RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Mohammed Seddik Benyahya- Jijel

Faculté Des Sciences Exactes Et Informatique

DÉPARTEMENT DE L'INFORMATIQUE

PROJET DE FIN D'ETUDE

PRÉSENTÉ POUR OBTENIR LE DIPLOME DE

MASTER ACADÉMIQUE

SPÉCIALITÉ: Intelligence Artificielle

THÉME

RECONNAISSANCE DES CHIFFRES À BASE DE SVM

Soutenu au Juin 2017

Présenté par :

Djaaboub Oussama

Ghibour Amal

Sous la direction de:

Mr. Azzedine Laater

Année universitaire : 2016-2017

Résumé

Dans ce travail, nous nous intéressons à la reconnaissance des chiffres manuscrits à base d'une nouvelle méthode d'apprentissage et de classification Inspirée de la théorie de l'apprentissage artificiel de Vapnik. Cette méthode appelée, Machines à Vecteurs de Support (SVM pour Support Vector Machine) a été adaptée et appliquée au problème de la reconnaissance des chiffres manuscrits. L'avantage des SVM est qu'un nombre restreint d'échantillons suffit seul à la détermination des vecteurs de support (SV) permettant la discrimination entre les classes contrairement aux autres classifieurs. Les résultats expérimentaux obtenus sur les chiffres de la base MNIST sont satisfaisants et très encourageants.

Mots Clés : apprentissage, classification, reconnaissance des chiffres manuscrits, machines à vecteurs de supports (SVM),

Abstract

In this work, we are interested in handwritten number recognition based on a new method of training and classification based on artificial training theory of Vapnik. This method named Support Vector Machines (SVM) has been adapted and applied to a problem of the handwritten number recognition. The advantage of SVM is that a restricted number of samples is sufficient to the determination of the vectors support (SV) allowing discrimination between the classes comparatively to the statistical estimation. Obtained results on image of MNIST base are satisfactory and very encouraging.

Key words: training, classification, handwritten number recognition, support vector machines (SVM).

Remerciement

Nos remerciements en premier lieu à **ALLAH** qui Nous a donné le pouvoir pour terminer ce modeste travail.

Nous tient à exprimer nos sentiments de gratitude et de reconnaissance à notre encadreur **Monsieur Azzedine Laater** professeur à l'université de Jijel, consacré leur temps à suivre de près l'évolution de ce projet, à orienter les différentes étapes et à pallier toutes les difficultés auxquelles nous avons eu à faire face. Nous avons lui le respect qu'imposent leur mérite et leur qualité humaine aussi bien que professionnelle.

Nous adressons notre reconnaissance à tous les membres qui nous fait l'honneur d'avoir participé à notre jury.

Nous somme aussi reconnaissant envers tous les enseignants qui ont contribué durant toutes nos études.

Finalement nous souhaitons vivement remercier nos collègues, pour avoir créé une ambiance chaleureuse et amicale tout au long de nos études.

Table Des Matières

Liste Des Figures VII
Liste Des Tableaux IX
Liste Des Algorithmes X
Liste Des Abreviations XI
Introduction Générale 1

CHAPITRE I : Généralités

1- Introduction 3
2- Reconnaissance d'écriture OCR 3
2-1- Forme de Documents 3
2-1-1- Documents Imprimés 3
2-1-2- Documents Manuscrits 4
2-2- Différents Aspects de Reconnaissance d'Ecriture 4
2-2-1- Reconnaissance En-ligne et Hors-ligne 4
2-2-2- Reconnaissance Globale et Analytique 5
3- Acquisition 5
4- Prétraitement 5
4-1- Binarisation 5
4-1-1- Seuillage Global 6
4-1-2- Seuillage Local 6
4-2- Filtrage 6
4-2-1- Filtrage Linéaire 6
4-2-1-1- Filtre Prewitt et Filtre Sobel 6
4-2-2- Filtrage Non Linéaire 7
4-2-2-1- Filtre par Élimination 7
4-2-2-2- Filtre de Gaussien 7
4-2-3- Filtrage Morphologique 7
4-2-3-1- L'érosion 7

4-2-3-2- Dilatation 7
4-3- La Normalisation 7
4-4- Extraction De Composantes Connexes 8
4-5- Extraction de Contour 8
4-6- Squelettisation d'une Image 8
5- Segmentation 9
5-1- Segmentation Explicite 9
5-2- Segmentation Implicite 9
6- Phase d'Analyse ou d'Extraction des Caractéristiques 10
7- Etape d'Apprentissage 10
7-1- Apprentissage Supervisé 10
7-2- Apprentissage Non Supervisé 11
8- Classification 11
9- Phase de Post-Traitement 11
10- Conclusion 11

CHAPITRE II : Les méthodes De Classifications

1- Introduction 12
2- Méthodes Markoviennes 12
2-1- Introduction 12
2-2- Définitions 12
2-2-1- Modèle De Markov Cachés 12
2-2-2- Topologie Des HMMS 13
2-2-2-1- HMM Ergodique 13
2-2-2-2- HMM Gauche-Droite 13
2-3- Modèles de Markov Caches « HMM Planaires» 14
2-4- Les Fonctionnalités d'un HMM 14
2-4-1- L'Évaluation 14

2-4-1-1- Problème d'Évaluation 14
2-4-1-1-1- Forward 14
2-4-1-1-2- Backward 15
2-4-2- L'apprentissage 16
2-4-2-1- L'Algorithme De Viterbi 16
2-4-3- Décodage 16
2-4-3-1- Problème Du Décodage 16
2-5- Avantages et Inconvénients 17
3- La Méthode du K Plus Proches Voisins 17
3-1- Introduction 17
3-2- Définitions 17
3-2-1- Principe 17
3-2-2- Formalisation mathématiques 18
3-3- Avantages et Inconvénients 18
3-4- Algorithme 18
4- Arbre de Décision 19
4-1- Introduction 19
4-2- Explications 19
4-3- Avantages et Inconvénients 20
5- Les Réseaux de Neurones 20
5-1- Introduction 20
5-2- Définitions 20
5-2-1- Un Neurone Formel 20
5-2-2- Le Réseaux de Neurones Artificiel 21
5-3- Perceptrons Multicouches 22
5-3-1- La Propagation 23
5-4- Avantages et Inconvénients 23
6- Classification Bayésienne 24

6-1- Introduction 24
6-2- Définition 24
6-3- Le Classifieur Bayésien Naïf 25
7- Classifieur Euclidien 25
8- Classifieur Quadratique 26
9- Conclusion 26

CHAPITRE III : Le Classifieur SVM

1- Introduction 27
2- Définition 27
3- Séparateur linéaire 27
3-1- Cas linéairement séparable (SVM à Marge Dure) 27
3-2- Cas linéairement non séparable (SVM à Marge Souple) 29
4- Séparation non linéaire 30
4-1- Cas non linéairement séparable 30
4-2- Condition de Mercer 31
4-3- Exemple de noyaux 32
5- L'Algorithme SMO (Sequential Minimal Optimization) 32
6- Les conditions de Karush-Kuhn-Tucker (KKT) 34
7- Calculer le seuil b 34
8- Architecture générale d'un SVM 34
9- Stratégie multi-classes 35
9-1- Approche un-contre-tous (1 VS R) 35
9-2- Approche un-contre-un (1 VS 1) 36
10- Conclusion 37

CHAPITRE IV : Implémentation du système proposé

1- Introduction 38
2- Acquisition 38

3- Prétraitement 39
3-1- Filtrage 39
3-1-1- Filtre Médian 39
3-2- Binarisation 40
3-2-1- La Méthode D'OTSU 40
3-3- Operateurs Morphologiques 41
3-3-1- L'Ouverture Et Fermeture 41
3-4- Squelettisation 42
3-4-1- Algorithme de Zhang et Suen 42
3-5- La Normalisation 43
3-5-1- Correction de l'inclinaison des lignes 43
3-5-1-1- La Transformation de Hough 44
3-5-2- Correction De L'inclinaison Des Lettres 44
3-5-2-1- Moment géométrique de deuxième ordre 45
3-5-3- Normalisation de taille 45
3-6- Détection De Contour 46
3-6-1- Méthode D'Approximation Du Contour 46
4- Segmentation 47
4-1- Rectangle englobant 47
5- Extraction des Caractéristiques 48
5-1- Histogramme de Gradient Orienté (HOG) 48
5-2- Données de pixels brutes 49
5-3- Moments de HU 50
6- Support Vecteur Machine 51
6-1- Pseudo Code pour SMO 51
7- Conclusion 53

CHAPITRE V : Implémentation et application

1- Introduction 54
2- L'Environnement De Développement 54
2-1- Le Langage Python 54
3- La Base De Données 54
4- Apprentissage et Test 54
5- Apprentissage Des SVMs 56
5-1- Paramètre De Régularisation C 56
5-2- Réglages Des Seuils De Tolérance Et å (=EPS) 57
5-3- Choix Du Noyau 57
5-4- Prétraitement De La Base 57
5-5- Protocole Expérimental 57
6- Métriques de performances 57
6-1- Taux de reconnaissance 57
6-2- La matrice de confusion 58
7- Description De Plateforme 58
8- Résultat Expérimentale 63
9- Comparaison Des Résultats avec d'autres classifieurs 69
10- Conclusion 69
Conclusion générale 70

Liste Des Figures

Figure I. 1 : Différents systèmes, représentations et approches de reconnaissance 3
Figure I. 2: Illustration de la binarisation d'une image en niveaux de gris 6
Figure I. 3: Image traitée par Dilatation et érosion 7
Figure I. 4: Exemple de normalisation de mots manuscrits 8
Figure I. 5: Le code de Freeman en 4-connexités et en 8 connexités 8
Figure I. 6: Codage à l'aide du code de Freeman 8
Figure I. 7: La squelettisation d'une Image. 9
Figure I. 8: Différentes segmentation pour le mot quatorze 10

Figure II. 1: Définition des paramètres d'un Modèle de Markov Caché 13
Figure II. 2: Modèle ergodique à 3 états 13
Figure II. 3: modèles gauche-droite à 4 états 13
Figure II. 4: Exemple d'architecture d'un PHMM 14
Figure II. 5: Représentation en treillis d'un HMM 15
Figure II. 6: Exemple de classification bi-classes avec un kppv 17
Figure II. 7: Classification avec les arbres de décision dans un arbre de décision 19
Figure II. 8: Neurone artificiel 21
Figure II. 9: Le modèle d'un mot est la concaténation des modèles HMMs. 22
Figure II. 10: Un perceptron à deux couches cachées. 22

Figure III. 1: Nuage de données d'apprentissage avec un Hyperplan optimal et une marge maximale 28
Figure III. 2: Répartition des exemples avec erreur 29
Figure III. 3: points non linéairement séparable et leur projection vers un autre espace d'une dimension supérieur. 30
Figure III. 4: Principe de fonctionnement d'un SVM (cas des chiffres manuscrits) 35
Figure III. 5: Nuage de points à 3 classes : l'approche un contre Tous 36
Figure III. 6: Une observation appartient à la zone d'ambigüité 37

Figure IV. 1: Architecture du système proposé 38
Figure IV. 2: Chiffres (NDG) 38

Figure IV. 3: Chiffres (RGB) 38
Figure IV. 4: Chiffres (JPEG) 39
Figure IV. 5: Chiffres (N,B) 39
Figure IV. 6: La méthode Otsu après l'application de filtre médian 40
Figure IV. 7: Techniques fermeture 40
Figure IV. 8: Techniques Ouverture 40
Figure IV. 9: Application d'algorithme Zhang Et Suen 42
Figure IV. 10: Représentation d'une droite dans l'espace de Hough 43
Figure IV. 11: Les méthodes de base pour la normalisation: (a) Skew normalization 43
Figure IV. 12: Les méthodes de base pour la normalisation: (b) Slant normalization 44
Figure IV. 13: Les méthodes de base pour la normalisation: (b) size normalization 44
Figure IV. 14: Représentation graphique Des angles α(k), β(k), Tc(k), βc(k) 45
Figure IV. 15: Phénomène de dégénérescence 46
Figure IV. 16: Application d'algorithme d'approximation du contour 46
Figure IV. 17: La déduction du rectangle englobant 46
Figure IV. 18: Rectangle englobante restantes a l'issue de la 1ère étape 47
Figure IV. 19: Rectangle englobante après l'application de la condition 47
Figure IV. 20: Mappage de pixel 48
Figure IV. 21: La conversion de 2D matrice à 1D matrice 48

Figure V. 1: Forme des chiffres dans la base 55
Figure V. 2: Forme des étiquettes dans la base 55
Figure V. 3: Exemples de la base de données MNIST 56
Figure V. 4: Test et Apprentissage de la base de données MNIST 56
Figure V. 5: La fenêtre principale de la plateforme 58
Figure V. 6: les listes de menu 59
Figure V. 7: La barre d'outille 59
Figure V. 8: Page image 60
Figure V. 9: Le graphe de probabilité 61
Figure V. 10: Fenêtre des informations de SVM 61
Figure V. 11: Fenêtre de créer SVM 62
Figure V. 12: Fenêtre de classification 62

Liste Des Tableaux

Tableau II. 1: Quelques fonctions de transfert usuelles. x est le vecteur d'entrée 21

Tableau IV. 1: Table des conditions à chaque itération 43

Tableau V. 1 : Répartition de la base MNIST pour les chiffres. 55

Tableau V. 2: Matrices de confusion pour 1 vs 1 : (HOG, la base N-Norm, noyau linéaire, C=1, tolé=0.001).. 63

Tableau V. 3: Matrice de confusion pour 1 Vs Reste (HOG, la base N-Norm, noyau linéaire, C=1, tolé=0.001)..... 63

Tableau V. 4: Matrice de confusion pour 1 Vs Reste (HOG, la base deslant, noyau RBF, C=1, tolé=0.001). 64

Tableau V. 5: Matrice de confusion pour 1 Vs 1 (Hu, la base N-Norm, noyau poly, degré =2, C=1, tolé=0.001).....64

Tableau V. 6: Matrice de confusion pour 1 Vs 1 (Hu, la base N-Norm, noyau RBF, sigma=1, C=1, tolé=0.001).....65

Tableau V. 7: Matrice de confusion pour 1 Vs 1 (HOG, la base N-Norm, noyau sigmoïde, C =1, tolé=0.001). 65

Tableau V. 8: Matrice de confusion pour 1 Vs 1 (HOG, la base, normalisé : squelette, noyau linéaire, C =1, tolé=0.001).....66

Tableau V. 9: Matrice de confusion pour 1 Vs 1 (HOG, base normalisé (deslant, contour), noyau linéaire, segma=0.0073 C=2.8, tolé=0.001) 66

Tableau V. 10: Matrice de confusion pour 1 Vs 1 (HOG, la base normalisé (deslant), noyau RBF, sigma =1, C =1, tolé=0.001).....67

Tableau V. 11: : Matrice de confusion pour 1 Vs 1 (Raw, la base normalisé (deslant), noyau RBF, sigma= 0.008, C =3, tolé=0.001).....67

Tableau V. 12: Résumer des matrice de confusion précédents et avec d'autre testes.....68

Tableau V. 13: Comparaison des résultats avec d'autre classifieurs 69

Liste Des Algorithmes

Algorithme II. 1: La Récursion Forward Pour HMM 15
Algorithme II. 2: La Récursion Backward pour HMM 15
Algorithme II. 3: De Viterbi « Backtracking » 16
Algorithme II. 4: De K Plus Proches Voisins 19
Algorithme II. 5: De Propagation 23

Algorithme IV. 1: Filtre Médian 40
Algorithme IV. 2: Méthode D'OTSU 41
Algorithme IV. 3: Dilatation Et Erosion 42
Algorithme IV. 4: Transformation Hough 43
Algorithme IV. 5: Approximation Du Contour 46
Algorithme IV. 6: Rectangle englobant 48
Algorithme IV. 7: De histogramme de gradient orienté 49
Algorithme IV. 8: Donneés des pixels brutes 50
Algorithme IV. 9: SMO Simplifié 52

Liste Des Abréviations

O.C.R : Système de Reconnaissance d'Ecriture (Optical Character Recognition).

CXX : Une composante connexe.

HMMS : Chaîne de Markov Cachées (Hidden Markov Models).

PHMM : Chaîne de Markov Cachées Planaires (Planar Hidden Markov Models).

K-PPV : K-plus proches voisins.

KNN : k-Nearest-Neighbor

SVM : Support Vecteur Machine

RBF : Réseau à Base de Fonction radiale

SMO : Optimisation séquentielle minimale (Sequential Minimal Optimization)

KKT : Karush-Kuhn-Tucker

QP : Problème Quadratique

RGB : Format d'image Rouge Vert Bleu

JPEG : Joint Photographic Expert Group

MNIST : (Mixed National Institute of Standards and Technology)

Introduction Générale

Depuis plusieurs années, de nombreux travaux de recherche ont porté sur la reconnaissance des chiffres manuscrits. Deux grandes classes d'application sont aujourd'hui à l'étude : les applications bancaires ou postales, qualifiés de hors lignes ou statiques par exemple en France, deux millions de chèques sont lus chaque jour sans intervention humaine et les applications destinés à la bureautique, qualifiés de en lignes ou dynamiques ou temps réel [01] [02].

Dans notre mémoire, nous nous intéressons au problème de reconnaissance d'écriture manuscrite hors-ligne, qui demeure un problème difficile car l'entrée de ces systèmes est une image. D'autres problèmes doivent également surmontés comme par exemple, comment on peut enlever le bruit comme des éléments du fond d'image, comment on peut traiter le manque des traits etc. De plus, il n'y a pas d'informations supplémentaires comme le cas d'En-ligne. D'ailleurs, pour être utilisés largement, ces systèmes doivent traiter rapidement avec le taux de reconnaissance élevé.

Par contre dans le cas de reconnaissance en ligne l'utilisateur écrit sur une table spéciale, il y a moins de bruit. De plus, on peut déterminer comment un caractère est écrit, c'est à dire, l'ordre de traits constituants ce caractère. D'ailleurs, la contrainte du temps de reconnaissance n'est pas stricte, on peut utiliser des algorithmes complexes. C'est pour quoi le taux de reconnaissance de ces systèmes est assez élevé. [03]

Les chiffres manuscrits hors – ligne reste aujourd'hui un thème de recherche ouvert. Avec en effet, bien que le nombre de classes naturelles soit très réduit (chiffres '0' à '9'), on trouve à l'intérieur de chacune d'entre elles, une très grande variabilité de l'écriture, de plus, les conditions souvent relativement précaires dans les quelles sont écrits les chiffres (chèques écrits rapidement sur un coin de table) et la variabilité du matériel utilisé (utilisation de divers stylos, de différentes qualités de papier) tendent à complexifier la reconnaissance.

Plusieurs générations de machines d'apprentissage ont vu le jour dans le but de classifier, de catégoriser ou de prédire des structures particulières dans les données. Mais la plupart de ces techniques éprouvent de grandes difficultés à traiter les données de très haute dimension. Pour surmonter ce problème, on procède souvent par la sélection d'une partie des attributs des données pour réduire la dimension de l'espace d'entrée. Mais dans ce cas on aura besoin d'utiliser des hypothèses simplificatrices qui ne se vérifient pas toujours en pratique.

Par ailleurs, une méthode issue récemment d'une formulation de la théorie de l'apprentissage statistique due en grande partie à l'ouvrage de Vapnik en 1995 intitulé *the nature of learning statistical theory* [04]. Cette technique est appelée *Support Vector Machines (SVM)* ou machines à vecteurs de support. Bien que récemment proposée, elle a fait l'objet d'un nombre important de publications. Le SVM est un modèle discriminant qui tente de minimiser les erreurs d'apprentissage tout en maximisant la marge séparant les données des classes. La maximisation de la marge est une méthode de régularisation qui réduit la complexité du classifieur.

De nombreux travaux ont démontré la supériorité du SVM sur les méthodes discriminantes classiques. Sa robustesse vis-à-vis de la dimensionnalité des données et son pouvoir accru de généralisation, font que le SVM est nettement plus avantageux [05]. Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent.

L'objet de ce mémoire est l'implémentation des SVMs pour la reconnaissance des chiffres manuscrits. Il est organisé en cinq chapitres. Le premier chapitre présente les différents blocs, schéma standard de reconnaissance des caractères manuscrits en génale.

Le deuxième chapitre décrit en détail les différents classifieurs portant sur l'écriture. On commence d'abord par le Modèle de Markov caché qu'est un très important classifieur a cause de son large utilisation dans le domaine de la reconnaissance de l'écriture, k plus proches voisins, méthode de l'arbre de décision et réseaux de nuerons…

Le troisième chapitre mettant l'accent plus précisément sur le classifieur SVM avec leur présentation de la formulation mathématique. Une description détaillée de l'algorithme SMO est également présentée dans ce chapitre.

Le quatrième chapitre décrit en détaille notre système proposé pour la reconnaissance les chiffres manuscrits avec les différentes étapes a base SVM.

Le cinquième chapitre décrit les résultats expérimentaux obtenus sur une base de données de chiffres ainsi que l'interface que nous avons développée en python 2.7.2 sous l'environnement Windows.

Enfin, nous terminons avec une conclusion générale qui résume l'apport de notre travail.

CHAPITRE I :

Généralité

1- Introduction

2- Reconnaissance d'écriture OCR

3- Acquisition

4- Prétraitement

5- Segmentation :

6- Phase d'Analyse ou d'Extraction des Caractéristiques

7- Etape d'Apprentissage

8- Classification

9- Phase de Post-Traitement

10- Conclusion

1- Introduction

La reconnaissance de caractères et de l'écriture est mieux connue sous le nom d'O.C.R (Optical Character Recognition). En général est un domaine actif de recherche pour la science informatique depuis la fin des années 1950. Le but de la reconnaissance de l'écriture est de transformer un texte écrit en une représentation compréhensible par une machine et apte à être manipulée par les logiciels de traitements de textes.

La tâche de reconnaissance de l'écriture n'est pas triviale car les mots possèdent une infinité de représentations due au fait que chaque personne possède une écriture qui lui est propre, qu'il existe de nombreuses polices de caractères pour l'imprimé avec de nombreux styles (gras, italique, souligné, etc.) et des mises en pages différentes et complexes. Tous ces facteurs rendent la reconnaissance de l'écriture bien complexe.

Parmi les domaines possibles d'application, on trouve le domaine postal pour la reconnaissance des adresses et le tri du courrier, le domaine administratif pour la gestion électronique des flux de documents, les bibliothèques numériques pour l'indexation de documents et la recherche d'information, biométrie pour l'identification du scripteur…etc.

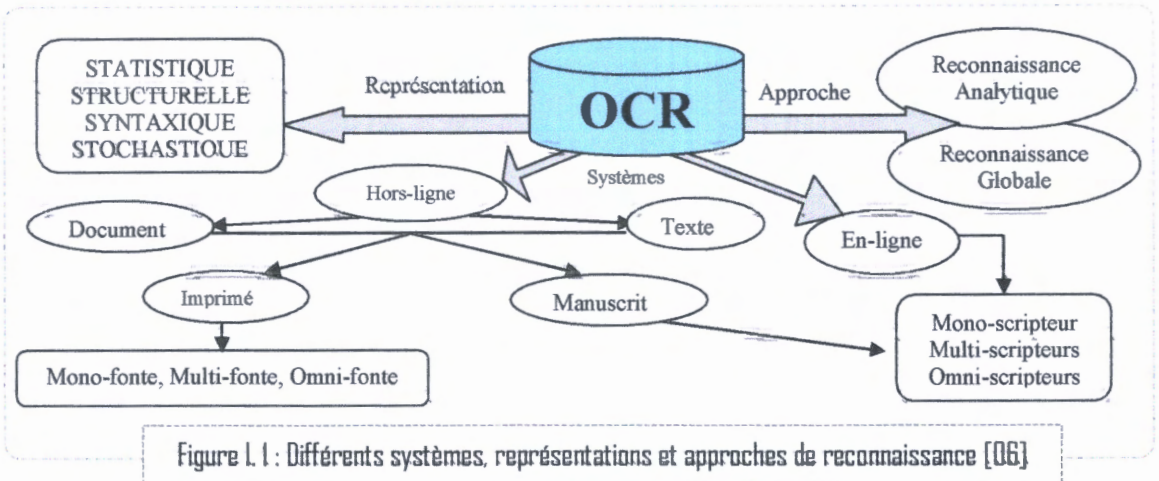


Figure I. 1 : Différents systèmes, représentations et approches de reconnaissance [06]

2- Reconnaissance d'écriture OCR

2-1- Forme de documents

2-1-1- Documents imprimés

Les caractères imprimés sont dans le cas général alignés horizontalement et séparés verticalement, ce qui simplifie la phase de lecture. La forme des caractères est définie par un style calligraphique (fonte) qui constitue un modèle pour l'identification. [06]

Dans ce cas, la reconnaissance peut être mono-fonte, multi-fonte ou omni-fonte: [07] [06]

Un système est dit **mono-fonte** s'il ne peut reconnaître qu'une seule fonte à la fois c'est à dire qu'il ne connaît de graphisme que d'une fonte unique. C'est le cas le plus simple de reconnaissance de caractères imprimés. Un système est dit **multi-fonte** s'il est capable de reconnaître divers types de fontes parmi un ensemble de fontes préalablement apprises. Et un système **omni-fonte** est capable de reconnaître n'importe quelle
fonte, généralement sans apprentissage préalable. Cependant ceci est quasiment impossible car il existe des milliers de fontes dont certaines illisible par l'homme (sauf bien sure pour celui qui l'a conçue) et avec un logiciel de création de fonte n'importe qui peut concevoir des fontes à sa guise.

2-1-2- Documents manuscrites

L'écriture manuscrite comme la parole fait partie de nos modes d'expression les plus évolués en tous les cas les deux sont très complexes, c'est un moyen particulier de codage de l'information qui concerne nos idées, pensées, nos sentiments, d'une autre façon la reconnaissance de l'écriture en général possède deux caractéristiques : Un liée au scripteur par exemple vérifier une signature d'un auteur. L'autre liée au sens de ce qui est écrit (contextuelle). [08]

Dans le cas du manuscrit, la reconnaissance peut être mono-scripteur, multi-scripteur ou omni-scripteur [09].

Un système est dit **Mono-scripteur** : (propres au scripteur) : un seul scripteur peut utiliser le système de reconnaissance après apprentissage de son écriture.

Un système est dit **Multi-scripteur** : (propres à l'écriture manuscrite): c'est que le système peut identifier et reconnaître l'écriture pour un certain nombre de scripteurs.

Un système est dit **Omni-scripteur** : (propres à n'importe quelle écriture manuscrite): le système est censé reconnaître toutes les écritures. Dans ce cas, la variabilité intra-scripteur s'ajoute à la variabilité inter-scripteur.

2-2- Différents aspects de reconnaissance d'écriture

2-2-1- Reconnaissance En-ligne et Hors-ligne

- **En-ligne:** Dans le cas de la reconnaissance *En-ligne* (dynamique) le système reçois les images d'entée de données en temps réel, ce qui permet d'intégrer les mouvements du stylo électronique (Tracé Dynamique) et pressure information [10], et calcule la relation entre les points pour extraire les caractéristiques afin de reconnaître les symboles au fur et à mesure qu'ils sont écrits à la main [11] [12].

- **Hors-ligne :** (ou en différé, ou encore statique) est obtenue par la saisie d'un texte déjà existant, obtenue par un scanner ou une caméra. Dans ce cas, on dispose d'une image binaire ou en niveaux de gris, ayant perdu toute information temporelle sur l'ordre des points. De plus, ce mode introduit une difficulté supplémentaire relative à la variabilité du tracé en épaisseur et en connectivité, nécessitant l'application de techniques de prétraitement [13].

2-2-2- Reconnaissance globale et analytique

Deux approches s'opposent en reconnaissance des mots : globale et analytique :

- **Les méthodes globales**: sont a priori séduisantes puisqu'elles ne rencontrent pas les difficultés liées aux ambiguïtés provenant de la segmentation [14]. Toutefois, elles ne peuvent se concevoir que dans le cas d'un vocabulaire limité, ou dans le cas de vocabulaires de taille plus grande mais limitée dynamiquement. [15]
- **Les méthodes analytiques :** cherchent à identifier les graphèmes issus de la segmentation (fragments de lettres, des lettres ou des regroupements de lettres) pour reconstituer les mots [16]. Elles présentent l'intérêt de pouvoir se généraliser à la reconnaissance d'un vocabulaire étendu ou limité dynamiquement, à partir d'une phase d'apprentissage des classes de graphèmes. [17]

3- Acquisition

La phase d'acquisition consiste à capter l'image d'un texte au moyen des capteurs physiques (scanner, caméra,…) et de la convertir en grandeurs numériques adaptés au système de traitement, avec un minimum de dégradation possible. [18] [19]

4- Prétraitement

Typiquement au moyen d'un scanner en noir et blanc ou en niveau de gris regroupe l'ensemble des opérations visant au bon conditionnement du message écrit et qui sont indispensables à son identification.

Elle inclut plusieurs opérations qui sont appliquées afin de réduire le volume de données à traiter et à éliminer autant que possible les bruits et les variabilités. Parmi les opérations de prétraitement généralement utilisées nous pouvons citer :

4-1- Binarisation

Elle permet de passer d'une image de niveaux de gris à une image binaire composée de deux valeurs 0 et 1, plus simple à traiter (Figure I. 2).

En général, on utilise un seuil de binarisation approprié qui traduit la limite des contrastes fort et faible dans l'image. [20]

Figure I. 2: Illustration de la binarisation d'une image en niveaux de gris

4-1-1- Seuillage global

C'est la technique de binarisation la plus simple, il consiste à prendre un seuil identique pour toute l'image. Chaque pixel de l'image est comparé à ce seuil : ceux de niveaux de gris inférieurs sont mis à 0 «noir », ceux mis à 1 «blanc». [21]

4-1-2- Seuillage local

Le principe du seuillage local est d'utiliser un voisinage centré sur le pixel considéré pour déterminer quel seuil utiliser. Cette fenêtre peut avoir différentes tailles, souvent en fonction de la taille moyenne du texte dans le document. [22] [23]

4-2- Filtrage

4-2-1- Filtrage linéaire

Une technique simple, qui consiste à remplacer chaque point par une moyenne calculée sur son voisinage, Notons Pi, pixel considéré, sa valeur est:

$P_i=\Sigma \{A_j * P_j\}; j=1...n$, **Le masque (voisinage):**

X_4	X_3	X_2
X_5	X_0	X_1
X_6	X_7	X_8

Tel que : A_j représente les coefficients du masque utilisé. P_j représente les valeurs du voisinage considéré. [24]

4-2-1-1- Filtre Prewitt et filtre Sobel

Sont des filtres linéaire ou les coefficients du masque peuvent être positifs ou négatifs et leur somme égale a 0. Ce type de filtre est utilisé pour la détection et le renforcement des contours [25].

Filtre de Prewitt :

1	1	1
1	2	1
-1	-1	-1

Filtre Sobel :

Gx

-1	-2	-1
0	0	0
1	2	1

Gy

-1	1	1
-2	0	2
-1	0	1

4-2-2- Filtrage non linéaire

Affecte au pixel considéré une valeur particulière de son voisinage ; au lieu de la moyenne pondérée exemple [26]:

4-2-2-1- Filtre par élimination

Il se base sur la formule suivante : Si $\left|\frac{\sum_{i=0}^{i=8} X_i}{N} - X_0\right| > S$ alors $X_0 = \frac{\sum_{i=0}^{i=8} X_i}{N}$

4-2-2-2- Filtre de gaussien

Il est défini par l'équation suivante pour calculer les nouvelles valeurs de l'image améliorée :

ImagAméliorée = ImagOriginale * G ;

$$\text{Tel que : } G(x, y, \sigma) = \frac{1}{\sigma(\sqrt{2\pi})^2} * \text{Exp} - \frac{x^2}{2\sigma^2}$$

4-2-3- Filtrage morphologique

Elles englobent les transformations qui suivent :

-Transformation par érosion et Transformation par dilatation d'image (lissage d'image).

-L'ouverture et la fermeture morphologique appliquée à une image (chapitre III).

4-2-3-1- L'érosion : Si un pixel 'p' est noir (p(x, y)=0), et il y a au moins 3 pixels, ou 7 pixels dans les 4 -voisins, ou les 8-voisins respectivement, qui sont blancs (P voisin(x, y)=1), alors on affecte respectivement a ce pixel la couleur blanche (p(x, y)=1), c. -à-d on efface ce pixel, (C'est un lissage d'un ou deux pixels d'une forme connexe) (Figure I. 3) . [27]

4-2-3-2- Dilatation : C'est l'inverse de la transformation par dilatation, si un pixel 'p' est blanc (p(x,y)=1), et il y a au moins 3 pixels, ou 7 pixels dans les 4 -voisins, ou les 8- voisins respectivement, qui sont noir (P voisin(x,y)=0), alors on affecte a ce pixel la couleur noir (p(x,y)=0) (Figure I. 3). [27]

Figure I. 3 Image traitée par Dilatation et érosion

4-3- La Normalisation

Après la normalisation de la taille, les images de tous les caractères se retrouvent définies dans une matrice de même taille, Pour faciliter les traitements ultérieurs.

Cette opération introduit généralement de légères déformations sur les images. Cependant certains traits caractéristiques tels que la hampe dans des caractères peuvent être éliminées à la suite de la normalisation, ce qui peut entraîner à des confusions entre certains caractères (Figure I. 4). [28] [27]

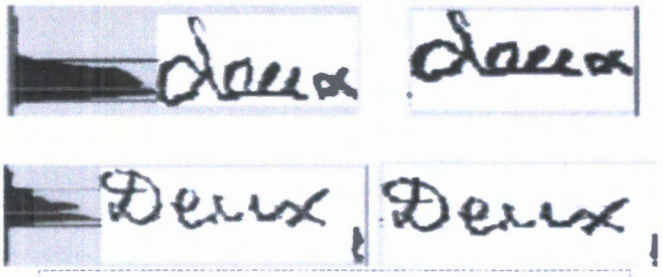


Figure I. 4: Exemple de normalisation de mots manuscrits

4-4- Extraction de composantes connexes

Une composante connexe (CXX) est un ensemble de points liés dans le plan. Elle peut correspondre à un point diacritique, un accent, au corps d'un caractère ou d'une chaîne de caractères... Une fois localisés les CXX sont regroupées pour former les mots. Cette technique est utilisée pour le repérage des points diacritiques dans les images de textes arabes [29] [28].

4-5- Extraction de contour

La chaîne de Freeman est la méthode la plus utilisée de description des contours dans les images [30]. C'est une technique de représentation des directions de contour (on code la direction le long du contour dans un repère absolu lors du parcours du contour à partir d'une origine donnée). Les directions peuvent se présenter en 4 connexités (codage sur deux bits) ou en 8 connexités (codage sur 3 bits). Les codes des contours sont donnés par la figure I.5 et figure I.6. [31]

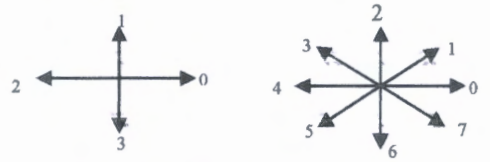
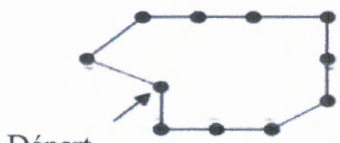


Figure I. 5: Le code de Freeman en 4-connexités (à gauche) et en 8 connexités (à droite).



Code 31000665442

Figure I. 6: Codage à l'aide du code de Freeman

4-6- Squelettisation d'une image

Très utilisée dans le domaine de la 'reconnaissance des formes', cette opération consiste à transformer une image binaire en un 'squelette'.

Le squelette est un ensemble de lignes d'épaisseurs infiniment petites. La squelettisation doit préserver la connexité de l'image (Figure I. 7).

En d'autres termes, cette opération ne doit ni séparer les éléments connexes, ni raccorder les éléments non connexes. Le but est de simplifier l'image du caractère en une image à « ligne » plus facile à traiter en la réduisant au tracé du caractère. [32]

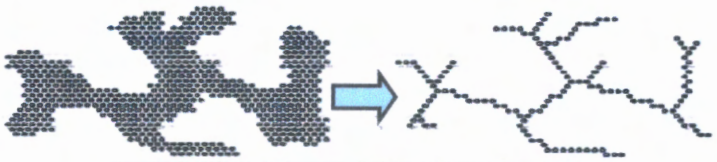


Figure I. 7: La squelettisation d'une Image.

5- Segmentation

En reconnaissance de l'écriture manuscrite, les données à traiter sont des images. La mise en œuvre d'une étape de segmentation permet de diviser l'image en différentes imagettes de taille moins importantes qui peuvent être des graphèmes, des lettres ou bien des sous mots. Cependant une imagette reste une matrice de pixels. [33] [9]

5-1- Segmentation explicite

La segmentation explicite, s'appuie sur un découpage à priori de l'image en sous-unités qui peuvent être des lettres ou des graphèmes (Figure -9 ci-dessous). Cette décomposition se base directement sur une analyse morphologique du texte ou de mot, ou sur la détection points caractéristiques tels que les points d'intersection, les points d'inflexion, les boucles à l'intérieur du texte ou de mot pour localiser les points de segmentation potentiels [34].

Dans le cas de segmentation en graphèmes, les textes ou les mots sont alors reconnus non comme une suite de lettres reconnues.

5-2- Segmentation implicite

La segmentation est dite implicite lorsque celle-ci est basée sur un moteur de reconnaissance pour valider et classer les hypothèses de segmentation (recherche de chemin des points de segmentation possibles) (Figure I-8 ci-dessous). Dans ce cas, la segmentation et la reconnaissance sont réalisées conjointement, d'où le nom parfois employé de "segmentation-reconnaissance intégrée". [20]

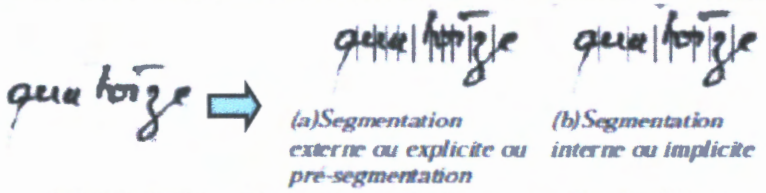


Figure I. 8: Différentes segmentation pour le mot quatorze [35]

6- Phase d'analyse ou d'extraction des caractéristiques

Les attributs caractéristiques aussi appelés primitives peuvent être groupés en trois classes selon qu'ils soient extraits à partir du mot entier (attributs de haut niveau), des lettres (attributs de moyen niveau) ou des lettres secondaires (attributs de bas niveau).

Les attributs de bas niveau sont extraits à partir des morceaux de lettres ; leurs formes sont élémentaires telles que de petites lignes, des courbures, des barres...Etc. Ils représentent, en général, des caractéristiques géométriques simples. Dans la plupart des cas, les primitives de bas niveau sont des déformations des éléments d'un ensemble de base de primitives où la déformation en elle-même pourra être employée comme primitive. Alors que les attributs à moyen niveau sont les attributs représentant la moyenne des régions locales et qui sont généralement préférables pour la segmentation explicite, tandis que les attributs de haut niveau comprennent la détection des éléments structuraux, ils ne dépendent pas du style d'écriture et sont donc stables par rapport à la variabilité cursive. Ils se sont généralement les boucles, les ascendantes et les descendantes d'un caractère [18].

7- Etape d'apprentissage

Avant de pouvoir prendre une décision, il faut acquérir des connaissances et les organiser en modèles de référence ou *classes*. C'est durant la phase d'apprentissage qu'est réalisé ce travail. Lorsque le concepteur (ou professeur) indique le nom de la forme d'entrée, l'apprentissage est dit supervisé. Si la construction des classes est automatique, on parle d'apprentissage non supervisé. [36] [37]

7-1- Apprentissage supervisé

L'apprentissage est dit supervisé si les différentes familles des formes sont connues a priori et si la tâche d'apprentissage est guidée par un superviseur ou professeur,

C'est-à-dire le concepteur, indique, pour chaque forme échantillon rentrée,

Le nom de la famille qui la contient. La tâche d'apprentissage tente de conserver ses liens de parenté en répartissant les familles dans des classes séparées entre elles. [18][36]

7-2- Apprentissage non supervisé

On l'appelle aussi, suivant l'approche utilisée, classification automatique, inférence ou encore apprentissage sans professeur. Il s'agit, à partir d'échantillon de référence et de règles de regroupement ou de modélisation, de construire automatiquement les classes ou les modèles sans intervention de l'opérateur. Ce mode d'apprentissage nécessite un nombre élevé d'échantillons et des règles de construction précise et non contradictoires pour bien assurer la formation des classes. [18]

8- Classification

Dans le processus complet d'un système de reconnaissance de formes, la classification joue rôle important en se prononçant sur l'appartenance d'une forme à une classe.

L'idée principale de la classification est d'attribuer un exemple (une forme) non connu à une classe prédéfinie à partir de la description en paramètres de la forme. (Les exemples des classifieurs dans le chapitre II)

9- Phase de post-traitement

L'objectif du post-traitement est l'amélioration du taux de reconnaissance des mots (par opposition au taux de reconnaissance du caractère). Cette phase est souvent implémentée comme un ensemble d'outils relatifs à la fréquence d'apparition des caractères dans une chaîne, aux lexiques et à d'autres informations contextuelles [29] [38].

10- Conclusion

Nous avons présenté dans ce chapitre une vision introductive, certains concepts généraux liés à la reconnaissance des caractères et de l'écriture en générale et comment préparer l'image pour les traitements ultérieurs.

CHAPITRE II :

Les méthodes De Classifications

1- Introduction

2- Méthodes Markoviennes

3- La Méthode du K Plus Proches Voisins

4- Arbre de Décision

5- Les Réseaux de Neurones

6- Classification Bayésienne

7- Classifieur Euclidien

8- Classifieur Quadratique

9- Conclusion

1- Introduction

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets a partir de certains traits descriptifs. Elles s'appliquent a un grand nombre d'activités Les humaines et conviennent en particulier au problème de la prise de décision automatisée. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples. Un exemple consiste en la description d'un cas avec la classification correspondante.

Citons quelques exemples de classification :

Un exemple courant d'application de la classification est le tri automatique du courrier en fonction du code postal ou de l'adresse manuscrite. Pour un système d'interprétation du code postal, on dispose au départ d'une base de données (un ensemble d'apprentissage) constituée de quelques milliers de chiffres, provenant de scripteurs différents, et identifiés, par un être humain, comme appartenant à une classe donnée, c'est-à-dire comme étant un 0, un 1,..., un 9. A partir de ces exemples munis d'une classe, ou étiquette, l'objectif est de calculer une fonction capable d'associer automatiquement une classe à un chiffre manuscrit ne provenant pas de l'ensemble d'apprentissage.

La classification est l'élaboration d'une règle de décision qui transforme les attributs caractérisant les formes en appartenance à une classe; passage de l'espace de représentation vers l'espace de décision Richard [39]. La classification consiste alors à identifier les classes auxquelles appartiennent les formes à partir des caractéristiques préalablement choisies et calculés. L'algorithme ou la procédure qui réalise cette application est appelé classifieur.

2- Méthodes Markoviennes

2-1- Introduction

Les modèles de Markov cachés sont une méthode utilisée depuis longtemps pour la modélisation de séquences. Leur simplicité s'applique avec succès à une grande diversité de tâches : la reconnaissance de la parole, la reconnaissance de l'écriture manuscrite ou imprimée ou encore la bio-informatique [40].

2-2- Définitions

2-2-3- Modèle De Markov Cachés

Etats: $S= \{S_1, S_2,..., S_N\}$

Processus passe d'un état à un autre, générant une suite où les éléments appartiennent à l'ensemble d'état $S= S_{i1}, S_{i2},..., S_{ik},...$

Propriété de la chaine de markov: probabilité dépend uniquement de l'état précédent: P (b_{ik} | S_{i1}, S_{i2},…, S_{ik-1}) = P (b_{ik} | S_{ik-1}).

Les états ne sont pas visibles, mais chaque état génère un état observable parmi un certain nombre: {v_1, v_2,...., v_M}.

Pour définir un modèle de markov, on doit définir: matrice des probabilités de transition A= ($a_{i,j}$), $a_{i,j}$= P (b_i | S_j), matrice des probabilités des états observables B= (b_i (v_m)), b_i(vm) = P (v_m | S_i), et un vecteur de probabilités initiales $\pi = \pi_i$, π_i = P (S_i). Le modèle est donné par M= (A, B, π) (voire figure II.1). [41] [42]

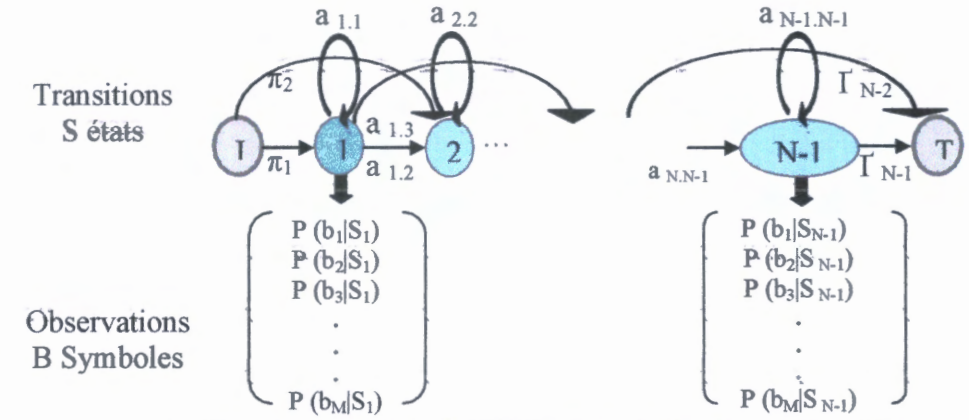


Figure II. 1: Définition des paramètres d'un Modèle de Markov Caché [42]

2-2-2- Topologie Des HMMS

Il y a deux types de HMMs. Ce sont le modèle ergodique et le modèle gauche-droite [44].

2-2-2-1- HMM Ergodique

C'est un modèle sans contraintes où toutes les transitions d'un état vers l'autre sont possibles (Figure II.2).

c'est-à-dire $a_{i,j} > 0 \ \forall (i,j) \in [1, N]$.

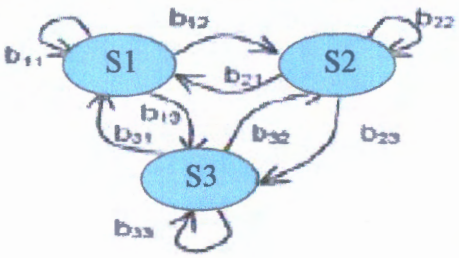


Figure II. 2: Modèle ergodique à 3 états [45]

2-2-2-2- HMM Gauche-Droite

C'est un modèle où il y a des contraintes sur des transitions :

Seulement la transition d'un état ayant un indice bas vers un état ayant un indice haut est acceptée (Figure II.1), c'est-à-dire: $a_{ij} = 0 \ si \ i > j \ \forall (i, j) \in [1.N]$

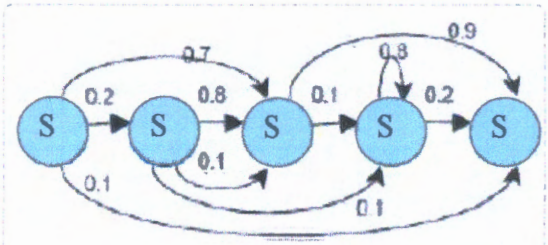


Figure II. 3: modèles gauche-droite à 4 états [45]

2-3- Modèles de Markov Caches « HMM Planaires»

Les HMM classiques, qui considèrent la forme comme un signal unidimensionnel, ont été utilisés avec succès en reconnaissance de l'écriture. Mais la nature 2D de l'écriture permet de penser que des améliorations plus importantes peuvent être apportées en étendant les HMM à deux dimensions. Ce qui a donné une extension des HMM appelée HMM pseudo-2D ou planaires (PHMM). En pratique, Les PHMMs sont des HMM où la probabilité d'observation dans chaque état est donnée par un HMM secondaire. L'architecture générale d'un PHMM inclut un modèle principal composé de super-états auxquels sont associés des modèles secondaires (voir figure II.4). [46]

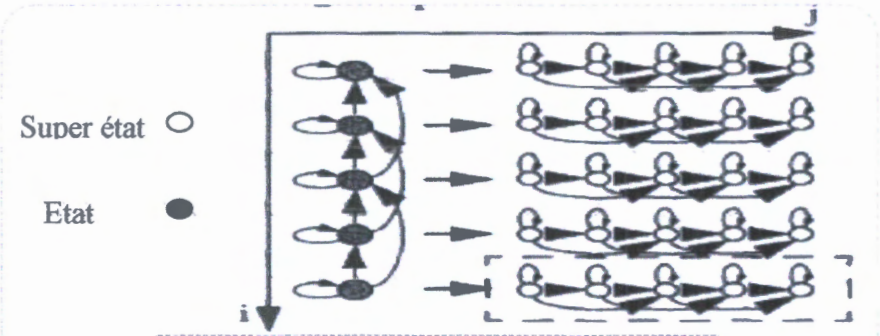


Figure II. 4: Exemple d'architecture d'un PHMM [47]

2-4- Les Fonctionnalités d'un HMM

2-4-1- L'Évaluation

2-4-1-1- Problème d'Évaluation

- Etant donné un HMM M= (A, B, π) et une séquence observée $O=o_1, o_2,...,o_K$,calculer la probabilité que M génère la séquence O.
- Essayer de trouver la séquence $O= o_1, o_2,..., o_K$ En considérant toutes les chaines possibles d'état est pas le plus intelligent en pratique, car il y a N_K séquences de K symboles pris parmi N(voir figure II.5). [48]
- Algorithme du **Forward-Backward** plus efficace [49]:

2-4-1-1-1- Forward

On définit la «forward variable» $\alpha_k(i)$ comme la probabilité d'observer la séquence d'observation partielle $o_1\ o_2\ ...\ o_k$ **AVEC** l'état caché à la date k qui est si :

$$\alpha_k(i) = P\ (o_1, o_2,... o_k, q_k= s_i)$$

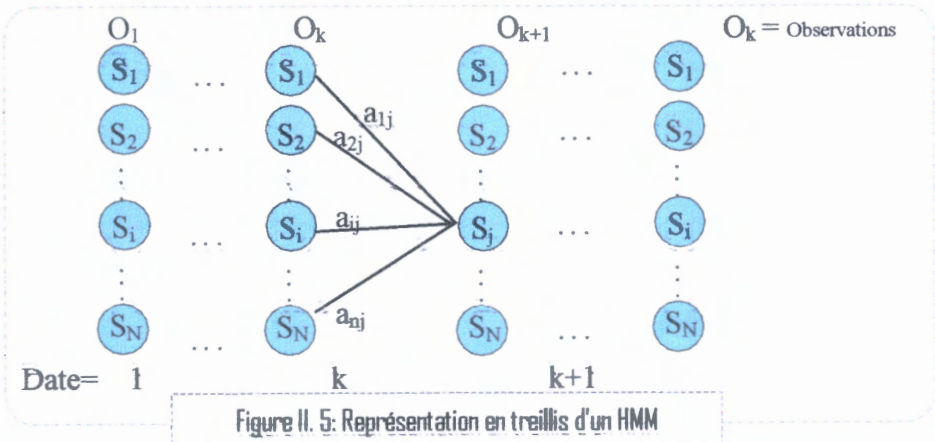


Figure II. 5: Représentation en treillis d'un HMM

Algorithme II. 1: La Récursion Forward Pour HMM

- **Initialisation:**

$\alpha_1(i) = P(o_1, q_1 = s_i) = \pi_i\, b_i(o_1)$, $1 <= i <= N$.

- **récursivité «en avant»:**

$\alpha_{k+1}(i) = P(o_1, o_2, \ldots o_{k+1}, q_{k+1} = s_j) =$
$\sum_i P(o_1, o_2, \ldots o_{k+1}, q_k = s_i, q_{k+1} = s_j) =$
$\sum_i P(o_1, o_2, \ldots o_k, q_k = s_i)\, a_{ij}\, b_j(o_{k+1}) =$
$\sum_i P\, \alpha_k(i)\, a_{ij}]\, b_j(o_{k+1})$, $1 <= j <= N$, $1 <= k <= K-1$.

- **Terminaison:**

$P(o_1, o_2, \ldots o_K) = +i\, P(o_1, o_2, \ldots o_K, q_K = s_i) = \sum_i a_k(i)$

- **Complexité :**

N^2K opérations.

2-4-1-1-2- Backward

On définit, $\beta_k(i)$ comme la probabilité d'observer la séquence partielle $o_{k+1}\, o_{k+2} \ldots o_K$ ET que l'état caché à la date k est si : $\beta_k(i) = P(o_{k+1}\, o_{k+2} \ldots o_K \mid q_k = s_i)$.

Algorithme II. 2: La Récursion Backward pour HMM

- **Initialisation:**

$\beta_k(i) = 1$, $1 <= i <= N$.

- **Backward récursion:**

$\beta_k(j) = P(o_{k+1}, o_{k+2}, \ldots o_K \mid q_k = s_j) =$
$\sum_i P(o_{k+1}, o_{k+2}, \ldots, o_k, q_{k+1} = si \mid q_k = s_j) =$
$\sum_i P(o_{k+2}, o_{k+3}, \ldots, o_K \mid q_{k+1} = s_i)\, a_{ji}\, b_i(o_{k+1}) =$
$\sum_i \beta_{k+1}(i)\, a_{ji}\, b_i(o_{k+1})$, $1 <= j <= N$, $1 <= k <= K-1$.

- **Terminaison:**

$P(o_1, o_2, \ldots, o_K) = \sum_i P(o_1, o_2, \ldots, o_K, q_1 = s_i) =$
$\sum_i P(o_1, o_2, \ldots, o_K \mid q_1 = s_i)\, P(q_1 = s_i) = \sum_i \beta_1(i)\, b_i(o_1)\, \pi_i$

2-4-2- L'apprentissage

Les méthodes d'apprentissage d'un HMM les plus utilisées sont : l'algorithme de Viterbi, et l'algorithme Baum-Welch [50].On va parler de Viterbi :

2-4-2-1- L'Algorithme De Viterbi

- Principe L'Algorithme de Viterbi « Backtracking » : [50]

Si un meilleur chemin se terminant en $q_k = s_j$ passe par $q_{k-1} =$ si alors il devrait coïncider avec le meilleur chemin terminant en $q_{k-1} = s_i$.

$\delta_k(i) = \max P(q_1 \dots q_{k-1}, q_k = s_j, o_1\ o_2 \dots o_k) = \max_i [a_{ij}\ b_j(o_k) \max P(q_1 \dots q_{k-1} = s_i, o_1\ o_2 \dots o_{k-1})]$

Pour remonter le chemin, garder l'info que le prédécesseur de s_j était s_i.

Algorithme II. 3: De Viterbi « Backtracking »

- **Initialisation:**

  $\delta_k(i) = \max P(q_1 = s_i, o_1) = \pi_i\ b_i(o_1),\ 1 <= i <= N.$

- **Récursion avant:**

  $\delta_k(j) = \max P(q_1 \dots q_{k-1}, q_k = s_j, o_1\ o_2 \dots o_k) =$
  $\max_i [a_{ij}\ b_j(o_k) \max P(q_1 \dots q_{k-1} = s_i, o_1\ o_2 \dots o_{k-1})] =$
  $\max_i\ a_{ij}\ b_j(o_k)\ \delta_{k-1}(i)],\ 1 <= j <= N,\ 2 <= k <= K.$

- **Terminaison:**

  choix de la « meilleure fin » (probabilité maximale) à la date K maxi $[\delta_K(i)]$

- On reconstruit le meilleur chemin.

Algorithme similaire à la récursion « Backward » avant du problème d'évaluation, en substituant $\sum$ par max, associé à une remontée de chemin.

2-4-3- Décodage

2-4-3-1- Problème Du Décodage

Etant donné $M = (A, B, \pi)$ et la séquence observée $O = o_1, o_2, \dots o_K$, déterminer la séquence d'état caché la plus probable qui a généré cette séquence.

On cherche la séquence d'état $Q = q_1 \dots q_K$ qui maximise $P(Q \mid o_1, o_2, \dots o_K)$, ou, de manière équivalente $P(Q, o_1\ o_2 \dots o_K)$.

Evaluation brutale se fait en temps exponentiel. On peut aussi utiliser l'algorithme de **Viterbi**.

On défini $\delta_k(i)$ comme la probabilité maximale de générer la séquence d'observations $o_1\ o_2 \dots o_k$ quand on parcourt la séquence d'états cachés $q_1 \dots q_{k-1}$ et en arrivant à $q_k = s_i$.

$\delta_k(i) = \max P(q_1 \dots q_{k-1}, q_k = s_i, o_1\ o_2 \dots o_k)$

Le max étant pris sur tous les chemins possibles $q_1 \dots q_{k-1}$. [51]

2-5- Avantages et Inconvénients

- **Avantage :** Existence d algorithme puissants (tel que Viterbi) permettant de déterminer la solution optimale.
- **Inconvénient :** Nécessite de corpus de tailles assez volumineuses. [52]

3- La Méthode du K Plus Proches Voisins

3-1- Introduction

KPPV (ou *k*-NN: k-Nearest-Neighbor en anglais) est une méthode d'apprentissage supervisée et non paramétrique, c'est-à-dire elle ne demande pas une phase d'apprentissage de paramètres. Ce type de classifieur non paramétrique est mis en œuvre, particulièrement, dans le cas où l'on ne dispose pas de connaissances a priori sur la distribution de probabilité des classes [53]. Dans [54], les auteurs suggèrent l'utilisation d'un *KPPV* comme une première tentative dans un nouveau problème de classification.

Un *KPPV* est basé sur l'hypothèse que les points proches dans l'espace de primitives sont susceptibles d'appartenir à la même classe.

3-2- Définitions

3-2-1- Principe

Son principe de décision consiste tout simplement à calculer la distance d'une forme inconnue x à tous les échantillons fournis. Puis on sélectionne les k plus proches échantillons et on affecte x à la classe majoritaire parmi ces k échantillons, et le choix de l'entier k et de la métrique de distance jouent un rôle primordial dans la méthode *KPPV*.

La figure II.6 a représente un problème de classification à 2 classes avec un algorithme *kppv* où $k = 1$ (a) et $k = 3$ (b). Les données étiquetées appartenant à la classe 1 et la classe 2 sont respectivement représentées par des carrés blancs et noirs. Une observation non étiquetée A (donnée à classer), est représentée par un carré gris. Afin de classer la donnée A dans un voisinage de k= 1, on recherche le plus proche voisin de A. Le cercle entoure le point à classer et son plus proche voisin. Le plus proche voisin de A est un point de la classe 1, d'où A sera donc affecté à la classe 1. [54]

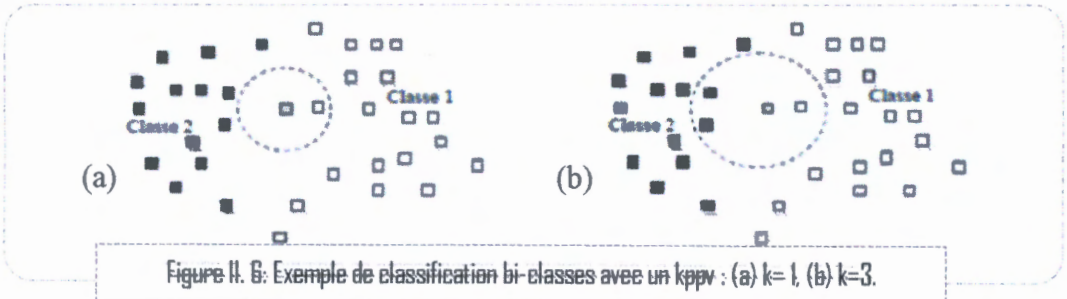


Figure II. 6: Exemple de classification bi-classes avec un kppv : (a) k= 1, (b) k=3.

3-2-2- Formalisation mathématiques

Soit :

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

$$\text{Argmax } y\in Y = \sum_{(x_i ,y_i)\in(x,T,d(.,.),k)} Iyi = y$$

Notons par $Xp = (x_{p1}; x_{p2}; \ldots; x_{pN})$ le vecteur caractéristique de l'entité p, avec N le nombre de caractéristiques et par p et q deux entités à comparer.

Les distances suivantes sont usuellement employées par les classifieurs Kppv : [55]

Distance Euclidienne : $D(X_p;X_q) = \sqrt{\sum_{i=1}^{N}(x_{pi} - x_{qi})^2}$

Distance de Manhattan : $D(X_p;X_q) = \sum_{i=1}^{N}|xpi - xqi|$

Distance de Minkowski : $D(X_p;X_q) = (\sum_{i=1}^{N}(x_{pi} - x_{qi})^r)^{1/r}$

Distance de Tchebychev : $D((X_p;X_q) = \max_{i=1}^{N}(|x_{pi} - x_{qi}|)$

3-3- Avantages et Inconvénients

La robustesse et la simplicité de la mise en œuvre sont les principaux avantages du *KPPV*. Cependant, Son efficacité dépend directement de la pertinence de la base d'apprentissage et notamment de sa densité dans les différentes régions de l'espace de données.

Son principal inconvénient est lié à une vitesse de classification faible dû au nombre important des distances à calculer .Outre, le *KPPV* est réputé comme classifieur lourd. La recherche des plus proches voisins est coûteuse, cela d'autant plus que la métrique utilisée est complexe, et que la base et la valeur de *k* sont grandes. Cependant, pour des problèmes particuliers (tels que la reconnaissance de chiffres), on peut réduire le nombre de données d'apprentissage et les organiser (prendre les échantillons les plus significatifs) [44].

3-4- Algorithme

Généralement la norme euclidienne est souvent employée comme mesure de distance, dans chaque étape de l'apprentissage, L'algorithme mémorise les k meilleurs exemples de l'ensemble d'apprentissage (kppv(x)) qui sont proches à l'exemple de test x.

Cet algorithme est souvent performant s'il y a suffisamment d'exemples d'apprentissage, mais demande un temps de prédiction très long pour passer tous les exemples afin de trouver les K meilleures solutions.

Soit L = $\{(x', c)|x' \in R^d, c \in C\}$ est un ensemble d'apprentissage. (x' est un exemple d'apprentissage et c sa classe qui lui est associée. [53]

Algorithme II. 4: De K Plus Proches Voisins

Début
  Pour chaque *(exemple* $(x', c) \in L$*)* **faire**
    Calculer la distance $D(x, x')$
  Fin
  Pour chaque $\{x' \in kppv(x)\}$ **faire**
    Compter le nombre d'occurrences de chaque classe
  Fin
    Attribuer à x la classe la plus fréquente
Fin

Soit x l'exemple de test dont on souhaite déterminer sa classe.

4- Arbre de Décision

4-1- Introduction

Les Arbres de décision sont des règles de classification basées sur des tests associés aux attributs organisés de manière arborescente : [56]

4-2- Explications

Le formalisme des arbres de décision permet de classier un nouvel objet en testant ses caractéristiques les unes à la suite des autres. La classification se fait à travers une séquence de questions dans laquelle chaque question dépend de la réponse à la question précédente. Cette séquence de questions est représentée par un arbre de décision dont les feuilles terminales représentent les classes. La figure II.2 illustre un exemple de classification sur des données continues en deux dimensions en utilisant les arbres de décision.

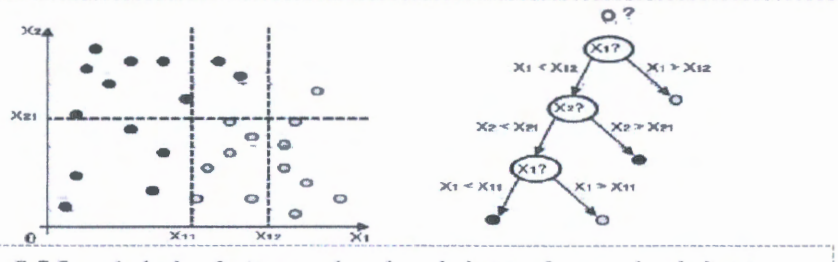


Figure II. 7: Exemple de classification avec les arbres de décision dans un arbre de décision

Dans un arbre de décision, chaque nœud de l'arbre fait la décision si une certaine donnée appartient à un certain ensemble de classe.

La succession des nœuds fait réduire le nombre de classes jusqu'à ce qu'on arrive à une seule classe. [57]

Chaque nœud de l'arbre contient un classifieur destiné à distinguer entre deux classes seulement parmi les quatre. A chaque niveau, on élimine une classe jusqu'à ce qu'une classe reste. Le choix d'une stratégie se fait par rapport au problème traité car il n'y a aucune étude comparative permettant de favoriser l'une par rapport à l'autre. La troisième méthode construit le minimum de classifieurs SVM, mais elle n'est pas évidente à mettre en œuvre car pour certains problèmes, il est difficile de trouver les dichotomies.

4-3- Avantages et Inconvénients

Les arbres de décisions sont très répandus, à cause de la simplicité de lecture de leurs résultats et leur traitement naturels des cas multi-classe. Néanmoins, ils posent beaucoup des problèmes tels que [58]:

– La difficulté de manipulation des attributs numériques,

– L'espace nécessaire pour leur déduction,

– Leur non scalabilité.

5- Les Réseaux de Neurones

5-1- Introduction

Selon [59]:

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

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

Selon [60]:

Un réseau de neurones est un circuit composé d'un nombre très important d'unités de calcul simples basées sur des neurones. Chaque élément opère seulement sur l'information locale.

Chaque élément opère de façon asynchrone; il n'y a donc pas d'horloge générale pour le système.

5-2- Définitions

5-2-1- Un Neurone Formel

Peut être considéré comme une modélisation élémentaire d'un neurone biologique(figure II.8).

Le neurone reçoit en entrée un vecteur d'attributs numérique présentant la description d'une observation x, les éléments de ce vecteur xi sont pondérés par des poids synaptiques w_i, un biais w_0 est également ajouté. La sortie y du neurone est obtenue par l'application d'une fonction de transfert appelée aussi fonction d'activation [61]: $Z = \sum_{i=1}^{d} w_i x_i + w_0$ $y = f(z)$

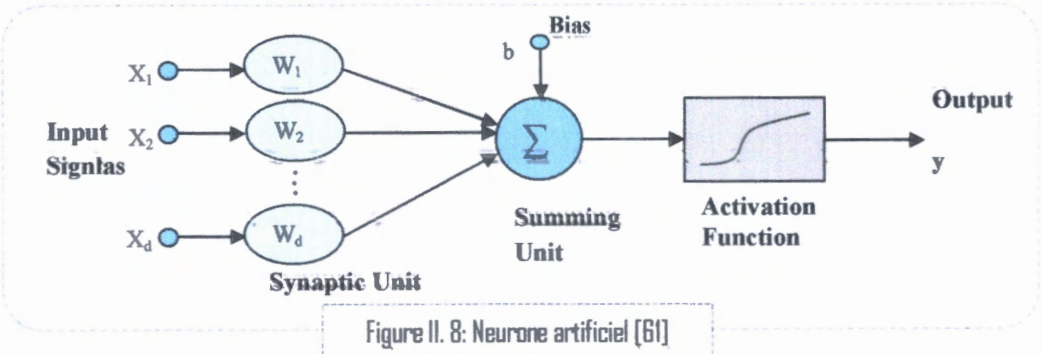


Figure II. 8: Neurone artificiel [61]

5-2-2- Le Réseaux de Neurones Artificiel

RNA est un réseau fortement connecté de processeurs élémentaire (neurones) fonctionnant en parallèle et disposant en couches. Tous les neurones d'une même couche ont la même fonction d'activation (voir tableau II.1). [53]

Les types de RNA sont aussi nombreux que leur définition est générale, ils se distinguent globalement par la fonction d'activation des neurones (tableau II.1), l'architecture du réseau (organisé ou non en couches, avec ou sans cycles) et le mode de connectivité (complètement ou localement connecté) (voir figure II.9).

Dans la partie suivante, nous présenterons rapidement les deux types de RNA les plus populaires : le Perceptron Multicouches (PMC) et le réseau RBF (Radial Basis Function).Aussi existe D'autre types tels que : les réseaux à convolution, carte de Kohonen et les réseaux polynomiales. [61]

Fonction lineaire	$\sum_i w_i \cdot x_i + w_0$
Fonction sigmoide	$\frac{1}{1+e^{-\lambda x}}$
Fonction tanh	$\frac{e^{2x}-1}{e^{2x}+1}$
Fonction softmax	$\frac{e^{x}}{\sum_i e^{x_i}}$
Fonction a base radilae de centre x_c	$\exp\left[-\frac{\|x-x_c\|^2}{2\sigma^2}\right]$

Tableaux II. 1: Quelques fonctions de transfert usuelles. x est le vecteur d'entrée

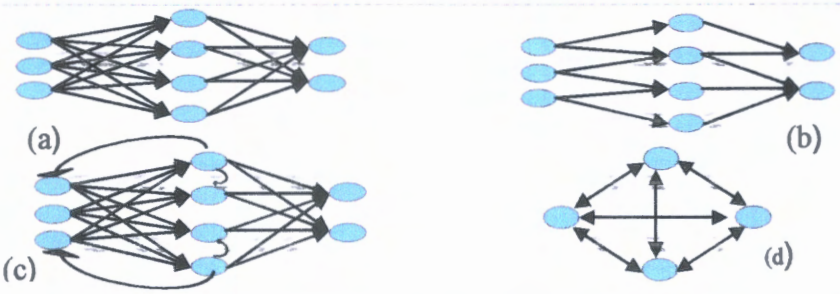


Figure II. 9: Le modèle d'un mot est la concaténation des modèles HMMs des caractères le composant. (a) réseaux multicouches; (b) à connexions locales; (c) à connexions récurrentes ; (d) à connexions complètes

5-3- Perceptrons Multicouches

Les perceptrons multicouches sont les réseaux de neurones les plus populaires et les plus simples. Ce sont des réseaux à propagation directe sans cycle, avec au mois une couche cachés, les neurones sont généralement complètement connectés et la fonction de transfert est de type sigmoïde (valeur dans [0,1]), tanh (valeur dans [-1,1]) ou softmax.

Ce classifieur a trouvé application dans beaucoup de domaines tels que la reconnaissance de caractères, la reconnaissance de visages, la prédiction, etc. notamment grâce à sa performance très intéressante, à son pouvoir de généralisation et à sa rapidité en phase de décision. Toutefois, son utilisation est liée à un certain nombre de problèmes (sur apprentissage, minima local, etc.).

Une description détaillée de ces problèmes est donnée dans [62].

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

1. $\forall i \in \{1,\ldots, C\}$, chaque couche possède neurones et entrées.
2. $\forall i \in \{1,\ldots, C-1\}$, $n_i = p_{i+1}$, de plus $p_1 = p$ et $n_c = n$.
3. les coefficients de la couche C_i sont notés $(w_1^i,\ldots, w_{ni}^i)$ cette couche définit une fonction F_i.
4. soit la suite $(Z_i)_{0\le i\le C}$ définie par: $Z_0 \in \forall i \in \{1,\ldots, C\}$, $Z_i = F_i(w_1^i,\ldots, w_{ni}^i, Z_{i-1})$.

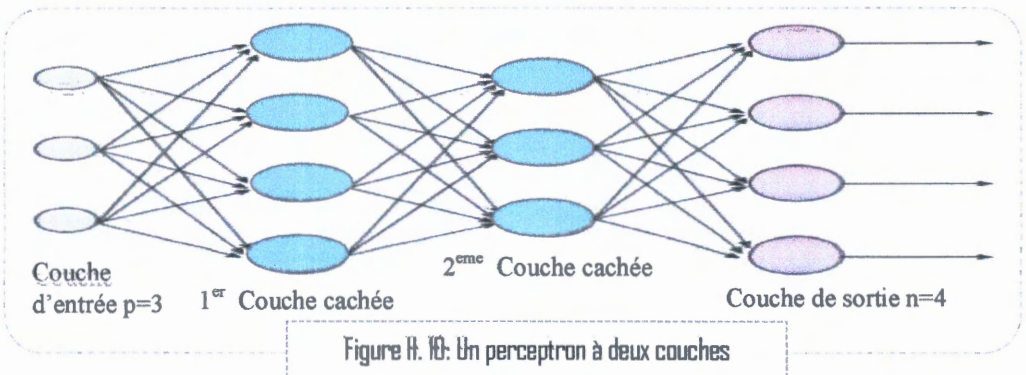


Figure II. 10: Un perceptron à deux couches

Souvent, on considère que les entrées forment la couche C_0 de manière à simplifier les écritures. Ainsi, chaque couche C_i du perceptron a pour entrées les sorties de la couche C_{i-1} .

Le mécanisme qui permet de calculer les sorties d'un réseau de neurones sachant ses poids est appelé *propagation* [62].

5-3-1- La Propagation

Cet algorithme s'applique à un réseau de neurones pour calculer les sorties de ce réseau connaissant ses poids ($W_{c,i,j}$) et ses entrées (x_j). [62]

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

Algorithme II. 5: De Propagation

Etape A : initialisation
Pour i=0 à C_0 faire
  $Z_{0,i} \leftarrow x_i$
Fin Pour
Etape B : Propagation
Pour c=1 à C **faire**
  Pour i=1 à C_c **faire**
    $Z_{c,i} \leftarrow 0$
    Pour j=1 à C_{i-1} **faire**
      $Z_{c,i} \leftarrow Z_{c,i} + W_{c,i,j}\, Z_{c-1,j}$
      Fin Pour
      $Z_{c,i} \leftarrow f(Z_{c,i} + b_{c,i})$
    Fin Pour
  Fin Pour
Fin Pour

5-4- Avantages et Inconvénients

L'avantage de ce modèle est le gain de temps considérable. Cependant, l'utilisation d'exemples pour apprendre apporte le risque de ne pouvoir résoudre que des situations déjà rencontrées, où un phénomène de sur-apprentissage qui spécialiserait le réseau uniquement sur les exemples connus sans généraliser. [63]

Par contre, ils représentent des problèmes remarquables qui ont limité leur évolution en face d'autres techniques tel que les SVMs [58]:

- Un réseau de neurones artificiel représente une boîte noire, et il est très difficile voire impossible de l'analyser et comprendre son fonctionnement en face d'un problème donné,

Ce qui empêche de choisir la structure (type, nombre de nœuds, organisation, connexions,...etc.) la mieux adaptée à ce problème.

- L'ordre de présentation des exemples d'entrainement au réseau influe directement sur les résultats obtenus. Pour surmonter ce problème, il est nécessaire de répéter au moins la phase d'entrainement avec plusieurs ordres différents des exemples, ce qui augmente considérablement le temps d'apprentissage.
- Dans le cas des bases de données, les réseaux de neurones artificiels ne permettent pas de traiter des exemples avec des attributs symboliques qu'après un encodage adapté, à l'inverse de plusieurs autres techniques d'apprentissages tel que les SVMs et les arbres de décision.

6- Classification Bayésienne

6-1- Introduction

Un classifieur probabiliste linéaire simple basée sur le théorème de Bayes qui suppose que les descripteurs (attributs) qui décrivent les objets de l'ensemble d'apprentissage sont indépendants.

La règle de décision bayésienne est un cas particulier de la règle général de décision avec coût, celle-ci consiste à associer à chaque classe un coût d'erreur de décision. Le critère à minimiser est, dans ce cas, le coût d'erreur qui traduit ainsi le risque de décision prise. La décision bayésienne associe, par exemple, un coût unitaire à chacune des erreurs et un coût nul à chacune des bonnes réponses. L'objectif est donc de construire un système dont la probabilité d'erreur globale est minimale. [64], cité dans [44]

6-2- Définition

La règle de Bayes permet de calculer ce qu'on appelle les probabilités a posteriori des classes. C'est à dire $P(w_i/x)$ à partir des probabilités a priori des classes $P(wi)$ et des fonctions de densité des probabilités $p(x/w_i)$.

$$P(w_i/x) = \frac{P(x/w_i) * P(w_i)}{P(x)} \quad \forall i=1,2 \quad \text{avec} \quad p(x) = \sum_{j=1}^{2} P(x/w_j) * P(w_j).$$

L'interprétation de $P(wi/x)$ est la suivante: ayant obtenu une forme (décrite par une valeur x de) $P(w_i/x)$ donne la probabilité d'avoir alors la classe w_i.

La règle de décision Bayésienne est alors pratiquement immédiate :

Par exemple, pour $i=1,2$, ayant remarqué une forme inconnue x (la forme est assimilée au paramètre qui la caractérise) on dira que la forme inconnue x est de la classe w_1 si :

$$P(w_1/x) > P(w_2/x)$$

Sinon elle est de la classe w_2 Soit sous une forme simplifiée:

REGLE DE DECISION BAYESIENNE

Ayant observé x on décide :

Si $P(w_1/x) > P(w_2/x)$
| w_1
Sinon
| w_2
Fin si

Ou encore

Si $P(x/w_1) * P(w_1) > P(x/w_2) * P(w_2)$
| w_1
Fin si

Dans cette règle on voit de façon particulièrement claire qu'il est nécessaire et suffisant de connaître pour toutes les classes cherchées la probabilité a priori et la loi de densité de probabilité de cette classe. [09]

6-3- Le Classifieur Bayésien Naïf

Si les attributs " v_i " de l'exemple « e » sont indépendant l'un de l'autre, le classifieur est appelé "classifieur de Bayes naïf " et la probabilité $P(e \mid C_i)$ est exprimée par le produit

$P(v_1|C_i) * ... * P(v_a|C_i)$.

On choisit pour « e » la classe Ci qui maximise la probabilité reformulée comme suit [65] :

$$P(C_i|e) = \frac{P(C_i)}{P(e)} \prod_{j=1}^{a} P(V_j|C_i)$$

Étant donné un ensemble d'apprentissage $X = (x_1, x_2, \ldots, x_n)$ où $x_i = (v_{i1}, v_{i2}, \ldots, v_{ip})$. Pour un nouvel élément $x = (v_1, \ldots, v_p)$: Le classifieur de Bayes estime les probabilités P(c) et P (x|c) à partir le corpus d'apprentissage X et puis emploie une règle de décision connue sous : *maximum a posteriori (MAP)* qui sélectionne l'hypothèse la plus probable. La classe correspondante est la fonction *CMAP* définie comme suit [66]:

$$C_{map} = argmgax_c\, p\left(\frac{c}{x}\right) = argmgax_c \frac{p\left(\frac{x}{c}\right) * P(C)}{P(X)} = argmgax_c\, p\left(\frac{x}{c}\right) P(C)$$

7- Classifieur Euclidien

Il s'agit de l'un des plus simples classifieurs qui puissent être conçus. La classe dont le vecteur de caractéristiques moyen est le plus proche, au sens de la distance euclidienne, du vecteur de caractéristiques de l'objet à classifier est assignée à ce dernier. Les fonctions discriminantes utilisées sont donc de la forme suivante [41] [67] :

$$\Phi i\,(X) = -1/2\,(X - u_i)^T\,(X - u_i)$$

Où $u_i = E\,[X / w_i]$ est le vecteur de caractéristiques moyen des éléments qui appartiennent à la classe w_i, $E\,[.]$ désignant l'opérateur d'espérance mathématique,

Et $(.)^T$ celui de transposition.

Le terme quadratique $X^T X$ est indépendant de la classe de l'objet, et les fonctions discriminantes peuvent également s'écrire:

$$\Phi i\,(X) = u_i^T X - \frac{1}{2} u_i^T u_i$$

Les frontières qui séparent les classes dans l'espace R^d sont ici linéaires. [68]

8- Classifieur Quadratique

Comme le nom l'indique, les frontières de décision fournies par ce type de classifieur sont décrites par des fonctions discriminantes quadratiques. Les fonctions discriminantes s'expriment [67]:

$$\Phi_i\,(X) = -1/2\,(X - u_i)^T \sum_i^{-1} (X - u_i)$$

Où $\sum_i = E\,[(X - u_i)^T\ (X - u_i)/w_i]$ est la matrice de covariance des vecteurs de caractéristiques de classe *wi*. En pratique, les vecteurs de caractéristiques moyens et les matrices de covariance ne peuvent qu'être estimées à partir des objets disponibles. Une estimation non biaisée à partir d'un ensemble fini de prototypes de chaque classe est donnée par [42]:

$$\hat{u}_i = \frac{1}{Ni} \sum_{X_K \in w_i} X_K \quad \text{et} \quad \widetilde{\sum} i = \frac{1}{Ni-1} \sum_{X_K \in w_i} (X_K - \hat{u}_i)\,(X_K - \hat{u}_i)^T$$

Où N_i est le nombre total d'objets de classe *wi* et X_k les vecteurs de caractéristiques qui représentent ces objets.

Dans le cas particulier où les composantes des vecteurs de caractéristiques ne sont pas corrélées entre elles, les matrices de covariances expérimentales sont diagonales.
L'expression des fonctions discriminantes se réduit alors à:

$$\Phi i\,(X) = -\frac{1}{2} \sum_{j=1}^{d} \frac{(x_j - \hat{u}_{ij})^2}{\tilde{\delta}_{ij}^2}$$

Où $\hat{u}_{ij}$ et $\tilde{\delta}_{ij}^2$ représentent respectivement la moyenne et la variance expérimentales de la jème composante du vecteur X, calculées sur l es éléments de la classe w_i.

9- Conclusion

Dans ce chapitre, L'état de l'art des méthodes de traitement d'écriture est donnée: Les Propriétés des Modèles de Markov Cachées à cause de son large utilisation dans le domaine de la reconnaissance de l'écriture, puis une description et un rappel de l'algorithme de k plus proches voisins, ensuite quelques propriétés de méthode de l'arbre de décision et de réseaux de nuerons aves leurs avantages et inconvénients . Enfin des descriptions des classifieurs Bayésien, euclidien et quadratique.

CHAPITRE III :

Le classifieur SVM

1- Introduction

2- Définition

3- Séparateur linéaire

4- Séparation non linéaire

5- L'Algorithme SMO (Sequential Minimal Optimization)

6- Les conditions de Karush-Kuhn-Tucker (KKT)

7- Calculer le seuil b

8- Architecture générale d'un SVM

9- Stratégie multi-classes

10- Conclusion

1- Introduction

Le classifieur SVM a été conçu pour une séparation de deux ensembles de donnée. Il est considéré donc comme un classifieur binaire.

Le but de SVM est de trouver un hyperplan qui va séparer et maximiser la marge de séparation entre deux classes. Le problème de recherche de l'hyperplan séparateur possède une formulation duale. Ceci est particulièrement intéressant car, sous cette formulation duale, le problème peut être résolu au moyen de méthode d'optimisation quadratique.

Différents algorithmes d'optimisation ont été développés parmi eux l'algorithme SMO (optimisation minimale et séquentielle) qui est un algorithme rapide proposé par platt [58] pour résoudre le problème de programmation quadratique des SVM.

2- Définition

Les SVM sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de la classification binaire et de la régression. Les SVM reposent sur deux idées principales: la notion de la marge maximale et la notion de la fonction noyau. La marge maximale est employée pour les problèmes de la classification linéaire. Elle représente la distance entre la frontière de séparation et les échantillons d'apprentissages les plus proches. Ces derniers sont les vecteurs supports. Les fonctions noyau sont employées dans le cas des problèmes de la classification non-linéaire pour transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension dans lequel il est probable qu'il existe de séparateurs linéaires [69].

Le classifieur SVM est un algorithme qui maximise la marge entre les classes du problème à résoudre et réduit au minimum l'erreur de classification.

L'objectif de la marge maximale est de faire séparer deux classes par un hyperplan de telle sorte la distance par rapport aux vecteurs supports soit maximale [70].

3- Séparateur linéaire

3-1- Cas linéairement séparable (SVM à Marge Dure)

L'hyperplan optimal de séparation est un classifieur de marge maximale qui sépare au mieux les deux classes et se trouvant à mi-distance des deux hyper plans H_1 et H_2 parallèles à H, d'équation respectives (voire Figure III.1): $H_1=w^t x + b > 0$, $H_2= w^t x+ b < 0$

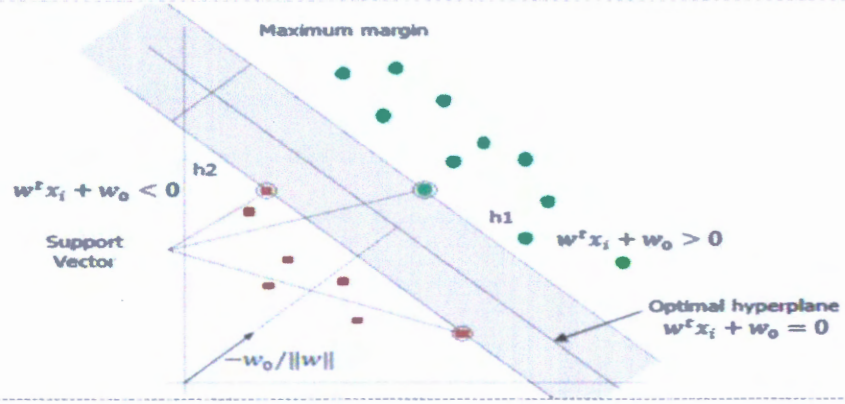


Figure III. 1: Nuage de données d'apprentissage avec un Hyperplan optimal et une marge maximale

Telle que :

- Il n'y a aucun point qui se situe entre H_1 et H_2
- Cette contrainte se traduit par les inégalités :

  $w^t x_i + b > 0$ si $y_i = 1$ Et $w^t x_i + b < 0$ si $y_i = -1$
- Ces deux inégalités peuvent être combinées en une seule : $y_i (w^T x_i + b) \geq 1$
- La distance ou la marge entre H_1 et H_2 est maximale.
- Dans ce cas, la distance entre H_1 et H_2 est donné par $M = \frac{2}{\|w\|}$
- Maximiser M revient à minimiser $\|w\|$ ou à minimiser $\|w\|^2$ avec :

  $\|w\| = w^T w$ (carré de la norme euclidienne du vecteur).

Le problème de séparation par hyperplan optimal peut être formulé comme suit :

$$\begin{cases} \text{Minimiser } \frac{1}{2}\|w\|^2 \\ \text{Sous contraintes : } \quad y_i (w^T x_i + b) \geq 1 \quad i = 1 \ldots n \end{cases}$$

Avec yi ∈ {=1,1} représente l'étiquette d'un modèle d'apprentissage x_i. (y_i =1 pour la classe 1, et -1 pour la classe 2).

La solution de ce problème correspond au point extrême du lagrangien primal [71]:

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

Où α_i sont les multiplicateurs de Lagrange. L'optimum de la fonction objective L_p peut être obtenu en la minimisant par rapport à w et b et en la maximisant par rapport aux α_i. À l'optimum de la fonction objective, ses dérivées par rapports aux variables w et b s'annulent ainsi que le produit des α_i aux contraintes (2) :

$$\begin{cases} \frac{\partial L(w,b,\alpha)}{\partial w} = 0, \\ \frac{\partial L(w,b,\alpha)}{\partial b} = 0, \\ \alpha_i [y_i (w^T x_i + b) - 1] = 0, \quad \alpha_i \geq 0 \end{cases} \quad (2)$$

De (2) on déduit :

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad \text{et} \qquad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3}$$

On remplacons dans (1), on obtient :

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \; y_i \; y_j \; x_i \; x_j$$

Cette fonction appelée Wolf Dual [28] permet de formuler le dual du problème d'optimisation :

$$\begin{cases} \text{Maximisation } \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \; y_i \; y_j \; x_i \; x_j \\ \text{Sous contraintes } \sum_{i=1}^{n} \alpha_i y_i = 0 \quad , \quad \alpha_i \geq 0 \end{cases}$$

Trouver un séparateur linéaire optimal revient à résoudre ce problème de programmation quadratique ou les α_i sont calculable est le **w** déduits a partir de l'équation (3).

On remplaçant la valeur de w de l'équation (3) dans la fonction de décision on obtient [71] [69]:

$$f(x) = \sum_{i=1}^{n} y_i \alpha_i \; (x_i, x) + b \tag{4}$$

Où " x_i " sont les vecteurs supports (SVs) lorsque les multiplicateurs de Lagrange (α_i) sont différent de zéro. Les SVs sont les modèles d'apprentissage qui se trouvent sur les frontières de la marge [71], α_i et b sont déterminés en résolvant un problème quadratique [74]. Dans la pratique, on utilisera les librairies SVM-Light de Joachim ou la méthode SMO implémentée par Platt [58].

3-2- Cas linéairement non séparable (SVM a Marge Souple)

La marge souple est considérée comme une relaxation de la marge dure justifiée par la présence des exemples mal classifiés appartenant a la marge (dite erreur de marge [71]).

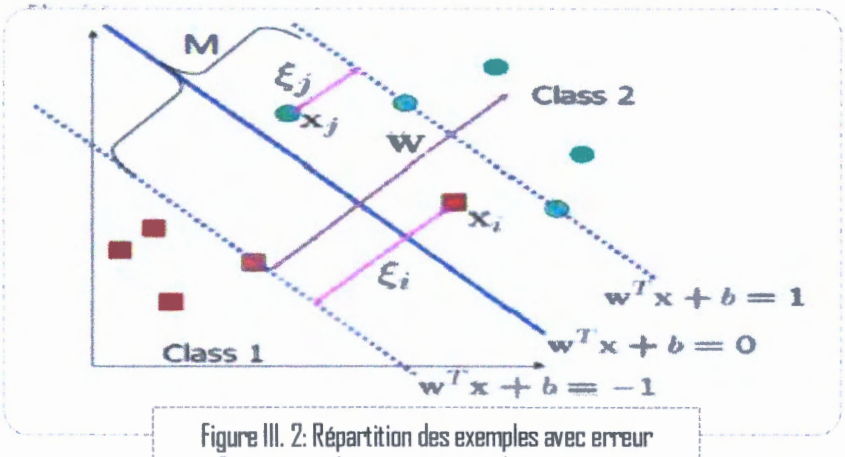


Figure III. 2: Répartition des exemples avec erreur

L'hyperplan optimal séparant les deux classes est celui qui sépare les données avec le minimum d'erreurs (voire Figure III.2), et satisfait donc comme suite [72] :

On relâche la contrainte de bon classement, initialement : $y_i (w^t x_i + b) > 1$

Devient: $y_i (w^t x_i + b) > 1 - \xi_i$

Et on ne doit plus minimiser : $\frac{1}{2} \|w\|^2$

Mais : $\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i$ Avec : $C \geq 0$.

C : représente une constante déterminante du compromis entre les deux objectifs opposés la minimisation de l'erreur et la maximalisation de la marge, la sélection de C reste intuitive vue qu'aucune méthode n'a était introduite pour le faire. [73]

La formulation duale du problème est similaire à celle du cas linéairement séparable sauf que les multiplicateurs de Lagrange deviennent bornés par C.

$$\begin{cases} \text{Maximisation } \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \; y_i \; y_j \; x_i \; x_j \\ \text{Sous contraintes } \sum_{i=1}^{n} \alpha_i y_i = 0 \quad , \quad 0 \leq \alpha_i \leq C \;\; i=1 \ldots m \end{cases} \tag{4}$$

La seule différence avec la SVM à marge dure est que les α_i ne peuvent pas dépasser C, La fonction de décision est alors calculée de la même manière que dans le cas des SVM à marge dure mais uniquement à base des vecteurs supports non bornés par :

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

4- Séparation non linéaire

4-1- Cas non linéairement séparable

Dans le cas linéaire, on pouvait transformer les données dans un espace où la classification serait plus aisée. Dans ce cas, l'espace de redescription utilisé le plus souvent est R (ensemble des nombres réels).

Il se trouve que pour des cas non linéaires, cet espace ne suffit pas pour classer les entrées. On passe donc dans un espace de grande dimension (voire Figure III.3) [69].

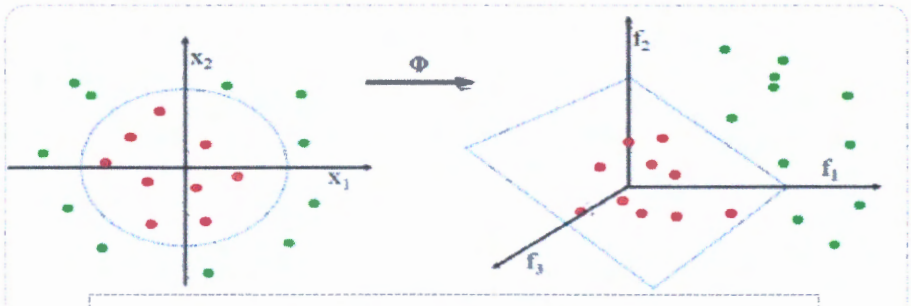


Figure III. 3: points non linéairement séparable et leur projection vers un autre espace d'une dimension supérieur.

La projection sur un espace de dimension supérieur permet d'effectuer des opérations linéaires équivalentes à des opérations non linéaires sur l'espace des entrées,

Cette projection est effectuée par la fonction de projection ϕ définis comme :

$\Phi : E \rightarrow F \quad , \quad x \rightarrow \Phi(x)$

D'où la fonction objective du problème d'optimisation sera reformulée comme :

$$\Phi(\alpha) = \text{Max} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j < \Phi(x_i), \Phi(x_j) >$$

Où $\Phi(x_i), \Phi(x_j)$ est le produit scalaire des deux images des vecteurs xi et xj dans le nouvel espace et dont le résultat est un scalaire. [69]

Le problème et sa solution ne dépendent que du produit scalaire $\Phi(x) \times \Phi(x_i)$. Plutôt que de choisir la transformation non-linéaire $\Phi : X \rightarrow F$, on choisit une fonction $K : XxX \rightarrow R$ (Nombres réels) appelée *fonction noyau*. [74].

Le produit scalaire imposé par la projection est plus complexe et très coûteux en calcul due a la grande dimension de Φ, d'autres fonctions dites fonction Kernel peuvent réaliser ce calcul sans faire de projection explicite vers d'autres espaces, l'utilisation de fonction Kernel pour éviter la projection est connu sous le nom de « Kernel Trick ». [73]

Une fonction Kernel (noyau) est définie comme :

$x^*x \rightarrow R \quad , \quad K(x_i, x_j) \rightarrow \Phi(x_i), \Phi(x_j)$ [19]

Pour remplacer la fonction de projection une fonction Kernel doit vérifier le théorème de Mercer qui énonce qu'une fonction Kernel représente le produit scalaire si elle est définie positive.

L'équation de l'hyperplan séparateur dans l'espace de redescription devient :

$$f(x) = \sum_{i=1}^{s} \alpha_i y_i K(x, x_i) + b \qquad (6)$$

S : représente les indices des vecteurs à support. [73]

4-2- Condition de Mercer

La matrice contenant les similarités entre tous les exemples de l'entraînement La fonction K (xi; xj) peut être vue comme une matrice G [n; n] dite de Gram représente les distances entre tous les exemples : [75] [76]

$$\begin{pmatrix} K(x_1; x_1) \dots K(x_1; x_n) \\ \vdots \quad \ddots \quad \vdots \\ K(x_n; x_n) \dots K(x_n; x_n) \end{pmatrix}$$

Théorème : (condition de Mercer) la fonction est un noyau si et $K(x, y) = XxX \rightarrow R$ seulement si :

$G = (K(x_i, y_j)), i, j = 1..n.$ Est définie positive.

Notons qu'une fonction $K : XxX \rightarrow R$ générant une matrice définie positive possède les trois propriétés fondamentales du produit scalaire : $\forall x_i, x_j \in X$

1. Positivité : $K(x_i, y_j) \geq 0$)

2. Symétrie : $K(x_i, y_j) \geq K(x_j, y_i)$

3. Inégalité de Cauchy-Shwartz : $|K(x_i, x_j| \leq \|x_i\| \cdot \|x_j\|$

La condition de Mercer nous indique si une fonction est un noyau mais ne fournit aucune information sur la fonction Φ (et donc sur l'espace des caractéristiques) introduit par ce noyau. [76], [77]

4-3- Exemple de noyaux

- **Noyau linéaire :** Si les données sont linéairement séparables, on n'a pas besoin de changer d'espace, et le produit scalaire suffit pour définir la fonction de décision :

$$K(x_i, x_j) = x_i^T x_j$$

- **Noyau polynomial :** Le noyau polynomial élève le produit scalaire à une puissance naturelle d : $K(x_i, x_j) = (x_i^T x_j)^d$

Si d = 1 le noyau devient linéaire. Le noyau polynomial dit non homogène $K(x_i, x_j) = (x_i^T x_j + C)^d$ est aussi utilisé.

- **Noyau RBF :** Les noyaux RBF (Radial Basis functions) sont des noyaux qui peuvent être écrits sous la forme : $K(x_i, x_j) = f(d(x_i, x_j))$ où d est une métrique sur X et f est une fonction dans R. Un exemple des noyaux RBF est le noyau Gaussien :

$$K(x_i, x_j) = e^{\left(-\frac{\|x_i - x_j\|^2}{2\delta^2}\right)}$$

Où δ est un réel positif qui représente la largeur de bande du noyau.

Noyau Sigmoïd : avec paramètres K et θ *avec*: $k(x, y) = \tanh(kx.\ y + \theta)$ [78] [79]

5- L'Algorithme SMO (Sequential Minimal Optimization)

L'algorithme Optimisation séquentielle minimale proposé par [80] est un algorithme qui permet de résoudre rapidement le problème quadratique (4) du SVM sans passer par toutes les étapes de résolution numérique d'un QP. L'idée principale des algorithmes de décomposition est de travailler avec un sous ensemble réduit de données du problème, garder les solutions et continuer avec le reste des données où les solutions antérieures doivent être encore testées. Le SMO prend cette idée à l'extrême : il optimise seulement deux vecteurs par itération (α_i, α_j). Cette optimisation admet une solution analytique.

- **Optimisation de α_1 et α_2 : (α_i et α_j)**

Rappelez la fonction objective que nous voulons optimiser :

Min $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \ y_i \ y_j \ K(x_i \ x_j)$

On peut l'ecrire comme :

$F(x) = \alpha_1 + \alpha_2 - \frac{1}{2} K(x_1, x_1) \ \alpha_1^2 - \frac{1}{2} K(x_2, x_2) \ \alpha_2^2 - 2y_1 y_2 K_{12} \alpha_1 \alpha_2 + y_1 \alpha_1 v_1 +$ $y_2 \alpha_2 v_2 + C_{St}$ où $V_i = \sum_{i=3}^{n} y_j \alpha_j^* \ K(x_i, x_j)$

En respectant les contraintes : $0 \leq \alpha_1, \alpha_2 \leq C$ et $\sum_{i=1}^{n} \alpha_i y_i = 0$

On dérive F(x) par rapport à α_2 et on obtient des expressions de cette variable en fonction de l'erreur de classification (6)

$\alpha_2^{new} = \alpha_2^{old} - \frac{y_2(E_1 - E_2)}{K}$ pour généraliser $\alpha_j^{new} = \alpha_j^{old} - \frac{y_2(E_i - E_j)}{K}$

Où $K = K(x_1, x_1) + K(x_2, x_2) + K(x_1, x_2)$

$E_i = f(x) - y_i = \left(\sum_{i=1}^{n} y_i \alpha_i \ K(x_i, x_j) + b \right) - y_i = u_i - y_i$ (6)

On peut considérer E_i comme l'erreur entre la sortie SVM sur l'ieme exemple et la véritable étiquette y (i). Cela peut être calculé en utilisant l'équation (2).

En calculant le paramètre K, on peut utiliser une fonction K (x_i, x_j) du noyau à la place du produit intérieur si désiré.

- Nous voulons trouver les limites L et H telles que $L \leq \alpha_2 \leq H$ doit contenir pour que α_2 satisfasse la contrainte que $0 \leq \alpha_2 \leq C$. On peut montrer que ceux-ci sont donnés par les éléments suivants:

If $y_1^{(i)} = y_2^{(j)}$, $L = \max(0, \alpha_{2(j)} - \alpha_{1(i)} - C)$, $H = \min(C, C + \alpha_{2(j)} - \alpha_{1(i)})$

If $y_1^{(i)} \neq y_2^{(j)}$, $L = \max(0, \alpha_{2(j)} - \alpha_{1(i)})$, $H = \min(C, C + \alpha_{2(j)} - \alpha_{1(i)})$

$$\alpha_{2(j)}^{new\_clipped} = \begin{cases} H & \text{if } \alpha_{2(j)}^{new} \geq H ; \\ \alpha_{2(j)}^{new}(i) & \text{if } L < \alpha_{2(j)}^{new} < H; \\ L & \text{if } \alpha_{2(j)}^{new} \leq L. \end{cases}$$

$$\alpha_{1(i)}^{new} = \alpha_{1(i)}^{old} + y_{1(i)} y_{2(j)} (\alpha_{2(j)}^{old} - \alpha_{2(j)}^{new\_clipped})$$

Le seul inconvénient de cette méthode est son critère d'arrêt basé sur les conditions de KKT, qui n'est pas toujours facile à contrôler. À l'heure actuelle, cette méthode est la plus courante pour appliquer les SVM à des problèmes de grande taille. [80]

6- Les conditions de Karush-Kuhn-Tucker (KKT)

Les conditions de Karush-Kuhn-Tucker (KKT) sont nécessaires et des conditions suffisantes pour un point optimal d'un problème de QP défini positif.

Les conditions KKT pour le problème QP (11) sont particulièrement simples. Le problème QP est résolu quand, pour tout i: [03]

$$\alpha_i = 0 \Rightarrow u_i \geq 1$$

$$\alpha_i = C \Rightarrow u_i \leq 1$$

$$0 < \alpha_i < C \Rightarrow u_i = 1.$$

Et $u_i = y_i (w^T x_i + b)$

7- Calculer le seuil b

Après avoir optimisé α_i et α_j, nous sélectionnons le seuil b de telle sorte que les conditions KKT soient satisfaites pour les $i^{ème}$ et $j^{ème}$ exemples. [80]

Si, après l'optimisation, α_i n'est pas à la limite (c'est-à-dire $0 < \alpha_i < C$), Alors le seuil b_1 suivant est valide, Car cela oblige le SVM à sortir y_1 (i) lorsque l'entrée est x_1 (i)

$$b_1 = E_1 + y_1(\alpha_1^{new} - \alpha_1) K(x_1, x_1) + y_2(\alpha_{2(j)}^{new\_clipped} - \alpha_2) K(x_1, x_2) + b$$

Le seuil suivant b2 est valide lorsque le nouveau α_2 n'est pas limité, Car il force la sortie du SVM à y_2 lorsque l'entrée est x_2:

$$b_2 = E_2 + y_1(\alpha_1^{new} - \alpha_1) K(x_1, x_2) + y_2(\alpha_{2(j)}^{new\_clipped} - \alpha_2) K(x_2, x_2) + b$$

Si les deux $0 < \alpha_i < C$ et $0 < \alpha_j < C$, alors ces deux seuils sont valides et ils seront égaux.

Si les deux nouveaux sont aux bornes (c'est-à-dire $\alpha_i = 0$ ou $\alpha_i = C$ et $\alpha_j = 0$ ou $\alpha_j = C$) alors tous les seuils entre b_1 et b_2 satisfont les conditions KKT,

SMO choisit le seuil à mi-chemin entre b_1 et b_2. Ceci donne l'équation complète pour b,

$$b := \begin{cases} b_1 & \text{if } 0 < \alpha_i < C \\ b_2 & \text{if } 0 < \alpha_j < C \\ (b_1 + b_2)/2 & \text{sinon} \end{cases}$$

8- Architecture générale d'un SVM

Une machine à vecteur support, recherche à l'aide d'une méthode d'optimisation, dans un ensemble d'exemples d'entrainement, des exemples, appelés vecteurs support, qui caractérisent la fonction de séparation. La machine calcule également des multiplicateurs

associés à ces vecteurs. Les vecteurs supports et leurs multiplicateurs sont utilisés pour calculer la fonction de décision pour un nouvel exemple.

Le schéma de la figure III. 4 résume l'architecture générale d'une SVM dans le cas de la reconnaissance des chiffres manuscrits.

La fonction noyau K est utilisée pour calculer la distance entre le vecteur à tester x et chaque vecteur support dans l'espace de caractéristique. Les résultats sont ensuite linéairement combinés en utilisant les multiplicateurs de Lagrange αi et ajoutés au biais b (w_0). Le résultat final f permet de décider à propos du nouveau vecteur : si f(x) est positive, il s'agit du chiffre "1", sinon, il s'agit d'un autre chiffre. [81]

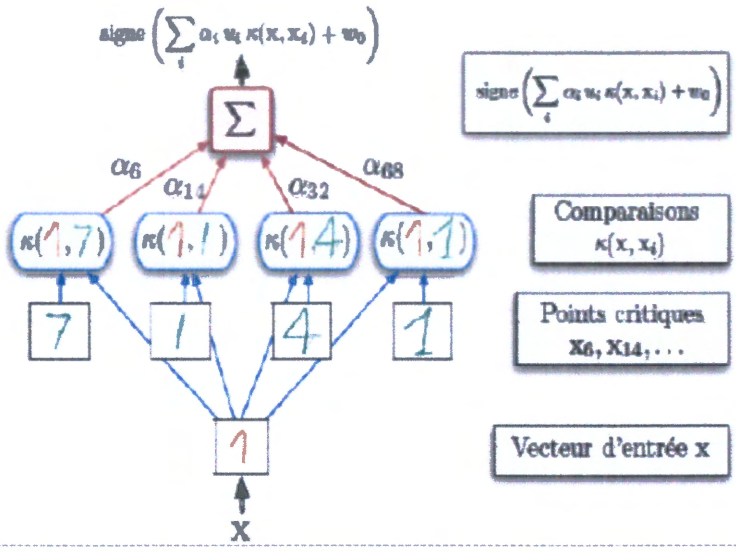


Figure III. 4: Principe de fonctionnement d'un SVM (cas des chiffres manuscrits)

9- Stratégie multi-classes

Bien que la méthode SVM soit adaptée aux problèmes de deux classes, elle peut être facilement transformée en un outil très efficace pour la classification N-classes.

Il existe deux méthodes pour résoudre un problème multi-classe avec des classifieurs SVM [70] [14]:

9-1- Approche un-contre-tous (1 VS R)

L'idée de cette stratégie est de construire autant de classifieurs que de classes.

Ainsi, durant l'apprentissage, tous les exemples appartenant à la classe considérée sont étiquetés positivement (+1) et tous les exemples n'appartenant pas à la classe sont étiquetés négativement (-1).

Un hyperplan f_k est défini pour chaque classe k par la fonction de décision suivante :

$f_k(x) = (< w_k, x > + b_k)$

$$\begin{cases} = \quad +1, si\ f_k\ x > 0 \\ 0, sinon \end{cases}$$

$k* = \text{Arg}1 \leq k \leq K\ ((x)$

Si une seule valeur (x) est égale à 1 et toutes les autres sont égales à 0.

On conclut que x appartient à la classe k. [82]

Or, il est possible que plusieurs sorties soient positives pour un exemple de test donné. Ceci est particulièrement le cas des données ambiguës (la région des points non classifiable) situées près des frontières de séparation des classes (figure III. 5). On utilise dans ce cas un vote majoritaire [83] pour attribuer l'exemple x à la classe k selon la formule:

$k* = \text{Arg}1 \leq k \leq K\ (w_k, x + b_k)*$

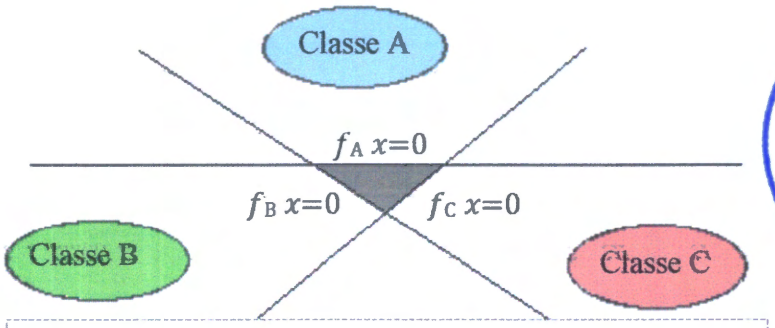


Figure III. 5: Nuage de points à 3 classes : l'approche un contre Tous

Souvent, la méthode 1vsR est critiquée à cause de son asymétrie [84], puisque chaque hyperplan est entrainé sur un nombre d'exemples négatifs beaucoup plus important que le nombre d'exemples positifs. Par exemple dans le cas des chiffres 0-9, le classifieur du chiffre '0' est entrainé sur des exemples positifs représentant '0' et des exemples négatifs représentant tous les autres chiffres (1-9). La méthode une contre une suivante est une méthode symétrique qui corrige ce problème.

9-2- Approche un-contre-un (1 VS 1)

L'approche un-contre-un est un cas spécial des méthodes de décomposition proposées par Dietterich et al [85].

Ce deuxième algorithme utilise un modèle SVM pour chaque paire de classes. Alors le nombre total de modèles utilisés pour un problème de m classes est $m\ (m-1)/2$ modèles. Chaque modèle réalise une classification binaire et sépare les données de deux classes.

Il considère les données de la première classe comme des éléments positifs et les données de la deuxième classe comme des éléments négatifs.

La pluparts des systèmes de classification dans cette catégorie utilisent le vote majoritaire comme un critère de décision, f_{ij} (x) est la fonction de décision qui sépare la classe i de la classe j par : f_{ij} (x) = - f_{ij} (x), Au cours de l'étape de classement (figure III.6), la stratégie de vote majoritaire est calculée comme suit: [86] [58] [03]

Argmax i =1...N $\sum_{j\neq i,i=1}^{N} sign\ (f_{ij}\ (x))$

Cependant, lorsque deux classes ont le même score, x ne sera pas classé. Ce concept est présenté à la figure III.6.

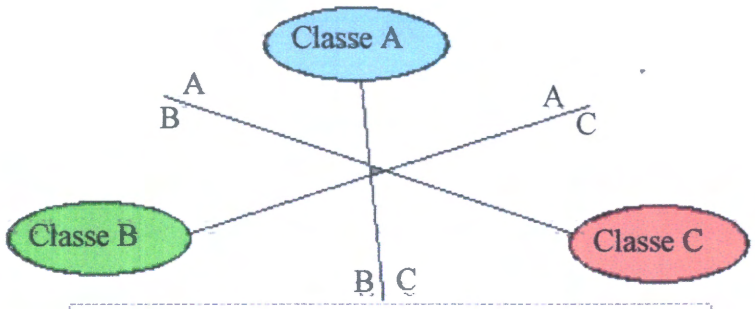


Figure III. 6: Une observation appartient à la zone d'ambigüité

Plusieurs solutions ont été proposées pour résoudre ce problème. On cite par exemple celle de Hsu et Lin [87] qui utilise une décision aléatoire.

10- Conclusion

Nous avons présenté dans ce chapitre, une méthode de classification à base des SVM. On a été consacré sur une présentation de la formulation mathématique des SVM et l'algorithme utilisé pour son optimisation. Celle-ci est fondée sur l'utilisation de l'algorithme SMO, Son principe repose sur l'optimisation de deux Multiplicateurs de lagrangien à chaque étape.

CHAPITRE IV :

Implémentation Du Système Proposé

1- Introduction

2- Acquisition

3- Prétraitement

4- Segmentation

5- Extraction des caractéristiques

6- Support Vecteur Machine

1- Introduction

L'objectif que nous nous sommes assignés s'articule autour du développement d'un système pour la reconnaissance des chiffres manuscrits. La figure IV.1 montre la méthodologie adoptée pour la reconnaissance. Chacune d'elles regroupe plusieurs opérations.

Dans ce qui suit, nous allons expliquer les différentes techniques utilisées pour la réalisation de ce système.

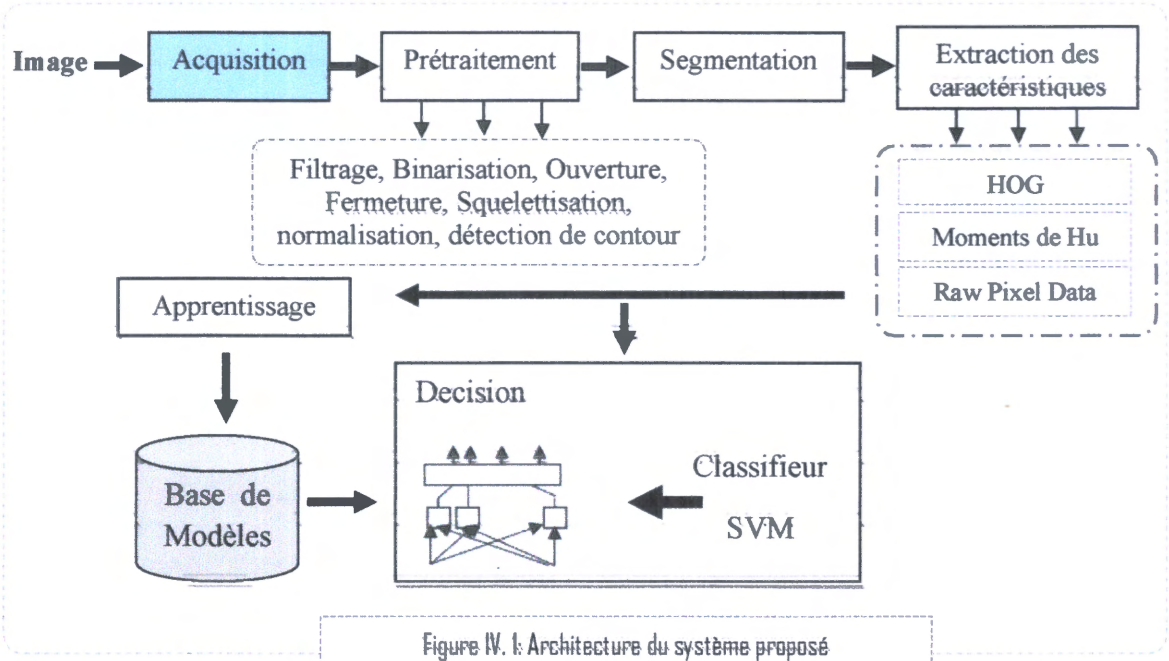


Figure IV. 1: Architecture du système proposé

2- Acquisition

Les chiffres ont été écrits sur des pages blanches avec un stylo bleu (ou rouge ou gris,….), de 0 à 9.

L'acquisition a été effectuée avec un scanner EPSON STYLUS SX130 à une résolution de 300 dpi (et parfois de 600 dpi), qui a permis d'avoir des images de bonne qualité.

On a 3 types d'image scanné : Couleur (RGB) - Niveaux de gris (0-256) - Noir & Blanc(0,1) et l'image dessiner (JPEG)

Des exemples sont présentés dans les figures IV.2, IV.3, IV.4, IV5 :

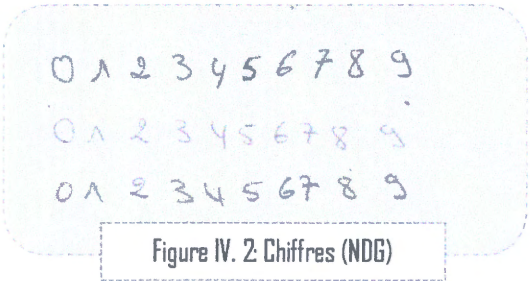


Figure IV. 2: Chiffres (NDG)

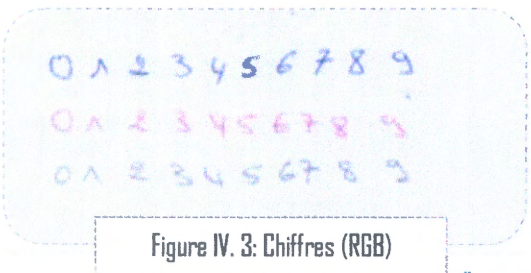


Figure IV. 3: Chiffres (RGB)

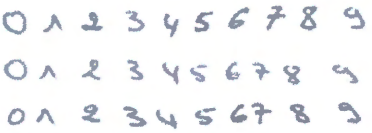


Figure IV. 4: Chiffres (N, B)

Figure IV. 5: Chiffres dessiner (JPEG)

3- Prétraitement

3-1- Filtrage

Une phase de filtrage des images est réalisée afin de réduire le bruit. Pour nos formulaires, on a appliqué un filtre médian qui procède tout d'abord par la transformation de l'image en niveaux de gris car La technique de ce filtre est largement utilisée en traitement d'images numériques.

En traitement d'image, les tailles des fenêtres utilisées pour le filtrage médian sont généralement impaires : 3 × 3 ; 5 × 5 ; 7 × 7. On a choisi d'utiliser la taille 3 × 3.

3-1-1- Filtre Médian

Le filtre médian est un filtre non linéaire, l'idée principale de ce dernier est de remplacer chaque pixel par la valeur médiane de son voisinage.

- Eliminer les événements de taille inférieure à la moitié de la taille de la fenêtre d'observation (élimination de bruits type impulsionnel) [88].

- Les événements les plus grands conservent leur dynamique, les frontières restent aussi plus précises.

Exemple :

1. Considérons neuf pixels en niveaux de gris, dont une valeur est aberrante (ici 255):

2	4	12
2	255	3
7	9	3

2. Le filtre médian va d'abord trier ces valeurs par ordre croissant :

   2, 2, 3, 3, **4**, 7, 9, 12, 255

3. Et prendre la valeur médiane (la cinquième valeur),

   Ici la valeur 4. La sortie du filtre donnera :

2	4	12
2	4	3
7	9	3

Le principe de l'algorithme utilisé est le suivant :

Algorithme IV. 1: Filtre Médian

Soit *V* un voisinage d'un pixel *p* ;
Soit *n* le nombre de pixels dans *V* ;
Soit *T* un tableau ;
I[x] représente le niveau de gris de *x*.
Pour tout *x* dans *V*, mettre *I[x]* dans *T* ;
Ordonner *T* ;
T [n/2] est la nouvelle luminance de *p* ;

3-2- Binarisation

La binarisation est un cas particulier de seuillage qui vise en principe à classer les pixels de l'image traitée en deux classes (noire et blanche) : les pixels de premier plan associés au texte et les pixels de l'arrière-plan associés au fond. Les images que nous avons utilisé dans notre système peuvent être en différents modes (niveau de gris, couleurs…). A ce niveau, nous avons opté pour la méthode d'Otsu (figure IV.6) suivante [89].

3-2-1- La Méthode D'OTSU

Cette méthode consiste à déterminer un seuil unique globalement sur toute l'image. Les statistiques sont alors calculées pour chaque niveau d'intensité i, i.e. pour tous les seuils possibles. Dans le cadre de la binarisation par la méthode d'Otsu, la séparation se fait à partir des moments des deux premiers ordres à savoir la moyenne et l'écart type. De plus, dans l'optique de rendre le procédée indépendant du nombre de points dans l'image, on normalise l'histogramme [89]. Ainsi on a :

$$\text{Histogramme}_{\text{normalisé}} : \frac{n_i}{\sum n_i}$$

Ou n_i représente le nombre de pixels de niveau i dans l'image.

Comme énoncée précédemment, la séparation se fait à partir de la moyenne et de la variance.

Des lors on a :

$$\text{Moy} = \sum_{i=1}^{k} i * \text{Histogramme}_{\text{normalisé}}(i) \quad \textbf{ET} \quad \text{Var} = \sum_{i=1}^{k} \text{Histogramme}_{\text{normalisé}}(i)$$

Enfin on réalisé pour chaque valeur de k le calcul suivant :

$$S^2(k) = \text{Var}(k) * (1 - \text{Var}(k)) * (\text{Moy}(255) * \text{Var}(k) - \text{Moy}(k))^2$$

Pour K=1…255

Dés lors le niveau qui maximise la fonction critère est considère comme seuil pour la binarisation de l'image. Ainsi la valeur de seuil est obtenue lorsque pour un k donnée on a :

$$S^2(k) \equiv \text{Max}(S^2(k))$$

Algorithme IV. 2: Méthode D'OTSU

1- Calculer l'histogramme et les probabilités de chaque niveau d'intensité

3- Passer à travers tous les seuils possibles K=1 … 255

4- Calculer $S^2(k)$

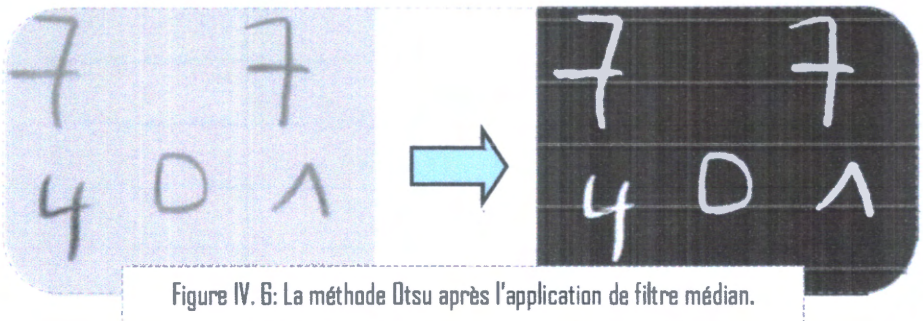


Figure IV. 6: La méthode Otsu après l'application de filtre médian.

3-3- Operateurs Morphologiques

Le but des opérations morphologiques est la correction, l'amélioration et la préparation de l'image à traiter; parce que chaque fois qu'une image est convertie d'une forme a une autre, sa qualité est plus ou moins affectée [90]. Ce type de traitement vise donc à éliminer les bruits, et à corriger les dégradations et les distorsions géométriques lors de l'acquisition de l'image afin d'améliorer et de restaurer l'image à traiter [91].

3-3-1- L'Ouverture Et Fermeture

L'ouverture (une érosion suivie d'une dilatation), permet d'ouvrir les petits trous et les espaces entre les objets qui se rapprochent suffisamment l'un de l'autre (figure IV.8) , Tandis que la fermeture (une dilatation suivie d'une érosion) permet de remplir les petits trous dans la forme(figure IV.7) . [92]

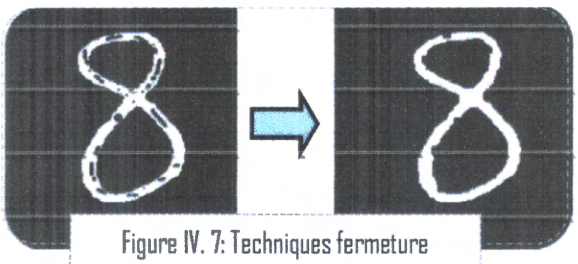


Figure IV. 7: Techniques fermeture

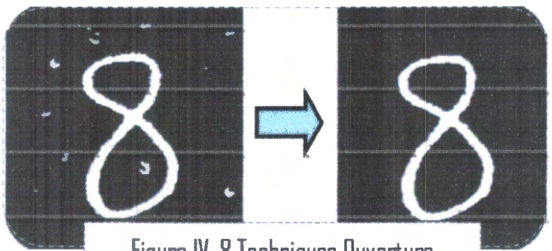


Figure IV. 8 Techniques Ouverture

Algorithme IV. 3: Dilatation Et Erosion

Algorithme : Parcours de l'image

Les algorithmes de morphologie mathématique parcourent l'image :

- En chaque pixel de départ (u (i,j)) on place l'élément structurant centré sur le pixel noir,
- Un test est réalise pour déterminer la couleur du pixel d'arrivée u' (i, j).

Algorithme : Dilatation

- **Algorithme :** Parcourir les pixels de l'image u,
  Pour chaque pixel u (i, j):
  1. Centrer l'élément structurant sur ce pixel,
  2. Considérerez les voisins du pixel dans l'élément structurant (gris + noir)
  3. Si l'un de ces pixels est en blanc, mettre u' (i, j) en blanc.

Algorithme : Erosion

- **Algorithme :** Parcourir les pixels de l'image u,
  Pour chaque pixel u (i, j):
  1. Centrer l'élément structurant sur ce pixel,
  2. Considérer les voisins du pixel (gris + noir de l'élément structurant)
  3. Si l'un de ces pixels est en noir, mettre u' (i, j) en noir.

3-4- Squelettisation

La squelettisation est l'une des techniques les plus utilisées dans la reconnaissance de l'écriture, notamment à des fins de l'extraction de caractéristiques morphologiques. Elle permet de simplifier l'image du caractère en une image plus facile à traiter en la réduisant à une forme avec une épaisseur de 1 pixel tout en conservant ses propriétés topologiques. Il existe de nombreuses méthodes de squelettisation.

Dans ce travail, nous avons appliqué l'algorithme de Zhang and Suen [93] (figure IV.9),

Réputé par son efficacité dans la préservation de la connectivité et de la topologie des caractères.

3-4-1- Algorithme de Zhang et Suen

L'algorithme de Zhang et Suen introduit deux critères pour décider si un pixel P doit être éliminé.

En premier lieu, il s'assure que le pixel considéré est un pixel noir et au moins l'un de ses voisins est blanc.

P1	P2	P3
P8	P	P4
P7	P6	P5

- La 1ère itération pour transformer le pixel en pixel blanc si les conditions suivantes dans (t) sont réalisées. (Voir Table IV.1)
- A la 2ème itération les conditions (1) et (2) ne changent pas plus les conditions suivantes dans (t) sont réalisées. (Voir Table IV.1)

Critère 1(première itération)	Critère 2(deuxième itération)
1. La connectivité est égale à 1 2. Il y a au moins 2 et au plus 6 voisins de valeur noire. 3. Au moins l'un des pixels P2, P4, P6 est blanc. 4. Au moins l'un des pixels P4, P6, P8 est blanc.	3. Au moins l'un des pixels P4, P2, P8 est blanc. 4. Au moins l'un des pixels P2, P6, P8 est blanc

Table IV. I: Table des conditions à chaque itération

Les pixels réalisant ces conditions doivent être supprimés. A la fin s'il n'y a aucun pixel à supprimer, alors l'algorithme s'arrête.

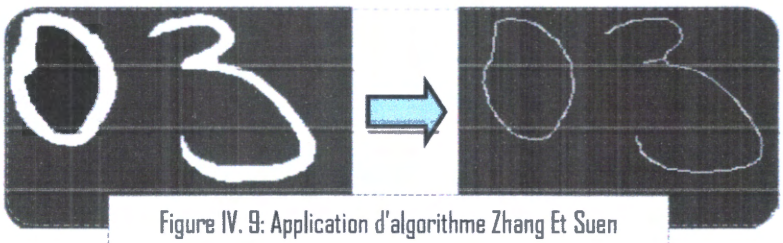


Figure IV. 9: Application d'algorithme Zhang Et Suen

3-5- La Normalisation

La méthode de normalisation est populairement utilisée dans la reconnaissance de caractères pour réduire tous les types de variations et pour obtenir des données standardisées.

Cependant, cela entraîne également une distorsion de forme excessive et élimine certaines informations utiles.

Nous avons utilisé 3 étapes pour normaliser les chiffres qui sont les suivantes:

3-5-1- Correction de l'inclinaison des lignes

En raison de la variation du style d'écriture, L'inclinaison peut nuire à l'efficacité de la reconnaissance et donc être détectée et corrigée par rapport à la ligne de base.

Diverses méthodes ont été utilisées pour «Skew normalization», Dans notre travaille nous nous choisissons La transformation Hough [94].

Elle est surtout utilisée dans la détection des lignes et d'angle d'inclinaison dans une image. Après la détection de faiblesse, Le caractère ou le mot est traduit à l'origine et tourné jusqu'à ce que la ligne de base soit horizontale (figure IV.11).

3-5-1-1- La Transformation de Hough

La transformée de Hough, est une méthode qui permet de détecter les formes paramétriques dans des images, en particulier elle est utilisée pour la détection des lignes dans une image. Chaque ligne peut être définie par l'équation suivante :

$d = x^*\cos\Theta + y^*\sin\Theta$ tel que d est la distance de la normale entre l'origine et la ligne, et Θ est l'angle entre la normale et l'axe des X [-90◦a 90◦] (figure IV.10).

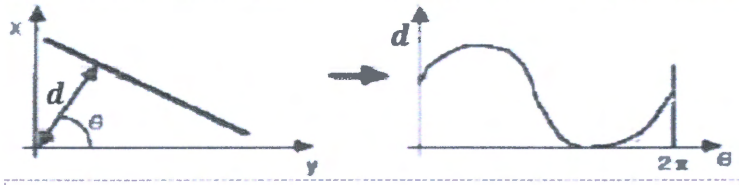


Figure IV. 10: Représentation d'une droite dans l'espace de Hough

Dans la pratique, la transformée de Hough est implémentée par un vecteur accumulateur qui est paramétré par les valeurs de ť et Θ.

Pour chaque pixel noir, la valeur de ť est calculée en faisant varier Θ l'intervalle [0°,π], le contenu de la cellule d'indice (ť, Θ) est incrémenté. Après avoir traité toute l'image, le maximum (ťm, Θ m) est recherché, où Θ m correspond à l'angle d'inclinaison. [95]

Algorithme IV. 4: Transformation Hough

1- Initialise H[d, Θ]

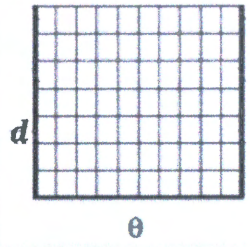
2- Pour chaque point I[x, y] dans l'image :

  2-1- Pour Θ = [Θ min a Θ max]

  2-2- H [d, Θ] +=1

3- Trouver la(s) valeur(s) de (d, Θ) pour H[d, Θ] est maximum

4- La ligne détecté sur l'image est données par : $d = x^*\cos\Theta + y^*\sin\Theta$



H : vecteur accumulateur

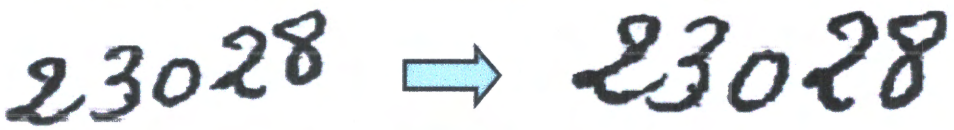


Figure IV. 11: Les méthodes de base pour la normalisation: (a) Skew normalization

3-5-2- Correction De L'inclinaison Des Lettres

L'inclinaison des caractères « Slant Normalization» généralement retrouvée dans le script cursif s'appelle slant. Alors, il est défini comme l'angle entre le plus long coup en un mot et la verticale,

La direction se réfère au mot inclinaison. La Correction de l'inclinaison des lettres est utilisée pour normaliser tous les caractères à une forme standard sans inclinaison.

De nombreuses méthodes ont été proposées pour détecter et corriger l'inclinaison des mots cursifs. On a utilisé le moment de second ordre car les moments de l'image nous donnent des informations importantes sur la structure et la densité de l'objet (figure IV.12).

3-5-2-1- Moment géométrique de deuxième ordre

Considérons une image binaire f (x; y), De largeur W et hauteur H et valeurs de pixels 0 et 1.

Les moments géométriques d'un p+q ordre de f sont donnés par:

$$m_{pq} = \sum_x \sum_y x^p \, y^q \; f(x, y)$$

Les moments centraux sont donnés par:

$$\mu_{pq} = \sum_x \sum_y (x - x_c)^p \, (y - y_c)^q \, f(x, y) \qquad \text{Ou} \qquad x_c = m_{10}/m_{00} \;\; \text{Et} \;\; y_c = m_{01}/m_{00},$$

Sont les coordonnées du centre de gravité d'un objet contenu dans cette image.

Les moments de second ordre μ_{20} et μ_{02} reflètent la quantité de pixels qui s'écartent du centre de gravité.

Nous les interprétons comme la taille de l'objet en direction x et y indépendamment.

Figure IV. 12: Les méthodes de base pour la normalisation: (b) Slant normalization

3-5-3- Normalisation de taille

Ou bien « Size Normalization» Il est utilisé pour ajuster la taille, Position et forme (dimension) de l'image du personnage (figure IV.13).

Cette étape est nécessaire pour réduire la variation de forme entre les images de la classe pour faciliter la génération de fonctionnalités et améliorer leur classification [61]

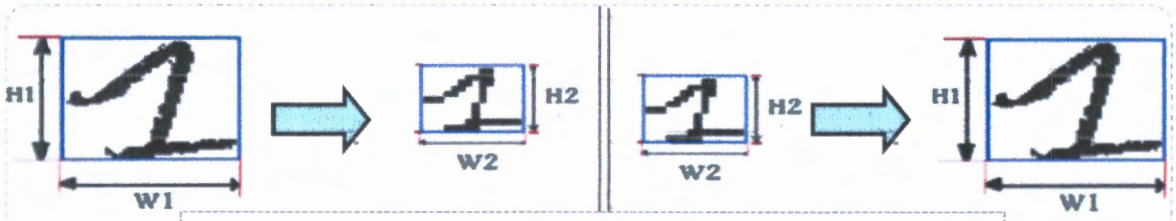


Figure IV. 13: Les méthodes de base pour la normalisation: (b) size normalization

3-6- Détection De Contour

Le contour est défini comme l'ensemble des pixels des mots de l'image ayant au moins un en commun avec le fond (en 4 ou 8 connexités) [96]. On appliquer l'algorithme d'approximation du contour (figure IV.16).

3-6-1- Méthode D'Approximation Du Contour

Soit $\alpha(k)$, l'angle entre le segment (P_0P_k) et l'axe horizontale positif comme montré à la figure IV.14 : $\alpha(k) = \text{arctage}\ \frac{(y_k - y_0)}{(x_k - x_0)}$

Soit $\beta(k)$, l'angle enter la tangente au disque centré en P_k et passant par P_0 et le segment (P_0P_k) figure IV.14: $\beta(k) = \arcsin \frac{\varepsilon}{\sqrt{(x_k - x_0)^2 + (y_k - y_0)^2}}$

Soit $T_c(k)$ respectivement $\beta_c(k)$ l'angle gui forme la tangente supérieure respectivement inférieure au disque centré en P_k avec l'axe horizontal positif (figure IV.14)

$$T_c(k) \equiv \alpha(k) + \beta(k) \quad , \quad \beta_c(k) \equiv \alpha(k) - \beta(k) \qquad [97]$$

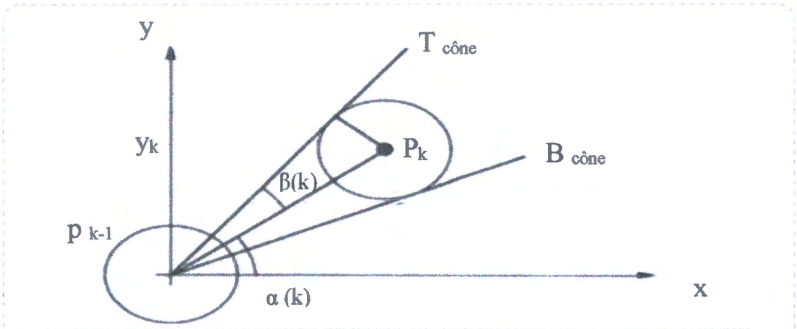


Figure IV. 14: Représentation graphique Des angles α(k), β(k), Tc(k), βc(k)

Soit $T_{min}(k_0)$, le minimum de $T_c(k)$ pour $k \leq k_0$ et $\beta_{max}(k_0)$, le maximum de $\beta_c(k)$ pour $k \leq k_0$.

Algorithme IV. 5: Approximation Du Contour

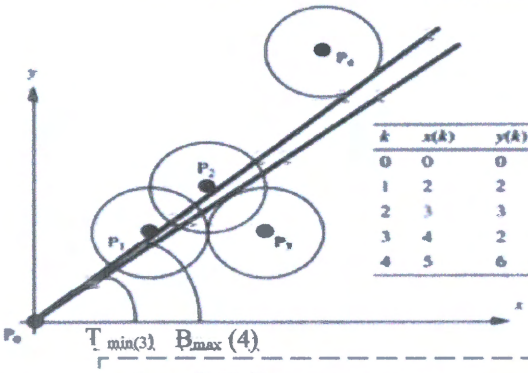
Règle 1 : en parcourant le contour depuis un point de départ arbitraire P0, on crier un segment (P_0P_{k-1}) au point k-1 si :

$\alpha(k) < B_{max}(k)$ ou $\alpha(k) > T_{min}(k)$

Règle 2 : au cas où les segments $(P_0, T_{min}(i))$ et $(P_0, B_{max}(i))$ se décroisent, c'est-à-dire si:

$T_{min}(i) < B_{max}(i)$

Alors le cône dégénère et l'algorithme d'approximation polygonale se termine avant que les n points du contour aient été visités. Le dernier point du polygone étant alors donné par (min (i, j)-1) comme l'illustre la figure IV.15.



k	x(k)	y(k)	α(k)	β(k)	$T_c(k)$	$B_c(k)$	$T_{min}(k)$	$B_{max}(k)$
0	0	0						
1	2	2	45	20.7	65.7	24.3	65.7	24.3
2	3	3	45	13.6	58.6	31.4	58.6	31.4
3	4	2	26.6	12.9	39.5	13.7	39.5	31.4
4	5	6	50.2	2.4	52.6	42.8	39.5	42.8

$T_{min(3)}$ B_{max} (4)

Figure IV. 15: Phénomène de dégénérescence Tmin (3) < Bmax (4), l'algorithme d'approximation se termine alors et le polygone finale est décrit par les points (P0, P2, P4)

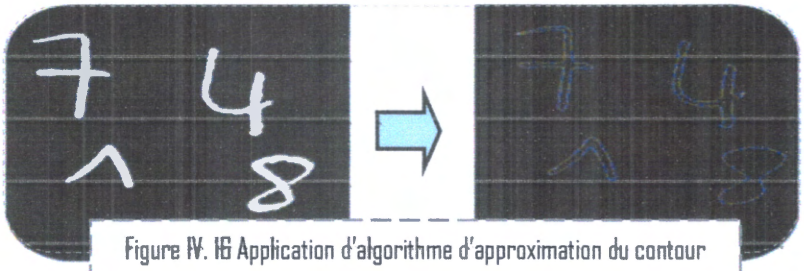


Figure IV. 16 Application d'algorithme d'approximation du contour

4- Segmentation

Dans un premier temps les contours des différentes composantes présentées dans l'image binaire sont extraits. Après nous faisons la segmentation avec une déduction de la boite englobante orientée de chacune des composantes. Chaque rectangle englobant contient un chiffre i.e. une de leur dimensions (largeur ou longueur), Ainsi, chaque composante soit gardée pour le reste du traitement.

4-1- Rectangle englobant

Cette méthode et appliquer après la détection de contour, puis l'extrême des Points signifie les points les plus bas , le haut, le plus à gauche et le plus à droite de l'objet (chiffre) (X_{min}, X_{max}, Y_{min}, Y_{max}), Enfin la déduction de la rectangle englobante d'après les 4 pixels (Y_{min}, X_{min}) (Y_{max}, X_{max}) (Y_{max}, X_{max}) (Y_{min}, X_{max}) comme l'illustre la figure IV.17.

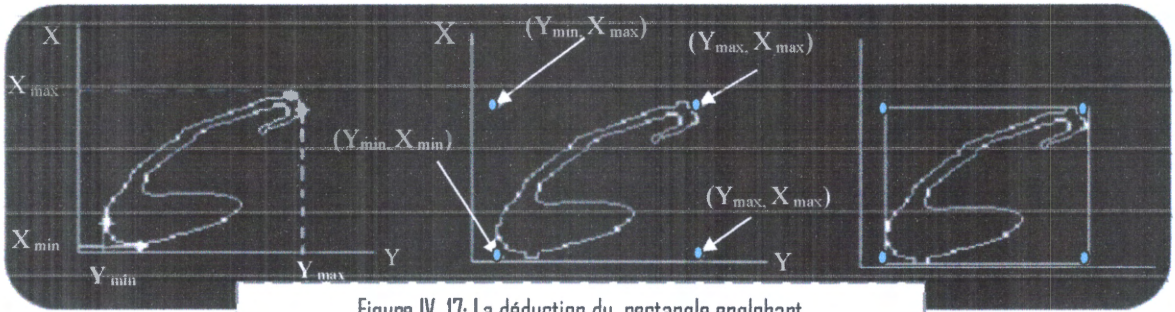


Figure IV. 17: La déduction du rectangle englobant

A l'issue de cette première étape, un grand nombre de composantes détecté (figure IV.18). Ont pu être retirées, considérées comme ne pouvant être une partie de notre travaille alors on applique la condition suivante : 13 < max (largeur ; longueur) < 300 alors (figure IV.19).

Algorithme IV. 6: Rectangle englobant

Règle 1: Trouver les points (X_{min}, X_{max}, Y_{min}, Y_{max})

Règle 2: D'après les points (Y_{min}, X_{min}) (Y_{max}, X_{max}) (Y_{max}, X_{max}) (Y_{min}, X_{max}) former la rectangle englobante et applique la condition : 13 < max (largeur ; longueur) < 300

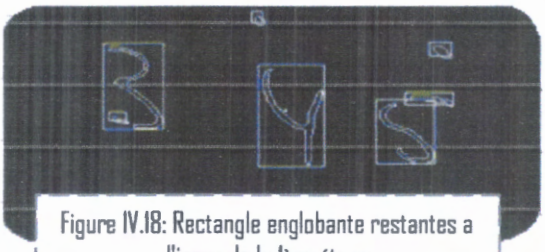


Figure IV.18: Rectangle englobante restantes a l'issue de la 1ère étape

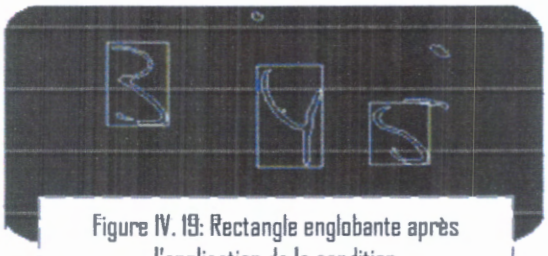


Figure IV. 19: Rectangle englobante après l'application de la condition

5- Extraction des Caractéristiques

L'extraction des caractéristiques est l'une des étapes les plus importantes de tout système de reconnaissance. Elle vise à extraire de l'image d'une classe donnée l'information pertinente permettant de la distinguer plus facilement des autres classes [77].

Dans notre application on a choisi d'exploiter trois méthodes : HOG (Histogramme de gradient orienté), Données de pixels brutes (Raw pixel data), les moments de Hu.

Chaque méthode a une taille de leur vecteur. Raw (784), HOG (16), Hu (7).

On applique les trois méthodes sur les chiffres de la base pour faire l'apprentissage.

5-1- Histogramme de Gradient Orienté (HOG)

Le descripteur HOG se base sur le calcul des orientations des gradients d'une image.

Le gradient d'un pixel d'une image f peut être calculé de différente façon de sorte à approcher la forme exacte $\nabla f = \frac{\delta f}{\delta x} x + \frac{\delta f}{\delta y} y$. Une méthode consiste à convoluer l'image par le filtre de Sobel (G_x, G_y) (chapitre I), Ainsi l'image $G = \sqrt{G_X^2 + G_Y^2}$ (La magnitude) contient les gradients de chaque pixel. L'image fournit par θ= atan $\left(\frac{G_y}{G_x}\right)$ (l'ongle) fournit les orientations des gradients en chaque pixel. [98]

Ayant maintenant l'orientation et l'intensité du gradient en chaque pixel, on découpe une zone autour du point clés à décrire en carrés de petite taille (de 4x4 à 12x12 pixels) et on crée un histogramme des orientations en pondérant chaque vote par l'intensité du gradient à ce pixel.

L'histogramme est ensuite normalisé pour éviter les disparités dues aux changements d'illumination. [99]

Algorithme IV. 7: De histogramme de gradient orienté

Règle 1 : Calculer gradient (G_x, G_y) 'Filtre Sobel'

Règle 2 : Calculer gradient magnitude and direction G et θ (en degrés)

Règle 3 : Calculer L'Histogramme des Gradients

5-1- Données de pixels brutes

Connu sous le nom Raw Pixel Data, Un classifieur SVM a besoin de fonctionnalités qu'il utilise pour reconnaître différents modèles. Les fonctionnalités étant des nombres. Dans ce cas, nous utilisons :

Les valeurs de couleur des pixels individuels (figure IV.20) :

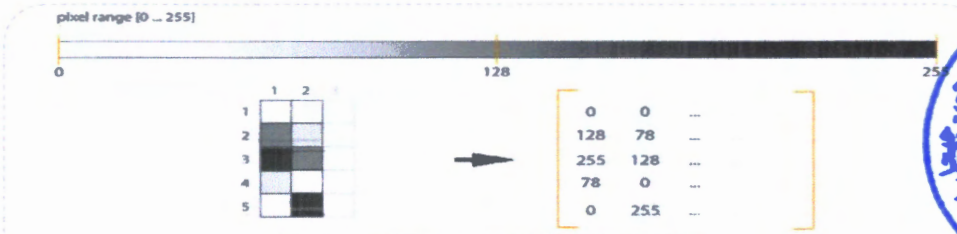


Figure IV. 20: Mappage de pixel

Les images dans l'ensemble de données MNIST (voire chapitre V) ont une largeur de 28 pixels et une hauteur de 28 pixels. Cela entraînera 784 fonctionnalités (28x28) que nous fournirons au SVM.

Pour former une SVM d'images, nous devons créer une matrice de formation. Dans cette matrice, chaque ligne correspond à une image et chaque colonne dans cette rangée correspond à une fonction d'image.

- Convertir notre image 2 dimensionnelle en 1 dimension (figure IV.21) :

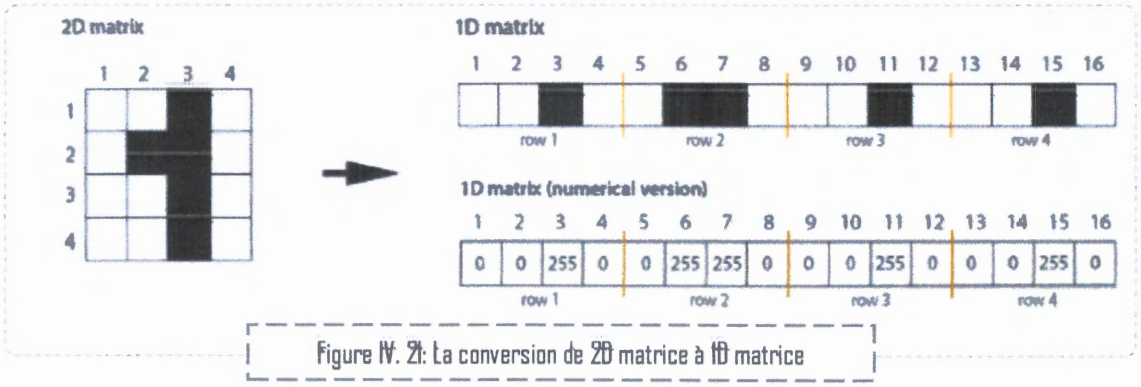


Figure IV. 21: La conversion de 2D matrice à 1D matrice

- Normaliser les pixels :

Les valeurs de pixels sont une échelle de gris entre 0 et 255. Parce que l'échelle est bien connue et bien conduite, Nous pouvons très rapidement normaliser les valeurs de pixels dans le rang -1 et 1 en divisant chaque valeur par un maximum de 255et multipliant en 0.2 puis moins 1 comme suite :

X =x/255 *0.2-1 alors

Algorithme IV. 8: Donneés des pixels brutes

Règle 1 : Fait le Mappage des pixels de la matrice 2D

Règle 2 : Convertir a une matrice 1D

Règle 2 : Normaliser les pixels

5-3- Moments de HU :

Hu a introduit la famille d'invariants portants son nom en utilisant les moments géométriques. Les moments géométriques représentent une double suite de coefficients. Le principe d'utilisation des moments est de sélectionner un sous-ensemble de M_{pq} qui contient suffisamment d'information pour caractériser l'image de façon unique, pour une application donnée [100].

Un moment géométrique général en ordre bidimensionnel (p + q) d'une image M × N est défini (dans le domaine discret) comme suit: [101]

$$m_{pq} = \sum_{x=0}^{x=M-1} \sum_{y=0}^{y=N-1} (x)^p (y)^q f(x,y)$$

Dans une image binaire, une zone d'un objet, ou le contour dans ce cas, est détenue par m00. Le centre de contour peut ensuite être calculé à partir de:

$$\overline{x} = \frac{m_{10}}{m_{00}} \quad , \quad \overline{y} = \frac{m_{01}}{m_{00}}$$

Où x et y sont des points sur les axes x et y, respectivement. En utilisant ce centroïde, Nous pouvons transformer le moment mentionné précédemment en un moment invariant de traduction en le redéfinissant en un moment central, défini comme :

$$\mu_{pq} = \sum_{x=0}^{x=M-1} \sum_{y=0}^{y=N-1} (x - \overline{x})^p (y - \overline{y})^q f(x, y)$$

Un moment central normalisé est invariant à l'échelle.

Par conséquent, le moment central peut être transformé en un moyen normalisé par :

$$\gamma = [(p + q)/2] + 1$$

$$\eta_{pq} = \mu_{pq} / \mu_{00}^{\gamma}$$

Enfin, pour rendre l'orientation du moment (rotation) invariable, Nous utilisons simplement η_{pq} sur les sept fonctionnalités de Hu ci-dessus au lieu d'un moment géométrique standard.

$M_1 = \eta_{20} + \eta_{02}$

$M_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$

$M_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$

$M_4 = (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2$

$M_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} - \eta_{12})[(\eta_{30} + \eta_{21})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

$M_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$

$M_7 = (3\eta_{21} - \eta_{03})(\eta_{30} - \eta_{12})[(\eta_{30} + \eta_{21})^2 - 3(\eta_{21} + \eta_{03})^2] + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2]$

Les six premiers moments sont invariants aux translations, aux rotations et aux changements d'échelle par contre le septième moment n'est pas invariant aux réflexions.

6- Support Vecteur Machine

Dans le SVM on applique l'algorithme de SMO avec toutes les étapes décrites en détaille dans le troisième chapitre.

6-1- Pseudo Code pour SMO

Dans cette section, nous présentons un pseudo-code pour l'algorithme de SMO simplifié. Soient les fonctions suivantes :

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

$$\text{If } y_1^{(i)} \neq y_2^{(j)},\ L = \max(0, \alpha_{2(j)} - \alpha_{1(i)}),\ H = \min(C, C + \alpha_{2(j)} - \alpha_{1(i)}) \quad (2)$$

$$\text{If } y_1^{(i)} = y_2^{(j)},\ L = \max(0, \alpha_{2(j)} - \alpha_{1(i)} - C),\ H = \min(C, C + \alpha_{2(j)} - \alpha_{1(i)}) \quad (3)$$

$$K = K(x_1, x_1) + K(x_2, x_2) + K(x_1, x_2) \quad (4)$$

$$\alpha_j^{new} = \alpha_j^{old} - \frac{y_2(E_i - E_j)}{K} \quad (5)$$

$$\alpha_{1(i)}^{new} = \alpha_{1(i)}^{old} + y_{1(i)} y_{2(j)} (\alpha_{2(j)}^{old} - \alpha_{2(j)}^{new\_clipped}) \quad (6)$$

$$b_1 = E_1 + y_1(\alpha_1^{new} - \alpha_1) K(x_1, x_1) + y_2(\alpha_{2(j)}^{new\_clipped} - \alpha_2) K(x_1, x_2) + b \quad (7)$$

$$b_2 = E_2 + y_1(\alpha_1^{new} - \alpha_1) K(x_1, x_2) + y_2(\alpha_{2(j)}^{new\_clipped} - \alpha_2) K(x_2, x_2) + b \quad (8)$$

$$b := \begin{cases} b_1 \text{ if } 0 < \alpha_i < C \\ b_2 \text{ if } 0 < \alpha_j < C \\ (b_1 + b_2)/2 \text{ sinon} \end{cases} \quad (9)$$

$$\alpha_{2(j)}^{new\_clipped} = \begin{cases} H & \text{if } \alpha_{2(j)}^{new} \geq H; \\ \alpha_{2(j)}^{new}(i) & \text{if } L < \alpha_{2(j)}^{new} < H; \\ L & \text{if } \alpha_{2(j)}^{new} \leq L. \end{cases} \quad (10)$$

Algorithme IV. 9: SMO Simplifié

Entrées:

C: Paramètre de régularisation

Tol: Tolérance numérique

Max passes: Valeur maximum

$(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$: données d'entraînement

Sortie:

$\alpha_i \in \Re^m$: Multiplicateurs de Lagrange

$b \in \Re$: Seuil de solution

- Initialiser $\alpha_i = 0, \forall i, b = 0$.
- Initialiser passes = 0.

- **Tant que** (passes < max passes)
  - Num_changer_alphas = 0.
  - **Pour** i = 1, . . . m,
    - Calculer $E_i = f(x^{(i)}) - y^{(i)}$ en utilisant (1).
    - **Si** $((y^{(i)} E_i < -Tol \;\&\&\; \alpha_i < C) \;\|\; (y(i) E_i > Tol \;\&\&\; \alpha_i > 0))$
      - Sélectionnez $j \neq i$ au hasard (random).
      - Calculer $E_j = f(x^{(j)}) - y^{(j)}$ en utilisant (2)
      - Sauvegarder Ancient (old) α's : $\alpha_i^{old} = \alpha_i$, $\alpha_j^{old} = \alpha_j$
      - Calculer L et H avec (2) ou (3).
      - **Si** (L = H)
        Continuer vers le prochain i.
      - Calculer K avec (4).
      - **Si** (K >= 0)
        Continuer vers le prochain i.
      - Calculez et saisissez une nouvelle valeur pour α_j en utilisant (5) et (10).
      - **Si** $(|\alpha_j - \alpha_j^{old}| < 10^{-5})$
        Continuer vers le prochain i.
      - Déterminez la valeur pour α_i en utilisant (6).
      - Calculez b_1 et b_2 en utilisant (7) et (8) respectivement.
      - Calculer b avec (9).

- Num_changer_alphas := Num_changer_alphas + 1.
- **Finsi**
- **Fin pour**
- **Si** (Num_changer_alphas = 0) Passes := passes + 1
- **Sinon** Passes := 0
- **Finsi**
- **End While**

7- Conclusion

Dans ce chapitre, nous avons présenté les différentes phases de notre système de reconnaissance des chiffres manuscrits hors-ligne à base des SVMs. Nous avons commencé par l'acquisition qui sert à scanner notre image (RGB. Niveaux de gris, Noir & Blanc) qui contient des chiffres écrit manuellement, suivi par le prétraitement qui regroupe plusieurs opérations : filtrage, binarisation, fermeture, squelettisation, normalisation, détection de contour et segmentation. Dans L'extraction des caractéristiques on a utilisés 3 moments qui vont servir dans la phase d'apprentissage.

CHAPITRE V :

Implémentation Et Application

1- Introduction

2- L'Environnement de développement

3- La Base de données

4- Apprentissage et Test

5- Apprentissage des SVMs

6- Métriques de performances

7- Description de plateforme

8- Résultat expérimentale

9- Comparaison des résultats avec d'autres classifieurs

10- Conclusion

1- Introduction

Dans ce chapitre nous allons présenter l'environnement de développement du système de reconnaissance des chiffres que nous avons réalisé, ainsi que l'implémentation SVM avec l'algorithme SMO en utilisant différents noyaux avec différents paramètres pour évaluer les performances des SVM.

2- L'Environnement de développement

Pour l'environnement logiciel, on a utilisé Windows 7 Édition Intégrale, processus i5-4210U, Ram 8G, système 64 bit. Notre choix s'est porté sur le langage python. Pour l'environnement de développement et d'exécution des programmes python, nous avons utilisé python 2.7.2 et pour l'interface en a utiliser le Qt design.

2-1- Le Langage Python

Python possède un certain nombre de caractéristiques qui ont largement contribué à son énorme succès :

- ✓ Python est portable est gratuit.
- ✓ La syntaxe de Python est très simple et combinée à des types de données évolués (listes, dictionnaires...).
- ✓ Python est orienté-objet. Il supporte l'héritage multiple et la surcharge des opérateurs. Dans son modèle objets, et en reprenant la terminologie de C++, toutes les méthodes sont virtuelles.
- ✓ Python est dynamique, orthogonal, réflectif, et introspectif.
- ✓ Enfin, Python est un langage qui continue à évoluer, soutenu par une communauté d'utilisateurs enthousiastes et responsables, dont la plupart sont des supporters du logiciel libre. Parallèlement à l'interpréteur principal, écrit en C et maintenu par le créateur du langage, un deuxième interpréteur, écrit en Java, est en cours de développement.

3- La Base de données

Les expériences ont été conduites sur une base de données composée de chiffres manuscrits isolés sous forme d'images niveau de gris (figure V.3).

Cette base, appelée la base de données MNIST (Mixed National Institute of Standards and Technology) est une grande base de chiffres manuscrits qui est couramment utilisé pour divers systèmes de traitement d'image avec une taille 70000 échantillons (figure V.1). [102]

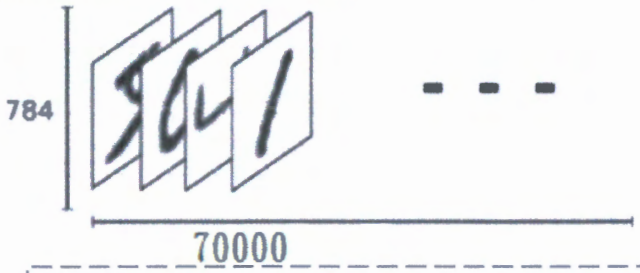


Figure V. 1: forme des chiffres dans la base

Chaque image dans MNIST possède une étiquette (label) correspondante, un nombre entre 0 et 9 représentant le chiffre tracé dans l'image (figure V.2).

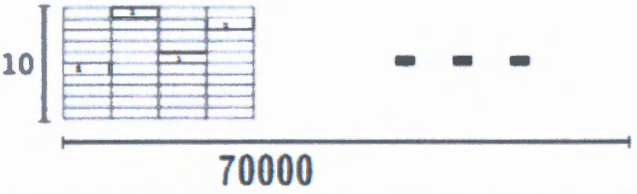


Figure V. 2: forme des étiquettes dans la base

La base de données est également largement utilisée pour l'apprentissage et les tests.

Elle a été créé par "re-mélange" les échantillons provenant des ensembles de données d'origine du NIST. En outre, les images en noir et blanc du NIST ont été normalisées insérer dans une boîte 28x28 pixel englobant. [103]

La base de données MNIST contient 60.000 images d'apprentissage et de 10.000 images de test (Table V.1).

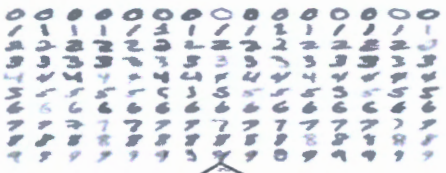
Classes	APPRENTISSAGE	TEST	TOTALE
0	5923	980	6903
1	6742	1135	7877
2	5958	1032	6990
3	6131	1010	7141
4	5842	982	6824
5	5421	892	6313
6	5918	958	6876
7	6265	1028	7293
8	5851	974	6825
9	5949	1009	6958
TOTALE	60000	10000	70000

Tableau V. 1 : Répartition de la base MNIST pour les chiffres.

Figure V. 3: Exemples de la base de données MNIST

4- Apprentissage et Test

Dans notre cas nous avons utilises 60000 pour l apprentissage et 10000 pour le test.



MNIST-Données (MNIST-Data)

[[0, 0, 0, ..., 0, 0, 0]
[0, 0, 0, ..., 0, 0, 0]
[0, 0, 0, ..., 0, 0, 0]
...,
[0, 0, 0, ..., 0, 0, 0]]

MNIST-Cible (MNIST-Target)

[0, 0, 0, ..., 0, 9, 9]

Data_Train	Label_ Train	Data_ Test	Label- Test
(Apprentissage -données)	(Apprentissage- Etiquettes)	(Test-Données)	(Test- Etiquettes)
[[0, 255, 0, ..., 0, 0, 0] [0, 0, 255, ..., 0, 0, 0] [0, 0, 0, ..., 0, 255, 0] ..., [0, 255, 0, ..., 0, 0, 0]]	[0, 4, 0, ..., 0, 9, 0]	[[0, 255, 0, ..., 0, 0, 0] [0, 0, 255, ..., 0, 0, 0] [0, 0, 0, ..., 0, 255, 0] ..., [0, 255, 0, ..., 0, 0, 0]]	[1, 0, 0, ..., 0,7, 0]

Figure V. 4: Test et Apprentissage de la base de données MNIST

5- Apprentissage des SVMs

Nous avons utilisé pour l'apprentissage multi classe les paramètres suivants :

5-1- Paramètre de régularisation C

Nous avons utilisé dans la phase d'apprentissage le hyper-paramètre C qui sert à fixer le compromis entre la minimisation de l'erreur d'apprentissage et la maximisation de la marge c'est un facteur de coût positif qui pénalise les exemples de formation mal classés. En pratique, le comportement du SVM est sensible à la valeur de C uniquement si les données d'apprentissage ne sont pas séparables. Dans ce cas, il existe des valeurs critiques qui peuvent compromettre la performance du classifieur. Une très grande valeur de C (quelques milliers) peut faire que la fonction objective minimisée par le SVM ne soit plus convexe et empêcherait sa convergence. Une très faible valeur de C tend à diminuer la capacité du classifieur.

5-2- Réglages des seuils de tolérance et å (=EPS) :

L'algorithme SMO est basé sur l'évaluation des conditions de KKT. Quand tous les multiplieurs vérifient ces conditions, l'algorithme s'arrête. En pratique ces conditions sont vérifiées à une erreur près *å*. Platt [103] a mentionné dans son article que *å* est typiquement choisi dans l'intervalle 10^{-2} et 10^{-3}.

La maximisation de la marge doit être tolérée de l'ordre : 10%, 0.1%...etc.

Dans notre cas nous avons fixé *å*, et la tolérance a 10^{-3}.

5-3- Choix du noyau

Quatre noyaux ont été utilisés dans notre travail :

Noyau linéaire, Noyau polynomial avec le paramètre de dégrées, Noyau RBF avec le paramètre sigma, Noyau Sigmoïde.

5-4- Prétraitement de la Base

Initialement la base est non normalisée c'est pourquoi on a besoin d'une étape de normalisation: correction de l'inclinaison des lignes (slant normalization), squelettisation et détection de contour.

Il est a noté que les prétraitements se font sur la base d'apprentissage et la base de test.

5-5- Protocole expérimental

Les expérimentations sont réalisées sur la base de données MNIST. Les méthodes d'extraction des caractéristiques utilisées sont : moments Hu, HOG et Données de pixels brutes (Raw pixel data).

6- Métriques de performances

On a utilisé deux métriques sont:

6-1- Taux de reconnaissance

La métrique intuitive utilisée est la précision du modèle appelée aussi le taux de reconnaissance.

Elle représente le rapport entre le nombre d'exemples correctement classés et le nombre total d'exemples testés. L'équation v.1 donne la formule utilisée.

$$P = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i)) \qquad \text{avec} \quad L = \begin{cases} 1 \text{ si } y_i = \hat{f}(x_i) \\ 0 \text{ sinon} \end{cases} \qquad \text{v.1}$$

Généralement, la précision est donnée sous forme de pourcentage ce qui nécessite de multiplier la précision de l'équation v.1 par 100. [104]

6-2- La matrice de confusion

La matrice de confusion est un critère d'évaluation de la classification. Cette matrice permet d'estimer les taux de bonne classification (ou de bonne reconnaissance) pour chaque modèle et également le taux global du système [03].Par exemple, si on considère un système de classification dont le but est de classer des prototypes en classe1 et en classe2,

On va vouloir savoir combien de prototypes seront faussement considérés comme du classe1 et combien de classe2 ne seront pas identifiés comme tels.

7- Description de plateforme

Nous présentons dans cette section la plate forme de reconnaissance de chiffres manuscrits à base de SVM en se basant sur les algorithmes qu'on a proposés.

Pour présenter les fonctionnalités de notre application, nous utilisons des captures d'écran qui montrent les différentes options intégrées :

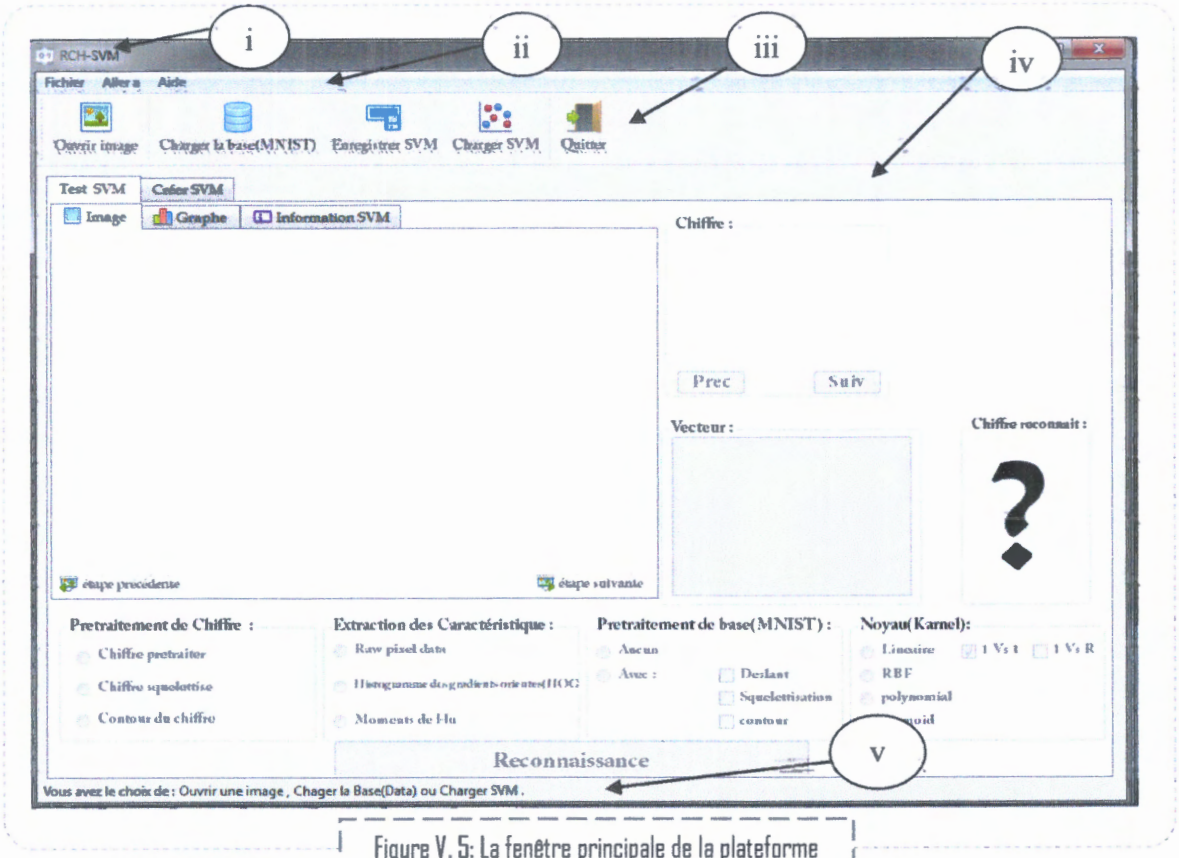


Figure V. 5: La fenêtre principale de la plateforme

i : Le nom de l'application RCH-SVM (abréviation de la Reconnaissance des CHiffres a base SVM :)

ii : Menu : Est considéré comme le point de contrôle de l'application. Il contient 3 items : Fichier, aller à, aide.

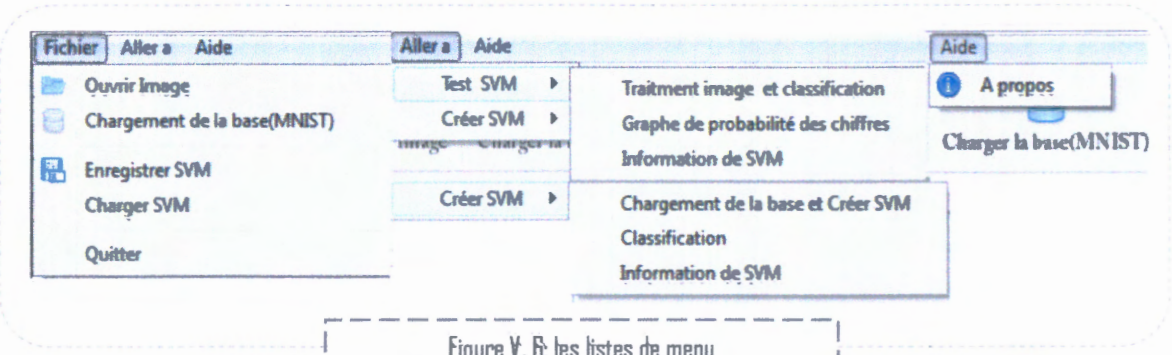


Figure V. 6: les listes de menu

- **Fichier :** ouvrir image, chargement de la base, enregistrer SVM, charger SVM, Quitter fonction les même options des items de barre d'outille.
- **Aller a :** test SVM et Créer SVM : leur option sont des raccourci des widgets comme : quand en click sur traitement image et classification la page image s'affichée.

iii : la barre d'outille, nous proposons un accès rapide aux fonctions les plus fréquemment utilisées dans le menu. Cela nous évite de devoir chercher ces fonctions dans les onglets.

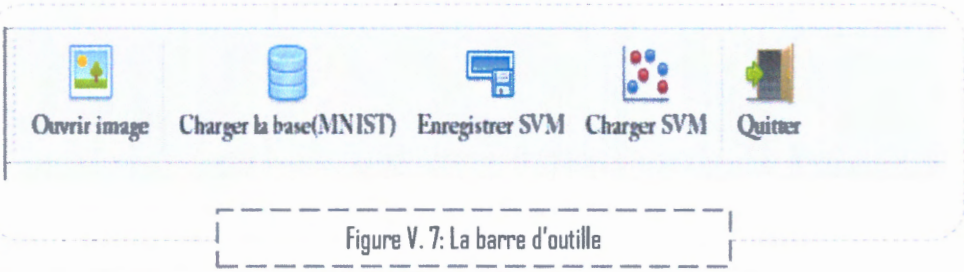


Figure V. 7: La barre d'outille

- **Ouvrir image :** pour ouvrir 1 image de trois type : jpeg, png et bmp et fait les traitements.
- **Charger la base:** pour charger la base MNIST.
- **Enregistrer SVM :** pour l'enregistrement le classifieur SVM en format (.pkl) (format pou sauvegarder les données de classifieur), les informations et les paramètres des SVM en format (. jpeg) et en format (.txt).
- **Charger SVM :** pour charger le classifieur SVM et le fichier texte qui le suive.
- **Quitter :** pour quitter l'application

iv : Widgets, contient 2 items : Test image et Créer SVM

1- Test SVM, Contient la page image, graphe et information SVM

1-1- Page Image

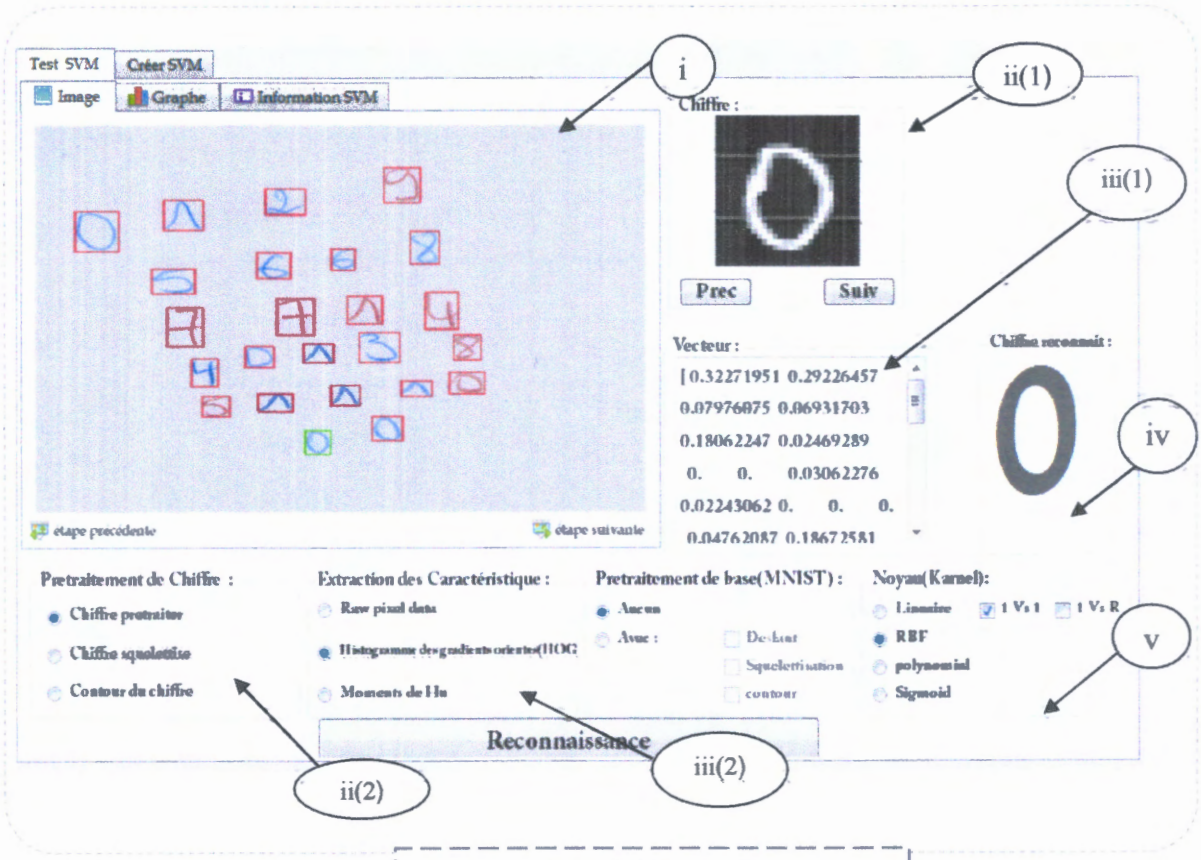


Figure V. 8: Page image

- **i :** Pour afficher l'image. Les buttons (l'étape précédente et l'étape suivante : sert pour afficher les résultats des traitements sur l'image (filtrage, binarisation, normalisation…)).
- **ii :** (1) : l'image qui représente le chiffre .Les buttons précédent et suivante pour défiler les autres images

  (2) : pour afficher les résultats de prétraitement.
- **iii :** (1) : vecteur des caractéristiques

  (2) : pour le choix L'extraction des caractéristiques
- **iv :** le chiffre connu.
- **v:** les paramètres de classifieur SVM et les traitements sur le chiffre d'image et sur la base (chargement manuelle de classifieur SVM).

1-2- Le graphe de probabilité :

Le pourcentage de reconnaissance de chiffre détecté.

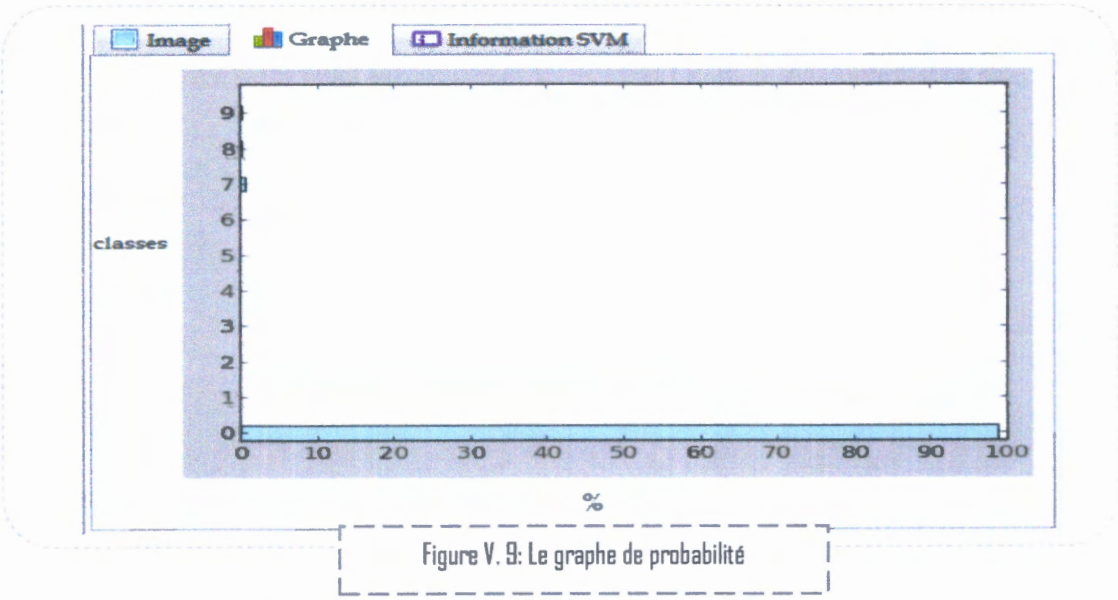


Figure V. 9: Le graphe de probabilité

1-3- les informations de SVM

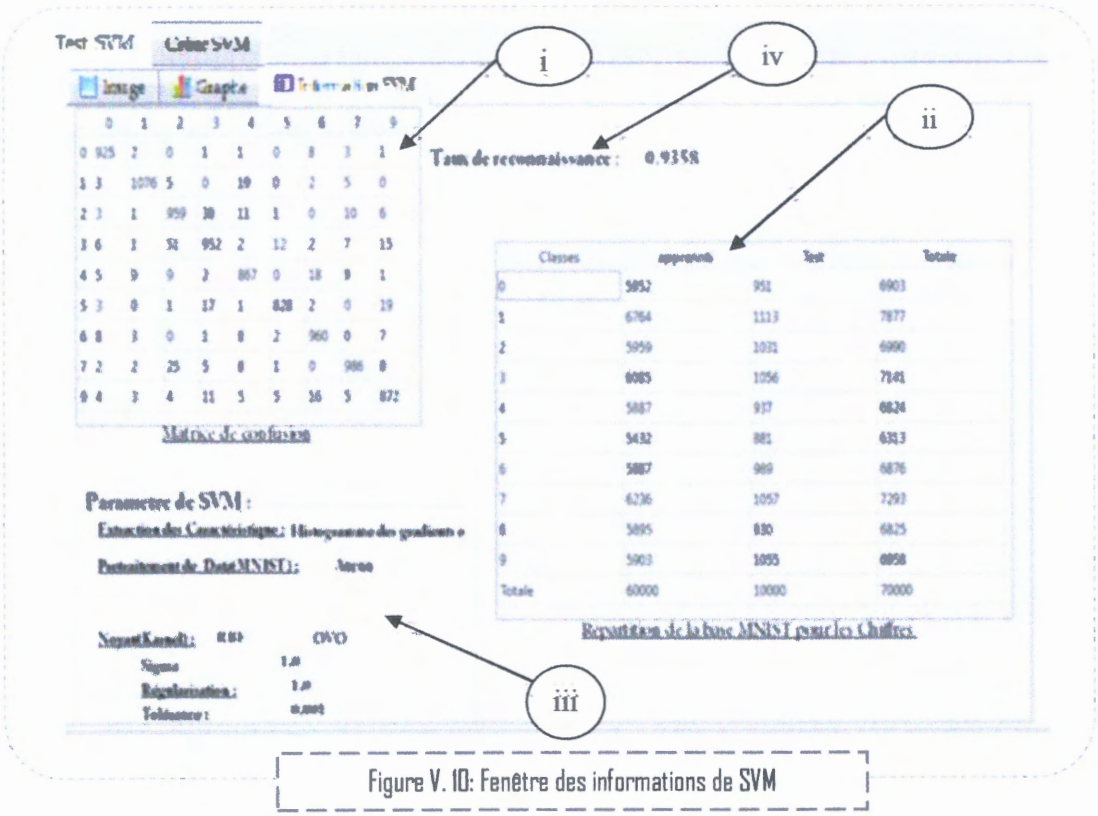


Figure V. 10: Fenêtre des informations de SVM

- **i :** matrice de confusion

- **ii :** table de Répartition de la base MNIST pour les chiffres.

- **iii :** paramètre de SVM

-**iv :** taux de reconnaissance

2- Créer SVM

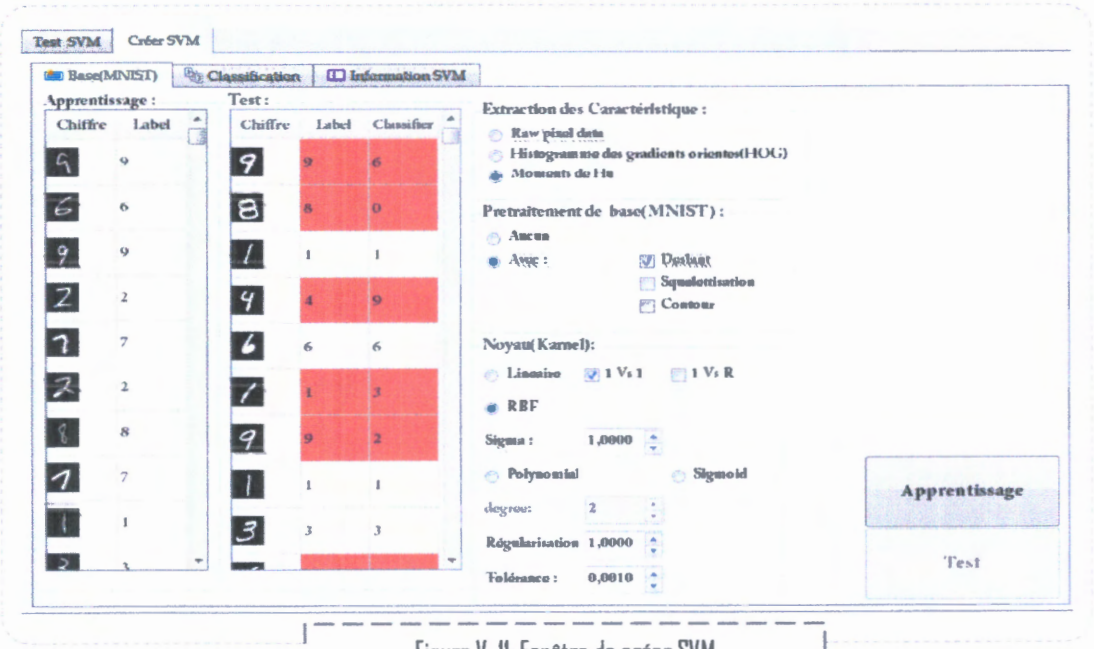


Figure V. 11: Fenêtre de créer SVM

2-1- Base (MNIST) : contient les paramètres de classifieur SVM et le traitement sur les chiffres d'image et sur la base.

- **Apprentissage :** on a 60000 chiffres avec leurs labels (échantillons).
- **Test :** on a 10000 chiffres avec les labels et leur classification, les lignes coloré en rouge est les chiffres mal reconnait. (après le teste)

2-2- Classification

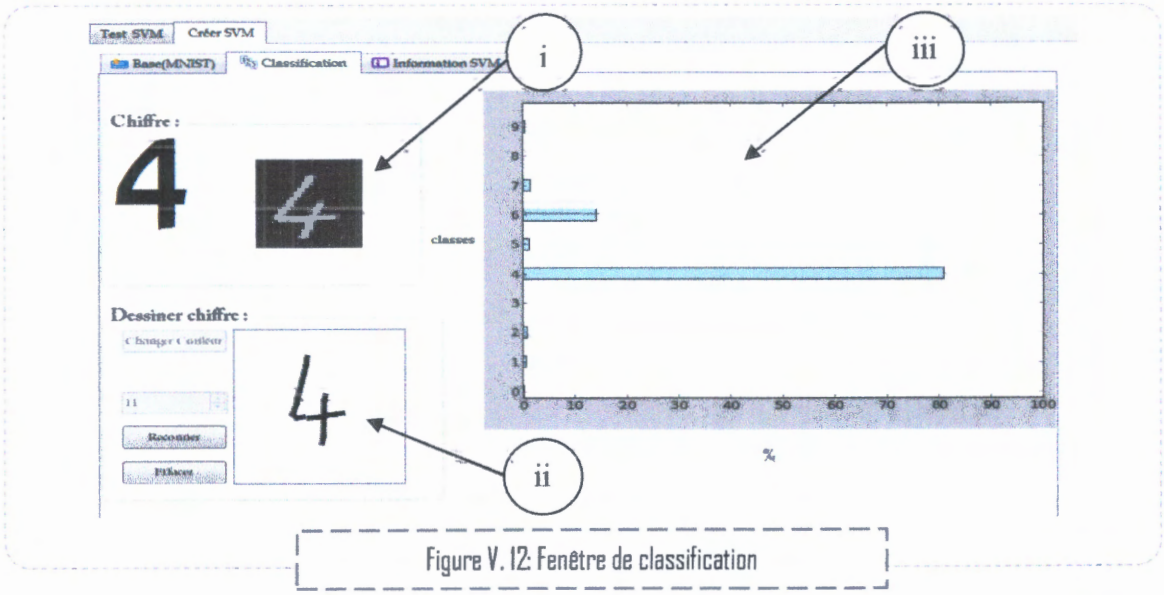


Figure V. 12: Fenêtre de classification

- **i:** Chiffre traité, **ii :** Pour Dessiner le chiffre, **iii:** la probabilité de reconnaissance de chiffre.

➢ **v :** barre des commentaires : à chaque fois accéder a une étape s'affichera plus des détailles sur cette opération.

8- Résultat expérimentale

Dans cette section, nous évaluons la classification multi-classe en utilisant l'approche un contre un et un contre tous pour les différents méthodes d'extraction des caractéristiques et différents paramètres des noyaux et l'état de la base (normalisé, non normalisé) en calculer la matrice de confusion et taux de reconnaissance dans chaque étape.

1- les paramètres : un contre un, Extraction des caractéristiques : HOG, la base et Non normalisé (N-Norm), choix du noyau Kernel linéaire, régularisation (C) =1, tolérance (tolé)=0.001.

	0	1	2	3	4	5	6	7	8	9
0	910	3	2	2	3	0	5	3	5	18
1	3	1066	7	0	20	0	3	4	8	2
2	7	2	911	58	19	2	0	22	8	2
3	6	2	75	983	4	12	3	21	12	28
4	7	11	15	14	819	3	25	9	11	23
5	2	0	2	15	1	802	6	1	25	27
6	8	7	0	1	13	2	943	0	15	0
7	8	4	35	20	9	1	0	926	19	35
8	7	11	8	14	15	16	32	13	800	14
9	32	6	2	34	27	28	6	41	25	854

Tableau V. 2: Matrices de confusion pour 1 vs 1 : (HOG, la base N-Norm, noyau linéaire, C=1, tolé=0.001).

Résultat : taux de reconnaissance 89.24 % Et l'erreur 10.76%

2- les paramètres : un contre tous, Extraction des caractéristiques : HOG, la base N-Norm, noyau Kernel linéaire, C=1, tolé=0.001.

	0	1	2	3	4	5	6	7	8	9
0	899	11	3	8	3	1	6	3	5	12
1	2	1063	9	0	16	0	3	4	11	5
2	13	3	881	67	21	3	1	21	17	4
3	8	2	79	856	16	21	3	21	16	34
4	9	19	31	20	752	1	44	10	23	28
5	2	0	4	27	3	776	6	5	33	25
6	13	7	1	2	9	7	933	0	15	2
7	12	5	37	24	17	0	1	902	23	36
8	7	11	36	21	22	13	49	19	736	16
9	50	6	7	39	37	22	7	63	42	782

Tableau V. 3: Matrice de confusion pour 1 Vs Reste (HOG, la base N-Norm, noyau linéaire, C=1, tolé=0.001).

Résultat : taux de reconnaissance 85.80 % et l'erreur 14.2%

En gardant les mêmes paramètres sauf la normalisation nous avons obtenu un taux de reconnaissance égale à 92.68% pour un contre tous et 94.25 % pour un contre un.

3- les paramètres : un contre tous, Extraction des caractéristiques :

HOG, la base et normalisé (deslant), noyau Kernel RBF,sigma=1.0, C =1, tolé =0.001.

	0	1	2	3	4	5	6	7	8	9
0	847	1	3	3	2	24	8	2	2	55
1	5	995	11	6	4	8	15	20	20	16
2	10	7	811	64	9	6	58	32	32	14
3	11	6	73	752	40	25	46	36	36	38
4	7	35	29	28	27	87	19	35	35	65
5	9	6	14	56	646	37	17	28	28	40
6	28	10	2	15	38	769	0	31	31	38
7	17	21	46	26	8	5	857	24	24	32
8	7	47	38	25	41	66	26	55	55	50
9	116	37	16	43	39	66	57	45	45	545

Tableau V. 4: Matrice de confusion pour I Vs Reste (HOG, la base deslant, noyau RBF, sigma=1, C=1, tolé=0.001).

Résultat : taux de reconnaissance 73.92 % et l'erreur 25%

✓ Le taux de la convergence de la stratégie un contre un est meilleur que taux de la convergence de la stratégie un contre tous (reste) mais on trouve le contraire concernant le temps d'exécution. Donc on va continuer avec la stratégie **un contre un** :

1- Extraction des caractéristiques : Moments de Hu, la base N-Norm, noyau Kernel polynomial (poly), degré =2, C =1, tolé =0.001.

	0	1	2	3	4	5	6	7	8	9
0	860	6	0	34	0	0	7	1	16	27
1	33	911	3	5	20	0	32	5	56	48
2	17	0	294	653	7	0	6	3	36	15
3	12	0	3	1003	1	0	0	4	19	14
4	39	9	12	42	532	0	113	3	74	113
5	16	0	2	356	2	322	16	0	129	38
6	62	4	1	1	5	12	817	0	84	3
7	34	2	20	144	9	0	3	720	42	83
8	14	6	3	89	1	0	36	0	774	7
9	99	2	1	149	18	1	4	12	69	700

Tableau V. 5: Matrice de confusion pour I Vs I (Hu, la base N-Norm, noyau poly, degré =2, C=1, tolé=0.001).

Résultat : taux de reconnaissance 69.33% et l'erreur 30.67%.

2- Extraction des caractéristiques : Moments de Hu, la base N-Norm, noyau Kernel RBF, sigma =1, C =1, tolé =0.001.

	0	1	2	3	4	5	6	7	8	9
0	609	60	4	78	16	0	0	2	179	3
1	21	896	6	130	0	1	0	19	39	1
2	122	63	77	100	128	5	146	143	76	171
3	199	250	22	266	39	4	2	30	218	26
4	418	13	32	50	141	0	43	34	123	83
5	159	148	28	212	42	2	9	50	191	40
6	57	6	37	15	83	1	403	228	18	141
7	27	32	24	47	29	2	137	565	33	161
8	326	83	7	161	65	1	2	5	265	15
9	97	7	79	39	113	3	184	242	47	244

Tableau V. 6: Matrice de confusion pour 1 Vs 1 (Hu, la base N-Norm, noyau RBF, sigma=1, C=1, tolé=0.001).

Résultat : taux de reconnaissance 34.68% et l'erreur 65.32%

3- Extraction des caractéristiques : HOG, la base N-Norm, noyau Kernel sigmoïde, C =1, tolé =0.001.

	0	1	2	3	4	5	6	7	8	9
0	951	0	0	0	0	0	0	0	0	0
1	1113	0	0	0	0	0	0	0	0	0
2	1031	0	0	0	0	0	0	0	0	0
3	1056	0	0	0	0	0	0	0	0	0
4	9	0	0	0	0	0	0	0	0	0
5	16	0	0	0	0	0	0	0	0	0
6	62	0	0	0	0	0	0	0	0	0
7	34	0	0	0	0	0	0	0	0	0
8	14	0	0	0	0	0	0	0	0	0
9	99	0	0	0	0	0	0	0	0	0

Tableau V. 7: Matrice de confusion pour 1 Vs 1 (HOG, la base N-Norm, noyau sigmoïde, C =1, tolé=0.001).

Résultat : taux de reconnaissance 11.13% et l'erreur 88.87%

- Concernait choix 1, 2 et 3 : moment de Hu (avec noyau Polynomial) donne des résultats moyens et prend du temps pour l'exécution (plus de 42 min).HOG (avec noyau sigmoïde) donnes des mal résultats et pour le temps d'exécution la même chose que polynomial.

4- Extraction des caractéristiques : HOG, la base et normalisé (squelette), noyau Kernel linéaire, C =1, tolé =0.001.

	0	1	2	3	4	5	6	7	8	9
0	884	1	7	3	1	0	19	8	4	24
1	6	1040	6	2	21	1	5	13	14	5
2	11	9	775	118	14	1	23	40	26	14
3	8	7	85	767	20	40	8	43	27	51
4	14	12	43	19	650	12	92	10	33	52
5	3	2	3	66	4	688	39	5	21	50
6	34	25	16	2	58	35	717	0	68	34
7	13	10	40	51	6	7	2	888	10	30
8	12	35	34	15	20	25	86	3	640	60
9	81	31	19	41	28	68	37	48	84	618

Tableau V. 8: Matrice de confusion pour 1 Vs 1 (HOG, la base, normalisé : squelette, noyau linéaire, C =1, tolé=0.001).

Résultat : taux de reconnaissance 76.67% l'erreur 23.33.

5- Extraction des caractéristiques : HOG, la base et normalisé : (deslant, contour), noyau Kernel RBF, sigma=0.0073, C =2.8, tolé =0.001.

	0	1	2	3	4	5	6	7	8	9
0	903	2	2	0	2	0	11	2	2	27
1	1	1069	5	0	30	1	0	1	1	5
2	12	0	878	47	11	3	9	33	24	14
3	22	0	44	844	12	27	20	38	30	19
4	1	10	12	5	680	6	103	16	11	93
5	6	2	10	48	16	691	30	16	37	25
6	13	6	0	8	24	15	869	3	9	42
7	4	4	27	15	32	2	4	898	9	62
8	9	1	34	22	25	30	34	9	734	23
9	71	3	12	17	69	14	71	37	11	750

Tableau V. 9: Matrice de confusion pour 1 Vs 1 (HOG, base normalisé (deslant, contour), noyau RBF, sigma=0.0073 C=2.8, tolé=0.001)

Résultat : taux de reconnaissance 83.16% et l'erreur 16.84 %.

6- Extraction des caractéristiques : HOG, la base et normalisé (deslant), noyau Kernel RBF, sigma =1, C =1, tolé = 0.001.

	0	1	2	3	4	5	6	7	8	9
	934	2	0	0	0	0	10	2	1	2
1	0	1090	2	0	15	0	0	3	2	1
2	3	0	983	25	7	0	0	6	7	0
3	1	0	22	1008	0	6	0	8	4	7
4	3	3	4	5	886	1	13	5	3	14
5	3	0	0	5	2	851	3	1	9	7
6	3	5	0	0	5	2	967	0	7	0
7	2	0	17	3	10	0	0	1007	4	14
8	8	3	2	6	5	2	10	1	890	3
9	2	2	2	14	15	7	2	25	3	683

Tableau V. 10: Matrice de confusion pour 1 Vs 1 (HOG, la base normalisé (deslant), noyau RBF, sigma =1, C =1, tolé=0.001)

Résultat : taux de reconnaissance 95.99% et l'erreur 4.01%.

✓ La méthode HOG (noyau RBF) ne prend pas temps pour l'exécution et nous donne des bons résultats Table V.10, la normalisation de base (squelette et prend temps pour l'exécution)

7- Extraction des caractéristiques : Donnes des pixels brutes, la base et normalisé : deslant, noyau Kernel RBF, sigma= 0.008, C=3, tolé = 0.001.

	0	1	2	3	4	5	6	7	8	9
0	945	0	0	0	0	1	4	0	0	1
1	0	1110	1	0	0	0	0	1	1	0
2	1	1	1019	1	1	0	0	1	4	0
3	0	0	5	1040	0	5	0	3	3	0
4	0	1	0	0	929	0	2	1	1	3
5	2	0	0	4	0	868	4	0	2	1
6	3	1	2	0	0	0	981	0	2	0
7	0	2	3	0	0	0	0	888	2	5
8	0	3	1	1	2	2	1	1	640	0
9	2	2	1	0	8	2	0	8	2	1030

Tableau V. 11: Matrice de confusion pour 1 Vs 1 (Raw, la base normalisé (deslant), noyau RBF, sigma= 0.008, C =3, tolé=0.001)

Résultat : taux de reconnaissance 98.86 % et l'erreur 1.14%

- Malgré que la méthode de données des pixels brutes (avec noyau RBF) est prend temps pour l'exécution mais donne très bonne résultats.

Les Méthodes	Données des pixels Brutes		Histogramme De Gradient Orienté									Moments De Hu	
Noyaux	RBF		Lineaire			RBF			Segmoid	Poly		poly	RBF
Parametres Noyau	sigma = 0.008	sigma = 0.005				Sigma =0.0073	Sigma =1	Sigma =1		Degré =1	Degré= 3	Degré = 2	sigma = 1
	C= 3	C= 2	C= 1	C=2	C= 1	C= 1	C= 1	C= 1	C=2	C= 1	C= 1	C= 1	C=1
	Tol= 10^{-3}	Tol= 10^{-3}	Tol= 10^{-3}	Tol= 10^{-3}	Tol= 10^{-3}	Tol= 10^{-3}	Tol= 10^{-3}	Tol= 10^{-3}	Tol= 10^{-3}	Tol= 10^{-3}	Tol=10^{-3}	Tol= 10^{-3}	Tol= 10^{-3}
L'etat de la base	deslant	deslant	Non-norm	squelette	Non-norm	deslant contour	Deslant	deslant	Non-norm	deslant	deslant	Non-norm	Non-norm
La strategie	1 Vs 1	1 Vs 1	1 Vs 1	1 VS 1	1 Vs R	1 Vs 1	1 Vs 1	1 Vs R	1 Vs 1	1 Vs 1	1 Vs 1	1 Vs 1	1 Vs 1
0	945	943	910	884	899	903	934	847	951	936	951	860	609
1	1110	1110	1066	1040	1063	1069	1090	995	0	1078	0	911	896
2	1019	1017	911	775	881	878	983	811	0	945	0	294	77
3	1040	1039	983	767	856	844	1008	752	0	971	0	1003	266
4	929	928	819	650	752	680	886	605	0	828	0	532	141
5	868	869	802	688	776	691	851	646	0	811	0	322	2
6	981	981	943	717	933	869	967	769	0	962	0	817	403
7	888	1043	926	888	902	898	1007	857	0	942	0	720	565
8	640	918	800	640	736	734	890	565	0	867	0	774	265
9	1030	103	854	618	782	750	683	545	0	950	0	700	244
	98.86	**98.78**	**89.24**	76.67	85.80	83.16	**95.99**	73.92	11.13	**92.90**	11.13	69.33	34.68

Table VI. 12: Résumer des matrice de confusion précédents et avec d'autre testes

Donc:

- Le paramètre de régularité C prend la valeur [1…3].
- A chaque fois le degré de polynomial augmente le taux de reconnaissance diminué.
- A chaque fois en donnes des petites valeurs de sigma nous donné des bons résultats.
- L'état de la base normalisé (Deslant (Correction De L'inclinaison Des Lettres)) donne des bons résultats.
- Données de pixels brutes nous donne le meilleur résultat avec le taux de reconnaissance : 98.86 % par rapport a la méthode HOG et moments de Hu.

9- Comparaison des résultats avec d'autres classifieurs

Le tableau V. 13 montre une comparaison de notre résultat avec d'autres classifieurs : HMMs [57] (méthode markovienne), Kppv [104] (k plus proche voisin) et les réseaux de neurone [104]. Nous remarquons que notre résultat est très satisfaisant.

	SVM	HMMs	Kppv	Réseaux Neurone
L'erreur %	1.14	2.7	3.09	2.5

Tableau V.13: Comparaison des résultats avec d'autres classifieurs

10- Conclusion

Ce chapitre a été consacré à l'évaluation de la reconnaissance des chiffres manuscrits a base de classifieur SVM. Les résultats obtenus montrent que le taux de la reconnaissance dépend du choix des méthodes d'extraction des caractéristiques et du noyau et du réglage de ses paramètres et l'état de la base.

Conclusion Générale

L'objectif de ce travail est l'étude de la reconnaissance des chiffres manuscrits avec l'implémentation des machines à vecteurs de support ou SVM (Support Vector Machines).

En général, la plupart des méthodes d'apprentissage machine comme réseau de neurones possèdent un grand nombre de paramètres d'apprentissage à fixer par l'utilisateur. En contrepartie, la formulation élégante de SVM laisse très peu de place aux paramètres utilisateurs.

Différents algorithme d'optimisation sont disponibles parmi eux l'algorithme SMO (optimisation minimale et séquentielle du risque) qui est un algorithme simple et rapide proposé pour résoudre le problème de la programmation quadratique des *SVM* sans la nécessité de stocker une grande matrice en mémoire et sans une routine numérique itérative pour chaque sous-problème.

La méthode des SVM que nous avons implémentée a été testée sur des chiffres manuscrits de base (MNIST) avec 60000 chiffres pour l'apprentissage et 10000 pour le test en considérant les deux stratégies, un contre tous et un contre un, en utilisant différents types de noyaux.

Les résultats obtenus montrent que le taux de reconnaissance dépend des choix des méthodes d'extraction des caractéristiques ainsi que du noyau et du réglage de ses paramètres.

En utilisant les données des pixels brutes, une base normalisée, noyau Kernel RBF, régularisation =3, tolérance=0.008 on a eu un très bon taux de reconnaissance qui atteint 98.86 %.

Parmi les difficultés rencontrées dans cette expérience c'est le temps critique concernant la phase d'apprentissage et qui dure environ 42 min.

En perspectives nous proposons de développer d'autres méthodes d'extraction des caractéristiques et d'autres types de noyaux et pourquoi pas de les utiliser sur d'autres bases de données dont le but d'améliorer le taux de reconnaissance.

BIBLIOGRAPHI

[01]	F. Ali and Th. Pavlidis, 'Syntactic Recognition of handwritten Numerals '. IEEETransaction on Systems, Man and Cybernetics, vol. n°7, 1977.
[02]	C. Tappert, C. Y. Suen, and T. Wakahana, ' The state of the art in on-line handwrittin grecognition '. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 12,1990.
[03]	N. Ayat, ' Sélection de modèle automatique des machines à vecteurs de support ' application à la reconnaissance d'images de chiffres manuscrits, Montréal, Canada. 2003.
[04]	V.Vapnik, ' the nature of statistical learning throry '. springer-verlag, New_York, USA. 1995.
[05]	V. K. Govindan, 'Character recognition - A review '; Department of Electrical communication engineering, India; CNED'94, 1990.
[06]	Ben Amara ' Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée ' Thèse de doctorat, Université des sciences, des Techniques et de médecine de Tunis II, 1999.
[07]	J. Anigbogu: 'Reconnaissance de textes imprimés mutifontes à l'aide de modèles stochastiques et métriques '. Thèse de doctorat, Université de Nancy I, 1992.
[08]	M.M.M. Fahmy, S.Al Ali: 'Automatic recognition of handwritten Arabic characters using their geometrical features '. Studies in informatics and control journal (SIC journal), vol. 10, No 2, 2001.
[09]	Mr Azizi Rebiai , 'Une approche hybride pour la reconnaissance d'ecriture arabe manuscrit', Traitement du signal , 2006/2007
[10]	P. Smrž et al: 'Off-line Recognition of Cursive Handwritten Czech text'. Université de Masaryk, Février 1998
[11]	Al-Rashaideh H: 'Preprocessing phase for Arabic Word Handwritten Recognition '. Institut d'informatique et autimatisme, Tom 6, N0 1,2006, cmp. Russie, February 26, 2006.

[12]	P.M. Lallican, C. Viarp-Gaudin, S. Knerr : ' From off-line to on-line handwriting recognition '. Proc. 7th workshop on frontiers in handwriting recognition, Amsterdam 2000.
[13]	A. Belaïd : ' Reconnaissance automatique de l'ecriture et du document '. LORIA-CNRS, Campus scientifique B.P. Vandoeuvre-Lès-nancy, France.
[14]	C. Parisse. '*Reconnaissance* hors ligne de mots manuscrits sans contraintes d'écriture, ni de vocabulaire'. Actes de CNED'92, Nancy, juil. 1992.
[15]	C. Parisse. 'Global word shape processing in off-line recognition of handwriting'. I.E.E.E., Trans. On An. And Mach. Int., 18 (4), April 1996.
[16]	K.M. Sayre. 'Machine recognition of handwritten words: A project report. Pattern Recognition', septembre 1973.
[17]	R.G. Casey, E. Lecolinet. 'A survey of methods and strategies in character segmentation '.IEEE Trans. Pattern Analysis and Machine Intelligence, juillet1996.
[18]	S. Haitaamar : ' segmentation de texte en caractere pour la reconnaissance optique de l'ecriture arabe'. Université EL-HADJ LAKHDHAR Batna, Juillet 2007
[19]	J. Park ' Hierarchical character recognition and it's use in handwritten word/phrase recognition', Université de New York, Novembre 1999.
[20]	A. Zahour, " Une Méthode de Reconnaissance de l'Ecriture Arabe Cursive", Thèse de Doctorat, 1990.
[21]	M. R. Gupta, N. P. Jacobson, E. K. Garcia, « OCR binarization and image pre-processing for searching historical documents », Pattern Recognition, 40, 2007.
[22]	J. Bernsen. Dynamic thresholding of grey-level images. In Proc. Eighth Int 'l Conf. On Pattern Recognition, 1986
[23]	W. Niblack. An introduction to digital image processing. Prentice Hall (July 1986), 1986.
[24]	Support de cours : ANALYSE D'IMAGE (techniques instrumentales) École des Mines de Saint-Etienne, 1997.
[25]	Lionel Lacassagne, Light Speed Labelling : un nouvel algorithme d'étiquetage en composantes connexes, 1997.

[26]	P. Kornprobst, Contributions à la Restauration d'Images et à l'Analyse de Séquences: Approches Variationnelles et Solutions de Viscosité, Thèse de Doctorat, Université de Nice-Sophia Antipolis, 1998.
[27]	Maged Mohamed Mahmoud Fahmy, Automatic Recognition Of Handwritten Arabic Characters Using Their Geometrical Feature , 2000.
[28]	A. Belaïd, Analyse de document : Extraction d'éléments (support de cours), 2006.
[29]	S. Snoussi Maddouri, Modèle perceptif neuronal à vision globale -locale pour la reconnaissance de mots manuscrits arabes, 2002.
[30]	P. Gaspart, Acquisition et traitement d'image, cours d'electronique, realisé par Gaspart, université libre du Bruxelles, 2004.
[31]	Rangachar. Kasturi, Lawrence. 0' Gorman, Venu. Govindraju, document image analysis, Sadhana, vol. 27, Part. 1, Feb.2002.
[32]	Sylvain Chevalier, Reconnaissance d'écriture manuscrite par des techniques markoviennes: une approche bidimensionnelle et générique. Thèse de doctorat de l'Université René Descartes - Paris 5, 2004.
[33]	Abderrazak Zahour, et AL,Contribution à la segmentation de textes manuscrits anciens ,2004
[34]	Blumenstein M., Cheng C. K., Liu X. Y., "New Preprocessing techniques for Handwritten Word Recognition", Proc. of the 2nd IASTED Conf. On Visualization, Imaging and Image Processing, pp. 480-484, 2002.
[35]	E.Augustin. Reconnaissance de mots manuscrits par systèmes hybrides Réseaux de Neurones et Modèles de Markov Cachés, thèse de doctorat, Université RENE DESCARTES - PARIS V, 2001.
[36]	Trevor Hastie and Patrice Y. Simard, Metrics and Models for Handwritten Character Recognition. Statistical science Vol. 13 N° 1, 1998.
[37]	Muhammad Sarfraz, Computer -Aided intelligent recognition techniq ues and applications (section: Offline Arabic Character Recognition) King Fahd University of Petroleum and Minerals, Kingdom of Saudi Arabia, 2005.
[38]	I.S.I.Abuhaiba, S.A.Mahmoud, R.J.Green, "Recognition of Handwritten Cursive Arabic Character", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 16, , June 1994.

[39]	R. O. Duda, P. E. Hart, & D. G. Stork. Pattern Classification. (Second edition). New York:NWiley-Interscience, 2001.
[40]	P. Baldi, Y. Chauvin, T. Hunkapiller et M. A. McClure : Hidden Markov Models in molecular biology : new algorithms and applications. Advances in Neural Processing Systems, 1993.
[41]	F. Menasri. Contributions à la reconnaissance de l'écriture Arabe manuscrite. PhD thesis, Univ Paris Descartes, 2008.
[42]	Xavier Dupré, "Contributions à la reconnaissance de l'écriture cursive à l'aide de modèles de Markov cachés," Paris V, 2004.
[43]	E. Augustin. Reconnaissance de mots manuscrits par systèmes hybrides Réseaux de Neurones et Modèles de Markov Cachés. PhD thesis, Université Rene Descartes -Paris V, 2001.
[44]	Khalid Hallouli, "Reconnaissance de caractères par méthodes markoviennes et réseaux baésiens," Paris, Thèse de doctorat 2004.
[45]	Mustapha Amrouch, Reconnaissance de caractères imprimés et manuscrits, textes et documents basée sur les modèles de Markov cachés, Université Ibn Zohr Faculté des Sciences d'Agadir.
[46]	A Belaïd , G. Saon. Utilisation des processus markoviens en reconnaissance de l'écriture. Traitement du signal, Vol. 14, N°. 2, 1997.
[47]	L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of IEEE 77 (2), 1989.
[48]	L. Bréhelin, O. Gascuel, G. Caraux, Hidden Markov Models with patterns and their application to intregated circuit testing, ECML 2000.
[49]	A. Kundu, Y. He, P. Bahl, Recognition of handwritten word: First and second order Hidden Markov Model based approach, in the proceedings of CVPR 88,1988.
[50]	L .Lebart, A. Morineau, J.-P. Fénelon. Traitement des données statistiques. Ed. Dunod, Paris, 1982.
[51]	O. Agazi & S. Kuo. Hidden Markov model based optical character recognition in the presence of deterministic transformations. Pattern Recognition, 1993.

[52]	Piet Mertens, Verbum ex machina (TALN vol. 1), Actes de la 13e conférence sur le traitement automatique des langues naturelles, Volume 1, avril 2006
[53]	C. Chatelain. Systèmes de reconnaissance de l'écriture manuscrite.
[54]	S. J. Russell and P. Norvig, "Artificial Intelligence, A Modern Approche, Second Edition". Pearson Education, Inc., Upper Saddle River, New Jersey, 2002.
[55]	H. Chouaib , 'Sélection de caractéristiques: méthodes et applications, Informatique, le 8 juillet 2011,
[56]	R. Rakotomalala. Arbres de décision. Revue Modulad, 33 :163–187, 2005.
[57]	A. DJEFFAL. Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données.2011/2012
[58]	J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification", In advances in Neural Information Processing Systems12, 2000.
[59]	S. Haykin, Neural Networks: AComprehensive Foundation. New York: Macmillan, 1994.
[60]	A. Nigrin, Networks for Pattern Recognition. Cambridge: MA: The MIT Press, 1993.
[61]	M. Cheriet, N. Kharma , L.C. Liu and C. Y. Suen. Character recognition systems, Aguide for students and practioners. Published by John Wiley & Sons, Inc., Hoboken, New Jersey.2007.
[62]	C. Chatelain. Extraction de séquences numériques dans des documents manuscrits quelconques. PhD thesis, Univ Rouen, 2006.
[63]	Moad Benkiniouar, Mohamed Benmohamed. "Méthodes d'identification et de reconnaissance de visages en temps réel basées sur AdaBoost" Article P2-3,2005.
[64]	C.K.Chow, "Statistical Independence and Threshold Functions", IEEE Transactions of Electrical Computer", 1965.

[65]	D.Weissenbacher, A.Nazarenko. "A bayesian classifier for the recognition of the impersonal occurrences of the it pronoun". In Proceedings of the 6th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC'07). 2007.
[66]	I. Pop, "An approach of the Naive Bayes classifier for the document classification", General Mathematics Vol. 14, No. 4 (2006), 135–138. 2006.
[67]	BOUKHAROUBA Abdelhak. Contribution à la segmentation et à la reconnaissance de l'écriture arabe manuscrite. Traitement du signal, le 16/11/2011.
[68]	Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2), 1998.
[69]	A.Borji, et M. Hamidi. Support Vector Machines for Persian Font recognition. In: International Conference onn Computer, Electrical, Systems Science, and Engineering (CESSE 2007),
[70]	S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", Informatica 31 (2007). 2007.
[71]	Colin Campbell, Yiming Ying "Learning with Support Vector Machines SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING #10", Morgan & Claypool publishers 2011.
[72]	J.-O. Moussafir, "Machines à Noyaux", 2 mars 2005.
[73]	Bernhard Scholkopf, Alexander J. Smola "Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond", the MIT Press 2002.
[74]	G.G. Rajput, S. M. Mali, "Fourier Descriptor based Isolated Marathi Handwritten Numeral Recognition", *International Journal of Computer Applications* Volume 3 – No.4. 2010.
[75]	R. Courant and D. Hilbert. "Méthods of Mathematical Physics", Interscience, 1953.
[76]	J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 1909.

[77]	H. König. Eigenvalue distribution of compact operators. Birkhäuser Verlag Basel Bost on Stuttgart,
[78]	M.Antkowiak, "Artificial Neural Networks vs. Support Vector Machines for Skin Diseases Recognition". 2006
[79]	P. Mahé, L. Ait-Ali :" Projet d'apprentissage statistique SVM pour l'apprentissage non supervise".DEA MVA, Février 2003.hesis, Umea University. 2006.
[80]	John. C. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", Technical Report MSR-TR-98-14, April (1998).
[81]	Bernhard Scholkopf, Alexander J. Smola "Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond", the MIT Press 2002
[82]	V. Vapnik, Statistical Learning Theory. Wiley, New York (1998)
[83]	B. Romaniuk, Modélisation linéaire et non linéaire en rdf statistique (Méthodes à noyaux) support de cours, 2004.
[84]	Scholkopf and A. Smola, "Leaning with Kernels". MIT Press, 2001.
[85]	Thomas G. Dietterich and G.Bakiri. "Solving multiclass learning problems via error-correcting output codes". Journal of Artificial Intelligence Research, 1995.
[86]	M. Schmidt and H. Gish. "Speaker identification via support vector machines", In Proc. ICASSP, 1996.
[87]	C.W. Hsu, C.J. Lin, A comparison of methods for multi-class Support Vector Machines. IEEE Transactions on Neural Networks, 2002.
[88]	B.B. Chaudhuri. Digital Document Processing: Major Directions and Recent,Advances. Springer, London, 2007.
[89]	N.Otsu. « A threshold selection method from grey-level histograms ». IEEE Trans. Syst. Man. Cybern, vol.SMC-8, 1978.
[90]	Henry, Analyse de document : Analyse de l'image (support de cours), 2005.
[91]	Marc Parizeau, Livre : Réseaux de neurones, université LAVAL Automne 2004.
[92]	T. Kanungo and R. M. Haralick. Character recognition using mathematical

	morphology,
[93]	T. Y. Zhang, C. Y. Suen. A Fast Parallel Algorithm for Thinning Digital Patterns. Image Processing and Computer Vision, vol. 27, no. 3, 1984.
[94	Recognition System Based on Genetic Algorithms ". The Internationnal Arab Journal of Information Technology, Vol.3, No.3, July 2006
[95]	Amin A., Fischer S., Parkinson T., Shiu R., "Fast algorithm for skew detection", *SPIE Proceedings*,Vol. 2661 ,Janvier 1996.
[96]	I. Zitouni and R.S Arikaya, "Arabic diacritic restoration approach based on maximum entropy models", *Computer Speech and Language*, Elsevier, vol. 23, 2009.
[97]	Murat KuntReconnaissance des formes et analyse de scènes, traitement de l'information, volume3, 2000
[98]	Dalal, B.Troggs, histograms of oriented gradient for human detection INCVPR, 2015.
[99]	Jonathan Guyomard, spécification des images de cassini pour la reconnaissance automatique de symboles et premiers résultats, 2012.
[100]	M. Sarfraz. Computer-Aided Intelligent Recognition: Techniques and Applications, Wiley, England, 2005.
[101]	M. Cheriet, N. Kharma, C.L. Liu, and C.Y. Suen. Character Recognition Systems: A Guide for Students and Practitioners, John Wiley & Sons, New Jersey-United States of America (USA), 2007.
[102]	L. Deng, « The MNIST Database of Handwritten Digit Images for Machine Learning Research (Best of the Web) », *IEEE Signal Processing Magazine*, vol. 29, no 6, novembre 2012,
[103]	Platt, John C. "Using analytic QP and sparseness to speed training of support vector machines". Advances in neural information processing systems,(1999),