



ROYAUME DU MAROC
Université Sultan Moulay Slimane
Faculté des Sciences et Techniques
Département d'informatique
Béni Mellal



N° d'ordre : 59 / 2014

Centre d'Etudes Doctorales « Sciences et Techniques »
Formation Doctorale « Mathématiques et Physique Appliquées »

THÈSE

Présentée par

Badre - eddine EL KESSAB

Pour obtention du grade de

Docteur

Spécialité : Informatique

La reconnaissance automatique des caractères : Application au Tifinagh et aux chiffres Arabes

Soutenue publiquement le 18 octobre 2014 devant les membres du jury :

Pr. Ahmed ZEGHAL	Faculté des Sciences et Techniques, Béni Mellal	Président
Pr. Mohammed ABBAD	Faculté des Sciences, Rabat	Rapporteur
Pr. Brahim MINAOUI	Faculté des Sciences et Techniques, Béni Mellal	Rapporteur
Pr. Mohamed SABRI	Faculté des Sciences et Techniques, Béni Mellal	Rapporteur
Pr. Cherki DAOUI	Faculté des Sciences et Techniques, Béni Mellal	Directeur de thèse
Pr. Belaid BOUIKHALENE	Faculté Polydisciplinaire, Béni Mellal	Co-directeur de thèse
Pr. Mohamed FAKIR	Faculté des Sciences et Techniques, Béni Mellal	Co-directeur de thèse

A mes parents, ma famille et mes amis

REMERCIEMENTS

A quelques jours de déposer mon manuscrit, au moment où mes encadrants sont plongés dans leurs dernières relectures, je profite de quelques heures de répit pour me remémorer mes cinq années de thèse. En cinq ans, je suis passé de l'enthousiasme de commencer un nouveau projet, aux doutes et au découragement face aux nombreuses difficultés et à l'exigence de ce travail, à la satisfaction enfin de le voir progresser et évoluer à un jeune chercheur débutant beaucoup plus confiant en lui-même. Cette évolution est le fruit de nombreuses rencontres que je souhaite remercier ici pour toute l'aide, le soutien et les conseils qu'ils m'ont apportés pendant ces cinq années qui est pour moi une aventure incroyable.

Ce manuscrit conclut cinq ans de travail, je tiens en ces quelques lignes à exprimer ma reconnaissance envers tous ceux qui de près ou de loin y ont contribué.

Tout d'abord, Je souhaite remercier très chaleureusement mon directeur de thèse, **Mr. Cherki DAOUI**, pour son aide très précieuse, tant matérielle, qu'intellectuelle et morale. Son regard à la fois critique et bienveillant sur mes textes, son immense culture, les discussions que nous avons pu avoir lors de nos rencontres, ont nourri et approfondi ma réflexion, m'ont ouvert des pistes de recherche et avec ses encouragements, ont grandement contribué à la réalisation de ce travail.

Je remercie également mon co-encadrant de thèse **Mr. Belaid BOUIKHALENE** pour m'avoir accompagné pour mes premiers pas dans le monde de la recherche, dont la passion pour la recherche m'a entraîné dans cette aventure en réunissant les conditions de faisabilité d'une thèse de didactique à la Réunion. Son soutien sans faille et ses conseils m'ont permis de participer à des rencontres de didactique qui ont été sources d'enrichissement de ma réflexion.

J'adresse de chaleureux remerciements à mon co-encadrant de thèse, **Mr. Mohammed FAKIR**, pour son attention de tout instant sur mes travaux, pour ses conseils avisés et son écoute qui ont été prépondérants pour la bonne réussite de cette thèse. Son énergie et sa confiance ont été des éléments moteurs pour moi. J'ai pris un grand plaisir à travailler avec lui.

Je n'oublie pas d'exprimer ma reconnaissance à mes collègues au laboratoire Traitement d'Information et Aide à la Décision au sein de la Faculté des sciences et techniques de Béni Mellal. L'administration du CED, précédente et actuelle, de la commission recherche une écoute favorable à la compréhension de ce besoin de recherche didactique de ses formateurs. Je les remercie de m'avoir ainsi soutenu dans ce travail de longue haleine.

Pour avoir accepté de rapporter ce travail, j'assure ma reconnaissance à Mr. Mohammed ABBAD, Mr. Brahim MINAOUI et Mr. Mohamed SABRI ainsi plus particulièrement Mr. Ahmed ZEGHAL pour avoir assuré en plus le rôle de président de jury.

Je dédie ce mémoire à ma famille qui m'a accompagné tout au long de ces années de travail intense.

Merci enfin au lecteur qui par essence justifie la rédaction de ce document.

RÉSUMÉ

La reconnaissance automatique des caractères est un traitement informatique qui a pour but de traduire un texte écrit à un texte codé numériquement. Cela veut dire que le système est tout comme le cerveau humain, reconnaît des mots et des phrases existant dans un langage connu plutôt qu'une succession de caractères. Depuis la fin des années 1950, la reconnaissance de caractères est devenue un domaine actif de recherche pour la science informatique. Au début, on pensait qu'il s'agissait d'un problème facile, mais il apparut qu'il s'agissait d'un sujet beaucoup plus laborieux. Il faudra encore de nombreuses décennies aux ordinateurs, s'ils y parviennent un jour, pour lire tous les documents avec la même précision que les êtres humains. Les domaines d'application de la reconnaissance de caractères sont nombreux on peut citer à titre d'exemples la reconnaissance des documents administratifs, la reconnaissance des chèques bancaires, la reconnaissance des signatures, etc.

Ce travail de thèse propose deux catégories de systèmes de reconnaissance de caractères, la première catégorie s'intéresse à la reconnaissance des caractères Tifinaghs imprimés, manuscrits et cursifs. Tandis que la deuxième est consacrée à la reconnaissance des chiffres Arabes extraits de la base de données standard MNIST.

Avant de citer les différentes méthodologies envisagées pour atteindre notre objectif, donnons d'abord juste un aperçu concernant les trois phases de construction d'un système de reconnaissance de formes :

- ◆ Le prétraitement qui sert à améliorer la qualité des images à reconnaître afin de pouvoir les rendre trop lisibles et manipulables par un certain ordinateur, cette amélioration a pour buts la réduction des bruits, l'élimination des informations redondantes et parasites, la correction des formes mal positionnées dans ses images, etc.
- ◆ L'extraction des primitives servant à extraire des quantités d'informations assez significatives et propres à chacune des formes, ce qui permettra par la suite de réaliser une discrimination inter-formes, ceci par conséquent pourra faciliter l'étape suivante.

- ◆ L'apprentissage – classification ayant pour objectif de faire ramener un système de reconnaissance capable de mieux apprendre les différentes formes en vue de les bien classifier autrement dit les reconnaître avec justesse.

Notons aussi que pour bien comparer et améliorer les performances des systèmes proposés dans cette thèse, on a pu diversifier les techniques utilisées dans chaque phase. En effet, on a employé dans la première phase les techniques de seuillage, de centrage, de normalisation, d'amincissement et de rotation. En ce qui concerne la deuxième phase on a adopté les méthodes de Morphologie mathématique, le zonage et le zig-zag. Tandis qu'en troisième phase on a agréé le perceptron multicouche, les supports vecteurs machines, les k-plus proches voisins et finalement le modèle de Markov caché.

Après avoir mis en œuvre ces systèmes de reconnaissance, on a obtenu des résultats satisfaisants et lorsqu'on a combiné les différentes méthodes utilisées dans chacune des phases, on a réellement pu bien améliorer ces résultats.

Mots-clés : Reconnaissance de formes, bases de données des caractères Tifinaghs imprimés, manuscrits et cursifs, bases de données des chiffres Arabes MNIST, prétraitement des images, seuillage, centrage, normalisation, amincissement, rotation, extraction des primitives, morphologie mathématique, dilatation, zonage, zig-zag, apprentissage-classification, perceptron multicouche, supports vecteurs machines, k-plus proches voisins, modèle de Markov caché.

ABSTRACT

The automatic character recognition is processed by computer, which has as goal to translate a written text to a digitally encoded text. This means that the system is just like the human brain that recognizes the words and sentences existing in a known language rather than a sequence of characters. Since the late 1950s, the character recognition has become an active domain of research in computer science. In the beginning, we thought that it was an easy problem, but it appeared that it was a much more laborious subject, it will still take many decades to computers, if they succeed one day, to read all the documents with the same precision as human beings. There are many fields of character recognition applications; we can cite as examples the recognition of administrative documents, recognition of bank checks, recognition of signatures, etc.

This thesis proposes two categories of character recognition systems; the first category is interested to the recognition of printed, manuscript and cursive Tifinaghs characters. While the second is devoted to the recognition of Arab numerals that are extracted from the standard database MNIST.

Before citing the different envisaged methodologies to achieve our goal, we give an overview concerning the three phases of the construction of a pattern recognition system:

- ◆ Pre-processing which serves to improve the quality of images to recognize in order to be able to make them too readable and manipulable by a computer, this improvement aims the noise reduction, the elimination of redundant and parasites information's, the correction of the forms badly positioned in its images, etc.
- ◆ The extraction of primitives serving to extract some quantities' information enough significant and proper to each of the forms, this will allow thereafter to achieve an inter-discrimination of forms and can facilitate the next step.
- ◆ Learning – classification aimed to bring back a recognition system able to better learn the different forms in order to well classify them, in other words, recognize them with correctness.

We also note that in order to well compare and improve between the performances of the proposed systems in this thesis. We have diversified the techniques used in each phase; in fact, we used in the first phase the techniques of thresholding, centring, normalization, thinning and rotation. As regards, in the second phase we adopted the methods of mathematical morphology, the zoning and zig-zag. While in the third stage, we have chosen the multilayer perceptron, the support vector machines, the k-nearest neighbour and finally the hidden Markov model.

After having implemented these recognition systems, we have obtained satisfactory results, and when we have combined the classical methods used in each phase, we really have been able to improve these results.

Keywords: Pattern recognition, printed, manuscript and cursive Tifinagh character databases, Arab numerals MNIST database, pre-processing images, thresholding, centering, thinning, rotation, features extraction, mathematical morphology, dilation, zoning, zig-zag, learning – classification, multilayer perceptron, supports vectors machines, k-nearest neighbors, hidden Markov model.

ملخص

التعرف الأوتوماتيكي على الحروف هو عبارة عن معالجة آلية الهدف منها ترجمة نص مكتوب إلى آخر مرموز بشكل رقمي، إن طريقة إشتغال هذا النظام هي مماثلة إلى حد بعيد لطريقة إشتغال الدماغ البشري الذي يمكنه التعرف على الكلمات والعبارات الموجودة في نص ما. منذ أواخر الخمسينيات من القرن الماضي أصبح التعرف على الحروف مجالا يشهد العديد من الأبحاث وخاصة بعد إدماج أدوات جديدة في هذا المجال ونخص بالذكر هنا الحواسيب.

في البداية اعتقدنا أن الأمر يتعلق بموضوع بحث إلى حد ما سهل، لكن بعد ذلك أدركنا أن الأمر أكثر تعقيدا ويتطلب المزيد من العقود ليتم خلالها اكتشاف أنواع جديدة من الحواسيب أكثر تطورا تتمكن من التعرف على الوثائق المكتوبة بدقة تماثل تلك التي يتميز بها العقل البشري. مجالات تطبيق التعرف على الحروف هي عمليا عديدة نذكر منها التعرف على الشيكات المصرفية، الوثائق الإدارية، والإمضاءات، إلخ.

موضوع هذه الأطروحة يقترح إنجاز مقاربتين في مجال التعرف الأوتوماتيكي، حيث الأولى تهتم بالتعرف على الحروف الأمازيغية المسماة تيفيناغ سواء المطبوعة منها أو المكتوبة بخط اليد بشكل مباشر أو غير مباشر. والمقاربة الثانية تهتم التعرف على الأرقام العربية المكتوبة بخط اليد والمستخرجة من قاعدة البيانات العالمية المسماة "إم نيسيت".

قبل التطرق إلى مختلف المنهجيات التي استعملناها في كلا المقاربتين، نقدم لمحة مختصرة بخصوص المراحل الثلاثة التي يتكون منها نظام تعرف ما:

- ◆ العلاج القبلي لصور الحروف والأرقام والذي يهدف إلى تحسين جودتها.
- ◆ استخراج مميزات الصور والذي تهدف إلى استخلاص كمية من المعلومات ذات دلالة كافية لكل صورة.
- ◆ التعلم – التصنيف وهدفه يتلخص في جعل نظام التعرف قادرا على تمييز مختلف الأنماط لتسهيل عملية التعرف عليها بشكل جيد.

نشير إلى أنه بغية القيام بمقارنة وكذا من أجل تحسين أداء مختلف أنظمة التعرف المقترحة في هذه الأطروحة قمنا بتنويع التقنيات المستعملة في كل مرحلة من مراحل التعرف، في المرحلة الأولى استعملنا تقنيات مستوى العتبة، التمرکز والتوحيد ثم الدوران. في المرحلة الثانية قمنا باستخدام طرق المورفولوجيا الرياضية، التمديد، والتقسيم. أما بخصوص المرحلة الثالثة فضلنا الاستعانة بالطرق الرياضية: شبكة الخلايا العصبية الاصطناعية، فواصل هامش واسع، طريقة البحث عن أقرب شبيه، نماذج ماركوف المخفية.

بعد تطبيق مختلف أنظمة التعرف حصلنا على نتائج مرضية، لكن عندما قمنا بتوليفة مكونة من مختلف الطرق المستعملة في كل مرحلة من مراحل التعرف تمكنا فعلا من تحسين هذه النتائج.

كلمات البحث: التعرف على الأنماط، قاعدة البيانات، الحروف والأرقام، المعالجة القبلية، مستوى العتبة، التمرکز، التوحيد، الدوران، استخراج المميزات، التقسيم، التمديد، المورفولوجيا الرياضية، التعلم – التصنيف، الخلايا العصبية، فواصل هامش واسع، طريقة البحث عن أقرب شبيه، نماذج ماركوف المخفية.

TABLE DES MATIERES

	Page
REMERCIEMENTS	1
RÉSUMÉ	4
ABSTRACT	6
ملخص.....	8
TABLE DES MATIERES	9
LISTE DES FIGURES	16
LISTE DES TABLEAUX	21
LISTE DES ALGORITHMES	23
LISTE DES ABRÉVIATIONS ET SIGLES	24
CHAPITRE I INTRODUCTION GENERALE.....	26
I.1 Objectifs et problématiques	26
I.2 Motivations et contributions	27
I.3 Plan du rapport	29
I.4 Liste des publications & communications	31
I.4.1 Publications.....	31
I.4.2 Communications	32
PARTIE 1 ETAT DE L'ART, PRETRAITEMENT ET EXTRACTION DES PRIMITIVES	34
CHAPITRE II ETAT DE L'ART : LES DIFFERENTES TECHNIQUES UTILISEES DANS LA RECONNAISSANCE DES CARACTERES	35
II.1 Introduction	36
II.2 Mise en action d'un système de reconnaissance	37
II.2.1 Prétraitement de l'image	37
II.2.2 Extraction des primitives	38
II.2.3 Apprentissage - classification des formes	38
II.2.3.1 Distance Euclidienne	39
II.2.3.1.1 Définition	39
II.2.3.1.2 Reconnaissance des formes par la distance Euclidienne	39
II. 2.3.2 Réseaux de neurones RNs	39
II.2.3.2.1 Définition	39

Table des matières

II.2.3.2.2	Reconnaissance des formes par les RNs	40
II.2.3.3	Supports vecteurs machines SVM	41
II.2.3.3.1	Définition	41
II.2.3.3.2	Reconnaissance des formes par les SVMs	41
II.2.3.4	Méthode des K-plus proches voisins	42
II.2.3.4.1	Définition	42
II.2.3.4.2	Reconnaissance des formes par les K-ppv	43
II.2.3.5	Modèle de Markov caché MMC	43
II.2.3.5.1	Définition	43
II.2.3.5.2	Reconnaissance des formes par le MMC	45
II.3	Conclusion	46
CHAPITRE III	PRETRAITEMENT ET EXTRACTION DES PRIMITIVES	47
III.1	Acquisition de l'image	48
III.1.1	Le mode statique	49
III.1.2	Le mode dynamique	49
III.2	Prétraitement des images	50
III.2.1	Numérisation	51
III.2.2	Seuillage	51
III.2.3	Normalisation	51
III.2.4	Amincissement	52
III.2.5	Rotation.....	52
III.2.6	Centrage	53
III.2.7	Segmentation	53
III.2.7.1	La projection horizontale	53
III.2.7.2	La projection verticale	54
III.2.7.3	Le chevauchement des caractères	55
III.3	Extraction des primitives	55
III.3.1	Primitives statistiques	56
III.3.2	Primitives structurelles	56
III.3.2.1	Méthodes d'extraction des primitives simples (non hybrides)	56
III.3.2.1.1	La Morphologie mathématique	56

Table des matières

III.3.2.1.1.1	Dilatation par la Morphologie mathématique	57
III.3.2.1.1.2	Dilatation par un algorithme	57
III.3.2.1.1.3	Détection des zones caractéristiques de l'image	58
III.3.2.1.1.3.1	Extraction de la zone caractéristique d'Est	58
III.3.2.1.1.3.2	Extraction de la zone caractéristique d'Ouest	59
III.3.2.1.1.3.3	Extraction de la zone caractéristique du Sud	60
III.3.2.1.1.3.4	Extraction de la zone caractéristique du Nord	60
III.3.2.1.1.3.5	Extraction de la zone caractéristique Centrale	61
III.3.2.1.1.3.6	Vecteur d'extraction	61
III.3.2.1.2	Le Zonage	62
III.3.2.1.2.1	Extraction par le Zonage	62
III.3.2.1.2.2	Vecteur d'extraction	63
III.3.2.1.3	Le Zig-Zag	63
III.3.2.1.3.1	Extraction par le Zig-Zag	63
III.3.2.1.3.2	Vecteur d'extraction	63
III.3.2.2	Méthodes contribuées (hybrides)	64
III.3.2.2.1	Extraction par Morphologie – Zonage	64
III.3.2.2.2	Extraction par Morphologie – Zig Zag	64
III.3.2.2.3	Extraction par Morphologie – Zig Zag – Zonage	65
III.4	Conclusion	66
PARTIE 2 APRENTISSAGE – CLASSIFICATION, TESTS ET RESULTATS		67
CHAPITRE IV : L'APPRENTISSAGE – CLASSIFICATION		68
IV.1	Méthodes d'apprentissage – classification supervisées	70
IV.1.1	Les réseaux de neurones (RNs)	70
IV.1.1.1	Définition	71
IV.1.1.1.1	Le neurone biologique	71
IV.1.1.1.2	Le neurone formel	72
IV.1.1.1.3	Fonction d'activation	73
IV.1.1.2	Architecture d'un réseau de neurones	74
IV.1.1.2.1	Le perceptron monocouche	75

Table des matières

IV.1.1.2.2	Le perceptron multicouche	76
IV.1.1.3	Apprentissage – classification par les réseaux de neurones	77
IV.1.1.3.1	Définition	77
IV.1.1.3.2	Séparation des données	77
IV.1.1.3.2.1	Cas des problèmes linéairement séparables	77
IV.1.1.3.2.2	Cas des problèmes non linéairement séparables	80
IV.1.2	Les supports vecteurs machines (SVMs)	83
IV.1.2.1	Introduction	83
IV.1.2.2	Les SVMs binaires	84
IV.1.2.2.1	Les SVMs linéaires	84
IV.1.2.2.1.1	Le problème primal	86
IV.1.2.2.1.2	Le problème dual	87
IV.1.2.3	Les SVMs non linéaires	89
IV.1.2.4	Les SVMs multi-classes	91
IV.1.3	Les K-plus proches voisins (K-ppv)	91
IV.1.3.1	Principe de classification par K-ppv	92
IV.1.3.2	L'importance du choix de la distance	93
IV.1.3.3	L'importance du choix de nombre K	94
IV.1.3.4	Algorithme de K-ppv	94
IV.2	Méthodes d'apprentissage – classification non supervisées	95
IV.2.1	Le modèle de Markov caché (MMC)	95
IV.2.1.1	Processus stochastique	95
IV.2.1.2	Chaine de Markov	96
IV.2.1.3	Modèle de Markov caché	97
IV.2.1.3.1	Élément d'un MMC	97
IV.2.1.3.2	Les topologies de MMC	98
IV.2.1.3.3	Types de densité d'observations	98
IV.2.1.3.4	Problèmes fondamentaux associés au MMC	99
IV.2.1.3.4.1	Problème d'évaluation	99
IV.2.1.3.4.2	Problème d'apprentissage ou d'entraînement	100
IV.2.1.3.4.3	Problème de reconnaissance ou décodage	100
IV.2.1.4	Classification par le MMC	101

Table des matières

IV.2.2	Système Hybride PMC / MMC	102
IV.3	Conclusion	103
CHAPITRE V : TESTS ET RESULTATS		104
V.1	Introduction	105
V.2	Base de données des caractères Tifinaghs	105
V.2.1	Définition des caractères Tifinaghs	105
V.2.2	Base de données	106
V.2.2.1	Caractères Tifinaghs isolés manuscrits	106
V.2.2.2	Caractères Tifinaghs isolés imprimés	106
V.2.2.3	Caractères Tifinaghs isolés cursifs	107
V.2.2.4	Textes Tifinaghs	108
V.3	Base de données des chiffres Arabes	108
V.3.1	Chiffres manuscrits	108
V.3.2	Chiffres cursifs.....	109
V.4	Mise en œuvre des différents systèmes de reconnaissance de formes	109
V.4.1	Reconnaissance des chiffres Arabes	109
V.4.1.1	Reconnaissance des chiffres Arabes manuscrits MNIST par PMC ...	109
V.4.1.2	Reconnaissance des chiffres Arabes manuscrits MNIST par SVM .	113
V.4.1.3	Reconnaissance des chiffres Arabes manuscrits MNIST par K-ppv .	115
V.4.1.4	Reconnaissance des chiffres Arabes cursifs	120
V.4.2	Reconnaissance des caractères Tifinaghs isolés	123
V.4.2.1	Reconnaissance des caractères Tifinaghs imprimés et manuscrits	123
V.4.2.1.1	Reconnaissance par le PMC	123
V.4.2.1.2	Reconnaissance par le PMC et MMC.....	126
V.4.2.1.3	Reconnaissance des caractères Tifinaghs cursifs	129
V.4.2.1.3.1	Reconnaissance par le PMC	129
V.4.2.1.3.2	Reconnaissance par les SVMs	132
V.4.3	Reconnaissance des textes Tifinaghs	135
V.5	Conclusion	144

Table des matières

CHAPITRE VI : CONCLUSION GENERALE ET PERSPECTIVES	145
VI.1 Conclusion	145
VI.2 Perspectives	148
BIBLIOGRAPHIE	149

SOMMAIRE DES ANNEXES

	Page
ANNEXE A :	Interface graphique associée à l'application 164
ANNEXE B :	Algorithmes de dilatation 170
B1	Algorithme de dilatation vers l'Ouest 170
B2	Algorithme de dilatation vers le Nord 170
B3	Algorithme de dilatation vers le Sud 171
ANNEXE C :	Zones caractéristiques 172
C1	La zone caractéristique d'Est 172
C2	La zone caractéristique d'Ouest 172
C3	La zone caractéristique du Nord 173
C4	La zone caractéristique du Sud 173
C5	La zone caractéristique Centrale 174
ANNEXE D :	Démonstration de l'algorithme de Widrow-Hoff 175
ANNEXE E :	Démonstration de l'algorithme de rétro-propagation du gradient 176
E1	La correction de poids pour la couche de sortie 179
E2	La correction de poids pour la couche cachée 180
ANNEXE F :	Les paramètres du modèle de Markov caché 182
F1	Algorithme de Backward 182
F2	Algorithme de Forward 182
F3	Algorithme de Baum-Welch 183
F4	Algorithme de Viterbi 184
ANNEXE G :	Supports vecteurs machines 185
G1	Principe de Lagrange 185
G2	Principe de Karush - Kuhn - Tucker KKT 185
G3	Théorème de Mercer 186

LISTE DES FIGURES

	Page
FIGURE II.1	Le système de reconnaissance des formes 36
FIGURE II.2	Exemple du perceptron multicouche à une couche caché (l'un des modèles des réseaux de neurones) 40
FIGURE II.3	Linéarisation d'une séparation non linéairement séparable 41
FIGURE II.4	4-ppv appartenant aux trois classes différentes 43
FIGURE II.5	Les états cachés et observables du modèle MMC 44
FIGURE III.6	Exemple des caractères Tifinaghs acquis par un scanner et qui sont écrits par un marqueur 49
FIGURE III.7	Exemple des caractères Tifinaghs acquis par un scanner et qui sont écrits par un stylo 49
FIGURE III.8	Exemple des caractères Tifinaghs cursifs 50
FIGURE III.9	Image d'un caractère Tifinagh avant et après le seuillage 51
FIGURE III.10	Images d'un caractère Tifinagh avant la normalisation 51
FIGURE III.11	Images d'un caractère Tifinagh après la normalisation 52
FIGURE III.12	Image avant et après la technique d'amincissement 52
FIGURE III.13	L'image du caractère Tifinagh avant et après avoir la rotée 52
FIGURE III.14	L'image du caractère Tifinagh dans des différentes positions 53
FIGURE III.15	Image du caractère Tifinagh après le centrage 53
FIGURE III.16	Texte français avant une segmentation horizontale 54
FIGURE III.17	Calcul de l'histogramme d'une projection horizontale 54
FIGURE III.18	Texte français après une segmentation horizontale 54
FIGURE III.19	Texte avant une segmentation verticale 54
FIGURE III.20	Calcul de l'histogramme d'une projection verticale 55
FIGURE III.21	Texte après la segmentation verticale 55

Liste des figures

FIGURE III.22	Dilatation d'un caractère Tifinagh vers L'EST	57
FIGURE III.23	Dilatations d'un caractère Tifinagh vers L'OUEST, NORD et SUD	57
FIGURE III.24	La zone caractéristique d'Est	59
FIGURE III.25	La zone caractéristique d'Ouest	60
FIGURE III.26	La zone caractéristique du Sud	60
FIGURE III.27	La zone caractéristique du Nord	61
FIGURE III.28	La zone caractéristique centrale	61
FIGURE III.29	Exemples d'extraction des primitives par le Zonage d'un caractère Tifinagh « yar »	62
FIGURE III.30	Exemples d'extraction des primitives par le Zig-Zag d'un caractère Tifinagh « yar »	63
FIGURE III.31	Exemple d'extraction des primitives par la dilatation et le Zonage d'un caractère Tifinagh « yar »	64
FIGURE III.32	Exemples d'extraction des primitives par la dilatation et le Zig-Zag d'un caractère Tifinagh « yar »	65
FIGURE III.33	Exemples d'extraction des primitives par hybridation des trois méthodes du caractère « yar »	65
FIGURE IV.34	Représentation de la forme biologique d'un neurone	72
FIGURE IV.35	Architecture d'un neurone formel	72
FIGURE IV.36	La représentation graphique de la fonction sigmoïde	73
FIGURE IV.37	La représentation graphique de la fonction $\tanh(x)$	73
FIGURE IV.38	La représentation graphique de la fonction linéaire	74
FIGURE IV.39	La représentation graphique de la fonction Heaviside	74
FIGURE IV.40	La représentation graphique de la fonction signe	74
FIGURE IV.41	Architecture du perceptron	75
FIGURE IV.42	Architecture du perceptron multicouche	76

Liste des figures

FIGURE IV.43	La séparation linéaire	78
FIGURE IV.44	La séparation de « OU logique »	79
FIGURE IV.45	La séparation de « ET logique »	80
FIGURE IV.46	L'impossibilité de séparation de « XOR »	80
FIGURE IV.47	Séparation linéaire entre deux classes avec un SVM	85
FIGURE IV.48	Différentes démarches pour trouver le problème primal	87
FIGURE IV.49	Présentation de la marge souple	89
FIGURE IV.50	Transformation d'une séparation non linéaire dans \mathbb{R}^2 en une autre linéaire dans \mathbb{R}^3	89
FIGURE IV.51	Les frontières de décision obtenues par trois SVM à fonction noyau Linéaire, polynomial et RBF	91
FIGURE IV.52	5-ppv du vecteur inconnu X qui appartient à deux classes différentes	92
FIGURE IV.53	Chaine de Markov modélisée par un graphe	97
FIGURE IV.54	Les modèles : Ergodique, Gauche-droite : parallèle	98
FIGURE IV.55	Représentation de décodage	101
FIGURE IV.56	Processus de reconnaissance par un MMC	102
FIGURE IV.57	Schéma global de conception du modèle hybride PMC/MMC	103
FIGURE V.58	L'alphabet Tifinagh	106
FIGURE V.59	Exemple d'un texte acquis par un scanner avec un marqueur	108
FIGURE V.60	Exemple d'un texte acquis par un scanner avec un stylo.....	108
FIGURE V.61	Chiffres isolés de la base MNIST	109
FIGURE V.62	La comparaison entre le taux de reconnaissance des deux bases de test et validation	111
FIGURE V.63	La reconnaissance de chiffres Arabes par le perceptron multicouche	112

Liste des figures

FIGURE V.64	Taux de reconnaissance des chiffres Arabes MNIST en fonction de nombre des zones avec SVM	115
FIGURE V.65	Taux de reconnaissance des chiffres Arabes MNIST en fonction de nombre des zones avec K-ppv	117
FIGURE V.66	Le taux de reconnaissance des chiffres en fonctions des méthodes d'extractions avec K-ppv	118
FIGURE V.67	Interface graphique mise en place pour reconnaitre les chiffres	120
FIGURE V.68	La reconnaissance de chiffres cursifs par PMC et K-PPV	122
FIGURE V.69	Interface graphique mise en place pour reconnaitre les chiffres cursifs	123
FIGURE V.70	Taux de reconnaissances des caractères Tifinaghs manuscrits et imprimés avec le PMC	124
FIGURE V.71	Image extraites de la base des caractères Tifinaghs manuscrits	125
FIGURE V.72	Image extraites de la base des caractères Tifinaghs imprimés	125
FIGURE V.73	Les taux de reconnaissance des caractères Tifinaghs imprimés par le PMC, MMC et PMC+ MMC	127
FIGURE V.74	Les taux de reconnaissance des caractères Tifinaghs manuscrits par le PMC, MMC et PMC+MMC	129
FIGURE V.75	Les taux de la reconnaissance des caractères Tifinaghs cursifs par la méthode de Morphologie en fonction du nombre des neurones.....	131
FIGURE V.76	Les taux de reconnaissance des caractères Tifinaghs cursifs par la Morphologie et le Zonage en fonction du nombre des neurones	132
FIGURE V.77	Les taux de la reconnaissance des caractères Tifinaghs cursifs par la méthode de Morphologie +Zonage par SVM	134
FIGURE V.78	Interface graphique mise en place pour reconnaitre les caractères Tifinaghs cursifs	135
FIGURE V.79	Exemple d'un texte Tifinagh	136
FIGURE V.80	La segmentation en ligne du texte Tifinagh	137

Liste des figures

FIGURE V.81	La segmentation de la première ligne	137
FIGURE V.82	La segmentation de la deuxième ligne	137
FIGURE V.83	La segmentation de la troisième ligne	137
FIGURE V.84	La segmentation de la quatrième ligne	137
FIGURE V.85	La segmentation de la cinquième ligne	137
FIGURE V.86	La segmentation en colonne de la troisième ligne	138
FIGURE V.87	Représentation graphique de la segmentation en colonne de la troisième ligne	138
FIGURE V.88	Caractères segmentés en colonne	139
FIGURE V.89	Interface graphique mise en place pour reconnaître un texte Tifinagh	140
FIGURE V.90	Taux de reconnaissance d'un texte Tifinagh écrit par un marqueur	142
FIGURE V.91	Taux de reconnaissance d'un texte Tifinagh écrit par un stylo	143
FIGURE A.92	Interface de la page d'accueil	164
FIGURE A.93	Interface mise en place pour le processus de la reconnaissance	165
FIGURE A.94	Interface graphique mise en place pour le prétraitement	166
FIGURE A.95	Interface graphique mise en place pour l'extraction par la Morphologie	167
FIGURE A.96	Interface graphique mise en place pour l'extraction par le Zonage ...	168
FIGURE A.97	Interface graphique mise en place pour la reconnaissance	169
FIGURE E.98	Le perceptron multicouche n'ayant qu'une seule couche cachée	176

LISTE DES TABLEAUX

		Page
TABLEAU V.I	Exemple de quelques caractères Tifinaghs manuscrits	106
TABLEAU V.II	Exemple de quelques caractères Tifinaghs imprimés.....	107
TABLEAU V.III	Exemple de quelques caractères Tifinaghs cursifs	107
TABLEAU V.IV	Exemple de quelques chiffres cursifs	109
TABLEAU V.V	Le taux de reconnaissance de plusieurs ensemble des chiffres par PMC	111
TABLEAU V.VI	Le taux de reconnaissance obtenu de chaque chiffre par PMC	112
TABLEAU V.VII	La matrice de confusion des chiffres par PMC	113
TABLEAU V.VIII	Le taux de reconnaissance des chiffres Arabes MNIST en fonction de nombre des zones par SVM	114
TABLEAU V.IX	Le taux de reconnaissance des chiffres Arabes MNIST en fonction de nombre des zones par K-ppv.	116
TABLEAU V.X	Le taux de reconnaissance des chiffres en fonctions des méthodes d'extractions avec K-ppv	118
TABLEAU V.XI	Le taux de reconnaissance de chaque chiffre cursif par PMC et K-ppv	121
TABLEAU V.XII	Le taux de reconnaissance de plusieurs ensemble des caractères par PMC	124
TABLEAU V.XIII	Les taux de reconnaissance τ_{rc} et τ_{rg} des caractères imprimés pour les trois méthodes MMC, PMC et PMC+MMC	126
TABLEAU V.XIV	Les taux de reconnaissance τ_{rc} et τ_{rg} des caractères manuscrits pour les trois méthodes MMC, PMC et PMC+MMC	128
TABLEAU V.XV	Les taux de reconnaissance obtenus par la Morphologie et Morphologie + Zonage par PMC	130
TABLEAU V.XVI	Les taux de reconnaissance obtenus par la Morphologie et Morphologie + Zonage par SVM.....	133
TABLEAU V.XVII	Le taux d'apparition des caractères Tifinaghs	140

TABLEAU V.XVIII	Taux de reconnaissance d'un texte Tifinagh écrit par un marqueur	141
TABLEAU V.XIX	Taux de reconnaissance d'un texte Tifinagh écrit par un stylo	142

LISTE DES ALGORITHMES

	Page
ALGORITHME III.1	La dilatation vers l'EST 58
ALGORITHME V.2	L'algorithme de Widrow-Hoff 77
ALGORITHME V.3	L'algorithme de rétro-propagation du gradient 82
ALGORITHME V.4	Algorithme de K-ppv 94
ALGORITHME V.5	Algorithme de Forward 99
ALGORITHME V.6	Algorithme de Backward 99
ALGORITHME V.7	Algorithme de Baum-Welch 100
ALGORITHME V.8	Algorithme de Viterbi 101
ALGORITHME B.9	Algorithmes de dilatation 170
ALGORITHME B1.10	Algorithmes de dilatation vers l'Ouest 170
ALGORITHME B2.11	Algorithmes de dilatation vers le Nord 170
ALGORITHME B3.12	Algorithmes de dilatation vers le Sud 171

LISTE DES ABRÉVIATIONS ET SIGLES

RN :	Réseaux de neurones
PMC :	Perceptron multicouches
MMC :	Modèle de Markov caché
ZE :	Zone caractéristique d'Est
ZO :	Zone caractéristique d'Ouest
ZS :	Zone caractéristique du Sud
ZN :	Zone caractéristique du Nord
ZC :	Zone caractéristique Centrale
N_{ze} :	Nombre de pixels de niveau 1 dans la zone caractéristique Est
N_{zo} :	Nombre de pixels de niveau 1 dans la zone caractéristique Ouest
N_{zs} :	Nombre de pixels de niveau 1 dans la zone caractéristique Sud
N_{zn} :	Nombre de pixels de niveau 1 dans la zone caractéristique Nord
N_{zc} :	Nombre de pixels de niveau 1 dans la zone caractéristique Centrale
N_{pixels} :	Nombre de pixels dans l'image traitée
V_{extraction} :	Vecteur d'extraction caractéristique
φ :	Fonction d'activation
W :	Matrice de connexion pour la couche cachée
Z :	Matrice de connexion pour la couche de sortie
X :	Vecteur d'entrée
O :	Vecteur de sortie désiré
T :	Vecteur de sortie trouvé

Liste des abréviations et sigles

η :	Coefficient d'apprentissage
A :	Matrice de transitions
B :	Matrice d'observations
Π :	Vecteur d'initialisation
λ :	Modèle de Markov
SVM :	Supports vecteurs machines
K-ppv :	K- plus proches voisins

CHAPITRE I

INTRODUCTION GENERALE

Contenu du chapitre

<i>I.1</i>	<i>Objectifs et problématiques</i>	<i>26</i>
<i>I.2</i>	<i>Motivations et contributions</i>	<i>27</i>
<i>I.3</i>	<i>Plan du rapport</i>	<i>29</i>
<i>I.4</i>	<i>Liste des publications & communications</i>	<i>31</i>
<i>I.4.1</i>	<i>Publications</i>	<i>31</i>
<i>I.4.2</i>	<i>Communications</i>	<i>32</i>

I.1 Objectifs et problématiques

Récemment le domaine du traitement de l'information a effectivement connu une mutation géante et exceptionnelle en vertu d'insertion des nouvelles machines intelligentes notamment les ordinateurs [56, 81, 85, 97, 106, 131, 140, 162]. En effet, depuis une soixantaine d'années, le progrès technologique a rendu la relation entre l'homme et la machine trop faisable et trop rentable. Parmi les relations importantes entre l'homme et l'ordinateur, il y a celle qui est consacrée à la reconnaissance automatique de l'écriture des caractères avec ses différentes formes imprimées ou manuscrites. Ce type de reconnaissance occupe actuellement une grande place dans plusieurs secteurs tels que la reconnaissance des chèques bancaires, la reconnaissance des matricules des automobiles, la reconnaissance des documents et des dossiers administratifs, la reconnaissance des signatures, le traitement des adresses postales et tri automatique, la lecture des documents patrimoniaux et l'indexation d'archives bibliothécaires.

L'informatisation d'un tel caractère ou d'un certain texte reste un problème trop difficile à automatiser en tenant compte de la grande variabilité liée aux scripteurs et aux styles d'écriture. Certainement, le fait de lire ou d'écrire un texte par un homme n'est pas une tâche évidemment réalisable par un ordinateur. De plus, l'approche n'est pas la même selon qu'il s'agira de reconnaître un caractère imprimé ou manuscrit, l'écriture manuscrite est sans aucun doute est plus complexe que celle imprimée en raison de son extrême variabilité y inclus les

ligatures remarquables et le graphisme qui est inégalement ordonné, ceci est encore trop aperçu lorsqu'on passe d'une personne à une autre. Même pour une personne donnée, ce type d'écriture n'est jamais parfaitement stable, ceci rend intuitivement la reconnaissance trop compliquée. Tandis que dans le cas imprimé, les caractères sont bien alignés et souvent bien séparés verticalement, ce qui permettra de faciliter d'une façon ou d'une autre la reconnaissance.

Ainsi, avec toutes ces problématiques parvenant de la reconnaissance des caractères, en particulier quand il s'agira de ceux manuscrits, on a opté dans ce travail à la reconnaissance automatique des caractères comme un objectif à aboutir.

I.2 Motivations et contributions

La reconnaissance automatique de l'écriture manuscrite ou imprimée reste un sujet de recherche très actif. Elle a réellement connu des progrès exponentiels du fait qu'une multitude de travaux qui ont été réalisés particulièrement durant la dernière décennie [12, 13, 22, 29, 58, 106, 132, 140, 179], s'intéressent à la combinaison entre que ce soient des techniques de prétraitement ou méthodes d'extraction des primitives, ou des diverses méthodes d'apprentissage - classification en vue de mettre en place des systèmes de reconnaissance plus efficaces [30, 95, 99, 108, 159, 174, 196].

Durant les dernières années, plusieurs recherches scientifiques ont été effectuées sur la reconnaissance des caractères latins [30, 57], japonais, chinois [119, 120, 122, 125, 185, 193, 195], indiens [19, 26, 29, 84, 102, 110, 128, 129] et arabes [3, 8, 9, 12, 68, 105, 143, 176, 179], ce qui a permis le développement de plusieurs approches de reconnaissance automatique de l'écriture de ces langues. Cependant, les caractères Tifinaghs sont très peu traités, quelques tentatives ont été menées pour améliorer la situation actuelle [14, 16, 17, 70, 71].

Introduction

À présent, la reconnaissance des caractères Tifinaghs reste encore un sujet intéressant et large de recherches et d'expérimentations. En effet au niveau national, les mises en œuvre qui ont été réalisées afin d'utiliser la langue Amazighe dans les différents secteurs publiques tels que son intégration graduelle dans les médias nationaux écrits et audiovisuels. Ainsi que sa généralisation dans la vie publique comme son insertion progressive dans le secteur d'enseignement et d'éducation à travers tout le pays. Tout cela avait effectivement généré un besoin qui s'agrandira de plus en plus et jour après jour de faire renforcer énormément de recherches scientifiques pour l'automatisation de l'écriture Amazighe. D'autre part tenant compte de l'importance ascendante de cette langue, le leader mondial de conception et de création de logiciels et de solutions vient d'intégrer l'Amazighe à Windows. De ce fait cette langue rejoint ainsi les 109 langues utilisées dans le monde informatique. Tout ce dynamisme nous a bien motivé, mieux encouragé de choisir un sujet s'intéressant à la reconnaissance des caractères Tifinaghs y inclus ceux imprimés, manuscrits et cursifs.

Signalons aussi que pour diversifier nos travaux, on a élaboré de nouveaux systèmes de reconnaissance des chiffres Arabes manuscrits ainsi que cursifs.

Pour fixer les idées et bien mettre en évidence tous les systèmes de reconnaissances que nous avons réalisés, on a choisi dans la phase de prétraitement plusieurs techniques qui sont le seuillage, le centrage, la normalisation et l'amincissement. En ce qui concerne la phase d'extraction des primitives, on a adopté de nombreuses méthodes qui font partie de la famille de méthodes dites structurelles (voir chapitre III) comme la Morphologie mathématique, le Zonage et le Zig-Zag. Finalement la phase d'apprentissage-classification a été effectuée via diverses méthodes mathématiques qui sont divisés en deux catégories principales supervisée (Perceptron multicouches, Séparateurs à vaste marge et K-plus proches voisins) et non supervisée (Modèle de Markov caché).

I.3 Plan du rapport

Ce rapport est organisé selon la planification suivante :

D'abord on a commencé par une introduction générale. Ensuite, on présentera les différentes problématiques rencontrées dans un tel système de reconnaissance des caractères. Après, on donnera quelques définitions et descriptions au sens large du mot des différentes phases d'un certain système de reconnaissance. Le rapport se terminera par une citation et une analyse de tous les résultats issus des expériences réalisées, tout en comparant ces systèmes pour en déduire les plus performants.

Chapitre 1 : Introduction générale

Ce chapitre contient trois paragraphes principaux qui se sont respectivement objectifs et problématiques, motivations et contributions, et plan du rapport. Le premier paragraphe présente les différents facteurs qui ont rendu le domaine de la reconnaissance de l'écriture trop attrayant, et les problématiques rencontrées et les défis à relever dans ce domaine. Le deuxième paragraphe cite les différentes raisons qui nous ont motivé et nous ont encouragé à choisir ce sujet de thèse ainsi que nos contributions. Le dernier paragraphe explique comment ce rapport a été planifié.

Chapitre 2 : Etat de l'art des différentes techniques utilisées pour la reconnaissance des caractères

Ce chapitre est consacré aux différentes méthodes et techniques utilisées dans un système de reconnaissance des caractères, basé sur les trois phases fondamentales qui sont le prétraitement, l'extraction des primitives et l'apprentissage-classification. Pour donner un éclaircissement concernant chacune de ces phases, on a expliqué le pourquoi et le comment de la mise en œuvre de toutes les démarches à suivre durant chaque phase.

Chapitre 3 : Prétraitement et Extraction des primitives

Ce chapitre présente toutes les techniques qu'on a utilisées en vue de prétraiter les images des caractères à reconnaître. Ainsi que toutes les méthodes d'extraction des primitives qui sont très

Introduction

significatives et trop propres pour chaque forme à reconnaître. Au cours des différents systèmes de reconnaissance réalisés, on a fait appel à plusieurs techniques de prétraitement à savoir le seuillage, le centrage, la normalisation, l'amincissement et la rotation. Également, on a adopté suffisamment de méthodes pour extraire efficacement les primitives des caractères à savoir la Morphologie, le Zonage et le Zig-Zag et aussi quelques combinaisons deux à deux ou trois à trois de ces méthodes.

Chapitre 4 : Les méthodes d'apprentissage – classification

Ce chapitre explique bien toutes les méthodes d'apprentissage-classification qu'on a considérées dans nos expériences que ce soient celles qui font partie d'un type supervisé ou du type non supervisé. En effet ces explications traitent d'une manière bien détaillée les fondements théoriques ainsi que pratiques de ces méthodes. A ce propos, parmi les méthodes supervisées, on a mis en action le perceptron multicouche, les supports vecteurs machines et les k-plus proches voisins. En revanche, on s'est focalisé au modèle de Markov caché quand il s'agit des méthodes résultant du type non supervisé.

Chapitre 6 : Tests et résultats

Dans ce dernier chapitre, on a cité extensivement tous les résultats obtenus après avoir mis en œuvre tous les tests. Afin de bien expliquer ces résultats on y insèrera des analyses et des commentaires y compris des comparaisons entre les performances des différents systèmes de reconnaissance réalisés.

Enfin, on terminera ce rapport par une conclusion générale ainsi que des perspectives en vue de poursuivre nos travaux de recherche dans le domaine global de la reconnaissance automatique des formes.

I.4 Liste des publications & communications

I.4.1 Publications :

- B. El kessab, C. Daoui, B. Bouikhalene, M. Fakir, K.Moro. **EXTRACTION METHOD OF HANDWRITTEN DIGIT RECOGNITION TESTED ON THE MNIST DATABASE**, International Journal of Advanced Science and Technology Vol. 50, January, 2013
- B. El kessab, C. Daoui, B. Bouikhalene, M. Fakir, K.Moro. **HANDWRITTEN TIFINAGH TEXT RECOGNITION USING NEURAL NETWORKS AND HIDDEN MARKOV MODELS**, International Journal of Computer Applications (0975 – 8887) Volume 75– No.18, August 2013
- B. El kessab, B. Bouikhalene, C. Daoui, M. Fakir, K.Moro. **UTILISATION DES RESEAUX DE NEURONES POUR LA RECONNAISSANCE DES CARACTERES TIFINAGHE MANUSCRITS**, Journal Méditerranéen de Télécommunication, vo.2, n°1, janvier 2012
- B. El kessab, C. Daoui, K.Moro, B. Bouikhalene, M. Fakir. **RECOGNITION OF HANDWRITTEN TIFINAGH CHARACTERS USING A MULTILAYER NEURAL NETWORKS AND HIDDEN MARKOV MODEL**, Global Journal of Computer Science and Technology Volume 11 Issue 15 September 2011
- K.Moro, M. Fakir, B. El kessab, B. Bouikhalene, C. Daoui. **COMPARISON OF TWO FEATURE EXTRACTION METHODS BASED ON THE RAW FORM AND HIS SKELETON FOR GUJARATI HANDWRITTEN DIGITS**, FACTA UNIVERSITATIS (NIS). SER. MATH. INFORM. Vol 28, No 2 (2013), 161-178.
- K.Moro, M. Fakir, B. El kessab, B. Bouikhalene, C. Daoui. **GUJARATI HANDWRITTEN NUMERAL OPTICAL CHARACTER THROUGH NEURAL NETWORK AND SKELETONIZATION**, Jurnal Sistem Komputer - Vol.3, No.1, Juni 2013.

Introduction

- K.Moro, M. Fakir, B. El kessab, B. Bouikhalene, C. Daoui. **COMPARISON OF TWO FEATURE EXTRACTION METHOD BASED ON THE RAW FORM AND HIS SKELETON FOR MNIST DATABASE**, International Journal of Computer Science Research and Application 2012, Vol. 2, Issue 3, pp.13-22
- K.Moro, M. Fakir, B. El kessab, B. Bouikhalene, C. Daoui. **FEATURE EXTRACTION BASED ON THE RAW FORM AND HIS SKELETON**, GJCAT, Vol 1 (3), 2011
- B. EL KESSAB, C. DAOU, R. SALOUAN (2014), **SOME COMPARATIVE STUDIES FOR CURSIVE HANDWRITTEN TIFINAGH CHARACTERS RECOGNITION SYSTEMS**, soumis au Journal: International Journal of Advanced Science and Technology.
- B. EL KESSAB, C. DAOU, B. BOUIKHALENE, R. SALOUAN (2014), **A COMPARATIVE STUDY BETWEEN THE SUPPORT VECTORS MACHINES AND THE K-NEAREST NEIGHBORS IN THE HANDWRITTEN LATIN NUMERALS RECOGNITION**, soumis au Journal: Elsevier Pattern Recognition Letters.

I.4.2 Communications :

- B. El kessab, C. Daoui, B. Bouikhalene, M. Fakir, R. Salouan, K.Moro. **HANDWRITTEN DIGITS RECOGNITION USING THE K-NEAREST NEIGHBOURS AND HIDDEN MARKOV MODEL**, CBI' 2014, Beni Mellal MOROCCO, April 29,30, 2014.
- B. El kessab, C. Daoui, B. Bouikhalene, M. Fakir, K.Moro. **METHOD ON MATHEMATICAL MORPHOLOGY AND NEURAL NETWORKS FOR THE RECOGNITION OF HANDWRITTEN TIFINAGHE CHARACTERS**, ICMCS' 2011, Ouarzazate MOROCCO, April 07,09, 2011.

Introduction

- B. El kessab, C. Daoui, B. Bouikhalene, M. Fakir, K.Moro. **APPROACH BASED ON MATHEMATICAL MORPHOLOGY, NEURAL NETWORKS AND HIDDEN MARKOV MODELS FOR THE RECOGNITION OF HANDWRITTEN TIFINAGHE CHARACTERS**, SITACAM'11, Agadir-Maroc, 13-14 Mai 2011
- B. El kessab, C. Daoui, K.Moro, B. Bouikhalene, M. Fakir. **UTILISATION DES RESEAUX DE NEURONES POUR LA RECONNAISSANCE DE CHIFFRES MANUSCRITS**, La Première Journée Nationale de Mathématiques et applications en Informatique et Télécommunication, Béni Mellal, 02 Juillet 2011.
- K.Moro, B. El kessab, M. Fakir, B. Bouikhalene, S. Safi. **CONTRIBUTION A LA RECONNAISSANCE DES CARACTERES TIFINAGH PAR UTILISATION DES RESEAUX DE NEURONES ET LA SQUELETTISATION**, SITACAM'11, Agadir-Maroc, 13-14 Mai 2011
- R.Salouan, S. Safi, B. Bouikhalene, B. El kessab. **EASTERN PRINTED ARABIC NOISY NUMERALS RECOGNITION**, CBI' 2014, Beni Mellal MOROCCO, April 29,30, 2014.

PARTIE 1: ETAT DE L'ART, PRETRAITEMENT ET EXTRACTION DES PRIMITIVES

CHAPITRE II

ETAT DE L'ART : LES DIFFERENTES TECHNIQUES UTILISEES DANS LA RECONNAISSANCE DES CARACTERES

La reconnaissance optique des caractères (OCR : Optical Character Recognition) est considérée aujourd'hui comme l'une des applications les plus puissantes et plus performantes dans la reconnaissance automatique des formes. En effet ce domaine est tellement dynamique au niveau des recherches et des développements.

En terme simple, les étapes nécessaires pour pouvoir mettre en place un tel OCR sont les trois phases habituelles déjà citées qui sont le prétraitement, l'extraction des primitives et l'apprentissage-classification.

Plusieurs méthodes ont été utilisées dans ce propos selon chaque phase et suivant le type de caractère traité (isolé ou non, manuscrit ou imprimé) et selon la langue considérée.

Contenu du chapitre

<i>II.1</i>	<i>Introduction</i>	<i>36</i>
<i>II.2</i>	<i>Mise en action d'un système de reconnaissance</i>	<i>37</i>
<i>II.2.1</i>	<i>Prétraitement de l'image</i>	<i>37</i>
<i>II.2.2</i>	<i>Extraction des primitives</i>	<i>38</i>
<i>II.2.3</i>	<i>Apprentissage - classification des formes</i>	<i>38</i>
<i>II.2.3.1</i>	<i>Distance Euclidienne</i>	<i>39</i>
<i>II.2.3.1.1</i>	<i>Définition</i>	<i>39</i>
<i>II.2.3.1.2</i>	<i>Reconnaissance des formes par la distance Euclidienne</i>	<i>39</i>
<i>II.2.3.2</i>	<i>Réseaux de neurones RNs</i>	<i>39</i>
<i>II.2.3.2.1</i>	<i>Définition</i>	<i>39</i>
<i>II.2.3.2.2</i>	<i>Reconnaissance des formes par les RNs</i>	<i>40</i>
<i>II.2.3.3</i>	<i>Supports vecteurs machines SVM</i>	<i>41</i>
<i>II.2.3.3.1</i>	<i>Définition</i>	<i>41</i>
<i>II.2.3.3.2</i>	<i>Reconnaissance des formes par les SVMs</i>	<i>41</i>
<i>II.2.3.4</i>	<i>Méthode des K-plus proches voisins K-ppv</i>	<i>42</i>
<i>II.2.3.4.1</i>	<i>Définition</i>	<i>42</i>

<i>II.2.3.4.2</i>	<i>Reconnaissance des formes par les K-ppv</i>	<i>43</i>
<i>II.2.3.5</i>	<i>Modèle de Markov Caché MMC</i>	<i>43</i>
<i>II.2.3.5.1</i>	<i>Définition</i>	<i>43</i>
<i>II.2.3.5.2</i>	<i>Reconnaissance des formes par le MMC</i>	<i>45</i>
<i>II.3</i>	<i>Conclusion</i>	<i>46</i>

II.1 Introduction

La reconnaissance de formes y inclus l'OCR consiste à faire associer une étiquette à une forme donnée afin de déterminer la classe pour laquelle la forme s'appartient. En vue de pouvoir réaliser un tel système de reconnaissance des formes, il fallait suivre forcément les étapes suivantes :

- Numérisation : s'effectuera pour que la représentation des formes à reconnaître soit facilement manipulable par un ordinateur.
- Prétraitement : sert à éliminer des bruits, normaliser, ré-échantillonner et améliorer des contrastes, etc.
- Extraction des primitives : ayant pour objectif de représenter les formes d'une manière compatible avec les méthodes d'apprentissage et de décision utilisées.
- Apprentissage-classification : consiste à rendre un tel système de reconnaissance très capable d'apprendre les différentes formes en vue de les classifier justement par la suite.

Les étapes d'instauration d'un certain système OCR peuvent être schématisées comme suit:

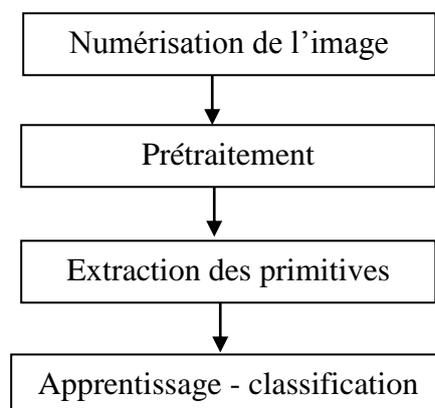


Figure II.1 : Le système de reconnaissance des formes

II.2 Mise en action d'un système de reconnaissance

Un tel système de reconnaissance des formes sert à pouvoir analyser une forme qui peut être un caractère, un chiffre, une empreinte ou une signature, etc. Tout en la donnant d'une manière simple une version informatique, ce qui facilitera par la suite sa lisibilité par une machine intelligente particulièrement un ordinateur. Pour ce faire et afin qu'un système de reconnaissance des formes puisse se mettre en action, les étapes suivantes doivent être forcément suivies :

II.2.1 Prétraitement de l'image :

D'abord avant de procéder à la reconnaissance des formes une étape de prétraitement est obligatoire. Cette étape ayant pour objectif d'améliorer la qualité de l'image traitée du fait que celle-ci a été en principe est mal visualisée, cette amélioration est effectivement accessible si on fera appel aux diverses techniques servant à supprimer les informations parasites ou redondantes, réduire autant que possible le bruit existant dans les images brutes, redresser les formes inclinées ou déformées dans les images, corriger le contraste et binariser les images. Ceci permettra par conséquent d'obtenir des images trop nettes, ce qui facilitera bien évidemment sa reconnaissance.

Plusieurs techniques utilisées dans ce niveau comme :

1. La numérisation : sert à discrétiser les images afin de faciliter leur lisibilité par les ordinateurs.
2. Le seuillage : permet de faire binariser les images c'est-à-dire les ramener n'incluant que les couleurs noirs et blancs selon un seuil fixé au paravent.
3. Le centrage : permet de bien positionner la forme englobée par l'image juste au centre de cette dernière.
4. La normalisation a pour but de rendre les formes contenues dans les images ayant des tailles égales.
5. L'amincissement est une technique qui permet d'uniformiser les différentes épaisseurs des formes.
6. La rotation est une opération qui s'utilise en vue de bien localiser les formes dans les images qui les incluent.

7. La segmentation s'exploite pour décomposer les textes existants dans des images en des mots ou bien en des caractères.

II.2.2 Extraction des primitives :

C'est l'étape nécessaire qui précède la phase de reconnaissance des caractères, elle consiste à effectuer une extraction des caractéristiques dites aussi des primitives qui ne sont rien d'autres que des quantités d'informations trop significatives caractérisant l'image à reconnaître, Cette extraction est réalisé par des méthodes structurelles (Zonage, Zig-Zig, ...) ou statistiques (les moments de Hu, de Zernike, ...), en vue de convertir l'image traitée en un vecteur caractéristique de données, ce qui permettra de bien faciliter son apprentissage ainsi que sa classification.

Les méthodes exploitées dans ce travail sont :

1. La Morphologie mathématique qui permettra de faire dilater, éroder, ouvrir ou fermer les formes afin de pouvoir extraire efficacement ses primitives.
2. Le Zonage qui consiste à découper une image d'une telle forme en plusieurs zones puis mettre en comptage le nombre de pixels appartenant à la forme (qui sont des pixels blancs si cette forme ayant un allure blanc dans une image noire ou bien sont des pixels noirs dans le cas contraire) dans chaque zone ce qui rendra la conversion de l'image en un vecteur tout à fait permmissible.
3. Le Zig-Zag servant à effectuer un balayage d'une façon zigzagué de l'image colonne par colonne puis linge par ligne que ce soit horizontal ou vertical ou parallèle au diagonale ou au anti diagonale, puis à compter le nombre de pixels qui appartiennent à la forme dans chaque ligne ou colonne balayée. Tous ces nombres seront stockés dans un vecteur.

II.2.3 Apprentissage - classification des formes :

Juste après la phase d'extraction, une autre phase d'apprentissage - classification est forcément nécessaire afin de ramener le système de reconnaissance capable de bien apprendre les images des formes de la base d'apprentissage pour qu'il pourra classifier correctement les autres images de la base de test.

Cette phase pouvant être effectuable en vertu de plusieurs méthodes à titre d'exemples la distance Euclidienne, les réseaux de neurones, les séparateurs à vaste marge, les k-plus proches voisins et les modèles de Markov cachés, etc.

Alors que le comment de la mise en œuvre de chacune de ces méthodes peut être expliqué comme suit :

II.2.3.1 Distance Euclidienne

II.2.3.1.1 Définition

Dans un espace vectoriel normé $(\mathbf{E}, \|\cdot\|)$, on pourra toujours définir de manière canonique une distance \mathbf{d} à partir d'une norme. En effet, il nous suffira de mettre :

$$\forall (x, y) \in \mathbf{E} \times \mathbf{E}, \quad \mathbf{d}(x, y) = \|x - y\| \quad (1)$$

En particulier, dans \mathbb{R}^n ($n \in \mathbb{N}$), on pourra définir par plusieurs manières la distance entre deux vecteurs, cependant la distance la plus utilisée couramment est celle Euclidienne (ou 2-distance).

Etant donné dans \mathbb{R}^n deux vecteurs $X = (x_1, x_2, \dots, x_n)$ et $Y = (y_1, y_2, \dots, y_n)$, la distance Euclidienne \mathbf{d}_e entre eux est par définition :

$$\mathbf{d}_e(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2)$$

II.2.3.1.2 Reconnaissance des formes par la distance Euclidienne

La reconnaissance par cette méthode s'effectuera comme suit :

Chacune des images des formes de la base d'apprentissage doit être convertie en un vecteur dont les composantes ne sont rien d'autres que ses primitives extraites. Durant la phase de classification, une image d'une telle forme inconnue sera présentée également sous forme d'un vecteur pour lequel la distance Euclidienne entre celui-ci et chacun des vecteurs modélisant les images des formes de la base d'apprentissage sera calculée. La reconnaissance sera attribuée à la classe dont l'image présentant la distance minimale entre celle-ci et l'image inconnue.

II.2.3.2 Réseaux de neurones RNs

II.2.3.2.1 Définition

Les réseaux de neurones (RNs) ou en anglais Neural Networks (NNs) ont connu un grand succès dans le domaine de la reconnaissance des formes (caractères, paroles, visages, ...) [10, 61, 76, 77, 93, 131, 135, 140, 143] grâce à la mise en œuvre de leurs algorithmes

d'apprentissage et de classification [35, 55, 86, 91]. L'apprentissage d'un tel réseau de neurones a pour but de rendre ce réseau ayant un pouvoir de bien apprendre les différentes formes présentées sous forme de vecteurs à ses entrées en vue d'être capable de fournir des sorties désirées, cela est accessible via des algorithmes trop compliqués d'apprentissage qui s'appuient sur plusieurs ré-estimations itératives de ses paramètres s'appelés les poids de connexions (qui représentent la jointure entre l'entrée et la sortie du réseau) afin d'aboutir en fin de compte aux sorties désirées. Dans le contexte de ces algorithmes, on cite ceux de la correction d'erreurs de Widrow-hoff et de retro propagation du gradient, etc. (Voir Figure II.2).

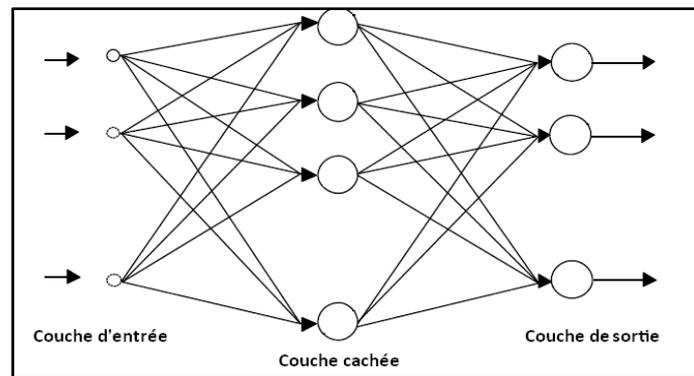


Figure II.2 : Exemple du perceptron multicouche à une couche cachée (l'un des modèles des réseaux de neurones)

II.2.3.2.2 Reconnaissance des formes par le RNs

Les démarches de cette reconnaissance pourront s'interpréter comme suit :

Dès que le réseau est mis en processus d'apprentissage, autrement dit les poids de connexions (qui ne sont que des matrices ré-estimées ou optimales) calculées juste après l'obtention des sorties désirées ou voulues, doivent être sauvegardés pour former ce qu'on appelle une base d'apprentissage, notons que chacune de ces sorties représente en fait une forme parmi celles de la base d'apprentissage.

Durant la phase de la reconnaissance, on présentera à l'entrée du réseau de neurones une image d'une forme inconnue sous forme d'un vecteur contenant ses primitives, puis on calculera le vecteur de sortie fournie par ce réseau via le calcul des produits matriciels entre ce vecteur inconnu et les coefficients de poids optimaux. Puis on se tiendra de calculer les différentes distances Euclidiennes entre ce vecteur de sortie et chacun des vecteurs de sorties

désirées. La forme qu'ayant la distance minimale entre la sortie désirée qui représente cette forme et le vecteur de sortie inconnue est effectivement la forme reconnue.

II.2.3.3 Supports vecteurs machines SVM

II.2.3.3.1 Définition

Les séparateurs à vaste marge en anglais Support Vector Machine (SVM) est une méthode statistique d'apprentissage-classification [2, 44, 66, 94, 101, 138]. Ils sont introduits par le mathématicien russe Vladimir Vapnik [188]. Les SVMs sont un ensemble d'algorithmes d'apprentissage qui permettent de bien discriminer entre les vecteurs. L'idée principale peut être expliquée tout brièvement que si deux classes qui ne sont pas en principe linéairement séparables dans un espace ayant une dimension quelconque peuvent les être séparables dans un autre espace de dimension plus grande que celle du premier (voir Figure II.3).

La séparation s'effectuera par la maximisation autant que possible de la marge entre un ensemble des vecteurs spéciaux d'une classe et un autre ensemble des autres vecteurs spéciaux aussi mais de l'autre classe. Ces vecteurs sont nommés les supports vecteurs, ce qui permettra par la suite de déterminer une frontière de séparation représentée par une fonction de décision séparant ces deux classes de part et d'autre d'une façon optimale.

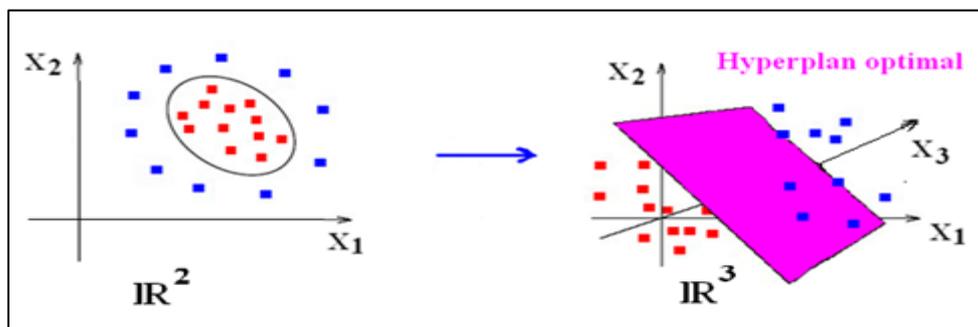


Figure II.3: Linéarisation d'une séparation non linéairement séparable

La généralisation des SVMs dans des séparations multi-classes pourra être faisable par deux stratégies principales pour lesquelles la première s'appelle un contre tous tandis que la deuxième se note un contre un.

II.2.3.3.2 Reconnaissance des formes par les SVMs

La première approche nommée un-contre-tous consiste à utiliser pour N formes données N classificateurs SVMs permettront de faire discriminer d'une manière optimale entre une

forme qui est déjà convertie en un vecteur et qui est existante dans une classe contre les autres formes présentées également dans une autre classe opposée et ainsi de suite pour chaque forme on fera le même processus. On obtiendra donc en fin de compte N fonctions de décision f_1, f_2, \dots, f_N qui doivent être stockées dans une base d'apprentissage.

La règle de reconnaissance utilisée dans ce cas est le maximum, ou l'on affectera une forme inconnue X à la classe de la forme dont la fonction de décision séparant celle-ci au reste des autres formes qui a donné la plus grande valeur de l'image de vecteur de la forme inconnue X après avoir calculé toutes les images du vecteur de la forme inconnue par toutes les N fonctions de décision:

$$\text{Classe (X)} = \arg \max_{i=1,2,\dots,N} (f_i (X)) \quad (3)$$

L'autre approche notée un-contre-un consiste à mettre en œuvre $\frac{N(N-1)}{2}$ classificateurs ou fonctions de décision pour chaque paire de classes possibles dont chacune d'elles présentant une seule forme. A ce moment-là, on attribuera à une forme inconnue la classe majoritaire parmi les $\frac{N(N-1)}{2}$ fonctions de décision.

II.2.3.4 Méthode des K-plus proches voisins K-ppv

II.2.3.4.1 Définition

La méthode des K-plus proches voisins (K-ppv) est une méthode de raisonnement à partir des cas. Elle fait partie de l'idée de prendre des décisions en recherchant un ou des cas similaires déjà résolus en mémoire [122, 124, 163, 171, 194, 196].

Il s'agit d'une des méthodes les plus simples dans le domaine d'apprentissage automatique supervisé. Elle est basée sur un algorithme appartenant à une famille des algorithmes dits paresseux.

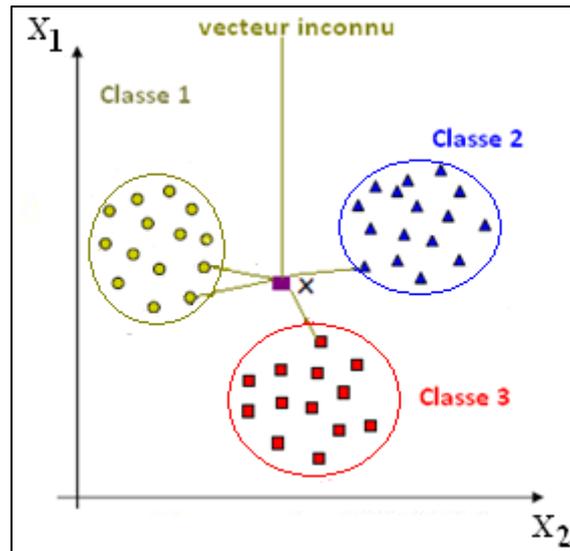


Figure II.4: 4-ppv appartenant aux trois classes différentes

La mise en évidence du classificateur K-ppv peut être présentée comme suit :

Étant donnée une base d'apprentissage des données (vecteurs), pour prédire la classe d'une nouvelle donnée inconnue, le classificateur K-ppv cherchera les K plus proches voisins de cette nouvelle donnée afin de prédire la réponse la plus fréquente de ces K-plus proches voisins. La méthode utilise dans ce contexte-là, deux paramètres qui sont un nombre K et une fonction de similarité (une certaine distance) pour pouvoir comparer la nouvelle donnée à celles déjà classées.

II.2.3.4.2 Reconnaissance des formes par les K-ppv

Une forme inconnue est présentée sous forme d'un vecteur dont les composantes sont les primitives de la forme inconnue, puis la distance Euclidienne entre ce vecteur et chacun des autres de la base d'apprentissage sera calculée, ensuite un ensemble de K-plus proches voisins du vecteur inconnu doit être déterminé. Finalement la reconnaissance sera assignée à la classe de la forme contenant le plus grand nombre des plus proches voisins parmi tous les K-ppv.

II.2.3.5 Modèle de Markov caché MMC

II.2.3.5.1 Définition

Le modèle de Markov caché (MMC) ou en anglais Hidden Markov Model (HMM), est un modèle stochastique qui est largement utilisé en particulier dans la reconnaissance des formes, l'intelligence artificielle et le traitement de signal [11, 17, 33, 36, 45, 57, 70, 168]. Un modèle

de Markov caché unidimensionnel, est une chaîne de Markov stationnaire où l'observation est une fonction probabiliste de l'état, qui est caractérisée par un système à état doublement stochastique constituant deux processus. Le premier est un processus de changement d'état appelé processus caché ou interne et qui n'est pas observable. Cependant, il peut être observable par l'intermédiaire d'un second processus d'émission d'un ensemble d'observations appelé processus externe (voir Figure II.5).

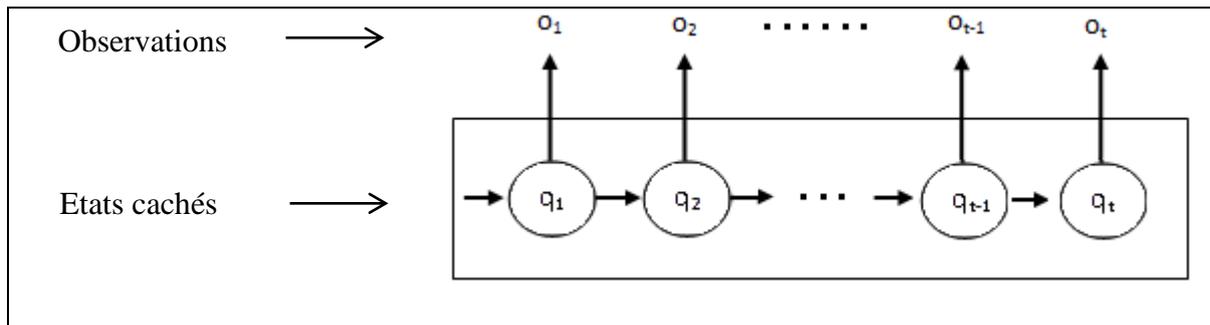


Figure II.5 : Les états cachés et observables du modèle MMC

Le comportement d'un modèle de Markov caché est lié donc aux deux suites de variables aléatoires qui sont associées respectivement à la composante observable et à celle cachée du modèle stochastique.

- La suite cachée correspond à la suite des états $q_1, q_2, q_3 \dots, q_T$ représentée par Q (1:T) avec les $q_i \in \{S_1, S_2, S_3 \dots, S_N\}$ l'ensemble de n états de modèles.
- La suite observable correspondant à la séquence d'observations $o_1, o_2, o_3 \dots, o_T$, notée par O (1:T) qui peut être continue ou discrète selon l'application donnée.

Un modèle de Markov caché est souvent noté par un ensemble des paramètres $\lambda = (\Pi, A, B)$

Où :

$$A = \{a_{i,j}\}_{1 \leq i,j \leq N} \quad (4)$$

$$a_{i,j} = P(q_t = S_j | q_{t-1} = S_i) \quad (5)$$

$$\sum_{j=1}^N a_{i,j} = 1 \quad (6)$$

On désigne par O_t le symbole observé à l'instant t et B est la matrice des probabilités d'émission des observations dans les états.

$$\Pi = \{\pi_{i,j}\}_{1 \leq i,j \leq N} \quad (7)$$

$$\pi_i = P(q_1 = S_i) \quad (8)$$

$$\sum_{i=1}^N \pi_i = 1 \quad (9)$$

Π est le vecteur des probabilités initiales des états.

II.2.3.5.2 Reconnaissance des formes par le MMC

Chaque image d'une forme qui est convertie en un vecteur pour lequel les composantes ne sont rien d'autres que les primitives de la forme, alors que ce vecteur sera utilisé comme une observation d'un MMC propre à chacune des formes $\lambda = (\Pi, A, B)$ dont les paramètres doivent être ré-estimés en vertu de l'algorithme de Baum-Welch en vue de maximiser autant que possible la probabilité générée par l'observation sachant le modèle λ , puis tous les modèles optimaux seront sauvegardés juste pour former une base d'apprentissage.

Afin de reconnaître une forme inconnue convertie auparavant en un vecteur représentant les observations qui ne sont rien d'autres que les primitives de cette forme, il fallait faire appel chacun des modèles optimaux pour calculer la probabilité qui est engendrée par l'observation inconnue à l'aide de l'algorithme de Forward ou celui de Viterbi. La reconnaissance sera à ce moment-là assignée à la classe de la forme dont le modèle optimal est celui qui a donné la plus grande probabilité.

II.3 Conclusion

La reconnaissance des caractères est en fait un domaine très riche et très dynamique en raison des recherches et des nouveautés réalisées jour après jour dans ce domaine.

Elle a pour objectif d'identifier ou bien de reconnaître des images acquises par exemple d'une numérisation via un scanner. Ce processus pourra être réalisé si les conditions suivantes sont satisfaites : le prétraitement servant à améliorer la qualité des images, puis l'extraction des primitives qui permet une différenciation inter-images et finalement l'apprentissage-classification ayant pour but de ramener le système de reconnaissance trop capable d'apprendre des images dites d'apprentissage en vue de mieux reconnaître ou classifier celles dites de test.

Notons que les performances d'un tel système de reconnaissance sont forcément pertinentes aux méthodes utilisées que ce soient pour prétraiter ou pour extraire les primitives, ou afin d'apprendre et de classifier les formes.

CHAPITRE III

PRETRAITEMENT ET EXTRACTION DES PRIMITIVES

D'abord, avant de procéder à faire reconnaître une image d'une telle forme représentée habituellement sous forme d'une matrice, il fallait d'abord obligatoirement la prétraiter en vertu de plusieurs techniques exploitées dans ce sens qui servent en principe de positiver la qualité de l'image en terme de netteté et de clarté, puis extraire autant d'informations significatives caractérisant cette image à l'aide de nombreuses méthodes usées dans ce contexte. A ce moment-là, et après avoir achevé ces deux opérations la reconnaissance de l'image sera d'une façon ou d'une autre facile.

On s'est rendu compte alors qu'avant de mettre en place un tel système de reconnaissance des formes, une phase de prétraitement puis une autre d'extraction des primitives sont forcément nécessaires.

Contenu du chapitre

<i>III.1</i>	<i>Acquisition de l'image</i>	<i>48</i>
<i>III.1.1</i>	<i>Le mode statique</i>	<i>49</i>
<i>III.1.2</i>	<i>Le mode dynamique</i>	<i>49</i>
<i>III.2</i>	<i>Prétraitement des images</i>	<i>50</i>
<i>III.2.1</i>	<i>Numérisation</i>	<i>51</i>
<i>III.2.2</i>	<i>Seuillage</i>	<i>51</i>
<i>III.2.3</i>	<i>Normalisation</i>	<i>51</i>
<i>III.2.4</i>	<i>Amincissement</i>	<i>52</i>
<i>III.2.5</i>	<i>Rotation</i>	<i>52</i>
<i>III.2.6</i>	<i>Centrage</i>	<i>53</i>
<i>III.2.7</i>	<i>La segmentation</i>	<i>53</i>
<i>III.2.7.1</i>	<i>La projection horizontale</i>	<i>53</i>
<i>III.2.7.2</i>	<i>La projection verticale</i>	<i>54</i>
<i>III.2.7.3</i>	<i>Le chevauchement des caractères</i>	<i>55</i>
<i>III.3</i>	<i>Extraction des primitives</i>	<i>55</i>
<i>III.3.1</i>	<i>Primitives statistiques</i>	<i>56</i>
<i>III.3.2</i>	<i>Primitives structurelles</i>	<i>56</i>
<i>III.3.2.1</i>	<i>Méthodes d'extraction des primitives simples (non hybrides)</i>	<i>56</i>

<i>III.3.2.1.1 La Morphologie mathématique</i>	56
<i>III.3.2.1.1.1 Dilatation par la Morphologie mathématique</i>	57
<i>III.3.2.1.1.2 Dilatation par un algorithme</i>	57
<i>III.3.2.1.1.3 Détection des zones caractéristiques de l'image</i>	58
<i>III.3.2.1.1.3.1 Extraction de la zone caractéristique d'Est</i>	58
<i>III.3.2.1.1.3.2 Extraction de la zone caractéristique d'Ouest</i>	59
<i>III.3.2.1.1.3.3 Extraction de la zone caractéristique du Sud</i>	60
<i>III.3.2.1.1.3.4 Extraction de la zone caractéristique du Nord</i>	60
<i>III.3.2.1.1.3.5 Extraction de la zone caractéristique Centrale</i>	61
<i>III.3.2.1.1.3.6 Vecteur d'extraction</i>	61
<i>III.3.2.1.2 Le Zonage</i>	62
<i>III.3.2.1.2.1 Extraction par le Zonage</i>	62
<i>III.3.2.1.2.2 Vecteur d'extraction</i>	63
<i>III.3.2.1.3 Le Zig-Zag</i>	63
<i>III.3.2.1.3.1 Extraction par le Zig-Zag</i>	63
<i>III.3.2.1.3.2 Vecteur d'extraction</i>	63
<i>III.3.2.2 Méthodes contribuées (hybrides)</i>	64
<i>III.3.2.2.1 Extraction par Morphologie – Zonage</i>	64
<i>III.3.2.2.2 Extraction par Morphologie – Zig Zag</i>	64
<i>III.3.2.2.3 Extraction par Morphologie – Zig Zag – Zonage</i>	65
<i>III.4 Conclusion</i>	66

III.1 Acquisition de l'image

Elle se considère comme la phase préliminaire d'un certain système de reconnaissance de formes, effectivement en fait recours à cette technique en vue de faire digitaliser les images afin de pouvoir les rendre lisibles par un ordinateur. Cette tâche est plus ou moins difficile en raison de la diversité des formats et de la qualité de présentation des formes dans les images, dans ce contexte-là, on se restreint parmi ces formes aux écritures des caractères.

En pratique, cette technique fait partie de deux modes principaux de saisie qui se sont le mode statique pour les caractères déjà écrits et le mode dynamique pour les caractères manuscrits à reconnaître tout exactement en cours de son écriture.

III.1.1 Le mode statique

Le mode statique nommé aussi la reconnaissance hors ligne utilise essentiellement des scanners afin de convertir l'écriture en image numérique. Plusieurs types de scanners existent dans le marché fournissant le choix des images soient en noir et blanc, soient ayant un niveau de gris, soient des images en couleurs.

➤ Exemple d'acquisition en mode statique

Avant de pouvoir acquérir une telle image contenant une écriture quelconque, celle-ci doit se présenter dans un papier blanc à un scanner qui permet de le transformer en une image binaire de format JPEG (voir Figures III.6, 7).

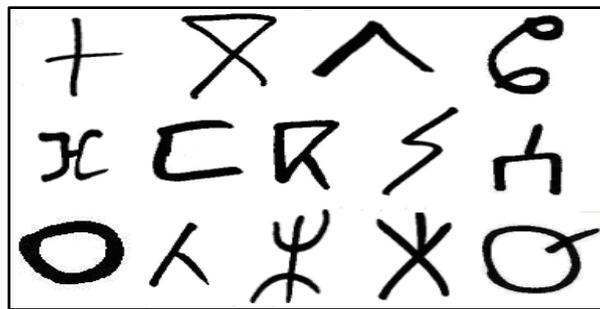


Figure III.6 : Exemple des caractères Tifinaghs acquis par un scanner et qui sont écrits par un marqueur

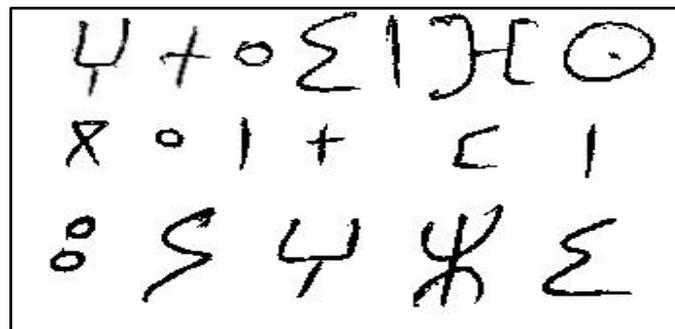


Figure III.7 : Exemple des caractères Tifinaghs acquis par un scanner et qui sont écrits par un stylo

III.1.2 Le mode dynamique

Le mode dynamique ou la reconnaissance en ligne des caractères est un mode de reconnaissance qui se déroule durant l'écriture c.-à-d. celle-ci est saisie à l'aide d'un stylo digital sur une tablette ou un écran. Juste après sa reconnaissance sera réalisée.

➤ *Exemple d'acquisition en mode dynamique*

Inversement au mode statique, cette fois-ci il s'agit d'une saisie d'écriture qui doit se reconnaître juste avant sa saisie. Plus précisément, on parlera dans ce cas d'une reconnaissance des caractères cursifs (voir Figure III.8).



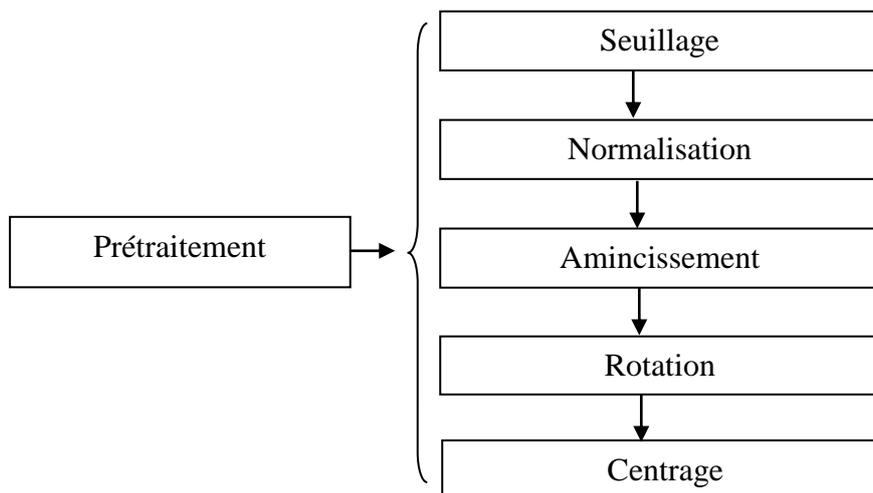
Figure III.8 : Exemple des caractères Tifinaghs cursifs

III.2 Prétraitement des images

Du fait que les images à reconnaître sont en principe bien évidemment bruitées, et qu'elles contiennent également plusieurs informations parasites ainsi redondantes, et qu'elles ne représentent pas assez de caractéristiques dites aussi primitives utiles facilitant sa reconnaissance. De plus qu'elles englobent des formes souvent présentent des granularités et des manquements ou des empâtements. Donc une étape de prétraitement sera forcément primordiale en vue d'éliminer ou plus précisément réduire que possible les différents types de bruit dû aux plusieurs causes à titre d'exemples parmi elles : l'encre du verso traversant le papier et dont la trace est visible sur la feuille du manuscrit, la mauvaise qualité du papier utilisée dans l'écriture du caractère et la sorte du stylo utilisé, etc.

Outre à la réduction du bruit le prétraitement permet également de pouvoir réaliser plusieurs opérations telles que la binarisation, le redressement, le amincissement et la normalisation ce qui nous permettra par conséquent d'effectuer une bonne extraction des primitives que nous la expliquerons ultérieurement. En bref toutes les techniques de prétraitement servent à améliorer la qualité des images à reconnaître.

Donc la phase de prétraitement peut être schématisée comme suit :



De ce fait, on expliquera ces différentes techniques de prétraitement :

III.2.1 Numérisation

La numérisation a pour rôle est de transformer l'image utilisée dans le traitement sous une forme matricielle, autrement dit pouvoir obtenir des données numériques manipulables aisément par l'ordinateur.

III.2.2 Seuillage

Dés que l'acquisition de l'image d'un caractère manuscrit est achevée, on fera rendre celle-ci en format binaire en vertu de la technique de seuillage qui consiste à ramener cette image ne contient que des valeurs 0 (couleur noire) et 1 (couleur blanche) selon un seuil préfixé.

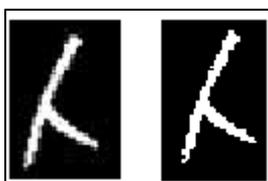


Figure III.9 : Image d'un caractère TIFINAGH avant et après le seuillage

III.2.3 Normalisation

La taille d'un caractère peut varier d'une écriture à l'autre, ce qui peut causer effectivement une instabilité des paramètres. Pour surmonter cet inconvénient on procédera à effectuer une technique de normalisation qui sert à égaliser la taille des caractères (voir Figure III.11).



Figure III.10 : Images d'un caractère Tifinagh avant la normalisation



Figure III.11 : Images d'un caractère Tifinagh après la normalisation

III.2.4 Amincissement

Les caractères acquis ayant en principe des épaisseurs variables, ce qui peut produire des instabilités dans les formes de ces caractères. Il est donc nécessaire de rendre chaque caractère ayant une épaisseur égale à un pixel avant d'extraire ses primitives. Cette technique constitue l'étape d'amincissement du caractère.

A savoir que les deux critères retenus pour les méthodes d'amincissement sont les suivants :

- L'épaisseur du caractère aminci doit être avec exactitude un pixel rien plus que ça.
- Le caractère aminci doit conserver les propriétés topologiques de sa forme comme le nombre de parties, le nombre de trous et la connexité (voir Figure III.12).



Figure III.12 : Image avant et après la technique d'amincissement

III.2.5 Rotation

Il arrive souvent que les caractères sont mal positionnés dans ses images, en terme d'être inclinés par des angles plus ou moins grandes vis-à-vis à l'axe horizontal ainsi qu'à celui vertical ce qui présentera une vraie problématique dans le processus de reconnaissance, néanmoins, en vue de surpasser cet obstacle on fera recours d'une opération de rotation de l'image de telle sorte que les angles d'inclinaisons convergent vers zéro.

En revanche, la détection d'inclinaison des caractères se fait à l'aide de la transformée de Hough [1, 152].



Figure III.13 : L'image d'un caractère Tifinagh avant et après avoir la rotée

III.2.6 Centrage

En principe les caractères à reconnaître sont mal localisés dans ses images, pour obtenir de résultats satisfaisants concernant sa reconnaissance il fallait bien que ses positions soient juste au centre de ses images, ceci est accessible en vertu de la technique du centrage (voir Figure III.15).



Figure III.14 : L'image du caractère Tifinagh dans des différentes positions



Figure III.15: Image du caractère Tifinagh après le centrage

III.2.7 Segmentation

Il s'agit cette fois-ci d'une technique de prétraitement [82, 174] qui ne s'utilise que dans la reconnaissance des textes et qui sert à décomposer l'image d'un texte en plusieurs entités qui ne sont que des mots, des caractères ou des graphèmes et qui font partie d'un alphabet prédéfini selon le but visé. Par ailleurs, la segmentation permet de réduire la complexité des modules de traitements utilisés par la suite. En effet cette technique constitue une étape fondamentale dans la lecture automatique des textes manuscrits cursifs. Pratiquement, elle se commencera d'abord par faire segmenter un tel texte donné en lignes, puis en mots et finalement en caractères.

La segmentation donc permet d'isoler dans une image les lignes d'un texte les unes aux autres, de même pour les caractères formant chaque ligne. Ensuite la segmentation s'achèvera si les caractères sont projetés en premier temps horizontalement puis verticalement.

III.2.7.1 La projection horizontale

La projection horizontale consiste à compter le nombre de pixels blancs dans chaque ligne, puis on tracera le nombre de pixels trouvés dans chaque ligne en fonction du nombre de lignes (voir Figure III.17). Enfin on doit isoler les lignes de texte les unes aux autres.

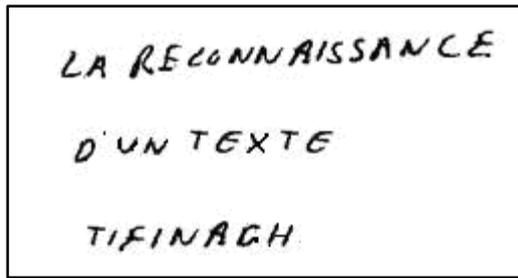


Figure III.16 : Texte français avant une segmentation horizontale

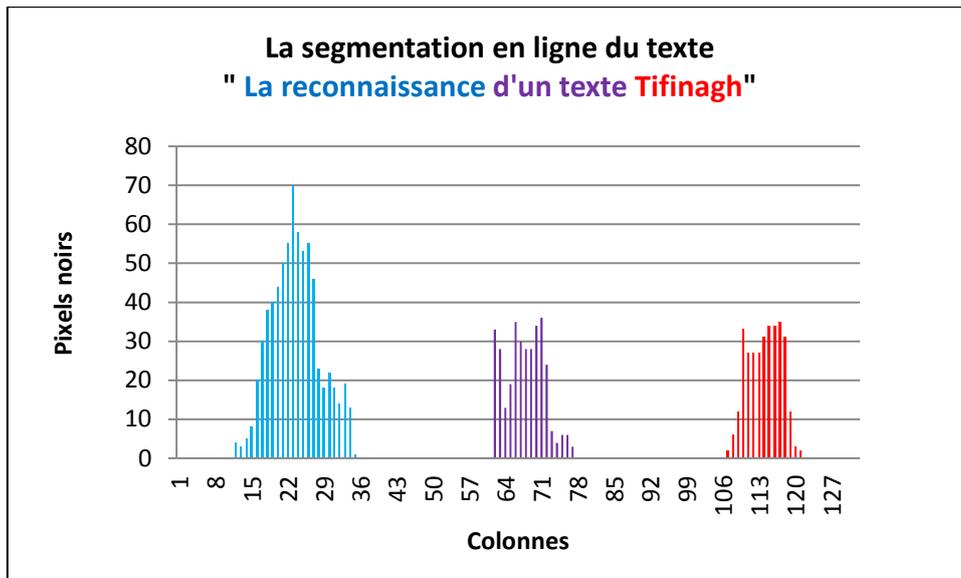


Figure III.17 : Calcul de l’histogramme d’une projection horizontale.

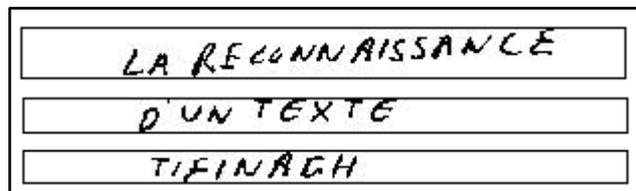


Figure III.18 : Texte français après une segmentation horizontale

III.2.7.2 La projection verticale

La projection verticale consiste à compter le nombre de pixels blancs dans chaque colonne, puis on tracera le nombre de pixels trouvés dans chaque colonne en fonction du nombre de colonnes (voir Figure III.20). Juste après l’isolation des lignes avec la projection horizontale, on pourra séparer des caractères dans chaque colonne.

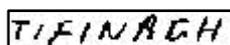


Figure III.19 : Texte avant une segmentation verticale

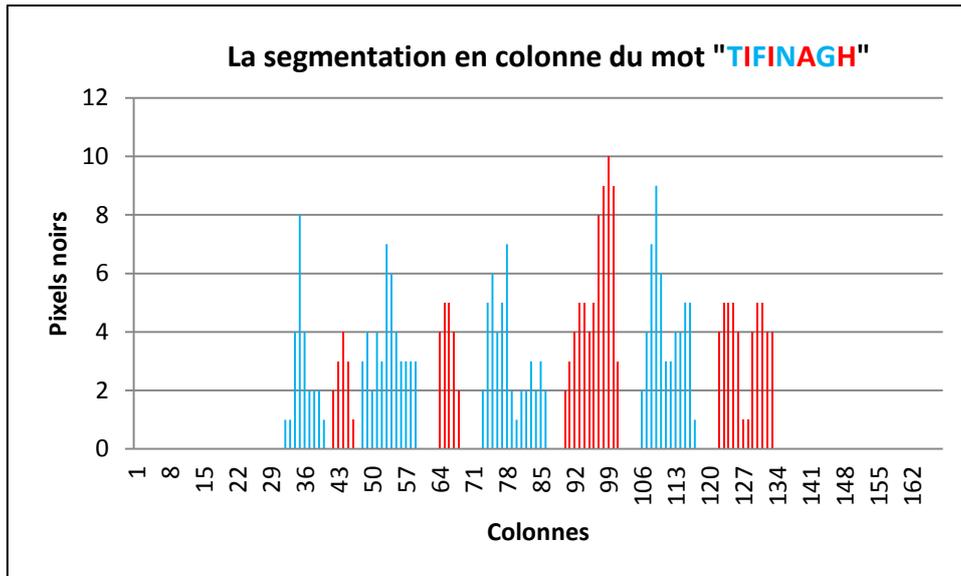


Figure III.20 : Calcul de l’histogramme d’une projection verticale.



Figure III.21 : Texte après la segmentation verticale

III.2.7.3 Le Chevauchement des caractères

Le chevauchement de deux ou de plusieurs caractères constitue bien un grand problème dans la reconnaissance des ceux-ci du fait qu’il rend une telle segmentation entre eux et qui précède sa reconnaissance trop difficile, ce problème est en principe dû à une écriture trop inclinée, ou à cause d’un alignement entre les lignes du texte écrit.

En effet, les caractères Tifinaghs ne présentent pas tellement de chevauchements du fait qu’ils sont écrits d’une manière séparée au sein d’un texte Tifinagh, de plus pour qu’on puisse avoir une garantie concernant le surpassement de cet inconvénient, on a opté de préfixer d’abord un seuil avant de faire segmenter les textes et les caractères ce qui permettra par conséquent d’éviter totalement le problème de chevauchement.

III.3 Extraction des primitives

Cette phase est très importante dans un certain système de reconnaissance, arguant que s’elle est mal conçue on aura certainement des résultats insatisfaisants fournis par le processus de reconnaissance, en fait, cette phase nous permettra de pouvoir convertir les images des formes présentées sous formes matricielles en des vecteurs dont les composantes ne sont rien d’autres

que des informations propres caractérisant efficacement chacune de ces formes, elle est évident que tant ce nombre de composantes s'augmentera tant que la quantité des informations extraites s'élèvera également ce qui facilitera par conséquent d'aboutir à une discrimination trop importante entre les différentes formes.

En revanche, l'extraction des primitives [52, 53, 184, 185, 198] se divise en deux catégories principales qui se sont les primitives statistiques et celles structurelles.

III.3.1 Primitives statistiques

Les primitives statistiques ont pour objectif de représenter les informations distribuées dans toute l'image. Elles sont calculées généralement par l'estimation de la densité de pixels appartenant à l'image à reconnaître soit complètement ou soit partiellement, tout en utilisant des mesures parfaitement statistiques à titre d'exemples:

- Les moments de : Hu, Zernike, Krawtchouk
- Les transformées de : Fourier-Mellin, Radon

III.3.2 Primitives structurelles

Il s'agit cette fois-ci des primitives qui décrivent les propriétés caractérisant la forme que ce soient topologiques ou géométriques. Ces primitives sont extraites à partir de la représentation de la forme comme le squelette ou le contour, etc. En effet les primitives structurelles possédant une grande capacité discriminative, ce qui va bien évidemment influencer sur la rapidité de la décision d'appartenance à une certaine classe durant la phase de classification.

III.3.2.1 Méthodes d'extraction des primitives simples (non hybrides)

III.3.2.1.1 La Morphologie mathématique

La Morphologie mathématique [18, 69, 70, 98, 177, 181] est une technique d'analyse de structures. Elle est fondée en principe en vue de résoudre quelques problèmes rencontrés au niveau du traitement d'images. Parmi les idées de base de la Morphologie mathématique, il y avait celle s'appuyant sur la détermination d'abord de ce que s'appelle l'élément structurant qui n'est qu'une partie de l'image ou par autre expression un ensemble de pixels rectiligne ou circulaire ou bien d'autres avant de pouvoir effectuer une telle Morphologie mathématique.

Dans notre approche on se restreint sur la dilatation arguant que celle-ci est la seule opération morphologique qui permettra de pouvoir nous fournir des primitives comme il le faut. De ce fait on a dilaté les images des formes par les deux techniques suivantes :

III.3.2.1.1.1 Dilatation par la Morphologie mathématique

La dilatation est une transformation basée sur l'intersection entre l'objet de l'image A à l'aide d'un élément structurant noté B. Elle est définie par la formule suivante :

$$\text{Dilatation (A, B)} = \{x \in A / B_x \cap A \neq \emptyset\} \quad (10)$$

Où A est l'objet de l'image (les pixels blancs), B est l'élément structurant qui est un ensemble particulier ayant un centre x, une géométrie et une taille connue (dans ce travail c'est une demi-droite).

- Exemple de la dilatation du caractère suivant vers l'Est.



Figure III.22 : Dilatation d'un caractère Tifinagh vers L'EST

Et on fait la même chose pour les autres directions Ouest, Nord et Sud.



Figure III.23 : Dilatations d'un caractère Tifinagh vers L'OUEST, le NORD et le SUD

III.3.2.1.1.2 Dilatation par un algorithme

L'algorithme utilisé pour la dilatation est appuyé sur le changement des valeurs des pixels (noirs ou blancs) de l'image traitée.

➤ **Algorithme pour la dilatation vers l'Est**

```
Répéter
  Pour chaque ligne / colonne faire
    1) Si Image (Ligne, Colonne) = 1
    2) {
    3) Image (Ligne, Colonne + 1) = 1
    4) Incrémentation
    5) }
    6) La Dilatation de l'image vers L'East.
  Fin Si
Fin
Jusqu'à « aucune modification n'est possible sur l'image »
```

Voir annexe B pour les autres algorithmes de dilatation vers l'**Ouest**, le **Nord** et le **Sud**.

III.3.2.1.1.3 Détection des zones caractéristiques de l'image

On détermine pour chaque caractère les paramètres discriminants qui ne sont que des zones caractéristiques qui peuvent être détectables en calculant les intersections entre les images dilatées vers les quatre directions habituelles qui se sont l'Est, l'Ouest, le Nord et le Sud. De cette façon on pourra réellement définir cinq types de zones caractéristiques : Est, Ouest, Nord, Sud, et la zone centrale dont chacune d'entre elles s'obtiendra de la manière suivante :

III.3.2.1.1.3.1 Extraction de la zone caractéristique d'Est :

Un point de l'image (voir Figure III.24-a) appartient à la zone caractéristique Est (voir Figure III.24-b) si et seulement si :

- Ce point n'appartient pas à l'objet du caractère (les pixels blancs).
- A partir de ce point, en se déplaçant en ligne droite vers l'Est, on ne croise pas l'objet (les pixels blancs).

- A partir de ce point, en se déplaçant en ligne droite vers le Sud, le Nord, l'Ouest on croise l'objet (voir Figure III.24-a). Le résultat de l'extraction est illustré dans la Figure III.24-b.

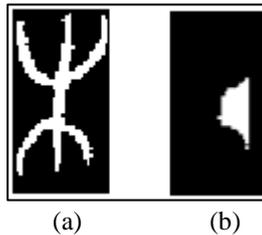


Figure III.24 : (a) : Image du caractère Tifinagh « yaz »
(b) : La zone caractéristique d'Est (**ZE**)

Avec,

ZE : l'image de la zone caractéristique EST sera donnée par :

$$\mathbf{ZE} = (\mathbf{ImageOuest}) \text{ and } (\mathbf{ImageNord}) \text{ and } (\mathbf{ImageSud}) \text{ and } (\mathbf{I}) \quad (11)$$

And : ET logique (pixel par pixel).

ImageOuest : la dilatation de l'image traitée vers l'Ouest.

ImageNord : la dilatation de l'image traitée vers le Nord.

ImageSud : la dilatation de l'image traitée vers le Sud.

I : l'inversion de l'image initiale.

De la même manière on trouve les autres zones caractéristiques (Voir Annexe C).

III.3.2.1.1.3.2 Extraction de la zone caractéristique d'Ouest

Un point de l'image (voir Figure III.25-a) appartient à la zone caractéristique Ouest si et seulement si :

- Ce point n'appartient pas à l'objet du caractère (les pixels blancs).
- A partir de ce point, en se déplaçant en ligne droite vers l'Ouest, on ne croise pas l'objet.
- A partir de ce point, en se déplaçant en ligne droite vers le Sud, le Nord, l'Est on croise l'objet (voir Figure III.25-a). Le résultat de l'extraction est illustré dans la Figure III.25-b.

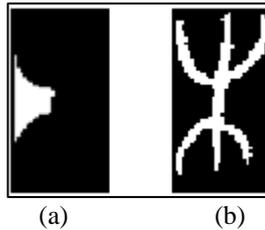


Figure III.25 : (a) : Image du caractère Tifinagh « yaz »

(b): La zone caractéristique d’Ouest (ZO)

III.3.2.1.1.3.3 Extraction de la zone caractéristique du Sud :

Un point de l’image (voir Figure III.26-a) appartient à la zone caractéristique Sud si et seulement si :

- Ce point n’appartient pas à l’objet du caractère.
- A partir de ce point, en se déplaçant en ligne droite vers le Sud, on ne croise pas l’objet.
- A partir de ce point, en se déplaçant en ligne droite vers le Nord, l’Est et l’Ouest on croise l’objet (voir Figure III.26-a). Le résultat de l’extraction est illustré dans la Figure III.26-b.

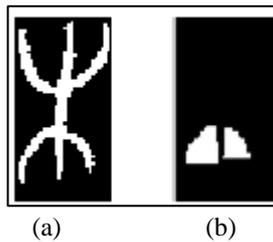


Figure III.26 : (a) : Image du caractère Tifinagh « yaz »

(b) : La zone caractéristique du Sud (ZS)

III.3.2.1.1.3.4 Extraction de la zone caractéristique du Nord :

Un point de l’image (voir Figure III.27-a) appartient à la zone caractéristique Nord si et seulement si :

- Ce point n’appartient pas à l’objet du caractère.
- A partir de ce point, en se déplaçant en ligne droite vers le Nord, on ne croise pas l’objet.
- A partir de ce point, en se déplaçant en ligne droite vers le Sud, l’Est et l’Ouest on croise l’objet (voir Figure III.27-a). Le résultat de l’extraction est illustré dans la Figure III.27-b.

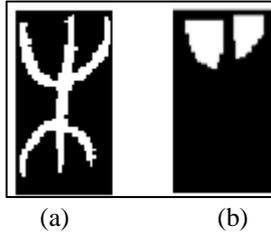


Figure III.27 : (a) : Image du caractère Tifinagh « yaz »
 (b) : La zone caractéristique du **Nord (ZN)**

III.3.2.1.1.3.5 Extraction de la zone caractéristique Centrale :

Un point de l'image (voir Figure III.28-a) appartient à une zone caractéristique centrale si:

- Ce point n'appartient à la limite de l'objet du caractère.
- A partir de ce point, en se déplaçant en ligne droite vers le Sud, le Nord, l'Est et l'Ouest on croise l'objet (voir Figure III.28-a). Le résultat de l'extraction est illustré dans la Figure III.28-b.

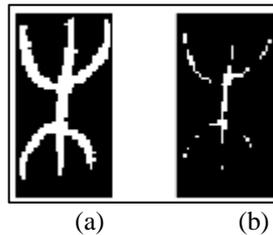


Figure III.28 : (a) : Image du caractère Tifinagh « yaz »
 (b) : La zone caractéristique **Centrale (ZC)**

III.3.2.1.1.3.6 Vecteur d'extraction

La méthode utilisée pour déterminer le vecteur d'extraction est basée sur les paramètres caractéristiques extraites à partir des surfaces des différents types de zones Est, Ouest, Centrale, Nord et Sud.

Chaque caractère est caractérisé par un vecteur d'extraction ayant cinq composantes.

$$\mathbf{V}_{\text{extraction}} = (\mathbf{X}_{ze}, \mathbf{X}_{zw}, \mathbf{X}_{zc}, \mathbf{X}_{zn}, \mathbf{X}_{zs}) \quad (12)$$

Ces composantes sont :

$$X_{ze} = \frac{N_{ze}}{N_{\text{pixels}}}$$

$$X_{zw} = \frac{N_{zw}}{N_{\text{pixels}}}$$

$$X_{zn} = \frac{N_{zn}}{N_{\text{pixels}}}$$

$$X_{zs} = \frac{N_{zs}}{N_{\text{pixels}}}$$

$$X_{zc} = \frac{N_{zc}}{N_{\text{pixels}}}$$

N_{ze} : Le nombre de pixels de niveau 1 dans l'image **ZE**.

N_{zw} : Le nombre de pixels de niveau 1 dans l'image **ZW**.

N_{zn} : Le nombre de pixels de niveau 1 dans l'image **ZN**.

N_{zs} : Le nombre de pixels de niveau 1 dans l'image **ZS**.

N_{zc} : Le nombre de pixels de niveau 1 dans l'image **ZC**.

N_{pixels} : Le nombre de pixels dans l'image traitée.

III.3.2.1.2 Le Zonage

III.3.2.1.2.1 Extraction par le Zonage

D'abord, étant donnée une image noire englobant un chiffre qui est écrit en blanc comme l'indique la figure ci-dessous, la méthode de Zonage [33, 92, 93, 96, 101, 159] consiste à découper cette image en plusieurs blocs ou zones carrées ou rectangulaires ou bien triangulaires. Puis dans chacune d'entre elles on comptera le nombre de pixels blancs ce qui permettra par la suite de convertir l'image en un vecteur ayant un nombre de composantes qui égale à celui du nombre de zones.

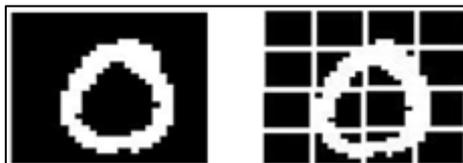


Figure III.29: – Exemples d'extraction des primitives par le Zonage d'un caractère Tifinagh « yar »

III.3.2.1.2.2 Vecteur d'extraction

- Chaque image est caractérisée par un vecteur de plusieurs composants.

$$\mathbf{V}_{\text{extraction}} = (N_{\text{pixels_blancs_zone}_1}, \dots, N_{\text{pixels_blancs_zone}_N}) \quad (13)$$

Avec :

$N_{\text{pixels_blancs_zone}_i}$: est le nombre de pixels blancs dans la zone i , $i=1,2,\dots$, ou N est le nombre total du zones pour lesquelles l'image a été découpée.

III.3.2.1.3 Le Zig-Zag

III.3.2.1.3.1 Extraction par le Zig-Zag

Cette méthode [104, 142, 151] consiste à faire balayer l'image ligne par ligne puis colonne par colonne puis le long de toute ligne parallèle au diagonal et également celle parallèle au anti-diagonal de l'image, ensuite le nombre de pixels blancs calculés dans chacune de ces lignes et de ces colonnes représentera en fait une composante d'un vecteur dont l'image s'est transformée (voir Figure III.30).

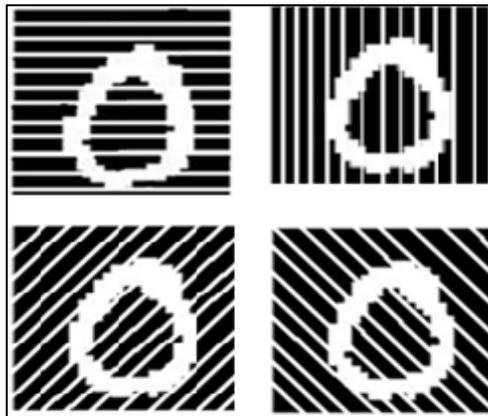


Figure III.30: Exemples d'extraction des primitives par Zig-Zag d'un caractère Tifinagh « yar »

III.3.2.1.3.2 Vecteur d'extraction

Le résultat issu de l'opération de Zig-Zag est que l'image s'est convertie en un vecteur présenté comme suit :

$$\mathbf{V}_{\text{Ligne}} = (N_{\text{pixelsLigne1}}, N_{\text{pixelsLigne2}}, N_{\text{pixelsLigne3}}, \dots, N_{\text{pixelsLigneN}}) \quad (14)$$

$$\mathbf{V}_{\text{Colonne}} = (N_{\text{pixelsColonne1}}, N_{\text{pixelsColonne2}}, N_{\text{pixelsColonne3}}, \dots, N_{\text{pixelsColonneN}}) \quad (15)$$

$$V_{\text{Diagonale}} = (N_{\text{pixelsDiagonale1}}, N_{\text{pixelsDiagonale2}}, N_{\text{pixelsDiagonale3}}, \dots, N_{\text{pixelsDiagonaleN}}) \quad (16)$$

$$V_{\text{AntiDiagonale}} = (N_{\text{pixelsAntiDiagonale1}}, N_{\text{pixelsAntiDiagonale2}}, \dots, N_{\text{pixelsAntiDiagonaleN}}) \quad (17)$$

$$V_{\text{extraction}} = (V_{\text{Ligne}}, V_{\text{Colonne}}, V_{\text{Diagonale}}, V_{\text{AntiDiagonale}}) \quad (18)$$

III.3.2.2 Méthodes contribuées (hybrides)

En fait, les méthodes présentées dans ce propos-là sont les méthodes qu'on a pu réaliser comme des contributions dans cette thèse élaborées en vue de bien améliorer les systèmes de reconnaissance qu'on a mis en action en termes de fiabilité et d'efficacité, ainsi que par ailleurs de mettre notre empreinte sur ce domaine de reconnaissance des caractères au sens de tenter à créer des méthodes non classiques.

III.3.2.2.1 Extraction par morphologie - zonage

Cette méthode consiste d'abord à faire dilater l'image (voir annexe B), puis effectuer un Zonage au sein de celle-ci. En effet la figure ci-dessous illustre le processus de cette hybridation.



Figure III.31: Exemple d'extraction des primitives par la dilatation et le Zonage d'un caractère Tifinagh « yar »

III.3.2.2.2 Extraction par Morphologie – Zig-Zag

Pour l'extraction des primitives à partir du nombre de pixels avec la dilatation, on cherchera à compter le nombre de pixels blancs dans chaque ligne, chaque colonne, chaque diagonale et chaque anti-diagonale mais après avoir dilaté l'image traitée (voir Figure III.32).

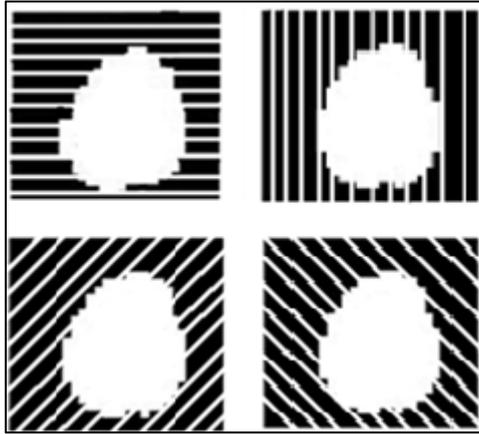


Figure III.32: Exemples d'extraction des primitives par la dilatation et le Zig-Zag d'un caractère Tifinagh « yar »

III.3.2.2.3 Extraction par Morphologie – Zig Zag – Zonage

Cette opération consiste qu'après avoir dilaté l'image à faire zoner celle-ci mais ce n'est pas comme ce qu'on a effectué précédemment, cependant il s'agit cette fois-ci de zoner l'image d'une manière zigzagüe, autrement dit les zones dans lesquelles l'image est découpée ne sont que des rectangles horizontaux ainsi verticaux et des trapézoïdes parallèles au diagonale et aussi au anti diagonale de l'image. Plus tard, on comptera le nombre de pixels blancs dans chacune de ces zones puis regrouper tous ces nombres dans un vecteur (voir Figure III.33)

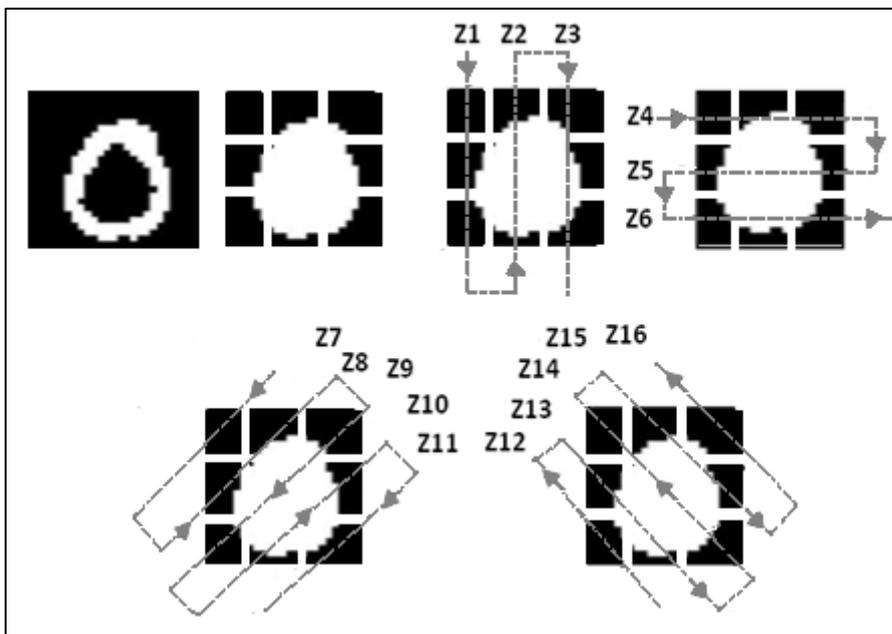


Figure III.33: Exemples d'extraction des primitives par l'hybridation des trois méthodes du caractère « yar »

III.4 Conclusion

Dans la reconnaissance des formes, le prétraitement reste une étape à la fois préliminaire et primordiale en termes de son rôle dans la réduction de tous les types du bruit présenté dans les images des formes, ainsi que dans l'élimination des informations redondantes et parasites, et également dans la correction des différents défauts pertinents à la forme que celle-ci soit mal centrée, plus ou moins déformée, etc.

Pour ce faire, En vue de prétraiter une telle image englobant une forme quelconque, une multitude de techniques font intervenir au sein de cette tendance, quelques-unes parmi elles par exemple : la binarisation, la normalisation, l'amincissement et le centrage, etc.

Cependant, même si les formes sont prétraitées sa reconnaissance directe à partir de son image est presque impossible. D'où la nécessité de réaliser une extraction des primitives, à partir de la représentation des pixels de l'image prétraitée, celle-ci doit se convertir en un vecteur pour lequel les composantes sont au sens informatique des informations propres caractérisant cette forme, la finalité envisagée de cette conversion est d'avoir facilité l'apprentissage puis la classification des formes.

Afin de mettre en œuvre cette extraction, nombreuses méthodes sont disponibles dans ce propos comme le Zonage, le Zig-Zag et les opérations morphologiques y inclus la dilatation, l'érosion, l'ouverture et la fermeture, etc.

PARTIE 2 : L'APPRENTISSAGE – CLASSIFICATION, TESTS ET RESULTATS

CHAPITRE IV

L'APPRENTISSAGE - CLASSIFICATION

L'apprentissage est une technique d'identification permettant de produire automatiquement des règles à partir d'une base de donnée d'apprentissage contenant des certains échantillons font partie en principe de nombreuses classes.

En fait, l'apprentissage est souvent suivi d'une étape de classification en raison que les deux sont forcément deux processus conjugués, pratiquement, il existe plusieurs types d'apprentissage que se soient d'origine neuronale, statistique ou probabiliste, etc.

Le but de la classification sert à pouvoir attribuer des objets qui ont été déjà inter-discriminés durant la phase d'apprentissage dans leurs classes correctement.

Dans ce travail, plusieurs méthodes d'apprentissage–classification sont utilisées à savoir les réseaux de neurones, les séparateur à vaste marge, les K- plus proches voisins et les modèles de Markov cachés. Ces méthodes se divisent en deux catégories principales supervisées et non-supervisées.

Contenu du chapitre

<i>IV.1</i>	<i>Méthodes d'apprentissage – classification supervisées</i>	<i>70</i>
<i>IV.1.1</i>	<i>Les réseaux de neurones (RN's)</i>	<i>70</i>
<i>IV.1.1.1</i>	<i>Définition</i>	<i>71</i>
<i>IV.1.1.1.1</i>	<i>Le neurone biologique</i>	<i>71</i>
<i>IV.1.1.1.2</i>	<i>Le neurone formel</i>	<i>72</i>
<i>IV.1.1.1.3</i>	<i>Fonction d'activation</i>	<i>73</i>
<i>IV.1.1.2</i>	<i>Architecture d'un réseau de neurones</i>	<i>74</i>
<i>IV.1.1.2.1</i>	<i>Le perceptron monocouche</i>	<i>75</i>
<i>IV.1.1.2.2</i>	<i>Le perceptron multicouche</i>	<i>76</i>
<i>IV.1.1.3</i>	<i>Apprentissage – classification par les réseaux de neurones</i>	<i>77</i>
<i>IV.1.1.3.1</i>	<i>Définition</i>	<i>77</i>
<i>IV.1.1.3.2</i>	<i>Séparation des données</i>	<i>77</i>
<i>IV.1.1.3.2.1</i>	<i>Cas des problèmes linéairement séparables</i>	<i>77</i>
<i>IV.1.1.3.2.2</i>	<i>Cas des problèmes non linéairement séparables</i>	<i>80</i>

<i>JV.1.2</i>	<i>Les supports vecteurs machines (SVMs)</i>	<i>83</i>
<i>JV.1.2.1</i>	<i>Introduction</i>	<i>83</i>
<i>JV.1.2.2</i>	<i>Les SVMs linaires</i>	<i>84</i>
<i>JV.1.2.2.1</i>	<i>Les SVMs linaires</i>	<i>84</i>
<i>JV.1.2.2.1.1</i>	<i>Le problème primal</i>	<i>86</i>
<i>JV.1.2.2.1.2</i>	<i>Le problème dual</i>	<i>87</i>
<i>JV.1.2.3</i>	<i>Les SVMs non linaires</i>	<i>89</i>
<i>JV.1.2.4</i>	<i>Les SVMs multi-classes</i>	<i>91</i>
<i>JV.1.3</i>	<i>Les K-plus proches voisins (K-ppv)</i>	<i>91</i>
<i>JV.1.3.1</i>	<i>Principe de classification par K-ppv</i>	<i>92</i>
<i>JV.1.3.2</i>	<i>L'importance du choix de la distance</i>	<i>93</i>
<i>JV.1.3.3</i>	<i>L'importance du choix de nombre K</i>	<i>94</i>
<i>JV.1.3.4</i>	<i>Algorithme de K-ppv</i>	<i>94</i>
<i>JV.2</i>	<i>Méthodes d'apprentissage – classification non supervisées</i>	<i>95</i>
<i>JV.2.1</i>	<i>Le modèle de Markov caché (MMC)</i>	<i>95</i>
<i>JV.2.1.1</i>	<i>Processus stochastique</i>	<i>95</i>
<i>JV.2.1.2</i>	<i>Chaine de Markov</i>	<i>96</i>
<i>JV.2.1.3</i>	<i>Modèles de Markov cachés</i>	<i>97</i>
<i>JV.2.1.3.1</i>	<i>Élément d'un MMC</i>	<i>97</i>
<i>JV.2.1.3.2</i>	<i>Les topologies de MMC</i>	<i>98</i>
<i>JV.2.1.3.3</i>	<i>Types de densité d'observations</i>	<i>98</i>
<i>JV.2.1.3.4</i>	<i>Problèmes fondamentaux associés au MMC</i>	<i>99</i>
<i>JV.2.1.3.4.1</i>	<i>Problème d'évaluation</i>	<i>99</i>
<i>JV.2.1.3.4.2</i>	<i>Problème d'apprentissage ou d'entraînement</i>	<i>100</i>
<i>JV.2.1.3.4.3</i>	<i>Problème de reconnaissance ou décodage</i>	<i>100</i>
<i>JV.2.1.4</i>	<i>Classification par le MMC</i>	<i>101</i>
<i>JV.2.2</i>	<i>Système Hybride P.MC / MMC</i>	<i>102</i>
<i>JV.3</i>	<i>Conclusion</i>	<i>103</i>

IV.1 Méthodes d'apprentissage – classification supervisées

L'apprentissage supervisé est un moyen primordial servant à rendre un système de reconnaissance des formes capable de bien apprendre ses entrées en vue de fournir des sorties désirées, autrement dit il va falloir forcer le système de converger vers ces sorties tout en ré-estimant d'une manière itérative ses paramètres tels que des matrices, des vecteurs, etc. Il s'agit en fait d'un problème d'optimisation.

IV.1.1 Les réseaux de neurones

L'intelligence artificielle a pour objectif de reproduire un comportement intelligent similaire à l'être humain en vertu d'une machine automatique comme l'ordinateur, sachant que certains problèmes qui sont facilement plus ou moins résolus par les êtres humains peuvent les être également par les ordinateurs. A titre d'exemples de ces problèmes: la traduction automatique, la reconnaissance des: signatures, visages, paroles etc.

Le principe fondamental de fonctionnement des réseaux de neurones s'est inspiré des propriétés du cerveau humain afin de produire des systèmes intelligents ayant une capacité élevée concernant la résolution des problèmes qui sont résolus aisément par l'esprit humain. La différence entre les réseaux de neurones et les ordinateurs réside dans leurs architectures.

Tout à fait, un ordinateur est composé d'un processeur effectuant la totalité des tâches prévues. Tandis que le cerveau humain est formé d'un très grand nombre des cellules élémentaires appelées neurones et qui se sont interconnectées par des connexions lentes.

Après avoir comparé la mise en action de ces deux systèmes, la remarque fondamentale est que l'ordinateur sert à résoudre des problèmes bien posés tant que le cerveau humain peut résoudre des problèmes mal posés en plus. En bref l'intelligence artificielle n'est rien d'autres qu'une modélisation du cerveau humain qui est capable de réaliser les deux tâches principales l'apprentissage et la reconnaissance qui se considèrent le principe d'apparition de l'intelligence artificielle.

Historiquement, les réseaux de neurones étaient bien développés d'une manière synchronisée avec la mutation géante de l'informatique. En 1943 McCulloch et Pitts [130] ont débutés des travaux concernant les réseaux de neurones. Six ans plus tard, la notion d'apprentissage a été mise en évidence par Donald Hebb [91]. Cette dernière date passée de neuf ans, Rosenblat

[166] introduisait le perceptron qui est le modèle le plus classique et le plus simple des réseaux de neurones.

En principe, la tâche envisagée par un perceptron est d'avoir déterminé un ensemble de valeurs que peuvent prendre certains paramètres propres au perceptron s'appelant les coefficients des poids (matrices) afin de pouvoir obtenir des valeurs désirées à la sortie du perceptron.

En vue d'aboutir à cette conséquence, le type d'apprentissage utilisé dans ce cas est supervisé, plus précisément il s'agit de celui de Widrow-Hoff [191, 192] qui consiste à rendre le perceptron capable d'apprendre des échantillons pré-présentés à lui via ses entrées, afin d'obtenir des valeurs bien voulues à ses sorties. Ceci est réalisable après avoir modifié d'une manière répétitive les paramètres de connexion.

Pour bien éclairer la tâche prévue par le perceptron, il faut préciser que celui-ci n'est capable que d'apprendre ainsi que classifier des objets qui sont parfaitement linéairement séparables. Cette incapacité est due à l'architecture du perceptron qui ne comprend qu'une couche d'entrée et une autre de sortie pas plus.

Pour surmonter cet inconvénient, Rosenblatt réussissait d'introduire un nouveau modèle du perceptron qui s'appelle le perceptron multicouche [49] en vue de résoudre des problèmes qui sont non-linéairement séparables.

Dans ce propos, l'apprentissage d'un tel perceptron multicouche s'effectuera en vertu de plusieurs algorithmes spécifiquement celui de rétro propagation du gradient [4, 116].

IV.1.1.1 Définition

Les réseaux de neurones [7, 49, 50, 69, 70, 78, 146, 176] ont été fondés en principe afin de simuler les propriétés du cerveau humain en vue de mettre en place des systèmes de calcul ayant une capacité importante pour résoudre des problèmes trop complexes.

IV.1.1.1.1 Le neurone biologique

Le neurone biologique est une cellule vivante qui est composée des éléments biologiques suivants : un corps cellulaire, un axone, des synapses et des dendrites qui peuvent contacter avec des milliers d'autres neurones.

Le neurone biologique sert d'assurer la transmission des signaux bioélectrique portant le nom influx nerveux via l'axone et qui se diffuse par le canal d'un réseau qui n'est rien d'autres que les dendrites (voir Figure IV.34).

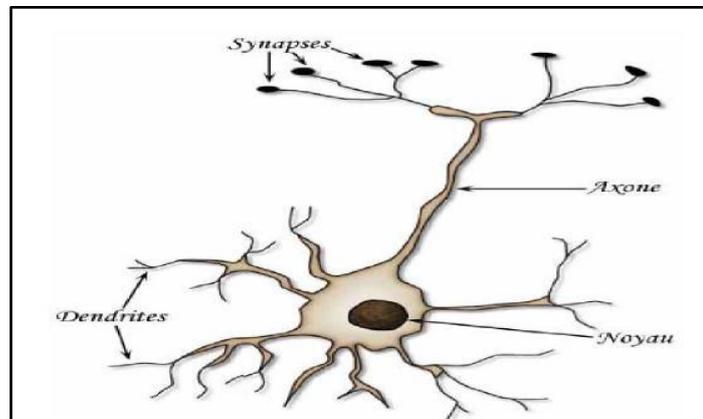


Figure IV.34 : Représentation de la forme biologique d'un neurone [21]

IV.1.1.1.2 Le neurone formel

Le neurone formel est l'unité de base ou l'unité élémentaire au niveau de l'architecture du réseau de neurones. Il est connecté :

- En entrée : à plusieurs sources d'informations.
- En sortie : il renvoie des informations.

Plus extensivement, un neurone formel ou artificiel est un processeur élémentaire qui reçoit des variables aux entrées qui peuvent être booléennes, binaires (0,1), bipolaires (-1,1) ou réelles. Chacune des entrées est associé un poids W_{ij} représentant la force de la connexion et liant la sortie du neurone qui peut être binaire, bipolaire ou réelle (voir Figure IV.35).

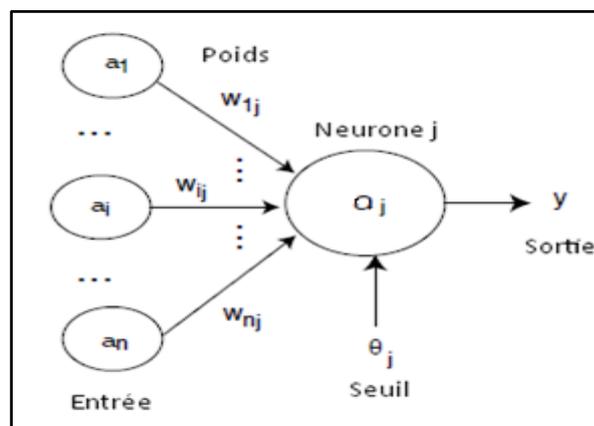


Figure IV.35 : Architecture d'un neurone formel

La sortie du neurone notée y est calculée en fonction de son état d'activation :

$$y = \varphi(\sum_{i=1}^n W_{ij} * a_i) \quad (19)$$

Avec :

φ : Fonction d'activation ou de transfert du neurone.

$W_{ij, (1 \leq i \leq n, 1 \leq j \leq m)}$: Matrice des poids liant les neurones d'entrées et la sortie.

a_i : Les neurones d'entrée

θ_j : Le seuil

y : La sortie

- L'activation d'un neurone est soumise à la condition suivante :

$$\text{Le neurone est} = \begin{cases} \text{actif} & \text{si } y \geq \text{seuil} \\ \text{inactif} & \text{sinon} \end{cases} \quad (20)$$

IV.1.1.1.3 Fonction d'activation

Cette fonction permet de définir l'état interne du neurone en fonction de son entrée totale, donnons à titre d'exemple quelques fonctions qui sont les plus utilisées.

✓ La fonction sigmoïde ou logistique : $f(x) = \frac{1}{(1+\exp(-x))}$

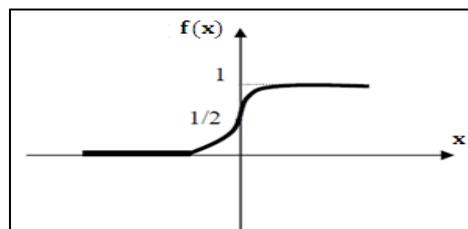


Figure IV.36 : La représentation graphique de la fonction sigmoïde

✓ La fonction tangente hyperbolique : $f(x) = \tanh(x)$.

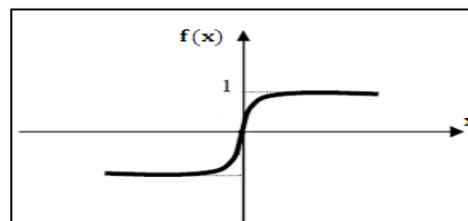


Figure IV.37 : La représentation graphique de la fonction tanh(x)

- ✓ La fonction linéaire : $f(x) = x$

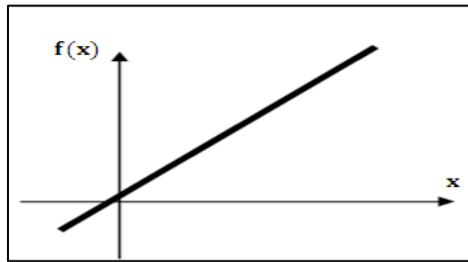


Figure IV.38 : La représentation graphique de la fonction linéaire

- ✓ La fonction à seuil:

- La fonction Heaviside : $H(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases}$

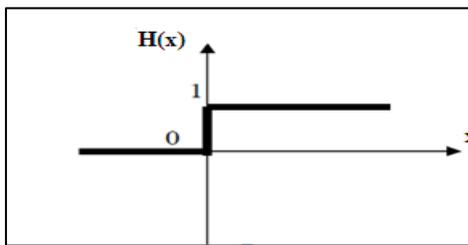


Figure IV.39 : La représentation graphique de la fonction Heaviside

- La fonction Signe : $Sgn(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ -1 & \text{sinon} \end{cases}$

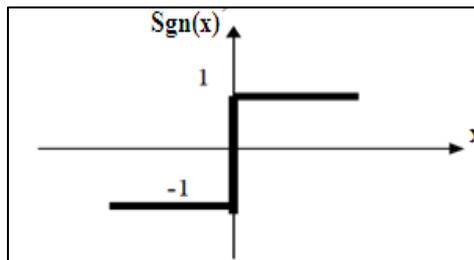


Figure IV.40 : La représentation graphique de la fonction signe

IV.1.1.2 Architecture d'un réseau de neurones

L'architecture d'un réseau de neurones peut prendre plusieurs formes selon les problèmes traités par ce réseau. A ce moment-là, deux paramètres font intervenir en vue de mettre en place cette architecture, le premier est de nature physique, et le deuxième fait parti de l'algorithmique, à savoir :

- Le nombre de neurones ainsi que celui de couches formant le réseau.
- L'algorithme d'apprentissage adopté pour optimiser les valeurs des coefficients synaptiques du réseau afin que celui-ci puisse efficacement résoudre les problèmes traités.

IV.1.1.2.1 Le perceptron monocouche

Ce type de réseau de neurones [166] se considère comme le modèle le plus classique et le plus simple parmi tous les types des réseaux.

Il s'agit en fait d'un réseau de neurones monocouche, plus explicitement n'ayant qu'une seule couche de sortie qui présente sa réponse et une couche d'entrée regroupant ses entrées se note la rétine du réseau, aucune couche cachée est intervenue dans son architecture, ces deux couches sont inter-reliées par les coefficients des poids W_{ij} . C'est pourquoi il se décrit par le réseau le plus simple (voir Figure IV.41).

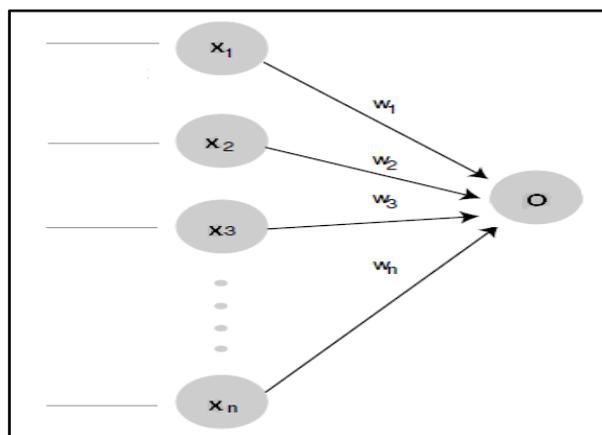


Figure IV.41: Architecture du perceptron.

Avec :

n : Le nombre total de neurones de la couche d'entrée et de coefficients synaptiques.

W_i : Les coefficients des poids (ou synaptiques) entre le $i^{\text{ème}}$ neurone d'entrée et la sortie.

Le perceptron calculera une sortie y en fonction de n variables d'entrées (x_1, x_2, \dots, x_n) selon la formule suivante.

$$y = \begin{cases} 1 & \text{si } \sum_{i=1}^n W_i * x_i > W_0 \\ 0 & \text{sinon} \end{cases} \quad (21)$$

W_0 : Seuil à préfixer.

IV.1.1.2.2 Le perceptron multicouche

Compte tenu de l'insuffisance du perceptron monocouche dans la résolution des problèmes trop compliqués notamment la séparation des données non linéairement séparables, les chercheurs dans le domaine de l'intelligence artificielle ont réussi de surpasser cette problématique après avoir développé un nouveau réseau de neurones nommé le perceptron multicouche (PMC) [49, 165] qu'ayant une capacité très élevée d'exécuter plusieurs tâches trop complexes à titre d'exemples pouvoir séparer des données non linéairement séparables.

Concernant l'architecture de ce type du réseau de neurones, elle est composée d'une couche d'entrée suivie d'une ou plusieurs couches cachées puis une couche de sortie (voir Figure IV.42).

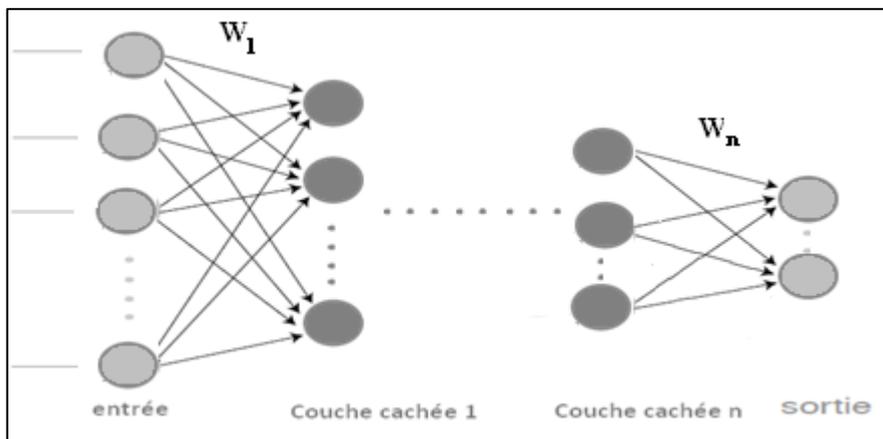


Figure IV.42: Architecture du perceptron multicouche.

Le PMC est composé :

- D'une couche d'entrée : l'ensemble de neurones d'entrées.
- D'une couche de sortie : l'ensemble de neurones de sortie.
- D'une ou plusieurs couches cachées situées entre la couche d'entrée et la couche de sortie connectées entre eux et qui n'ayant aucun contact avec l'extérieur. Chacune de ces couches est composée d'un certain nombre de neurones.

Les connexions $W_i, i=1, \dots, n$ sont toutes orientées séquentiellement partant de la couche d'entrée vers la couche de sortie, autrement dit chaque neurone d'une telle couche est connecté à ceux de la couche suivante.

IV.1.1.3 Apprentissage – classification par les réseaux de neurones

IV.1.1.3.1 Définition

Dans ce type d'apprentissage, Lorsque les exemples sont constitués de couples de valeurs c.-à-d. des données d'entrées et des sorties désirées, on cherchera à déterminer les coefficients des poids du réseau afin de pouvoir fournir des sorties voulues. Il s'agira donc de faire d'exiger le réseau de converger vers un état final précis.

A ce moment-là, plusieurs types d'algorithmes peuvent être utilisés selon le type d'apprentissage envisagé qui peut être :

IV.1.1.3.2 Séparation des données

IV.1.1.3.2.1 Cas des problèmes linéairement séparables

En vue de réaliser une telle tâche de séparation qui inclue à la fois un apprentissage et une classification, on est tenu d'abord opter un modèle de réseaux de neurones adéquat.

Puisque cette paragraphe se restreindra sur une séparation linéaire, il nous suffira d'utiliser le perceptron monocouche, en fait ce type de séparation ne nécessite pas forcément la présence d'une couche cachée. On fera recours d'adopter un algorithme d'apprentissage approprié. Dans ce propos on citera le plus fréquent qui est celui de Widrow-Hoff [191, 192] qui se présentera comme suit:

Algorithme V.2 : L'algorithme de Widrow-Hoff

Algorithme de Widrow-Hoff :

Entrée :

- 1- Un ensemble des données $\mathbf{x}_1, \dots, \mathbf{x}_n$.
- 2- α : coefficient d'apprentissage, $\alpha \in [0, 1]$.
- 3- ϵ : l'erreur entre la sortie désirée (théorique) et la sortie calculée (réelle).
- 4- Initialisation aléatoire des coefficients de poids $\mathbf{w}_i, i=1, \dots, n$

Répéter

Pour i de 1 à n

$$\begin{aligned}
 W_i(t + 1) &= W_i(t) + \Delta(W_i) \\
 &= W_i(t) + \alpha * (t_i - y_i) * x_i
 \end{aligned}$$

Jusqu'au $|t_i - y_i| < \epsilon$

Ou :

t_i : Sortie théorique.

y_i : Sortie réelle.

Fin pour

Fin répéter

La sortie y est obtenue par l'équation suivante :

$$y = \sum_{i=1}^n W_i * x_i \quad (22)$$

La mise à jour itérative des coefficients des poids W_i permettra d'avoir une sortie réelle très proche de celle qui est théorique, autrement dit l'erreur ϵ converge vers zéro, ce qui facilitera par conséquent la détermination d'un hyperplan optimal réalisant cette séparation envisagée.

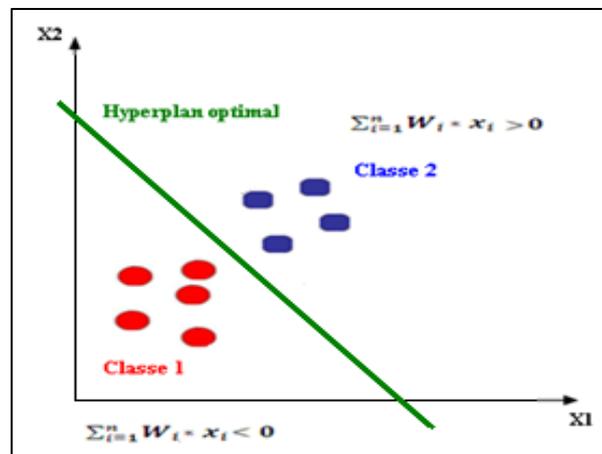


Figure IV.43 : La séparation linéaire

Dans ce propos, pour bien expliquer comment cette séparation se passera, on présente les deux exemples suivants :

- Exemple 1 : « **OU logique** »

$$\text{« OU »} \begin{cases} 00 & \rightarrow 0 \\ 01 & \rightarrow 1 \\ 10 & \rightarrow 1 \\ 11 & \rightarrow 1 \end{cases}$$

La sortie fournie par le perceptron dans ce cas est donnée par :

$$y = \begin{cases} 1 & \text{si } \sum_{i=1}^n W_i * x_i > 0 \\ 0 & \text{sinon} \end{cases} \quad (23)$$

Ce qui implique que le perceptron peut séparer le "OU" logique (voir Figure IV.44).

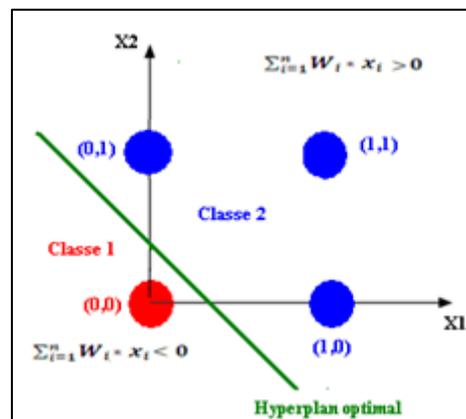


Figure IV.44 : La séparation de « OU logique »

- Exemple 2 : « **ET logique** »

$$\text{« ET »} \begin{cases} 00 & \rightarrow 0 \\ 01 & \rightarrow 0 \\ 10 & \rightarrow 0 \\ 11 & \rightarrow 1 \end{cases}$$

- Le perceptron peut aussi séparer le "ET" logique (voir Figure IV.45).

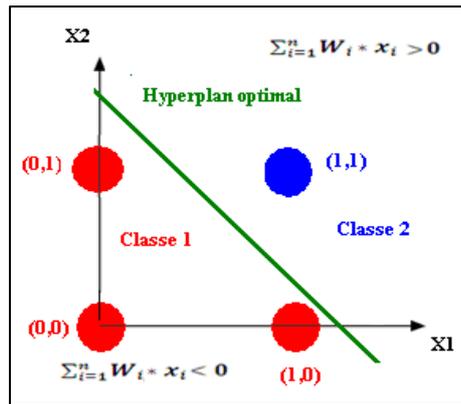


Figure IV.45 : La séparation de « ET logique »

IV.1.1.3.2.2 Cas des problèmes non linéairement séparables

Si les données non linéairement séparables, il n'existe aucun hyperplan permettant de séparer d'une manière optimale entre ces données. D'où la nécessité d'introduire ou plutôt de développer d'autres architectures neuronales afin de mise en œuvre des séparations entre des données qui sont en principe non linéairement séparables.

Pour bien expliquer ce cas, on présente l'exemple de « XOR » ou « OU exclusif ».

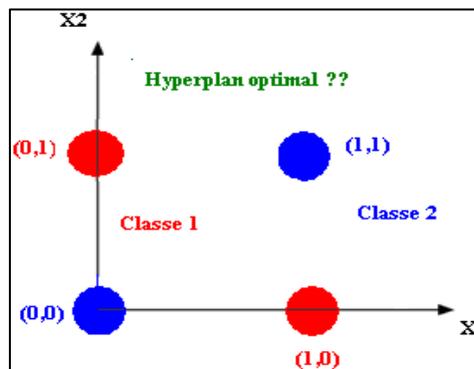


Figure IV.46 : L'impossibilité de séparation de « XOR »

Pour rendre ce problème linéairement séparable, il faudra utiliser un perceptron multicouche dont l'apprentissage s'effectuera par plusieurs algorithmes [35, 39, 62], parmi ceux il y'a celui nommé retro-propagation du gradient [4, 86, 115, 167, 178].

En fait, l'algorithme du gradient est une méthode d'optimisation ou minimisation des erreurs utilisée pour résoudre des problèmes que se soient linéaires ou pas. Elle consiste à déterminer le minimum d'une fonction de plusieurs variables ou paramètres par un ensemble des

itérations successives. Le gradient d'une fonction matricielle f est défini comme la matrice des dérivées partielles de cette fonction. Il est donné par :

$$\nabla f(X) = \frac{\partial f(X)}{\partial x_i} \quad (24)$$

Ou $X = (x_1, x_2, \dots, x_n)$ est le vecteur d'entrée, on cherchera à minimiser la fonction f de n variables.

En fait, le déroulement de cet algorithme contient les étapes suivantes :

➤ La correction des erreurs pour la couche de sortie :

On cherche à modifier tout en débutant par la couche de sortie et en terminant par la couche d'entrée, c'est pourquoi cet algorithme portant le nom rétro-propagation du gradient.

On cherche à mettre à jour les valeurs des coefficients synaptiques notées Z liant la couche cachée et la couche de sortie, ceci est réalisable via le calcul du gradient de l'erreur quadratique notée E par rapport à Z qui se calculera en utilisant la règle de dérivation des fonctions composées (réécrites en notation matricielle).

$$Z_{t+1} = Z_t + \Delta_Z E \quad (25)$$

$$= Z_t - \alpha * \nabla_Z E \quad (26)$$

$$= Z_t - \alpha * \frac{\partial E}{\partial Z} \quad (27)$$

- La $k^{\text{ème}}$ composante de l'erreur quadratique est donnée par :

$$E_k = \frac{1}{2} * (t_k - o_k)^2 \quad (28)$$

Après un développement des calculs on aura :

$$Z_{t+1} = Z_t + \alpha * \delta_{\text{sortie},k} + h_k \quad (29)$$

Ou :

- Les matrices de réponse des couches cachée et de sortie sont respectivement données par:

$$h = \frac{1}{(1 + \exp(-x * W))} \quad (30)$$

$$o = \frac{1}{(1 + \exp(-h * Z))} \quad (31)$$

- Les signaux d'erreur des couches de sortie et cachée sont respectivement donnés par :

$$\delta_{sortie,k} = (o_k - t_k) * o_k * (1 - o_k) \quad (32)$$

$$\delta_{caché,k} = \delta_{sortie,k} * Z * h_k * (1 - h_k) \quad (33)$$

- La correction des erreurs pour la couche cachée :

On cherche à modifier les valeurs des coefficients synaptiques notés W liant la couche d'entrée et la couche de sortie :

$$W_{t+1} = W_t + \Delta_W(E) \quad (34)$$

$$= W_t - \alpha * \nabla_W(E) \quad (35)$$

$$= W_t - \alpha * \frac{\partial E}{\partial W} \quad (36)$$

Après avoir développé les calculs on aura :

$$W_{t+1} = W_t + \alpha * \delta_{caché,k} + x_k \quad (37)$$

Pour plus de détails concernant cet algorithme voir Annexe E.

Algorithme V.3 : L'algorithme de rétro-propagation du gradient

Algorithme de retro-propagation du gradient:

Entrée :

- 1- Un ensemble des données x_i, \dots, x_n .
- 2- α : coefficient d'apprentissage, $\alpha \in [0, 1]$.
- 3- E : l'erreur quadratique entre la sortie désirée (théorique) et la sortie calculée (réelle).
- 4- Initialisation aléatoire des coefficients de poids $W_{i, i=1, \dots, n}$ et $Z_{i, i=1, \dots, m}$

Répéter

Pour t de 1 à n

$$Z_{t+1} = Z_t + \alpha * \delta_{sortie,k} + h_k$$

$$W_{t+1} = W_t + \alpha * \delta_{caché,k} + x_k$$

Jusqu'au $|E| < \epsilon$

Ou :

E_k : L'erreur quadratique donné par : $E_k = \frac{1}{2} * (t_k - o_k)^2$

t_k : Sortie théorique.

y_k : Sortie réelle.

δ_{sortie} : Le signal d'erreur de la couche de sortie.

$\delta_{\text{cachée}}$: Le signal d'erreur de la couche cachée.

Fin pour

Fin répéter

En vue de bien séparer le problème de « XOR » avec un perceptron deux solutions sont envisagées :

- On transformera l'espace des entrées de dimension qui est bidimensionnel en un autre espace tridimensionnel ce qui permettra de linéariser cette séparation qui est en principe non linéaire.
- On ajoutera une couche cachée située entre l'entrée et la sortie du perceptron, autrement dit on fera remplacer le perceptron monocouche par un autre ayant une couche cachée.

IV.1.2 Supports vecteurs machines (SVM)

IV.1.2.1 Introduction

Les supports vecteurs machines (en anglais The Support Vectors Machines SVM) [5, 20, 47, 100, 126, 182] font partie d'une famille d'algorithmes d'apprentissage supervisé servant à séparer d'une manière optimale entre deux classes (classification binaire) dont chacune d'entre elles contient un ensemble de vecteurs ainsi qu'à classifier des nouveaux vecteurs inconnus. Cette séparation consiste à maximiser autant que possible la marge entre les deux classes en vertu de la recherche d'un hyperplan optimal séparant correctement les vecteurs des deux classes de part et d'autre.

Plus précisément la marge maximale doit être entre les vecteurs les plus proches de l'hyperplan de la première classe et ceux les plus proches également de la deuxième classe.

Ce type de vecteurs s'appelle vecteurs de supports qui se sont l'appuie dont les SVMs se reposent.

Le principe est donc de déterminer une fonction de discrimination ou de décision qui n'est rien d'autres que l'hyperplan optimal et qui permettra de bien séparer les deux classes ainsi que classifier des nouveaux vecteurs inconnus [25, 87, 134, 188].

Le principe de base des SVMs consiste à ramener le problème de la discrimination à celui qui est linéaire, par la recherche d'un hyperplan optimal. Deux idées permettent d'aboutir cet objectif :

- La première consiste à définir l'hyperplan comme une solution d'un problème d'optimisation quadratique sous contraintes dont la fonction objectif ne s'exprimera qu'à l'aide de produits scalaires entre les vecteurs.
- La deuxième sert à introduire une fonction noyau dont le produit scalaire induisant implicitement une transformation non linéaire des données vers un espace intermédiaire de plus grande dimension.

Sur le plan théorique, la fonction noyau définit un espace Hilbertien, dit auto-reproduisant et isométrique par la transformation non linéaire de l'espace initial et dans lequel le problème linéaire sera facilement résolu. L'introduction de noyaux, spécifiquement adaptés à une problématique donnée, lui confère une grande flexibilité pour s'adapter à des situations très diverses comme reconnaissance de formes, de caractères, et diagnostics, etc.

IV.1.2.2 Les SVMs binaires

Un SVM binaire a pour but est de déterminer un séparateur optimal entre une première classe contenant un ensemble de vecteurs et une seconde contenant un ensemble des autres vecteurs. La complexité de cette séparation dépend en fait de est-ce qu'il s'agit de deux classes linéairement séparables ou non.

IV.1.2.2.1 Les SVMs linéaires

C'est le cas le plus simple des SVMs, la séparation optimale entre les deux classes ne nécessite pas tellement de calculs car elle s'effectuera juste par un hyperplan. Pour bien mettre les points sur les i , on est tenu d'explicitement comment cette séparation se mettra en œuvre. De ce fait, étant donné un ensemble (espace de données) $E \subseteq \mathbb{R}^n$ de vecteurs $x_i, i=1,2,\dots,m$

Une partie de E porte l'étiquette 1, tandis que l'autre partie est étiquetée par la valeur -1.

L'ensemble des étiquettes se note $y = \{-1, 1\}$. On nomme les vecteurs de la première partie la classe 1, et ceux de la deuxième la classe 2.

L'objectif de SVM est de trouver un classificateur nommé hyperplan (classificateur linéaire) séparant d'une manière optimale entre ces deux classes de part et d'autre, autrement dit celui qui maximise autant que possible la marge entre les deux classes : les exemples situés d'un côté de l'hyperplan sont classés positivement et ceux qui situent de l'autre côté sont classés négativement.

Cet hyperplan est caractérisé par un vecteur poids $w \in \mathbb{R}^n$ (qui est perpendiculaire à l'hyperplan) et un scalaire noté le biais $b \in \mathbb{R}$ et s'exprimera en fonction de x par l'équation suivante :

$$f(x) = wx + b \quad (38)$$

L'équation de l'hyperplan optimal vérifie les propriétés suivantes :

$$f(x) = \begin{cases} wx + b \geq 0 & \text{si } x \text{ se situe au dessus de l'hyperplan optimale} \\ wx + b \leq 0 & \text{si } x \text{ se situe au dessous de l'hyperplan optimale} \\ wx + b = 0 & \text{si } x \text{ appartient à l'hyperplan optimale} \end{cases}$$

Au sens de classification, l'équation de cet hyperplan s'appelle la fonction de décision.

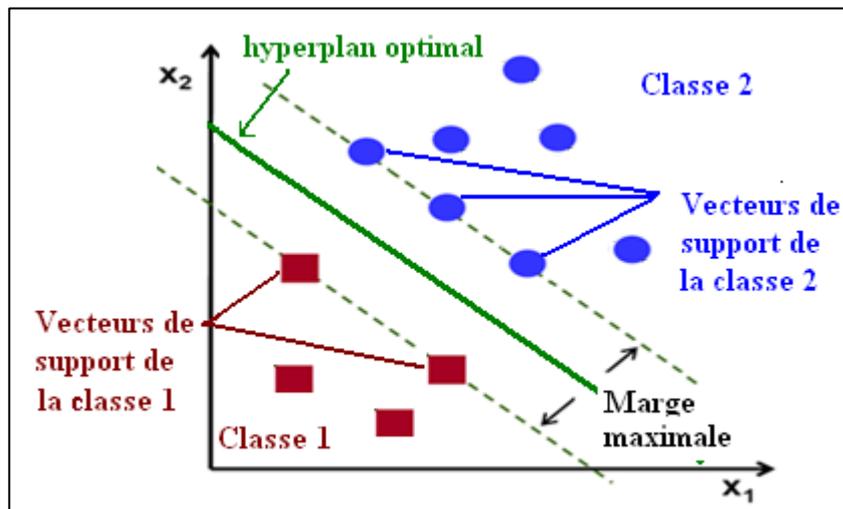


Figure IV.47 : Séparation linéaire entre deux classes avec un SVM

Maximiser cette marge entre les deux classes n'est rien d'autre que maximiser cette marge entre les vecteurs des deux classes qui sont les plus proches de l'hyperplan. Ces vecteurs s'appellent les vecteurs de support (voir figure ci-dessus).

Pour classer un nouveau vecteur inconnu \mathbf{X} , on calculera $f(\mathbf{X})$:

$$\text{Classe}(\mathbf{X}) = \text{Classe 1 si : } f(\mathbf{X}) \geq 0 \quad (39)$$

$$\text{Classe}(\mathbf{X}) = \text{Classe 2 si : } f(\mathbf{X}) \leq 0 \quad (40)$$

IV.1.2.2.1.1 Le problème primal

En vue de déterminer le problème primal des SVMs linéaires, on présentera les démarches suivantes :

Plus extensivement, cet hyperplan satisfait aux deux conditions suivantes :

$$\begin{cases} w \cdot x_i + b \geq 1 & \text{si } y_i = 1 \\ w \cdot x_i + b \leq -1 & \text{si } y_i = -1 \end{cases} \quad (41)$$

Ce qui est équivalent à :

$$y_i(w \cdot x_i + b) \geq 1 \quad \text{pour } i = 1, \dots, m \quad (42)$$

Comme nous l'avons déjà signalé, un hyperplan optimal est celui qui maximise autant que possible la marge que la note M qui représente la plus petite distance entre les différents vecteurs des deux classes à l'hyperplan.

Disons maximiser cette marge, cela est équivalent en fait à maximiser la somme des distances des deux classes par rapport à l'hyperplan. Ainsi, cette marge s'exprime par la formule mathématique suivante :

$$M = \min_{x_i | y_i = 1} \frac{w \cdot x + b}{\|w\|} - \min_{x_i | y_i = -1} \frac{w \cdot x + b}{\|w\|} \quad (43)$$

$$= \frac{1}{\|w\|} - \frac{-1}{\|w\|}$$

$$= \frac{2}{\|w\|} \quad (44)$$

Trouver l'hyperplan optimal revient donc à maximiser $\frac{2}{\|w\|}$, ce qui est équivalent à minimiser $\frac{1}{2} \|w\|^2$ sous la contrainte $y_i(w \cdot x_i + b) \geq 1$ pour tout $i=1,2,\dots,m$. Il s'agit alors d'un problème de minimisation d'une fonction objectif quadratique sous des contraintes linéaires.

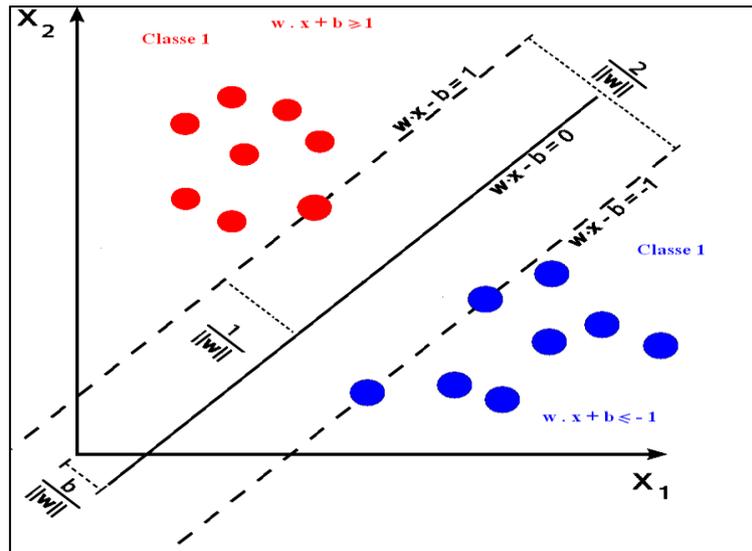


Figure IV.48 : Différentes démarches pour trouver le problème primal

De ce fait-la, le problème primal du SVM pourra s'exprimer par :

$$\begin{cases} \min_{w,b} \frac{1}{2} \|w\|^2 \\ \text{SC } y_i(wx_i + b) \geq 1 \quad \forall i = 1, \dots, n \end{cases} \quad (45)$$

IV.1.2.2.1.2 Le problème dual

D'après la théorie de l'optimisation [voir annexe G], et comme la fonction objectif à minimiser est $\frac{1}{2} \|w\|^2$ sous les contraintes $y_i(wx_i + b) \geq 1$, ce problème peut être résolu en vertu de l'opérateur Lagrangien [voir annexe G]:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i (wx_i + b) - 1] \quad (46)$$

Où les α_i étant les multiplicateurs de Lagrange.

En vue de pouvoir obtenir la formule explicite du problème dual il faudra d'abord annuler les dérivées partielles par rapport au vecteur poids w d'une part et par rapport au biais b d'autre part. D'où le problème dual est :

$$\begin{cases} \max_{\alpha} (\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i x_j)) \\ \text{SC } \sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha_i \geq 0 \quad \forall i = 1, \dots, n \end{cases} \quad (47)$$

La résolution de ce type des équations est un problème qui fait partie de la programmation quadratique convexe [48], qui en pratique peut être traitée par une des implémentations disponibles.

En fait, les multiplicateurs de Lagrange α_i vérifiant l'annulation des dérivées partielles correspondent aux vecteurs de support. Tous les autres vecteurs ont un α_i nul. Finalement, l'équation de l'hyperplan séparateur (fonction de décision) est donnée par :

$$f(x) = (w^*x) + b = \sum_{i=1}^n \alpha_i^* y_i \cdot (x \cdot x_i) + b^* \quad (48)$$

Où les α_i^* sont les α_i non nuls et b^* est trouvé en remplaçant les coordonnées d'un vecteur de support x_i de classe y_i dans $w x_i + b > 0$ si $y_i = 1$ ou dans $w x_i + b < 0$ si $y_i = -1$.

Remarquons que la complexité de la décision est forcément pertinente au nombre de vecteurs de support. On aura alors une décision d'autant plus rapide si le nombre de vecteurs de support est faible et vice-versa.

Néanmoins, dans les problèmes réels, il est toutefois rare que les classes soient linéairement séparables. A ce moment-là, la contrainte de bon classement qui est définie initialement par :

$$y_i(w x_i + b) \geq 1 \quad (49)$$

Doit être relâchée par l'intermédiaire d'un paramètre ξ_i pour se devenir :

$$y_i(w x_i + b) \geq 1 - \xi_i \quad (50)$$

Dans ce cas, la plus grande marge possible entre l'hyperplan et les vecteurs de support, initialement obtenue par minimisation de $\frac{1}{2} \|w\|^2$ doit désormais être obtenue par la minimisation de :

$$\begin{cases} \min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ \text{SC } y_i(w x_i + b) \geq 1 - \xi_i \quad \forall i = 1, \dots, n \end{cases} \quad (51)$$

Où :

- C désigne un paramètre strictement positif à fixer auparavant.
- $\xi_{i,i=1,\dots,n}$ s'appellent les variables de ressort.

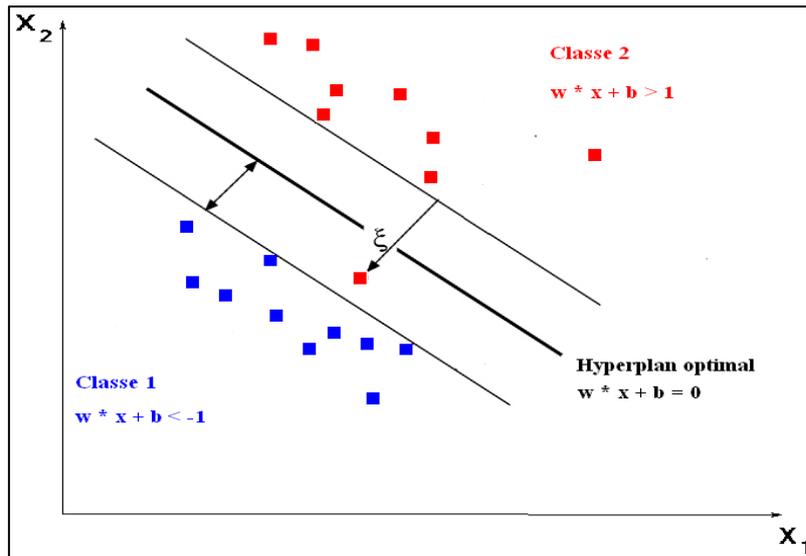


Figure IV.49 : Présentation de la marge souple

Intuitivement, on se rend compte sans aucun doute de présenter une méthode de séparation linéaire assez limitée. D'où la nécessité d'introduire des nouveaux modèles des SVMs pouvant traiter efficacement les problèmes non linéairement séparables.

IV.1.2.3 Les SVMs non linéaires

Il s'agit cette fois-ci des deux classes qui sont en principe non linéairement séparables, l'idée des SVMs dans ce cas est de rendre cette séparation linéaire. Cela pourra être accessible via une transformation de l'espace des données de dimension n en un autre espace de plus grande dimension p qui s'appelle l'espace de ré-description (voir Figure IV.50).

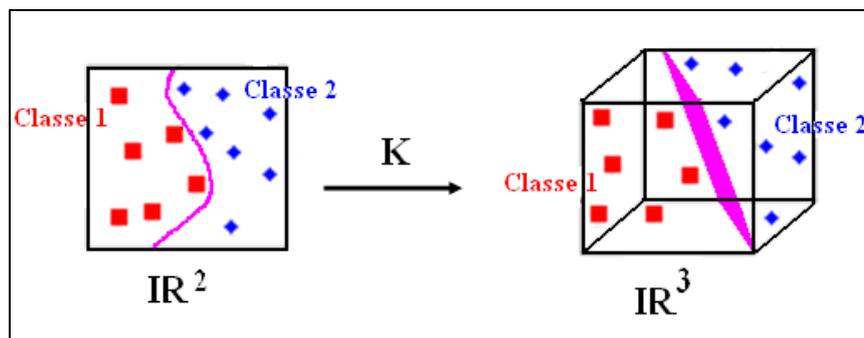


Figure IV.50: Transformation d'une séparation non linéaire dans \mathbb{R}^2 en une autre linéaire dans \mathbb{R}^3

Pratiquement, on fait recours à introduire les fonctions noyau satisfaisant d'abord aux conditions dites de Mercer [voir annexe G] qu'en fait permettront de réaliser des

séparations non linéaires. Effectivement, on montre que l'équation de l'hyperplan séparateur peut être reformulée en fonction d'un produit scalaire (x, x_i) :

$$f(x) = (w^* x) + b^* = \sum_{i=1}^n \alpha_i^* y_i \cdot (x, x_i) + b^* \quad (52)$$

Justement, le terme (x, x_i) pourra remplacer par n'importe quelle fonction noyau $K(x, x_i)$ réalisant un produit scalaire. L'équation de l'hyperplan deviendra:

$$f(x) = \sum_{i=1}^n \alpha_i^* y_i \cdot K(x, x_i) + b^* \quad (53)$$

Dans ce cas le problème dual deviendra comme suit :

$$\begin{cases} \max_{\alpha} (\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j)) \\ \text{SC} \quad \sum_{i=1}^n \alpha_i y_i = 0, \alpha_i \geq 0 \quad \forall i = 1, \dots, n \end{cases} \quad (54)$$

◇ La fonction de décision sera :

$$f(x) = \sum_{i=1}^n \alpha_i^* y_i K(x, x_i) + b \quad (55)$$

Les fonctions noyaux couramment utilisées sont :

◇ Le noyau linéaire :

$$K(x, x_i) = x * x_i \quad (56)$$

◇ Le noyau fonction de base radiale (RBF) :

$$K(x, x_i) = e^{-\frac{\|x-x_i\|^2}{2\sigma^2}} \quad (57)$$

◇ Le noyau polynomial :

$$K(x, x_i) = (x * x_i + c)^2 \quad (58)$$

◇ Le noyau sigmoïde :

$$K(x, x_i) = \tanh(x * x_i + c) \quad (59)$$

La figure suivante montre des exemples de frontières de décision obtenues avec différents types de noyaux :

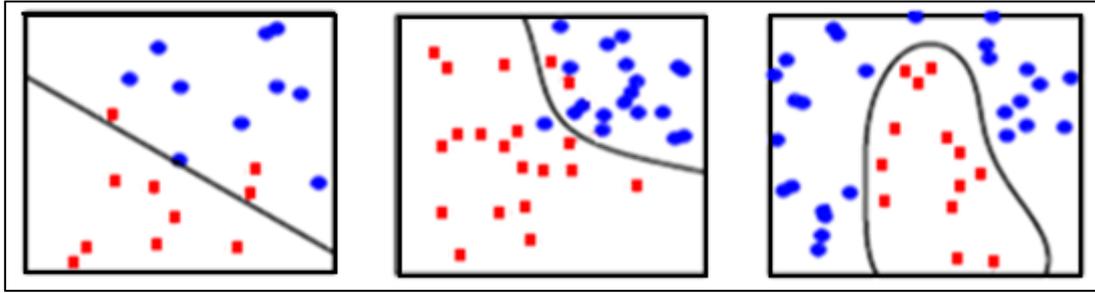


Figure IV.51: Les frontières de décision obtenues par trois SVM à fonction noyau Linéaire, polynomial et RBF.

IV.1.2.4 Les SVMs multi-classes

Rappelons qu'en principe, les SVMs sont des classificateurs binaires. Bien que certains travaux cherchent à généraliser l'utilisation des SVMs dans des problèmes de multi classes [134, 179]. En réalité, le problème du multi classes est généralement traité par combinaison de classificateurs binaires. Pour un problème à N classes, il existe deux catégories principales de méthodes de combinaison :

- La première approche notée un-contre-tous consiste à utiliser N classificateurs permettront de faire la discrimination de chacune des classes contre toutes les autres. La règle de décision utilisée dans ce cas est généralement le maximum, ou l'on affectera au vecteur inconnu la classe associée au SVM dont la sortie est la plus grande.
- L'autre approche nommée un-contre-un consiste à mettre en œuvre $\frac{N(N-1)}{2}$ classificateurs pour chaque paire de classes possible. On attribuera cette fois-ci à un vecteur inconnu la classe majoritaire parmi les $\frac{N(N-1)}{2}$ fonctions de décision.

IV.1.3 Les K-plus proches voisins (K-ppv)

La recherche de voisinage est utilisée dans de nombreux domaines, telles que la reconnaissance des formes, les techniques de compression des données y inclus des images, signaux, textes, et autres types de mesures, etc. En vue que cette recherche soit accessible, il faudra en principe mise en œuvre des méthodes s'articulant sur des fonctions dites de similarités particulièrement la distance Euclidienne tenant compte son efficacité, ainsi que sa simplicité.

Parmi ces méthodes, il y a celle qui est nommée les k-plus proches voisins (K-ppv) [23, 28, 65, 80, 124, 171, 194, 196] qui fait partie de la famille d'apprentissage supervisé.

IV.1.3.1 Principe de classification par K-ppv

Il s'agit d'une approche très simple et directe du fait qu'elle ne nécessite pas carrément aucun apprentissage mais tout simplement juste un stockage des données d'apprentissage. Son principe est comme suit :

Une donnée de classe inconnue est comparée à toutes les données déjà stockées. On choisira pour cette nouvelle donnée la classe majoritaire parmi ses K-plus proches voisins (voir Figure IV.52).

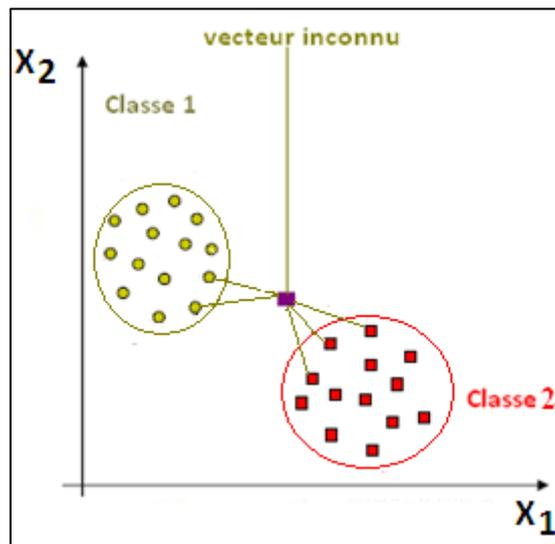


Figure IV.52: 5-ppv du vecteur inconnu X qui appartiennent à deux classes différentes

Compte tenu des données présentées dans la figure ci-dessous, la classe de vecteur X est celle portant le numéro 1.

Cependant, quelle classification choisit-t-elle en cas d'égalité des nombres des plus proches voisins ?

Dans ce propos, les solutions suivantes sont envisagées :

1. Augmenter la valeur de k pour rendre une telle classe dominante par rapport aux autres.
2. Tirer au hasard la classe parmi les classes ambiguës.
3. Pondérer des exemples par leur distance au vecteur inconnu X.
4. Attribuer le vecteur inconnu X à la classe dont appartient le très plus proche voisin, c'est-à-dire celui qu'ayant la distance la plus minimale de X parmi tous ses K-ppv.

En revanche, La méthode de K-ppv nécessite justement :

- Un entier K.
- Une base d'apprentissage.
- Une distance.

Mais de quelle distance s'agira-t-il ?

IV.1.3.2 L'importance du choix de la distance

Pour ce faire, Le choix d'une telle distance est primordial au bon fonctionnement de la méthode des K-ppv. En effet, une telle distance entre deux vecteurs donnés $X, Y \in \mathbb{R}^n$ doit avoir les quatre propriétés :

- $D(X, Y) \geq 0$ (non-négativité)
- $D(X, Y) = 0$ si et seulement si $X = Y$ (réflexivité)
- $D(X, Y) = D(Y, X)$ (symétrie)
- $D(X, Y) + D(Y, Z) \geq D(X, Z), \forall Z \in \mathbb{R}^n$ (inégalité triangulaire)

A ce moment-là, plusieurs types des distances entre X et Y sont disponibles, à savoir :

- Distance Euclidienne :

$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (60)$$

- Distance «city block» ou «Manhattan» (somme des valeurs absolues) :

$$\sqrt{\sum_{i=1}^n |x_i - y_i|} \quad (61)$$

- Distance de Tchebychev :

$$\lim_{p \rightarrow \infty} \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p} = \sup_{1 \leq i \leq n} |x_i - y_i| \quad (62)$$

- Distance de Minkowski :

$$\sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p} \quad (63)$$

- Ces distances peuvent être exprimées tout brièvement par :

$$d_\lambda(X, Y) = \left[\sum_{i=1}^n |x_i - y_i|^\lambda \right]^{\frac{1}{\lambda}} \quad (64)$$

- $\lambda=1$ distance de Manhattan.

- $\lambda=2$ distance Euclidienne.
- $\lambda \rightarrow \infty$ distance Chebyshev.
- $\lambda \rightarrow p$ distance Minkowski.

IV.1.3.3 L'importance du choix de nombre K

Le choix du paramètre k dans la règle des K -ppv a une influence tout à fait directe sur le style du classificateur ainsi déterminé. Effectivement, une faible valeur de K va donner un classificateur de bonne résolution (définir des frontières compliquées entre classes) mais très sensible au bruit sur les échantillons et sur le vecteur à classer. Une valeur grande de k aura un comportement inverse, lissant la frontière mais peu sensible au bruit. A savoir donc que :

- K grand:
 - Moins sensible au bruit.
 - Une grande base d'apprentissage permet une plus grande valeur de K .
- K petit :
 - Rend mieux compte de structures fines.
 - Nécessaire pour des petites bases d'apprentissage.

IV.1.3.4 Algorithme de K-ppv

Algorithme V.4 : Algorithme de K -ppv

➤ **Algorithme de K-ppv:**

Entrée :

- 1- Un ensemble des vecteurs $X_1, X_2, \dots, X_n \in \mathbb{R}^p$.
- 2- $k \in \mathbb{N}$.
- 3- Un vecteur inconnu X_{test} .

Répéter

Pour i de 1 à n

Calculer la distance Euclidienne

$$D(X_{\text{test}}, X_i)$$

Fin pour

Pour i de 1 à n

Fixer le nombre k.

Déterminer les k-plus proches voisins de X_{test} .

Pour chaque $X_i \in K\text{-ppv}(X_{\text{test}})$

Compter le nombre d'occurrence de ces plus proches voisins de chaque classe.

Fin pour

Fin pour

Attribuer à X_{test} la classe la plus fréquente.

Fin répéter

IV.2 Méthodes d'apprentissage – classification non supervisées

Il s'agit cette fois-ci d'un type d'apprentissage qui consiste à faire rendre un tel système de reconnaissance ayant des valeurs d'entrée connues capable de réaliser une auto-adaptation afin de pouvoir produire des valeurs de sortie très proches à celles d'entrée. En d'autre terme, le système devra être convergé librement vers un état final quelconque.

IV.2.1 Modèle de Markov caché (MMC)

IV.2.1.1 Processus stochastique

Un **processus stochastique** (ou processus aléatoire) est une séquence X_1, X_2, \dots, X_N de variables aléatoires fondées sur un même ensemble fondamental noté espace des états $E = \{1, 2, \dots, N\}$. Les valeurs possibles des variables aléatoires sont appelées les états possibles du processus. La variable X_t représente l'état du processus au temps t, $t = \{1, 2, \dots, T\}$. (on dit aussi l'observation au temps t).

Les différentes variables aléatoires ne sont pas en général indépendantes les unes aux autres. En effet cette dépendance est la propriété dans laquelle se réside l'utilité des processus stochastiques.

En revanche, Pour définir entièrement un processus stochastique, il suffira d'expliciter :

- Une loi de probabilité de la première variable aléatoire X_1 qui détermine donc l'état du processus lors de la première observation.
- Pour toute valeur de $t > 1$ la probabilité conditionnelle :

$$P(X_t = j \mid X_{t-1} = i_{t-1}, \dots, X_1 = i_1) \quad (65)$$

IV.2.1.2 Chaîne de Markov

Une **chaîne de Markov** ou **processus Markovien** est un type particulier de processus stochastiques qui satisfait aux deux conditions suivantes :

- L'état au temps t du processus ne dépend que de son état au temps $t-1$:

$$P(X_t = j \mid X_{t-1} = i_{t-1}, \dots, X_1 = i_1) = P(X_t = j \mid X_{t-1} = i_{t-1}) \quad (66)$$

Compte tenu cette définition, un processus Markovien est dit processus sans mémoire.

- La probabilité de passage d'un état i à un état j ne varie pas avec le temps :

$$\forall t, 1 < t \leq N, P(X_t = j \mid X_{t-1} = i) = C \quad (67)$$

Une chaîne de Markov peut être décrit par une matrice de transition A telle que :

$$A(i, j) = P(X_t = j \mid X_{t-1} = i), 1 < t \leq N \quad (68)$$

Avec $A(i, j) \geq 0, \forall i, j$ et

$$\sum_{j=1}^N A(i, j) = 1 \quad \forall i \quad (69)$$

L'état du processus à l'instant 1 notée π de la variable X_1 :

$\pi(i) = P(X_1 = i)$, Une chaîne de Markov peut aussi être représentée par un automate fini :

- Chaque état du processus est représenté par un état de l'automate
- Une transition de l'état i à l'état j est étiqueté par la probabilité $A(i, j)$.

A s'appelle la matrice des probabilités de transition, c'est une matrice stochastique, ceci veut dire que la somme de toutes les probabilités situées dans chaque ligne de A vaut 1. Une chaîne de Markov pourra être présentée sous forme d'un graphe probabiliste ou les nœuds sont les états tandis que les arcs sont les probabilités de transition du processus.

Exemple : Soit une chaîne de Markov définie par les paramètres suivants :

Un espace d'états : $E = \{1, 2, 3, 4\}$, et une matrice des probabilités de transition :

$$A = \begin{pmatrix} \frac{3}{5} & \frac{3}{10} & 0 & \frac{1}{10} \\ \frac{1}{10} & \frac{2}{5} & \frac{1}{2} & 0 \\ \frac{3}{4} & 0 & \frac{1}{5} & \frac{1}{20} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Le graphe associé à cette chaîne de Markov est :

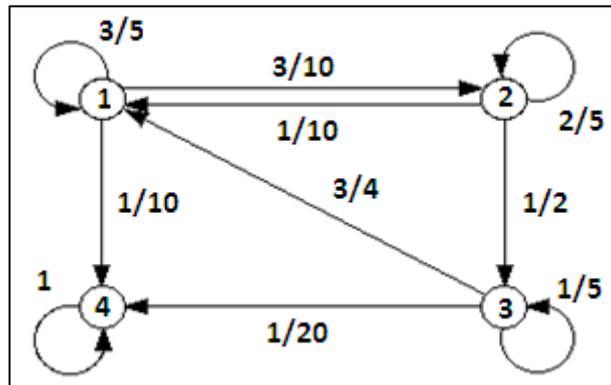


Figure IV.53: Chaîne de Markov modélisée par un graphe

IV.2.1.3 Modèle de Markov caché

Dans les chaînes de Markov, les observations correspondent aux états du processus. Dans un modèle de Markov caché, on ne peut observer directement les états du processus, mais des symboles (appelés aussi observables) émis par les états selon une certaine loi de probabilité. Au vu d'une séquence d'observation on ne peut savoir par quelle séquence d'états (ou chemin) le processus est passé, d'où le nom de modèles de Markov cachés (MMC). On distingue le processus $X = X_1, X_2, \dots, X_T$ qui représente l'évolution des états du MMC et le processus $O = O_1, O_2, \dots, O_T$ qui représente la suite des symboles émis par le MMC.

IV.2.1.3.1 Eléments d'un MMC

Un MMC est défini par un $\lambda = (A, B, \Pi)$ où :

- A est la matrice des probabilités de transition d'un état vers un autre. La probabilité de transition d'un état i vers un état j ($P(X_t = j | X_{t-1} = i)$) est notée $A(i, j)$. Comme dans le cas des chaînes de Markov, la somme des probabilités des transitions émanant d'un état vaut 1 :

$$\sum_{j=1}^N A(i, j) = 1 \quad \forall i \in S \quad (70)$$

- B est la matrice de probabilités des observations $\{b_1, b_2, \dots, b_M\}$ des symboles notés $\{v_1, v_2, \dots, v_M\}$ émis par les états vérifiant :

$$\sum_{i=1}^M B(i, j) = 1 \quad \forall i \in E \quad (71)$$

- π est la loi de probabilité de l'état initial $\pi(i) = P(X_1 = i)$, et on a :

$$\sum_{i=1}^N \pi_i = 1 \quad (72)$$

IV.2.1.3.2 Les topologies de MMC

Les principales topologies de modèles de Markov cachés sont celles ergodique, gauche-droite et parallèles.

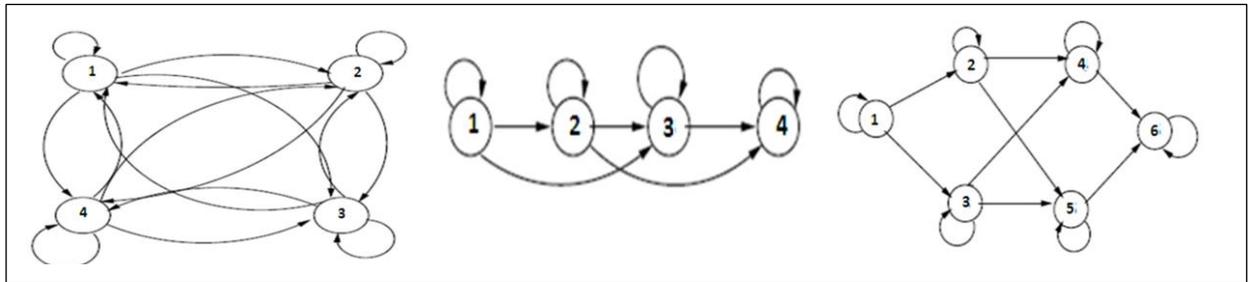


Figure IV.54: Les modèles : Ergodique, Gauche-droite : parallèle.

IV.2.1.3.3 Types de densité d'observations

Le modèle de Markov caché $\lambda = (A, B, \Pi)$ expliqué ci-dessus concerne en fait les modèles dont les observations sont discrètes, cependant quand il s'agira des observations continues, le modèle sera cette fois-ci exprimé par le :

$$\lambda_i = \{A, \Pi, \mu_i, \sigma_i\}. \quad (73)$$

- La matrice des probabilités des observations est donnée par:

$$b_i(v_k) = P(O_t = v_k | q_t = i) = \frac{1}{(2\pi)^{\frac{1}{2}} \sigma_i} e^{-\frac{(O_t - \mu_i)^2}{2\sigma_i^2}} \quad (74)$$

Ou :

μ_i et σ_i sont respectivement le vecteur moyen et le vecteur écart type caractérisant chacun des états occupé par le processus.

IV.2.1.3.4 Problèmes fondamentaux associés au MMC

Lorsqu'on travaille avec un modèle de Markov caché correspondant à un ensemble de séquences, l'utilisation du MMC permet de résoudre trois problèmes de base qui sont posés par Rabiner dans [155, 156].

IV.2.1.3.4.1 Problème d'évaluation :

Il existe plusieurs algorithmes afin d'évaluer la vraisemblance $P(O|\lambda)$ d'une séquence d'observation O , on se restreint à l'algorithme Forward - backward [161]. Le déroulement dont lequel cet algorithme s'effectuera se présente comme suit:

✓ L'algorithme Avant - Arrière (Forward - Backward)

Etant donné une suite d'observations $O=O_1O_2\dots O_T$ et une suite d'états associée $Q = q_1q_2\dots q_T$. La probabilité d'observer O , connaissant le modèle λ est égale à la somme suivant tous les chemins d'états possibles Q des probabilités conjointes de O .

$$P(O|Q) = \sum_Q P(O, Q|\lambda) \quad (75)$$

Algorithme V.5 : Algorithme de Forward

➤ Algorithme de Forward (voir Annexe F):

Initialisation : $\alpha_t(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N$

Réurrence : $\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) a_{ij}] \times b_j(O_{t+1}) \quad t \in [1, T - 1], 1 \leq j \leq N$

Terminaison : $P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$

Algorithme V.6 : Algorithme de Backward

➤ Algorithme de Backward (voir Annexe F):

Initialisation : $\beta_T(i) = 1 \quad 1 \leq i \leq N$

Réurrence : $\beta_t(i) = [\sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)] \quad T - 1 \geq t \geq 1, 1 \leq j \leq N$

Terminaison : $P(O|\lambda) = \sum_{i,j=1}^N \pi_j b_j(O_1) \beta_1(i)$

IV.2.1.3.4.2 Problème d'apprentissage ou d'entraînement

L'idée de l'apprentissage est d'ajuster les paramètres du modèle pour maximiser la vraisemblance. Étant donnée une base d'apprentissage composée de séquences d'observations, les paramètres d'un MMC sont premièrement initialisés puis, ré-estimés itérativement en augmentant la vraisemblance du modèle vis à vis des séquences d'observations d'entraînement. Le processus d'entraînement d'un modèle s'arrête quand la vraisemblance du modèle atteint une valeur maximale.

- Choisir un ensemble initial de paramètres λ_0
- Ré-estimer $\lambda = (A, B, \Pi)$ en calculant λ_1 à partir de λ_0 et ainsi de suite.
- Tant que (il y'a augmentation de $P(O | \lambda)$) ou (Il y a encore des itérations à Faire)

Algorithme V.7 : Algorithme de Baum Welch

➤ **Algorithme de Baum Welch (voir Annexe F):**

Fixer les valeurs initiales :

$$a_{ij}^0, b_j^0(k), \pi_i^0 \quad 1 \leq i, j \leq N, 1 \leq k \leq N$$

Calculer à l'aide des fonctions de Forward-Backward :

$$\xi_t(i, j) \quad \gamma_t(i) \quad 1 \leq i, j \leq N, 1 \leq t \leq T - 1$$

Et en utilisant les formules de ré-estimation.

Recommencer en 2 jusqu'à ce qu'un certain critère de convergence soit rempli.

IV.2.1.3.4.3 Problème de reconnaissance ou décodage :

Le but du décodage consiste à déterminer le chemin correspond à l'observation $O = (o_1, o_2, o_3 \dots, o_t)$, c'est-à-dire qui maximise la vraisemblance de la séquence d'observations en cherchant la meilleure suite d'états $Q = (q_1, q_2, q_3 \dots, q_t)$.

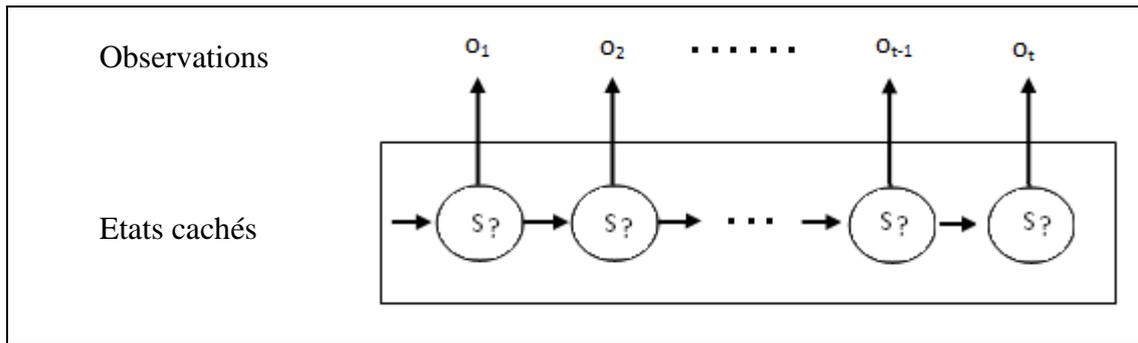


Figure IV.55 : Représentation de décodage.

La recherche de la meilleure suite d'états est basée sur l'algorithme de Viterbi [75 ,161] qui est un algorithme récursif qui permet de trouver à partir d'une suite d'observations, une solution optimale au problème d'estimation de la suite d'états.

Algorithme V.8 : Algorithme de Viterbi

➤ **Algorithme de Viterbi (voir Annexe F):**

Initialisation : $t = 1, 1 \leq i \leq N, \delta_1(i) = \pi_i b_i(O_1)$

Récurrence : $t \in [2, N], 1 \leq i \leq N$

$$\delta_{t+1}(i) = \max_{j=1 \dots n} (\delta_t(j) * a_{j,i}) * b_i(O_{t+1})$$

$$\psi_t(i) = \arg \max_{j=1 \dots n} (\delta_{t-1}(j) * a_{j,i})$$

Terminaison : $S(N) = \arg \max_i (\delta_T(i))$

Retour en arrière : $N-1 \geq i \geq 1 S(t) = \psi_{t+1}(s(t+1))$

IV.2.1.4 Classification par le MMC

Après avoir ré-estimé en vertu de l'algorithme de Baum-Welch [150] (voir annexe F) chacun des modèles initiaux $\lambda_0 = (A_0, B_0, \Pi_0)$ modélisant chaque vecteur d'observations en vue de maximiser la probabilité qui a engendré l'observation sachant le modèle initial. Tous les modèles optimaux seront stockés afin de former une base d'apprentissage.

Afin de reconnaître une observation inconnue, il faut calculer la probabilité générant cette observation par tous les modèles optimaux à l'aide de l'algorithme de Forward ou celui de Viterbi. La reconnaissance sera à ce moment-là ramenée à la classe d'observation pour laquelle le modèle optimal est celui qui a donné la probabilité la plus élevée.

Le processus de traitement pour le modèle de Markov commence après l'extraction des caractéristiques. Dans notre modélisation chaque état correspond à un élément du vecteur d'extraction. Nous avons considéré le vecteur d'extraction des caractéristiques à partir de l'image de caractère comme des observations pour notre modèle. Connaissant la classe à laquelle appartient le caractère celui-ci est comparé aux modèles λ_k .

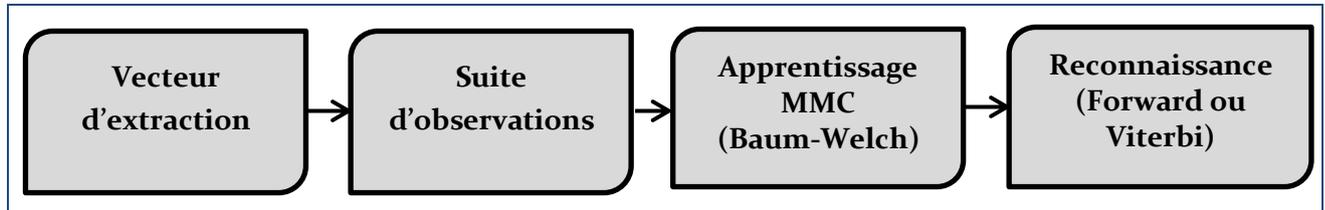


Figure IV.56 : Processus de reconnaissance par un MMC

IV.2.2 Système Hybride PMC/MMC

Le modèle hybride PMC-MMC est un modèle utilisé dans le domaine de la reconnaissance des formes pour lequel le PMC s'exploite comme un estimateur des probabilités à posteriori. Effectivement, dès que les sorties désirées par le PMC sont acquises, celles-ci s'utilisent comme des vecteurs d'observations présentés au MMC qui doit être s'entraîner par l'algorithme de Baum-Welch juste pour maximiser la probabilité d'apparition de ces observations, juste après l'achèvement de cet entraînement (apprentissage), chacun des caractères c présentera par la suite une probabilité maximale $P(O_c|\lambda_c)$.

Tous ces modèles optimaux λ^* doivent être enregistrés avec les matrices de connexions W^* et Z^* du PMC dans une base d'apprentissage.

Après avoir classifié un tel caractère de test en vertu du PMC via la production d'un vecteur de sortie, celui-ci doit se présenter comme un vecteur d'observation dont lequel on calculera sa probabilité d'apparition par tous les λ^* à l'aide de l'algorithme de Forward.

La reconnaissance sera attribuée au caractère dont le modèle optimal est celui qui a donné la probabilité maximale $\max(P(O|\lambda_k))$

$$\text{Classe de chiffre test} = \arg(\max_i(P(\text{Observation} | \lambda_i^*))) \quad (76)$$

En fait, dans ce propos, plusieurs systèmes ont été développés [11, 67, 68, 70, 79, 133, 147, 197] Ces systèmes présentent beaucoup d'avantages par rapport aux approches purement markoviennes. Cependant, il n'est pas évident de pouvoir mettre en œuvre ou plutôt en évidence un tel système qui fait partie de cette hybridation en raison du nombre élevé de

paramètres à ajuster ainsi que la grande quantité de données d'apprentissage nécessaires permettant à réaliser ce système.

Dans ce cadre, et juste pour savoir comment cette hybridation s'effectuera-t-elle, on est tenu de présenter le schéma suivant :

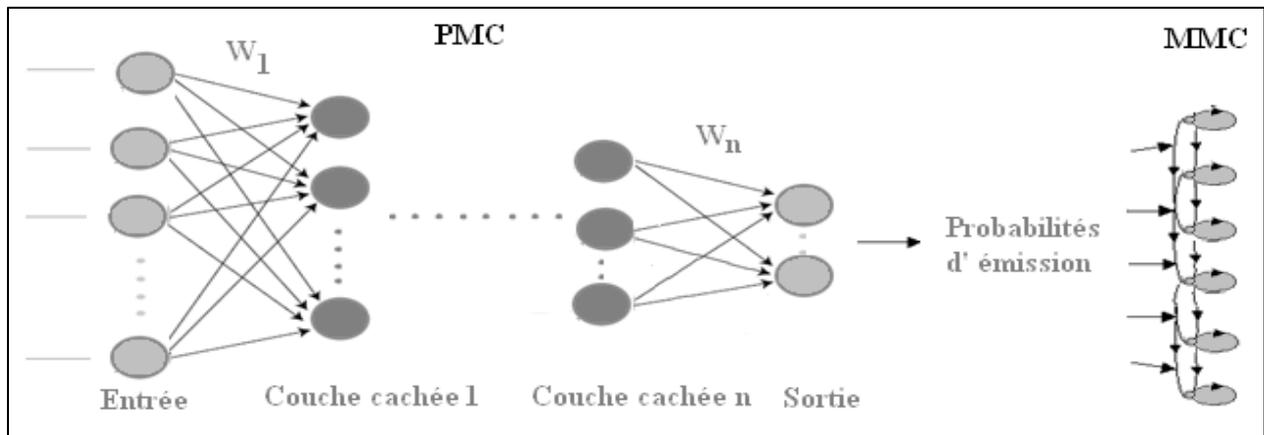


Figure IV.57: Schéma global de conception du modèle hybride PMC/MMC

IV.3 Conclusion

Plusieurs méthodes sont utilisées dans l'apprentissage et la classification par les différents systèmes de reconnaissance des formes. Les performances de ces méthodes varie de l'une à l'autre selon le type d'apprentissage c'est-à-dire supervisé ou non, ainsi que selon le genre du système de reconnaissance qui peut être neuronale, statistique ou probabiliste, etc.

Dans ce propos, l'apprentissage supervisé est une technique privilégiée permettant de ramener un certain système de reconnaissance des formes puissant, d'apprendre d'une façon efficace ces formes présentées dans ses entrées afin de pouvoir fournir des sorties bien voulues, tout se passe comme si le système est obligé de converger forcément vers ces sorties. Au niveau des méthodes s'appuyant sur ce type d'apprentissage, il y a les réseaux de neurones, les séparateurs à vaste marge et les K-plus proches voisins, etc.

En revanche, l'apprentissage non supervisé est tel que juste les valeurs d'entrées d'un système de reconnaissance sont connues auparavant, il fallait donc rendre ce système adaptatif en vue de fournir des sorties très proches aux valeurs d'entrées. Plus précisément, il faudra laisser le système converger d'une manière tout à fait libre vers n'importe qu'elles valeurs de sorties. A titre d'exemple le modèle de Markov caché repose sur cette sorte d'apprentissage.

CHAPITRE V

TESTS ET RESULTATS

Ce chapitre se considère comme la pierre de l'angle de notre thèse, du fait qu'il est consacré à la mise en œuvre de plusieurs systèmes de reconnaissance des chiffres Arabes sous forme manuscrits ou cursifs et la reconnaissance des caractères Tifinaghs sous forme imprimés, manuscrits et cursifs ainsi que les textes. En effet, afin de bien mettre en valeur ces systèmes, on est intéressé à réaliser des comparaisons entre leurs performances.

Contenu du chapitre

<i>V.1</i>	<i>Introduction</i>	<i>105</i>
<i>V.2</i>	<i>Base de données des caractères Tifinaghs</i>	<i>105</i>
<i>V.2.1</i>	<i>Définition des caractères Tifinaghs</i>	<i>105</i>
<i>V.2.2</i>	<i>Base de données</i>	<i>106</i>
<i>V.2.2.1</i>	<i>Caractères Tifinaghs isolés manuscrits</i>	<i>106</i>
<i>V.2.2.2</i>	<i>Caractères Tifinaghs isolés imprimés</i>	<i>106</i>
<i>V.2.2.3</i>	<i>Caractères Tifinaghs isolés cursifs</i>	<i>107</i>
<i>V.2.2.4</i>	<i>Textes Tifinaghs</i>	<i>108</i>
<i>V.3</i>	<i>Base de données des chiffres Arabes</i>	<i>108</i>
<i>V.3.1</i>	<i>Chiffres manuscrits</i>	<i>108</i>
<i>V.3.2</i>	<i>Chiffres cursifs</i>	<i>109</i>
<i>V. 4</i>	<i>Mise en œuvre des différents systèmes de reconnaissance de formes</i>	<i>109</i>
<i>V.4.1</i>	<i>Reconnaissance des chiffres Arabes</i>	<i>109</i>
<i>V.4.1.1</i>	<i>Reconnaissance des chiffres Arabes manuscrits MNJSS par PMC</i>	<i>109</i>
<i>V.4.1.2</i>	<i>Reconnaissance des chiffres Arabes manuscrits MNJSS par SVM</i>	<i>113</i>
<i>V.4.1.3</i>	<i>Reconnaissance des chiffres Arabes manuscrits MNJSS par K-ppv</i>	<i>115</i>
<i>V.4.1.4</i>	<i>Reconnaissance des chiffres Arabes cursifs</i>	<i>120</i>
<i>V.4.2</i>	<i>Reconnaissance des caractères Tifinaghs isolés</i>	<i>123</i>
<i>V.4.2.1</i>	<i>Reconnaissance des caractères Tifinaghs imprimés et manuscrits</i>	<i>123</i>
<i>V.4.2.1.1</i>	<i>Reconnaissance par le PMC</i>	<i>123</i>
<i>V.4.2.1.2</i>	<i>Reconnaissance par le PMC et MMC</i>	<i>126</i>
<i>V.4.2.1.3</i>	<i>Reconnaissance des caractères Tifinaghs cursifs</i>	<i>129</i>
<i>V.4.2.1.3.1</i>	<i>Reconnaissance par le PMC</i>	<i>129</i>

<i>V.4.2.1.3.2</i>	<i>Reconnaissance par les SVMs</i>	<i>132</i>
<i>V.4.3</i>	<i>Reconnaissance des textes Tifinaghs</i>	<i>135</i>
<i>V.5</i>	<i>Conclusion</i>	<i>144</i>

V.1 Introduction

En vue de pouvoir comparer entre les performances des différentes méthodes utilisées dans un système de reconnaissance des formes que ce soient celles d'extraction des primitives ou celles d'apprentissage – classification, on a mis en œuvre plusieurs systèmes contenant des différentes techniques d'extraction hybrides et non hybrides, à savoir la morphologie mathématique, le zonage et le zig-zag, et contenant également diverses méthodes d'apprentissage – classification qui sont le perceptron multicouches, les séparateurs à vaste marge, les k-plus proches voisins et les modèles de Markov cachés.

V.2 Bases de données des caractères Tifinaghs

Il s'agit de quatre base de données formées de plusieurs caractères isolés manuscrits, imprimés et cursifs, et également des textes **Tifinaghs**.

V.2.1 Définition des caractères Tifinaghs

Le Tifinagh est un alphabet consonantique utilisé par l'amazighe au nord d'Afrique. Il est formé de trente trois caractères selon IRCAM (Institut Royal de la Culture Amazighe). Il a subi des modifications et des variations depuis son origine jusqu'aujourd'hui. Parmi ses origines:

- Le Tifinagh saharien.
- Le Tifinagh Touareg.

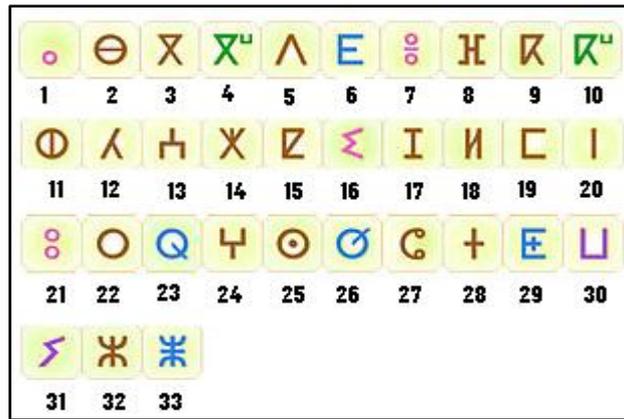


Figure V.58 : L'alphabet Tifinagh

V.2.2 Base de données

V.2.2.1 Caractères Tifinaghs isolés manuscrits

Cette base comprend des caractères Tifinaghs dont chacun d'entre eux est écrit par différents scripteurs:

Tableau V.I : Exemple de quelques caractères Tifinaghs manuscrits

Caractères manuscrits	Scripteur 1	Scripteur2	Scripteur3	Scripteur4	Scripteur5
ⵜ	ⵜ	ⵜ	ⵜ	ⵜ	ⵜ
ⵍ	ⵍ	ⵍ	ⵍ	ⵍ	ⵍ
ⵓ	ⵓ	ⵓ	ⵓ	ⵓ	ⵓ
ⵏ	ⵏ	ⵏ	ⵏ	ⵏ	ⵏ
ⵉ	ⵉ	ⵉ	ⵉ	ⵉ	ⵉ
ⵔ	ⵔ	ⵔ	ⵔ	ⵔ	ⵔ

V.2.2.2 Caractères Tifinaghs isolés imprimés

Le tableau ci-dessous contient un ensemble des caractères pour lesquels chacun parmi eux se diffère à ses homologues que ce soit au niveau de dimensionnement, ou d'aspect morphologique.

Tableau V.II: Exemple de quelques caractères Tifinaghs imprimés

Caractères imprimés	Forme 1	Forme 2	Forme 3	Forme 4	Forme 5

V.2.2.3 Caractères Tifinaghs isolés cursifs

Chaque caractère parmi ceux de cette base est écrit par cinq scripteurs différents :

Tableau V.III: Exemple de quelques caractères Tifinaghs cursifs

Caractères cursifs	Scripteur 1	Scripteur2	Scripteur3	Scripteur4	Scripteur5

V.2.2.4 Textes Tifinaghs

Cette base incluant différents types de textes Tifinaghs dont les premiers sont écrits par un marqueur et les deuxièmes par un stylo (voir Figures V.59-60).

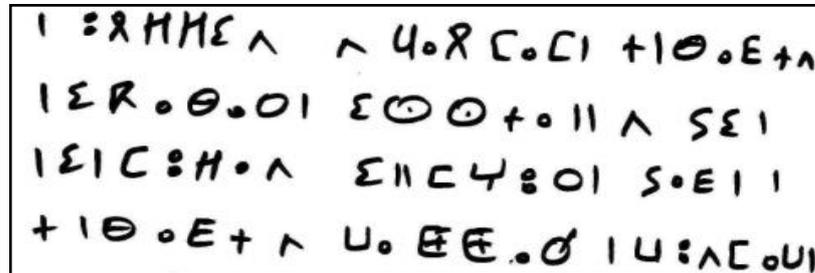


Figure V.59: Exemple d'un texte acquis par un scanner avec un marqueur

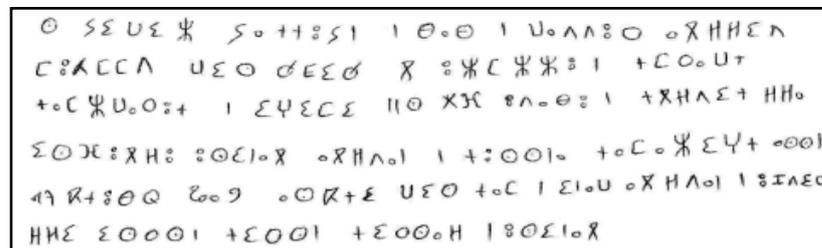


Figure V.60: Exemple d'un texte acquis par un scanner avec un stylo

V.3 Bases de données des chiffres Arabes

V.3.1 Chiffres manuscrits

Il s'agit cette fois-ci de deux bases de données, la première est celle de MNIST qui est gratuite et disponible sur le Web [114]. Elle se constitue des images des chiffres manuscrits. Elle est composée de 70000 chiffres (60000 en apprentissage) et (10000 en test). Les images sont stockées dans une base de données au format CDB. Du code C++ ou Matlab fourni par les auteurs permet d'extraire les chiffres isolés. La deuxième base inclut des chiffres cursifs (voir Figure V.61).



Figure V.61: Chiffres isolés de la base MNIST

V.3.2 Chiffres cursifs

Le tableau ci-dessous comprend ces chiffres et qui sont écrits par différents scripts

Tableau V.IV: Exemple de quelques chiffres cursifs

Chiffres cursifs	Scripteur 1	Scripteur2	Scripteur3	Scripteur4	Scripteur5
0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	9	9	9	9	9

V.4 Mise en œuvre des différents systèmes de reconnaissance de formes

V.4.1 Reconnaissance des chiffres Arabes

V.4.1.1 Reconnaissance des chiffres Arabes manuscrits MNIST par le PMC

En vue de mettre en action ce système on a adopté les données suivantes :

- ◆ Le prétraitement des images des chiffres a été réalisé par les techniques suivantes :
 - La normalisation des tailles des images à la valeur 28x28 pixels.

- Le seuillage dont le seuil est fixé à la valeur 0.5.
- Le centrage
- La rotation

A savoir que les techniques de prétraitements citées ci-dessus resteront les mêmes dans tous les systèmes de reconnaissance que nous ferons par la suite, sauf une contre indication.

- ◆ L'extraction des primitives a été effectuée par les méthodes suivantes :
 - La dilatation avec un élément structurant (la Morphologie mathématique)
 - La dilatation avec un algorithme sans aucun élément structurant.
- ◆ L'apprentissage – classification a été caractérisé par les données suivantes :
 - Le nombre des vecteurs d'entrée est égal à 10 dont chacun d'eux ayant 7 composantes.
 - Le nombre de neurones de la couche d'entrée est égal à 7 neurones.
 - Le nombre de neurones de la couche cachée est égal à 7 neurones.
 - Le nombre de neurones de la couche de sortie est égal à 10 neurones.
 - Le coefficient d'apprentissage η est fixé à la valeur 0.9.
 - La fonction d'activation utilisée est la fonction sigmoïde $F(\mathbf{x}) = \frac{1}{1+e^{-x}}$
 - L'algorithme adopté est celui de retro-propagation du gradient.

Après avoir mis en œuvre ce système de reconnaissance on a obtenu les valeurs de taux de reconnaissance de chaque chiffre τ_{rc} ou celui global τ_{rg} c'est-à-dire de tous les chiffres (10 chiffres) ou celui d'erreur de chaque chiffre τ_{ec} ou d'erreur global τ_{eg}

On rappelle que ces quatre taux $\in [0 \ 1]$ et sont inter-liés par les relations suivantes :

$$\tau_{rc} = \frac{\text{nombre de fois ou le chiffre est bien reconnu}}{\text{nombre total de fois ou le chiffre est présenté comme un chiffre de test}} \quad (77)$$

$$\tau_{rg} = \frac{\sum_{i=1}^{10} \tau_{rc,i}}{10} \quad (78)$$

$$\tau_{ec} = \frac{\text{nombre de fois ou le chiffre est mal reconnu}}{\text{nombre total de fois ou le chiffre est présenté comme un chiffre de test}} \quad (79)$$

Ou bien :

$$\tau_{rc} + \tau_{ec} = 100\% = 1 \quad (80)$$

$$\tau_{eg} = \frac{\sum_{i=1}^{10} \tau_{ec,i}}{10} \quad (81)$$

Tableau V.V: Le taux de reconnaissance de plusieurs ensembles des chiffres par PMC

Ensemble des chiffres Manuscripts	Nombre des chiffres	Base de validation $\tau_{cg} \times \%$	Base de test $\tau_{cg} \times \%$
1 Ensemble	10	61.50	40.22
10 Ensembles	100	71.90	68.98
50 Ensembles	500	82.00	77.43
100 Ensembles	1000	82.00	79.28
Nombre des images de test		1.000	60.000

- Le graphe associé est :

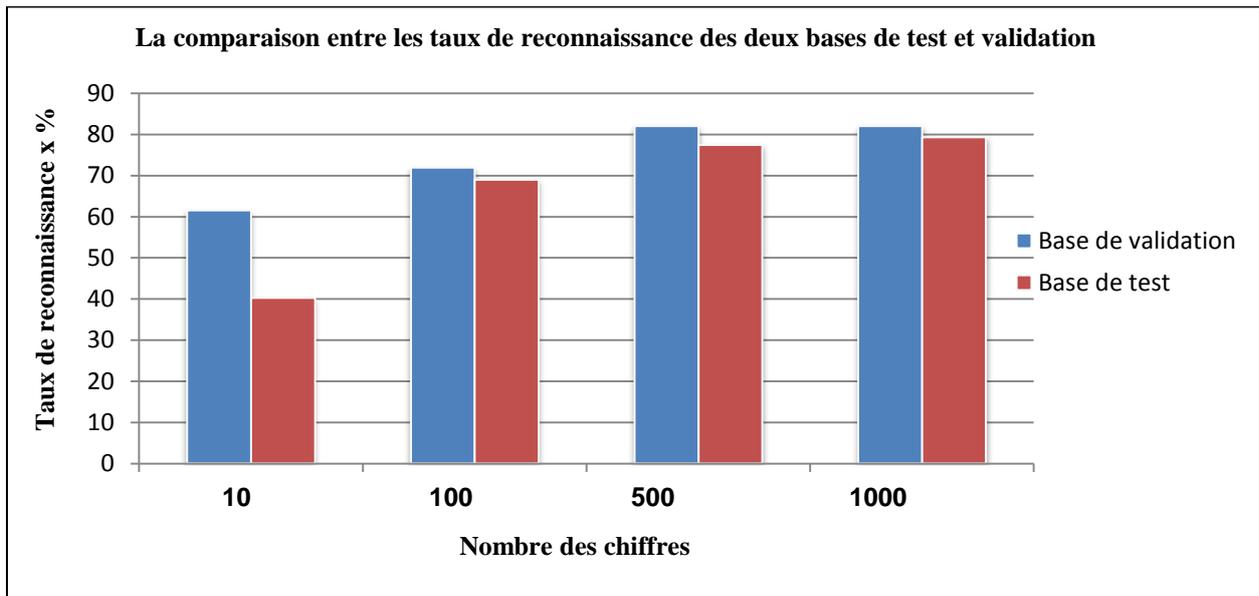


Figure V.62: La comparaison entre le taux de reconnaissance des deux bases de test et validation

- **Analyse et commentaire :**

Dans ce système de reconnaissance on a tenté d'une façon ou d'une autre d'optimiser la mise en œuvre du perceptron multicouche autrement dit le rendre plus capable de fournir des taux de reconnaissance trop élevés.

De ce fait l'idée consiste à faire varier progressivement le nombre d'échantillons des images des chiffres d'entrée en partant de 10 puis 100 puis 500 tout en terminant par 1000 images ce qui permettra d'augmenter le taux de reconnaissance global de la base des chiffres de test tout en introduisant une base des chiffres intermédiaire nommée la base de validation qui a pour rôle d'avoir évité autant que possible ce que s'appelle le sur-apprentissage, en extension cette base permet en fait de pouvoir obtenir les plus meilleurs paramètres du perceptron ce qui le

ramènera plus capable d'apprendre ainsi de classifier les chiffres d'une manière tout à fait correcte.

En plus de cela, les plus meilleurs paramètres obtenus sont sauvegardés en vue de les exploiter dans la reconnaissance des chiffres de test.

Le tableau ci-dessous englobe toutes les valeurs des taux de reconnaissance et d'erreur obtenues, ainsi que le temps de reconnaissance de chaque chiffre noté t_{rc} et celui global nommé t_{rg} . La relation entre les deux s'exprimera par :

$$t_{rg} = \sum_{i=1}^{10} t_{rc,i} \quad (82)$$

Tableau V.VI: Le Taux de reconnaissance obtenu de chaque chiffre par PMC

Chiffres	τ_{rc} x %	τ_{ec} x %	t_{rc} (s)
0	86.45	13.55	91.64
1	94.39	05.61	112.58
2	88.73	11.27	95.63
3	77.02	22.98	76.71
4	76.12	23.88	84.29
5	84.10	15.90	70.77
6	78.81	21.19	85.91
7	77.12	22.88	90.33
8	79.03	20.97	85.37
9	49.64	50.36	93.90
Grandeur globaux	τ_{rg} : 79.14%	τ_{eg} : 20.71%	t_{rg} : 887.13 s

– Le graphe associé est :

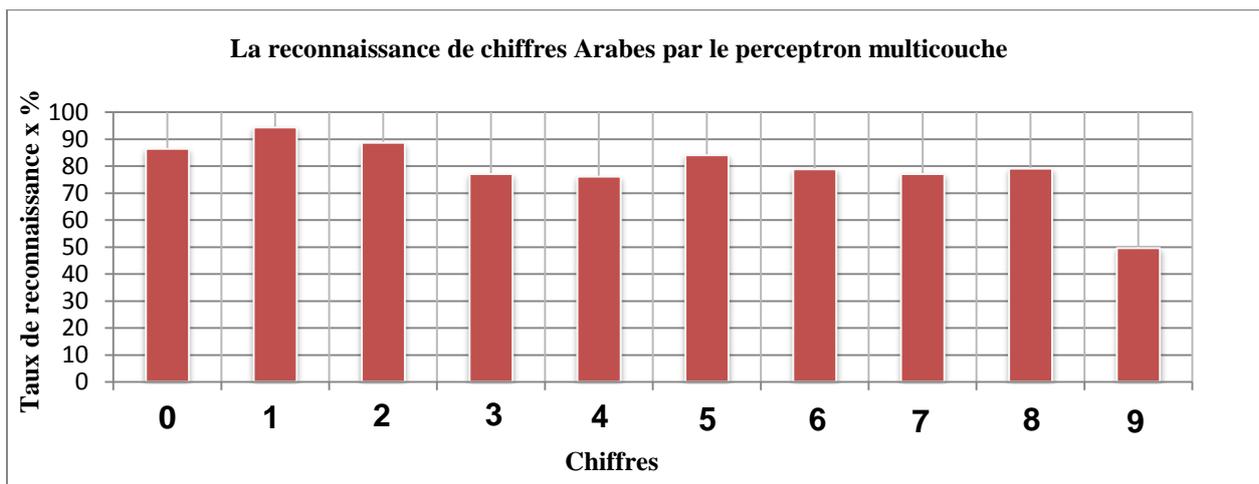


Figure V.63: La reconnaissance de chiffres Arabes par le perceptron multicouche

Tableau V.VII : La matrice de confusion des chiffres par PMC (donnée en %)

Perceptron multicouches										
Chiffres	0	1	2	3	4	5	6	7	8	9
0	86.45	00.81	01.10	00.01	01.40	00.27	00.22	00.17	06.36	03.21
1	00.02	94.39	01.00	00.84	00.02	00.48	00.44	00.18	01.03	01.59
2	00.09	01.42	88.73	04.33	00.82	00.65	00.03	01.00	02.00	00.93
3	00.06	00.52	00.96	77.02	01.18	00.45	00.00	14.53	04.95	00.32
4	01.21	02.92	00.34	01.44	77.94	02.85	03.30	00.58	03.59	05.83
5	00.03	01.37	01.27	01.03	00.74	84.10	06.73	00.22	02.58	01.92
6	02.29	02.41	00.81	00.10	06.21	02.43	78.81	00.00	06.78	00.15
7	00.04	03.47	06.66	03.83	01.50	00.04	00.00	77.12	04.84	02.48
8	04.82	02.58	01.50	02.03	04.08	00.22	00.70	03.16	79.03	01.86
9	04.85	09.89	08.34	00.65	02.70	08.14	02.51	01.90	11.37	49.64

- **Analyse et commentaire :**

En tenant compte des résultats présentés ci-dessus on pourra constater que la base de validation est un facteur nécessaire pour positiver les performances du perceptron. En fait, on a effectué plusieurs expériences sans et avec une base de validation ayant un nombre d'échantillons qui s'augmentera de plus en plus, on s'est bien rendu compte que le perceptron s'est ramené plus performant quand on a fait élever le nombre des échantillons.

En revanche, la matrice de confusion contient dix lignes/colonnes, à savoir qu'un tel élément situé à la $i^{\text{ème}}$ ligne et la $j^{\text{ème}}$ colonne correspond au taux de reconnaissance de chiffre j sachant que i est le chiffre de test.

Notons que si $i = j$ on dit que le chiffre est bien reconnu, et il est mal reconnu ailleurs.

V.4.1.2 Reconnaissance des chiffres Arabes manuscrits par SVM

On se propose maintenant de comparer entre les performances des différentes méthodes d'extraction des primitives tout en utilisant dans la phase d'apprentissage-classification les supports vecteurs machines.

Afin d'appliquer cette reconnaissance on a utilisé les données suivantes :

- ◆ L'extraction des primitives a été effectuée par les méthodes suivantes :

- Le Zonage.
- La dilatation (la Morphologie mathématique).
- Le Zig-Zag.
- La dilatation + Zig-Zag.

◆ L'apprentissage – classification a été effectué selon les données suivantes :

- La fonction noyau utilisé est celle de RBF ayant un écart type égale à 0.5.
- La stratégie utilisée pour les SVMs multi-classes est celle d'un contre tous.

Les valeurs obtenues des taux de reconnaissance de chaque chiffre ainsi que celui global sont regroupés dans le tableau suivant :

Tableau V.VIII: Le taux de reconnaissance des chiffres Arabes MNIST en fonction de nombre des zones par SVM

Chiffres MNIST	SVM					
	Zonage		$\tau_{rc} \times \%$ pour Morphologie	Zig-Zag	Morphologie	Morphologie + Zig-Zag
	$\tau_{rc} \times \%$ Pour 4 Zones	$\tau_{rc} \times \%$ Pour 9 Zones		$\tau_{rc} \times \%$ Pour 9 Zones	$\tau_{rc} \times \%$ Pour 9 Zones	$\tau_{rc} \times \%$ Pour 9 Zones
0	70.00	92.00	60.00	75.00	99.00	98.00
1	70.00	98.00	99.00	99.00	99.00	97.00
2	40.00	87.00	99.00	90.00	99.00	92.00
3	70.00	57.00	90.00	60.00	70.00	93.00
4	50.00	88.00	80.00	93.00	85.00	95.00
5	60.00	52.00	80.00	50.00	60.00	50.00
6	50.00	93.00	40.00	70.00	70.00	70.00
7	90.00	67.00	60.00	90.00	70.00	70.00
8	80.00	90.00	50.00	70.00	80.00	95.00
9	60.00	82.00	90.00	50.00	60.00	60.00
$\tau_{rg} \times \%$	64.00	80.60	74.80	74,70	79.20	82.00

– Le graphe associé est :

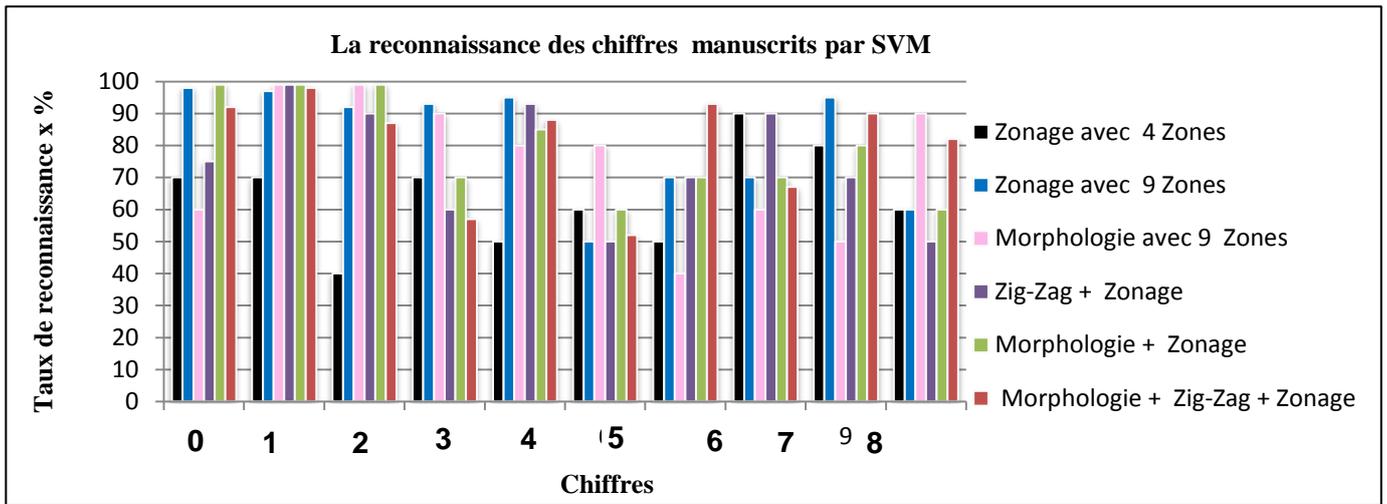


Figure V.64: Taux de reconnaissance des chiffres Arabes MNIST en fonction de nombre des zones avec SVM

- **Analyse et commentaire :**

Tout en exploitant les résultats produits par ce système de reconnaissance, on pourra effectivement conclure que :

Le taux de reconnaissance global obtenu varie d'une méthode d'extraction à une autre, plus précisément il croît d'une manière progressive quand on utilise un Zonage contenant 4 zones(64%), puis un Zig-Zag effectué selon 9 zones (74.7%), puis la dilatation (74.8%), puis une autre faite selon 9 zones (79.20%), puis un Zonage appliqué selon 9 zones (80.60%) et finalement la dilatation combinée avec un Zig-Zag et un Zonage contenant 9 zones (82%).

De ce fait, on déduit que le taux de reconnaissance global le plus médiocre est celui fourni par le Zonage effectué selon 4 zones tandis que le meilleur taux est obtenu par la méthode hybride qui combine à la fois la Morphologie, le Zig-Zag et le Zonage selon 9 zones.

V.4.1.3 Reconnaissance des chiffres Arabes manuscrits MNIST par K-ppv

On a réalisé en effet deux systèmes de reconnaissances ou on a opté la méthode de Zonage dans le premier, et les méthodes de Morphologie, zig-zig et l'hybridation entre les deux dans le deuxième.

- *Reconnaissance des chiffres Arabes manuscrits MNIST par la méthode de Zonage*

Le but envisagé de ce système reconnaissance est de pouvoir comparer entre les taux τ_{rc} et τ_{rg} obtenus pour chacune des valeurs de nombre de zones puis du nombre de plus proches voisins K. autrement dit savoir l'impact de variation de ces deux paramètres sur les performances de ce système de reconnaissance.

Les valeurs des taux τ_{rc} et τ_{rg} sont regroupés dans le tableau suivant :

Tableau V.IX: Le taux de reconnaissance des chiffres Arabes MNIST en fonction de nombre des zones par K-ppv

Chiffres MNIST	K-ppv							
	τ_{rc} x % pour K = 6		τ_{rc} x % pour K = 10		τ_{rc} x % pour K= 15		τ_{rc} x % pour K = 20	
	4 Zones	9 Zones	4 Zones	9 Zones	4 Zones	9 Zones	4 Zones	9 Zones
0	73.33	90.00	81.67	90.00	81.67	90.00	81.67	91.67
1	85.00	98.33	85.00	98.33	88.33	99.00	86.67	98.33
2	58.33	93.33	63.33	93.33	68.33	91.67	75.00	91.67
3	88.33	76.67	81.67	80.00	76.67	76.67	85.00	86.67
4	18.33	99.00	15.00	98.33	11.67	98.33	13.33	99.00
5	21.67	30.00	20.00	41.67	16.67	55.00	14.00	48.33
6	71.67	66.67	70.00	66.67	78.33	80.00	90.00	80.00
7	68.33	71.67	80.00	90.00	80.00	95.00	78.33	95.00
8	45.00	43.33	50.00	41.67	50.00	56.67	50.00	51.67
9	43.33	66.67	46.67	65.00	51.67	70.00	50.00	70.00
τ_{rg} x %	57.50	73.50	59.30	76.50	60.03	81.10	62.40	81.30

– Le graphe associé est :

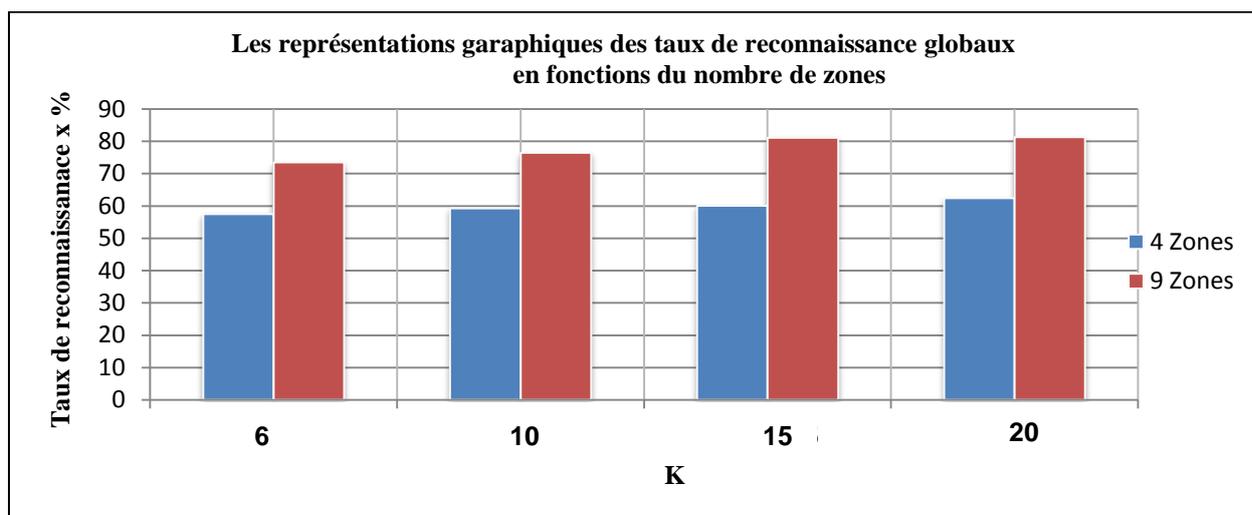


Figure V.65: Taux de reconnaissance des chiffres Arabes MNIST en fonction de nombre des zones avec K-ppv

- **Analyse et commentaire :**

Compte tenu des résultats cités dans le tableau ainsi que dans le graphe ci-dessus, on constate que quand on augmente soit le nombre de zones ou soit le nombre K de plus proche voisins, on aura par conséquent une augmentation de τ_{rc} et de τ_{rg} .

Effectivement, le fait d'augmenter le nombre de zones dans une image d'un tel chiffre ne peut signifier qu'à augmenter également la quantité des informations ou des caractéristiques extraites de ce chiffre ce qui le rend trop discriminant aux autres ce qui permettra par la suite de bien faciliter sa reconnaissance correcte. De l'autre côté, disons qu'on a fait élever la valeur de K, ceci s'expliquera par une élévation également du nombre de plus proches voisins autrement dit le nombre de fois ou le chiffre de test appartiendra à sa vraie classe.

- ***Reconnaissance des chiffres Arabes manuscrits MNIST par les méthodes de Morphologie, Zig-Zag et hybridation entre Morphologie et Zig-Zag***

L'objectif prévu cette fois-ci est d'un côté de pouvoir savoir l'effet de variation de nombre de plus proche voisins K sur les taux τ_{rc} et τ_{rg} dans chacune de ces méthodes d'extraction. Et de comparer entre les performances de ces méthodes de l'autre côté.

Les valeurs des taux τ_{rc} et τ_{rg} sont regroupés dans le tableau suivant :

Tableau V.X : Le taux de reconnaissance des chiffres en fonctions des méthodes d'extractions avec K-ppv

Chiffres MNIST	K-ppv											
	$\tau_{rc} \text{ x \% pour } K = 6$			$\tau_{rc} \text{ x \% pour } K = 10$			$\tau_{rc} \text{ x \% pour } K = 15$			$\tau_{rc} \text{ x \% pour } K = 20$		
	Zig-Zag avec Morph	Zig-Zag	Morph	Zig-Zag avec Morph	Zig-Zag	Morph	Zig-Zag avec Morph	Zig-Zag	Morph	Zig-Zag avec Morph	Zig-Zag	Morph
0	99.00	95.00	63.33	99.00	96.67	56.67	99.00	96.67	55.00	99.00	96.67	58.33
1	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00	99.00
2	88.33	90.00	83.33	80.00	90.00	85.00	76.67	88.33	85.00	73.33	83.33	86.67
3	88.33	85.00	90.00	91.67	86.67	90.00	95.00	85.00	90.00	95.00	88.33	88.33
4	96.67	95.00	91.67	99.00	93.33	93.33	99.00	93.33	96.67	99.00	95.00	93.33
5	31.67	41.67	56.67	45.00	58,33	71.76	48.33	61.67	73.33	43.33	61.67	78.33
6	98.33	90.00	95.00	99.00	96.67	96.67	99.00	96.67	98.33	99.00	96.67	98.33
7	95.00	99.00	68.33	99.00	99.00	63.33	98.33	99.00	46.67	98.33	99.00	35.00
8	81.67	68.33	60.00	80.00	55.00	56.67	76.67	60.00	58.33	76.67	60.00	65.00
9	85.00	58.33	66.57	73.33	56.67	48.33	75.00	73.33	63.33	75.00	78.33	61.67
$\tau_{rg} \text{ x \%}$	86.30	82.13	77.39	86.50	83.13	76.07	86.60	85.30	76.56	85.76	85.80	76.79

- Le graphe associé est :

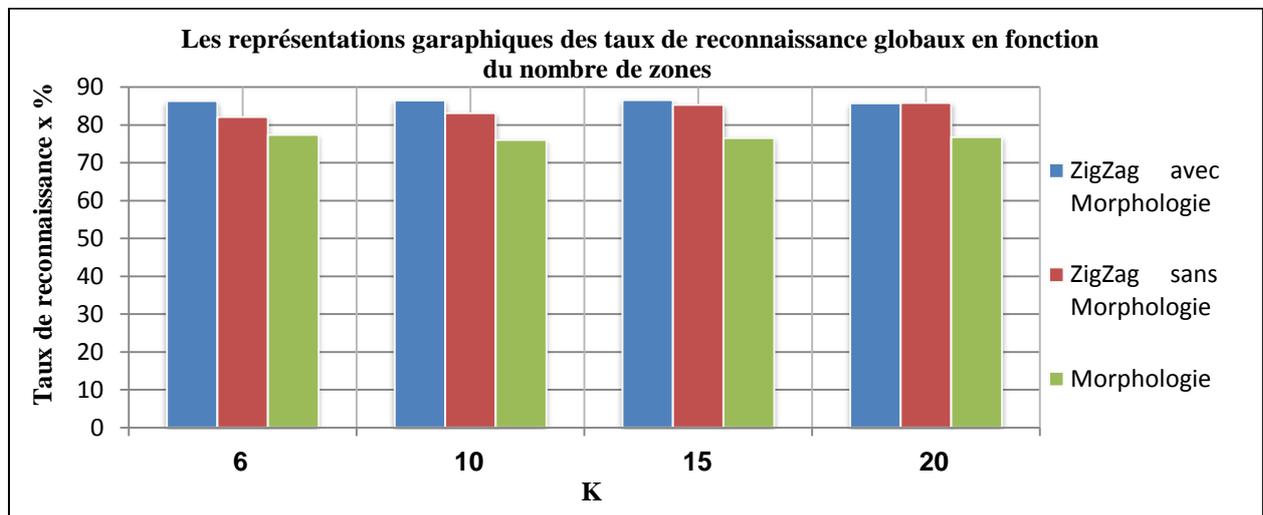


Figure V.66: Le taux de reconnaissance des chiffres en fonctions des méthodes d'extractions avec K-ppv

- **Analyse et commentaire :**

En s'appuyant sur ces résultats, on peut constater que la méthode hybride est celle qui a fourni le taux de reconnaissance global le plus élevé puis le Zig-Zag et finalement la Morphologie. L'augmentation du nombre de plus proches voisins K , se traduit par une augmentation du taux de reconnaissance global.

- ** Notion d'interface graphique**

Dans le sens de bien mettre en valeur la reconnaissance des chiffres Arabes manuscrits rien ne nous empêchera de faire introduire des outils qui permettront de réaliser cette finalité. C'est pourquoi on a fait appel à l'interface graphique présentée ci-dessous qui pourra se mettre en action si on suit certaines démarches.

Avant d'expliquer le comment de son activation, donnons juste une idée à propos des éléments constituant cette interface :

- un bouton comme son nom l'indique permet de télécharger une image d'un tel chiffre de test à partir d'une base englobant un ensemble de chiffres et qui est non représentée sur cet interface.
- Un bouton nommé sélection permet de choisir un chiffre de test à partir d'une sous interface contenant une base des images de ce chiffre, cette base est celle située au-dessous du bouton télécharger.
- Un bouton appelé prétraitement qui sert à nettoyer le chiffre de test qui sera apparu après avoir le nettoyé sur une sous interface située juste au-dessous de ce bouton.
- Un bouton qui se note dilatation qui sert à faire dilater le chiffre vers les quatre directions : Est-West-Nord-Sud.
- Un bouton d'extraction des primitives qui permettra d'extraire les caractéristiques du chiffre dilaté.
- Un bouton a pour rôle de réaliser un Zonage au sein de l'image de chiffre de test, le résultat de ce Zonage pourra être visualisé sur une sous interface se trouve justement au-dessous de ce bouton.
- Enfin, cinq boutons dont chacun d'entre eux permettra de réaliser la phase d'apprentissage – classification à savoir qu'il y ont a les réseaux de neurones, le modèle de Markov caché, les K -plus proches voisins, les supports vecteurs machines et le modèle hybride combinant à la fois le perceptron multicouche et le modèle de Markov caché.

Notons que la mise en action d'un tel bouton se fera juste par une seule petite clique sur celui-ci.

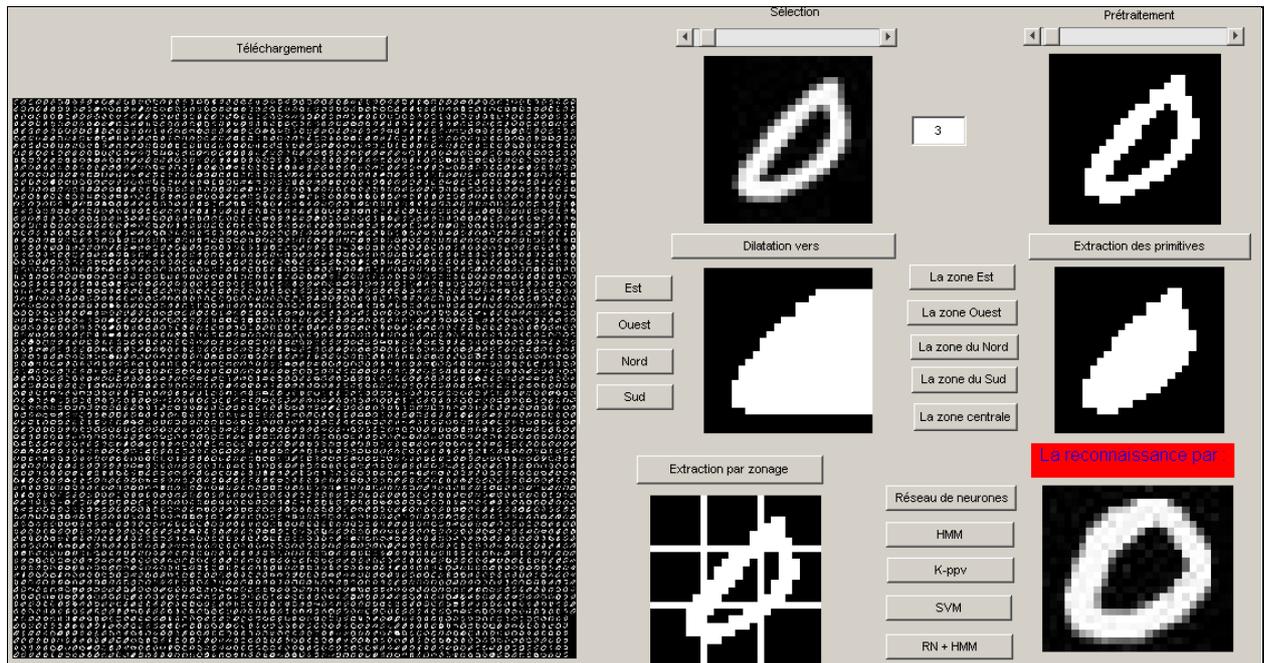


Figure V.67: Interface graphique mise en place pour reconnaître les chiffres Arabes

V.4.1.4 Reconnaissance des chiffres Arabes cursifs

Pour bien mettre en valeur cette reconnaissance, La finalité prévue de celle-ci est de pouvoir comparer entre les performances des deux méthodes d'apprentissage-classification qui sont le perceptron multicouches et les K-plus proches voisins tout en utilisant durant la phase d'extraction des primitives la dilatation morphologique.

Pour ce faire, le contexte de premier système de reconnaissance par le perceptron multicouches est de faire varier le nombre des chiffres d'apprentissage en les donnant les valeurs {1, 10, 20} et pour chacune d'entre elles on calculera le taux de reconnaissance de chaque chiffre τ_{rc} , puis on comparera entre les trois taux obtenus, de même pour le taux de reconnaissance global τ_{rg} c'est-à-dire celui de tous les chiffres.

En revanche, le processus de deuxième système de reconnaissance par la méthode de K-plus proches voisins s'effectuera tout en faisant changer K en l'attribuant les valeurs {6, 10, 15}. Ensuite on va comparer entre les trois taux de reconnaissance τ_{rc} obtenus pour chacune des valeurs de K d'une part ainsi qu'entre les trois τ_{rg} d'autre part.

Finalement on procédera de comparer entre le taux de reconnaissance τ_{rc} le plus élevé obtenu par le premier système de reconnaissance et celui obtenu par le deuxième système.

On fera la même chose pour le taux de reconnaissance global τ_{rg} .

Après avoir mis en œuvre ces deux systèmes de reconnaissance on a obtenu les résultats présentés dans le tableau suivant :

Tableau V.XI: Le taux de reconnaissance de chaque chiffre cursif par PMC et K-ppv

Chiffres	PMC			K - ppv		
	Nombre des chiffres			τ_c x % Pour K= 6	τ_c x % Pour K= 10	τ_c x % Pour K= 15
	τ_c x % Pour 1 chiffre	τ_c x % Pour 10 chiffres	τ_c x % Pour 20 chiffres			
0	78.00	100.00	100.00	89.00	91.00	99.00
1	87.00	97.00	97.00	91.00	92.00	96.00
2	20.00	89.00	95.00	82.00	86.00	92.00
3	80.00	51.00	79.00	74.00	79.00	81.00
4	96.00	63.00	94.00	82.00	86.00	86.00
5	40.00	93.00	93.00	77.00	79.00	89.00
6	47.00	92.00	94.00	65.00	70.00	76.00
7	82.00	82.00	98.00	55.00	64.00	74.00
8	35.00	45.00	52.00	64.00	75.00	73.00
9	46.00	85.00	91.00	58.00	61.00	78.00
τ_g x %	57.22	79.70	89.20	73.70	78.30	84.40

– Les graphes associés sont :

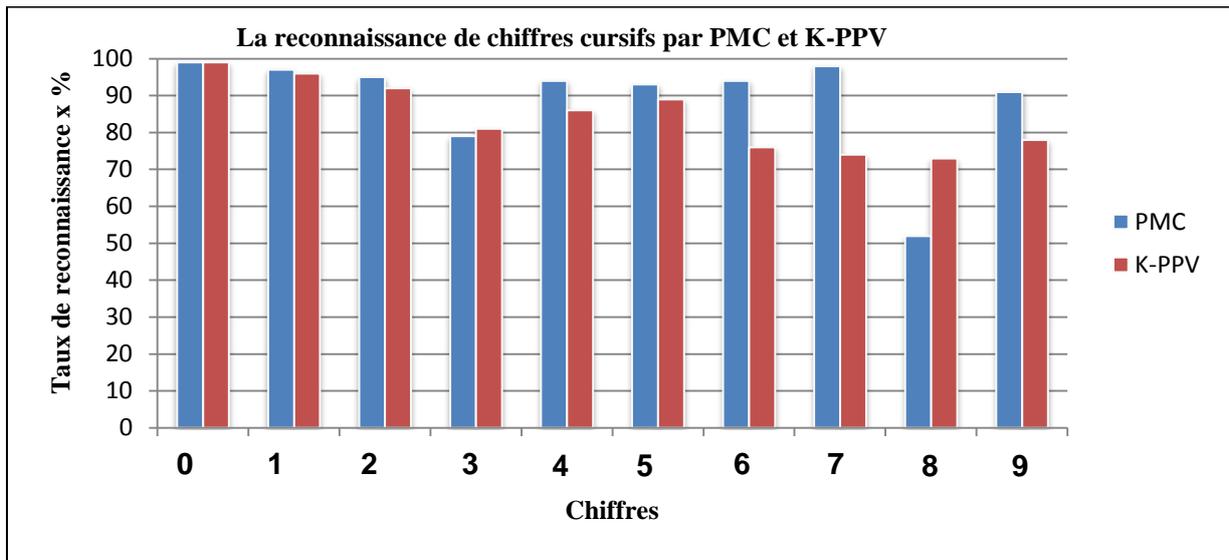


Figure V.68: La reconnaissance de chiffres cursifs par PMC et K-PPV

- **Analyse et commentaire :**

Vu les résultats obtenus d'un côté lorsqu'on a opté un perceptron multicouche, on pourra déduire que le taux de reconnaissance que ce soit de chaque chiffre τ_{rc} ou global τ_{rg} le plus élevé est celui fourni quand on a utilisé 20 chiffres. Et de l'autre côté lorsqu'on a adopté la méthode de K-ppv le taux τ_{rc} ainsi que τ_{rg} les plus élevés sont ceux donnés quand K vaut 15.

D'autre part le τ_{rc} ainsi que τ_{rg} donné par un perceptron multicouche tout en utilisant 20 chiffres est en général plus grand ou égale à celui fourni par les K-ppv sachant que K est fixé à la valeur 15. On constate en fin de compte que le perceptron multicouche est plus performant que les K-ppv.

Les deux systèmes de reconnaissance sont modélisés par l'interface graphique suivante :

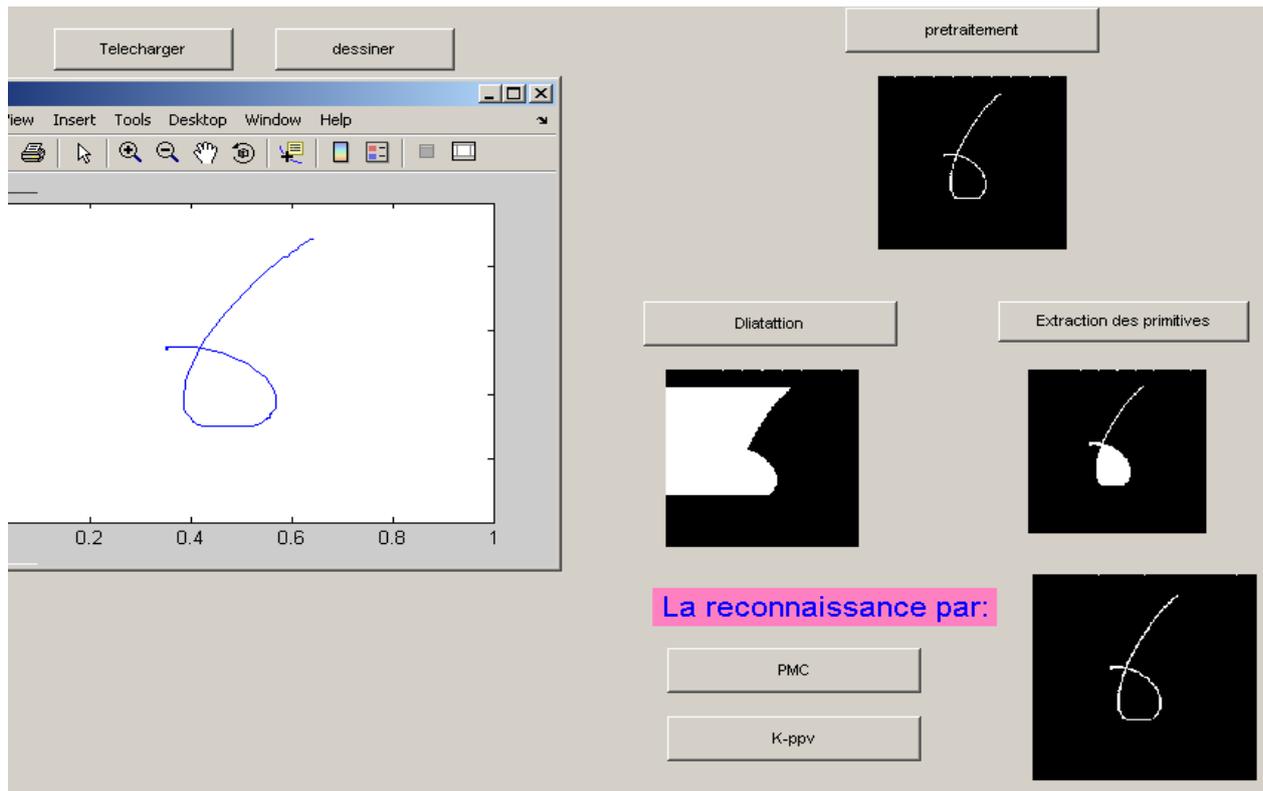


Figure V.69: Interface graphique mise en place pour reconnaître les chiffres cursifs

V.4.2 Reconnaissance des caractères Tifinaghs isolés

V.4.2.1 Reconnaissance des caractères Tifinaghs imprimés et manuscrits

V.4.2.1.1 Reconnaissance par le PMC

Notre finalité envisagée dans ce système de reconnaissance est de comparer entre le taux de reconnaissance global τ_{rg} des caractères manuscrits et celui des caractères imprimés tout en optant les données suivantes :

- ◆ L'extraction des primitives a été effectuée par la méthode de dilatation (la morphologie mathématique).
- ◆ L'apprentissage – classification a été fait pour :
 - Le nombre des vecteurs d'entrée est égal à 18 dont chacun d'eux ayant 7 composantes.
 - Le nombre de neurones de la couche d'entrée est égal à 7 neurones.
 - Le nombre de neurones de la couche cachée est égal à 7 neurones.
 - Le nombre de neurones de la couche de sortie est égal à 18 neurones.
 - Le coefficient d'apprentissage η est fixé à la valeur 0.9.

- La fonction d'activation utilisée est la fonction sigmoïde $F(x) = \frac{1}{1+e^{-x}}$
- L'algorithme adopté est celui de retro-propagation du gradient.

Tableau V.XII: Le taux de reconnaissance de plusieurs ensembles des caractères par PMC

Ensembles des caractères	Nombre des caractères	$\tau_{rg} \times \%$ pour une base de caractères de test manuscrits	$\tau_{rg} \times \%$ pour une base de caractères de test imprimés
1 Ensemble	18	62.43	60.20
10 Ensembles	180	78.88	60.36
20 Ensembles	360	81.57	61.86
30 Ensembles	540	81.75	62.46
40 Ensembles	720	84.23	63.83
50 Ensembles	900	84.91	63.88
Numbers of images for test		2340 images	2340 images

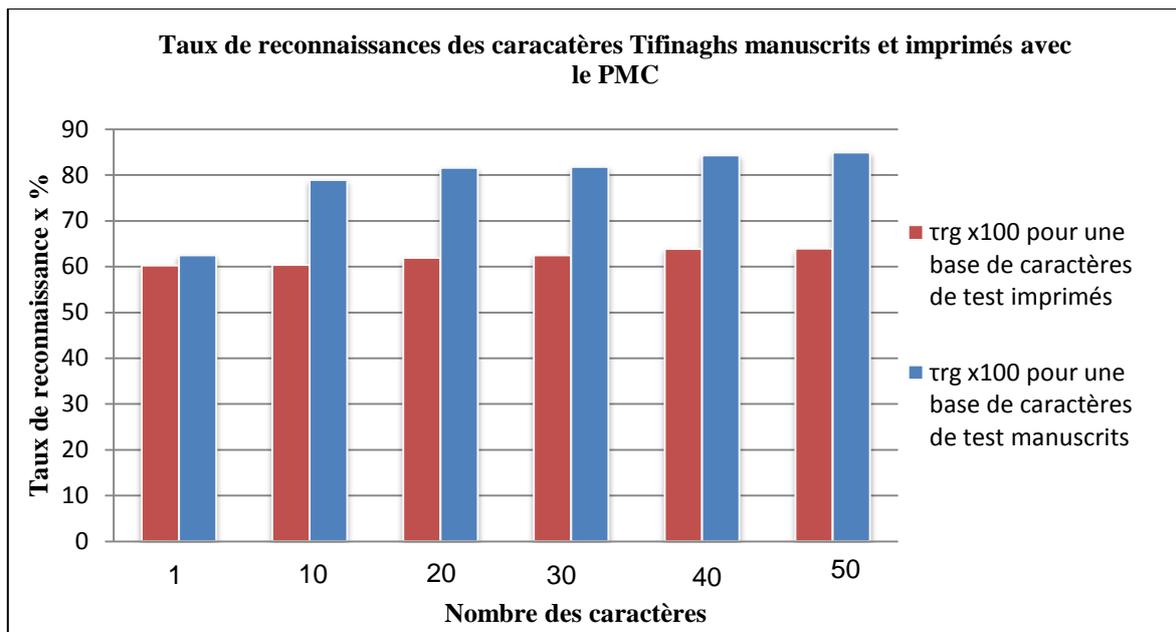


Figure V.70: Taux de reconnaissances des caractères Tifinaghs manuscrits et imprimés avec le PMC

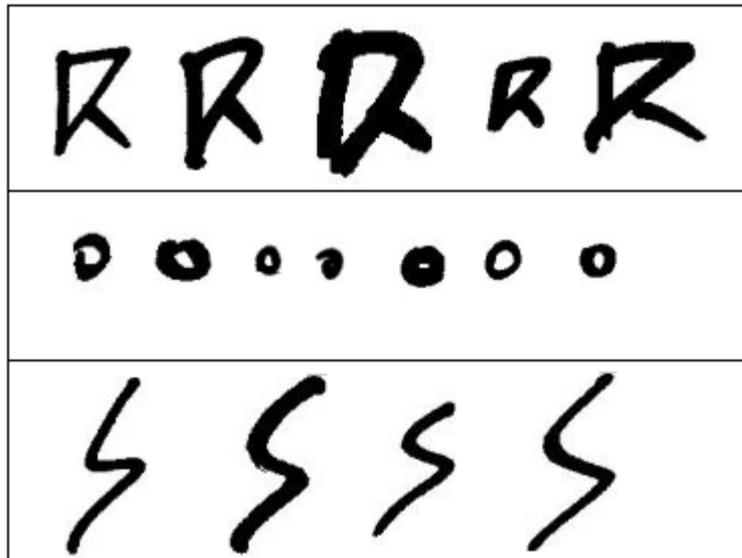


Figure V.71 : Images extraites de la base des caractères Tifinaghs manuscrits



Figure V.72 : Images extraites de la base des caractères Tifinaghs imprimés

- **Analyse et commentaire :**

Compte tenu des résultats issus de ces deux systèmes de reconnaissance, on constate que d'une part lorsqu'on augmente le nombre des ensembles des caractères on aura également une augmentation du taux de reconnaissance global que ce soit des caractères manuscrits ou imprimés.

D'autre part, on remarque que le taux de reconnaissance des caractères manuscrits est plus grand que celui de ceux imprimés, dans ce contexte-là ce résultat est habituellement insolite. Mais dans notre cas il est intuitivement acceptable pensant que bien que les caractères de la

première base sont manuscrits (voir figure V.71), chacun d'entre eux n'est pas trop différent à ses homologues en termes de taille, du scripteur d'écriture et du squelette ce qui permettra d'avoir une extraction très précise de ses primitives par la méthode de morphologie mathématique, et ce qui rendra par la suite sa reconnaissance assez régulière. Tandis que les caractères de la deuxième base même s'ils sont imprimés (voir figure V.72), chacun d'eux est tout à fait différent à ses homologues au niveau de la taille, du scripteur d'écriture et du squelette ce qui engendrera une mauvaise reconnaissance à cause d'un problème d'extraction des primitives par la méthode de morphologie mathématique. Tout cela vérifie bien la défection de cette méthode quand il s'agit de reconnaître des caractères, imprimés ou manuscrits, dont chacun d'eux est très différent à ses homologues.

V.4.2.1.2 Reconnaissance par le PMC et MMC

On se propose de réaliser une comparaison entre les performances des trois systèmes de reconnaissance qui sont le perceptron multicouche et le modèle de Markov caché puis une hybridation entre les deux dans la reconnaissance des caractères imprimés puis manuscrits.

- *Cas des caractères imprimés.*

Les résultats obtenus sont cités dans ce tableau :

Tableau XIII: Les taux de reconnaissance τ_{rc} et τ_{rg} des caractères imprimés pour les trois méthodes MMC, PMC et PMC+MMC

Caractères	τ_{rc} x % de PMC	τ_{rc} x % de MMC	τ_{rc} x % de PMC+MMC
	77,01	85,00	94,71
	67,06	86,00	88,24
	68,82	64,71	65,29
	66,47	81,41	70,00
	53,53	82,41	73,53
	78,24	83,00	74,12
	52,94	89,24	72,35

ⵏ	89,00	88,82	92,00
ⵍ	59,41	72,00	77,65
ⵑ	76,41	71,00	88,24
ⵑ	77,06	88,82	87,00
ⵏ	65,29	88,82	76,00
ⵍ	99,00	89,00	89,00
ⵏ	69,00	73,00	98,24
ⵏ	50,00	88,00	62,35
ⵏ	67,06	58,64	76,47
ⵏ	65,00	72,35	82,35
ⵑ	78,82	87,00	98,82
$\tau_{rg} \text{ x } \%$	63,83	77,00	87,77

- Le graphe associé est :

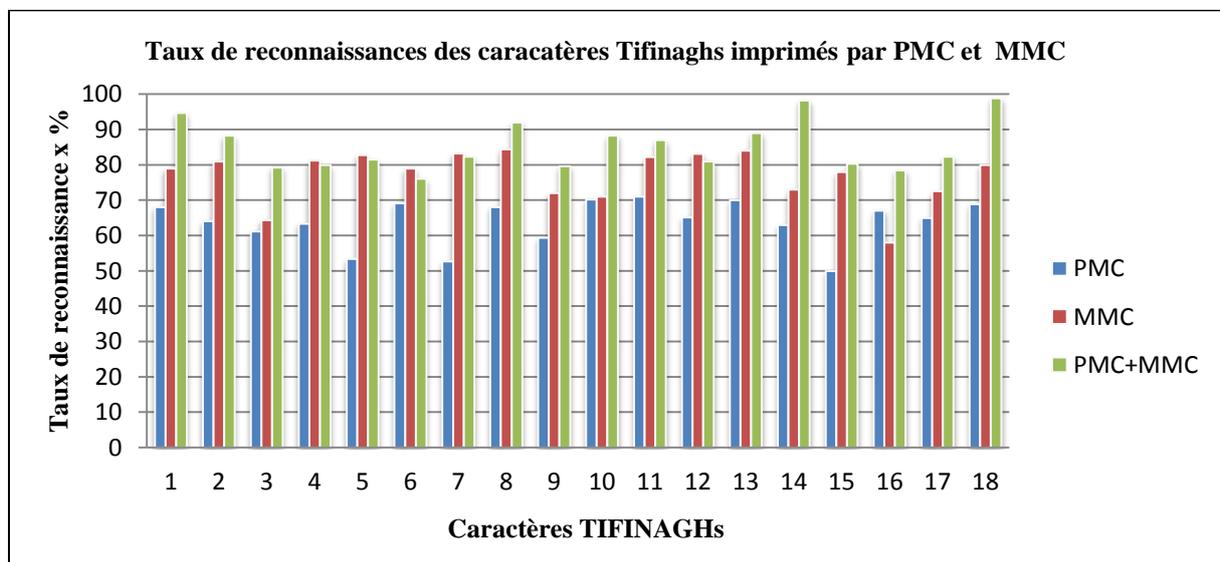


Figure V.73 : Les taux de reconnaissance des caractères Tifinaghs imprimés par le PMC, MMC et PMC+ MMC

▪ **Cas des caractères manuscrits.**

On présente les résultats acquis dans le tableau ci-dessous :

Tableau XIV: Les taux de reconnaissance τ_{rc} et τ_{rg} des caractères manuscrits pour les trois méthodes MMC, PMC et PMC+MMC

Caractères	τ_{rc} x % de PMC	τ_{rc} x % de MMC	τ_{rc} x % de PMC+MMC
R	92,31	100	100
O	93,08	98,46	99,00
X	94,92	93,00	95,00
^	97,00	100	100
H	95,98	100	100
X	68,15	100	100
I	63,77	100	100
S	80,00	87,92	89,00
E	84,00	100	100
Q	100	100	100
F	72,00	83,08	85,00
^	75,25	59,85	60,00
B	85,00	88,15	86,00
E	0100	0100	0100
H	87,00	89,00	90,00
I	92,69	92,31	91,00
H	79,00	100	100
O	67,00	69,23	69,00
τ_{rg} x %	84.91	92.22	92.30

- Le graphe associé est :

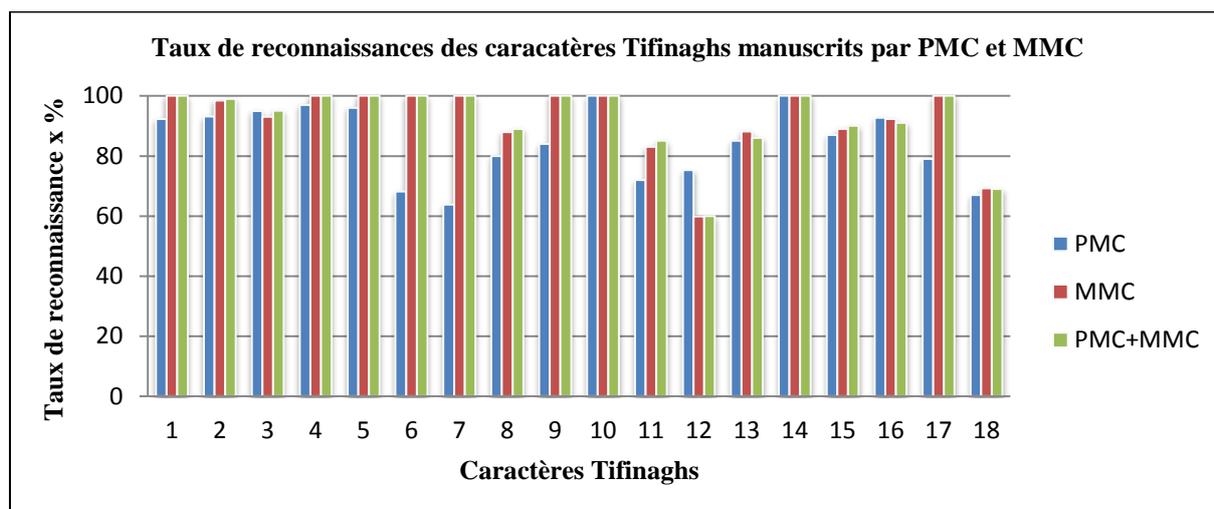


Figure V.74: Le taux de reconnaissance des caractères Tifinaghs manuscrits par le PMC, MMC et PMC+ MMC

- **Analyse et commentaire :**

En tenant compte des résultats présentés ci-dessus on pourra constater que le modèle de Markov caché est plus performant que le perceptron multicouche, et que les deux sont moins performants que le modèle hybride c'est-à-dire qui combine à la fois les deux modèles dans la reconnaissance des caractères Tifinaghs que ce soient imprimés ou manuscrits. Et que ces derniers sont plus reconnaissables que ceux imprimés.

V.4.2.1.3 Reconnaissance des caractères Tifinaghs cursifs

V.4.2.1.3.1 Reconnaissance par le PMC

Notre tendance dans cette reconnaissance est de réaliser un système servant à reconnaître des caractères Tifinaghs cursifs tout en mettant en œuvre dans :

- la phase d'extraction la méthode de Morphologie ainsi que celle hybridée qui combine la Morphologie et Zonage.
- La phase d'apprentissage-classification le perceptron multicouche.

En effet, le but prévu de cette reconnaissance est de pouvoir comparer entre les performances de ces deux méthodes d'extraction tout en faisant varier le nombre des neurones de la couche cachée du perceptron en vue de savoir l'impact de cette variation sur ces performances.

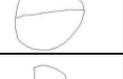
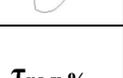
De ce fait, on a opté dans la phase d'apprentissage – classification les données suivantes

- Le nombre des vecteurs d'entrée est égal à 15 dont chacun d'eux ayant 9 composantes.
- Le nombre de neurones de la couche d'entrée est égal à 9 neurones.
- Le nombre de neurones de la couche cachée prend les valeurs suivantes **{8, 9, 10, 11, 12}**.
- Le nombre de neurones de la couche de sortie est égal à 15 neurones.
- Le coefficient d'apprentissage η est fixé à la valeur 0.9.
- La fonction d'activation utilisée est la fonction sigmoïde $F(x) = \frac{1}{1+e^{-x}}$
- L'algorithme adopté est celui de retro-propagation du gradient.

Le tableau présenté ci-dessous englobe tous les résultats obtenus.

Tableau V.XV: Les taux de reconnaissance obtenus par la Morphologie et Morphologie + Zonage par PMC

Chiffres MNIST	Perceptron multicouches									
	$T_{rc} \ x \ %$ par Morphologie					$T_{rc} \ x \ %$ par Morphologie + Zonage				
	Nombre de neurones de la couche cachée					Nombre de neurones de la couche cachée				
	8	9	10	11	12	8	9	10	11	12
	90.00	85.00	80.00	85.00	65.00	90.00	90.00	90.00	100.00	100.00
	80.00	70.00	100.00	95.00	100.00	95.00	100.00	95.00	100.00	100.00
	80.00	65.00	80.00	65.00	65.00	60.00	60.00	80.00	60.00	60.00
	70.00	40.00	15.00	60.00	25.00	10.00	10.00	20.00	25.00	15.00
	70.00	80.00	80.00	75.00	50.00	100.00	100.00	100.00	90.00	100.00
	85.00	85.00	85.00	85.00	85.00	70.00	85.00	85.00	75.00	70.00
	95.00	95.00	95.00	80.00	90.00	50.00	75.00	90.00	90.00	90.00
	95.00	95.00	85.00	95.00	95.00	95.00	95.00	95.00	95.00	95.00
	80.00	45.00	40.00	25.00	60.00	10.00	90.00	50.00	45.00	25.00
	35.00	35.00	35.00	40.00	35.00	40.00	40.00	40.00	50.00	45.00

	45.00	50.00	35.00	45.00	45.00	45.00	40.00	35.00	45.00	70.00
	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
	95.00	90.00	95.00	95.00	90.00	65.00	95.00	95.00	95.00	95.00
	60.00	60.00	55.00	60.00	55.00	70.00	60.00	65.00	60.00	60.00
	95.00	95.00	95.00	95.00	95.00	95.00	95.00	95.00	95.00	95.00
$\tau_{rc} \times \%$	73.44	62.75	67.20	68.75	65.94	62.20	70.94	65.56	70.32	75.63

Et les graphes correspondants sont :

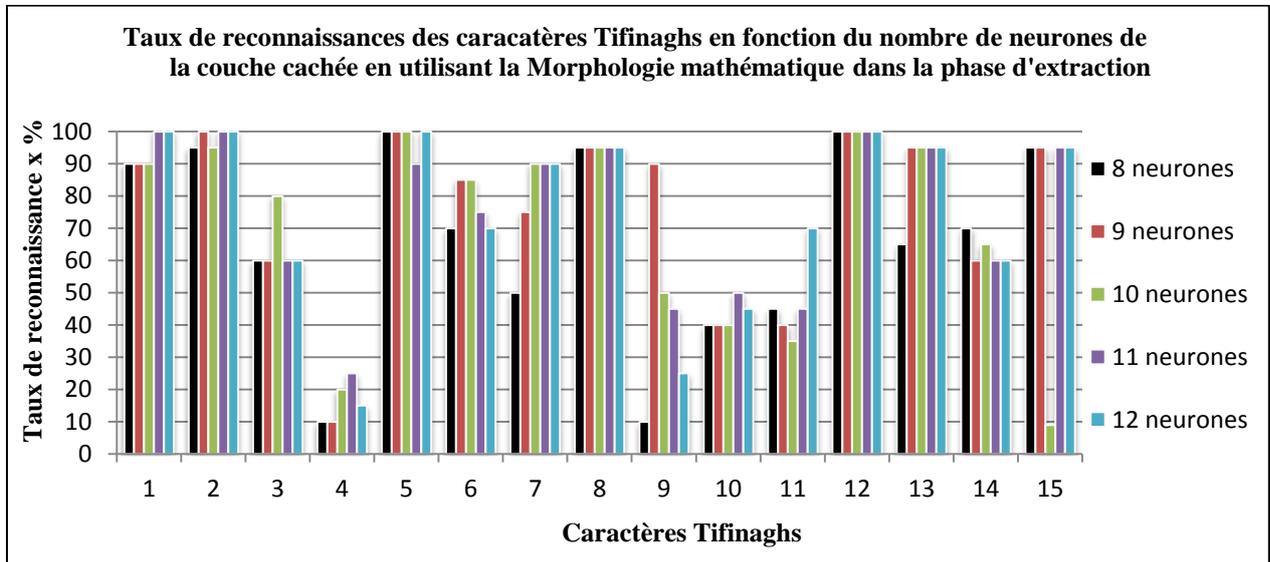


Figure V.75: Les taux de la reconnaissance des caractères Tifinaghs cursifs par la méthode de Morphologie en fonction du nombre des neurones.

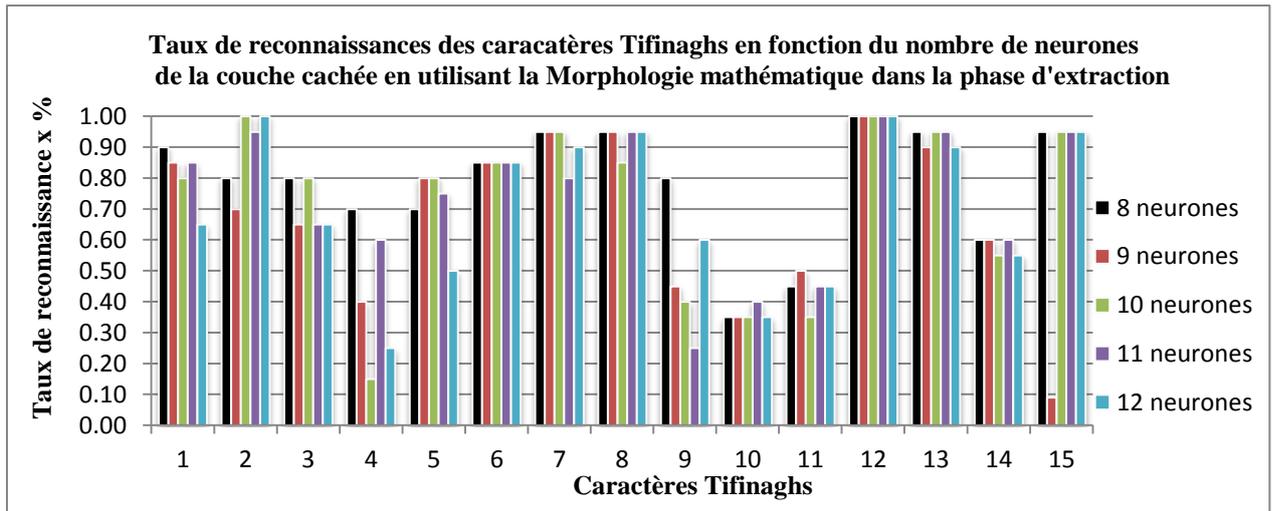


Figure V.76: Les taux de la reconnaissance des caractères Tifinaghs cursifs par la méthode de Morphologie + Zonage en fonction du nombre des neurones.

- **Analyse et commentaire :**

Après avoir analysé les valeurs des taux de reconnaissance τ_{rc} et τ_{rg} acquises, on déduit que la méthode de Morphologie + Zonage est plus efficace que celle qui se restreint uniquement sur la Morphologie. Et quand en faisant augmenter le nombre des neurones de la couche cachée ceci n'implique pas forcément l'augmentation des valeurs de ces taux.

V.4.2.1.3.2 Reconnaissance par les supports vecteurs machines

Le but visé durant cette reconnaissance est de pouvoir comparer en terme des performances entre différentes méthodes d'extraction des primitives qui se sont le Zonage, la Morphologie et la méthode hybride combinant les deux, tout en fait appel dans la phase d'apprentissage – classification dont la Fonction noyau adoptée est celle de RBF d'écart type $\sigma = 0.9$, et la stratégie de multi-classification utilisée est celle d'un contre tous.

Les valeurs issues des taux de reconnaissance de chaque caractère ainsi que celui global sont regroupés dans le tableau suivant :

Tableau V.XVI: Les taux de reconnaissance obtenus par la Morphologie et Morphologie + Zonage par SVM

Caractères Tifinaghs	SVM		
	τ_{rc} (en %) par Zonage	τ_{rc} (en %) par Morphologie	τ_{rc} (en %) par Morphologie + Zonage
	60.00	56.00	80.00
	65.00	90.00	70.00
	55.00	58.00	60.00
	50.00	54.00	50.00
	50.00	50.00	55.00
	70.00	55.00	80.00
	90.00	62.00	90.00
	70.00	51.00	98.00
	70.00	50.00	95.00
	50.00	40.00	56.00
	50.00	45.00	60.00
	80.00	92.31	93.00
	90.00	62.00	92.00
	60.00	60.00	60.00
	60.00	56.00	54.00
$\tau_{rc} \times \%$	63.33	58.75	72.86

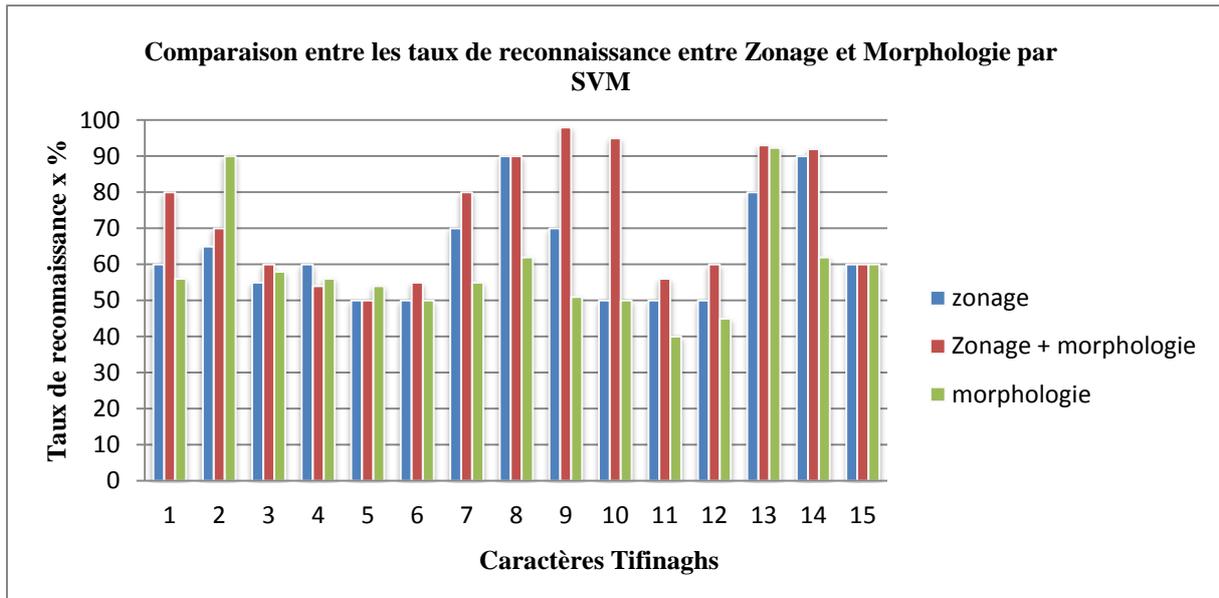


Figure V.77 : Les taux de la reconnaissance des caractères Tifinaghs cursifs par la méthode de Morphologie + Zonage par SVM.

- **Analyse et commentaire :**

En prenant en compte les résultats obtenus après avoir mis en œuvre de ce système de reconnaissance, on pourra effectivement constater que la méthode hybride Morphologie + Zonage est celle qui est la plus performante suivie de celle du Zonage puis celle de Morphologie.

En revanche, en vue de donner en fait un plus à ces deux systèmes de reconnaissance, on présente l'interface graphique suivante qui englobe toutes les phases des deux systèmes.

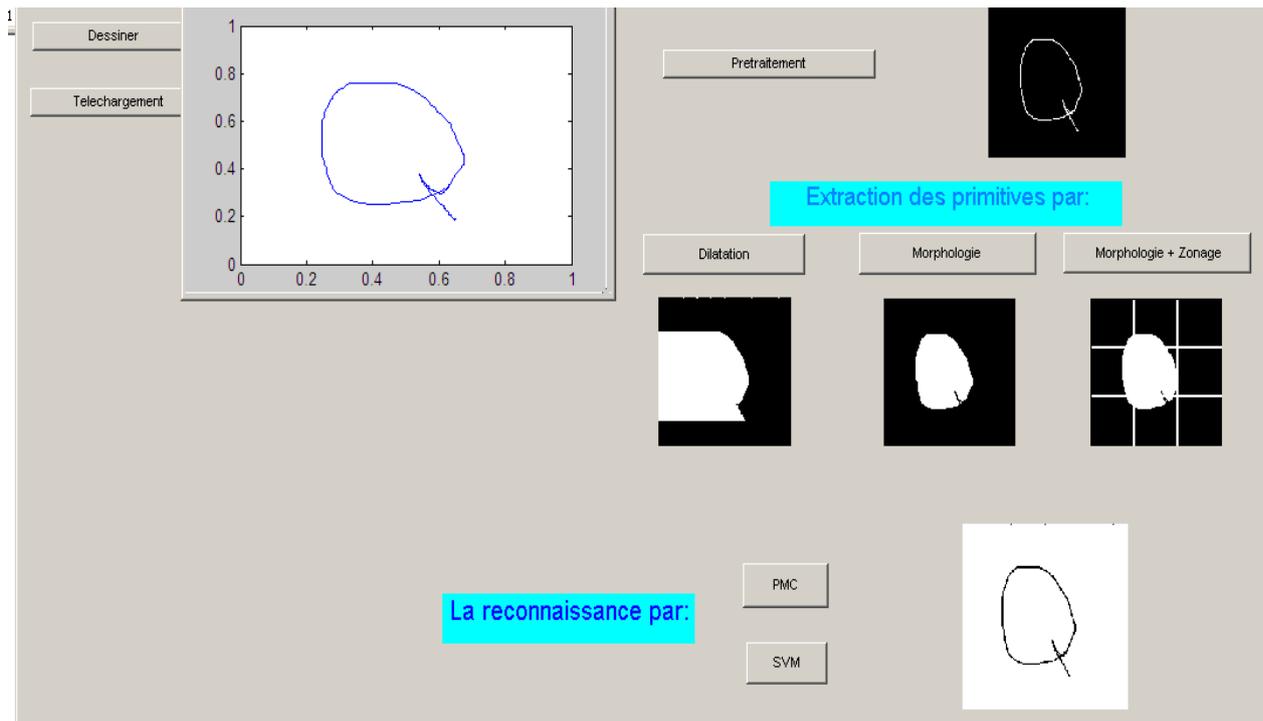


Figure V.78 : Interface graphique mise en place pour reconnaître les caractères Tifinaghs cursifs

V.4.3 Reconnaissance des textes Tifinaghs

Afin de diversifier nos travaux qui s'intéressent aux caractères Tifinaghs, on adopte à mettre en place un système de reconnaissance de ces caractères mais cette fois-ci ceux qui ne sont pas isolés c'est-à-dire des textes tout en utilisant dans la phase d'extraction la Morphologie et dans la phase d'apprentissage-classification un perceptron multicouche dont les données restent les mêmes que celles du celui utilisé dans la reconnaissance des caractères Tifinaghs cursifs sauf que le nombre des neurones de la couche cachée a été fixé à la valeur 9.

Pour simplifier la complexité de cette reconnaissance, on procède à segmenter chaque texte formé d'un ensemble de caractères de telle manière à bien séparer chaque caractère à celui qui se suivra, puis on fait reconnaître chacun de ces caractères au lieu de texte complet.

On se propose de comparer entre le taux de reconnaissance des textes écrits par un stylo et celui des textes écrits par un marqueur.

C'est pour cela on a choisi deux bases de textes de test dont la première contient 11000 caractères écrits en stylo tandis que la deuxième comprend 11000 caractères écrits en marqueur.

✚ La segmentation en Ligne

Dans le contexte de faire segmenter en ligne le texte Tifinagh présenté dans la figure ci-dessous, on est tenu d'abord de compter le nombre de pixels noirs dans chaque ligne, puis on procèdera de représenter graphiquement ces nombres calculés en fonctions des indices des lignes. En effet, ce graphe n'est rien d'autres que l'histogramme de l'image (voir Figure V.80). Ensuite celui-ci s'exploitera pour pouvoir subdiviser l'image originale en plusieurs sous images pour lesquelles chacune d'entre elles englobant exactement un ensemble de caractères qui sont distribués tout à fait horizontalement (voir Figures V.81-85).

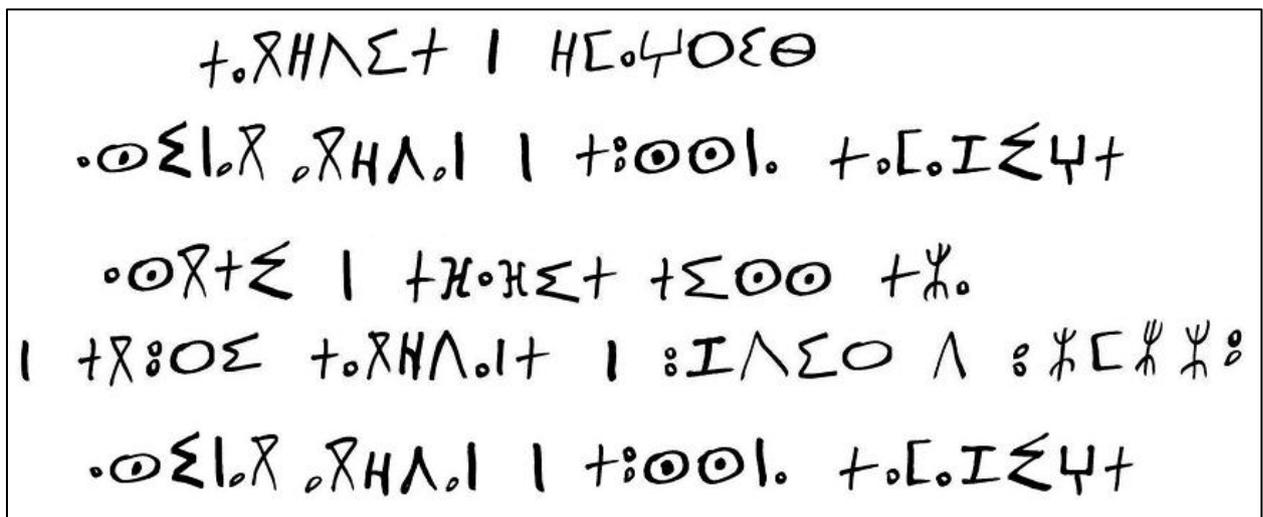


Figure V.79: Exemple d'un texte Tifinagh

La segmentation en colonne

Quand il s'agit de ce type de segmentation on suivra les mêmes démarches qu'on a fait dans le cas précédant, en fait, il nous suffira de remplacer les lignes de l'image par les colonnes (voir Figures V.86-88).



Figure V.86: La segmentation en colonne de la troisième ligne

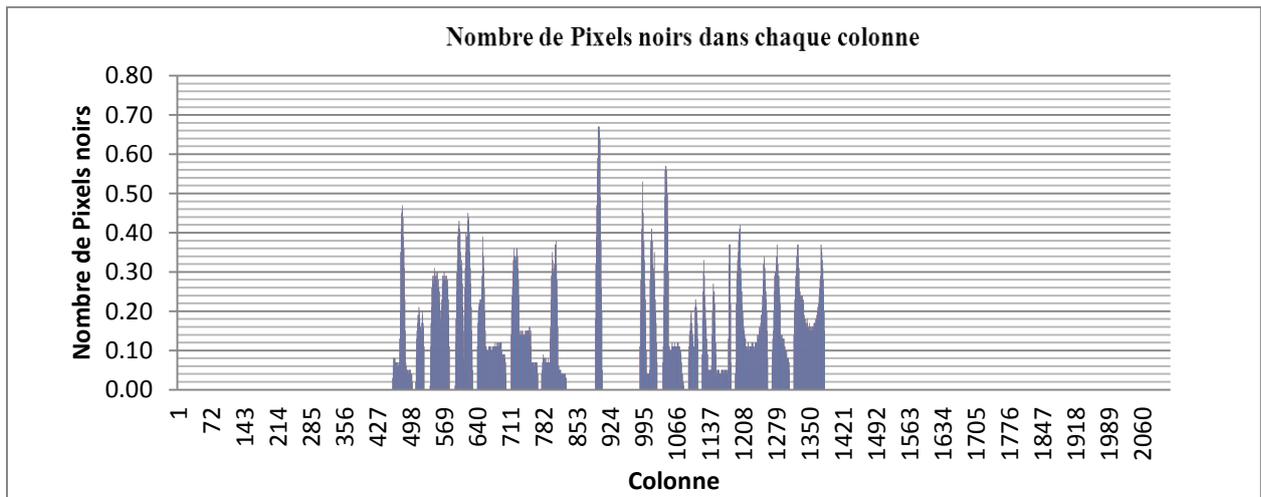


Figure V.87: Représentation graphique de la segmentation en colonne de la troisième ligne



Figure V.88: Caractères segmentés en colonne

Afin de bien mettre en valeur le processus de segmentation, on a élaboré l'interface graphique présentée ci-dessous et qui regroupe d'une manière trop significative toutes les démarches suivies dans ce processus.

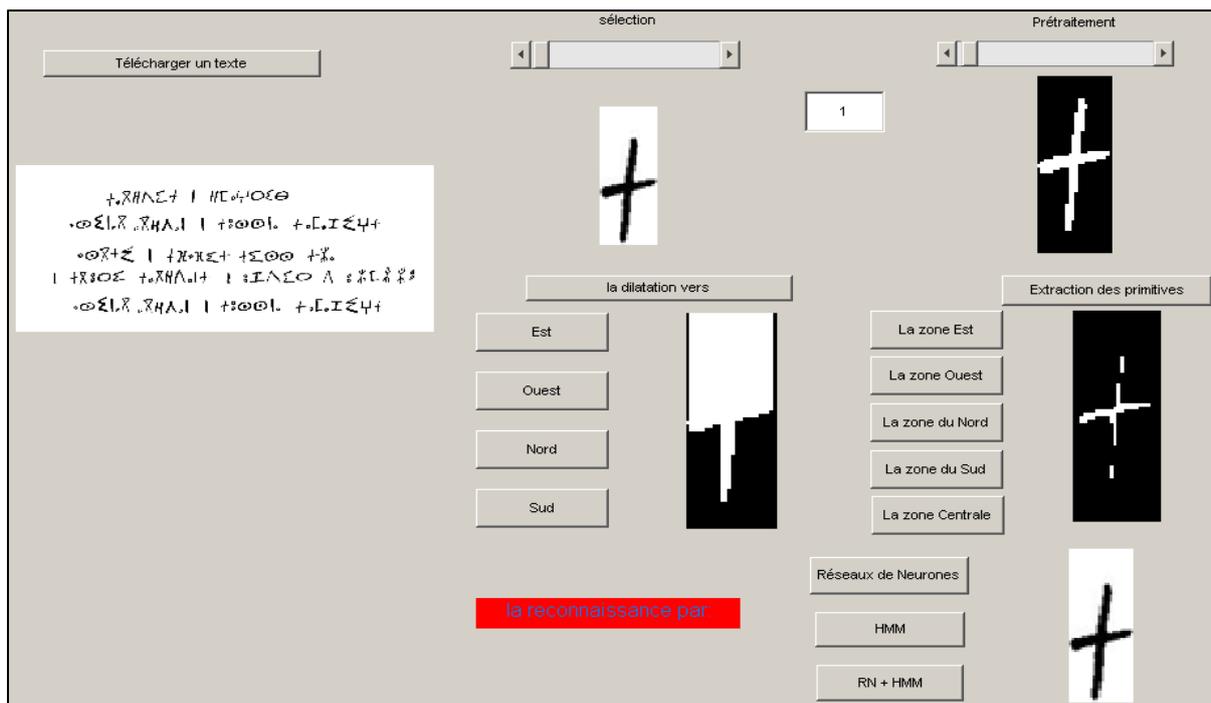


Figure V.89: Interface graphique mise en place pour reconnaître un texte Tifinagh

Tableau V.XVII: Le taux d'apparition des caractères Tifinaghs

ⵏ	01.19	ⵉ	00.38	ⵓ	00.49	ⵙ	02.80	ⵔ	00.95	ⵖ	00.62
ⵏ	00.89	ⵓ	07.41	ⵓ	09.77	ⵓ	05.56	ⵓ	02.04	ⵓ	00.89
ⵏ	00.90	ⵓ	00.89	ⵓ	15.28	ⵓ	00.89	ⵓ	11.58	ⵓ	09.08
ⵓ	03.71	ⵓ	06.30	ⵓ	00.23	ⵓ	03.55	ⵓ	01.72	ⵓ	02.70
ⵓ	00.27	ⵓ	00.83	ⵓ	00.30	ⵓ	01.13	ⵓ	00.35	ⵓ	05.38
ⵓ		00.89		ⵓ		00.11		ⵓ		00.03	

❖ Quand le texte est écrit par un marqueur, on regroupe tous les résultats obtenus dans le tableau suivant :

Tableau V.XVIII: Le Taux de reconnaissance d'un texte Tifinagh écrit par un marqueur

Perceptron multicouches			
Caractère	$\tau_{rc} \text{ x } \%$	Caractère	$\tau_{rc} \text{ x } \%$
ⵛ	62.38	ⵝ	74.62
ⵛ	82.28	ⵞ	71.02
ⵛ	79.06	ⵟ	73.51
ⵛ	70.16	ⵠ	82.15
ⵛ	76.00	ⵡ	65.00
ⵛ	77.51	ⵢ	70.00
ⵛ	75.63	ⵣ	72.53
ⵛ	73.72	ⵤ	70.93
ⵛ	74.21	ⵥ	70.00
ⵛ	77.00	ⵦ	80.60
ⵛ	80.13	ⵧ	73.68
ⵛ	65.34	⵨	64.44
ⵛ	85.79	⵩	70.00
ⵛ	74.62	⵪	85.94
ⵛ	80.21	⵫	73.24
$\tau_{rg} \text{ x } \%$	74.38		

- Le graphe associé est :

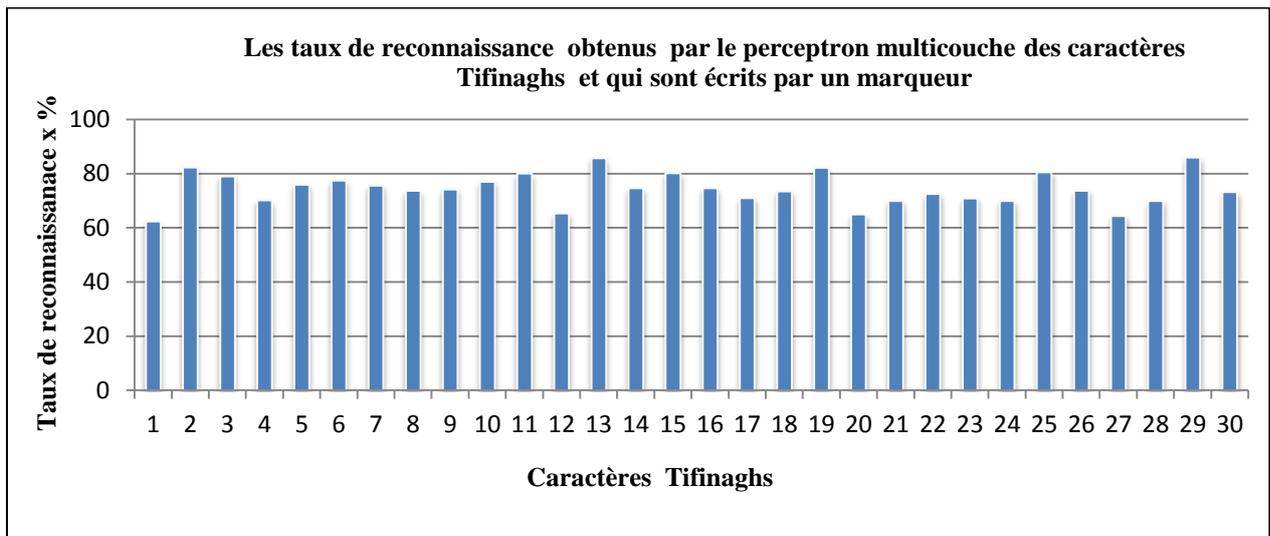


Figure V.90: Taux de reconnaissance d'un texte Tifinagh écrit par un marqueur

❖ Dans le cas où le texte est écrit par un stylo, tous les résultats obtenus sont présentés dans le tableau suivant :

Tableau V.XIX: Le Taux de reconnaissance d'un texte Tifinaghs écrit par un stylo

Perceptron multicouches			
Caractère	$\tau_{rc} x \%$	Caractère	$\tau_{rc} x \%$
ⵛ	56.33	ⵝ	70.38
ⵏ	70.59	ⵞ	79.00
ⵍ	73.33	ⵟ	77.00
ⵍ	70.95	ⵠ	71.74
ⵍ	73.06	ⵡ	71.00
ⵍ	70.00	ⵢ	73.25
ⵍ	76.00	ⵣ	74.72

H	79.00	ⵇ	74.08
ⵇ	69.43	ⵉ	73.19
I	72.19	ⵏ	78.56
O	72.10	ⵓ	70.82
Q	61.47	ⵔ	69.44
Y	84.23	ⵕ	74.14
ⵇ	67.34	ⵖ	82.39
E	71.00	ⵗ	77.96
$\tau_{rg} \times \%$	72.81		

- Le graphe associé est :

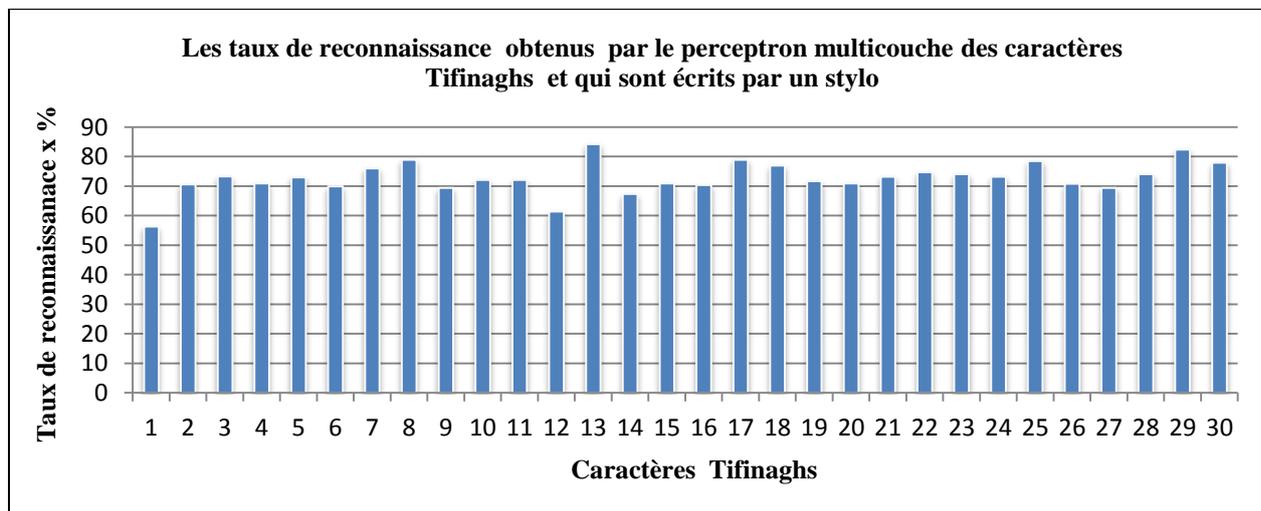


Figure V.91: Taux de reconnaissance d'un texte Tifinaghs écrit par un stylo

- **Analyse et commentaire :**

Vu les valeurs des taux de reconnaissance obtenues, on conclut que les textes qui sont écrits par un marqueur sont plus reconnaissables que ceux qui sont écrits par un stylo

V.5 CONCLUSION

Dans ce chapitre, on a présenté les divers systèmes de reconnaissance qu'on a réalisé incluant ceux intéressés aux caractères Tifinaghs englobant ce qui sont imprimés, manuscrits, et cursifs ainsi que des textes manuscrits. Et ceux qui sont concernés aux chiffres Arabes manuscrits et cursifs. La raison pour laquelle on a tenté de diversifier ces systèmes est de pouvoir comparer entre leurs performances.

En effet, on a utilisé plusieurs techniques de prétraitement telles que le seuillage, le centrage, la normalisation, la rotation, l'amincissement et la segmentation, et des nombreuses méthodes d'extraction des primitives à savoir la Morphologie, le Zonage et le Zig-Zag, et également une multitude des méthodes d'apprentissage-classification qui sont le perceptron multicouche, les supports vecteurs machines, les K-plus proches voisins et le modèle de Markov caché .

Les finalités ciblées ainsi que les objectifs visés est de bien positiver et mieux mettre en valeur ces systèmes, c'est pourquoi on a tenté de les rendre trop sophistiqués et trop fiables. Ceci a été accessible quand on a pu faire hybrider entre ces différentes méthodes que ce soient au niveau d'extraction des primitives ou au plan d'apprentissage-classification. Justement en vertu de ces hybridations, on a obtenu de bons résultats en terme du taux de reconnaissance malgré qu'elles consomment beaucoup de temps afin qu'elles puissent s'exécuter par rapport aux méthodes non hybrides.

CHAPITRE VI : CONCLUSION GÉNÉRALE ET PERSPECTIVES

Contenu du chapitre

<i>VJ.1</i>	<i>Conclusion</i>	<i>145</i>
<i>VJ.2</i>	<i>Perspectives</i>	<i>148</i>

VII.1 Conclusion

Le travail présenté dans cette thèse s'intéresse à deux types de reconnaissance de caractères : la reconnaissance des chiffres Arabes manuscrits ainsi que cursifs et la reconnaissance des caractères Tifinaghs y inclus imprimés, manuscrits, cursifs et des textes. D'une manière récapitulative, on tente à ce moment de donner une brève idée significative expliquant le pourquoi ainsi que le comment du déroulement de tout ce qu'on a accompli durant cette thèse. Dans ce contexte-là, on a bien pensé en principe qu'en vue de bien opter une stratégie concrète qui nous permettra de mettre en réalité nos objectifs envisagés, on a cru que celle-ci doit avoir d'abord un scénario se caractérisant par une philosophie qui s'articulera extrêmement sur les éléments fondamentaux suivants :

- Les motivations.
- Les objectifs prévus.
- Les problématiques.
- Les solutions (systèmes simples).
- Comparaison entre les résultats obtenus et ceux prévus.
- Résultats probants.
- Amélioration des solutions (systèmes hybrides).
- Solutions trop probants.

En effet, on a d'abord débuté ce mémoire par un extrait des motivations et des différents facteurs qui nous ont énormément encouragés à choisir le domaine de la reconnaissance des formes comme sujet de thèse. Ensuite, on a déterminé nos objectifs prévus et nos finalités ciblées puisque cette détermination nous permettra de passer de vouloir au pouvoir. Néanmoins, ce passage est accessible à condition que si on a mis sous lumière en premier temps toutes les problématiques qui empêcheront nos objectifs d'être opérationnels. Parmi les diverses difficultés rencontrées, il faut citer que la reconnaissance du caractère manuscrit reste

un défi à relever vis-à-vis du cas imprimé. Ces complications sont dues à la multitude de possibilités d'écrire manuellement un tel même caractère par un ou plusieurs personnes.

En outre, la détection de ces problématiques signifie qu'il faut trouver soigneusement des remèdes appropriées et des solutions adéquates. On a opté en première phase à des solutions simples qui visent à mettre en place des systèmes de reconnaissance simples (non hybrides). Précisément, on a proposé plusieurs systèmes de reconnaissances en utilisant d'une part différentes méthodes d'extraction des primitives à savoir la morphologie mathématique, le zonage et le Zig-Zag. D'autre part, en faisant appel à de nombreuses méthodes d'apprentissage-classification qui sont divisées en deux catégories principales supervisée et non supervisée. Parmi celles de la première catégorie on a opté le perceptron multicouche, les supports vecteurs machines et les K-plus proches voisins. Tandis qu'en deuxième catégorie, on s'est focalisé au modèle de Markov caché.

En pratique, ces derniers systèmes simples ont fourni des résultats d'une façon ou d'une autre moins probants vis-à-vis de ce qu'on souhaitera avoir en fin de compte. Ceci nous a poussés à proposer d'autres solutions plus satisfaisantes, ainsi on a essayé de réaliser des combinaisons entre les différentes méthodes qui interviennent dans les trois phases de construction d'un système de reconnaissance. Effectivement, les systèmes de reconnaissance hybrides réalisés ont donné des résultats plus probants.

Également, afin de mettre en valeur ce travail, on a fait une comparaison entre les performances de tous les systèmes réalisés avec des commentaires et des analyses des résultats obtenus, exprimés à la fin de ce mémoire.

Notons aussi qu'on a choisi la base de données des chiffres Arabes manuscrits MNIST et quatre bases de données des caractères Tifinaghs : la première englobe ceux imprimés, la deuxième contient ceux manuscrits, tandis que la troisième comprend ceux qui sont cursifs, finalement la dernière base contient des textes.

En conclusion, les études que nous avons menées englobent des mises en évidence, en place, en œuvre, finalement en action de plusieurs systèmes de reconnaissance de caractères, simples ou hybrides, avec des analyses et des commentaires fournies à la fin de la thèse. On en déduit que les méthodes hybrides que ce soient au niveau d'extraction des primitives ou au niveau d'apprentissage-classification sont celles qui donnent les résultats les plus probants.

Pour être bien précis concernant tous les travaux effectués durant cette thèse. On est tenu de passer à la langue des chiffres par ce qu'elle est tellement significative, autrement dit on

parlera de ces travaux selon un point de vue statistique. En effet, on a pu mettre en œuvre en total 29 systèmes de reconnaissance qui sont répartis comme suit :

- 10 systèmes réalisés par PMC.
- 2 systèmes réalisés par MMC.
- 9 systèmes réalisés par SVM.
- 6 systèmes réalisés par K-PPV.
- 2 systèmes réalisés par PMC/MMC.

Signalons que tous ces systèmes sont accompagnés par :

- 19 Tableaux regroupant tous les résultats obtenus.
- 16 Représentations graphiques.
- 10 Interfaces graphiques modélisant les systèmes de reconnaissance.

VII.2 Perspectives

En perspectives, on ambitionnera de promouvoir nos acquis, et de bien renforcer nos potentiels dans le domaine de reconnaissance des formes. Dans ce sens-là, nous voudrions réaliser des systèmes de reconnaissance plus développés et plus fiables permettant d'obtenir des taux de reconnaissance très probants et dans des durées raisonnables.

En fait, quand il s'agira d'augmenter la fiabilité des systèmes ceci sera accessible par la mise en œuvre de nouvelles hybridations que ce soient au niveau des techniques de prétraitement ou des méthodes d'extraction des primitives ou bien au niveau d'apprentissage-classification. Pour la minimisation de temps de reconnaissance, elle peut être faisable en subdivisant un programme de reconnaissance en plusieurs sous programmes rapidement exécutables et utiliser ensuite les techniques de parallélisme informatique.

Dans ce contexte, on peut fait appel à des modèles statistiques comme les moments invariants notamment ceux de Krawtchouk, de Hahn et de Zernike, ainsi que quelques transformées mathématiques particulièrement celles de Radon, de Fourier-Mellin, etc. D'autre part, dans le cadre des méthodes d'apprentissage-classification hybrides, on pense utiliser de nouvelles combinaisons à titre d'exemples K-PPV/MMC, SVM/MMC et PMC/K-PPV.

Finalement, nous envisagerons aussi d'élargir l'application de ces systèmes de reconnaissance à d'autres caractères : Arabes, Latins, Grecques, Indiens, etc. Ainsi que la reconnaissance en ligne des textes Tifinaghs.

Bibliographie

- [1] E. Achtert, C. Böhm, J. David, P. Kröger, and A. Zimek. Global Correlation Clustering Based on the Hough Transform. *Statistical Analysis and Data Mining*, vol 1(3), pp. 111-127, 2008.
- [2] M. Adankon and M. Cheriet. Model selection for the ls-svm. Application to Handwriting. *Pattern Recognition*, Volume 42, Issue 12, Pages 3264-3270, December 2009.
- [3] K. Addakiri and M. Bahaj. On-line Handwritten Arabic Character Recognition using Artificial Neural Network. *International Journal of Computer Applications*, volume 55, number 13, 2012.
- [4] A. Aggarwal. Handwritten Devanagari Characters Recognition using Gradient Features *IJARCSSE*, page 85-90, Vol. 2 Issue 5 May 2012.
- [5] A. R. Ahmad, C. Viard-Gaudin, M. Khalid, and R. Yusof. Online Handwriting Recognition using Support Vector Machine. *Proceedings of the Second International Conference on Artificial Intelligence in Engineering & Technology*, Kota Kinabalu, Sabah, Malaysia, August 3-5 2004.
- [6] P. Ahmed and Y. Al-Ohali. Arabic Character Recognition: Progress and Challenges. *Journal of King Saud University - Computer and Information Sciences*. Volume 12, Pages 85–116, 2000.
- [7] S. Al-Ma'adeed. Recognition of off-line handwritten Arabic words using neural network. *Proceeding of GMAI'06, International Conference on Geometric Modeling and Imaging*, pp. 141-114, London, England, July 2006.
- [8] S. Alma'adeed, C. Higgins, and D. Elliman. Off-line recognition of handwritten Arabic words using multiple hidden Markov models. *Knowledge-Based Systems*, vol. 17, pp. 75-79, 2004.
- [9] S. Al-Ma'adeed, C. Higgins, and D. Elliman. Recognition of off-line handwritten Arabic words using hidden Markov model approach. *Proceedings of ICPR'02, 16th International Conference on Pattern Recognition*, Vol. 3, pp. 481-484, Quebec City, Canada, August 2002.
- [10] S. Alma'adeed. Recognition of Off-Line Handwritten Arabic Words Using Neural Network, *proc. of the Geometric Modeling and Imaging - New Trends*, 2006.
- [11] A. Al khateeb, J. Ren, J. Jiang, and H. Al-muhtaseb. Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking. *Pattern Recognition Letters archive volume 32 Issue 8*, Pages 1081-1088, June, 2011.

- [12] J. H. Al khateeb, O. Pauplin, J. Ren, J. Jiang. Performance of hidden Markov model and dynamic Bayesian network classifiers on handwritten Arabic word recognition Knowledge-Based Systems recognition, Volume 24, Issue 5, Pages 680-688, July 2011.
- [13] F. Álvaro, J. A. Sánchez, J. M. Benedí Recognition of on-line handwritten mathematical expressions 2D stochastic context-free grammars and hidden Markov. Pattern Recognition Letters, Volume 35, Pages 58-67, 1 January 2014.
- [14] M. Ameer, A. Bouhjar, F. Boukhris, A. Boukous, A. Boumalk, M. Elmedlaoui, E. Iazzi, and H. Souifi. Initiation à la langue Amazighe. Publications de l'IRCAM, CAL, Rabat, 2004.
- [15] A. Amin and S. Fischer. A document skew detection method using the hough transform. Pattern Anal. Appl., vol. 3, no. 3, pp. 243-253, 2000.
- [16] M. Amrouch, Y. Es-Saady, A. Rachidi, M. El Yassa, and D. Mammas. A Novel Feature Set for Recognition of Printed Amazigh Text using Maximum Deviation and HMM. International Journal of Computer Applications (0975 – 8887), Volume 44– No12, April 2012.
- [17] M. Amrouch, Y. Es-Saady, M. El Yassa, and D. Mammas. Handwritten Amazigh character recognition system based on continuous HMMs and directional features. International Journal of Modern Engineering Research, vol. 2, 2, pp. 436-441, 2012.
- [18] J. Angulo and J. Serra. Automatic analysis of DNA microray images using mathematical morphology. Bioinformatics, vol 19, no 5, pp. 553-562, Mar 2003.
- [19] B. Anuradhasrinivas, A. Agarwal, and R. Rao. An overview of OCR research in indian scripts, International Journal of Computer Science and Engineering System, vol. 2, 2, pp. 141–153, April 2008.
- [20] S. Arora, D. Bhattacharjee, M. Nasipuri, L. Malik, M. Kundu, and D. K. Basu. Performance Comparison of SVM and ANN for Handwritten Devnagari Character Recognition, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 6, May 2010.
- [21] J. Arnaud. Affective Computing: Apport des Processus Emotionnels aux Systèmes Artificiels. Laboratoire GRAPPA, Université de Lille III – Charles de Gaulle, 2003.
- [22] D. Arrivault, N. Richard, C. Fernandez-Maloigne, and P. Bouyer. Collaboration between statistical and structural approaches for old handwritten characters recognition. In Lecture Notes on COMPUTER SCIENCE, éditeur : 5th IAPR International Workshop, Graph Based Representation in Pattern Recognition, Poitiers, France, volume 3434, pages 291–300. Springer, Mars 2005.
- [23] V. Athitsos and S. Sclaroff, Boosting Nearest Neighbor Classifiers for Multiclass Recognition, IEEE Workshop on Learning in Computer Vision and Pattern Recognition, June 2005.

- [24] E. Augustin. Reconnaissance de mots manuscrits par systèmes hybrides Réseaux de Neurones et Modèles de Markov Cachés, these de doctorat, Université RENE DESCARTES - PARIS V, 2001.
- [25] N.E. Ayat, M. Cheriet, and C.Y. Suen. Empirical error based optimization of svm kernels. Application to digit image recognition. In Proceedings of the International Workshop on Handwriting Recognition, Niagara on the lake, Canada, 2002.
- [26] S. Baboo, P. Subashini, and M. Krishnaveni. Combining Self-Organizing Maps and Radial Basis Function Networks for Tamil handwritten Character Recognition. ICGST-GVIP Journal, vol. 9, Issue 4, 2009.
- [27] S. Bag, G. Harit, and P. Bhowmick. Recognition of Bangala compound characters using structural. Pattern Recognition, Volume 47, Issue 3, March 2014, Pages 1187-1201.
- [28] L. Baoli, Y. Shiwen, and L. Qin. An Improved k-Nearest Neighbor Algorithm for Text Categorization. Proceedings of the 20th International Conference on Computer Processing of Oriental Languages, Shenyang, China, 2003.
- [29] N.P. Banashree, D. Andhre, R. Vasanta, and P.S. Satyanarayana. OCR for script identification of Hindi (Devnagari) numerals using error diffusion Halftoning Algorithm with neural classifier, Proceedings of World Academy of Science Engineering and Technology 20 (2007) 46–50.
- [30] T. Barbu. A pattern recognition approach to image segmentation. Proceedings of the romanian academy. Series A. Volume 4. Number 2, 2003.
- [31] S. Barve. Optical Character Recognition Using Artificial Neural Network. International Journal of Advanced Research in Computer Engineering & Technology", Vol. 1, No. 4, 2012.
- [32] N. Ben Amara, A. Belaïd, and N. Ellouze. Utilisation des modèles Markoviens en reconnaissance de l'écriture arabe état de l'art. Colloque International Francophone sur l'Ecrit et le Document (CIFED'00), Lyon, France, pp.181- 191, 2000.
- [33] A. Benouareh, A. Ennaji, and M. Sallami. Semi-continuous HMMs with explicit state duration for unconstrained Arabic word modeling and recognition. Pattern Recognition Letters archive. Volume 29 Issue 12, pages 1742-1752. September, 2008.
- [34] R. Bertolami and H. Bunke. Hidden Markov model-based ensemble methods for offline handwritten text line recognition. Pattern Recognition, Volume 41, Issue 11, November 2008, Pages 3452-3460.
- [35] D. P. Bertsekas and J. N. Tsitsiklis. Gradient convergence in gradient methods with errors. SIAM J. on Optimization, 10(3):627–642, July 1999.
- [36] M. Bicego, E. Pekalska, D. M.J. Tax, R. P.W. Duin. Component-based discriminative classification for hidden Markov models. Pattern Recognition, Volume 42, Issue 11, November 2009, Pages 2637-2648.

- [37] C. M. Bishop, "Pattern recognition and machine learning", Springer, 2006.
- [38] F. Bortolozzi, Jr.A. Britto, L.S. Oliveira, and M. Morita. Document Analysis, chapitre Recent Advances in Handwriting Recognition, pages 1–31. ISBN 8177647849. Umapada Pal et al, 2005.
- [39] L. Bottou. Online algorithms and stochastic approximations. In David Saad, editor, Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK, 1998.
- [40] A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. Journal of Machine Learning Research, 6 :1579-1619, September 2005.
- [41] D. Bouchaffra Embedding HMMs-based models in a Euclidean space : the topological hidden Markov models. Pattern Recognition, Volume 43, Issue 7, July 2010, Pages 2590-2607.
- [42] H. Bronni, J. Iacono, J. Katajainen, P. Morin, J. Morrison, and G. Toussaint. In-place planar convex hull algorithms. In Latin American Theoretical Informatics, LATIN2002, Cancun, Mexico, pages 494–507, 2002.
- [43] H. Byun and S. Lee. Applications of Support Vector Machines for Pattern Recognition: A Survey. In the Proceedings of the First International Workshop: Pattern Recognition with Support Vector Machines, pp. 213-236, Niagara Falls, Canada 2002.
- [44] F. Camastra A SVM-based cursive character. Pattern Recognition, Volume 40, Issue 12, December 2007, Pages 3721-3727.
- [45] P. R. Cavalin, R. Sabourin, C. Y. Suen, and A. S. Britto Evaluation of incremental learning algorithms for HMM in the recognition of alphanumeric characters. Pattern Recognition, Volume 42, Issue 12, December 2009, Pages 3241-3253.
- [46] A. Chalabi. MT-Based Transparent Arabization of the Internet TARJIM.COM”, In White, J.S. (Ed) AMTA Springer: Verlag Berlin Heidelberg, pp. 189-191, 2000.
- [47] O. Chapelle. Training a support vector machine in the primal. Neural computation, 19(5) : 1155-1178, 2007.
- [48] O. Chapelle, V. Vapnik, O. Bousquet and S. Mukherjee. Choosing Multiple Parameters for Support Vector Machines. Machine Learning 46 (1-3): 131-159, 2002.
- [49] B. B. Chaudhuri and U. Bhattacharya. Efficient training and improved performance of multilayer perceptron in pattern classification. Neurocomputing, vol. 34, no. 1–4, 2000, pp. 11–27.
- [50] Z. Cheng and Y. Ma. A research about pattern recognition of control chart using probability neural network. In: Proc. ISECS, 2008, pp. 140-145.
- [51] C. Choisy and A. Belaid. Couplage d’une vision locale par HMM et globale par RN pour la reconnaissance de mots manuscrits pages 345-354, CIFED, 2002.

- [52] A. Choudhary, R. Rishi, S. Ahlawat. A New Character Segmentation Approach for Off-Line Cursive Handwritten Words. *Procedia Computer Science*, Volume 17, 2013, Pages 88-95.
- [53] A. Choudhary, R. Rishi, S. Ahlawat. Off –line Handwritten Character Recognition Using Features Extracted from Binarization Technique. *AASRI Procedia*, Volume 4, 2013, Pages 306-312.
- [54] S. Connell, A.K. Jain, Writer adaptation of on-line handwriting models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2): 329-342, 2002.
- [55] A. Cornuéjols and L. Miclet. *Apprentissages artificiels, Concept et algorithmes*, Eyrolles. 2002.
- [56] K. Daoudi, D. Fohr, and C. Antoine. Continuous Multiband Speech Recognition using Bayesian Networks *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Trento Italy, December 2001.
- [57] R. L. Das, B. K. Prasad, and G. Sanyal. HMM based Offline Handwritten Writer Independent English Character Recognition using Global and Local Feature Extraction. *International Journal of Computer Applications (0975 – 8887)* volume 46 – No.10, May 2012.
- [58] N. Das, J. M. Reddy, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, D.K. Basu. A statistical - topological feature combination for recognition of handwritten numerals *Applied Soft Computing*, Volume 12, Issue 8, August 2012, Pages 2486-2495.
- [59] R. Dass and S. Asht. Handwritten Numeral Pattern Recognition Using Neural Network *International Journal of Computer Science and Telecommunications*. Volume 3, Issue 9, September 2012.
- [60] F. Delbos and J.Ch. Gilbert. Global linear convergence of an augmented Lagrangian algorithm for solving convex quadratic optimization problems. *Journal of Convex Analysis*, 12, 45-69. 2005.
- [61] A. A. Desai. Gujarati handwritten numeral optical character reorganization through neural network. *Pattern Recognition* 43, 2010, 2582-2589.
- [62] P. Dianne, O'Leary, *Linear and nonlinear Conjugate gradient-related Methods*, AMS-SIAM, 1996, « Conjugate gradient and related KMP algorithms : the Beginnings ».
- [63] J. J. Ding, P. Y. Lin, and H. H. Chen. Generalized zigzag scanning algorithm for non-square blocks. *Springer-Verlag Berlin, Heidelberg*, pp. 252-262, 2011.
- [64] T. M. T. Do and T. Artières. Learning mixture models with support vector machines for Sequence classification and segmentation. *Pattern Recognition*, Volume 42, Issue 12, December 2009, Pages 3224-3230.

- [65] C. Domeniconi, J. Peng, and D. Gunopulos. Locally adaptive metric nearest-neighbor classification. *PAMI*, 24(9):1281– 1285, 2002.
- [66] H. Drucker, D. Wu, and V. Vapnik. Support vector machines for spam categorization. *IEEE transactions on Neural Networks*, 10 5, 1999, pp. 1048-1055.
- [67] F. Einsele, R. Ingold, and J. Hennebert. A HMM-Based Approach to Recognize Ultra Low Resolution Anti-Aliased Words. In *Second International Conference on Pattern Recognition and Machine Intelligence (PREMI 2007)*, volume 4815 de LNCS, pages 511–518, Kolkata, Inde, Décembre 2007. Springer.
- [68] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel. Arabic Hand- writing Recognition Using Baseline Dependent Features and Hidden Markov Modeling. *Proc. Eighth Intl Conf. Document Analysis and Recognition*, pp. 893-897, 2005.
- [69] B. El kessab, C. Daoui, B. Bouikhalene, M. Fakir, and K.Moro. Extraction Method of Handwritten Digit Recognition Tested on the MNIST Database. *International Journal of Advanced Science and Technology* Vol. 50, January, 2013.
- [70] B. El kessab, C. Daoui, B. Bouikhalene, M. Fakir, and K.Moro. Handwritten Tifinagh Text Recognition using Neural Networks and Hidden Markov Models, *International Journal of Computer Applications (0975 – 8887) Volume 75– No.18, August 2013.*
- [71] B. El kessab, B. Bouikhalene, C. Daoui, M. Fakir, and K.Moro. Utilisation des Réseaux de Neurones pour la Reconnaissance des Caractères Tifinaghe Manuscrits. *Mediterranean Telecommunication Journal*, vo.2, n°1, janvier 2012.
- [72] B. El kessab, C. Daoui, K.Moro, B. Bouikhalene, and M. Fakir. Recognition of Handwritten Tifinagh Characters Using a Multilayer Neural Networks and Hidden Markov Model, *Global Journal of Computer Science and Technology* Volume 11 Issue 15 September 2011.
- [73] Y. Es Saady, A. Rachidi, M. El Yassa, and D. Mammass. Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character ». *International Journal of Advanced Science and Technology* Vol. 33, August, 2011.
- [74] L. A. F. Fernandes and M. M. Oliveira. Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognition*, Elsevier, Volume 41, Issue 1, pp. 299–314, 2008.
- [75] G. D. Forney. The Viterbi Algorithm. *Proceedings of IEEE*, Vol. 61, N°. 3, pp.268-278, March 1973.
- [76] L. M. Fu. Analysis of the dimensionality of neural networks for pattern recognition. *Pattern Recognition*, Volume 23, Issue10, 1990, Pages 1131-1140.
- [77] K. Fukushima. Recognition of partly occluded patterns: A neural network model. *Biol. Cybernet.*, vol. 84, 2001, pp. 251–259.

- [78] B. Gabrys and A. Bargiela. General fuzzy min–max neural network for clustering and classification. *IEEE Trans. Neural Networks*, vol. 11, no. 3, 2000, pp. 769–783.
- [79] W. Gao, G. Fang, D. Zhao, Y. Chen. A chinese sign language recognition system based on SOFM /SRN/HMM. *Pattern Recognition*, Volume 37, Issue 12, December 2004, Pages 2389-2402
- [80] Q. B. Gao and Z. Z. Wang. Center-based nearest neighbor classifier. *Pattern Recognition*, Volume 40, Issue 1, January 2007, Pages 346-349.
- [81] C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11): 1408-1423, 2004.
- [82] N. Garg, L. Kaur and M. K. Jindal. Segmentation of Handwritten Hindi Text. *International Journal of Computer Applications (IJCA)*, Vol. 1, No. 4, pp.22-26, 2010.
- [83] N. Garg, L. Kaur and M. K. Jindal. The Segmentation of Half Characters in Handwritten Hindi Text. *Proceedings of the ICISIL 2011*, Springer, pp.48-53, 2011.
- [84] N. K.Garg, L. Kaur, M. Jindal. Recognition of Offline andwritten Hindi Text Using SVM, *International Journal of Image Processing (IJIP)*, Volume (7) : Issue (4) : 2013.
- [85] G. Giacinto, F. Roli, and L. Bruzzone. Combination of neural and statistical algorithms for supervised classification of remote-sensing images, *Pattern Recognition Letters*, 21, pp 385-397, 2000.
- [86] L. Gong, C. Liu, Y. Li, and F.Yuan. Training Feed-forward Neural Networks Using the Gradient Descent Method with the Optimal Stepsize. *Journal of Computational Information Systems* 8: 4, 2012, pp 1359–1371.
- [87] H.P. Graf, E. Cosatto, L. Bottou, I. Durdanovic, and V. Vapnik. Parallel Support Vector Machines: The Cascade SVM. In Lawrence K. Saul, Y. Weiss, and L. Bottou, editors, *Neural Information Processing Systems*, pages 521-528, Cambridge, MA, 2004. MIT Press.
- [88] Y. Guermeur, A. Eliseeff, and H. PaugamMoisy. A new multiclass svm based on a uniform convergence result. *IJCNN*, vol. IV, pages 183–188, 2000.
- [89] K. Hallouli, L. Likforman-Sulem, and M. Sigelle. A comparative study between decision fusion and fusion data in Markovian printed character recognition In 16th International Conference on Pattern Recognition Quebec, Canada, August 2002.
- [90] X. He and P. Shi. An Efficient Iris Segmentation Method for Recognition, pp 120-126, 2005. Springer Berlin Heidelberg.
- [91] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. New York: Wiley and Sons. ISBN 9780471367277, 1949.

- [92] R. S. Hegadi. Recognition of Printed Kannada Numerals based on Zoning Method. International Journal of Computer Applications (0975 – 8878) on National Conference of Advanced Computing and Communications - NCACC, April 2012.
- [93] A. Hennig and N. Sherkat. Exploiting zoning based on approximating splines in cursive script recognition. Pattern Recognition, Volume 35, Issue 2, February 2002, Pages 445-454.
- [94] C. Hou, F. Nie, C. Zhang, D. Yi, and Y. Wu. Multiple rank multi-linear SVM for matrix data classification. Pattern Recognition, Volume 47, Issue 1, January 2014, Pages 454-469.
- [95] C. Huang, H. Ai, Y. Li, and S. Lao. High-performance rotation invariant multiview face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(4): 671-686, 2007.
- [96] D. Impedovo and G. Pirlo. Zoning methods for handwritten character recognition: A survey. Pattern Recognition, 47(3):969–981, 2014.
- [97] S. Impedovo, F.M. Mangini, and D. BarbuZZi A novel prototype generation technique for handwritten digit. Pattern Recognition, Volume 47, Issue 3, March 2014, Pages 1002-1010.
- [98] M. Iwanowski and M. Swierez. Pattern Recognition Using Morphological Class Distribution Functions and Classification Trees. Springer, pp. 143-154, 2011.
- [99] A.K. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1): 4-37, 2000.
- [100] A. Jindal, R. Dhir, and R. Rani. Diagonal features and SVM classifier for handwritten Gurumukhi character recognition”, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, 5, pp. 505-508, 2012.
- [101] J. John, K.V. Pramod, and K. Balakrishnan. Unconstrained Handwritten Malayam Character Recognition using Wavelet Transform and Support vector Machine. Procedia Engineering, Volume30, 2012, Pages 598-605.
- [102] J. Jomy, K. V. Pramod, and B. Kannan. Handwritten Character Recognition of SouthIndian Scripts: A Review. Computer Vision and Pattern Recognition, 2011.
- [103] C. Jou and H. C. Lee Handwritten numeral recognition based on simplified structural classification and fuzzy memberships. Expert Systems with Applications, Volume 36, Issue 9, November 2009, Pages 11858-11863.
- [104] M. Kamel and A. Campilho. An Iris Recognition Method Based On Zigzag Collarett Area and Asymmetrical Support Vector Machines. ICIAR 2007, pp. 854-865, 2007.
- [105] M. Khorsheed. Recognising handwritten Arabic manuscripts using a single hidden Markov model. Pattern Recognition Letters, vol. 24, pp. 2235-2242, 2003.

- [106] J.B. Kim, S.C. Kee, and J.Y. Kim. Fast detection of multiview face and eye based on cascaded classifier, *Machine vision and Applications*, 116-119, 2005.
- [107] Y. Kimori, E. Katayama, N. Morone, and T. Kodama. Fractal dimension analysis and mathematical morphology of structural changes in actin filaments imaged by electron microscopy. *J Struct Biol* 2011, 176:1-8.
- [108] G. Koch, L. Heutte, and T. Paquet. Automatic extraction of numerical sequences in handwritten incoming mail documents. *Pattern Recognition Letters*, 26(8):1118–1127, juin 2005.
- [109] B. B. Kodada and K. M. Shivakumar. Unconstrained Handwritten Kannada Numeral Recognition, *IJIEE* 2013, Vol.3(2): 230-232.
- [110] M. Kumar, M. K. Jindal, and R. K. Sharma. SVM based offline handwritten Gurmukhi character recognition, proceeding of International Workshop on Soft Computing Applications and Knowledge Discovery, pp. 51-62. 2011.
- [111] E.M. Kussul, and T.N. Baidyk. Rosenblatt perceptrons for handwritten digit recognition, *Proceedings of International Joint Conference on Neural Networks IJCNN 2* 1516-1520, 2001.
- [112] L. Lazli, A. Chebira, M. T. Laskri, and K. Madani. Hybrid HMM/ANN System Using Fuzzy Clustering for Speech and Medical Pattern recognition, pp. 550-557, Springer-verlag Berlin Heidelberg 2011.
- [113] G. Lebrun. Utilisation des pyramides combinatoires pour la détection de manuscrits hiéroglyphiques. *Mémoire de D.E.A., ENSICAEN*, 2005.
- [114] Y. LeCun. THE MNIST handwritten digit database, 1998.
- [115] Y. Le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*. San Mateo, Morgan Kaufmann. 396-404, 1990.
- [116] Y. LeCun, L. Bottou, Y. Bengio, P. Hafner. Gradientbased learning applied to document recognition, *Proceedings of the IEEE* 86 (11) 2278-2344, 1998.
- [117] N. Le Roux, M. W. Schmidt, and F. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *NIPS*, 2012.
- [118] K.C. Leung, C.H. Leung. Recognition of handwritten Chinese characters by critical region analysis. *Pattern Recognition*, Volume 43, Issue 3, March 2010, Pages 949-961
- [119] C. L. Liu, F. Yin, Q. F. Wang, and D. H. Wang. Chinese Handwriting Recognition Competition. *Proc. 11th ICDAR*, Beijing, China, 2011, pp. 1464-1469.

- [120] C. L. Liu, F. Yin, D.H. Wang, and Q. F.Wang. Chinese Handwriting Recognition Contest 2010, Proc. 2010 Chinese Conf. Pattern Recognition (CCPR), Chongqing, 2010.
- [121] J. Liu, P. Gader. Neural networks with enhanced outlier rejection ability for off-line handwritten word. Pattern Recognition, Volume 35, Issue 10, October 2002, Pages 2061-2071.
- [122] L. Li and L. Zhang, J. SU. Handwritten character recognition via direction string and NN matching. Journal of China Universities of Posts and Telecommunications, Volume 19, Supplement 2, October 2012, Pages 160-165,196.
- [123] Y. Li and B. L. Lu. Feature selection based on loss-margin of Nearest neighbor classification. Pattern Recognition, Volume 42, Issue 9, September 2009, Pages 1914-1921.
- [124] Z. Liu, Q. Pan, J. Dezert. A new belief-based K-nearest neighbor classification. Pattern Recognition, Volume 46, Issue 3, March 2013, Pages 834-844.
- [125] C. L. Liu, F. Yin, D. H. Wang, Q. F. Wang. Online and offline handwritten Chinese Character recognition : Benchmarking on new. Pattern Recognition, Volume 46, Issue 1, January 2013, Pages 155-162.
- [126] S.A. Mahmoud and S.O. Olatunji. Automatic recognition of off-line handwritten Arabic (Indian) numerals using support vector and extreme learning machines. Int. Journal of Imaging and Robotics, vol. 2, pp. 34–53, 2009.
- [127] S. A. Mahmoud and S. M. Awaida. Rrecognition of off-line handwritten Arabic (Indian) numerals using multi-scale features and supports vectors machines vs. Hidden Markov models, The Arabian Journal for Science and Engineering, Volume 34, Number 2B October 2009.
- [128] A. Majumdar and B. B. Chaudhuri. A MLP classifier for both printed and handwritten Bangla numeral recognition. Proceeding of Indian Conference on Vision, Graphics and Image Processing, pp. 796-804. 2006.
- [129] S. Manipal, D.H Manjaiah, R. Bera, and H.N. Ashoka. Based Feature Extraction and Statistical Classification Technique for Kannada Handwritten Numeral Recognition, International Journal of Computer Science & Engineering Technology (IJCSSET) 2010.
- [130] W. S. McCulloch and W. Pitts. A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943.
- [131] P. Melin. Modular Neural Networks and Type-2 Fuzzy Systems for Pattern Recognition. Springer Volume 389 2012.
- [132] J. Milgram, R. Sabourin, and M. Cheriet. Combining Model-based and Discriminative Approaches in a Modular Two-stage Classification System: Application to Isolated Handwritten Digit Recognition. Electronic Letters on Computer Vision and Image Analysis, 5(2):1-15, 2005.

- [133] A. Mohamed, K.N. R. Nair. HMM/ANN hybrid model for continuous Malayalam speech recognition. *Procedia Engineering*, Volume 30, 2012, Pages 616-622.
- [134] T. C.Mota and A. Thomé. One-Against-All-Based Multiclass SVM Strategies Applied to Vehicle Plate Character Recognition, *IJCNN*, 2009.
- [135] A. P. Nagare. License Plate Character Recognition System using Neural Network. *International Journal of Computer Applications*, Volume 25, No. 10, July 2011, pp. 36-39.
- [136] P. Natarajan, S. Saleem, R. Prasad, E. MacRostie, and K. Subramanian. Multilingual Off-line Handwriting Recognition Using Hidden Markov Models: A script independent Approach. *Springer Book Chapter on Arabic and Chinese Handwriting Recognition*, ISSN:0302-9743, VOL. 4768, pp. 235-250, March 2008.
- [137] J. M. Nazzal, I. M. EL-Emary, and S. A. Najim. Multilayer preceptron neural network (MLPs) for analyzing the properties of Jordan oil shale. *World Applied Science Journal*, vol. 5, 5, pp. 546-552. 2008.
- [138] X. Niu and C. Y. Suen. Anovel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognition*, Volume 45, Issue 4, April 2012, Pages 1318-1325.
- [139] I. Nouretdinov, S. G. Costafreda, A. Gammerman, A. Chervonenkis, V. Vovk, V. Vapnik, and C. H. Y. Fu. Machine learning classification with confidence: Application of transductive conformal predictors to MRI-based diagnostic and prognostic markers in depression, *NeuroImage*, 56, 809-813, 2011.
- [140] I.S. Oh and C.Y. Suen. A class-modular feed-forward neural network for handwriting recognition, *Pattern Recognition*, 35: 229-244, 2002.
- [141] G. Ososkov. Effective neural network approach to image recognition and control, *Proceedings of International Conference on Physics and Control 1 (2003)* 242-246.
- [142] M. Padmaa and Y. Venkataramani. ZIG-ZAG PVD – A Nontraditional Approach. *International Journal of Computer Applications (0975 – 8887)*, Volume 5– No.7, August 2010.
- [143] M. T. Parvez and S. A. Mahmoud. Arabic handwritten recognition using structural and Syntactic pattern attributes. *Pattern Recognition*, Volume 46, Issue 1, January 2013, Pages 141-154.
- [144] P.M. Patil and T.R. Sontakke Rotation, scale and translation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network. *Pattern Recognition*, Volume 40, Issue 7, July 2007, Pages 2110-2117.
- [145] M. Pechwitz and V. Maergner. HMM Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT- Database”, *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, IEEE Computer Society, 2003, p. 890.

- [146] N. G. Pedrajas and D. O. Boyer. A cooperative constructive method for neural networks for pattern. *Pattern Recognition*, Volume 40, Issue1, January 2007, Pages 80-98.
- [147] E. Poisson. Architecture et Apprentissage d'un Système Hybride Neuro-Markovien pour la Reconnaissance de l'Écriture Manuscrite En-Ligne. Rapport de thèse, Univ de Nantes, 2005.
- [148] J. Pradeep, E. Srinivasan and S. Himavathi. Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network. *International Journal of Computer Science and Information Technology (IJCSIT)*, vol. 3, no. 1, pp. 27-38, Feb 2011.
- [149] J. Pradeep, E. Srinivasan, and S. Himavathi. Diagonal feature extraction based handwritten character system using Neural network", *International Journal of Computer Applications*, vol. 8, 9, pp. 17-22. 2010.
- [150] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. «Section 16.2. Viterbi Decoding ». *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York: Cambridge University Press, 2007.
- [151] K. Roy and P. Bhattacharya. An Iris Recognition Method Based On Zigzag Collarete Area and Asymmetrical Support Vector Machines. *IEEE*, pp. 861-865, 2006.
- [152] T. Rabbani and F. v. d. Heuvel. Efficient hough transform for automatic detection of cylinders in point clouds" in *Proceedings of the 11th Annual Conference of the Advanced School for Computing and Imaging (ASCI '05)*, The Netherlands, June 2005.
- [153] M. Rigamonti, D. Lalanne, F. Evéquo, and R. Ingold. Browsing Multimedia Archives Through Intra- and Multimodal Cross-Documents Links. In *Machine Learning for Multimodal Interaction (MLMI'05)*, volume 3869 de LNCS, pages 114–125. Springer, 2005.
- [154] K. Roy, C. Chaudhuri, M. Kundu, M. Nasipuri, and D. K. Basu. Comparison of the Multi Layer Perceptron and the Nearest Neighbor Classifier for Handwritten Numeral Recognition *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING* 21, 1247-1259, 2005.
- [155] L. Rabiner, and B. Juang, *Fundamentals of speech recognition*. Prentice Hall, 1993.
- [156] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, pp. 257 – 286, 1989.
- [157] A. Rahman and B. Verma. Effet of ensemble classifier composition on offline cursive character. *Information Processing & Management*, Volume 49, Issue 4, July 2013, Pages 852-864.
- [158] A.F.R. Rahman, R. Rahman, and M.C. Fairhurst Recognition of handwritten Bengali characters : a novel multistage approach. *Pattern Recognition*, Volume 35, Issue 5, May 2002, Pages 997-1006.

- [159] S.V. Rajashekararadhya and P.V. Ranjan. Efficient zone based feature extraction algorithm for handwritten numeral recognition of popular south Indian scripts, *Journal of Theoretical and Applied Information Technology* 7 (1) 1171–1180, 2009.
- [160] G. G. Rajput, R. Horakeri, and S. Chandrakant. Printed and Handwritten Mixed Kannada Numerals Recognition Using SVM G.G. Rajput. (IJCSE) *International Journal on Computer Science and Engineering* Vol. 02, No. 05, 2010, 1622-1626.
- [161] E. Ramasso, M. Rombaut, and D. Pellerin. Algorithmes Forward-Backward et décodage de Viterbi dans le cadre du Modèle des Croyances Transférables Application à la classification de séquences vidéos. Colloque GRETSI, 11-14 septembre 2007, Troyes.
- [162] Y. Rodriguez, F. Cardinaux, S. Bengio, and J. Mariethoz. Measuring the performance of face localization systems, *Image and Vision Computing*, 24(8): 882:893, 2006.
- [163] S. H. Rodríguez, J. F. M. Trinidad, J. Ariel C. Ochoa. Fast k most similar neighbor classifier for mixed data (tree k-MSN). *Pattern Recognition*, Volume 43, Issue 3, March 2010, Pages873-886.
- [164] O. F. Romero, B. G. Berdiñas, B. P. Sánchez, and A. A. Betanzos A new convex objective function for thr supervised learning of single-layer neural networks. *Pattern Recognition*, Volume 43, Issue 5, May 2010, Pages 1984-1992.
- [165] F. Rosenblatt. *Principles of Neurodynamics*. Spartan, 1962.
- [166] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [167] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*, 1986.
- [168] M. Sabri Recognition of writer-independent off-line handwritten Arabic (Indian) numerals using hidden Markov models. *Signal Processing*, Volume 88, Issue 4, April 2008, Pages 844-857.
- [169] R. Sadykhov, O. Malenko, A.N. Limovich and M.L. Elinger. Hybrid system for recognition of handwritten symbols on the base of structural methods and neural networks. In *Vision Interface Conference - Minsk - Belarussie*, page 223, 2002.
- [170] S. Sagiroujlu, E. Besdoc and M. Erler. Contro chart pattern recognition using artificial neural networks. *Turkish Journal of Electrical Engineering*, vol. 8, (2000), pp. 137-147.
- [171] N. A. Samsudin and A. P. Bradley Nearest neighbour group-based classification. *Pattern Recognition*, Volume 43, Issue 10, October 2010, Pages 3458-3467.
- [172] M. P. Schambach. Recurrent HMMs and Cursive Handwriting Recognition Graphs. *Proc. 10th Int'l Conf. Document Analysis and Recognition*, pp. 1146-1150, July 2009.

- [173] Y. X. Shao, C. H. Wang, and B. H. Xiao, L. B. Zhang Self-generation Voting Based Method for Handwritten Chinese Character. *Acta Automatica Sinica*, Volume 39, Issue 4, April 2013, Pages 450-454.
- [174] B. Shaw, S. K. Parui, and M. Shridhar. A segmentation based approach to offline handwritten Devanagari word recognition. *Proceedings of IEEE International Conference on Information Technology*, pp: 256–257, 2008.
- [175] M. Schmitt and J. Mattioli. *Morphologie Mathématique. Logique Mathématiques Informatique*. Masson, Paris, 1993.
- [176] A. Selamat, C.C. Ng. Arabic script web page language identifications using decision tree neural networks. *Pattern Recognition*, Volume 44, Issue 1, January 2011, Pages 133-144.
- [177] J. Serra. *Image Analysis and Mathematical Morphology. II : Theoretical Advances*. Academic Press. London, 1988.
- [178] M. P. Singh and V.S. Dhaka. Handwritten character recognition using modified gradient descent technique of neural networks and representation of conjugate descent for training patterns. *IJE TRANSACTIONS A: Basics Vol. 22, No. 2 (June 2009)* 145-158.
- [179] G. Sinha and J. kumar. Arabic numeral Recognition Using SVM Classifier. *International Journal of Emerging Research in Management & Technology ISSN: 2278-9359 (Volume-2, Issue-5), May 2013*.
- [180] J. A. Snyman. *Practical Mathematical Optimization : An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms*. Springer Publishing, 2005.
- [181] P. Soille. *Morphological Image Analysis : Principles and Applications*. 2nd edition, 2003.
- [182] D.K. Srivastava and L. Bhambhu. Data classification using Support Vector Machine. *International Journal of Theoretical and Applied Information Technology*, vol. 12, 1, pp. 1-7, 2009.
- [183] T. H. Su, T. W. Zhang, D. J. Guan, and H. J. Huang, Off-Line Recognition of Realistic Chinese Handwriting Using Segmentation-Free Strategy, *Pattern Recognition*, vol.42, no.1, pp.167-182, 2008.
- [184] R.M. Suresh, S. Arumugam. Fuzzy technique based recognition of handwritten characters. *Image and Vision Computing*, Volume 25, Issue 2, February 2007, Pages 230-239.
- [185] J. Tan, J. H. Lai, C. D. Wang, W. X. Wang, and X. X. Zuo. A new handwritten character segmentation method based on nonlinear clustering. *Neurocomputing*, Volume 89, 15 July 2012, Pages 213-219.
- [186] B. Tatyana and K. Ernst. Application of neural classifier for flat image recognition in the process of microdevice assembly, 7278-7803. *IEEE*, 2002.

- [187] V. Vapnik. Estimation of Dependences Based on Empirical Data. Springer, 2006.
- [188] V. Vapnik. Support-vector networks. *Machine Learning* 20(3): 273, 1995.
- [189] Y. Wang, X. Wei, L. Han, and X. Wu. A Novel Character Recognition Algorithm Based on Hidden Markov Models, pp. 298-305, Springer, 2009.
- [190] J. Weston, R. Collobert, F. Sinz, L. Bottou, and V. Vapnik. Inference with the Universum. In *Proceedings of the Twenty-third International Conference on Machine Learning (ICML)*, 2006.
- [191] B. Widrow and Hoff. Adaptive switching circuits, in 1960 IRE WESCON Convention Record, Part 4, New York: IRE, pp. 96–104, 1960.
- [192] B. Widrow and M. A. Lehr. 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation, *Proc. IEEE*, 78:1415–1441, 1990.
- [193] D. H. Wang, C. L. Liu, X. D. Zhou. An approach for real time recognition of online Chinese handwritten sentences. *Pattern Recognition*, Volume 45, Issue 10, October 2012, Pages 3661-3675.
- [194] T. Wakahara and Y. Yamashita. K-NN classification of handwritten characters via accelerated GAT correlation. *Pattern Recognition*, Volume 47, Issue 3, March 2014, Pages 994-1001.
- [195] H. Xiaobing. A printed Chinese character recognition method. *IEEE*, pages 2904 – 2907, June 2011.
- [196] J. Yang and D. Zhang. From classifiers to discriminators : A nearest neighbor rule induced discriminant analysis. *Pattern Recognition*, Volume 44, Issue 7, July 2011, Pages 1387-1402.
- [197] S. Yoon, Y. K. Ham, R. H. Park. Hybrid approaches to frontal view face recognition using the hidden Markov model and neural network. *Pattern Recognition*, Volume 31, Issue 3, March 1998, Pages 283-293 K.
- [198] W. Zeng, X. X. Meng, C. L. Yang, and L. Huang. Feature extraction for handwritten characters using Delaunay triangulation. *Computers & Graphics*, Volume 30, Issue 5, October 2006, Pages 779-786.

ANNEXES

Annexe A : Interface graphique associée à l'application

En vue de bien positiver les travaux qu'on a réalisé dans cette thèse, on a conçu les interfaces graphiques présentées ci-dessous et qui regroupent d'une manière probante ainsi significative toutes les sortes des systèmes de reconnaissance qu'on a mis en action.

Royaume du Maroc
Université Sultan Moulay Slimane
Faculté des Sciences et Techniques
Département d'informatique
Béni Mellal

THÈSE
**La reconnaissance automatique des caractères: Application au Tifinagh
et aux chiffres Arabes**

Présentée par:
Mr. Badre-eddine EL KESSAB

Soutenue publiquement le 18 octobre 2014 devant les membres du jury:

Pr. Ahmed ZEGHAL	: Faculté des Sciences et Techniques de Béni Mellal	Président
Pr. Mohammed ABBAD	: Faculté des Sciences de Rabat	Rapporteur
Pr. Brahim MINAOUI	: Faculté des Sciences et Techniques de Béni Mellal	Rapporteur
Pr. Mohamed SABRI	: Faculté des Sciences et Techniques de Béni Mellal	Rapporteur
Pr. Cherki DAOUI	: Faculté des Sciences et Techniques de Béni Mellal	Directeur de thèse
Pr. Belaid BOUIKHALENE	: Faculté Polydisciplinaire de Béni Mellal	Co-directeur de thèse
Pr. Mohamed FAKIR	: Faculté des Sciences et Techniques de Béni Mellal	Co-directeur de thèse

[Suivant](#)

Figure A.92: Interface de la page d'accueil



Figure A.93: Interface graphique mise en place pour le processus de la reconnaissance

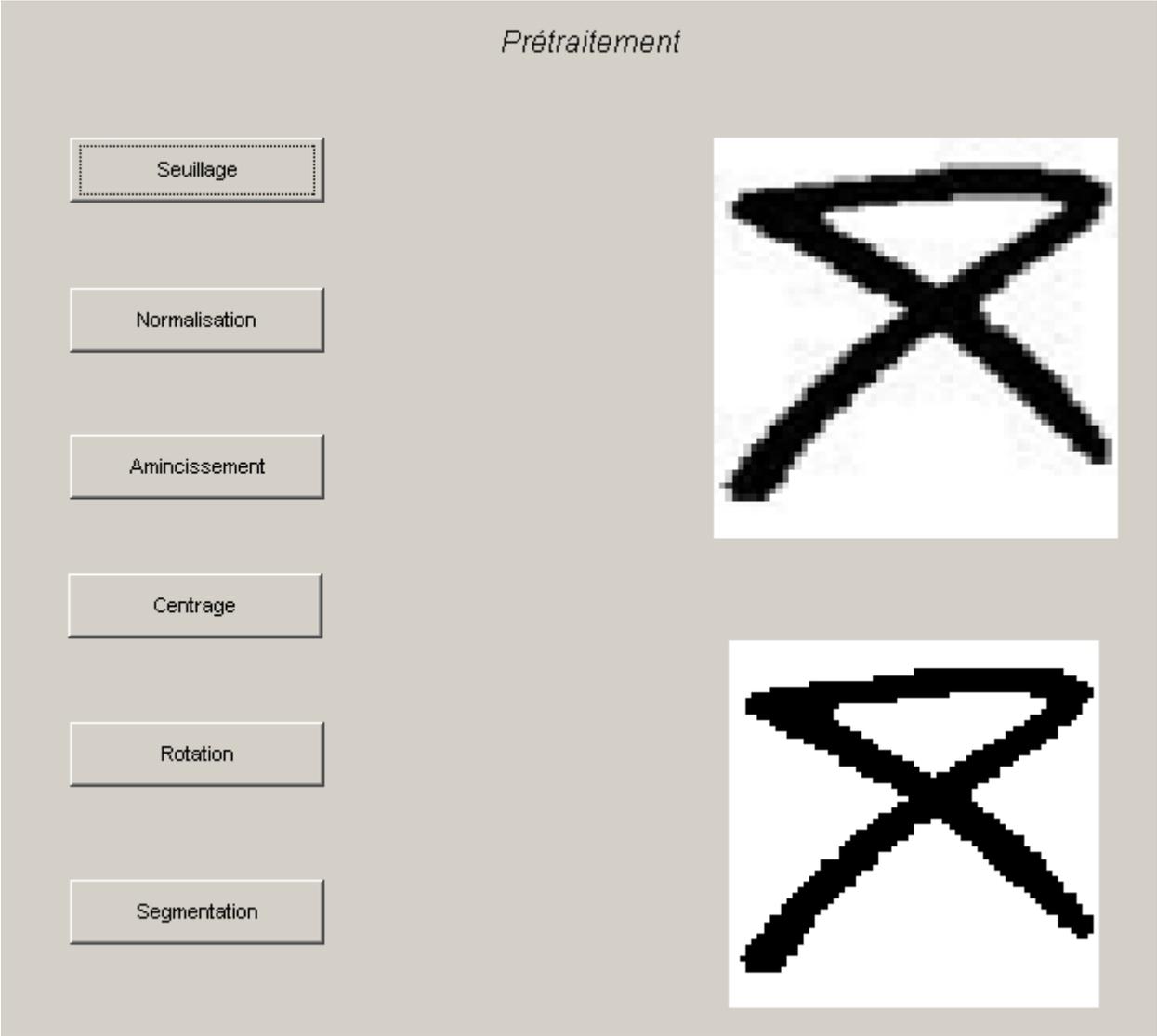


Figure A.94: Interface graphique mise en place pour le prétraitement

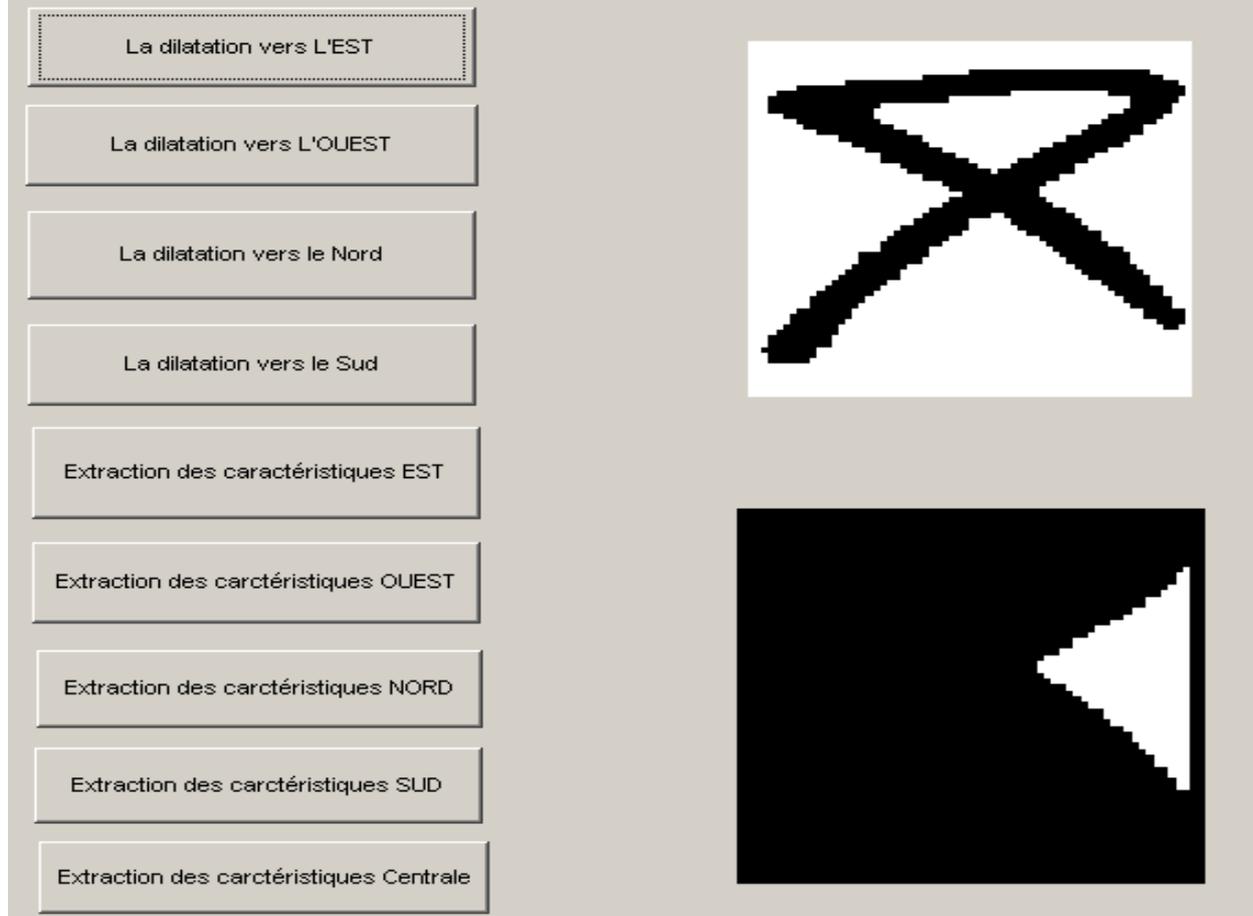


Figure .95: Interface graphique mise en place pour l'extraction des primitives avec la Morphologie

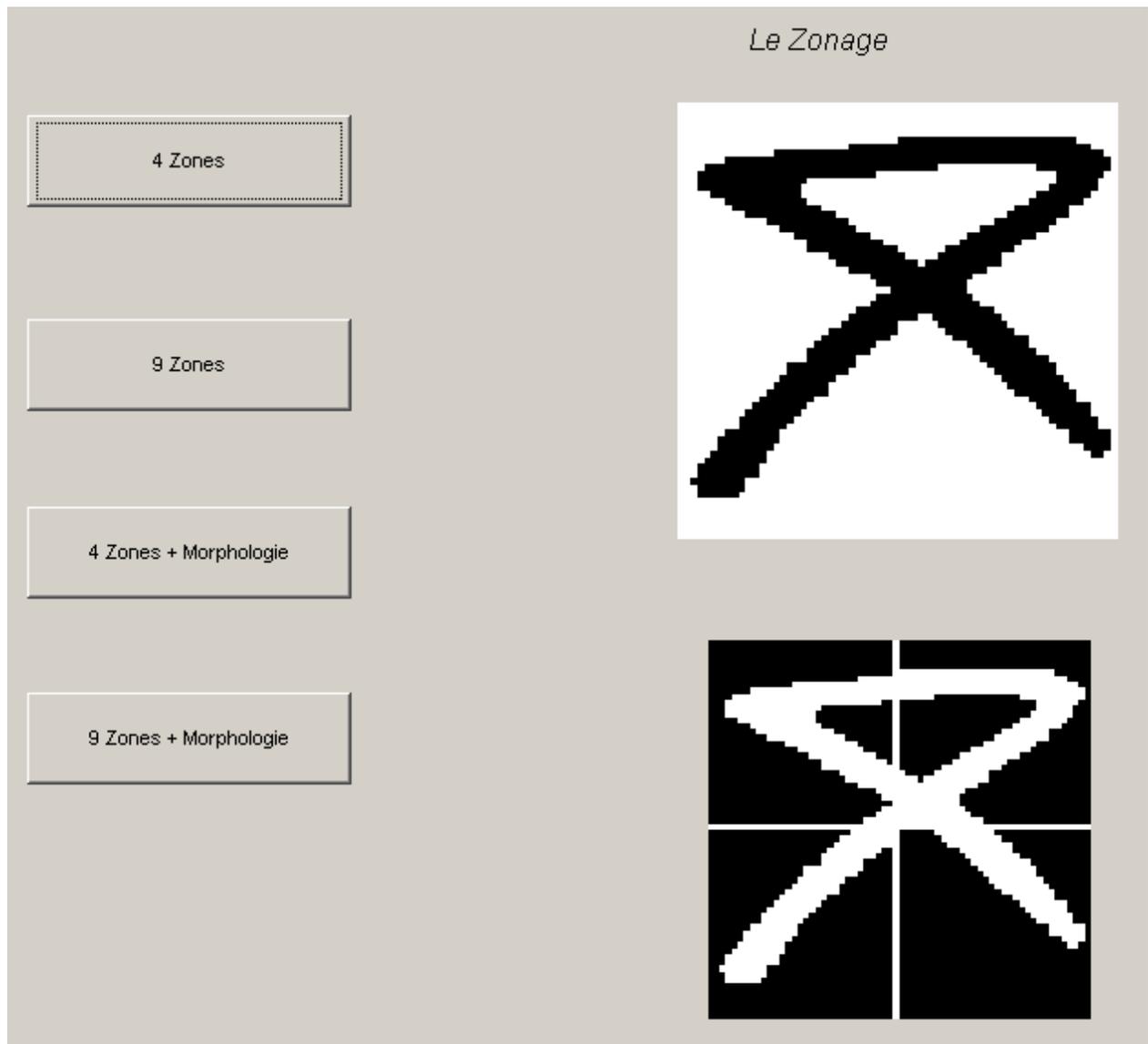


Figure V.96: Interface graphique mise en place pour l'extraction des primitives avec le Zonage

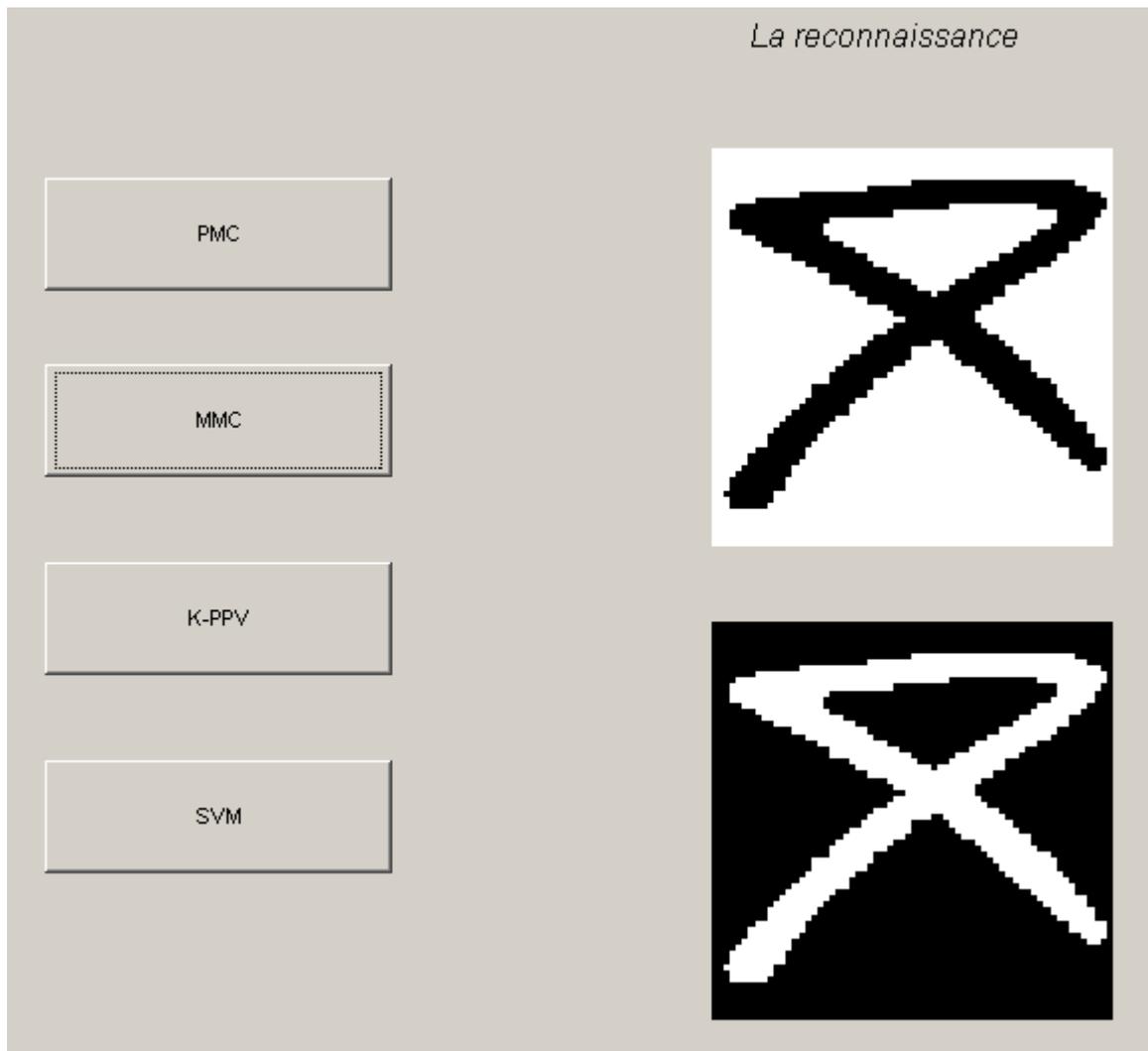


Figure V.97: Interface graphique mise en place pour la reconnaissance

Annexe B : Algorithmes de dilatation

B1 : Algorithme de dilatation vers l'Ouest

Répéter

Pour chaque ligne / colonne faire

- 1) Si Image (Ligne, Colonne) = 1
 - 2) {
 - 3) Image (Ligne -1, Colonne) = 1
 - 4) Incrémentation
 - 5) }
 - 6) La Dilatation de l'image vers L'Ouest.
- Fin Si**

Fin

Jusqu'à « aucune modification n'est possible sur l'image »

B2 : Algorithme de dilatation vers le Nord

Répéter

Pour chaque ligne / colonne faire

- 1) Si Image (Ligne, Colonne) = 1
 - 2) {
 - 3) Image (Ligne, Colonne -1) = 1
 - 4) Incrémentation
 - 5) }
 - 6) La Dilatation de l'image vers le Nord.
- Fin Si**

Fin

Jusqu'à « aucune modification n'est possible sur l'image »

B3 : Algorithme de dilatation vers le Sud

Répéter

Pour chaque ligne / colonne faire

- 1) Si Image (Ligne, Colonne) = 1
- 2) {
- 3) Image (Ligne, Colonne + 1) = 1
- 4) Incrémentation
- 5) }
- 6) La Dilatation de l'image vers le Sud.

Fin Si

Fin

Jusqu'à « aucune modification n'est possible sur l'image »

Annexe C : Zones caractéristiques

C1 : La zone caractéristique d'Est

ZE : l'image de la zone caractéristique **Est** est donnée par :

ZE = (ImageOuest) and (ImageNord) and (ImageSud) and (I)

Avec

And : ET logique (pixel par pixel).

ImageEst : la dilatation de l'image traitée vers l'Est.

ImageOuest : la dilatation de l'image traitée vers l'Ouest.

ImageNord : la dilatation de l'image traitée vers le Nord.

ImageSud : la dilatation de l'image traitée vers le Sud.

I : l'inversion de l'image initiale.

C2 : La zone caractéristique d'Ouest

ZO : l'image de la zone caractéristique **Ouest** est donnée par :

ZO = (ImageEst) and (ImageNord) and (ImageSud) and (I)

Avec

And : ET logique (pixel par pixel).

ImageEst : la dilatation de l'image traitée vers l'Est.

ImageOuest : la dilatation de l'image traitée vers l'Ouest.

ImageNord : la dilatation de l'image traitée vers le Nord.

ImageSud : la dilatation de l'image traitée vers le Sud.

I : l'inversion de l'image initiale.

C3 : La zone caractéristique du Nord

ZN : l'image de la zone caractéristique **Nord** est donnée par :

ZN = (ImageEst) and (ImageOuest) and (ImageSud) and (I)

Avec

And : ET logique (pixel par pixel).

ImageEst : la dilatation de l'image traitée vers l'Est.

ImageOuest : la dilatation de l'image traitée vers l'Ouest.

ImageNord : la dilatation de l'image traitée vers le Nord.

ImageSud : la dilatation de l'image traitée vers le Sud.

I : l'inversion de l'image initiale.

C4 : La zone caractéristique du Sud

ZS : l'image de la zone caractéristique **Sud** est donnée par :

ZS = (ImageEst) and (ImageOuest) and (ImageNord) and (I)

Avec

And : ET logique (pixel par pixel).

ImageEst : la dilatation de l'image traitée vers l'Est.

ImageOuest : la dilatation de l'image traitée vers l'Ouest.

ImageNord : la dilatation de l'image traitée vers le Nord.

ImageSud : la dilatation de l'image traitée vers le Sud.

I : l'inversion de l'image initiale.

C5 : La zone caractéristique Centrale

ZC : l'image de la zone caractéristique **Centrale** est donnée par :

ZC = (ImageEst) and (ImageOuest) and (ImageNord) and (ImageSud) and (I)

Avec

And : ET logique (pixel par pixel).

ImageEst : la dilatation de l'image traitée vers l'Est.

ImageOuest : la dilatation de l'image traitée vers l'Ouest.

ImageNord : la dilatation de l'image traitée vers le Nord.

ImageSud : la dilatation de l'image traitée vers le Sud.

I : l'inversion de l'image initiale.

Annexe D: Démonstration de l'algorithme de Widrow-Hoff

L'algorithme de Widrow-Hoff est donné par la formule suivante :

$$W_j(t + 1) = W_j(t) + \Delta_{W_j}(E) \quad (83)$$

$$= W_j(t) - \alpha * \nabla_{W_j}(E) \quad (84)$$

$$= W_j(t) - \alpha * \frac{\partial E}{\partial W_j} \quad (85)$$

Avec:

$\alpha \in [0, 1]$ est le taux d'apprentissage.

W_j est le $j^{\text{ème}}$ élément de la matrice de poids de connexion.

t désigne l'itération courante.

E est l'erreur quadratique $E = \frac{1}{2}(y - o)^2$, où y la réponse théorique et o la sortie du neurone est donnée par

$$o = \begin{cases} 1 & \text{si } \sum_{j=1}^t W_j * x_j > 0 \\ 0 & \text{sinon} \end{cases} \quad (86)$$

Où x_j est le $j^{\text{ème}}$ élément du vecteur d'entrée.

On obtient donc :

$$W_j(t + 1) = W_j(t) - \alpha * \frac{\partial \left(\frac{1}{2} (y - \sum_{k=1}^{k=I} W_k * x_k)^2 \right)}{\partial W_j} \quad (87)$$

$$= W_j(t) - \alpha * (y - \sum_{k=1}^{k=I} W_k * x_k) * \frac{\partial (y - \sum_{k=1}^{k=I} W_k * x_k)}{\partial W_j} \quad (88)$$

$$= W_j(t) - \alpha * (y - \sum_{k=1}^{k=I} W_k * x_k) * (-x_j) \quad (89)$$

$$= W_j(t) + \alpha * (y - \sum_{k=1}^{k=I} W_k * x_k) * (x_j) \quad (90)$$

On en déduit que :

$$\Delta_{W_j}(E) = \alpha * (y - \sum_{k=1}^{k=I} W_k * x_k) * (x_j) \quad (91)$$

Annexe E : Démonstration de l'algorithme de rétro-propagation du gradient

Avant de présenter les différentes démarches suivies dans la démonstration de cet algorithme, rappelons d'abord le schéma représentant l'architecture de PMC pour que cette démonstration soit bien compréhensible.

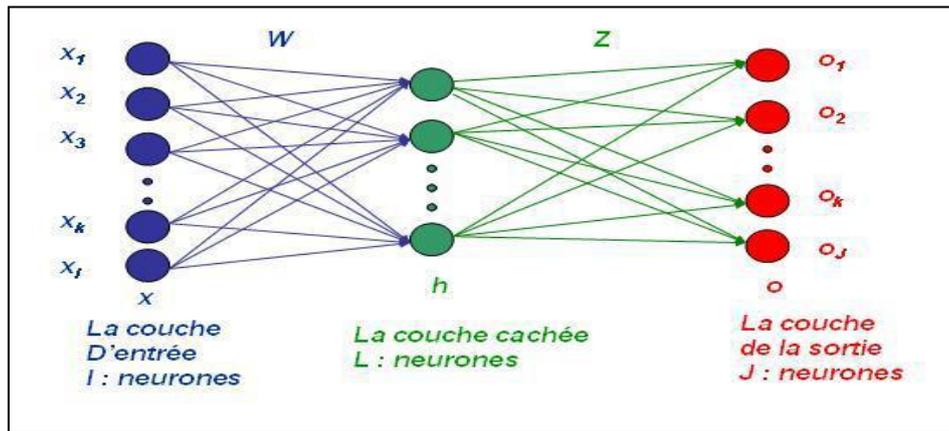


Figure E.98: Le perceptron multicouche n'ayant qu'une seule couche cachée

Les données présentées dans le schéma ci-dessus sont :

I : le nombre de neurones dans la couche d'entrée.

L : le nombre de neurones dans la couche cachée.

J : le nombre de neurones dans la couche de sortie.

K : le nombre de stimuli.

x : stimuli d'entrée.

Où

Le 1^{er} stimuli est noté $x_1 = \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{I1} \end{pmatrix}$

Le 1^{ème} stimuli est noté $x_1 = \begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{I1} \end{pmatrix}$

Le k^{ème} stimuli est noté $x_k = \begin{pmatrix} x_{1k} \\ x_{2k} \\ \vdots \\ x_{Ik} \end{pmatrix}$

Donc :

$$x = \begin{pmatrix} x_{11} & x_{1l} & x_{1k} \\ x_{21} & \dots & x_{2k} \\ \vdots & & \vdots \\ x_{l1} & x_{ll} & x_{lk} \end{pmatrix}$$

h : la réponse de la couche cachée.

Le 1^{er} stimuli est noté $h_1 = \begin{pmatrix} h_{11} \\ h_{21} \\ \vdots \\ h_{l1} \end{pmatrix}$

Le l^{ème} stimuli est noté $x_l = \begin{pmatrix} x_{1l} \\ x_{2l} \\ \vdots \\ x_{ll} \end{pmatrix}$

Le k^{ème} stimuli est noté $x_k = \begin{pmatrix} x_{1k} \\ x_{2k} \\ \vdots \\ x_{lk} \end{pmatrix}$

Donc

$$h = \begin{pmatrix} h_{11} & h_{1l} & h_{1k} \\ h_{21} & h_{2l} & h_{2k} \\ \vdots & \vdots & \vdots \\ h_{l1} & h_{ll} & h_{lk} \end{pmatrix}$$

o : la réponse théorique ou désirée de la couche de sortie.

Le 1^{er} stimuli est noté $o_1 = \begin{pmatrix} o_{11} \\ o_{21} \\ \vdots \\ o_{j1} \end{pmatrix}$

Le l^{ème} stimuli est noté $o_l = \begin{pmatrix} o_{1l} \\ o_{2l} \\ \vdots \\ o_{jl} \end{pmatrix}$

Le k^{ème} stimuli est noté $o_k = \begin{pmatrix} o_{1k} \\ o_{2k} \\ \vdots \\ o_{jk} \end{pmatrix}$

Donc

$$o = \begin{pmatrix} o_{11} & o_{1l} & o_{1k} \\ o_{21} & o_{2l} & o_{2k} \\ \vdots & \vdots & \vdots \\ o_{j1} & o_{jl} & o_{jk} \end{pmatrix}$$

W : la matrice d'ordre $L \times I$ des valeurs de la connexion reliant les cellules de la couche d'entrée aux celles de la couche cachée.

$$W = [W_1, W_2, W_3, \dots, W_L].$$

Avec

$$W_1 = \begin{pmatrix} W_{11} \\ W_{21} \\ \vdots \\ W_{L1} \end{pmatrix}$$

$$W_2 = \begin{pmatrix} W_{12} \\ W_{22} \\ \vdots \\ W_{L2} \end{pmatrix}$$

$$W_L = \begin{pmatrix} W_{1L} \\ W_{2L} \\ \vdots \\ W_{LL} \end{pmatrix}$$

Z : la matrice d'ordre $J \times L$ des valeurs de la connexion reliant les cellules de la couche cachée aux celles de la couche de la sortie

$$Z = [Z_1, Z_2, Z_3, \dots, Z_J].$$

Avec

$$Z_1 = \begin{pmatrix} Z_{11} \\ Z_{21} \\ \vdots \\ Z_{L1} \end{pmatrix}$$

$$Z_2 = \begin{pmatrix} Z_{12} \\ Z_{22} \\ \vdots \\ Z_{L2} \end{pmatrix}$$

$$Z_J = \begin{pmatrix} Z_{1J} \\ Z_{2J} \\ \vdots \\ Z_{LJ} \end{pmatrix}$$

Maintenant on pourra passer à la démonstration :

Lorsque le $k^{\text{ème}}$ stimuli est présenté en entrée, le vecteur de réponse des cellules de la couche cachée h_k est :

$$h_k = \varphi (W * x_k) \tag{92}$$

Tandis que le vecteur de réponse de la cellule de la couche de sortie est :

$$o_k = \varphi (Z * h_k) \tag{93}$$

Ou φ est la fonction sigmoïde dont la dérivée est donnée par :

$$\varphi'(x) = \varphi(x) * (1 - \varphi(x)) \quad (94)$$

En revanche, les matrices de réponse de la couche cachée ainsi de celle de sortie sont respectivement:

$$h = \frac{1}{(1 + \exp(-x * W))} \quad (95)$$

$$o = \frac{1}{(1 + \exp(-h * Z))} \quad (96)$$

L'erreur de la couche de sortie est : $e_k = (t_k - o_k)$. (97)

L'erreur quadratique est donnée par : $E_k = \frac{1}{2} * e_k^2$ (98)

E1 : La correction de poids de la couche de sortie

Le vecteur de réponse de la couche de sortie est donné par la formule suivante

$$o_k = \varphi(Z * h_k) \quad (99)$$

Le gradient de l'erreur par rapport à Z est donné par :

$$\nabla_Z E_k = \frac{\partial E_k}{\partial Z} \quad (100)$$

On obtient par la suite :

$$\nabla_Z E_k = \frac{\partial E_k}{\partial o_k} * \frac{\partial o_k}{\partial Z} \quad (101)$$

$$= \frac{\partial E_k}{\partial o_k} * \frac{\partial o_k}{\partial (Z * h_k)} * \frac{\partial (Z * h_k)}{\partial Z} \quad (102)$$

$$= (o_k - t_k) * o_k * (1 - o_k) * h_k \quad (103)$$

Donc :

$$\nabla_Z E_k = \delta_{sortie,k} * h_k \quad (104)$$

Or que la matrice de connexion de la couche de sortie est :

$$Z_{t+1} = Z_t + \eta * \Delta(Z) \quad (105)$$

Ce qui implique que :

$$Z_{t+1} = Z_t - \eta * \nabla_Z(E_k) \quad (106)$$

$$\begin{aligned} Z_{t+1} &= Z_t + \eta * (o_k - t_k) * o_k * (1 - o_k) * h_k \\ &= Z_t + \eta * \delta_{sortie,k} * h_k \end{aligned} \quad (107)$$

Avec $\delta_{sortie,k}$ est le signal d'erreur de la couche de sortie.

$$\delta_{sortie,k} = (o_k - t_k) * o_k * (1 - o_k) \quad (108)$$

E2 : La correction de poids de la couche cachée

La matrice de connexion de la couche cachée est :

$$W(t + 1) = W(t) + \eta * \Delta(W) \quad (109)$$

Ce qui implique que :

$$W(t + 1) = W_j(t) - \eta * \nabla_W(E_k) \quad (110)$$

On note $h_k = \varphi(W * x_k)$ la $k^{\text{ème}}$ composante du vecteur réponse de la couche de sortie. Le gradient de l'erreur par rapport à W est par définition est donné par :

$$\nabla_W E_k = \frac{\partial E_k}{\partial W} \quad (111)$$

$$= \frac{\partial E_k}{\partial o_k} * \frac{\partial o_k}{\partial W} \quad (112)$$

$$= \frac{\partial E_k}{\partial o_k} * \frac{\partial o_k}{\partial(Z * h_k)} * \frac{\partial(Z * h_k)}{\partial h_k} * \frac{\partial h_k}{\partial(W * x_k)} * \frac{\partial(W * x_k)}{\partial W} \quad (113)$$

$$= \frac{\partial E_k}{\partial o_k} * \frac{\partial \varphi(Z * h_k)}{\partial(Z * h_k)} * \frac{\partial(Z * h_k)}{\partial h_k} * \frac{\partial \varphi(W * x_k)}{\partial(W * x_k)} * \frac{\partial(W * x_k)}{\partial W} \quad (114)$$

La $k^{\text{ème}}$ composante du vecteur erreur quadratique $E_k = \frac{1}{2} * (t_k - o_k)^2$ ce qui implique que

$$\frac{\partial E_k}{\partial o_k} = -(t_k - o_k). \quad (115)$$

D'où :

$$\frac{\partial \varphi(W * x_k)}{\partial(W * x_k)} = h_k * (1 - h_k) \quad (116)$$

D'autre part et du fait que $\frac{\partial(W * x_k)}{\partial W} = x_k$ car x_k ne dépend pas du W .

$$\frac{\partial E_k}{\partial o_k} * \frac{\partial \varphi(Z * h_k)}{\partial (Z * h_k)} = (o_k - t_k) * o_k * (1 - o_k) \quad (117)$$

Et $\frac{\partial (Z * h_k)}{\partial h_k} = Z$ car Z ne dépend pas du h_k .

$$h_k = \varphi(W * o_k) \quad (118)$$

Donc :

$$\nabla_W E_k = (o_k - t_k) * o_k * (1 - o_k) * Z * h_k * (1 - h_k) * x_k \quad (119)$$

Et par suite :

$$\nabla_W E_k = \delta_{sortie,k} * Z * h_k * (1 - h_k) * x_k \quad (120)$$

Ce qui implique que :

$$W(t + 1) = W_j(t) + \eta * \delta_{sortie,k} * Z * h_k * (1 - h_k) * x_k \quad (121)$$

Enfin, on aura l'expression finale de la matrice de correction d'erreur de la couche de sortie qui est donnée par :

$$W_{t+1} = W_t + \eta * \delta_{caché,k} * x_k \quad (122)$$

Annexe F: Les paramètres du modèle de Markov caché

F1: Algorithme de Backward

$\beta_t(i)$ Peut se voir comme la probabilité d'observer la suite $(\mathbf{o}_{t+1}\mathbf{o}_{t+2}\dots \mathbf{o}_T)$, qui commence à l'instant $t+1$ et se termine (à l'instant t) à l'état S_i . L'algorithme **Backward** se détermine ainsi par:

Pour $i=1$ à N **Faire**

$$\beta_T(i) = 1$$

Finpour

Pour $t=T-1$ à 1 **Faire**

Pour $i=1$ à N **Faire**

$$\beta_t(i) = 1 = \sum_{j=1}^N a_{ij} \beta_{t+1}(j) b_j(O_{t+1})$$

Finpour

Finpour

$$P(O | \lambda) = \sum_{i=1}^N \beta_1(i)$$

F2: Algorithme de Forward

$\alpha_t(i)$ est la probabilité de la suite d'observations partielle $(\mathbf{o}_1\mathbf{o}_2\dots \mathbf{o}_t)$, se termine (à l'instant t) à l'état S_i . L'algorithme **Forward** se détermine ainsi par:

Pour $i=1$ à N **Faire**

$$\alpha_1(i) = \prod_i b_i(o_1)$$

Finpour

Pour $t=1$ à $T-1$ **Faire**

Pour $j=1$ à N **Faire**

$$\alpha_{t+1}(j) = 1 = (\sum_{i=1}^N \alpha_t(i) a_{ij}) b_j(O_{t+1})$$

Finpour

Finpour

$$P(O | \lambda) = \sum_{i=1}^N \alpha_T(i)$$

F3 : Algorithme de Baum Welch

L'algorithme de Baum-Welch est un algorithme à apprentissage. Son but étant la maximisation de la vraisemblance d'un modèle λ , celui-ci modifie substantiellement les paramètres du modèle étudié afin d'augmenter sa vraisemblance. L'algorithme réalise son optimisation en ré-estimant les différents paramètres (A, B et Π), suivant la (ou les) séquence(s) observée(s).

Les estimations peuvent donc logiquement se concevoir ainsi :

$$\Pi_i(k) = \text{probabilité d'être dans l'état } S \text{ à l'instant } t = 1 \quad (123)$$

$$a_{ij} = \frac{\text{nombre de transitions de l'état } S_i \text{ vers } S_j}{\text{nombre de fois où l'on quitte } S_i} \quad (124)$$

$$b_j(k) = \frac{\text{nombre de fois où l'on est dans l'état } S_j \text{ en observant le symbole } v_k}{\text{nombre de fois où l'on est dans l'état } S_i} \quad (125)$$

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \quad (126)$$

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \frac{\alpha_t(i) \beta_t(i)}{P(O | \lambda)} \quad (127)$$

$$\Pi_i = \gamma_1(i) \quad 1 \leq i \leq N$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (128)$$

$$1 \leq i \text{ et } j \leq N$$

$$b_i(k) = \frac{\sum_{t=1 \wedge o_t=v_k}^{T-1} \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)} \quad (129)$$

$$1 \leq i \leq N$$

Après avoir ré-estimer les différentes paramètres du modèle d'origine, l'algorithme recalcule la vraisemblance du nouveau modèle. Il va ensuite ré-itéré les différentes opérations de ré-estimation avec ce nouveau modèle, tant que la vraisemblance courante n'est pas maximale ($P(O | \lambda)=1$).

L'algorithme de Baum-Welch se construit ainsi :

```

Pour t= 1 à T Faire
  Pour i= 1 à N Faire
    Pour j= 1 à N Faire
      Calculer  $\xi_t(i,j)$ 
    Finpour
      Calculer  $\gamma_t(i)$ 
  Finpour
Finpour
Ré-estimer  $\lambda = (A, B, \Pi)$ 
Tantque (il y'a augmentation de  $P(O | \lambda)$ ) ou (Il y a encore des itérations à Faire)

```

F4: Algorithme de Viterbi

Cet algorithme est a pour but de trouver la séquence d'états la plus probable ayant produit la séquence mesurée. L'algorithme de Viterbi se construit ainsi :

```

Pour i= 1 à n Faire
   $\delta_1(i) = \pi_i b_i(O_1)$ 
   $\psi_1(i) = 0$ 
Finpour
T=2
Tant que t différent de T faire
  j= 1
  Pour j ≤ n Faire
    
$$\delta_t(i) = \max_{j=1\dots n} (\delta_{t-1}(j) * a_{j,i}) * b_i(O_t)$$

    
$$\psi_t(i) = \arg \max_{j=1\dots n} (\delta_{t-1}(j) * a_{j,i})$$

  j= j+1
Finpour
  t= t+1
Fin Tant que
 $P^* = \max_{i=1\dots n} (\delta_T(i))$ 
 $q_T^* = \arg \max_{i=1\dots n} (\delta_T(i))$ 
  t=T
Tant que t ≥ T faire
   $q_t^* = \psi_{t+1}(q_{t+1}^*)$ 
  t = t - 1
Fin tant que

```

Annexe G : Supports vecteurs machines

G1 : Principe de Lagrange (1788)

Pour résoudre un problème d'optimisation sous contrainte, il suffit de rechercher un point stationnaire z_0 de l'opérateur lagrangien $L(z, \alpha)$ de la fonction g à optimiser et les fonctions C_i^g exprimant les contraintes.

$$L(z, \alpha) = g(z) + \sum_{i=1}^m \alpha_i C_i^g(z) \quad (130)$$

Où

g s'appelle la fonction objectif dont on cherchera soit à maximiser ou à minimiser. Les $\alpha_i = (\alpha_1, \dots, \alpha_m)$ sont des constantes appelées multiplicateurs de Lagrange.

G2 : Principe de Karush-Kuhn-Tucker KKT (1951)

Avec des fonctions g et C_i^g convexe, il est toujours possible de trouver un point critique (z_0, α^*) vérifiant :

$$\min_z L(z, \alpha^*) = L(z_0, \alpha^*) = \max_{\alpha \geq 0} L(z_0, \alpha) \quad (131)$$

En appliquant le principe de Karush-Kuhn-Tucker, on est amené à rechercher un point-critique (w_0, b_0, α^0) . Le lagrangien correspondant à notre problème est :

$$L(w, b, \alpha) = \frac{1}{2} w \cdot w - \sum_{i=1}^m \alpha_i [y_i [(x_i \cdot w) + b] - 1] \quad (132)$$

Le lagrangien doit être minimal par rapport à w et b et maximal par rapport à $\alpha \geq 0$.

$L(w, b, \alpha)$ est minimal par rapport à b :

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \Leftrightarrow \sum_{i=1}^m \alpha_i y_i = 0 \quad (133)$$

$L(w, b, \alpha)$ est minimal par rapport à w :

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \Leftrightarrow w - \sum_{i=1}^m \alpha_i x_i y_i = 0 \quad (134)$$

$L(w, b, \alpha)$ est maximal par rapport à $\alpha \geq 0$.

En remplaçant (141) et (142) dans le lagrangien (140) on aura

$$L(w, b, \alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (135)$$

Ainsi notre problème est de maximiser $L(w, b, \alpha)$ sous la contrainte :

$$\sum_{i=1}^m \alpha_i y_i = 0; \alpha \geq 0 \quad (136)$$

Soit la solution $\alpha^0 = (\alpha_1^0, \dots, \alpha_m^0)$. Selon le théorème de KKT une condition nécessaire et suffisante pour que α^0 soit optimal est :

$$\alpha_i^0 [y_i [(w_0 \cdot x_0) + b_0] - 1] = 0 \text{ pour } i=1, \dots, m$$

$$\text{Ce qui veut dire que : } \alpha_i^0 = 0 \text{ où } y_i [(w_0 \cdot x_0) + b_0] = 1 \quad (137)$$

G3 : Théorème de Mercer :

Pour être sûr qu'une fonction symétrique $K(u, v)$ admet un développement de la forme suivante :

$$K(u, v) = \sum_{k=1}^{+\infty} \beta_k \psi_k(u) \cdot \psi_k(v) \quad (138)$$

Tel que les $\beta_k > 0$ (i.e. $K(u, v)$ décrit un produit interne dans l'espace des caractéristiques) il est nécessaire et suffisant que la condition suivante soit satisfaite :

$$\iint K(u, v) g(u) g(v) du dv \geq 0 \quad (139)$$

Pour toute fonction $g \neq 0$ avec :

$$\int g^2(z) dz \geq 0 \quad (140)$$

On appelle ces fonctions les noyaux de Hilbert-Schmidt. Plusieurs noyaux ont été utilisés par les chercheurs, en voici quelques-uns :

- Le noyau linéaire :

$$K(x, x_i) = x * x_i \quad (141)$$

- Le noyau fonction de base radiale (RBF) :

$$K(x, x_i) = e^{-\frac{\|x-x_i\|^2}{2\sigma^2}} \quad (142)$$

Où σ est l'écart type.

- Le noyau polynomial :

$$K(x, x_i) = (x * x_i + c)^d \quad (143)$$

Où d et c sont respectivement le degré du polynôme et un scalaire.

- Le noyau sigmoïde :

$$K(x, x_i) = \tanh(x * x_i + c) \quad (144)$$

En pratique, les fonctions noyaux s'exploitent bien dans les problèmes de séparation non linéaires par les supports vecteurs machines, dans ce cas cette séparation se réalisera en vertu d'une fonction de décision ou de classement qui permettra de classer un vecteur inconnu selon la forme suivante:

$$\text{Classe}(x) = \text{sign} [\sum_{x_i \in VS} \alpha_i^0 y_i K(x_i, x) + b_0] \quad (145)$$

En vue de bien expliquer comment cette séparation non linéaire s'effectuera, on présente l'exemple suivant :

Etant donné deux classes dont chacune d'entre elles contient un ensemble de vecteur $U \in \mathbb{R}^2$ et qui ne sont pas linéairement séparables, afin de réaliser cette séparation on passera de \mathbb{R}^2 à \mathbb{R}^3 , en fait ce passage est accessible à l'aide de la fonction noyau K (qui optée dans cet exemple un noyau polynomial de degré 2) qui n'est rien d'autre qu'un produit scalaire d'une transformation ψ qui est donnée par :

$$\psi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$U = (u_1, u_2) \rightarrow (u_1^2, \sqrt{2}u_1u_2, u_2^2) \quad (146)$$

D'où :

$$\begin{aligned} K(U, V) &= \psi(U) \cdot \psi(V), \quad \forall V \in \mathbb{R}^2 \\ &= (u_1^2, \sqrt{2}u_1u_2, u_2^2) \cdot \begin{pmatrix} v_1^2 \\ \sqrt{2}v_1v_2 \\ v_2^2 \end{pmatrix} \end{aligned} \quad (147)$$

$$= (u_1^2v_1^2 + 2u_1u_2v_1v_2 + u_2^2v_2^2) \quad (148)$$

$$= (u_1u_2 + v_1v_2)^2 \quad (149)$$

$$= [(u_1 \ u_2) \cdot (v_1 \ v_2)]^2 \quad (150)$$

$$= (U \cdot V)^2 \quad (151)$$