse d'un classifieur peut être divisée en trois catégories suivantes :

2.1. CLASSIFIEUR DE TYPE CLASSE

$$e_j(x) = \omega_i, i \in \{1, \dots, c\}$$

ω_i est la classe attribuée par le classifieur e_j à x . Dans ce cas, l'avis du classifieur est binaire. On peut alors représenter la réponse du classifieur par un vecteur binaire dans lequel l'indique la classe proposée par le classifieur. Un classifieur peut aussi produire un ensemble de classes. Il considère alors que la forme x appartient à une des classes de cet ensemble sans donner d'autres informations permettant de discriminer les classes.

2.2. CLASSIFIEUR DE TYPE RANG

$$e_j(x) = (r_{1,j}, r_{2,j}, \dots, r_{N,j})$$

$r_{i,j}$ est le rang attribué à la classe ω_i par le classifieur e_j . Il s'agit d'un classement sur les classes. Le classifieur indique ce classement en fournissant en sortie un vecteur des rangs de taille c . La classe placée au premier rang de la liste proposée par le classifieur est considérée comme la plus probable pour la forme x et la classe du dernier rang est la moins probable.

2.3. CLASSIFIEUR DE TYPE MESURE

$$e_j(x) = (m_{1,j}, m_{2,j}, \dots, m_{N,j})$$

$m_{i,j}$ est la mesure attribuée à la classe i par le classifieur e_j . Elle indique le niveau de confiance du classifieur dans sa proposition. La sortie du classifieur est donc un vecteur de mesures de taille c . Cette mesure, normalisée ou non, peut être une distance, une probabilité a posteriori, une valeur de confiance, un score, une fonction de croyance, une possibilité, une crédibilité, une mesure floue, etc.

3. LES RESEAUX DE NEURONES

3.1. INTRODUCTION

Selon [HAY94]:

Un réseau de neurones est un processeur massivement distribué en parallèle qui a une propension naturelle pour stocker de la connaissance empirique (*experiential knowledge* selon l'auteur) et la rendre disponible à l'usage. Il ressemble au cerveau sur deux aspects:

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

Selon [NIG93]:

Un réseau de neurones est un circuit composé d'un nombre très important d'unités de calcul simples basées sur des neurones. Chaque élément opère seulement sur l'information locale. Chaque élément opère de façon asynchrone; il n'y a donc pas d'horloge générale pour le système. Bilan

3.2. LE NEURONE SIMPLE

3.2.1. Le neurone de Mac Culloch-Pitts

Le premier neurone artificiel apparu est le neurone de [MCC43], il se caractérise principalement par une fonction d'activation de type limitation 0 ou 1. Il est montré dans la Figure III.1.

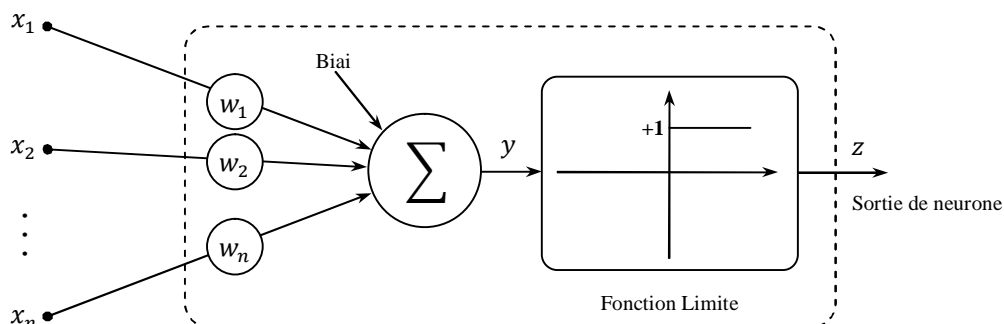


Figure III.1 : Le Neurone de Mac Culloch-Pitts [MCC43]

Le neurone formel est donc une modélisation mathématique qui reprend les principes du fonctionnement du neurone biologique, en particulier la sommation des entrées. Sachant qu'au niveau biologique, les synapses n'ont pas toutes la même «valeur» (les connexions entre les neurones étant plus ou moins fortes), les auteurs ont donc créé un algorithme qui pondère la somme de ses entrées par des poids synaptiques (coefficients de pondération). De plus, les 1 et les -1 en entrée sont là pour figurer une synapse excitatrice ou inhibitrice.

3.2.2. Le Perceptron

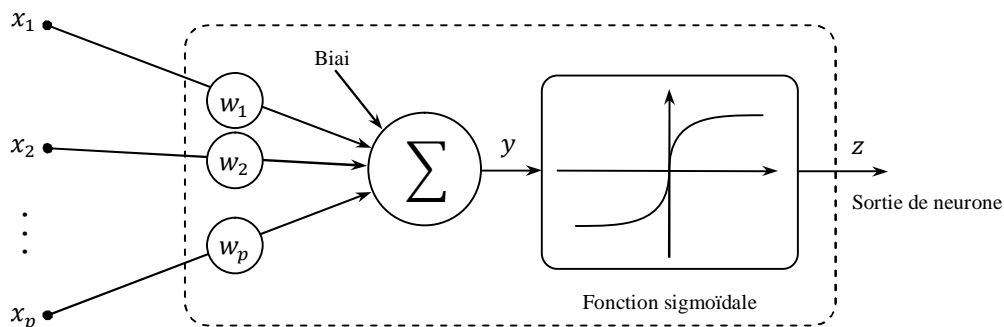


Figure III.2 : Le perceptron

Le perceptron se caractérise par une fonction d'activation continue et différentiable :

$$z = \frac{2}{1+e^{-By}} - 1$$

3.3. INTERPRETATION MATHÉMATIQUE

D'un point de vue mathématique, le neurone formel peut être représenté de la manière suivante:

Un neurone à p entrée est une fonction $f: \mathbb{R}^{p+1} \times \mathbb{R}^n \rightarrow \mathbb{R}$ définie par :

1. $g: \mathbb{R} \rightarrow \mathbb{R}$
2. $W \in \mathbb{R}^{p+1}, W = (w_1, w_2, \dots, w_{p+1})$
3. $\forall x \in \mathbb{R}^p, f(W, x) = g(\sum_{i=1}^p w_i x_i + w_{p+1})$
avec $x = (x_1, x_2, \dots, x_p)$

Cette définition est inspirée du neurone biologique, les poids jouant le rôle de synapses, le vecteur x celui des entrées et W celui des coefficients ou poids. Le coefficient w_{p+1} est appelé le biais et souvent noté b . La fonction g est appelée fonction de d'activation ou de transfert ou

encore fonction de seuil. A cette définition mathématique du neurone, une autre moins formelle et plus graphique (Figure III.2) lui est préférée. Ce schéma est également plus proche de sa définition biologique et dissocie mieux les rôles non symétriques des entrées et des poids. Des exemples de fonctions de transfert sont donnés par le Tableau III.1 dont les plus couramment utilisées sont les fonctions linéaire et sigmoïde.

La plupart des fonctions utilisées sont dérivables et cette propriété s'étend à tout assemblage de neurones, ce qui permet d'utiliser l'algorithme de rétro propagation découvert par [RUM86]. Ce dernier permet le calcul de la dérivée ouvre ainsi les portes des méthodes d'optimisation basées sur cette propriété.

Le vecteur $(x_1, x_2, \dots, x_p) \in \mathbb{R}^p$ joue le rôle des entrées.

y est appelé parfois le *potentiel*.

$$y = \sum_{i=1}^p w_i x_i + b$$

z est appelé la sortie du neurone.

f est appelée la fonction d'activation ou de transfert.

$$z = f(y) = f\left(\sum_{i=1}^p w_i x_i + b\right)$$

exemples de fonction de transfert ou de seuil	expression : $f(x) =$
Binaire	$\begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$
Signe	$\begin{cases} -1 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$
Linéaire	x
sigmoïde entre $[0, 1]$	$\frac{1}{1 + e^{-x}}$
sigmoïde entre $[-1, 1]$	$\frac{2}{1 + e^{-x}} - 1$

Tableau III.1 : Fonctions d'activation usuelles.

3.4. TYPES ET ARCHITECTURES DES RNA

3.4.1. Le Perceptron Multicouches

Les perceptrons multicouches sont les réseaux de neurones les plus populaires, du fait de leurs bonnes performances et de leur facilité de mise en œuvre.

Ce sont des réseaux de neurones à propagation directe sans cycles, avec au moins une couche cachée. La fonction de transfert des neurones est généralement de type sigmoïde ou \tanh à valeurs dans $[0; 1]$ ou dans $[-1; 1]$.

Un réseau de neurones multicouches à n sorties, p entrées et C couches est une liste de couches (C_1, \dots, C_C) connectées les unes aux autres de telle sorte que :

1. $\forall i \in \{1, \dots, C\}$, chaque couche C_i possède n_i neurones et p_i entrées
2. $\forall i \in \{1, \dots, C - 1\}$, $n_i = p_{i+1}$, de plus $p_1 = p$ et $n_C = n$
3. les coefficients de la couche C_i sont notés $(W_1^i, \dots, W_{n_i}^i)$ cette couche définit une fonction F_i
4. soit la suite $(Z_i)_{0 \leq i \leq C}$ définie par : $Z_0 \in \forall i \in \{1, \dots, C\}, Z_i = F_i(W_1^i, \dots, W_{n_i}^i, Z_{i-1})$

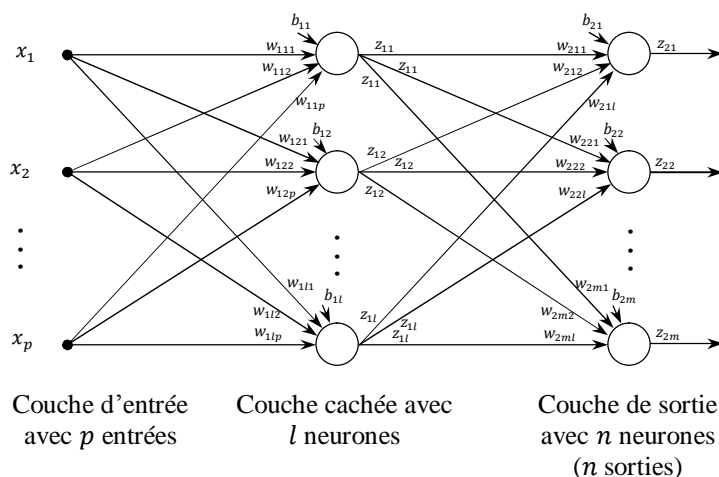


Figure III.3 : Un perceptron à une couche cachée unique

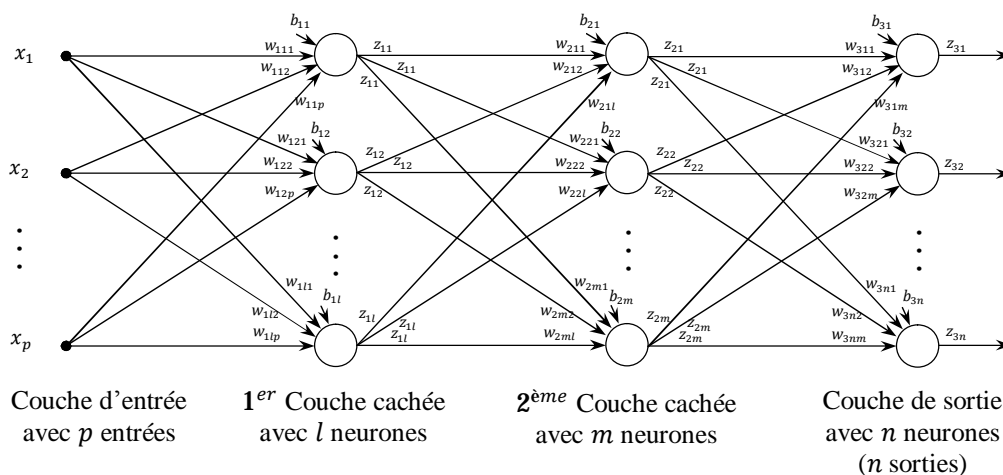


Figure III.4 : Un perceptron à deux couches cachées

Souvent, on considère que les entrées forment la couche C_0 de manière à simplifier les écritures. Ainsi, chaque couche C_i du perceptron a pour entrées les sorties de la couche C_{i-1} . Cette définition est plus facile à illustrer qu'à énoncer (Figure III.3 et Figure III.4).

Le mécanisme qui permet de calculer les sorties d'un réseau de neurones sachant ses poids est appelé *propagation*.

(x_1, \dots, x_p)	Les entrées
C	Nombre des couches
C_i	Nombre de neurones sur la couche i ($C_0 = p$)
$z_{c,i}$	Sortie du neurone i de la couche c par extension, $z_{0,i} = x_i$
$y_{c,i}$	Potentiel du neurone i de la couche c
$w_{c,i,j}$	Coefficient associé à l'entrée j du neurone i de la couche c
$b_{c,i}$	Biais du neurone i de la couche c
$f_{c,i}$	Fonction de seuil du neurone i de la couche c

3.4.1.1. La Propagation

Cet algorithme s'applique à un réseau de neurones pour calculer les sorties de ce réseau connaissant ses poids ($w_{c,i,j}$) et ses entrées (x_j).

Etape A : initialisation

pour $i = 1$ à C_0 **faire**

$$z_{0,i} \leftarrow x_i$$

fin pour

Etape B : Propagation

pour $c = 1$ à C **faire**

pour $i = 1$ à C_c **faire**

$$z_{c,i} \leftarrow 0$$

pour $j = 1$ à C_{i-1} **faire**

$$z_{c,i} \leftarrow z_{c,i} + w_{c,i,j} z_{c-1,j}$$

fin pour

$$z_{c,i} \leftarrow f(z_{c,i} + b_{c,i})$$

fin pour

fin pour

Les perceptrons multicouches définissent une classe de fonctions qui permet d'approximer n'importe quelle fonction continue à support compact.

3.4.2. Fonctions à bases radiales

Les réseaux de neurones de type RBF (Radial Basis Function) ont généralement une couche cachée. Les neurones de cette couche cachée sont de types gaussiens. Les neurones de sorties sont une combinaison linéaire des neurones gaussiens de la couche cachée, parfois suivis d'une fonction de transfert sigmoïde.

Les caractéristiques des neurones gaussiens (centres et variances) sont généralement estimées au début de l'apprentissage. Ces paramètres sont ensuite figés. L'apprentissage du réseau de neurones consiste à apprendre la couche de combinaison linéaire de sortie, par descente de gradient.

3.5. L'APPRENTISSAGE

L'apprentissage peut être considéré comme le problème de la mise à jour des poids des connexions au sein du réseau afin de réussir la tâche qui lui est demandée. L'apprentissage est la caractéristique principale de RNA, il peut se faire de différentes manières et selon de différentes règles.

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissage. Dans le cas de l'apprentissage supervisé, les exemples sont des couples (Entrée, Sortie associée) alors que l'on ne dispose que des valeurs (Entrée) pour l'apprentissage non supervisé. Remarquons cependant que les modèles à apprentissage non supervisé nécessitent avant la phase d'utilisation une étape de labellisation effectuée l'opérateur, qui n'est pas autre chose qu'une part de supervision.

3.5.1. La loi d'apprentissage de Hebb [HEB49]

S'il y a une couche de neurones qui partagent p entrées, la sortie du neurone j est $z_j = f_j(y_j) = f_j(\sum_{i=1}^{p+1} w_{ij}x_i)$, w_{p+1} représente le biais, la modification des poids est définie comme suite :

$$\Delta w_{ij} = \mu x_i z_j$$

$$w_{ij(t+1)} = w_{ij(t)} + \Delta w_{ij(t)}$$

Avec :

- z_j est la sortie du neurone j
- $w_{ij(t+1)}$ est le nouveau poids i du neurone j , $w_{ij(t)}$ est l'ancien poids i du neurone j
- μ représente le coefficient d'apprentissage

L'algorithme d'apprentissage modifie de façon itérative les poids pour adapter la réponse obtenue à la réponse désirée. Il s'agit en fait de modifier les poids lorsqu'il y a erreur seulement.

La règle d'apprentissage de Hebb est une règle d'apprentissage non supervisé.

3.5.2. La règle d'apprentissage du Perceptron

La règle d'apprentissage du perceptron [ROS58] s'applique sur un neurone de type signe, elle est définie comme suite :

$$\Delta w_{ij} = \mu x_i e_j$$

$$e_j = d_j - z_j = \text{signe} \left(\sum_{i=1}^{p+1} w_{ij} x_i \right)$$

Avec :

- e_j est l'erreur de la sortie j
- d_j est la sortie désirée j

L'apprentissage avec cette règle est un apprentissage supervisé.

3.5.3. Retro-propagation du gradient

L'algorithme de rétro-propagation a été développé en particulier par [RUM86]. Cet algorithme repose sur la minimisation de l'erreur quadratique entre les sorties calculées et celles souhaitées.

Le terme rétro-propagation du gradient provient du fait que l'erreur calculée en sortie est transmise en sens inverse vers l'entrée.

$$\Delta w_{c,i,j} = \mu f'(y_{c,i}) \delta_{c,i} z_{c-1,j}$$

Si c représente une couche de sortie alors : $\delta_{c,i} = d_i - z_{c,i}$

Si c représente une couche cachée alors : $\delta_{c,i} = \sum_{j=1}^{C_{c+1}} w_{j,i} \delta_{c-1,j}$

3.6. AVANTAGES ET INCONVENIENTS DES RESEAUX NEURONAUX

L'intérêt porté aux réseaux de neurones tient sa justification dans les propriétés intéressantes qu'ils possèdent; ils présentent aussi un certain nombre de limitations ou d'inconvénients. A partir de ces propriétés, les applications potentielles de l'approche neuronale peuvent être déduites [DAV93], [MEI00].

3.6.1. Avantages de l'approche neuronale

3.6.1.1. L'exploitation de connaissances empiriques :

L'apprentissage à partir d'exemples (méthode d'apprentissage empirique) se fait d'une façon assez simple et permet d'obtenir de bons résultats par rapport aux autres techniques d'apprentissage automatique.

3.6.1.2. La robustesse :

Dans les réseaux de neurones, la mémoire est distribuée, elle correspond à une carte d'activation de neurones. Cette carte est en quelque sorte un codage des faits mémorisés ce qui attribue à ces réseaux l'avantage de résister aux bruits (pannes) car la perte d'un élément ne correspond pas à la perte d'un fait mémorisé. Ainsi, un réseau peut bien fonctionner même quand des unités sont en panne. De plus, de nombreux modèles de réseaux donnent de bons résultats même quand leurs entrées sont bruitées.

3.6.1.3. La dégradation progressive :

Les réseaux, de par leur nature continue, ne fonctionnent pas en tout ou rien et leurs performances ont plutôt tendance à diminuer progressivement en cas de problème (bruit, panne, entrée inconnue...). Cette propriété est très recherchée car les systèmes cognitifs vivants montrent une telle faculté. Les réseaux permettent de bien généraliser les connaissances présentes dans la base d'apprentissage et sont moins sensibles aux perturbations que les systèmes symboliques. Le fait de travailler sur une représentation numérique des connaissances rend les réseaux plus adaptés pour manipuler des données quantitatives (valeurs continues). Les réseaux de neurones sont moins vulnérables aux données approximatives et à la présence de données incorrectes dans la base d'apprentissage.

3.6.1.4. Le parallélisme massif :

Les réseaux sont composés d'un ensemble d'unités de traitement de l'information qui peuvent opérer en parallèle. Bien que la plupart des implémentations des réseaux connexionnistes soient réalisées sur des simulateurs séquentiels, il est possible de faire des implémentations (logicielles ou matérielles) exploitant la possibilité d'activer simultanément les unités. La plupart des implémentations des réseaux de neurones peuvent être facilement converties d'une version séquentielle vers une version parallèle.

3.6.1.5. La prise en compte du non linéarité et du temps :

Les réseaux de neurones artificiels présentent l'intérêt de pouvoir prendre en compte la non linéarité (les fonctions d'activation sont en général non linéaires). Certains réseaux peuvent aussi prendre en compte les aspects temporels (cas des réseaux récurrents).

3.6.2. Inconvénients de l'approche neuronale

3.6.2.1. La difficulté de choix de l'architecture et des paramètres :

Il n'existe pas de méthode automatique pour choisir la meilleure architecture possible pour un problème donné. Il est assez difficile de trouver la bonne topologie du réseau ainsi que les bons paramètres de réglage de l'algorithme d'apprentissage. L'évolution du processus d'apprentissage est très influencée par ces deux éléments (l'architecture du réseau et les paramètres de réglage) et dépend beaucoup du type de problème traité. Le simple fait de changer la base d'apprentissage utilisée, peut nous obliger à reconfigurer le réseau en entier.

3.6.2.2. Le problème d'initialisation et de codage :

Les algorithmes d'apprentissage connexionniste sont en général très dépendants de l'état initial du réseau (initialisation aléatoire des poids) et de la configuration de la base d'apprentissage. Un mauvais choix des poids employés pour initialiser le réseau, de la méthode de codage des données, ou même de l'ordre des données, peut bloquer l'apprentissage ou poser des problèmes pour la convergence du réseau vers une bonne solution;

3.6.2.3. Le manque d'explicabilité :

Les connaissances acquises par le réseau sont codées par l'ensemble des valeurs des poids synaptiques ainsi que par la façon dont les unités sont interconnectées. Il est très difficile pour un être humain de les interpréter directement. Les réseaux connexionnistes sont en général des boîtes noires, où les connaissances restent enfermées et sont inintelligibles pour l'utilisateur ou pour l'expert. Un réseau ne peut pas expliquer le raisonnement qui l'a amené à une solution spécifique.

3.6.2.4. Le manque d'exploitation de connaissances théoriques :

Les réseaux classiques ne permettent pas de profiter des connaissances théoriques disponibles sur le domaine du problème traité. Ils sont dédiés à la manipulation de connaissances empiriques. Une façon simpliste de profiter des connaissances théoriques consiste à convertir des règles en exemples (prototypes). Cependant, cette méthode ne garantit pas que ces exemples aillent être bien représentés dans les connaissances du réseau à la fin de l'apprentissage, car nous sommes obligés à passer pour une phase d'apprentissage où se mélangent sans distinction des connaissances empiriques avec des connaissances théoriques codées par des exemples.

4. K PLUS PROCHES VOISINS

4.1. INTRODUCTION

L'algorithme des k plus proches voisins (KPPV) est certainement un des algorithmes les plus simples d'apprentissage automatique, il a été proposé par [COV91]. Il est motivé par l'idée que des entrées x_t semblables devraient avoir des cibles y_t semblables. Ainsi, pour bien définir un algorithme KPPV, il suffit de définir ce que veut dire "semblable" dans le contexte des entrées et de définir l'influence de ces voisins sur la prédiction de la cible pour une entrée de test.

Donc, pour obtenir une prédiction de la cible pour une entrée de test x , il suffit de trouver les k plus proches voisins selon une métrique déterminant jusqu'à quel point des entrées sont semblables (par exemple, la distance euclidienne ou norme L^2 , ou de façon plus générale la norme L^p de Minkowski) et d'utiliser ces k plus proches voisins pour prédire la cible de x . Dans un problème de classification, la prédiction correspond à la classe majoritaire parmi les k plus proches voisins, i.e. que l'ensemble des k plus proches voisins votent pour la classe correspondant à leur cible respective et la classe recueillant le plus de vote est choisie en tant que prédiction par l'algorithme.

4.2. FORMALISATION MATHÉMATIQUE

Soit :

- $T = \{(x_i, y_i) / i \in \{1, \dots, c\}\}$ un ensemble d'apprentissage, où $y_i \in Y = \{0, \dots, c - 1\}$ correspond à l'identité de la classe cible de l'entrée x_i .
- c le nombre de classes
- $d(.,.)$ une fonction de distance
- x une entrée de test
- $V = (x, T, d(.,.), k)$ l'ensemble des k plus proches voisins de x parmi les entrées de T ainsi que leur cible associée

La prédiction de classification par l'algorithme des k plus proches voisins est donc:

$$\operatorname{argmax}_{y \in Y} \sum_{(x_i, y_i) \in (x, T, d(.,.), k)} I_{y_i=y}$$

Une fonction de distance couramment utilisée est la distance euclidienne:

$$d(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

4.3. ALGORITHME

On définit un algorithme d'apprentissage en précisant sa procédure d'entraînement et de prédiction pour une nouvelle entrée de test. Etant donné que la procédure d'entraînement de l'algorithme KPPV consiste simplement à mettre en mémoire l'ensemble d'apprentissage T, voici donc la procédure de KKPV [LAR07][COV91]:

```

pour  $i \leftarrow 0$  à  $C - 1$ 
  faire  $c[i] \leftarrow 0$ 

pour  $i \leftarrow 0$  à  $k - 1$ 
  faire  $\begin{cases} \text{voisin}[i] \leftarrow \text{NULL} \\ \text{dist}[i] \leftarrow \text{inf} \end{cases}$ 

pour  $i \leftarrow 0$  à  $k - 1$ 
  faire  $\begin{cases} dt \leftarrow d(x_t, x) \\ j \leftarrow \text{argmax}_i \text{dists}[i] \\ \text{si } dt < \text{dists}[j] \\ \text{alors } \begin{cases} (x_i, y_i) \leftarrow \text{voisins}[i] \\ c[y_i] \leftarrow c[y_i] + 1 \end{cases} \end{cases}$ 

retourner  $(\text{argmax}_i c[i])$ 

```

5. MODELES DE MARKOV CACHES

5.1. INTRODUCTION

Les techniques de classification des séquences d'observations sont pour beaucoup différentes de celles employées pour des formes discrètes en nombre de classes fini. La principale différence est le fait que deux séquences d'une même classe ne sont pas forcément de longueurs identiques.

Les modèles de Markov cachés (Hidden Markov Models ou HMMs) ont été décrits avec leurs méthodes d'apprentissage et de décodage dès 1966 [PET66]. Ce sont ces algorithmes qui ont permis d'appliquer efficacement cet outil de modélisation. Le formalisme des MMC a

permis des progrès importants en reconnaissance automatique de la parole ces trente dernières années. Leur formalisme probabiliste et leur capacité de modélisation des séquences en ont fait un instrument quasiment universel dans ce domaine. Ces méthodes s'imposent depuis plus de dix ans dans d'autres domaines et notamment dans la reconnaissance de l'écriture manuscrite en ligne et hors ligne.

L'une des difficultés de la reconnaissance de la parole ou de l'écriture est sa segmentation en sous séquences stationnaires, pour leur reconnaissance. Or, selon la contradiction bien connue, il n'est pas possible de segmenter sans avoir reconnu les parties, ni de les reconnaître sans avoir segmenté la séquence. Les MMC permettent d'appliquer le paradigme de segmentation-reconnaissance simultanées. Après reconnaissance complète, il est possible de faire une segmentation globalement optimale du signal en zones stationnaires modélisées. C'est l'alignement temporel du signal sur une séquence de labels (phonèmes ou graphèmes, mots ...) [AUG01].

5.2. DEFINITIONS

5.2.1. Chaîne de Markov

Une chaîne de Markov discrète d'ordre n est un processus stochastique discret $X = \{X_t | t = 1, \dots, T\}$ avec des variables aléatoires discrètes (les réalisations de ces variables sont appelées états) [HAL04]:

$$P(X_t = q_{i_t} | X_{t-1} = q_{i_{t-1}}, \dots, X_1 = q_{i_1}) = P(X_t = q_{i_t} | X_{t-1} = q_{i_{t-1}}, \dots, X_{t-n} = q_{i_{t-n}})$$

$$\forall t \in [1, T]$$

Où $Q = \{q_1, \dots, q_n\}$ représente l'ensemble des états.

Dans la suite on n'utilise que les chaînes de Markov d'ordre 1.

5.2.2. Chaîne stationnaire

Une chaîne de Markov d'ordre 1 est stationnaire si pour tout t et k on a :

$$P(X_t = q_{i_t} | X_{t-1} = q_j) = P(X_{t+k} = q_i | X_{t+k-1} = q_j)$$

Dans ce cas, on définit une matrice de probabilité de transition $A = (a_{ij})$ telle que:

$$a_{ij} = P(X_t = q_j | X_{t-1} = q_i)$$

À un instant donné d'un processus quelconque.

5.3. UN MODELE DE MARKOV CACHE

Un modèle de Markov caché (HMM) est une chaîne de Markov cachée stationnaire où l'observation est une fonction probabiliste de l'état.

La le schéma d'un HMM.

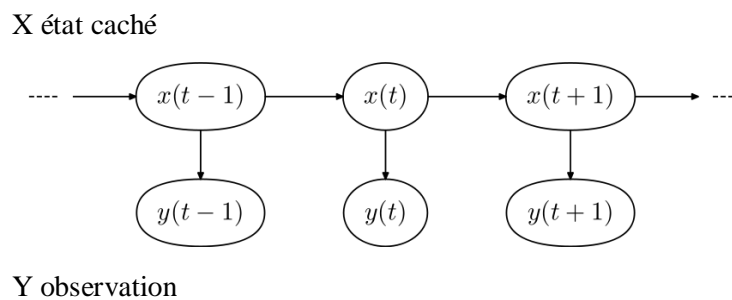


Figure III.5 : Modèle de Markov Caché

5.4. INTERPRETATION MATHEMATIQUE

Selon [DUP04] Un HMM discret du premier ordre est défini par:

- N états représentés par $Q = \{q_1, q_2, \dots, q_N\}$. On rappelle que l'on désigne un état au temps t par $q_t \in Q$.
- M symboles observés dans chaque état $V = \{o_1, o_2, \dots, o_M\}$. On désigne une observation au temps t par $O_t \in V$.
- La matrice de probabilité de transition entre les états $A = (a_{ij})_{1 \leq i, j \leq N}$ avec

$$a_{ij} = P(X_t = q_j | X_{t-1} = q_i) \quad \forall i, j \in [1, N]$$

$$a_{ij} \geq 0$$

$$\sum_{j=1}^N a_{ij} = 1 \quad \forall i \in [1, N]$$

5.5. LES TYPES DES HMMS

Selon la topologie du réseau des états, il y a deux types de HMMS. Ce sont le modèle ergodique et le modèle gauche-droite [HAL04].

5.5.1. HMM ergodique

C'est un modèle sans contraintes où toutes les transitions d'un état vers l'autre sont possibles, c'est à dire $a_{ij} > 0 \quad \forall (i, j) \in [1, N]$.

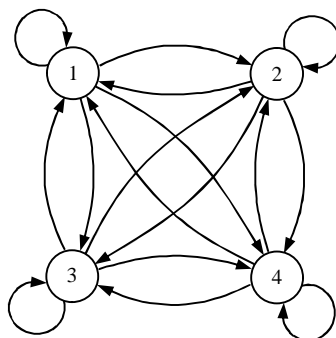


Figure III.6 : Modèle ergodique

5.5.2. HMM gauche-droite

C'est un modèle où il y a des contraintes sur des transitions : seulement la transition d'un état ayant un indice bas vers un état ayant un indice haut est acceptée. Il existe deux sous-types de ce modèle : le modèle parallèle et le modèle séquentiel.

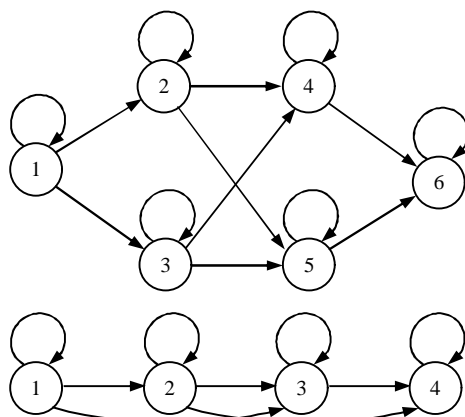


Figure III.7 : modèles gauche-droite

Le modèle séquentiel est souvent utilisé pour reconnaître la parole et l'écriture manuscrite car il est convenable à la modélisation des signaux et il y a moins de paramètres à estimer dans la phase de l'apprentissage. Dans ce modèle, la densité de probabilité initiale π n'est pas nécessaire [HAL04].

Selon les types des symboles des observations on a les HMMs discrets et les HMMs semi-continus.

6. CONCLUSION

Nous avons vu dans ce chapitre les réseaux de neurones et leurs principes de bases, ainsi que l'algorithme de KPPV, on va alors passer au quatrième chapitre où nous étudierons les techniques de combinaison des classifieurs d'une manière parallèle pour les utiliser ultérieurement dans la combinaison.

CHAPITRE IV

LA COMBINAISON DES SYSTÈMES DE CLASSIFICATION

Dans ce chapitre:

- ✓ Stratégies de combinaison.
- ✓ Taxonomies des méthodes de combinaison.

1. INTRODUCTION

Dans ce chapitre nous allons présenter un état de l'art des systèmes de combinaison de classifieurs. Nous présentons les stratégies de prises de décision possibles lorsque plusieurs classifieurs interviennent dans un processus de reconnaissance et les spécificités relatives à chaque stratégie. Nous nous focalisons en particulier sur la combinaison parallèle de classifieurs. Nous rappelons tout d'abord les différentes stratégies de combinaison, puis la combinaison parallèle des classifieurs de types classe, rang et mesure.

2. STRATEGIES DE COMBINAISON

2.1. APPROCHE SEQUENTIELLE

La combinaison séquentielle, appelée également combinaison série, est organisée en niveaux successifs de décision permettant de réduire progressivement le nombre de classes possibles. Dans chaque niveau, il existe un seul classifieur qui prend en compte la réponse fournie par le classifieur placé en amont afin de traiter les rejets ou confirmer la décision obtenue sur la forme qui lui est présentée (Figure IV.1). Une telle approche peut être vue comme un filtrage progressif des décisions dans la mesure où elle permet de diminuer au fur et à mesure l'ambiguïté sur la classe proposée. Cela permet généralement de diminuer le taux d'erreur globale de la chaîne de reconnaissance. Néanmoins, une combinaison de ce type demeure particulièrement sensible à l'ordre dans lequel sont placés les classifieurs. En effet, même s'ils ne nécessitent pas d'être les plus performants, les premiers classifieurs invoqués doivent être robustes, c'est-à-dire que la solution réelle de la forme à identifier doit apparaître dans les listes successives quelle que soit leur taille. En cas de mauvaise décision du premier classifieur, placé en amont de la série des classifieurs utilisés, l'erreur va se propager de façon irrévocable. Il faudra donc choisir judicieusement le premier classifieur afin d'éviter - autant que possible - l'apparition d'une telle situation. La combinaison séquentielle suppose donc une certaine connaissance a priori du comportement de chacun des classifieurs. Notons que dans cette approche, chaque classifieur est réglé en fonction du classifieur placé en amont de la chaîne. Une simple modification du premier classifieur peut provoquer un ré-paramétrage (réapprentissage) des classifieurs suivants [ZOU04].

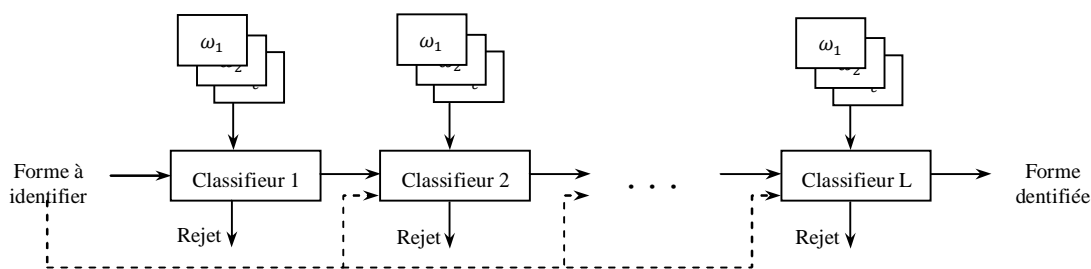


Figure IV.1 : Combinaison séquentielle de classifieurs

Le but ici n'est pas de décrire en détail les systèmes de combinaison séquentielle. Nous présentons maintenant deux exemples qui illustrent leur principe de fonctionnement.

L'approche proposée dans [GAD91] est basée sur trois étages de décision. Les deux premiers étages mettent en œuvre une comparaison directe du caractère à reconnaître avec l'ensemble des modèles, permettent de classer 70 à 80% des chiffres avec un taux d'erreur faible et sont capables de générer des décisions sur les classes d'appartenance des chiffres rejetés. Lorsque ces étages ne peuvent pas conclure, ils fournissent une liste d'hypothèses au dernier niveau de décision pour chercher le modèle dans une liste prédéfinie de modèles syntaxiques. Récemment, [PRE03] présentent un système composé de deux étages pour améliorer la reconnaissance de caractères manuscrits. Le premier étage est un classifieur non supervisé qui fournit des scores à chacune des classes. Le second étage est un classifieur neuronal qui sépare les paires de classes les plus ambiguës. Ce système séquentiel est basé sur l'idée que la classe correcte est systématiquement parmi les deux premières classes (celles ayant les probabilités les plus élevées) proposées par le premier classifieur. Les résultats expérimentaux montrent une amélioration de 30% par rapport à chacun des classifieurs utilisés pour une réponse de type classe dans un problème à 62 classes.

2.2. APPROCHE PARALLELE

A la différence de l'approche séquentielle, l'approche parallèle laisse dans un premier temps les différents classifieurs opérer indépendamment les uns des autres puis fusionne leurs réponses respectives. Cette fusion est faite soit de manière démocratique, dans le sens où elle ne favorise aucun classifieur par rapport à un autre, soit au contraire dirigée et, dans ce cas, on attribue à la réponse de chaque classifieur un poids en fonction de ses performances. L'ordre d'exécution des classifieurs n'intervient pas dans cette approche. La Figure IV.2 fournit une représentation de la combinaison parallèle de classifieurs[ZOU04].

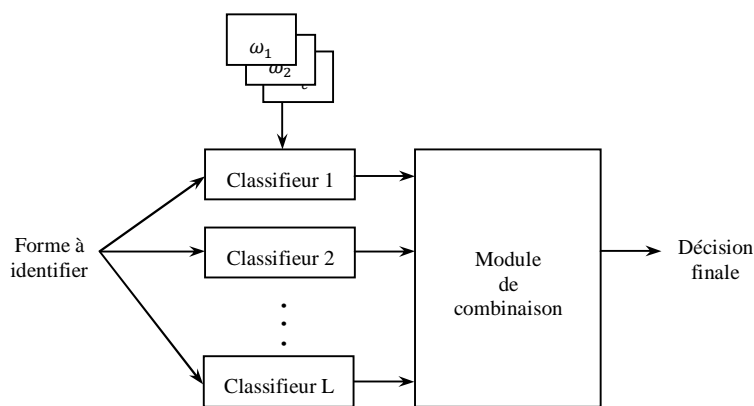


Figure IV.2 : Combinaison parallèle de classifieurs

L'inconvénient majeur de l'approche parallèle est qu'elle nécessite l'activation de tous les classifieurs du système qui doivent participer de manière concurrente et indépendante. Par contre, la décision finale est prise avec le maximum de connaissances mises à disposition par chaque classifieur. Dès lors se posent les problèmes de précision des informations fournies par les classificateurs et de la confiance qu'on peut accorder à chacun d'eux [ZOU04].

L'approche proposée dans [HUA95] nécessite que chacun des classifieurs fournisse une confiance (probabilité ou distance) associée à chaque proposition ou classe. La décision finale est prise dans un réseau de neurones à partir de la combinaison des différents résultats fournis par les classifieurs. Pour améliorer la reconnaissance de mots, [KIM00] proposent de combiner deux classifieurs, l'un de type HMM (Hidden Markov Model), l'autre de type MLP (Multi-Layer Perceptron). L'idée ici est que pour augmenter la complémentarité, les classifieurs doivent opérer avec des structures différentes. Les sorties du classifieur HMM sont normalisées avant la combinaison pour pouvoir les fusionner avec les sorties du MLP.

2.3. APPROCHE HYBRIDE

L'approche hybride consiste à combiner à la fois des architectures séquentielles et parallèles afin de tirer pleinement avantage de chacun des classifieurs utilisés. La Figure IV.3 présente un exemple de combinaison hybride dans laquelle on combine un classifieur en série avec deux classifieurs en parallèle.

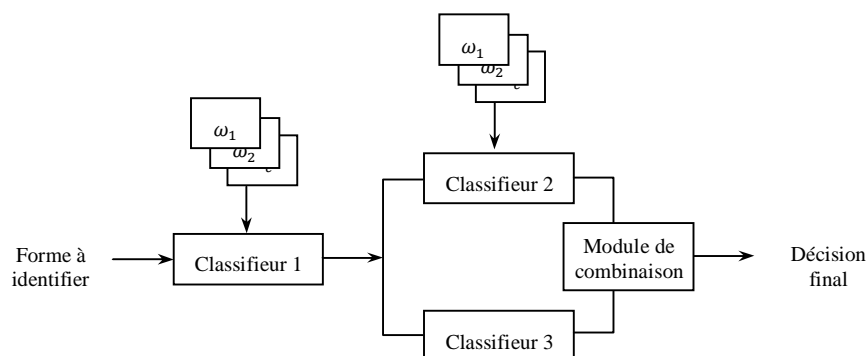


Figure IV.3 : Combinaison hybride de classificateurs

On peut citer dans ce cadre, les travaux de [KIM00] qui propose un système de reconnaissance de mots cursifs anglais extraits des chèques bancaires. Ce système hybride est composé de deux étages. Dans le premier étage, deux classificateurs (PMC) utilisant des vecteurs de caractéristiques différents sont combinés par un autre classificateur de même type. La coopération de ce dernier avec un autre classificateur de type HMM est réalisée dans l'étage suivant par une règle de multiplication. Un autre exemple est celui présenté dans [BEL02]. Il décrit un système de reconnaissance de chiffres manuscrits par combinaison hybride de réseaux neuronaux de type MLP et de machines à vecteurs de support SVM. Cette méthode de combinaison consiste à introduire des classificateurs SVM spécialisés pour chaque paire de classes numériques (0 à 9) uniquement dans le voisinage des surfaces de séparation générées par le réseau MLP entre les exemples d'apprentissage de ces mêmes paires de classes. Cette architecture de combinaison est fondée sur la constatation que les deux premières solutions de la couche de sortie du MLP contiennent presque systématiquement la bonne classe de la forme à classifier et que certaines paires de classes constituent la majorité des confusions générées par le MLP. Les SVM sont introduits pour détecter la bonne classe parmi les deux meilleures hypothèses de classification fournies par le réseau. Ce choix se résume à un problème de classification à deux classes (binaire). Cependant, cette méthode peut sembler fastidieuse car elle nécessite un classificateur SVM pour chaque paire de classes. Une seconde originalité de cette méthode réside dans l'introduction de SVM uniquement pour les paires de classes qui constituent la majorité des confusions (erreurs) du réseau MLP. Certains auteurs ont proposé d'effectuer des combinaisons conditionnelles. Ainsi Gasselin [GOS97] propose de classer les classificateurs selon leur performance et de traiter une forme inconnue par le premier classificateur. Il propose d'accepter sa décision, si la forme n'est pas rejetée. Dans le cas contraire, la décision sera prise suite à la combinaison du premier classificateur avec la sortie du deuxième classificateur. Le même raisonnement peut s'appliquer, jusqu'à ce que la forme soit classée ou que les sorties

de tous les classifieurs soient combinées. Cette combinaison conditionnelle permet de réduire efficacement les temps d'exécution. L'inconvénient est la nécessité de fixer plusieurs seuils de rejet associés aux différents niveaux [ZOU04].

3. TAXONOMIES DES METHODES DE COMBINAISON

3.1. COMBINAISON DE CLASSIFIEURS DE TYPE CLASSE

3.1.1. Vote avec seuil

Les méthodes de vote peuvent pratiquement toutes être dérivées de la règle avec seuil exprimée par:

$$E(x) = \begin{cases} \omega_i & \text{si } \sum_{j=1}^L e_{i,j} = \max_{t=1}^c \sum_{j=1}^L e_{t,j} \geq \lambda L \\ \text{rejet} & \text{sinon} \end{cases}$$

λ correspond à la proportion de classifieurs devant répondre la même classe pour que cette classe soit retenue comme résultat de la combinaison. Ainsi, pour $\lambda = 0$, il s'agit du vote à la pluralité où la classe qui reçoit le plus de votes est choisie comme classe finale. Connue aussi sous le nom First past the post [CRA96], cette méthode a la forme de vote la plus simple et est donc la plus facile à appliquer. Néanmoins, il y a rejet si toutes les classes ont le même nombre de votes. Dans ce cas, les risques de conflit sont particulièrement importants.

3.1.2. Majorité sans conflit

Le principe est identique à celui de la majorité unanime, à ceci près qu'on autorise les classifieurs à rejeter. Un classifieur proposant de rejeter l'élément n'a donc aucun poids dans ce système: lorsque tous les classifieurs rejettent sauf un, alors c'est la sortie de ce classifieur qui sera conservée comme résultat de la combinaison. Il n'y a rejet que lorsque tous les classifieurs ont proposé de rejeter l'élément ou en cas de conflit.

3.1.3. Majorité notoire

Dans ce cas, pour être désignée comme réponse finale, la classe majoritaire doit de plus se distinguer de la deuxième classe d'une différence supérieure à un certain seuil. Les résultats de tests dans [XUL92] montrent que cette méthode est meilleure que la majorité avec seuil surtout quand un minimum de fiabilité est exigé.

3.1.4. Vote avec pondération

Dans cette méthode, la réponse $e_{i,j}$ de chaque classifieur e_j est pondérée par un coefficient W_j indiquant son importance dans la combinaison.

$$E(x) = \begin{cases} \omega_i & \text{si } \sum_{j=1}^L w_j e_{i,j} = \max_{t=1}^c \sum_{j=1}^L w_j e_{t,j} \geq \lambda L \\ \text{rejet} & \text{sinon} \end{cases}$$

Il existe plusieurs façons de déterminer les coefficients W_j . Ils peuvent avoir été optimisés par algorithme génétique [LAM94]. Dans [ACH96], la forme d'entrée est attribuée à la classe pour laquelle la somme des votes qui sont pondérés par la fiabilité $\left(\frac{\text{taux de reconnaissance}}{100 - \text{taux de rejet}}\right)$ estimée de chacun des experts, est la plus élevée. Cette méthode n'introduit pas de critère de rejet.

3.2. COMBINAISON DE CLASSIFIEURS DE TYPE RANG

3.2.1. Borda Count et ses variantes

Il s'agit d'une procédure de vote proposée en 1770 par Jean Charles de Borda [BOR81] dans laquelle chaque voteur arrange les c candidats selon sa préférence en attribuant un nombre de points à chacun. Le candidat préféré reçoit $c - 1$ points, le candidat suivant reçoit $c - 2$, ainsi de suite. Le nombre de points du candidat placé à la fin de la liste est 0 . La règle de Borda offre donc un seul ensemble de rangs "échelle", $c - 1, c - 2, \dots, 0$. Certains auteurs attribuent les rangs autrement: c points à la classe placée en tête (top) de chaque liste de classifieurs, $c - 1$ à la classe suivante et 1 à la dernière classe [VAN00]. Pour un problème à deux classes, le Borda Count est un vote à la majorité. Pour une classe particulière ω_i ($i = 1, \dots, c$), le Borda Count BC (ω_i) est la somme de tous les rangs proposés par les classifieurs pour cette classe.

$$BC(\omega_i) = \sum_{j=1}^L r_{i,j}$$

$r_{i,j}$ est le rang attribué par le classifieur e_j à la classe ω_i . $r_{i,j} = c$ si la classe ω_i est placée au début de la liste. $r_{i,j} = c - 1$ si la classe ω_i est placée à la place suivante de la liste, etc. Les classes sont ensuite triées selon leur rang total. Le nouvel ordre détermine les rangs finaux.

3.2.2. Meilleur rang

La méthode du meilleur rang consiste à attribuer à chacune des classes le rang le plus élevé $MR(\omega_i)$ parmi les rangs proposés par les classifieurs et d'ordonner la liste selon ces rangs [Ho, 1992].

$$MR(\omega_i) = \max_{j=1}^L r_{i,j}$$

Cette méthode est adaptée pour des problèmes à plusieurs classes et peu de classifieurs non corrélés. Son avantage est qu'elle utilise la meilleure réponse de chaque classifieur. Cependant, il apparaît que cette méthode peut facilement provoquer des conflits lorsque plusieurs classes obtiennent le même meilleur rang. Ce problème peut être résolu en utilisant des critères supplémentaires [ZOU04].

3.2.3. Somme pondérée

Dans cette méthode, les rangs attribués par les classifieurs pour une classe ω_i sont pondérés par des coefficients W_j indiquant la crédibilité accordée à chaque classifieur e_j . Il s'agit d'une généralisation du borda count (lorsque $W = \mathbf{1}$ on obtient la règle du borda count). La somme pondérée (SP) d'une classe ω_i s'écrit:

$$SP(\omega_i) = \sum_{j=1}^L w_j r_{i,j}$$

La méthode proposée dans [VER01] permet de prendre en compte la différence entre les classifieurs pour améliorer la reconnaissance de mots. Il s'agit d'une modification du borda count (MBC). Pour L classifieurs proposant chacun des listes à K solutions mots ($K \leq c$), le MBC de chaque classe ω_i est calculé comme suit:

$$MBC(\omega_i) = \sum_{j=1}^L w_j * m_{i,j} * r_{i,j}$$

où w_j est le poids attribué à chaque classifieur e_j , $m_{i,j}$ est la mesure attribuée par chaque classifieur e_j à chaque mot ω_i de la liste. $r_{i,j}$ est le rang attribué à chaque classe ω_i par le

classifieur e_j . Il est égal à 0 si la classe ω_i ne figure pas dans la liste du classifieur e_j . $m_{i,j}$ est calculé comme suit:

$$r_{i,j} = 1 - \frac{pos_{ij}}{K}$$

Où pos_{ij} est la position du mot i dans la liste de solutions du classifieur e_j .

3.2.4. Intersection

Dans la méthode d'intersection, chaque classifieur propose une liste de classes ordonnées selon sa préférence. L'intersection de ces listes forme le résultat final. Ceci veut dire qu'une classe n'appartient à la liste finale que si elle existe dans toutes les listes des classifieurs. Les listes à combiner peuvent être de taille différente. Le seuil qui détermine la taille de chaque liste est calculé en apprentissage.

Ho [HOT92] propose une méthode qui délimite la taille des listes avant de les combiner. Pour cela, une base d'apprentissage B^a est utilisée. Chaque classifieur propose des rangs aux différentes formes de cette base. Le rang le plus élevé est considéré comme le seuil à utiliser dans la phase de test. Pour chaque forme de test, les classifieurs proposent des listes de classes, celles placées avant le seuil sont prises en compte dans l'intersection. Cette procédure est illustrée dans le Tableau IV.1 présentant les rangs attribués par 4 classifieurs à 5 données d'apprentissage. Selon ce tableau, 34 solutions doivent être prises du classifieur 1, 36 du classifieur 2, 29 du classifieur 3 et 24 du classifieur 4.

$B^a / e(j)$	$a(1)$	$a(2)$	$a(3)$	$a(4)$
a_1	3	12	1	24
a_2	1	5	29	12
a_3	34	3	4	6
a_4	4	36	5	5
a_5	16	7	3	4
seuil	34	36	29	24

Tableau IV.1 : Exemple de recherche des tailles des listes des classifieurs dans l'intersection [HOT92].

3.2.5. Union

Dans la méthode d'union, chaque classifieur propose aussi une liste des rangs reflétant sa préférence pour les classes. L'union de ces listes forme le résultat final. Une classe qui existe au moins une fois dans l'une des listes des classifieurs, appartient aussi à la liste finale. Contrairement à l'intersection, ici le nombre de classes de chaque classifieur doit être suffisamment petit pour ne pas avoir un ensemble trop important dans la liste finale. Pour sélectionner les seuils des listes des classifieurs, une procédure min-max peut être utilisée [HOT92]. Cette procédure se base sur les performances des classifieurs obtenues sur la base d'apprentissage B_a comme précédemment. Elle est illustrée dans le Tableau IV.2. La moitié gauche du tableau présente les réponses des classifieurs pour les formes d'apprentissage. Pour chacune des lignes, le meilleur (minimum) rang est déterminé (la moitié droite du tableau). Le maximum de tous ces minima est calculé pour chaque classifieur (colonne).

$B^a / e(j)$	$a(1)$	$a(2)$	$a(3)$	$a(4)$	$a(1)$	$a(2)$	$a(3)$	$a(4)$
a_1	3	12	1	24	0	0	1	0
a_2	1	5	29	12	1	0	0	0
a_3	34	3	4	6	0	3	0	0
a_4	4	36	5	5	4	0	0	0
a_5	16	2	3	4	0	2	0	0
Colmax					4	3	1	0

Tableau IV.2 : Procédure min-max pour déterminer les seuils dans l'union [HOT92]

3.3. COMBINAISON DE CLASSIFIEURS DE TYPE MESURE

3.3.1. Méthodes fixes

Le principe de base derrière les méthodes fixes est le suivant: les classifieurs sont indépendants et estiment des probabilités a posteriori des classes. Ainsi, pour reconnaître une forme x , on utilise une règle de décision $E(x)$ qui revient à choisir la classe ω_i pour laquelle la probabilité a posteriori P_i est la plus élevée [ZOU04]:

$$E(x) = \begin{cases} \omega_i & \text{si } \max_{i=1}^N P_i = \max_{m=1}^N P_m \\ \text{rejet} & \text{sinon} \end{cases}$$

La probabilité a posteriori P_m peut être calculée par l'une des règles suivantes:

3.3.1.1. La règle maximum

$$P_m = \max_{j=1}^L m_{i,j}$$

3.3.1.2. La règle minimum

$$P_m = \min_{j=1}^L m_{i,j}$$

3.3.1.3. La règle médiane

$$P_m = \begin{cases} \frac{m_{i,\frac{L}{2}} + m_{i,\frac{L}{2}+1}}{2} & \text{si } L \text{ est paire} \\ m_{i,\frac{L+1}{2}} & \text{si } L \text{ est impaire} \end{cases}$$

3.3.1.4. La règle produit

$$P_m = \prod_{j=1}^L m_{i,j}$$

3.3.1.5. La règle linéaire

$$P_m = \lambda \sum_{j=1}^L m_{i,j}$$

λ est une constante. Ainsi, pour $\lambda = 1$, on obtient la règle de somme. Si $\lambda = \frac{1}{L}$ alors la règle est la moyenne simple.

L'exemple suivant avec $c = 3$ et $L = 5$, permet d'éclaircir le fonctionnement de ces règles de combinaison.

	e_1	e_2	e_3	e_4	e_5	<i>Max</i>	<i>Min</i>	<i>Méd</i>	<i>Prod</i>	<i>Som</i>	<i>Moy</i>
ω_1	0.1	0.0	0.4	0.2	0.1	0.4	0.0	0.1	0.0	0.8	0.16
ω_2	0.5	0.0	0.3	0.7	0.8	0.8	0.0	0.5	0.0	2.3	0.46
ω_3	0.4	1.0	0.4	0.1	0.1	1.0	0.1	0.1	0.016	2.0	0.4

Tableau IV.3 : Exemple de la décision avec les méthodes fixes (Maximum, Minimum, Médian, Produit, Somme, Moyenne)

La décision finale obtenue par les règles minimum, maximum et produit est la classe 3 alors que celle obtenue par la règle de somme et moyenne est la classe 2.

3.3.2. Les règles pondérées

Ces méthodes consistent tout simplement à appliquer des pondérations aux sorties des classifieurs. Chaque probabilité a posteriori P_i d'une classe ω_i peut être obtenue par l'une des règles suivantes:

$$P_i = \lambda \sum_{j=1}^L w_j m_{i,j}$$

$$P_i = \prod_{j=1}^L m_{i,j}$$

W_j est le coefficient qui détermine l'importance attribuée au j^{ieme} classifieur dans la combinaison ($j = 1, \dots, L$). Avec $\lambda = 1$, la première règle est nommée la somme pondérée. Pour $\lambda = i$, il s'agit de la moyenne pondérée.

La deuxième règle est le produit pondéré. D'autres règles qui se basent sur le même principe que cette méthode existent.

3.3.3. Les réseaux de neurones

La combinaison de classifieurs au niveau mesure peut aussi être réalisée en utilisant un réseau de neurones en tant que fonction de décision [WOL92][HUA95]. Les classifieurs dans l'ensemble sont au début appris sur la base entière ou sur des parties de cette base. Ensuite, une base de validation est classée en utilisant cet ensemble de classifieurs. La réponse à chaque forme est considérée comme une nouvelle donnée à apprendre par le réseau de neurones ce qui lui permet d'apprendre le comportement des classifieurs. Chaque forme de la base de test est alors classée au début par l'ensemble de classifieurs dont les sorties sont ensuite combinées par le réseau de neurones pour fournir la décision finale. Les sorties des classifieurs sont donc considérées pour le classifieur de fusion comme un nouvel ensemble de caractéristiques de chaque forme de test.

4. CONCLUSION

Dans ce chapitre nous avons obtenu une vue globale sur les systèmes multiclassifieurs, nous pouvons maintenant passer à le chapitre suivant et voir le travail proposé dans le domaine de la combinaison des classifieurs appliqué sur les mots arabes manuscrits.

CHAPITRE V

TRAVAIL PROPOSÉ ET CONCLUSION

Dans ce chapitre:

- ✓ Base de mots et outils de développement.
- ✓ Les phases du système proposé.

1. INTRODUCTION

Dans ce chapitre nous allons présenter notre travail, commençant par une présentation de la base de mots et les différents outils de développement utilisés, puis les caractéristiques choisies, ainsi que l'apprentissage et le test des classifieurs individuels, enfin la combinaison des trois classifieurs et voir si la combinaison des trois va augmenter le taux de reconnaissance moyen.

2. BASE DE MOTS ET OUTILS DE DEVELOPPEMENT

2.1. BASE DE MOTS

Les images des mots arabes utilisées dans ce projet sont celles qui sont librement disponibles (pour la recherche) dans la base de mots manuscrits des villes tunisiennes IFN/ENIT [PEC02]. Cette base de mots est construite par Pechwitz, et al, et avant cela il n'y a pas d'une base standard de mots manuscrits pour ce domaine.

Le Tableau V.1 illustre quelques échantillons des mots :

Mot arabe manuscrit	Mot arabe
	الشرايع
	الفايض
	المكالم
	برج الطويل
	الوديان
	القلعة الكبرى

Tableau V.1 : Des mots extrait de la base INF/ENIT.

2.2. OUTILS

2.2.1. Turbo Delphi 2006

Turbo Delphi est un environnement de développement intégré (IDE), créé par CodeGear, qui est destiné aux étudiants, professionnels et programmeurs amateurs.

Il existe deux versions de Turbo Delphi, une qui génère des applications Win32 (Turbo Delphi pour Windows), et une autre qui génère du « bytecode » pour Microsoft .NET CLR. Chaque version est disponible en deux éditions, une édition gratuite « Explorer », et une autre payante « Professionnel ».

Dans ce travail nous avons utilisé la version gratuite Turbo Delphi pour Windows édition Explorer.

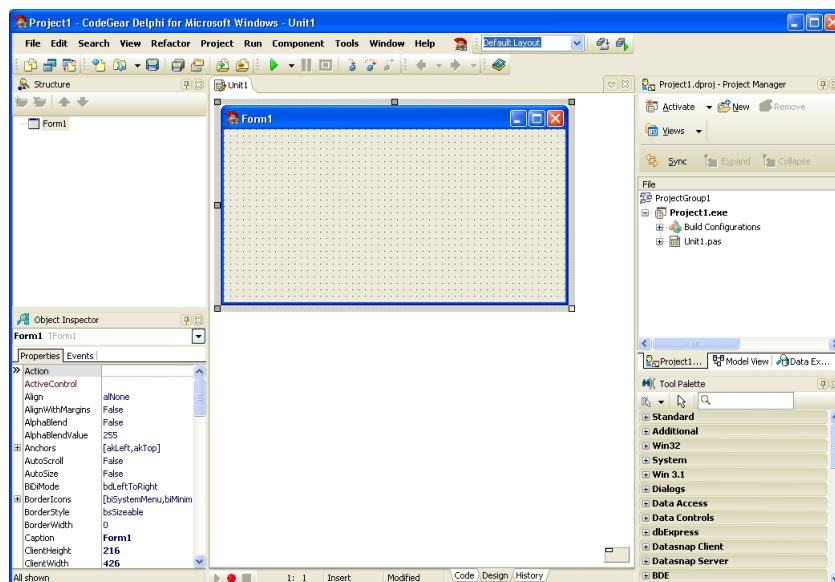


Figure V.1 : L'interface principale de l'IDE Delphi.

2.2.2. Mathworks MATLAB

MATLAB est à la fois un langage de programmation et un environnement de développement, il est développé et commercialisé par la société américaine The MathWorks. MATLAB est utilisé dans les domaines de l'éducation, de la recherche et de l'industrie pour le calcul numérique mais aussi dans les phases de développement de projets.

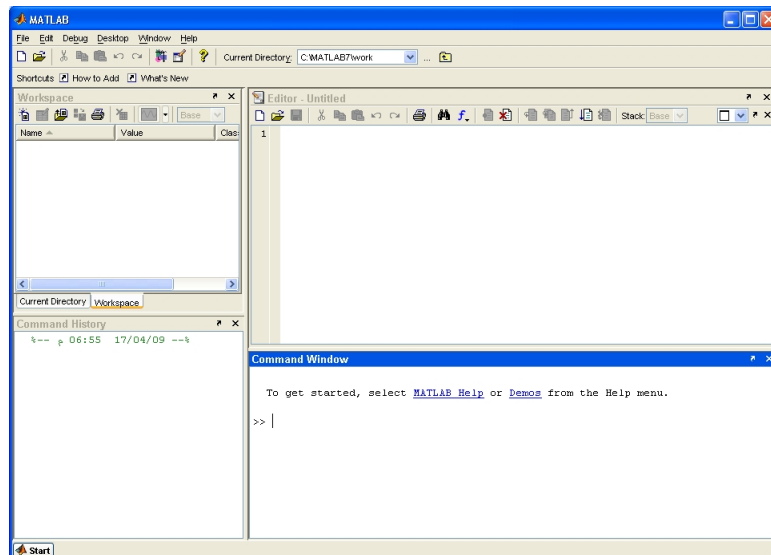


Figure V.2 : L'interface principale du MATLAB.

3. LES PHASES DU SYSTEME PROPOSE

3.1. ACQUISITION D'IMAGE

Les images sont acquises par Pechwitz, et al, [PEC02] d'une façon hors ligne.

3.2. PRETRAITEMENT

3.2.1. Seuillage

Le seuillage est déjà fait par Pechwitz, et al, [PEC02]. Alors cette étape va être ignorée. Et voici des échantillons de la base IFN/ENIT qui montrent le seuillage:

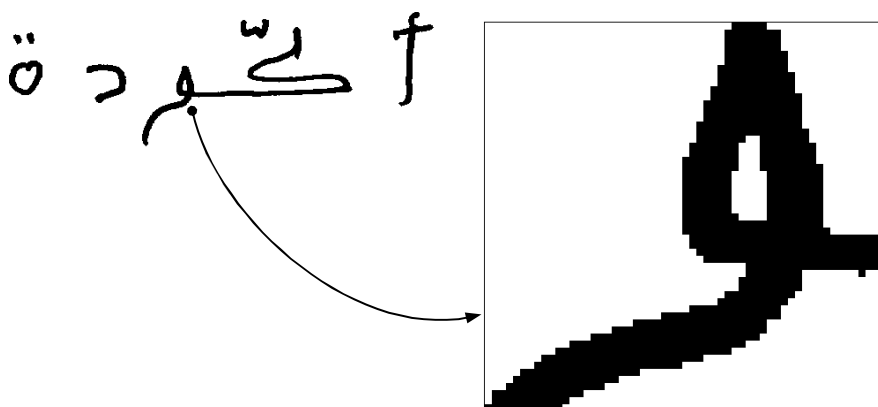


Figure V.3 : Le seuillage du mot «أكودة» fait par [PEC02].

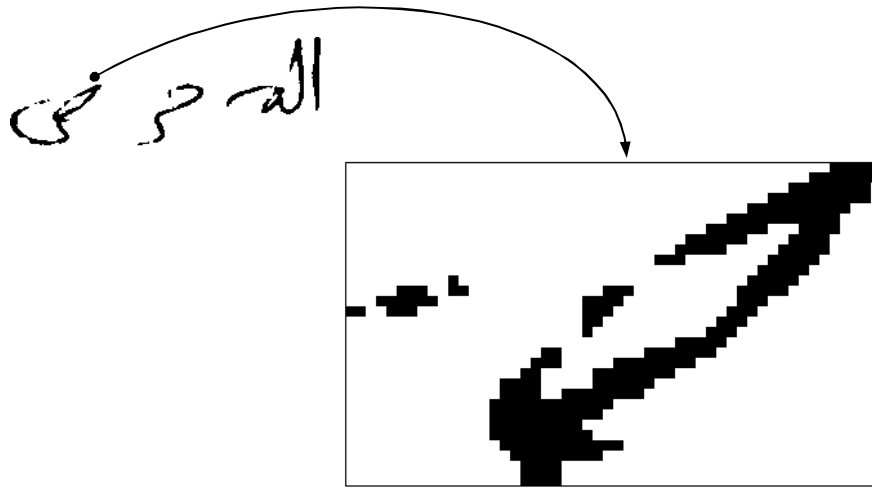


Figure V.4 : Le seuillage du mot « المحرص » fait par [PEC02].

On peut voir que le seuillage du deuxième mot est mal appliqué.

3.2.2. Squelettisation

Pour la squelettisation nous avons appliqué l'algorithme de Hilditch [HIL69], un exemple est donné dans la figure suivante.

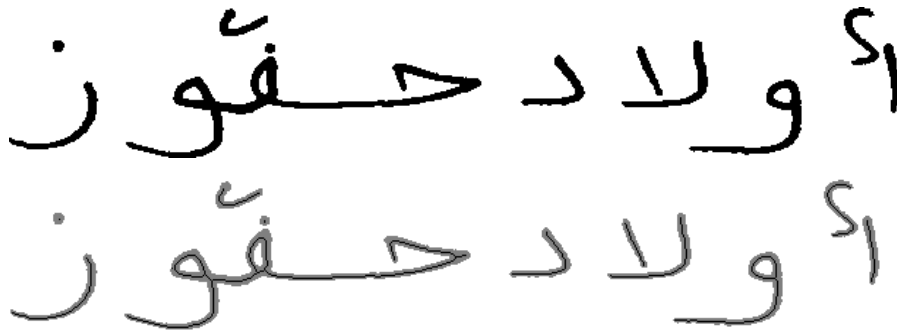


Figure V.5 : La squelettisation du mot « أولاد حفوز » par notre système.

3.2.3. Détection de la ligne de base

Ici nous avons utilisé deux méthodes de détection de la ligne de base, une détection basée sur l'image originale (en haut) et une détection basée sur le squelette (en bas), voir Figure V.6, Figure V.7.

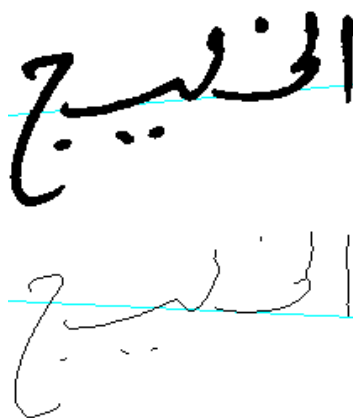


Figure V.6 : Détection de la ligne de base du mot « الخليج » par notre système.

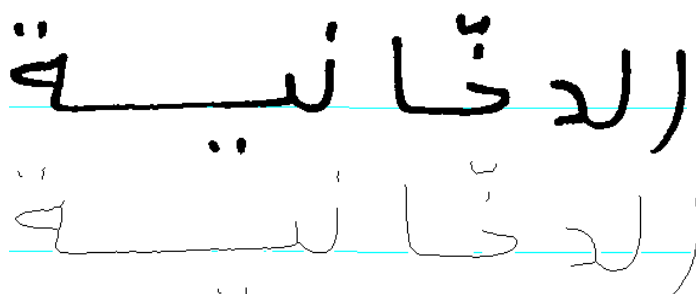


Figure V.7 : Détection de la ligne de base du mot « الدخانية » par notre système.

3.3. SEGMENTATION

3.3.1. Segmentation en blocs, lignes, mots et lettres

Comme les images contiennent un nom de ville chacune, la segmentation en blocs, en ligne, et en mots va être ignorée, ainsi que la segmentation en lettres n'a pas été appliquée.

3.3.2. Segmentation en graphèmes

Nous avons utilisé une segmentation en graphèmes pour l'extraction des caractéristiques globales, un graphème est considéré comme une ligne, quelque soit fermée ou non, qui n'a aucune attachement au milieu (Figure V.8).

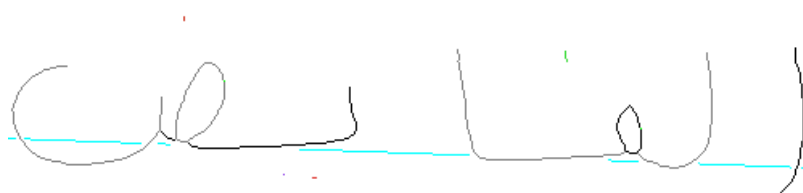


Figure V.8 : La segmentation en graphèmes de droite à gauche par notre système.

3.4. EXTRACTION DE CARACTERISTIQUES

Touts les caractéristiques suivantes sont basées sur la détection de la ligne de base et sur les graphèmes.

3.4.1. Nombres des ascendants et des descendants

Les ascendants sont tous graphèmes qui ont une hauteur en dessus de la ligne de base plus que 25% de l'hauteur totale du mot, ainsi que les descendants sont tous graphèmes qui ont une hauteur en dessous de la ligne de base plus que 25% de l'hauteur totale du mot.



Figure V.9 : L'extraction des ascendants et des descendants du mot « رأس الزراع » par notre système donne 4 ascendants et 4 descendants.

3.4.2. Nombres des points hauts et des points bas

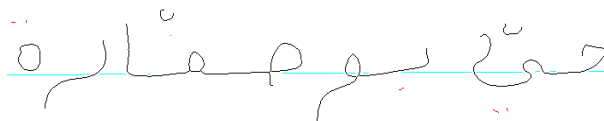


Figure V.10 : L'extraction des points hauts et bas du mot « حي بوصفارة » par notre système donne 1 point haut et 1 point bas.

3.4.3. Nombres des points doubles hauts et des points doubles bas

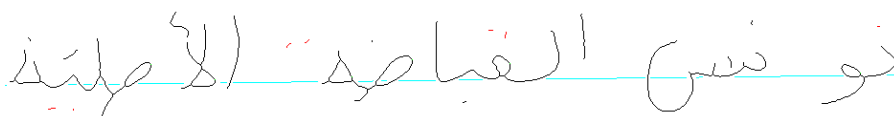


Figure V.11 : L'extraction des points doubles hauts et bas du mot « تونس القباضة الأصلية » par notre système donne 3 points doubles haut et 1 point double bas.

3.4.4. Nombres des points triples hauts



Figure V.12 : L'extraction des points triples hauts du mot « شعال » par notre système donne 1 point triple haut.

3.4.5. Nombres des boucles

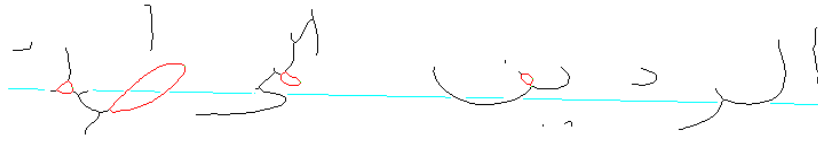


Figure V.13 : L'extraction des boucles du mot « الرديف المحطة » par notre système donne 4 boucles.

3.4.6. Nombres des composants connexes

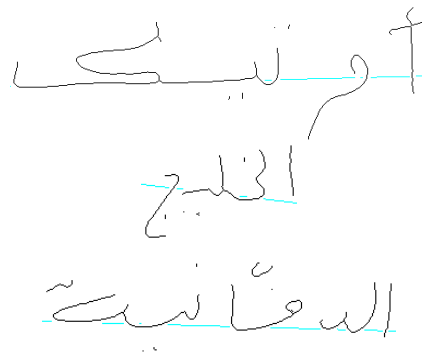


Figure V.14 : L'extraction des composants connexes du mot « أوتيك، الخليج، الدخانية » par notre système donne 3,3,5 respectivement.

3.4.7. Nombres des graphèmes

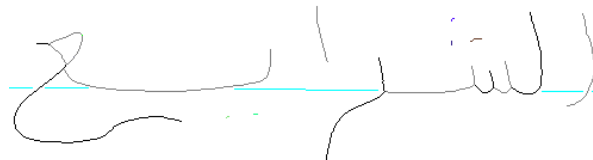


Figure V.15 : L'extraction des graphèmes du mot « حي بوصفارة » par notre système donne 10 graphèmes.

Maintenant, après l'extraction des dix caractéristiques, nous pouvons passer à l'apprentissage puis le test.

3.5. APPRENTISSAGE ET TEST

Cette étape d'apprentissage et de test n'existe qu'à la première fois. Après, elle va être remplacée par la phase de classification.

Nous avons extrait à partir de la base de mots IFN/ENIT un ensemble qui contient un vocabulaire de 38 mots, il contient en global 238 mots à partir des différents scripteurs, cet

ensemble a été divisé en deux sous ensembles, une pour l'apprentissage (75%), et une autre pour le test (25%).

3.5.1. Le premier classifieur

Le premier classifieur est un réseau de neurones avec une couche cachée créée sous Matlab avec la fonction **newff**, le réseau est détaillé comme suite :

- Une couche d'entrée avec 10 entrées correspondantes aux caractéristiques extraites.
- Une couche cachée avec 45 neurones, ce nombre de neurones a été trouvé expérimentalement.
- Une couche de sortie avec 38 neurones correspondants à la taille du vocabulaire.

L'apprentissage du réseau a donné un taux de reconnaissance de 46.77% de l'ensemble du test, la courbe de performance du réseau durant l'apprentissage est montrée par la figure suivante.

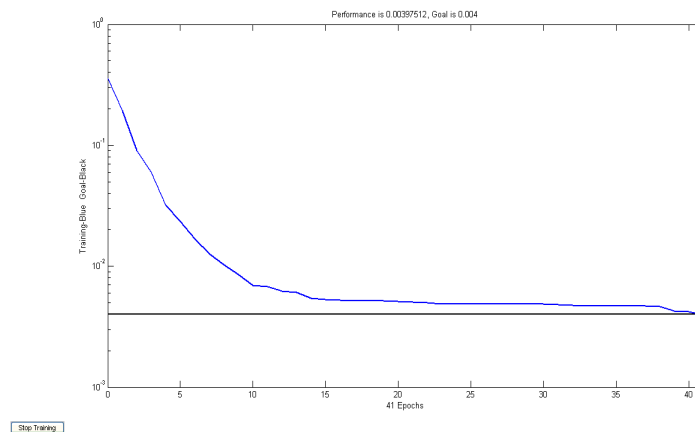


Figure V.16 : L'apprentissage du premier PMC.

L'erreur moyenne quadratique minimale obtenue par l'apprentissage du premier réseau de neurones est 0.00397512 après 20 époques (Figure V.16).

3.5.2. Le deuxième Classifieur

Le deuxième classifieur est aussi un réseau de neurones avec une couche cachée, le réseau est détaillé comme suite :

- Une couche d'entrée avec 10 entrées correspondantes aux caractéristiques extraites.

- Une couche cachée avec 39 neurones, ce nombre de neurones a été aussi trouvé expérimentalement.
- Une couche de sortie avec 38 neurones correspondants à la taille du vocabulaire.

L'apprentissage du réseau a donné un taux de reconnaissance de 59.68% de l'ensemble du test, la courbe de performance du réseau durant l'apprentissage est montrée par la figure suivante.

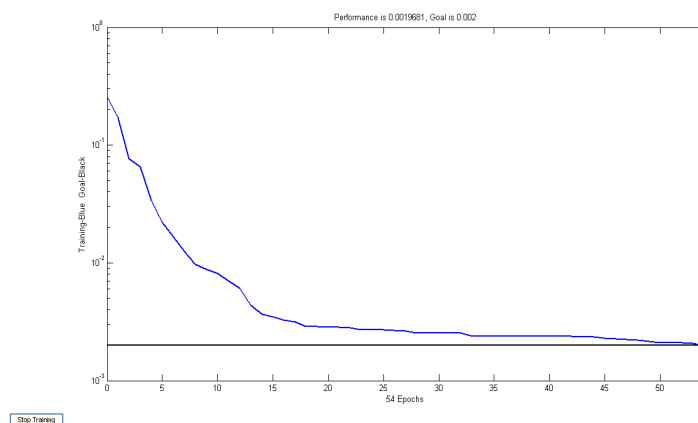


Figure V.17 : L'apprentissage du deuxième PMC.

L'erreur moyenne quadratique minimale obtenue par l'apprentissage du premier réseau de neurones est meilleure que celle obtenue avec le premier réseau, l'erreur est 0.0019681 après 54 époques (voir Figure V.17).

3.5.3. Le troisième classifieur

Le troisième classifieur utilise l'algorithme de K Plus Proche Voisins (KNN) pour la classification, alors aucun apprentissage n'est nécessaire.

La classification des images de l'ensemble de test donne un taux de reconnaissance 67.74% avec $k=1$, k a été trouvé expérimentalement par l'incrément de 1 à partir de 1 jusqu'à 10.

3.6. COMBINAISON

Après avoir vu les résultats des trois classifieurs individuels, on a combiné les trois parallèlement en utilisant une combinaison par somme [ZOU04], nous avons obtenu un taux de reconnaissance égale à 72.58%, la structure du système est montrée par la Figure V.18.

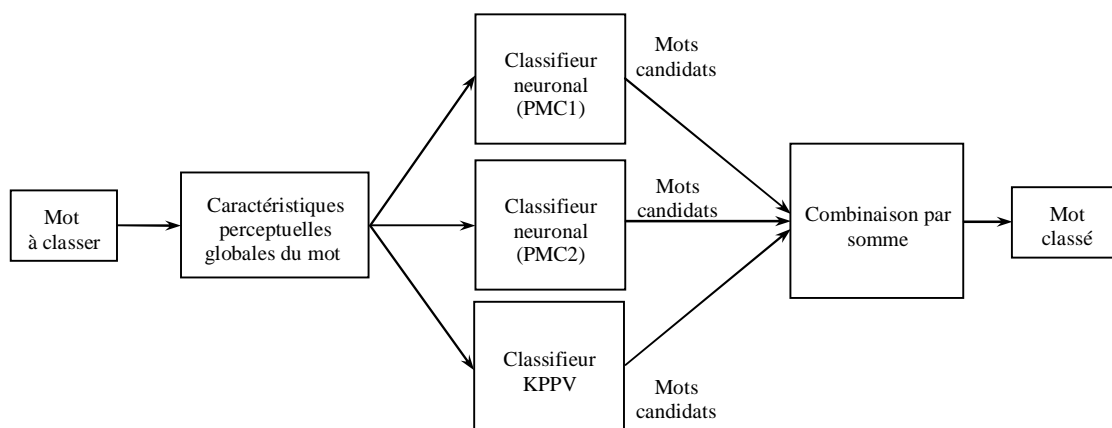


Figure V.18 : L'architecture du système multiclassifieur proposé.

Les résultats de performances des classificateurs individuels et du multiclassifieur sont résumés dans le Tableau V.2.

Classifieur	Taux de reconnaissance
PMC1	46.77%
PMC2	59.68%
KPPV	67.74%
PMC1 et PMC2 combinés	62.90%
PMC1 et KPPV combinés	67.74%
PMC2 et KPPV combinés	67.74%
Les trois combinés	72.58%

Tableau V.2 : Les résultats de performance avant et après la combinaison.

Voici quelques exemples de classification par les classifieurs individuels ainsi que par le multiclassifieur :

Code du mot	Libellé	Image réel	PMC1	PMC2	KPPV	PMC1+ PMC2	PMC1+ KPPV	PMC2+ KPPV	Les trois combinés	Observation
8	الفايض		8	8	8	8	8	8	8	Bien reconnue par les trois
5	الدخانية		5	5	5	5	5	5	5	Bien reconnue par les trois
12	المنزه 6.		12	12	12	12	12	12	12	Bien reconnue par les trois
35	مارث		7	35	35	35	35	35	35	Mal reconnue par le 1 ^{er}
20	حي.بو.صقارة		8	20	20	20	20	20	20	Mal reconnue par le 1 ^{er}
21	دوّار اللواته		10	21	10	21	10	10	10	Mal reconnue par le 2 ^{eme}
8	الفايض		8	12	8	8	8	8	8	Mal reconnue par le 2 ^{eme}
22	رأس الذراع		22	22	21	22	21	21	22	Mal reconnue par le 3 ^{eme}
5	الدخانية		5	5	8	5	8	8	5	Mal reconnue par le 3 ^{eme}
2	أوتيك		2	6	29	2	29	29	29	Bien reconnue par le 1 ^{er}
32	قرعة.التاظور		12	11	7	12	7	7	7	Mal reconnue par les trois

Tableau V.3 : Exemples de reconnaissance des mots arabes par le système proposé

4. CONCLUSION

Le taux de reconnaissance a été augmenté avec la combinaison par 4.84% par rapport au meilleur classifieur et par 14.52% par rapport à la moyenne des trois classifieurs, on déduit qu'avec l'utilisation de la combinaison des classifieurs on bénéficie des puissances individuelles des classifieurs, chacun dans son espace de spécialité pour un taux de reconnaissance plus élevé.

CONCLUSION GÉNÉRALE

CONCLUSION GÉNÉRALE

Ce mémoire porte sur un système multiclassifieurs de Réseaux de Neurones et K-Plus Proches Voisins pour la reconnaissance de l'écriture arabe manuscrite avec des scripteurs multiples et un lexique défini.

Un système hybride d'un KPPV et deux Perceptron Multi Couche a été réalisé pour un vocabulaire réduit et fixé à partir de la base de mots arabes INF/ENIT. Ce système obtient des performances meilleures qu'un système neuronal ou un système KPPV seules comme montré dans le dernier chapitre.

Par ce travail nous espérons que nous avons couvert une partie concernant le domaine de recherche de la combinaison des classifieurs appliquée sur l'écriture arabe manuscrite.

Dans l'avenir il est possible d'intégrer un classifieur basé sur les Modèles de Markov Cachés dans le multiclassifieur et essayer de bénéficier les puissances des Modèles de Markov Cachés montrées dans les travaux d'aujourd'hui [MEN08,HAL04,DUP04,AUG01,BEN99a].

Nous espérons dans l'avenir pouvoir intégrer un groupe de recherche dans ce domaine pour pouvoir contribuer dans la recherche ainsi que développer notre algorithme pour couvrir plus de mots arabes et plus de scripteurs et contribuer dans l'avancement des recherches dans le domaine de reconnaissance de l'écriture arabe.

BIBLIOGRAPHIE

- [ACH96] B. Achermann and H. Bunke, "Combination of classifiers on the decision level for face recognition," *Technical report*, 1996.
- [ALO02] Y. Al-Ohali, "Handwritten Word Recognition – Application to Arabic Cheque Processing," Montreal, Quebec, Canada, PhD Thesis 2002.
- [ALO03] Y. Al-Ohali, Mohamed Cheriet, and Ching Y. Suen, "Databases for recognition of handwritten arabic cheques," *Pattern Recognition*, vol. 36, no. 1, pp. 111-121, 2003.
- [AMI00] A. Amin, "Recognition of printed Arabic text based on global features and decision tree learning techniques," *Pattern Recognition*, vol. 33, pp. 1309-1323, 2000.
- [AUG01] Emmanuel Augustin, "Reconnaissance de mots manuscrits par systèmes hybrides Réseaux de Neurones et Modèles de Markov Cachés," PARIS V, Thèse de doctorat 2001.
- [BAD95] B. Al-Badr and S.A. Mahmoud, "Survey and bibliography of Arabic optical text recognition," *Signal processing*, vol. 41, pp. 49-77, 1995.
- [BEL02] A. Bellili, M. Gilloux, and P. Gallinari, "reconnaissance de chiffres manuscrits par un système hybride MLP-SVM," *13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et d'Intelligence Artificielle*, vol. 3, pp. 761-769, 2002.
- [BEL97] A. Belaid. (1997) *Analyse de documents: de l'image à la représentation par les normes de codage*.
- [BEN99a] N. Benamara, "Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée," Tunis II, Thèse de doctorat 1999.
- [BEN99b] A. Bennis, A. Zahour, and B. Taconet, "Extraction des lignes d'un texte manuscrit arabe," *Vision interface*, 1999.
- [BOR81] J.C. Borda, *Mémoire sur les élections au scrutin*. Paris: Histoire de l'Académie Royale des Sciences, 1781.
- [CAR03] F. Carmagnac, "Classification supervisée et semi-supervisée : contributions à la classification d'images de documents," PhD thesis 2003.
- [CAS96] R.G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 7, July 1996.

- [COV91] T.M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transaction on Information Theory*, vol. 13, no. 1, pp. 123-140, 1967.
- [CRA96] L. Cranor, "Declared-strategy voting: an instrument for group decision-making," Washington, PhD thesis 1996.
- [DAV93] E. Davalo and P. Naim, "Des réseau de neurones," Eyrolles, 1993.
- [DUD01] R. O. Duda, P. E. Hart, and and D. G. Stork, *Pattern Classification*, 2nd ed.: John Wiley & Sons, 2001.
- [DUP04] Xavier Dupré, "Contributions à la reconnaissance de l'écriture cursive à l'aide de modèles de Markov cachés," Paris V, 2004.
- [FAR05] N. Farah, L. Souici, and M. Sellami, "Arabic Word Recognition by Classifiers and Context," vol. 20, no. 3, pp. 402-410, 2005.
- [GAD91] L. Gader et al., "Recognition of handwritten digits using template and model matching," *Pattern Recognition*, vol. 24, no. 5, pp. 421-431, 1991.
- [GLA57] H. T. Glantz, "On the recognition of information with a digital computer," *J. ACM*, vol. 4, no. 2, pp. 178-188, 1957.
- [GOS97] B. Gosselin, "Cooperation of multilayer perceptron classifiers," *8th Workshop on Circuits, Systems and Signal Processing*, pp. 187-190, 1997.
- [GRE06] R. Gregory, Sargur N. Ball, Srihari, and Harish Srinivasan, "Versatile search of scanned arabic handwriting," *Summit on Arabic and Chinese Handwriting Recognition*, 2006.
- [HAL04] Khalid Hallouli, "Reconnaissance de caractères par méthodes markoviennes et réseaux baésiens," Paris, Thèse de doctorat 2004.
- [HAT90] J.P. Hatton, "Intelligence Artificielle et Reconnaissance des Formes," *Techniques de l'ingenieur*, vol. H2, 1990.
- [HAY94] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.
- [HEB49] D. Hebb, *The Organization of Behavior*. New York: Wiley, 1949.

- [HIL69] J. Hilditch, "Linear skeletons from square cupboards," *Machine Intelligence 4*, p. 404–420, 1969.
- [HOT92] T. Ho, "A theory of multiple classification to visual word recognition," Suny at Buffalo, New York, PhD thesis 1992.
- [HUA95] Y. Huang, K. Liu, and C. Suen, "The combination of multiple classifiers by neural network approach," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 9, no. 3, pp. 579-597, 1995.
- [JAC85] P. C. Jackson, *Introduction to Artificial Intelligence.*: Dover Publications Incorporated, 1985.
- [KHA99] N. Kharma, M. Ahmed, and M. Ward, "A new comprehensive database of handwritten arabic words, numbers, and signatures used for ocr testing," *Electrical and Computer Engineering*, vol. 2, pp. 766-768, 1999.
- [KHO03] M. S. Khorsheed, "Recognising handwritten Arabic manuscripts using a single hidden Markov model," *Pattern Recognition Letters*, vol. 24, pp. 2235-2242, 2003.
- [KIM00] J. Kim, K. Kim, C. Nadal, and C. Suen, "A methodology of combining HMM and MLP classifiers for cursive word recognition," *International Conference Document Analysis and Recognition*, vol. 2, pp. 319-322, 2000.
- [LAM94] L. Lam and C. Suen, "A theoretical analysis of the application of majority voting to pattern recognition," *International Conference on Pattern Recognition*, pp. 418-420, 1994.
- [LAR07] Olivier Delalleau and Hugo Larochelle, *Algorithme des k plus proches voisins*, Janvier 19, 2007.
- [MAD02] Somaya Al-Maadeed, Dave Elliman, and Colin A. Higgins, "A data base for arabic handwritten text recognition research," *iwfhr*, p. 485, 2002.
- [MAR01] J. P. Marques de Sa, *Pattern Recognition: Concepts, Methods and Applications.*: Springer, 2001.
- [MCC43] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.
- [MEI00] Michel Weinfeld, "Réseaux de neurones," 2000.

- [MEN08] Farès Menasri, "Contributions à la reconnaissance de l'écriture arabe manuscrite," UNIVERSITE PARIS DESCARTES, Thèse de doctorat 2008.
- [NIG93] A. Nigrin, *Networks for Pattern Recognition*. Cambridge: MA: The MIT Press, 1993.
- [OLI02] J. J. Oliveira, J. de Carvalho, C. Freitas, and R. Sabourin, *Feature sets evaluation for handwritten word recognition.*, 2002.
- [PEC02] M. Pechwitz, S. Snoussi Maddouri, V. Maergner, N. Ellouze, and H. Amiri, "IFN/ENIT database of handwritten arabic words," *CIFED*, 2002.
- [PEC03] M. Pechwitz and V. Märgner, "HMM based approach for handwritten Arabic word recognition using the IFN/ENIT – database," *International Conference on Document Analysis and Recognition*, pp. 890-894, 2003.
- [PET66] L.E. Petrie and T. Baum, *Annals Math. Statist.*, no. 37, pp. 1554-1563, 1966.
- [PRE03] L. Prevost, C. Michel-Sendis, A. Moises, L. Oudot, and M. Milgram, "Combining model-based and discriminative classifiers: application to hadwritten character recognition," *7th Internatinal Conference on Dcument Analysis and Recognition*, vol. 1, pp. 31-35, 2003.
- [ROS58] F. Rosenblatt, "The Perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386-408, 1958.
- [RUM86] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, vol. 1, pp. 318-362, 1986.
- [TAR] Khadir Mohamed Tarek, "Principe de base des réseaux de neurones artificiels et apprentissage," ANNABA, 0000.
- [THI94] E. Thiel, "Les distances de chanfrein en analyse d'images : fondements et applications," Grenoble 1, Thèse de doctorat 1994.
- [TRI95] Oivind Due Trier and Torfinn Taxt, "valuation of binarization methods for docu-," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3), p. 312–315, 1995.
- [VAN00] M. Van-Erp and L. Schomaker, "Variants of the borda count method for combining ranked classifier hypotheses," *7 th International Workshop on Frontiers in Handwriting Recognition*, pp. 443-452, 2000.

- [VER01] B. Verma, P. Gader, and W. Chen, "Fusion of multiple handwritten word recognition techniques," *Pattern Recognition Letters*, vol. 22, no. 9, pp. 991-998, 2001.
- [WOL92] D. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, 1992.
- [XUL92] L. Xu, A. Krzyzak, and C. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418-435, 1992.
- [ZOU04] H la Khoufi Zouari, "Contribution   l' valuation des m thodes de combinaison parall le de classifieurs par simulation," 2004.

LISTE DES TABLEAUX

Tableau I.1 : Les primitifs des tracés de fréquence cardiaque.....	14
Tableau I.2 : Types communs de caractéristiques de signal et d'image.....	18
Tableau II.1 : L'alphabet arabe.....	41
Tableau III.1 : Fonctions d'activation usuelles.....	56
Tableau IV.1 : Exemple de recherche des tailles des listes des classifieurs dans l'intersection [HOT92].....	78
Tableau IV.2 : Procédure min-max pour déterminer les seuils dans l'union [HOT92].....	79
Tableau IV.3 : Exemple de la décision avec les méthodes fixes (Maximum, Minimum, Médian, Produit, Somme, Moyenne).....	80
Tableau V.1 : Des mots extrait de la base INF/ENIT.	84
Tableau V.2 : Les résultats de performance avant et après la combinaison.	93
Tableau V.3 : Exemples de reconnaissance des mots arabes par le système proposé.....	94

LISTE DES FIGURES

Figure I.1 : Représentations possibles des prototypes «pomme verte» et «orange».	8
Figure I.2 : Exemples de «pomme rouge» et «orange vert » à être caractérisés par les caractéristiques forme et couleur.	8
Figure I.3 : (a) Image grise de la composante verte de l'image pomme, (b) Histogramme de l'intensité de la lumière pour la région rectangulaire de l'intérêt montré à (a).	9
Figure I.4 : (a) des pommes vertes et des oranges dans l'espace des caractéristiques, (b) Une pomme rouge qui « ressemble » à une orange et un problème d'orange verdâtre.	10
Figure I.5 : Prévisions des actions boursières d'un jour à venir. r_A, r_B, r_C sont des actions des trois entreprises. Approximation fonctionnel (cercles noirs) de la valeur vraie de r (courbe), est indiqué pour le 16 juin selon les valeurs des actions, le changement Euro - Dollar, les revenus de l'entreprise A pendant 6 mois.	13
Figure I.6 : (a) Le traçage de la fréquence cardiaque avec la balance verticale par BPM. (b) un détail de la première onde de haut en bas est montrée avec ses primitifs.	14
Figure I.7 : primitif d'onde pour le signal de la fréquence cardiaque : (a) Transitions ; (b) Non-transitions.	15
Figure I.8 : ECG de 4 classes agnostiques	16
Figure I.9 : Les espaces de reconnaissance de formes pour la classification des formes (maladies de cœur dans cette situation) en utilisant caractéristiques d'ECG.	16
Figure I.10 : Les approches de RF - S : supervisé, U : non supervisé, SC : classification statistique, NN : réseaux neuronaux, DC : groupement de données, SM : classification structurelle, SA : analyse syntaxique, GI : inférence grammaticale [MAR01].	19
Figure I.11 : Un système de reconnaissance de formes avec ses unités. Quelques systèmes n'ont pas de l'unité de prétraitement et/ou de post-traitement.	24
Figure I.12 : Les phases d'un projet de reconnaissance de formes. Note : l'évaluation de caractéristique à deux phases distinctes.	25
Figure II.1 : (a) Ecriture hors ligne (ensemble de pixels), (b) Ecriture en ligne (ensemble de coordonnées).	30
Figure II.2 : Squelettisation du mot 'الخليج'.	32
Figure II.3 : Axe médian d'une forme continue (trait pointillé) dont seul le contour est représenté [THI94].	32
Figure II.4 : Correction de l'inclinaison des lignes à l'aide d'histogrammes de projection horizontale. Les pics les plus prononcés correspondent à la rotation correcte [CAR03].	34
Figure II.5 : Inclinaison de l'écriture, et histogrammes de projection verticale correspondants.	35
Figure II.6 : Masques qui permettent de lisser le contour des formes. Les masques complémentaires s'obtiennent par rotation [MEN08].	36
Figure II.7 : Histogramme de projection horizontale et bande de base.	36
Figure II.8 : Points en arabe : un, deux ou trois points.	41
Figure II.9 : Voyelles en arabe.	42

Figure II.10 : Les ascendants et descendants sont entourés.	43
Figure II.11 : Un mot peut être composé de plusieurs composantes connexes (pseudo-mots).....	43
Figure II.12 : ﻻ sous forme d'une ou de deux composantes connexes.....	44
Figure II.13 : Ligatures verticales et inversion de l'ordre du tracé.....	44
Figure II.14 : Extraction de caractéristique avec une fenêtre glissante de trois colonnes.	49
Figure II.15 : Mot originale squelette approximation linéaire.....	49
Figure II.16 : Combinaison de classifieurs dans [FAR05].....	50
Figure III.1 : Le Neurone de Mac Culloch-Pitts [MCC43].....	54
Figure III.2 : Le perceptron	55
Figure III.3 : Un perceptron à une couche cachée unique	57
Figure III.4 : Un perceptron à deux couches cachées.....	58
Figure III.5 : Modèle de Markov Caché.....	67
Figure III.6 : Modèle ergodique.....	68
Figure III.7 : modèles gauche-droite.....	68
Figure IV.1 : Combinaison séquentielle de classifieurs.....	72
Figure IV.2 : Combinaison parallèle de classifieurs.....	73
Figure IV.3 : Combinaison hybride de classifieurs	74
Figure V.1 : L'interface principale de l'IDE Delphi.....	85
Figure V.2 : L'interface principale du MATLAB.....	86
Figure V.3 : Le seuillage du mot «أكودة» fait par [PEC02].	86
Figure V.4 : Le seuillage du mot «المحرص» fait par [PEC02].....	87
Figure V.5 : La squelettisation du mot «أولاد حفور» par notre système.	87
Figure V.6 : Détection de la ligne de base du mot «الخليج» par notre système.	88
Figure V.7 : Détection de la ligne de base du mot «الدخانية» par notre système.....	88
Figure V.8 : La segmentation en graphèmes de droite à gauche par notre système.....	88
Figure V.9 : L'extraction des ascendants et des descendants du mot « رأس الزراع » par notre système donne 4 ascendants et 4 descendants.....	89
Figure V.10 : L'extraction des points hauts et bas du mot «حي بوصفارة» par notre système donne 1 point haut et 1 point bas.	89
Figure V.11 : L'extraction des points doubles hauts et bas du mot « تونس القباضة الأصلية » par notre système donne 3 points doubles haut et 1 point double bas.	89
Figure V.12 : L'extraction des points triples hauts du mot «شعال» par notre système donne 1 point triple haut..	89
Figure V.13 : L'extraction des boucles du mot «الرديف المحطة» par notre système donne 4 boucles.	90
Figure V.14 : L'extraction des composants connexes du mot « أوتيك،الخليج، الدخانية » par notre système donne 3,3,5 respectivement.....	90

Figure V.15 : L'extraction des graphèmes du mot « حي بوصفارة » par notre système donne 10 graphèmes.	90
Figure V.16 : L'apprentissage du premier PMC.....	91
Figure V.17 : L'apprentissage du deuxième PMC.....	92
Figure V.18 : L'architecture du système multiclassifieur proposé.....	93

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ