

# Remerciements

*Je souhaite remercier en premier lieu mon directeur de thèse, Mr Moahammed Fakir pour m'avoir accueilli au sein de son équipe. Je lui suis également reconnaissant pour le temps conséquent qu'il m'a accordé, ses qualités pédagogiques et scientifiques, sa franchise et sa sympathie. J'ai beaucoup appris à ses côtés et je lui adresse ma gratitude pour tout cela.*

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

*Je remercie de même, Mr Daoui, Mr El Fachi et Mr Baslam qui ont assisté à mon encadrement durant la formation.*

*Ce travail de thèse ne voudrait rien dire sans remercier mes parents pour leur dévouement incommensurable, qui m'ont toujours soutenue et poussés à donner le meilleur de moi-même.*

*Enfin, je remercie tous mes collègues pour leurs sourires et leurs bonnes humeurs.*

# Plan

<b>Liste des publications</b>	<b>6</b>
<b>Introduction Générale</b>	<b>8</b>
<b>Chapitre 1 : Reconnaissance de l'Écriture Manuscrite</b>	<b>10</b>
1. Introduction	10
2. Notion d'OCR	10
3. Analyse de l'image	11
4. Différentes Approches de Classification	11
4.1. Approche Structurale	12
4.2. Classifieur Bayésien	12
4.3. Méthodes des K Plus Proches Voisins (KPPV)	12
4.4. Méthodes de séparation linéaire et non linéaire	13
4.5. Méthodes Connexionnistes	13
4.6. Méthodes Stochastiques	13
5. Modèles de Markov Cachés	14
6. Réseaux Bayésiens Statiques et Dynamiques	14
7. Conclusion	15
<b>Chapitre 2 : Prétraitement de l'image</b>	<b>16</b>
1. Introduction	16
2. Binarisation	16
2.1. Méthode de Bernsen	17
2.2. Méthode de Niblack	17
2.3. Méthode de Sauvola	17
2.4. Méthode de Wolf	18
2.5. Méthode de Nick	
2.6. Résultats	18
3. Squelettisation	20
3.1. Propriétés du Squelette	21
3.2. Définitions et Notations	22
3.3. Connexité	24
3.3.1. Nombre d'intersections de Rutovitz	24
3.3.2. Nombre d'intersections de Hilditch	24
3.4. Description des Algorithmes	25
3.4.1. Algorithme de Zhang et Suen (ZS)	25
3.4.1.1. Description	25
3.4.1.2. Propriétés	25
3.4.2. Algorithme de Lu et Wang (LW)	26
3.4.2.1. Description	26

3.4.2.2. Propriétés	26
3.4.3. Algorithme de Guo et Hall (GH)	26
3.4.3.1 Description	27
3.1.3.2. Propriétés	27
3.4.4. Algorithme de Rosenfeld (ROS)	27
3.4.4.1. Description	27
3.4.4.2. Propriétés	27
3.4.5. Algorithme de Host, Stewart, Clint et Perrot (HSCP)	27
3.4.5.1. Description	28
3.4.5.2. Propriétés	28
3.4.6. Algorithme de Hall (HSCPN)	28
3.4.6.1. Description	29
3.4.6.2. Propriétés	29
3.4.7. Algorithme de Guo et Hall (GH_AFP1)	29
3.4.7.1. Description	29
3.4.7.2. Propriétés	29
3.4.8. Algorithme de Guo et Hall (GH_AFP2)	30
3.4.8.1. Description	30
3.4.8.2. Propriétés	30
3.4.9. Algorithme de Guo et Hall (GH_AFP3)	30
3.4.9.1. Description	31
3.4.9.2. Propriétés	31
3.4.10. Algorithme de Chin, Wan, Stover et Iverson (CWSI)	31
3.4.10.1. Description	32
3.4.10.2. Propriétés	32
3.4.11. Algorithme de Wu et Tsai (WT)	33
3.4.11.1. Description	33
3.4.11.2. Propriétés	33
3.4.12. Algorithme de Manzarena et Bernard (MB_2)	33
3.4.12.1. Description	34
3.4.12.2. Propriétés	34
3.4.13. Algorithme de Gonzales et Wood (GW)	34
3.4.13.1. Description	34
3.4.13.2. Propriétés	34
3.4.14. Algorithme de Rutovitz (RUT)	35
3.4.14.1. Description	35
3.4.14.2. Propriétés	35
3.4.15. Algorithme de Deutsh (DEU)	35
3.4.15.1. Description	36
3.4.14.2. Propriétés	36
3.4.16. Algorithme FP-Isthmus	36
3.4.16.1. Description	37
3.4.16.2. Propriétés	37
3.4.17. Algorithme SI-Q-Isthmus	37

3.4.17.1. Description	38
3.4.17.2. Propriétés	38
3.4.18. Algorithme SF-k-Isthmus	38
3.4.18.1. Description	39
3.4.18.2. Propriétés	39
3.5. Résultats	39
4. Conclusion	41
<b>Chapitre 3 : Extraction des Caractéristiques</b>	<b>42</b>
1. Introduction	42
2. Objectifs de l'Extraction des Caractéristiques	42
3. Présentation de Quelques Méthodes d'Extraction des Caractéristiques	43
3.1. Moments de Hu	43
3.2. Descripteurs de Fourier Elliptiques	44
3.3. Profils et Contours	45
3.3.1. Les Primitives Perceptuelles	45
3.3.2. Les Primitives Directionnelles	46
3.3.3. Les Primitives Topologiques	46
4. Méthodes d'Extraction des Caractéristiques Utilisées	46
4.1. Méthode de la Somme	46
4.2. Méthode de la Distance	47
5. Conclusion	48
<b>Chapitre 4 : Classification</b>	<b>49</b>
1. Introduction	49
2. Réseaux de Neurones	49
2.1. Neurones Biologiques	50
2.2. Historique	51
2.3. Modèle Formel	52
2.4. Différentes Configurations du Réseau	52
2.5. Apprentissage	54
2.6. Réseaux de Neurones à Apprentissage Supervisé	54
2.6.1. Le Perceptron	54
2.6.2. Architecture à Couches Cachées : Rétro_Propagation	56
3.6.3. Evaluation de la Rétro_propagation	59
2.7. Champs d'Utilisation	60
2.8. Surapprentissage	60
3. K- Plus Proches Voisins	61
3.1. Algorithme 1 – PPV	61
3.2. Algorithme K – PPV	62
3.3. Quelques règles sur le choix de k	62
4. Bayes Naïve	63

4.1. Théorème de Bayes	63
4.2. Modèle bayésien naïf	64
4.3. Estimation de la valeur des paramètres	65
4.4. Classification à partir d'un modèle de probabilité	66
5. Sparse Representation	66
5.1. Principe	66
5.2. Traitement avec un bruit faible	67
5.3. Algorithme	68
6. Conclusion	68
<b>Chapitre 5 : Résultats Expérimentaux</b>	<b>69</b>
1. Introduction	69
2. Prétraitement	69
3. Extraction des Caractéristiques	70
4. Réseaux de Neurones	73
5. K - Plus Proches Voisins	75
6. Bayes Naïve	75
7. Sparse Representation	75
8. Résultats Expérimentaux	76
9. Conclusion	109
<b>Conclusion Générale</b>	<b>110</b>
<b>Annexe : Base de données</b>	<b>111</b>
<b>Références</b>	<b>113</b>
<b>Liste des Figures</b>	<b>120</b>
<b>Liste des Tableaux</b>	<b>124</b>

## Liste des publications

### Publications :

1. **Kamal Moro**, Mohamed Fakir, Badre-eddine El Kessab, Belaid Bouikhalene, Cherki Daoui. Comparison of Two Feature Extraction Methods Based on the Raw Form and his Skeleton for Gujarati Handwritten Digits. Facta Universitatis (NIS) Ser Math Inform Vol 28, No 2 pp: 161-178. 2013.
2. **Kamal Moro**, Mohamed Fakir, Badre-eddine El Kessab, Belaid Bouikhalene, Cherki Daoui. Gujarati Handwritten Numeral Optical Character Through Neural Network end Skeletonization. Jurnal Sistem Komputer - Vol 3 No 1, Juni 2013.
3. **Kamal Moro**, Mohamed Fakir, Badre-eddine El Kessab, Belaid Bouikhalene, Cherki Daoui. Feature Extraction Based on the Raw Form and his Skeleton. Global Journal of Computer Application and Technology. Vol 1(3) pp: 444-448. 2011.
4. Badre-eddine El Kessab, Cherki Daoui, Belaid Bouikhalene, Mohamed Fakir, **Kamal Moro**. Extraction Method of Handwritten Digit Recognition Tested on the MNIST Database. International Journal of Advanced Science and Technology. Vol 50, Junary 2013.
5. Badre-eddine El Kessab, Belaid Bouikhalene, Cherki Daoui, Mohamed Fakir, **Kamal Moro**. Utilisation des Réseaux de Neurones pour la Reconnaissance des Caractères Tifinaghe Manuscrits. Revue Méditerranéenne des Télécommunications vo.2, n°1, janvier 2012
6. Rachid El Yachi, **Kamal Moro**, Mohamed Fakir, Balaid Bouikhalene. On the Recognition of Tifinaghe Scripts. Journal of Theoretical and Applied Information Technology. Vol 20, N 2. 2010.

## Communications

1. **Kamal Moro**, Mohamed Fakir, Belaid Bouikhalene, Badre-Eddine El Kessab. Using the Bayes Naïve Method for recognition of MNIST character. First International Conference on Business Intelligence (CBI'14), April 29-30, 2014, Béni Mellal, Morocco.
2. **Kamal Moro**, Mohamed Fakir, Badre-eddine El Kessab, Belaid Bouikhalene, Cherki Daoui. Evaluation de quelques Techniques de Binarisation de caractères Tifinaghs manuscrits. 3ieme Symposium International de Traitement Automatique de la langue et Culture Amazigh. SITACAM'13, 2-4 Mai 2013, Béni Mellal, Morocco.
3. **Kamal Moro**, Mohamed Fakir, Badre-eddine El Kessab, Belaid Bouikhalene, Cherki Daoui. Contribution à la reconnaissance des caractères Tifinagh par utilisation des réseaux de neurones et la squelettisation. 2ieme Symposium International sur le Traitement Automatique de la Culture Amazighe. SITACAM 2011. 6-7 Mai 2011, Agadir, Morocco.
4. **Kamal Moro**, Mohamed Fakir, Badre-eddine El Kessab, Belaid Bouikhalene, Cherki Daoui. Skeletonization Methods Evaluation for Recognition of Printed Tifinaghe Characters. 1ier Symposium International sur le Traitement Automatique de la Culture Tifinaghe, défis et opportunités. SITACAM'09. 12-13 Décembre 2009, Agadir, Morocco.
5. **Kamal Moro**, Mohamed Fakir, Badre-eddine El Kessab, Belaid Bouikhalene, Cherki Daoui. Utilisation des Réseaux de Neurones par l'Utilisation des Chiffres Manuscrits. 1iere Journée Nationale de Mathématiques et Applications en Informatique et Télécommunications. 1ere JMIT'11. Juillet 2011, Béni Mellal, Morocco.
6. **Kamal Moro**, Badre-eddine El Kessab, Mohamed Fakir, Belaid Bouikhalene, Cherki Daoui. Evaluation de la Squelettisation pour la Reconnaissance des Caractères MNIST. 1iere Journée Nationale de Mathématiques et Applications en Informatique et Télécommunications. 1ere JMIT'11. Juillet 2011, Béni Mellal, Morocco.

## Introduction Générale

L'apparition des premières écritures, il y a près de 5500 ans, a révolutionné à ce point l'humanité, que celle-ci décida ultérieurement que son Histoire débiterait à ce moment. L'importance de l'écriture en tant que moyen de communication pour l'homme n'a fait que croître. Aujourd'hui, le développement constant de l'outil informatique rend nécessaire une communication toujours plus étroite entre l'homme et la machine. C'est à celle-ci de s'adapter aux moyens de communication que l'homme maîtrise à présent et qu'il se complaît à utiliser, telle que l'écriture.

La reconnaissance de l'écriture manuscrite est le vieux rêve de tous ceux qui ont eu besoin d'entrer des données dans un ordinateur. Il remonte à plus d'une trentaine d'années. Aujourd'hui, il existe plusieurs domaines dans lesquels la reconnaissance de l'écriture manuscrite est attendue avec impatience, par exemple dans le tri automatique du courrier, le traitement automatique de dossiers administratifs, des formulaires d'enquêtes, ou encore l'enregistrement des chèques bancaires. Ces applications montrent clairement les spécificités du domaine de la reconnaissance de l'écriture manuscrite par rapport à celui de la reconnaissance optique des caractères (OCR : Optical Character Recognition) qui concerne les caractères imprimés ou dactylographiés. Il est nécessaire de distinguer également la reconnaissance en ligne (on-line) de l'écriture manuscrite, qui relève plutôt de l'interfaçage entre l'homme et l'ordinateur (un stylo spécial est connecté à la machine et ne fonctionne que sur une tablette sensible), de la reconnaissance hors ligne (off-line). Seule la reconnaissance hors ligne sera considérée dans ce travail.

Dans la reconnaissance de l'écriture en général, nous distinguons deux grandes catégories d'applications : applications à vocabulaire limité, où généralement le nombre de mots à reconnaître constitue un lexique de taille inférieure à 100 mots ; et les applications à vocabulaire étendu, où plusieurs milliers, voire des dizaines de milliers de mots forment un dictionnaire d'appui.

Quelques systèmes de reconnaissance de l'écriture manuscrite ont été réalisés et sont opérationnels à ce jour. Cependant, ils sont spécifiques à un domaine précis et sont encore limités. Par exemple, en ce qui concerne la poste, la reconnaissance de l'écriture manuscrite, contrairement à celle des caractères imprimés, se limite au code postal en chiffres ainsi qu'à la ville en caractères majuscules.

La reconnaissance des caractères peut être considérée, à première vue, comme étant typiquement un problème de reconnaissance des formes. Elle s'articule autour des principes habituels : l'acquisition de l'information par un organe sensoriel (capteur), le prétraitement, l'extraction des caractéristiques et la classification de l'objet à partir de sa représentation codée parmi un ensemble de catégories ou classes possibles.

L'étape de prétraitement est l'une des premières tâches d'un système de reconnaissance après l'étape d'acquisition. Cette étape n'est pas spécifique à la reconnaissance du manuscrit mais fait partie de tout système de reconnaissance de forme visant à améliorer la qualité de l'information pour les étapes de traitements qui vont suivre. Dans le cadre du manuscrit, elle regroupe l'ensemble des processus visant au bon conditionnement du message écrit et qui sont indispensables à son identification.

Parmi les phases les plus importantes du prétraitement, on trouve la squelettisation, c'est le nom apporté à la méthode qui consiste à extraire le squelette d'une forme binaire, néanmoins, elle n'est pas indispensable. Plusieurs travaux de recherche ont utilisés la squelettisation dans le processus de reconnaissance des caractères [1,2,3,4], d'autres ont choisis d'effectuer le traitement sans passer par la squelettisation [5,6,7,8]. Notre approche consiste à effectuer le traitement sur la forme brute et sur le squelette du caractère, la moyenne des deux vecteurs d'extraction est celle utilisée dans la phase de classification. Les résultats expérimentaux montrent l'avantage de l'utilisation de cette méthode par rapport à d'autres approches classiques, ils ont été appliqués sur la base MNIST et les caractères Gujarati manuscrits. La Figure.1 montre le processus de reconnaissance utilisé.

La thèse est organisée de la manière suivante : dans le chapitre 1, nous présentons l'état de l'art de la reconnaissance de l'écriture manuscrite. Quelques méthodes de binarisation utilisant l'approche locale, et une description de la squelettisation et de 21 algorithmes sont présentées au chapitre 2. Dans le chapitre 3, nous présentons les méthodes d'extraction des caractéristiques utilisées dans cette thèse. Les réseaux de neurones, les K-PPV, la méthode de bayes naïve et Sparse Representation sont détaillés au chapitre 4, ce sont les méthodes de classification utilisées. Dans le chapitre 5, les résultats expérimentaux sont présentés et détaillés avec une comparaison avec d'autres approches classiques.

## Chapitre 1

# Reconnaissance de l'Écriture Manuscrite

### 1. Introduction

Chaque information écrite peut être reprise dans une chaîne de traitement informatisée avec différents objectifs (gestion électronique de documents rédaction, édition de rapports, archivage, tri...). La reconnaissance optique de caractères, en anglais, Optical Character Recognition (OCR) est une opération permettant d'effectuer la transformation d'un texte sous forme image en un texte sous forme d'un fichier informatique de façon rapide et automatique. Les systèmes de reconnaissance de caractères sont différents suivant que l'on traite isolément ou non. Dans le premier cas, il s'agit de la reconnaissance de la forme des caractères sans prise en compte du contexte. Dans le second, la reconnaissance prend en considération le contexte lexical ou celui de la structure du texte.

### 2. Notion d'OCR

La reconnaissance optique de caractères est un procédé permettant de récupérer les symboles dans les images de texte numérisées. Les images (par exemple d'une télécopie) consistent en des matrices de pixels. La tâche d'un OCR consiste à segmenter les images en lignes, mots, caractères puis à effectuer la reconnaissance des symboles. Il en résulte une transcription textuelle de l'image par laquelle des traitements automatiques sont possibles : recherche de mots, de noms, résumé,...etc. En général, l'OCR concerne le traitement d'un document numérisé. L'OCR connaît des applications pratiques dans plusieurs domaines :

- Les banques pour l'authentification des chèques, et les assurances pour la vérification de clauses de contrats.
- La poste pour la lecture des adresses et le tri automatique du courrier.
- Les télécommunications pour l'échange à distance de fichiers informatisés.

Jusqu'à maintenant, il n'existe pas de système universel d'OCR mais des voies d'approches qui dépendent du type des données traitées et de l'application envisagée. Pour la reconnaissance, on a toujours le problème concernant le style calligraphique (fonte) : monofonte, multifonte ou omnifonte. Un système est monofonte s'il ne connaît que l'alphabet d'une seule fonte. La reconnaissance est simple puisque l'alphabet représenté est réduit. Un système multifonte est capable de reconnaître plusieurs fontes parmi un ensemble de fontes préalablement apprises. Dans ce cas la reconnaissance est difficile puisque le système doit identifier les différentes formes, et gérer plus d'ambiguïtés. Un système est dit omnifonte s'il est capable de reconnaître toutes les fontes sans les avoir apprises, ce qui est trop difficile à réaliser.

### 3. Analyse de l'image

Le but de l'analyse est d'extraire les propriétés caractéristiques de l'image du caractère et de l'exprimer sous forme numérique ou symbolique. Ces caractéristiques doivent être invariantes aux déformations possibles intra-classe. Certains prétraitements peuvent être nécessaires. Parmi ces prétraitements, on trouve :

- Correction de l'inclinaison des lignes de texte consiste à redresser horizontalement les lignes d'écriture oblique. Plusieurs méthodes sont disponibles. Les deux plus populaires sont la transformée de Hough (appliquée sur les centres de gravité des composantes connexes), et les histogrammes de projection qui consistent à parcourir l'image selon des directions  $d$  proches de l'horizontale, et à compter le nombre de pixels noirs selon ces directions pour chaque ligne.
- La squelettisation sert à obtenir une épaisseur égale à 1 du trait d'écriture et à se ramener ainsi à une écriture linéaire. Le squelette doit préserver la forme, la connexité, la topologie, les extrémités du tracé, et ne doit pas introduire d'éléments parasites.
- La normalisation permet de ramener les images de caractères à des tailles standards. Cette phase peut être indispensable pour certains systèmes comme les réseaux neuronaux.

L'étape principale d'un système d'OCR est d'extraire de l'image de caractère des caractéristiques sur lesquelles la reconnaissance est fondée. Les caractéristiques utilisées peuvent être différentes suivant la technique utilisée pour la phase de reconnaissance. Globalement, les caractéristiques se divisent en trois types : locales, globales et structurelles [9].

- Les caractéristiques globales cherchent à représenter la forme générale d'un caractère et elles sont calculées sur l'image complète. Ces caractéristiques sont par exemple : les histogrammes de projections, les densités de points, les intersections avec les droites.
- Les caractéristiques locales sont calculées au niveau d'un pixel et explorent la partie d'image autour de ce pixel. Elles peuvent aussi avoir à explorer des parties éloignées du pixel courant.
- Les caractéristiques structurelles reflètent la structure du caractère (lettre ou chiffre) : présence de boucles fermées ou ouvertes, présence de jonctions caractéristiques entre segments.

### 4. Différentes Approches de Classification

La classification est une tâche difficile à réaliser selon l'application demandée. C'est pour cela que les chercheurs ont développé plusieurs approches. On distingue l'approche statistique et l'approche structurelle. Une différence essentielle réside dans la représentation de la forme : vecteur de caractéristiques dans l'approche statistique, agencement de primitives pour l'approche structurelle. Nous allons rappeler quelques une de ces approches couramment utilisées en les classant en 4 groupes : structurelle, statistique, neuronale et stochastique. Cette

classification comprend une part d'arbitraire car celle-ci n'est pas unique. Nous nous limitons ici à la présentation de méthodes à apprentissage supervisé.

#### 4.1.Approche Structurelle

Les méthodes structurelles se basent sur la structure physique des caractères. Elles cherchent à décomposer le caractère en primitives et à décrire leurs relations. Les primitives sont de type topologique comme un arc, une boucle, etc, et une relation peut être la position relative d'une primitive par rapport à une autre. Généralement, le nombre de primitives est assez limité et leur enchaînement peut se décrire par un ensemble de règles d'assemblages. Dans cette approche on distingue plusieurs méthodes :

- **Méthodes Syntaxiques :** Ces méthodes sont directement issues de la théorie des langages formels. Elles se basent sur une grammaire formelle. Chaque caractère est représenté par une phrase dans un langage où le vocabulaire est constitué de primitives. La reconnaissance consiste à déterminer si la phrase de la description du caractère peut être générée par la grammaire. Cette méthode a été utilisée dans [10] et [11]. L'inconvénient de ces méthodes est l'absence d'algorithmes efficaces pour l'inférence grammaticale directe.
- **Comparaison des Graphes :** Cette méthode consiste à construire un graphe où les nœuds contiennent les primitives et les liens entre ces primitives. Ainsi la reconnaissance consiste à faire une mise en correspondance entre ce graphe et d'autres graphes représentant des caractères de référence construits lors de la phase d'apprentissage. Cette méthode a été utilisée dans [12], [13] et [14] et elle donne des résultats acceptables.
- **Comparaison de Chaines :** Dans ce cas, les caractères sont représentés par des chaînes de primitives. La méthode consiste à mesurer la similitude entre les chaînes du caractère à reconnaître et un modèle de référence par calcul de distance [15].

#### 4.2.Classifieur Bayésien

La méthode statistique bayésienne à la reconnaissance de forme suppose que le problème de représentation des caractères admet un modèle probabiliste. Cette méthode définit l'appartenance d'un caractère à une classe avec un minimum d'erreur et évalue le risque de la décision à prendre pour un caractère donné. Nous cherchons donc la classe qui maximise la probabilité d'appartenance parmi l'ensemble des classes. L'intérêt de cette méthode est qu'elle repose sur des bases mathématiques très bien définies.

#### 4.3.Méthodes des k plus proches voisins (KPPV)

Ces méthodes consistent, étant donné un point  $x$  de  $R^m$  représentant le caractère à reconnaître, à déterminer la classe de chacun des  $k$  points les plus proches de  $x$  parmi l'ensemble des caractères d'apprentissage et à retenir pour la décision la classe la plus représentée [16]. Si  $k=1$ ,  $x$  est donc attribué à la classe de son plus proche voisin. De plus, ces méthodes s'inscrivent dans le cadre des méthodes non bayésiennes et non paramétriques. Les méthodes des KPPV sont lentes en phase de décision puisqu'il faut calculer un nombre très

important de distances. Nous remarquons que nous pouvons utiliser plusieurs types de distances parmi lesquelles la distance tangente qui utilise une approximation de Taylor du premier ordre [17]. Pour cette distance, il existe deux formes : la distance tangente unilatérale et la distance tangente bilatérale.

#### **4.4.Méthodes de séparation linéaire et non linéaire**

Ces méthodes statistiques sont basées sur la définition des fonctions permettant de séparer partiellement ou totalement des classes représentées par les vecteurs paramètres de leurs échantillons [18]. Ainsi on peut distinguer deux catégories de fonctions :

- Fonctions de discrimination linéaire [19] : dans ce cas, on parle d'un hyperplan séparateur. Donc le problème consiste en la recherche du meilleur hyperplan permettant de discriminer les classes. Dans le cas de non séparabilité des classes, on passe dans un espace de plus grande dimension où l'on cherche un séparateur linéaire : ce sont les SVM (Support Vector Machines) [20].
- Fonctions de discrimination non linéaire : fonctions linéaire par morceaux, séparation par des quadriques [15].

#### **4.5.Méthodes Connexionnistes**

Le principe de base du connexionnisme est que les phénomènes mentaux peuvent être décrits à l'aide de réseaux d'unités simples interconnectées. La forme des connexions et des unités peut varier selon les modèles. Par exemple, les unités d'un réseau peuvent représenter des neurones et les connexions peuvent représenter des synapses. Un autre type de modèle pourrait faire que chaque unité du réseau soit un mot et que chaque connexion soit un indicateur de la similarité sémantique.

Les réseaux de neurones sont, de loin, les modèles connexionnistes les plus répandus aujourd'hui. Beaucoup de recherches utilisant des réseaux de neurones se réclament du connexionnisme. Les réseaux connexionnistes répondent à deux caractéristiques :

- Chaque état mental peut être représenté comme un vecteur à  $n$  dimensions représentant les valeurs d'activation des unités neuronales.
- Le réseau peut apprendre en modifiant le poids des connexions entre ses unités. La force des connexions (ou poids) est généralement représentée comme une matrice à  $n$  dimensions.

#### **4.6.Méthodes Stochastiques**

L'approche stochastique consiste à utiliser la modélisation par une collection de variables aléatoires en prenant en compte la variabilité des caractères. Ainsi le caractère est considéré comme un signal continu observable dans le temps à différents endroits constituant les états d'observations. La reconnaissance consiste à comparer un échantillon de caractère de test à un modèle donné. Les paramètres des modèles sont calculés à l'aide des probabilités calculées de manière plus fine par apprentissage. Chaque modèle décrit ces états à l'aide des probabilités de transition d'état à état et de la loi de probabilité d'observation par état. Ainsi la comparaison consiste à chercher dans ce graphe d'états le chemin le plus probable

correspondant à une suite d'éléments observés dans la chaîne d'entrée. Ces méthodes sont robustes et fiables du fait de l'existence de bons algorithmes d'apprentissage.

#### **4.6.1. Modèles de Markov Cachés**

Un modèle de Markov Caché (HMM) [21] est défini par les données suivantes :

- Un ensemble d'états
- Un ensemble de symboles observables dans chaque état
- Une matrice des probabilités de transitions
- Une matrice des probabilités d'observation
- Un ensemble de densités de probabilité initiale

Ce modèle utilise essentiellement des données unidimensionnelles, ce qui a permis leur application directe en traitement de la parole, et ensuite pour l'OCR. On peut distinguer deux types de HMMs :

- Les HMMs discrets qui modélisent les observations avec des variables discrètes [22,23].
- Les HMMs semi-continus qui modélisent les observations avec un mélange de gaussiennes [24].

L'inconvénient est que les HMMs sont mono-dimensionnels alors que les images de caractères sont en dimension 2, ce qui donne des résultats parfois peu satisfaisants pour une application directe des HMMs.

#### **4.6.2. Réseaux Bayésiens Statiques et Dynamiques**

Un réseau bayésien statique (ou BN) [25] est un mariage entre la théorie des graphes et la théorie des probabilités. Ainsi un réseau bayésien est constitué d'un graphe acyclique orienté dont les nœuds sont des variables aléatoires qui peuvent avoir un nombre discret d'états possibles ou dont les valeurs sont continues selon une loi continue, et d'un ensemble de distributions locales de probabilité qui sont les paramètres du réseau. Pour chaque nœud on dispose d'une table de probabilités conditionnelles.

Récemment les réseaux bayésiens sont appliqués dans le domaine de l'écrit. Nous pouvons citer les travaux de Chow [26] dans lesquels il modélise les caractères on-line par les réseaux bayésiens : les variables qui représentent les positions des points caractéristiques, sont toutes observées. Souafi dans sa thèse [27] utilise un réseau bayésien classifieur pour la classification automatique de documents (apprentissage des structures des BNs en utilisant la méthode de programmation génétique). Enfin les réseaux bayésiens ont été utilisés pour la vérification automatique des signaux [28] (modélisation des dépendances entre composantes de la signature).

Les réseaux bayésiens dynamiques (DBN) sont une extension des réseaux bayésiens statistiques et sont également appliqués à la reconnaissance automatique de la parole [29].

Daoudi dans [30] propose une approche multibandes où le réseau bayésien permet une interaction entre sous-bandes du signal de parole.

## **5. Conclusion**

Dans ce chapitre, on a introduit des définitions et des notions d'OCR. Ensuite, on a présenté la façon dont l'image est analysée dans tout système de reconnaissance des caractères, après, on a expliqué les différentes approches de classification, à savoir, l'approche structurelle, les classifieurs bayésiens, les méthodes des K Plus Proches Voisins, les méthodes de séparation linéaire, connexionnistes et stochastiques. Dans les chapitres qui suivent, on présentera les différentes démarches utilisées dans le processus de reconnaissance.

## Chapitre 2

# Prétraitement de l'image

### 1. Introduction

Le prétraitement est une opération ou un ensemble d'opérations consistant à rendre des données brutes aptes à une analyse thématique. Cette phase comprend des processus de traitement d'image en général (atténuation du bruit, binarisation ...) et d'autres plus spécifiques au traitement de l'écriture manuscrite. Traditionnellement, les prétraitements contribuent à diminuer la variabilité intra-classes par des traitements de normalisation.

Dans ce qui suit, on décrit les notions de binarisation et de squelettisation qui se considèrent comme les fonctions les plus importantes dans l'opération de prétraitement, et on présente quelques algorithmes et ces résultats appliqués sur des caractères.

### 2. Binarisation

La binarisation est souvent la première étape dans les systèmes de traitement et d'analyse d'images [31,32], et plus particulièrement d'images de documents. Elle a comme but de diminuer la quantité d'informations présentes dans l'image, et de ne garder que les informations pertinentes, ce qui nous permet d'utiliser des méthodes d'analyse simple vis-à-vis des images en niveaux de gris ou en couleurs [33]. Les performances des étapes suivantes dans les systèmes d'analyse de documents, dépendent fortement des résultats de l'algorithme de binarisation utilisé, c'est pour cette raison que l'étape de binarisation doit être effectuée le plus fidèlement possible. Elle doit, d'une part, conserver le maximum d'informations et détails présents dans l'image en entrée, et de l'autre part, elle doit éliminer le bruit présent dans l'image. Le problème de binarisation de documents est plus ancien que l'analyse d'images de documents lui-même. Un très grand nombre de techniques de binarisation ont été proposées dans la littérature dont chacune d'entre elles est appropriée à un type d'images. En effet, plusieurs techniques ont été proposées ces deux dernières décennies pour la binarisation d'images de documents en niveaux de gris, d'autres sont applicables directement sur des images en couleurs.

Selon plusieurs travaux de recherche [34,35], les techniques de binarisation d'images en niveaux de gris peuvent être classées en deux catégories : seuillage globale, où un seul seuil est utilisé dans toute l'image pour la diviser en deux classes (texte et fond), et seuillage local où les valeurs des seuils sont déterminées localement, pixel par pixel ou bien région par région.

Dans ce chapitre, on présente un état de l'art de quelques méthodes locales de binarisation les plus fréquemment citées dans la littérature, à savoir la méthode de Bernsen [36], Niblack [37], Sauvola [38], Wolf [39] et Nick [40].

### 2.1. Méthode de Bernsen

C'est une méthode locale adaptative dont le seuil est calculé pour chaque pixel de l'image [36]. Ainsi pour chaque pixel de coordonnées  $(x, y)$ , le seuil est donné par :

$$T(x, y) = \frac{Z_{haut} + Z_{bas}}{2} \quad (1)$$

, Tel que  $Z_{bas}$  et  $Z_{haut}$  sont le niveau de gris le plus bas et le plus haut respectivement, dans une fenêtre carré  $r \times r$  centré sur le pixel  $(x, y)$ . Cependant si la mesure de contraste  $C(x, y) = (Z_{haut} - Z_{bas})$  est inférieure à un certain seuil  $l$ , alors le voisinage consiste en une seule classe: fond ou bien texte.

Sur les fenêtres de tailles petites, cette méthode garde la forme du caractère mais génère des points sur le voisinage du contour, mais lorsqu'on applique cet algorithme sur une fenêtre de taille  $9 \times 9$ , les résultats sont assez satisfaisants.

### 2.2. Méthode de Niblack

L'algorithme de Niblack [37] calcule un seuil local à chaque pixel en glissant une fenêtre rectangulaire sur toute l'image. Le seuil  $T$  est calculé en utilisant la moyenne  $m$  et l'écart-type de tous les pixels dans la fenêtre (voisinage du pixel en question). Ainsi le seuil  $T$  est donné par :

$$T = m + k * \sigma \quad (2)$$

, tel que  $k$  est un paramètre utilisé pour déterminer le nombre de pixels de contours considérés comme des pixels de l'objet, et prend des valeurs négatives ( $k$  est fixé - 0.2 par les auteurs).

La méthode de Niblack est efficace dans le cas des photos réels, cependant, dans le cas des caractères, les résultats sont moins satisfaisants.

### 2.3. Méthode de Sauvola

L'algorithme de Sauvola [38] est une modification de celui de Niblack, pour donner plus de performance dans les documents avec un fond contient de texture claire ou bien trop de variation et illumination inégal.

Dans la modification de Sauvola, la binarisation est donnée par :

$$T = m * \left( 1 - k * \left( 1 - \frac{\sigma}{R} \right) \right) \quad (3)$$

Où  $R$  est la portée dynamique de l'écart-type, et le paramètre  $k$  prend des valeurs positives dans l'intervalle  $[0.2, 0.5]$ . Dans ses tests, Sauvola utilise  $R=128$  et  $k=0.5$ .

La méthode de Sauvola est plus efficace que la méthode de Niblack dans le cas où le niveau de gris du texte est proche de 0 et celui du fond est proche de 255. Cependant, dans le cas où le niveau de gris des pixels du fond et du texte sont proches, les résultats sont moins satisfaisants.

#### 2.4. Méthode de Wolf

Pour s'adresser aux problèmes de l'algorithme de Sauvola (faible contraste, écart de niveaux de gris etc.), Wolf et al. [39] proposent de normaliser le contraste et la moyenne de niveaux de gris de l'image, et calculer le seuil par :

$$T = (1 - k) * m + k * M + k * \frac{\sigma}{R * (m - M)} \quad (4)$$

Tel que  $k$  est fixé à 0.5,  $M$  est le niveau de gris minimum de l'image et  $R$  vaut l'écart-type maximum de niveaux de gris obtenu sur toutes les fenêtres.

La méthode de Wolf donne des résultats satisfaisants, que ce soit sur les photos réels ou bien sur les caractères.

#### 2.5. Méthode Nick

L'avantage majeur de cette méthode est qu'elle améliore considérablement la binarisation des images de page « blanches » et claires, et dans le cas où l'image présente de faible contraste, en déplaçant vers le bas, le seuil de binarisation [35].

Le calcul du seuil est réalisé comme suit :

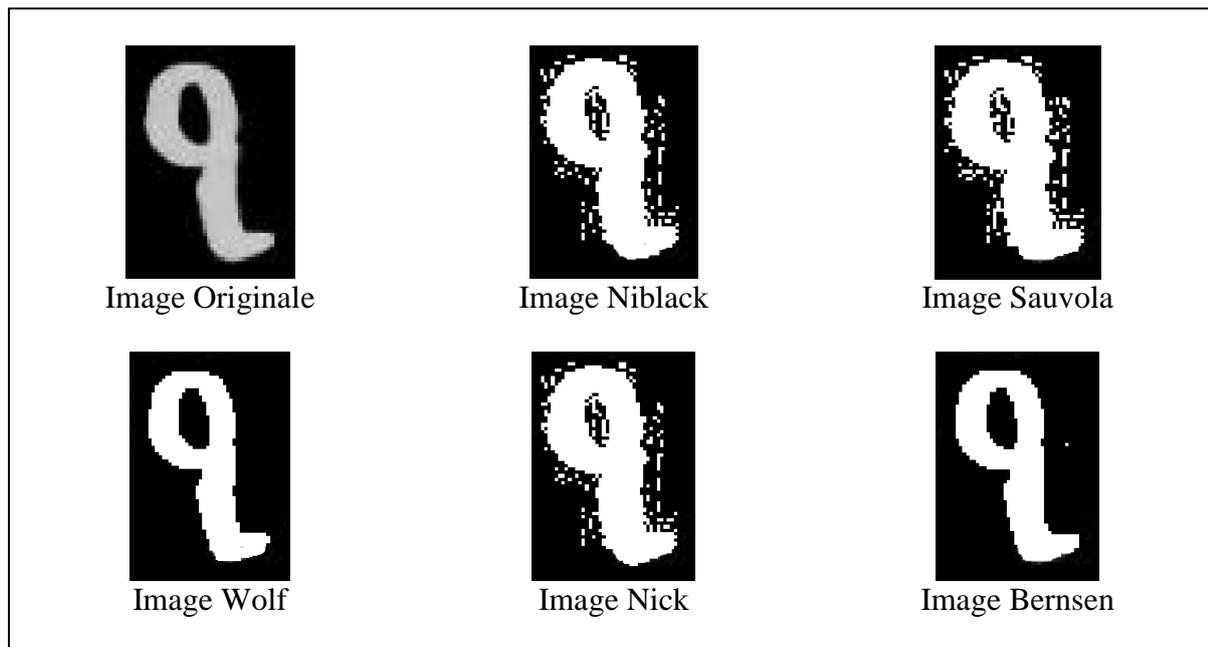
$$T = m + \sqrt{\frac{(\sum p_i^2 + m^2)}{NP}} \quad (5)$$

avec  $k$  est le facteur de Niblack varie entre -0.1 à -0.2 selon les besoins de l'application,  $m$ : le niveau de gris moyen,  $p_i$ : niveau de gris du pixel  $i$  et  $NP$  est le nombre total de pixels.

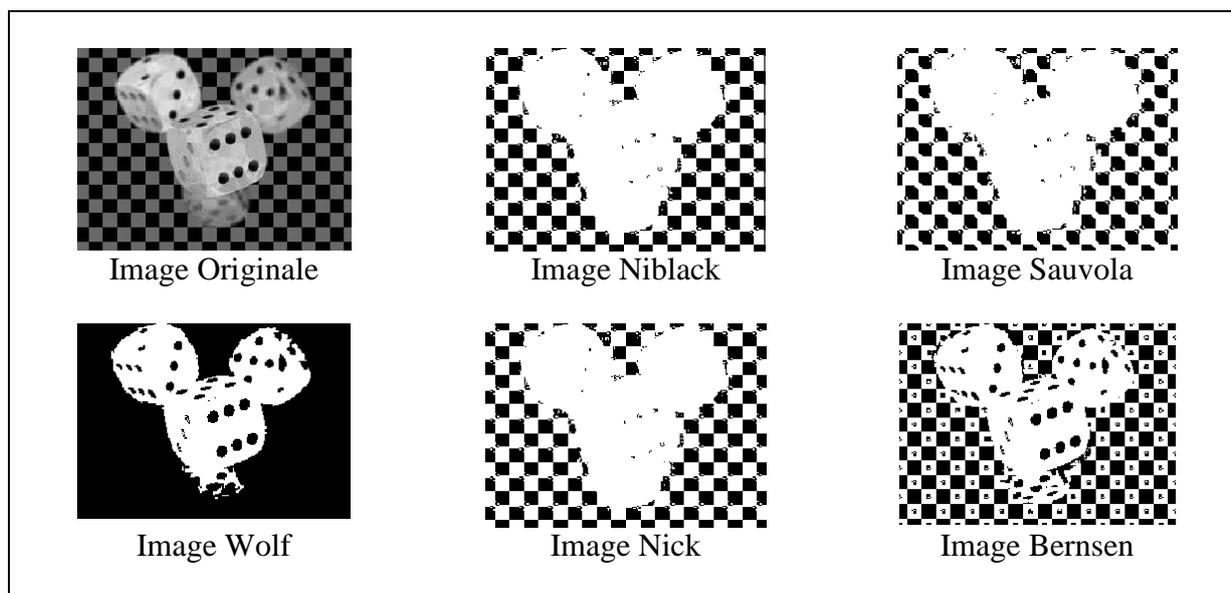
De même que la méthode de Niblack ou Sauvola, la méthode de Nick ne donne pas les résultats attendus sur les caractères.

#### 2.6. Résultats

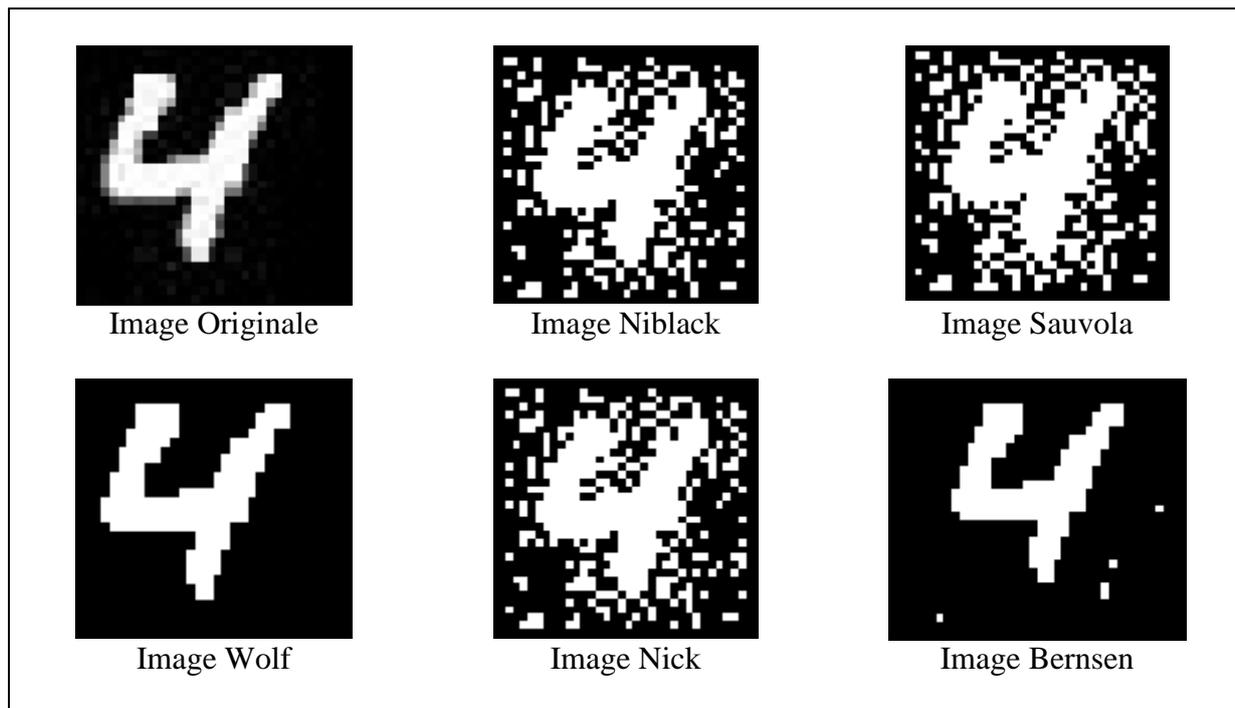
Dans ce paragraphe, nous allons mettre en application les différentes méthodes de binarisation décrites dans ce mémoire, les résultats expérimentaux ont été appliqués sur le caractère Gujarati '1', le caractère de la base MNIST '4' et sur une photo réelle (Figure.2.1, Figure.2.2, Figure.2.3).



**Figure.2.1.** Comparaison des algorithmes de binarisation avec une fenêtre  $r=9*9$  sur le caractère Gujarati



**Figure.2.2.** Comparaison des algorithmes de binarisation avec une fenêtre  $r=9*9$  sur une photo réelle

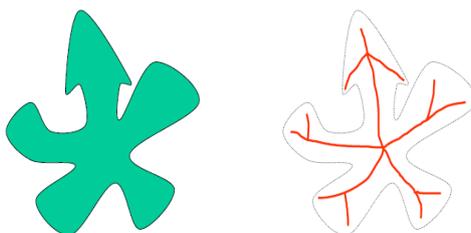


**Figure.2.3.** Comparaison des algorithmes de binarisation avec une fenêtre  $r=9*9$  sur un caractère MNIST

### 3. Squelettisation

Récemment, d'importants efforts ont été dirigés sur les algorithmes de squelettisation en traitement d'image et reconnaissance des formes. Un certain nombre d'algorithmes ont été proposés pour améliorer l'efficacité de calcul en tentant au même temps, de produire une représentation de l'axe médiane d'une région donnée. Typiquement, ceux-ci sont donc les algorithmes de squelettisation qui suppriment, par des procédés itératifs, les pixels de bord d'une région.

Dans le plan continu, le squelette d'une forme est un ensemble de lignes passant en son milieu (Figure.2.4). L'objectif de la squelettisation est de représenter un ensemble avec un minimum d'information, sous une forme qui soit la plus simple à extraire et commode à manipuler. C'est la notion d'axe médian d'une forme continu introduite par BLUM [40] en 1964 « Soit une prairie couverte par l'herbe sèche et  $\Omega$  un ensemble de point de cette prairie. Si tous les points du contour de  $\Omega$  sont enflammés simultanément, le feu se propage de manière homogène et s'étend à une vitesse constante. Le squelette de  $\Omega$  (noté  $MA(\Omega)$ ) est alors défini comme le lieu des points où les fronts enflammés se rencontrent. »



**Figure.2.4.** exemple de squelette

Il existe actuellement une grande variété de méthodes permettant de construire des squelettes à partir de formes. L'une des plus connues est l'amincissement topologique, celui-ci consiste à examiner les pixels de l'image binaire et effacer itérativement ceux qui n'appartiennent pas au squelette final. Les examens des pixels se fait de deux manières: Approche séquentielle où la suppression d'un point  $P$  à la  $n^{ieme}$  itération dépend des opérations faites jusqu'ici mais aussi sur les pixels déjà traités, et approche parallèle où Les pixels sont examinés de façon indépendante à chaque itération, la suppression de  $P$  à la  $n^{ieme}$  itération dépend des opérations réalisées à l'itération précédente.

### 3.1. Propriétés du squelette

- **Préservation de la géométrie:** Le squelette doit rendre compte des propriétés géométriques de la forme (Figure.2.5).



Figure.2.5. Préservation de la géométrie

- **Epaisseur nulle:** Le squelette doit être constitué de courbes sans épaisseur (Figure.2.6).

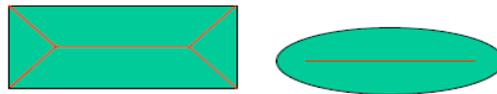


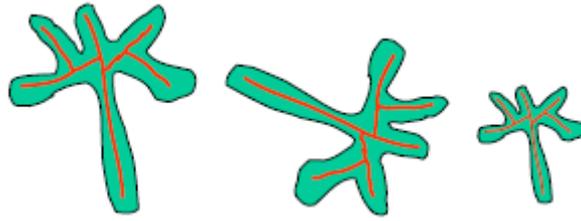
Figure.2.6. Epaisseur nulle

- **Préservation de la topologie:** le squelette doit conserver les relations de connexités: même nombres de composantes connexes, même nombres de trous par composantes (Figure.2.7).



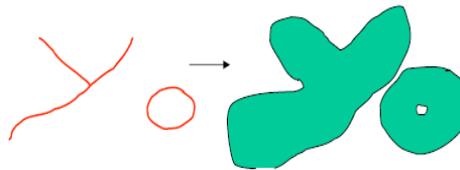
Figure.2.7. Préservation de la topologie

- **Invariance aux transformations affines:** Le squelette doit commuter avec la translation, la rotation et l'homothétie (Figure.2.8).



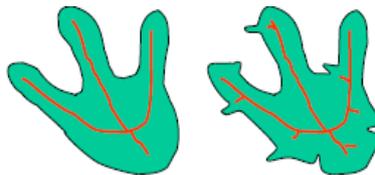
**Figure.2.8.** Invariance aux translations affines

- **Réversibilité:** Le squelette doit permettre de retrouver la forme originale (Figure.2.9).



**Figure.2.9.** Réversibilité

- **Continuité:** Une petite modification de la forme originale doit induire une petite modification du squelette (Figure.2.10).



**Figure.2.10.** Continuité

### 3.2. Définitions et notations

Soit un point  $P$  (Figure.2.11) et ses voisinages noté  $X1, X2, X3, X4, X5, X6, X7$  et  $X8$ .

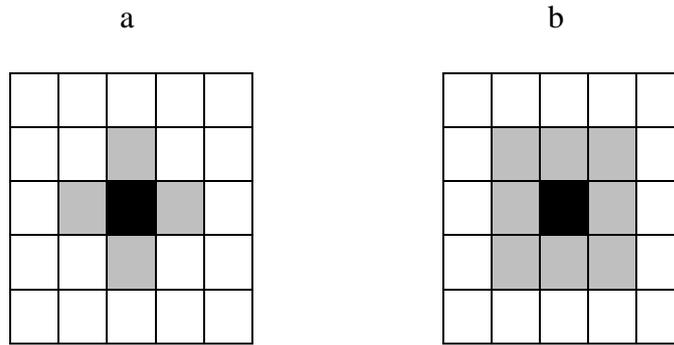
$X4$	$X3$	$X2$
$X5$	$P$	$X1$
$X6$	$X7$	$X8$

**Figure.2.11.** Un point  $P$  et ses voisinages

Nous noterons  $N(P)$  le voisinage d'un point  $P$  et  $b(P)$  le nombre des pixels noirs dans  $N(P)$ .

**Point de contour :** c'est un point qui a au moins un pixel blanc sur son 4-voisinage (Figure.2.12).

**Point de fin :** c'est un point tel que  $b(P) = 1$ .

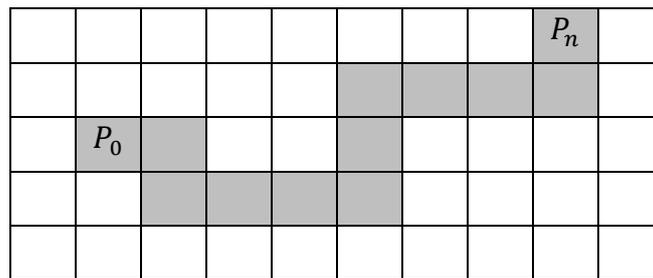


**Figure.2.12.** Principe des 4(-8) voisinages. a : Adjacence directe, b : adjacence indirecte ou diagonale

**k-chemin:** un k-chemin du pixel P au pixel q est une suite de  $x_0 \dots x_n$  (où n est un entier naturel) de pixels tels que  $x_0 = p$  et  $x_n = q$  et pour  $i=0, \dots, n-1$ ,  $x_i$  est k-adjacent à  $x_{i+1}$ .

**k-distance:** La k-distance  $d_k(p, q)$  entre 2 pixels p et q est la longueur minimum d'un k-chemin de p à q.

On appelle **chemin connexe** entre 2 points  $P_0$  et  $P_n$ , une suite de points  $P_0 \dots P_n$  tel que  $P_i$  et  $P_{i+1}$  sont adjacents. La Figure.2.13 montre un chemin 4-connexe.

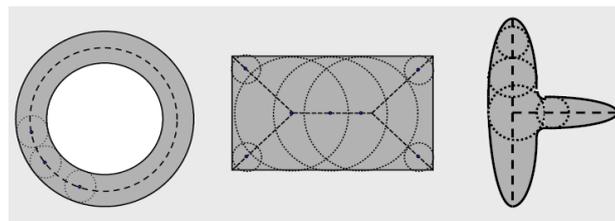


**Figure.2.13.** Chemin 4-connexe

**Boule maximale:** une boule B incluse dans un objet S est dite **maximale** s'il n'existe pas d'autre boule B' incluse dans S et la contenant strictement (Figure.2.14).

**Axe médian:** le squelette ou l'axe médian de S est l'ensemble des centres des boules maximales de S.

**Axe médian pondéré:** l'axe médian pondéré de S est l'ensemble des couples {centre, rayon} des boules maximales de S.

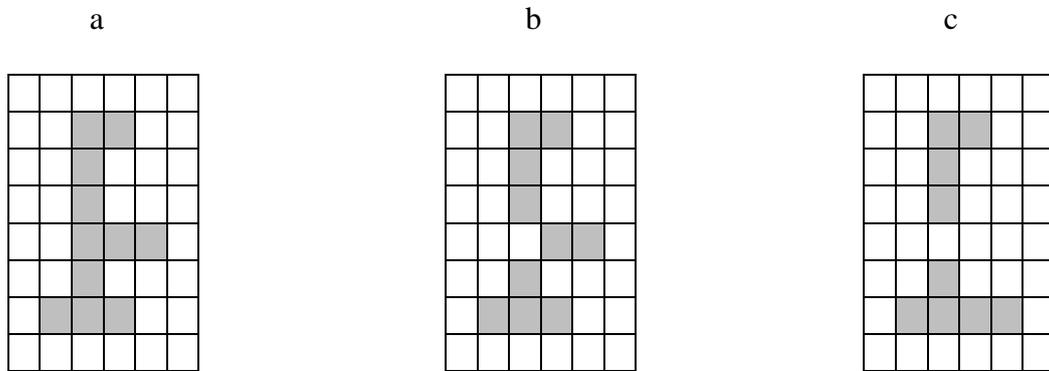


**Figure.2.14.** Boules maximales

### 3.3. Connexité

Soient P et Q deux points de l'image, et soit S un sous-ensemble de pixels contenant P et Q:

- On dit que P et Q sont **connectés** si et seulement s'il existe un chemin connexe inclus dans S entre P et Q.
- S est une **composante connexe** si et seulement si  $(\forall P, Q \in S)$  P et Q sont connectés (Figure 2.15).
- **Point 8-(-4) supprimable** : Un point P non nul est dit 8-(-4) supprimable si et seulement si sa mise à valeur nulle ne change pas la 8-(-4) connectivité de l'objet d'intérêt auquel il appartient.



**Figure.2.15.** Points 8-(-4) supprimables. a : Exemple de composantes 4-connexes, b : Exemple de composantes 8-connexes, c : Exemple de composantes non-connexes.

Un point 4-supprimable est 8-supprimable mais l'inverse n'est pas vrai.

L'amincissement d'une forme consiste à supprimer les points de contour qui ne sont pas des points de fin et qui sont 8-(-4) supprimables (suivant le choix de la connectivité à conserver).

#### 6.3.1. Nombre d'intersections de Rutovitz [41]

$$X_R(P) = \sum_{i=1}^8 |x_{i+1} - x_i| \text{ avec } x_9 = x_1 \quad (6)$$

$X_R(P)$  représente le nombre de transitions  $0 \rightarrow 1$  et  $1 \rightarrow 0$ . Il est égal à deux fois le nombre de 4-composantes dans  $N(P)$ .

Si  $X_R(P) = 2$  et  $b(P) > 1$ , il est certain que les pixels noirs de  $N(P)$  sont 4-connexes.

**Point extrémité** : Ce sont les pixels qui ont les points de contour nuls sauf l'un des pixels  $x_6$ ,  $x_7$  ou  $x_8$ .

### 6.3.2. Nombre d'intersections de Hilditch [42]

$$X_H(P) = \sum_{i=1}^4 b_i \quad (7)$$
$$b_i = \begin{cases} 1 & \text{si } x_{2i-1} = 0 \text{ et } (x_{2i} = 1 \text{ ou } x_{2i+1} = 1) \\ 0 & \text{sinon} \end{cases}$$

$X_H(P)$  représente le nombre de composantes 8-connexes dans le 8-voisinage de  $P$ . On le note souvent  $C(P)$ , dans la suite du rapport, on va utiliser la deuxième notation.

### 3.4. Description des algorithmes de squelettisation

Nous présentons dans cette section 21 algorithmes de squelettisation utilisant l'approche parallèle. Nous mettrons en évidence les avantages et les inconvénients de chacun de ces algorithmes.

#### 3.4.1. Algorithme de Zhang et Suen (ZS) [43]

C'est un algorithme parallèle de type deux sous-itérations, il est décrit ci-dessous :

Supprimer en parallèle les points  $P$  de l'objet vérifiant les conditions suivantes :

- 1-  $2 \leq b(P) \leq 6$
- 2-  $X_R(P) = 2$  (8)
- 3-  $x_1 = 0$  ou  $x_7 = 0$  ou  $x_3 = x_5 = 0$  (itérations paires)
- 4-  $x_3 = 0$  ou  $x_5 = 0$  ou  $x_1 = x_7 = 0$  (itérations impaires)

##### 3.4.1.1. Description

La première condition assure que  $P$  est un point de contour et que sa suppression ne provoquera pas de trou, la deuxième assure que les pixels noirs de  $N(P)$  sont 4-connexes, dans ce cas, si  $2 \leq b(P) \leq 6$ , il est supprimable sans risque de modification de la connexité de la forme. La troisième condition s'attache à retirer les points de bord *Est* ou *Sud* et les coins *Nord-Ouest*. La quatrième condition s'attache à retirer les points de bord *Nord* ou *Ouest* et les coins *Sud - Est*.

##### 3.4.1.2. Propriétés

- Conserve les relations de connectivité.
- Préserve la topologie de l'image originale.
- L'épaisseur du squelette est mince.
- Supprime des parties du squelette.
- Sensible aux bruits.
- réduit des lignes obliques d'épaisseur 2, en 2 points.

### 3.4.2. Algorithme de Lu et Wang (LW) [44]

C'est un algorithme parallèle d'amincissement, la première condition de l'algorithme de Zhang et Suen (ZS) est remplacée par la condition suivante :  $3 \leq b(P) \leq 6$ , on obtient ainsi l'algorithme de Lü et Wang :

1.  $3 \leq b(P) \leq 6$
2.  $X_R(P) = 2$  (9)
3.  $x_1 = 0$  ou  $x_7 = 0$  ou  $x_3 = x_5 = 0$  (itérations paires)
4.  $x_3 = 0$  ou  $x_5 = 0$  ou  $x_1 = x_7 = 0$  (itérations impaires)

#### 3.4.2.1. Description

C'est une amélioration de l'algorithme de Zhang et Suen décrit précédemment. Cet algorithme ne réduit pas les lignes obliques d'épaisseur 2, en 2 points, mais ceci présente l'inconvénient sur le fait que l'épaisseur du squelette n'est pas nulle.

#### 3.4.2.2. Propriétés

- Conserve les relations de connectivité.
- Préserve la topologie de l'image originale.
- L'épaisseur du squelette n'est pas mince.
- Supprime des parties du squelette.
- Sensible aux bruits.
- Ne réduit des lignes obliques d'épaisseur 2, en 2 points.

### 3.4.3. Algorithme de Guo et Hall (GH) [45]

Dans cet algorithme, les notations suivantes sont utilisées :

$$N_1(P) = (x_1 \vee x_2) + (x_3 \vee x_4) + (x_5 \vee x_6) + (x_7 \vee x_8) \quad (10)$$

$$N_2(P) = (x_2 \vee x_3) + (x_4 \vee x_5) + (x_6 \vee x_7) + (x_8 \vee x_1)$$

$$N(P) = \text{Min}[N_1(P), N_2(P)]$$

Cet algorithme se décrit comme suit :

Supprimer en parallèle les points  $P$  de l'objet vérifiant les conditions suivantes :

- 1-  $C(P) = 1$
- 2-  $2 \leq N(P) \leq 3$  (11)
- 3-  $(x_2 \vee x_4 \vee \bar{x}_8) \wedge x_1 = 0$  (itérations paires)

$$4- (x_6 \vee x_7 \vee \bar{x}_4) \wedge x_5 = 0 \quad (\text{itérations impaires})$$

### 3.4.3.1. Description

La première condition de l'algorithme cible les points 8-supprimables, la deuxième permet la suppression de points de rubans inclinés d'épaisseur 2, contrairement à l'algorithme de Lu et Wang. La troisième condition cible certains points de bord *Nord* ou *Ouest*. La quatrième cible certains points de bord *Sud* ou *Ouest*.

### 3.4.3.2. Propriétés

- Respecte les propriétés du squelette.
- Ne réduit pas les lignes obliques d'épaisseur 2, en gardant l'amincissement du squelette.
- Ne supprime pas des parties du squelette.

### 3.4.4. Algorithme de Rosenfeld (ROS) [46]

C'est un algorithme parallèle de type quatre sous-itérations qui est décrit ci dessous :

Supprimer en parallèle les points  $P$  de l'objet vérifiant les conditions suivantes :

- 1-  $C(P) = 1$
- 2-  $P$  n'est pas un point extrémité. (12)
- 3-  $x_i = 0$  avec  $i$  prend les valeurs 3, 7, 1, 5, 3,7, .... Pour les sous-itérations successives.

### 3.4.4.1. Description

Cet algorithme permet de supprimer en parallèle les points 8-supprimables non extrémités, ces points sont également des points *Nord*, *Sud*, *Est* ou *Ouest*.

### 3.4.4.2. Propriétés

- Préserve la géométrie de l'image originale.
- L'épaisseur du squelette est mince.
- Préserve la topologie de l'image.

### 3.4.5. Algorithme de Holt, Stewart, Clint et Perrott (HSCP) [47]

C'est un algorithme de type fortement parallèle, le schéma de squelettisation est décrit par l'algorithme suivant :

Supprimer en parallèle tout point de l'objet vérifiant la condition  $a$ ) sauf ceux qui vérifient au moins l'une des trois conditions de la condition  $b$ ).

Condition a)

$$2 \leq b(P) \leq 6$$

$$X_R(P) = 2$$

Condition b)

(13)

$x_3 = x_7 = 1$  et  $x_1$  a été retenu dans l'étape a).

$x_1 = x_5 = 1$  et  $x_7$  a été retenu dans l'étape a).

$x_1, x_7$  et  $x_8$  ont été retenus dans l'étape a).

### 3.4.5.1. Description

La condition b-1) préserve certains points de bord *Ouest* dans une bande verticale d'épaisseur 2. La condition b-2) préserve certains points de bord *Nord* dans une bande horizontale d'épaisseur 2. La condition b-3), quand à elle, préserve au moins un point dans un carré  $2 \times 2$ .

### 3.4.5.2. Propriétés

- Préserve la géométrie de l'image.
- Vérifie l'amincissement du squelette.
- Ne préserve pas la topologie de l'image.
- Supprime des parties du squelette.
- réduit des lignes obliques d'épaisseur 2, en 2 points.

### 3.4.6. Algorithme de Hall (HSCP) [48]

Cet algorithme est noté HSCP. La première inéquation de la *condition a)* l'algorithme précédent est remplacée par la l'inéquation suivante,  $3 \leq b(P) \leq 6$ , l'algorithme complet est décrit comme suit :

Condition a)

$$3 \leq b(P) \leq 6$$

$$X_R(P) = 2$$

Condition b)

(14)

$x_3 = x_7 = 1$  et  $x_1$  a été retenu dans l'étape a).

$x_1 = x_5 = 1$  et  $x_7$  a été retenu dans l'étape a).

$x_1, x_7$  et  $x_8$  ont été retenus dans l'étape a).

### 3.4.6.1. Description

C'est une amélioration de l'algorithme précédent, il résout le problème de suppression des lignes obliques d'épaisseur nulle, cependant, le squelette n'est pas assez mince et la topologie de l'image n'est pas préservée.

### 3.4.6.2. Propriétés

- Préserve la géométrie de l'image.
- Vérifie l'amincissement du squelette.
- Ne préserve pas la topologie de l'image.
- Supprime des parties du squelette.
- Ne réduit des lignes obliques d'épaisseur 2, en 2 points.

### 3.4.7. Algorithmes de Guo et Hall (GH\_AFP1) [49]

C'est un algorithme fortement parallèle utilisant un support de 11 points. Le schéma est décrit par l'algorithme suivant :

Supprimer en parallèle les points  $P$  de l'objet vérifiant les conditions suivantes :

- 1-  $X_R(P) = 2$
  - 2-  $P$  est 4-adjacent à  $\bar{x}$
  - 3-  $b(P) > 2$
  - 4-  $L(P) = 0$
- (15)

Avec  $L(P) = [(\bar{x}_3 \wedge x_7 \wedge \bar{Q}_2) \wedge (x_1 \vee x_2 \vee x_8) \wedge (x_4 \vee x_5 \vee x_6)] \vee (\bar{x}_1 \wedge x_5 \wedge \bar{Q}_1)$

$Q_1 = A(i, j - 2)$  et  $Q_2 = A(i + 1, j)$

$A$  représente la matrice des valeurs des pixels de l'image.

### 3.4.7.1. Description

On prend  $L(P) = D(P) \vee E(P)$

avec  $D(P) = [(\bar{x}_3 \wedge x_7 \wedge \bar{Q}_2) \wedge (x_1 \vee x_2 \vee x_8) \wedge (x_4 \vee x_5 \vee x_6)]$  (16)

et  $E(P) = (\bar{x}_1 \wedge x_5 \wedge \bar{Q}_1)$ .

Si  $P$  est un bord *Est* dans un ruban vertical d'épaisseur 2, alors  $L(P) = 1$  car  $E(P) = 1$ .

Si  $P$  est un bord *Nord* dans un ruban horizontal d'épaisseur 2, alors  $L(P) = 1$  car  $D(P) = 1$ .

### 3.4.7.2. Propriétés

- Le squelette est assez mince.
- Préserve la géométrie de l'image.

- Ne préserve pas la topologie de l'image.
- Les rubans diagonaux d'épaisseur 2 ne sont pas amincis

### 3.4.8. Algorithmes de Guo et Hall (GH\_AFP2) [49]

C'est un algorithme fortement parallèle, le schéma est décrit par l'algorithme suivant :

- 1-  $X_R(P) = 2$
- 2-  $P$  est 4-adjacent à  $\bar{x}$  (17)
- 3-  $b(P) > 2$
- 4- Le voisinage de  $P$  ne satisfait aucun des motifs suivants (Figure.2.16):

	0			1	1			0	0	0		
1	P	1		0	1	P	0		0	1	P	0
1	1	1			1	1			1	1	0	
	0									0		

**Figure.2.16.** Condition 4 de l'algorithme GH\_AFP2

#### 3.4.8.1. Description

La dernière condition de cet algorithme permet de ne pas perdre les rubans horizontaux ou verticaux d'épaisseur 2, ou un carré  $2 \times 2$ , tout en gardant la topologie de l'image. Cet algorithme est une version améliorée du précédent, il préserve la géométrie de l'image, mais il supprime des parties du squelette.

#### 3.4.8.2. Propriétés

- Préserve la géométrie de l'image.
- Les rubans diagonaux d'épaisseur 2 ne sont pas amincis
- Ne préserve pas la topologie de l'image.
- Supprime des parties du squelette
- Le squelette est n'assez pas assez mince.

### 3.4.9. Algorithmes de Guo et Hall (GH\_AFP3) [49]

C'est un algorithme fortement parallèle, il est décrit en suivant le schéma :

Supprimer en parallèle les points  $P$  de l'objet vérifiant l'une des conditions suivantes :

*Condition a)*

- 1-  $X_R(P) = 2$
- 2-  $P$  est 4-adjacent à  $\bar{x}$  (18)

3-  $b(P) > 2$

4- Le voisinage de  $P$  ne satisfait aucun des motifs suivants (Figure.2.17, Figure.2.18):

	0			1	1			0	0	0
1	P	1		0	1	P	0		0	1
1	1	1			1	1			1	1
	0									0

**Figure.2.17.** Condition a)-4 de l’algorithme GH\_AFP3

Condition b)

	0			0	0	0
1	P	1		0	1	P
1	1	1			1	1
						0

**Figure.2.18.** Condition b) de l’algorithme GH\_AFP3

### 3.4.9.1. Description

La troisième version de cet algorithme est plus performante, Les rubans obliques d’épaisseur 2 sont maintenant amincis, ce qui permet l’obtention d’un squelette plus mince, cependant, des partis du squelette sont supprimés, ce qui veut dire que cet algorithme ne préserve pas la topologie de l’image.

### 3.4.9.2. Propriétés

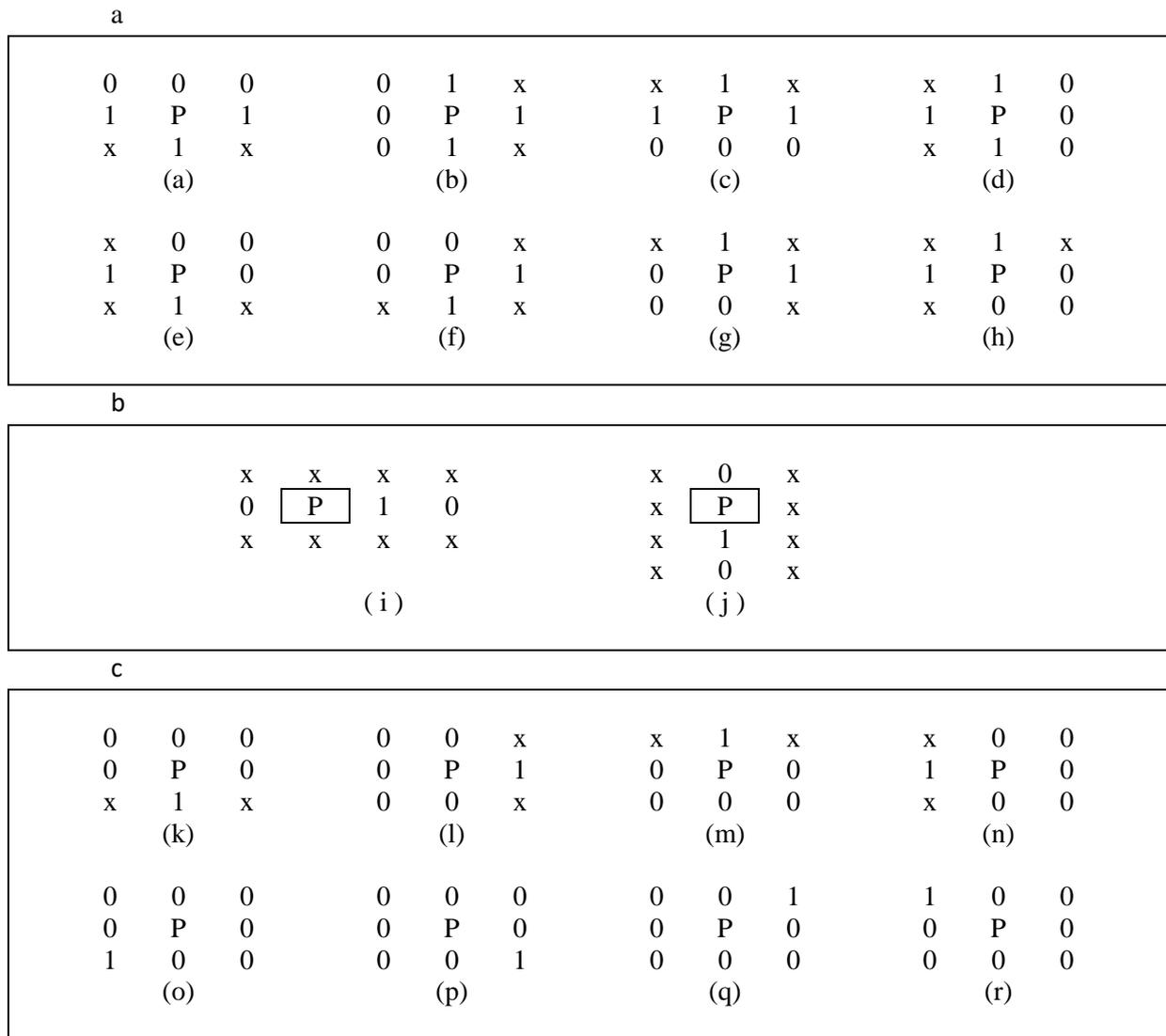
- Préserve la géométrie de l’image.
- Le squelette est assez mince.
- Les rubans diagonaux d’épaisseur 2 ne sont pas amincis
- Ne préserve pas la topologie de l’image.
- Supprime des partis du squelette

### 3.4.10. Algorithme de Chin, Wan, Stover et Iverson (CWSI) [50]

C’est un algorithme de type fortement parallèle d’un support de 15 points. Le schéma est décrit par l’algorithme suivant :

Supprimer en parallèle les points  $P$  de l’objet qui vérifient au moins une des *thinning* ou *trimming templates*, et qui ne vérifient pas une *restoring template* :

Le symbole “ $x$ ” signifie “appartient ou non à l’objet” (Figure.2.19):



**Figure.2.19.** Conditions de l’algorithme CWSI. a : thinning templates, b : restoring templates, c: trimming templates

### 3.4.10.1. Description

Cet algorithme est composé de thinning et trimming templates ciblant des candidats à la suppression auxquelles est nécessaire d’adjoindre des restoring templates, afin de ne pas supprimer des rubans horizontaux ou verticaux d’épaisseur 2 et le carré  $2 \times 2$ .

### 3.4.10.2. Propriétés

- Le squelette est assez mince.
- Préserve la géométrie de l’image.
- Préserve la topologie de l’image, mais pas dans tous les cas.
- Les rubans diagonaux d’épaisseur 2 sont supprimés.

### 3.4.11. Algorithme de Wu et Tsai (WT) [51]

C'est un algorithme de type fortement parallèle d'un support de 15 points. Le schéma est décrit par l'algorithme suivant :

Supprimer en parallèle les points  $P$  de l'objet qui vérifient au moins l'un des motifs suivants :

			1	1	y	1	1	1	y	1	1	x	y	0	y								
			1	P	0	1	P	1	0	P	1	1	1	P	1								
			1	1	y	y	0	y	y	1	1	x	1	1	1								
												x	1	x									
			(a)				(b)				(c)				(d)								
x	0	0	x	1	1	0	1	0	x	1	x	0	0	x	0	0	0						
1	P	0	0	P	1	0	P	1	1	P	0	0	P	1	0	P	1						
x	1	x	0	0	x	0	0	0	x	0	0	x	1	1	0	1	0						
			(e)				(f)				(g)				(h)				(i)				(j)
			0	0	0	1	0	0	1	1	1	0	0	1									
			0	P	0	1	P	0	0	P	0	0	P	1									
			1	1	1	1	0	0	0	0	0	0	0	1									
			(k)				(l)				(m)				(n)								

**Figure.2.20.** Conditions de l'algorithme WT

Le symbole “ $x$ ” signifie “appartient ou non à l'objet”, au moins l'un des deux symboles “ $y$ ” apparaissant dans un motif doit être un point du complémentaire de l'objet.

#### 3.4.11.1. Description

Cet algorithme est composé de trois parties : la première permet d'amincir des rubans horizontaux ou verticaux, la deuxième permet d'amincir des rubans diagonaux, la dernière permet la réduction de bruits apparaissant au voisinage de morceaux de lignes droites.

#### 3.4.11.2. Propriétés

- Préserve la géométrie de l'image.
- Préserve la topologie de l'image.
- N'est pas sensible aux bruits.
- Le squelette n'est pas assez mince.

### 3.4.12. Algorithme de Manzanera et Bernard (MB\_2) [52]

C'est un algorithme de type fortement parallèle. Le schéma est décrit par l'algorithme suivant :

Supprimer en parallèle les points  $P$  de l'objet vérifiant le motif  $\alpha_0$  ou le motif  $\alpha_1$ , et non le motif  $\beta$  (Figure.2.21):

		1		0	0			1	0
0	P	1	1	0	P	1		0	P
		1			1	1	1		
						1			
		$\alpha_0$			$\alpha_0$				$\beta$

**Figure.2.21.** Conditions de l'algorithme MB\_2

### 3.4.12.1. Description

Cet algorithme part du principe que la k-connexité du squelette est préservé si et seulement si la suppression de toute ou partie de l'ensemble ne change pas la connexité de la forme quel que soit l'ordre de suppression.

Manzanera et Bernard ont démontré que cette condition est équivalente à la suppression simultanée des pixels dont le voisinage est  $\alpha_0$  ou  $\alpha_1$  mais qui n'est pas  $\beta$ .

### 3.4.12.2. Propriétés

- Préserve la géométrie de l'image.
- Ne préserve pas la topologie de l'image.
- Supprime des parties du squelette.
- Les rubans d'épaisseur 2 horizontaux, verticaux ou obliques ne peuvent être amincis
- Assez long dans l'exécution.

### 3.4.13. L'algorithme de Gonzales et Wood (GW) [53]

Cet algorithme est fortement parallèle, il consiste à marquer comme supprimables les pixels qui vérifient la relation suivante :

$$3 \leq b(P) \leq 5 \tag{19}$$

### 3.4.13.1. Description

C'est une approche intuitive d'amincissement parallèle en une itération. Il est sensible aux bruits et a tendance à supprimer des parties du squelette.

### 3.4.13.2. Propriétés

- Préserve la géométrie de l'image.
- Préserve la topologie de l'image.
- Supprime des parties du squelette.
- La continuité du squelette n'est pas assurée.

### 3.4.14. L'algorithme de Rutovitz (RUT) [41]

Algorithme parallèle d'amincissement, il balaye l'image et supprime le pixel si et seulement si toutes les conditions suivantes sont vraies

- 1-  $2 \leq b(P) \leq 6$
  - 2-  $X_R(P) = 2$
  - 3-  $x_1x_3x_5 = 0$  ou  $X_R(x_3) \neq 2$
  - 4-  $x_1x_3x_7 = 0$  ou  $X_R(x_1) \neq 2$
- (20)

#### 3.4.14.1. Description

La troisième condition de cet algorithme permet de ne pas supprimer totalement les rubans verticaux, tandis que la quatrième permet de ne pas supprimer totalement les rubans horizontaux.

#### 3.4.14.2. Propriétés

- Préserve la géométrie de l'image.
- Préserve la topologie de l'image.
- Assure l'amincissement du squelette.
- conduit parfois à un squelette décalé par rapport à l'axe médian.
- ne réduit pas les lignes doubles 8-connexes.

### 3.4.15. L'algorithme de Deutsch (DEU) [54]

C'est un algorithme parallèle de type quatre sous-itérations. Le schéma est décrit par l'algorithme suivant :

Supprimer en parallèle les points  $P$  de l'objet qui vérifient les conditions suivantes :

*-Première sous-itération :*

- 1-  $X_R(P) = 0, 2$  ou  $4$
  - 2-  $b(P) \neq 1$
  - 3-  $x_1x_3x_5 = 0$
  - 4-  $x_1x_3x_7 = 0$
  - 5- Si  $X_R(P) = 4$  alors une des conditions suivantes doit être vraie :
    - (a)  $x_1x_7 = 1, x_2 + x_6 \neq 0$  et  $x_3 + x_4 + x_5 + x_8 = 0$
    - (b)  $x_1x_3 = 1, x_4 + x_8 \neq 0$  et  $x_2 + x_5 + x_6 + x_7 = 0$
- (21)

-Deuxieme sous-itération :

- 1-  $X_R(P) = 0, 2 \text{ ou } 4$
- 2-  $b(P) \neq 1$
- 3-  $x_1x_5x_7 = 0$
- 4-  $x_3x_5x_7 = 0$
- 5- Si  $X_R(P) = 4$  alors une des conditions suivantes doit être vraie :
  - (a')  $x_1x_5 = 1, x_2 + x_6 \neq 0$  et  $x_1 + x_4 + x_7 + x_8 = 0$ .
  - (b')  $x_3x_5 = 1, x_4 + x_8 \neq 0$  et  $x_1 + x_2 + x_3 + x_6 = 0$ .

(22)

### 3.4.15.1. Description

DEUTSCH a ajouté une condition à celles de RUTOVITZ pour résoudre le problème des lignes doubles 8-connexes et tenter d'aligner le squelette avec l'axe médian, en revanche, les carrés sont souvent complètement érodés.

### 3.4.15.2. Propriétés

- Préserve la géométrie de l'image.
- Préserve la topologie de l'image.
- Assure l'amincissement du squelette.
- Résout le problème de décalage du squelette par rapport à l'axe médian.
- Réduit les lignes doubles 8-connexes.
- Les carrés sont souvent complètement érodés.

### 3.4.16. Algorithme FP-Isthmus [55]

Dans la suite de ce chapitre, on utilise les notations suivantes:

$N_{4-8}(P)$ : l'ensemble des 4-8 voisinages de  $P$ .

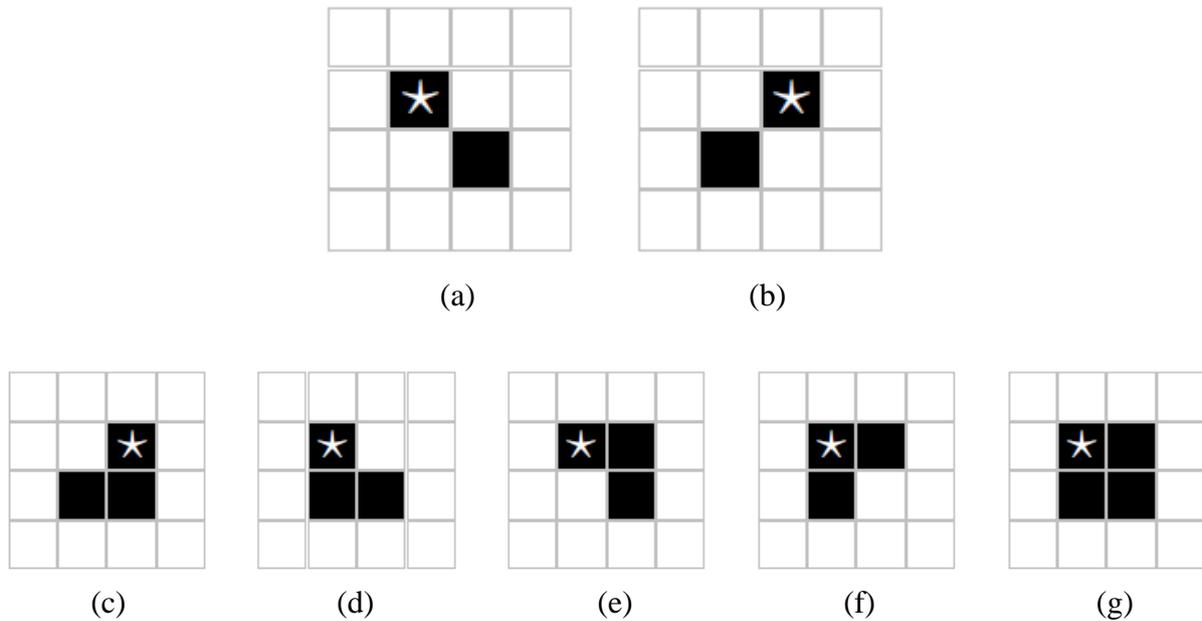
$$p < q \iff p_y < q_y \text{ ou } (p_y = q_y \text{ et } p_x < q_x) \quad (23)$$

$P$  est un point simple s'il est un point de contour et  $C(P)=1$ .

L'algorithme FP-Isthmus consiste à supprimer en parallèle les points  $P$  qui vérifient les conditions suivantes:

1.  $P$  est un point simple dans  $B$ .
2. Pour chaque point  $q \in N_4(P) \cap B$ ,  $P$  est simple dans  $B-\{q\}$ , ou  $q$  est simple dans  $B-\{p\}$ , ou  $q < P$ .
3.  $P$  ne coïncide pas avec les points marqués dans la Figure.2.22.

$B$  constitue l'image à traiter.



**Figure.2.22.** Conditions de la non suppression des points marqués 'x'

### 3.4.16.1. Description

Les conditions de suppression de cet algorithme sont le résultat d'un théorème décrit dans [54] pour préserver la topologie du squelette.

### 3.4.16.2. Propriétés

- Préserve la géométrie de l'image.
- Préserve la topologie de l'image.
- Supprime des parties du squelette.
- La continuité du squelette n'est pas assurée.

### 3.4.17. Algorithme SI-Q-Isthmus[54]

Un point  $P$  est N-border si  $P_N$  est un point appartenant à la forme (Figure.2.23)

$PNW$	$PN$	$PNE$
$PW$	$P$	$PE$
$PSW$	$PS$	$PSE$

**Figure.2.23.** Notation des voisinages 3\*3 d'un point  $P$

L'algorithme SI-Q-Isthmus supprime en parallèle les points qui vérifient les conditions suivantes:

Pour chaque point  $d \in Q$

1.  $P$  est simple et d-border.

2. Pour chaque point d-border  $q \in N_4(P)$ ,  $q$  est simple dans  $N_8(P) - \{q\}$ , ou  $q$  est simple dans  $N_8(q) - \{P\}$ , ou  $p < q$ .
3. Les conditions suivantes sont satisfaisantes:
  - Si  $d \in \{N, E, S, W\}$ , alors  $P$  ne coïncide pas avec les point marqués 'x' dans la Figure.2.19 (a) et (b).
  - Si  $d = NE$ , alors  $P$  ne coïncide pas avec les points marqués 'x' dans la Figure.2.19 (a), (b), (c), (e) et (f).
  - Si  $d = SW$ , alors  $P$  ne coïncide pas avec les points marqués 'x' dans la Figure.2.19 (a), (b), (c), (d) et (f).
  - Si  $d = NW$ , alors  $P$  ne coïncide pas avec les points marqués 'x' dans la Figure.2.19 (a), (b), (d), (e) et (f).
  - Si  $d = SE$ , alors  $P$  ne coïncide pas avec les points marqués 'x' dans la Figure.2.22 (a), (b), (c), (d) et (e).

avec  $d \in \{N, E, S, W, NE, SW, NW, SE\}$ .

$$Q = \langle NE, SW \rangle, \langle N, E, S, W \rangle, \langle NE, SW, NW, SE \rangle.$$

### 3.4.17.1. Description

De même que l'algorithme précédent, Les conditions de suppression de cet algorithme sont le résultat d'un théorème décrit dans [54] pour préserver la topologie du squelette.

### 3.4.17.2. Propriétés

- Préserve la géométrie de l'image.
- Préserve la topologie de l'image.
- Supprime des partis du squelette.
- La continuité du squelette n'est pas assurée.

### 3.18. Algorithme SF-k-Isthmus[55]

On considère les ensembles suivants:

$$S_2(i) = \{p = (x, y) | (x + y) = (i \bmod 2)\} \tag{24}$$

$$S_4(i) = \{p = (x, y) | 2 \cdot (y \bmod 2) + (x \bmod 2) = i\}$$

avec  $k = \{2, 4\}$  et  $i = \{1, \dots, k - 1\}$ .

L'algorithme SF-k-Isthmus consiste à supprimer les points  $P$  qui vérifient les conditions suivantes:

Pour  $i = 1$  jusqu'à  $k - 1$

$P$  est un point simple dans la sous-zone  $S_k(i)$ .

$P$  ne coïncide pas avec les point marqués 'x' dans la Figure.2.22 (a) et (b).

Fin pour

### 3.4.18.1. Description

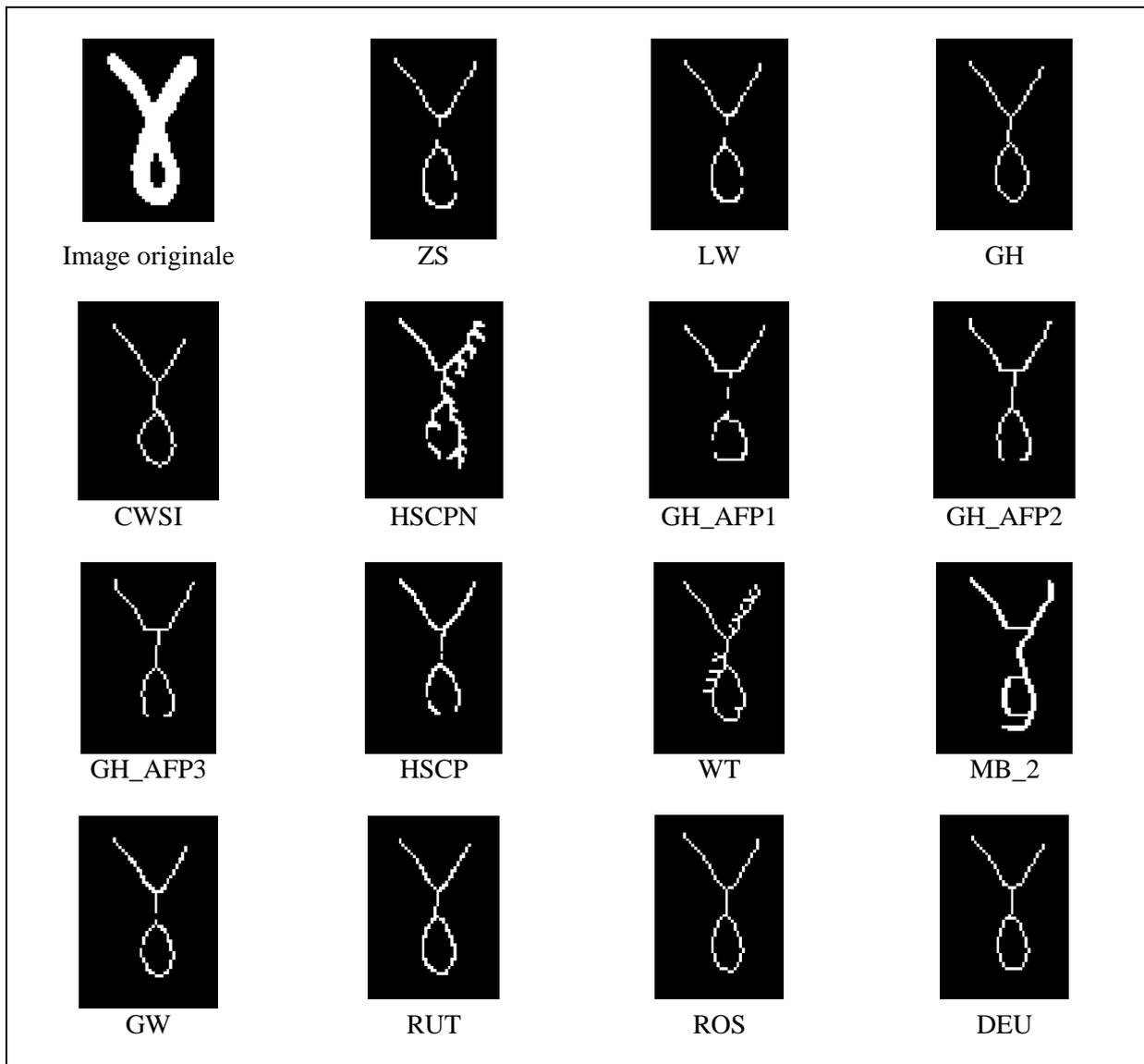
Les conditions de suppression de cet algorithme sont le résultat d'un théorème décrit dans [54] pour préserver la topologie du squelette.

### 3.4.18.2. Propriétés

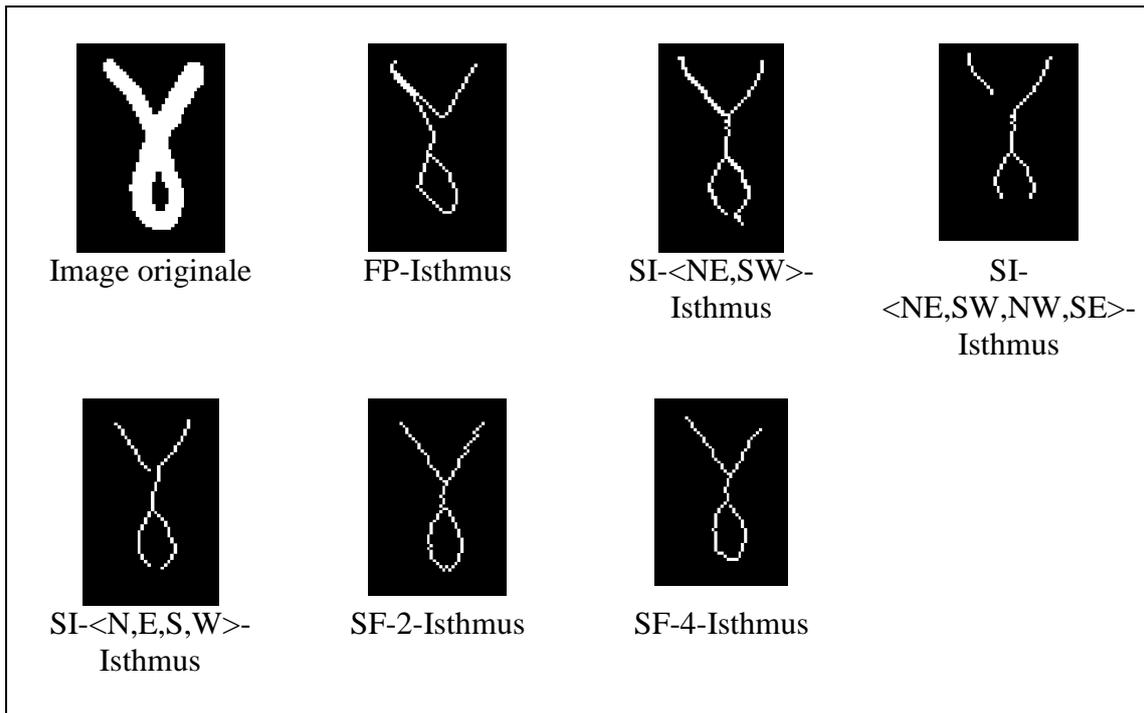
- Préserve la géométrie de l'image.
- Préserve la topologie de l'image.
- Supprime des parties du squelette.
- La continuité du squelette n'est pas assurée.

### 3.5. Résultats

Dans cet annexe, nous allons appliquer les algorithmes de squelettisation mentionnés précédemment sur le caractère Gujarati '1' et sur le caractère '4' de la base MNIST.



**Figure.2.24.** Comparaison des algorithmes de squelettisation sur le caractère Gujarati '4'

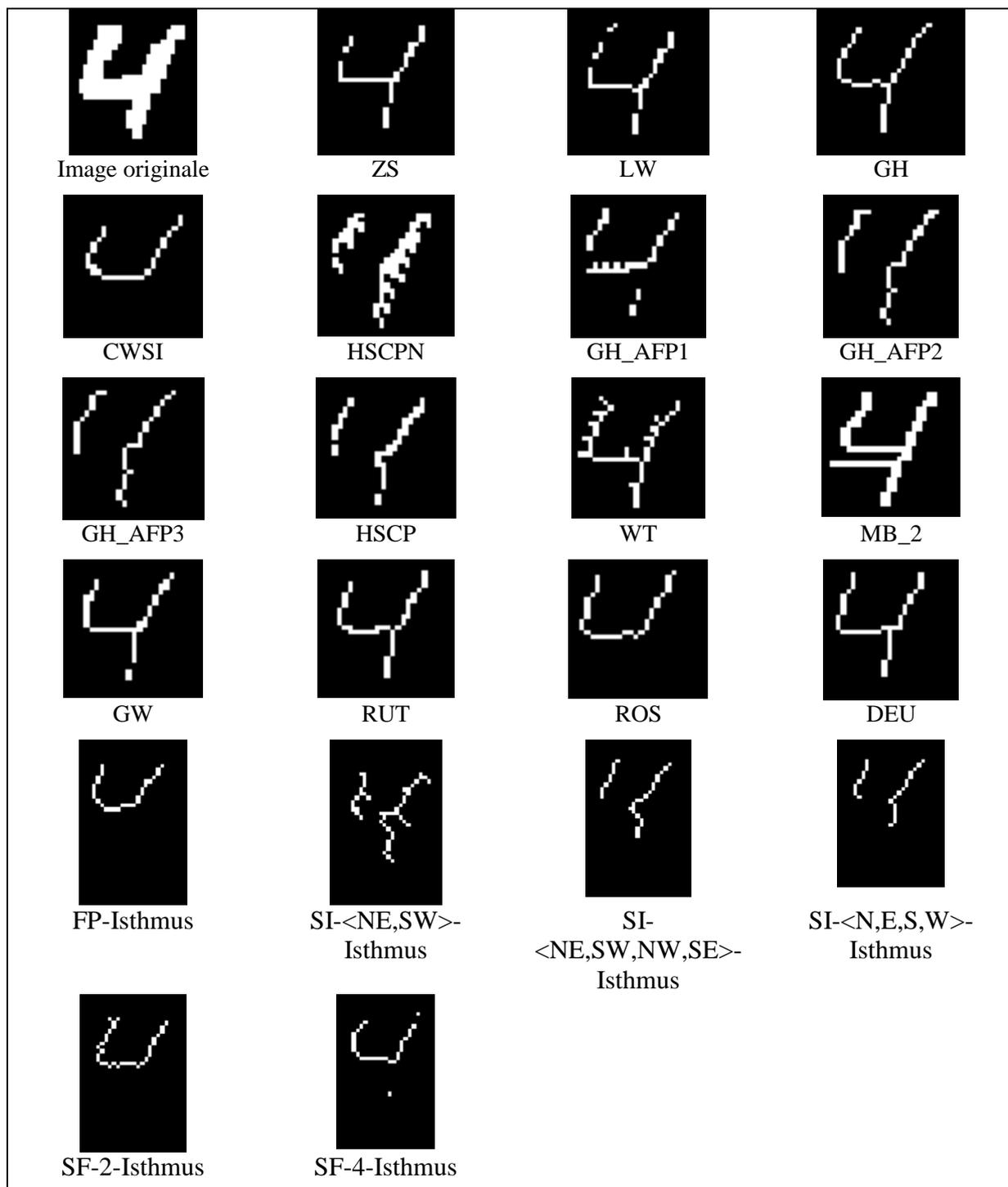


**Figure.2.24.** Comparaison des algorithmes de squelettisation sur le caractère Gujarati '4' (suite)

D'après les résultats effectués précédemment, on remarque que les algorithmes de Rosenfeld, Deutsch, Rutovitz et Guo\_Hall donnent d'assez bons résultats au niveau de la qualité du squelette, cependant, l'algorithme de Rutovitz crée un squelette décalé par rapport à l'axe médian, celui de Deutsch supprime souvent des formes carrés de l'image. En ce qui concerne l'algorithme de Rosenfeld, même si le nombre d'itérations est élevé du fait qu'il effectue quarts sous-itérations pour son implémentation, le temps de son exécution est acceptable, cependant, des fois, il supprime des parties du squelette. Pour l'algorithme de Guo\_Hall, il donne de bons résultats, le seul inconvénient avec cette méthode se pose au niveau de la continuité, c'est-à-dire qu'une petite modification de l'image originale n'entraîne pas de petites modifications sur le squelette (Figure.2.25).



**Figure.2.25.** Continuité de l'Algorithme de Guo-Hall



**Figure.2.26.** Comparaison des algorithmes de squelettisation sur le caractère '4' de la base MNIST

#### 4. Conclusion

Nous avons exposé dans ce chapitre deux parties essentielles dans la phase de prétraitement, à savoir la binarisation et la squelettisation. D'après l'étude comparative que nous avons effectuée, nous avons constaté que l'algorithme de Wolf dans le cas de la binarisation et l'algorithme de Guo-Hall pour le cas de la squelettisation sont ceux qui correspondent le mieux au contexte de la reconnaissance des caractères. Dans le chapitre qui suit, nous allons exposer la phase de l'extraction des caractéristiques.

## Chapitre 3

# Extraction des Caractéristiques

### 1. Introduction

L'extraction des caractéristiques est une étape de grande importance: si elle est mal conçue, il sera difficile, voire impossible, d'effectuer une reconnaissance efficace. Pour la reconnaissance de caractères manuscrits, les primitives recherchées doivent présenter une certaine invariance géométrique d'une part, et une certaine invariance statistique d'autre part. L'invariance géométrique signifie une tolérance vis-à-vis des opérateurs de translation, de rotation, et de changement d'échelle, tandis que l'invariance statistique signifie une tolérance vis-à-vis du bruit inévitablement présent sur chaque caractère.

Il existe un nombre important de caractéristiques possibles que l'on peut extraire d'une forme finie à deux dimensions. Il faut toutefois ne s'attarder qu'aux caractéristiques qui ont une pertinence possible pour la classification, ce qui suppose qu'au cours de la période de conception, le spécialiste s'attarde aux caractéristiques qui, selon une certaine technique de classification, apporteront les résultats les plus certains et les plus efficaces.

Dans ce chapitre, nous allons détailler les objectifs de l'extraction des caractéristiques, puis nous allons exposer quelques méthodes reconnues et les méthodes d'extraction utilisées dans cette thèse.

### 2. Objectifs de l'Extraction des Caractéristiques

Au cours de l'extraction des caractéristiques, plusieurs objectifs, qui précèdent la reconnaissance, peuvent être envisagés. Les principaux objectifs que nous définirons dans la perspective de la reconnaissance de l'écriture manuscrite sont : l'analyse, le paramétrage, la modélisation, le codage et la classification. [56]

- **L'Analyse**, dont la définition littérale est "la décomposition d'un tout en ses parties", consiste généralement en l'extraction d'un ensemble d'attributs caractéristiques du texte. Concrètement, l'analyse d'un texte manuscrit consiste à recueillir des informations statistiques telles que la disposition des lignes d'écriture, leur orientation, leur régularité, l'espacement des mots et des lettres, la régularité et l'inclinaison des lettres, l'épaisseur du trait, ainsi que la ligature des lettres à l'intérieur des mots.
- **Le Paramétrage** consiste à établir une liste d'attributs représentés par une variable binaire (attribut présent ou absent) ou multivaluée (proportionnelle à l'importance de l'attribut), qui ont été détectés et évalués dans l'image. A la différence de l'analyse, le paramétrage ne concerne qu'un mot ou qu'une lettre en vue de sa reconnaissance.
- **La Modélisation** est la construction d'une représentation approximative de la forme entière. A la différence du paramétrage, l'objectif est la réduction de l'information utile, au minimum nécessaire pour représenter complètement la forme, en particulier son aspect et sa structure. Si la modélisation est une description proche d'aspect de la

forme originale, on parle alors d'une schématisation, avec une approximation plus ou moins importante. Sinon, il s'agit d'un codage de l'information.

- **La classification** peut être considérée comme une identification partielle de l'information. L'objectif de la classification est, lorsque l'on ne dispose pas de toute l'information nécessaire à l'identification complète de la forme (cas de la reconnaissance), de déterminer quand même une catégorie à laquelle elle appartient. Le principe est que moins d'information est nécessaire pour distinguer les caractères que pour les reconnaître.

### 3. Présentation de Quelques Méthodes d'Extraction des Caractéristiques

Dans la suite de cette section, nous présenterons quelques grandes familles de primitives couramment utilisées :

#### 3.1. Moments de Hu

Les moments invariants ont reçu une attention considérable dans le domaine de reconnaissance des formes. En 1961, Hu [57] a été l'un des premiers à introduire les moments invariants. Il a pu obtenir un ensemble de moments invariants par changement d'échelle, translation et de rotation.

Les moments invariants de Hu d'ordre  $\leq 3$  sont utilisés pour la première fois dans la reconnaissance des images des avions et des bateaux [58,59]. Les moments d'ordre  $\leq 2$  sont aussi utilisés dans la reconnaissance des caractères [60].

Les moments d'ordre  $p+q$  de la fonction  $f(x, y)$  (qui est égal au niveau de gris dans une image binaire) sont définis par :

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x, y) dx dy \quad (25)$$

Dans le cas d'une image binaire, la fonction  $f$  est définie par :  $f(x, y) = 1$  sur la région fermée et  $f(x, y) = 0$  ailleurs.

Les moments centrés peuvent être exprimés par rapport au centre de gravité de l'objet  $(x_g, y_g)$ , ils sont calculés par :

$$\mu_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x - x_g)^p (y - y_g)^q f(x, y) dx dy \quad (26)$$

avec  $x_g = m_{10}/m_{00}$  et  $y_g = m_{01}/m_{00}$ .

Dans le cas d'une image binaire, l'équation précédente devient :

$$\mu_{pq} = \sum_{k=0}^p \sum_{j=0}^q (x - x_g)^k (y - y_g)^j f(x, y) \quad (27)$$

Hu a pu extraire, à partir de ces moments centraux normalisés, un ensemble de 6 moments qui sont invariants par translation, rotation et homothétie. Ces moments sont données ci-dessous :

$$M_1 = m_{20} + m_{02} \quad (28)$$

$$M_2 = (m_{20} - m_{02})^2 + 4m_{11}^2 \quad (29)$$

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

$$M_4 = (m_{30} + m_{12})^2 + (m_{21} + m_{03})^2 \quad (31)$$

$$M_5 = (m_{30} - 3m_{12})(m_{30} + m_{12})[(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] \quad (32)$$

$$+ (3m_{21} - m_{03})(m_{21} + m_{03})[3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2]$$

$$M_6 = (m_{20} - m_{02})[(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] \quad (33)$$

$$+ 4m_{11}(m_{30} + m_{12})(m_{21} + m_{03})$$

### 3.2. Descripteurs de Fourier Elliptiques

Les descripteurs de Fourier elliptiques sont introduits dans [61] pour décrire les formes des contours fermés. Une démonstration visuelle de leur fonctionnement est disponible sur [64].

Soit un contour de  $T$  pixels, dont chaque pixel à pour coordonnées  $(x(t), y(t))$ , avec  $t \in \{0, 1, \dots, T-1\}$ .

$x : t \rightarrow x(t)$  et  $y : t \rightarrow y(t)$  sont périodiques de période  $T$  (le temps nécessaire pour tracer l'ensemble du contour à vitesse constante). Ces signaux périodiques sont décomposables en séries de Fourier:

$$x(t) = A_0 + \sum_{n=1}^{\infty} \left[ a_n \cos \frac{2n\pi t}{T} + b_n \sin \frac{2n\pi t}{T} \right] \quad (34)$$

$$y(t) = C_0 + \sum_{n=1}^{\infty} \left[ c_n \cos \frac{2n\pi t}{T} + b_n \sin \frac{2n\pi t}{T} \right] \quad (35)$$

avec,

$$A_0 = \frac{1}{T} \int_0^T x(t) dt \quad (36)$$

$$C_0 = \frac{1}{T} \int_0^T y(t) dt \quad (37)$$

$$a_n = \frac{2}{T} \int_0^T x(t) \cos \frac{2n\pi t}{T} dt \quad (38)$$

$$b_n = \frac{2}{T} \int_0^T x(t) \sin \frac{2n\pi t}{T} dt \quad (39)$$

$$c_n = \frac{2}{T} \int_0^T y(t) \cos \frac{2n\pi t}{T} dt \quad (40)$$

$$d_n = \frac{2}{T} \int_0^T y(t) \sin \frac{2n\pi t}{T} dt \quad (41)$$

Différents niveaux d'approximation peuvent être obtenus en faisant varier le nombre d'harmoniques. L'approximation de rang N est obtenue comme:

$$x_N(t) = A_0 + \sum_{n=1}^T \left[ a_n \cos \frac{2n\pi t}{T} + d_n \sin \frac{2n\pi t}{T} \right] \quad (42)$$

$$y_N(t) = C_0 + \sum_{n=1}^N \left[ c_n \cos \frac{2n\pi t}{T} + d_n \sin \frac{2n\pi t}{T} \right] \quad (43)$$

Les coefficients  $a_n$ ,  $b_n$ ,  $c_n$  et  $d_n$  peuvent être utilisés comme primitives.

### 3.3. Profils et Contours

Dans [63], les auteurs présentent un certain nombre de primitives couramment utilisées en reconnaissance de l'écriture. Ils distinguent trois catégories :

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

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

#### 3.3.1. Primitives Perceptuelles

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

### 3.3.2. Primitives Directionnelles

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

### 3.3.3. Primitives Topologiques

Ce type de primitives est basé sur des densités de pixels. On peut par exemple projeter des images de tailles différentes dans une matrice de taille fixe. Les caractères extraits sont les valeurs des cellules de cette matrice.

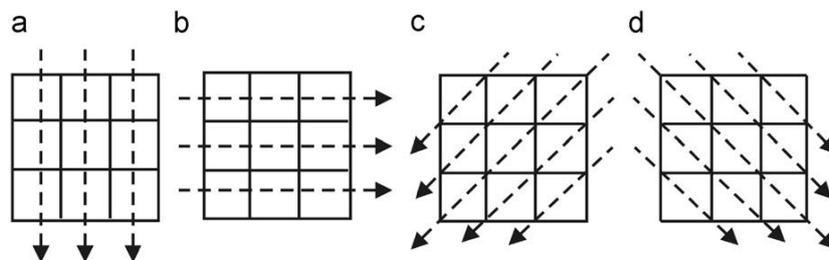
Dans ce type de primitives, on compte également des profils et histogrammes. Pour maintenir un vecteur de taille fixe, on divise l'image en un nombre fixe de bandes horizontales et verticales. Les caractéristiques sont les moyennes des valeurs sur ces bandes.

C'est ce type d'extraction des caractéristiques qui est utilisé dans ce rapport.

## 4. Méthodes d'Extraction des Caractéristiques Utilisées

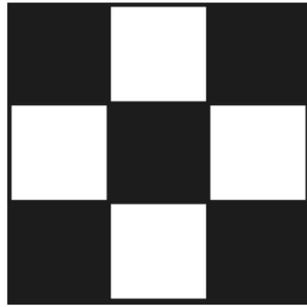
### 4.1. Méthode de la Somme

La première méthode qu'on décrit dans ce paragraphe est citée dans [7], elle consiste à effectuer la somme des pixels de l'image aux niveaux horizontaux, verticales et des diagonales (Figure.3.1).



**Figure.3.1.** Somme des pixels d'une matrice 3\*3 pixels. a : somme des pixels verticalement, b : somme des pixels horizontalement, c : somme des pixels sur les diagonales gauches, d : somme des pixels sur les diagonales droites.

Par exemple, en considérant la forme de la Figure.3.2, le vecteur d'extraction est (2,1,2,2,1,2,1,0,3,0,1,1,0,3,0,1). Le Tableau.3.1 décrit en détail ce vecteur.



**Figure.3.2** Forme 3\*3 pixels

<b>Figure.3.2</b>	<b>Vecteur d'Extraction</b>
A	(2,1,2)
B	(2,1,2)
C	(1,0,3,2,1)
D	(1,0,3,2,1)

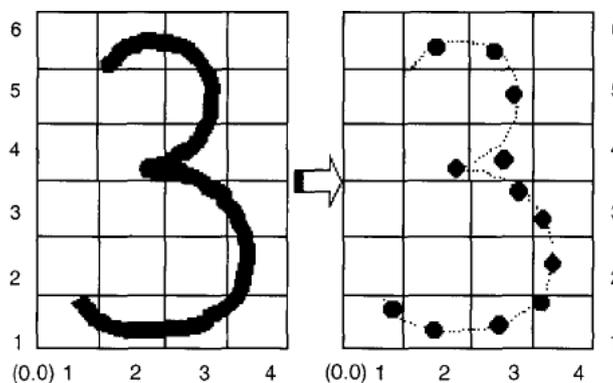
**Tableau.3.1.** Vecteurs d'Extraction

Dans la suite, on nome cette méthode: La méthode de la somme.

#### 4.2. Méthode de la Distance

La deuxième méthode utilisée dans cette thèse est l'approche par blocks citée dans [64,65,66], cette méthode exige la division de l'image en blocks, l'avantage majeur de cette approche découle de sa robustesse à de petites variations, la facilité de mise en œuvre et les taux de reconnaissances relativement élevés. Le choix des blocks et leurs nombres sont discutés dans la section 5.

En considérant le caractère de la Figure.3.3, l'image est divisée en  $6*4=24$  blocks, il pourrait y avoir des blocks qui sont vides, toutefois, toutes les cases prises en considération pour l'analyse dans un ordre séquentiel.



**Figure.3.3.** Division de l'image en blocks et le calcul des distances

En considérant le coin en bas à gauche comme l'origine absolue (0,0), la distance de coordination pour le  $k^{ieme}$  pixel dans le  $b^{ieme}$  block à l'emplacement (i,j) est calculée ainsi:

$$d_{kb} = \sqrt{(i^2 + j^2)} \quad (44)$$

En divisant la somme des distances des pixels noirs présents dans le block par le nombre total des pixels de chaque block, le vecteur de distance normalisé ( $\gamma_b$ ) est obtenu ainsi:

$$\gamma_b = \frac{1}{n_b} \sum_{k=1}^{n_b} d_k^b, \quad b = 1, 2, \dots, 24 \quad (45)$$

avec  $n_b$  est le nombre total des pixels pour le  $b^{ieme}$  block.

Par la suite, les 24  $\gamma_b$  correspondant aux 24 blocks constituent le vecteur d'extraction des caractéristiques, toutefois, pour les blocks vides, la valeur de la distance est zéro.

Dans la suite, on nome cette méthode: méthode de la distance.

## 5. Conclusion

Dans ce chapitre, on a présenté une vue générale sur l'extraction des caractéristiques et quelque méthodes largement utilisées dans la littérature. Le choix des primitives doit être très représentatif de l'ensemble des formes à prendre en compte. Ce choix est important car il conditionne toute la méthodologie mise en œuvre pour la reconnaissance. Il est irréversible et ne peut donc être remis en cause au cours de déroulement du système de reconnaissance.

## Chapitre 4

# Classification

### 1. Introduction

La classification est l'attribution d'une classe spécifique à un objet donné. Cette attribution a besoin d'un certain degré d'abstraction pour pouvoir extraire des généralités à partir des exemples dont on dispose.

Pour une machine, la classification de visages, de données médicales, de formes, sont toutes des tâches assez difficiles. Par exemple, dans le cas de la reconnaissance de caractères manuscrits, il est difficile d'énoncer une description générale qui tienne compte de toutes les variations particulières de chaque caractère. Une autre approche qui peut être utilisée pour cette tâche est celle de l'apprentissage. Ainsi, le critère pour décider si une image correspond ou non à une lettre 'A' consiste à comparer si cette image est suffisamment similaire à des 'A' vus auparavant. De ce point de vue, on ne calcule pas la classification de caractères: elle doit être apprise à partir d'exemples.

Nous présentons dans ce chapitre deux méthodes de classification largement répandues: les réseaux de neurones,  $k$  plus proches voisins ( $k$ -PPV) et la méthode de bayes naïve et Sparse Representation.

### 2. Réseaux de Neurones

Depuis plusieurs années, les réseaux de neurones artificiels, et particulièrement les *perceptrons multicouches*, se sont montrés très efficaces dans le domaine de la reconnaissance statistique de formes. Ces systèmes sont composés d'un ensemble structuré d'unités de traitement, appelées *neurones*, qui fonctionnent en parallèle et qui sont fortement interconnectées. Ces neurones réalisent chacun une fonction non linéaire de leurs entrées, qui est déterminée par un ensemble de paramètres dont les valeurs sont établies à la suite d'un apprentissage. Les non-linéarités présentes dans ces systèmes connexionnistes, ainsi que l'apprentissage discriminant qu'ils subissent, rendent ces derniers particulièrement bien adaptés aux tâches de classification. Cependant, leurs performances sont fortement affectées par la qualité de représentation des objets à classer. Pour des caractères, cela implique une grande dimension de représentation et, dans ce cas, les perceptrons multicouches peuvent devenir délicats à entraîner. Ils ne peuvent en outre être performants que si les caractéristiques qui leur sont fournies, sont pertinentes. Enfin, des réseaux de neurones de grande taille sont aussi très exigeants en puissance de calcul, ce qui restreint fortement leurs possibilités d'application.

En informatique, on appelle réseau de neurones un ensemble d'entités (les neurones) interconnectées. Dans la grande majorité des cas, les neurones sont en fait des fonctions calculées par un programme informatique, mais ils sont parfois réalisés sur des circuits électroniques. Les neurones sont caractérisés par un état d'excitation qui dépend de celui des neurones situés en amont, ainsi que de la force des liens qui les relient.

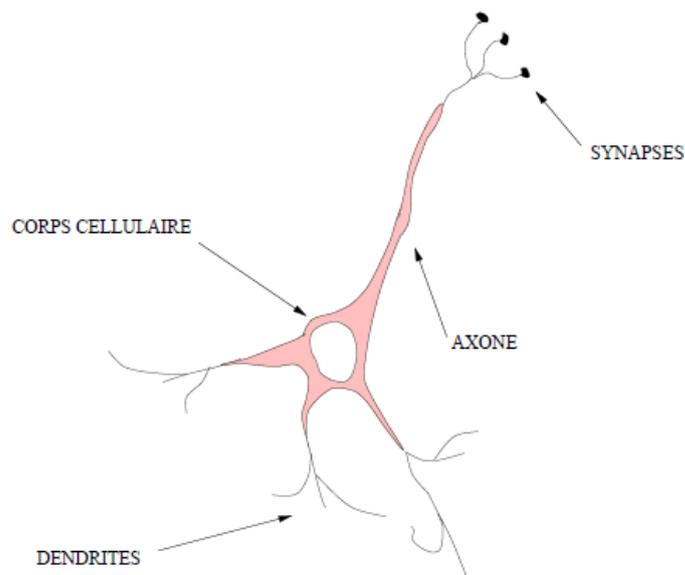
Étudions de plus près le fonctionnement d'un réseau neuronal classique.

## 2.1. Neurone Biologique

Comme les réseaux de neurones mis au point par les informaticiens sont largement inspirés de ce que la biologie nous apprend sur ceux que l'on trouve chez les êtres vivants, il convient d'abord de décrire brièvement le modèle biologique.

En biologie, un neurone est une cellule nerveuse dont la fonction est de transmettre un signal électrique dans certaines conditions. Il agit comme un relai entre une couche de neurones et celle qui la suit. Les caractéristiques des neurones sont encore mal connues (et font l'objet de recherches) mais on connaît leur principe d'action.

Le corps d'un neurone est relié d'une part à un ensemble de dendrites (entrées des neurones) et d'autre part à un axone, partie étirée de la cellule, qui représentera pour nous sa sortie. Le neurone étudié est connecté aux neurones qui l'environnent : il reçoit au niveau de ses dendrites les signaux électriques des neurones "en amont", propagés par les axones de ces derniers. Les charges électriques s'accumulent dans le neurone jusqu'à dépasser un certain seuil : à ce moment la transmission du signal électrique se déclenche via son axone vers d'autres neurones "en aval" (Figure.4.1).



**Figure.4.1** Un neurone biologique et ses principaux composants

Chacun de ces neurones est par ailleurs fort complexe. Essentiellement, il s'agit de tissu vivant et de chimie. Les spécialistes des neurones biologiques commencent à peine à comprendre quelques uns de leurs mécanismes internes. On croit en général que leurs différentes fonctions neuronales, y compris celle de la mémoire, sont stockées au niveau des connexions (synapses) entre les neurones. C'est ce genre de théorie qui a inspiré la plupart des architectures de réseaux de neurones artificiels que nous aborderons dans ce cours. L'apprentissage consiste alors soit à établir de nouvelles connexions, soit à en modifier des existantes.

## 2.2. Historique

Reproduire l'intelligence de l'être humain constitue sans aucun doute le rêve le plus passionnant de beaucoup de chercheurs de notre siècle. Ce n'est qu'après la fin de la Seconde Guerre mondiale qu'ont commencé à apparaître quelques voies de recherche dans ce domaine. L'une des premières est sans doute celle issue de la neurobiologie. En effet, son idée de base consiste à modéliser le premier niveau de l'intelligence humaine : c'est le neurone biologique. En 1948, deux chercheurs américains, Mac Culloch et Pitts [67], ont donné naissance au premier modèle mathématique de neurone biologique, qu'ils ont appelé : **le neurone formel**. Les entrées d'un tel neurone sont pondérées par des coefficients que l'on appelle "poids synaptiques". Le neurone est modélisé en tant qu'automate binaire ayant les deux états actif (+1) et inactif (-1).

Au début des années 80, un certain nombre de publications scientifiques ont permis de relancer les travaux de recherche sur le sujet des réseaux de neurones [68,69] et depuis le premier congrès international sur les réseaux de neurones en 1987, on assiste à une explosion incroyable des colloques et des congrès autour de ce sujet. Les raisons de cette explosion peuvent être résumées ainsi :

- plusieurs architectures mettant en coopération un certain nombre de neurones formels, ont permis d'apporter des solutions du moins équivalentes aux solutions déjà existantes.
- l'utilisation du principe de l'apprentissage d'une base contenant des exemples tirés d'un problème réel, réduit considérablement les difficultés rencontrées pour la formalisation de tels problèmes ;
- la facilité de développement d'une solution neuronale dans la résolution d'un problème est très attractive et permet à des non spécialistes d'intégrer de telles solutions très rapidement.

Une révolution survient alors dans le domaine des réseaux de neurones artificiels : une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non-linéaires : le *perceptron multicouche* ne possède pas les défauts mis en évidence par Marvin Minsky. Proposé pour la première fois par Werbos [70], le Perceptron Multi-Couche apparaît en 1986. Ces systèmes reposent sur la *rétropropagation du gradient de l'erreur* dans des systèmes à plusieurs couches.

### 2.3. Modèle Formel

Les neurones formels sont dotés de caractéristiques inspirées de celles des neurones biologiques que nous avons passées en revue dans la section précédente:

- **Le potentiel d'activation des cellules nerveuses:** il s'agit ici d'une valeur numérique qui peut être transmise à des neurones en aval. Un neurone formel ne peut transmettre qu'une valeur unique qui correspond à son état d'activation.
- Les dendrites: des neurones biologiques leur permettent de recevoir différents signaux de l'extérieur. De la même manière, un neurone formel peut recevoir des signaux  $x_i$  de plusieurs neurones. Ces signaux sont combinés en un signal d'entrée unique

$$E = \sum w_i x_i \quad (46)$$

où les  $w_i$  sont les poids affectés aux signaux extérieurs.

- Les synapses: les nombres  $w_i$  pondèrent les signaux émis par les différents neurones situés en amont. On retrouve ici l'analogie des synapses qui peuvent être inhibitrices ( $w_i < 0$ ), ou excitatrices ( $w_i > 0$ ).
- une fonction d'activation 's' gère l'état du neurone formel. Généralement, si  $s \cong 1$  le neurone est excité et il est au repos si  $s \cong -1$  ou  $A \cong 0$  selon les cas. L'allure de cette fonction est généralement telle qu'il existe un seuil d'activation du neurone (Figure.4.2): le neurone n'est excité que s'il reçoit un signal d'entrée E supérieur à ce seuil s.

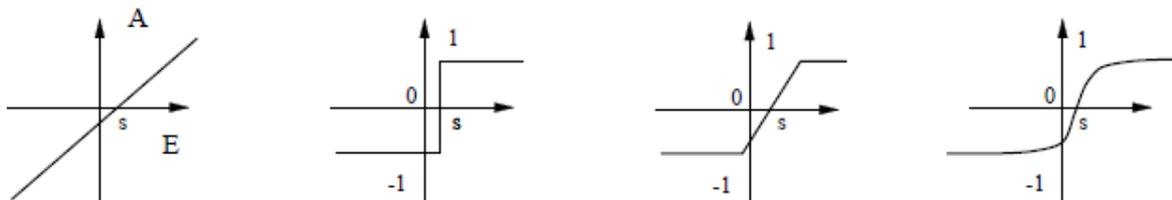


Figure.4.2. Quelques fonctions d'activation

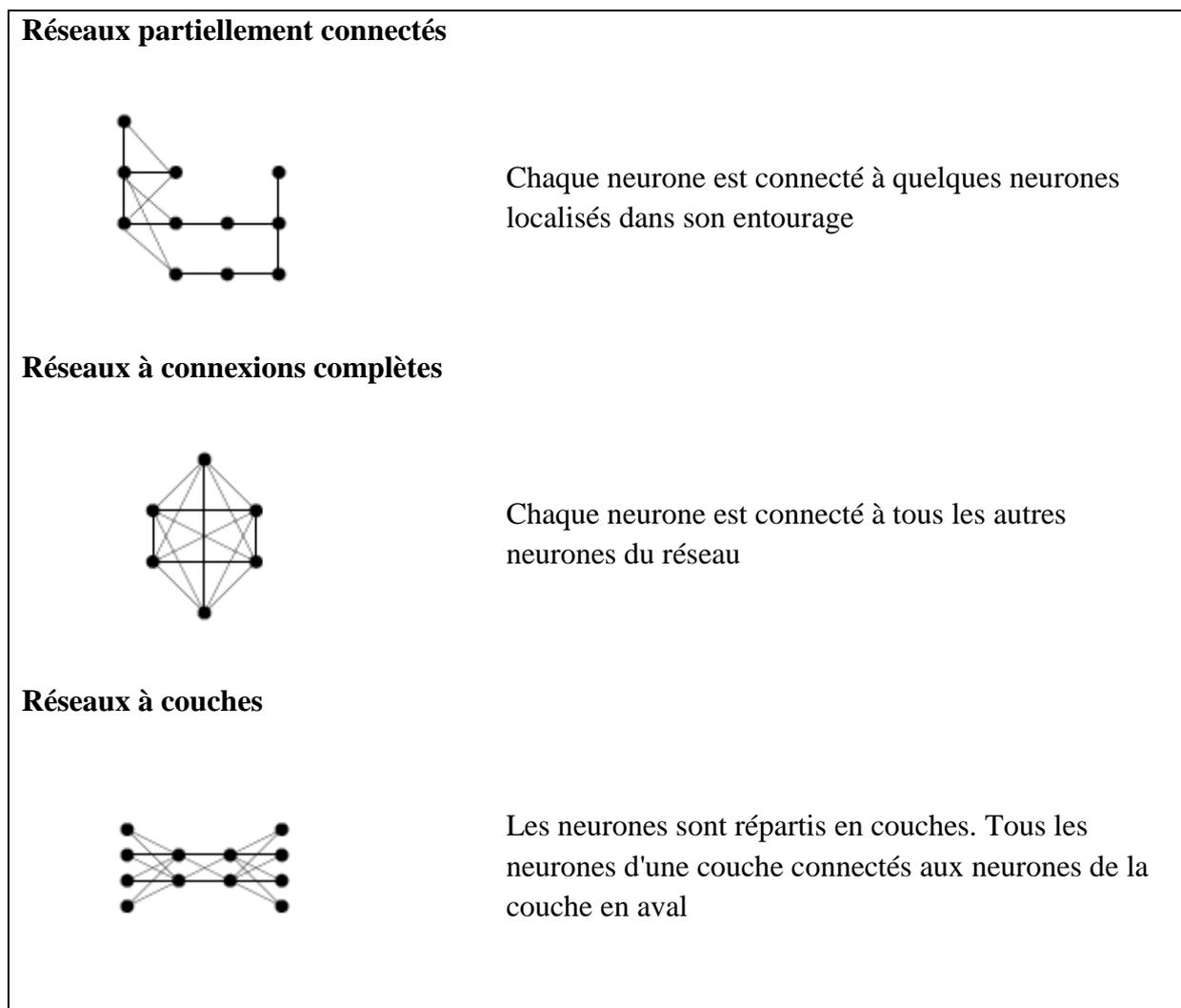
### 2.4. Différentes configurations du réseau

La configuration d'un réseau de neurones peut être quelconque mais quelques schémas typiques sont souvent utilisés, que l'on peut ranger suivant la Figure.4.4.

D'autres réseaux ont été créés suivant des configurations différentes, par exemple les réseaux à couches et à connexion locales ou encore à connexions récurrentes (où il existe des connexions qui ramènent l'information en arrière).

Notons que pour un problème donné, le choix de la configuration d'un réseau de neurones, du nombre optima de neurones qui le constituent, n'est pas trivial. Nous pouvons insister sur le fait que toute l'information contenue dans un réseau de neurones réside dans:

- La configuration du réseau (le nombre de neurones, le nombre de liens entre les neurones, le nombre de couches différentes, etc...).
- La force (définie par les nombres  $w_i$ ) des connexions qui lient les neurones entre eux, en leur permettant de transmettre des signaux avec une plus ou moins grande importance, et avec une action inhibitrice ou excitatrice.
- La loi qui détermine les réactions des neurones en fonction des informations qu'ils reçoivent (la fonction d'activation).



**Figure.4.3.** Différentes configurations de réseaux

Le problème de la détermination de la configuration optimale d'un réseau a longtemps constitué une question ouverte, mais il existe actuellement diverses méthodes basées sur des tests statistiques pour y répondre [71].

Une fois l'architecture du réseau et la fonction d'activation des neurones déterminés, il reste à fixer les valeurs des pondérations des liens qui les relient. C'est le but de ce qu'on appelle l'apprentissage.

## 2.5. Apprentissage

L'apprentissage d'un réseau de neurones formels consiste à déterminer les poids  $w_i$  optimaux suivant le problème à résoudre. Rappelons que l'architecture du réseau est déjà déterminée à ce stade. On distingue deux types de réseaux de neurones en fonction du type d'apprentissage auquel ils sont soumis:

- Les réseaux à apprentissage supervisé, qui sont généralement destinés à reproduire un processus quelconque (chimique, mécanique, financier ...) dont on connaît seulement quelques variables et les résultats correspondants.
- Les réseaux à apprentissage non supervisé, utilisés par exemple en classification lorsque les classes auxquelles doivent appartenir les données ne sont pas connues a priori.

Les données servant à l'apprentissage sont donc différentes dans les deux cas, puisqu'elles consistent en une série de couples (entrée; sortie désirée correspondante) pour les apprentissages supervisés, tandis que pour les apprentissages non supervisés, les données sont uniquement des entrées.

## 2.6. Les réseaux de neurones à apprentissage supervisé

L'apprentissage supervisé d'un réseau de neurones consiste à modifier les poids tant que la réponse correspondant à chaque entrée n'est pas assez proche de la réponse souhaitée.

### 2.6.1. Le perceptron:

Le perceptron peut être considéré comme le premier des réseaux de neurones. Il fut mis au point dans les années cinquante par Rosenblatt [72]. Comme son nom l'indique, le perceptron se présentait comme un modèle de l'activité perceptive. Le but du perceptron est d'associer des configurations présentées en entrée à des réponses. Le perceptron se compose, pour l'essentiel, de deux couches de neurones (Figure.4.5). La première couche était appelée, à l'origine, la rétine du perceptron. La deuxième couche donne la réponse du perceptron correspondant à la stimulation donnée en entrée.

Par exemple, imaginons un perceptron composé d'une rétine de  $10 \times 10 = 100$  cellules et d'une couche de sortie composée de 26 cellules. La tâche du perceptron sera de reconnaître les lettres présentées sur la rétine. Chaque lettre devra être associée à une cellule de sortie. Ainsi, lorsque la lettre A est présentée sur la rétine, la première cellule de sortie doit être active, et toutes les autres passives. Lorsque la lettre E est présentée sur la rétine, la cinquième cellule de sortie doit être active, et toutes les autres passives, etc.

Les cellules de la première couche répondent en *oui/non*. La réponse oui correspond à une valeur 1 en sortie pour le neurone. La réponse non correspond à une valeur 0 en sortie pour le neurone. Les cellules d'entrée sont reliées aux cellules de sortie grâce à des synapses. Les

cellules de sortie évaluent l'intensité de la stimulation en provenance des cellules de la rétine en effectuant la somme des intensités des cellules actives. Avec une formule:

$$a_j = \sum_i^I x_i w_{ij} \quad (47)$$

Avec

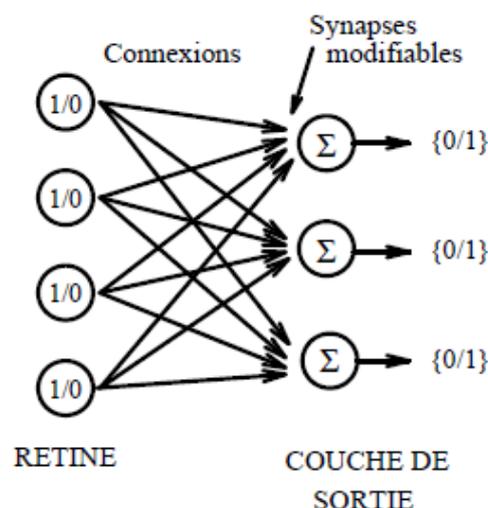
- $a_j$ : activation de la  $j^{ieme}$  cellule de sortie.
- $x_i$ : valeur de sortie (0 ou 1) de la  $i^{ieme}$  cellule de la rétine.
- $w_{ij}$  : intensité de la connexion entre la  $i^{ieme}$  cellule d'entrée et la  $j^{ieme}$  cellule de sortie.

Les cellules de sortie deviennent actives si leur degré d'activation dépasse un seuil fixé noté  $\theta_j$ . Le plus souvent, le seuil  $\theta_j$  est fixé à zéro, mais sans que cela soit nécessaire. Avec une formule:

$$o_j = \begin{cases} 0 & \text{pour } a_j \leq \theta_j \\ 1 & \text{pour } a_j > \theta_j \end{cases} \quad (48)$$

Un des premiers perceptrons mis au point par Rosenblatt s'appelait Mark 1. Il comportait une rétine de  $20 \times 20 = 400$  cellules. Les synapses étaient représentées par des séries de potentiomètres. Mark 1 occupait l'équivalent d'un appartement de deux pièces.

Le principal problème pour le perceptron est d'arriver à trouver un ensemble de valeurs pour les synapses tel que les configurations d'entrée se traduisent par les réponses voulues. On peut envisager plusieurs règles d'apprentissage, la plus connue porte le nom de Widrow-Hoff [71]. Pour apprendre, le perceptron doit savoir qu'il a commis une erreur, et il doit connaître la réponse qu'il aurait dû donner.



**Figure.4.4.** Exemple de perceptron

La règle d'apprentissage est locale dans le sens que chaque cellule de sortie apprend sans avoir besoin de connaître la réponse des autres cellules. Seule importe l'information qui lui est apportée par les cellules de la rétine. La règle en elle-même est très simple. Tout d'abord, une cellule ne modifie l'intensité de ses synapses que lorsqu'elle se trompe. Si la cellule de sortie est active quand elle devrait être inactive, alors elle diminue l'intensité des synapses correspondant à des cellules actives de la rétine. Cela revient à diminuer l'intensité de la stimulation arrivant à la cellule de sortie. Si la cellule de sortie est inactive quand elle devrait être active, alors elle augmente l'intensité des synapses correspondant à des cellules actives de la rétine. Cela revient à augmenter l'intensité de la stimulation arrivant à la cellule de sortie.

On présente, donc, l'ensemble des stimuli à apprendre au perceptron dans un ordre arbitraire, et on permet au perceptron d'apprendre. Si après une première présentation de l'ensemble des stimuli, le perceptron commet encore des erreurs, on présente de nouveau l'ensemble des stimuli dans un nouvel ordre arbitraire en permettant aux cellules du perceptron d'apprendre. La procédure se poursuit ainsi jusqu'à ce que le perceptron soit capable de donner toutes les réponses correctes.

De manière plus générale, la règle d'apprentissage de Widrow-Hoff s'écrit:

$$w_{ij}^{(t+1)} = w_{ij}^t + \mu(t_j - o_j)x_i = w_{ij}^t + \Delta w_{ij} \quad (49)$$

Avec

- $\Delta w_{ij}$  : changement à effectuer pour la valeur  $w_{ij}$ .
- $x_i$  : valeur de sortie (0 ou 1) de la  $i^{\text{ème}}$  cellule de la rétine.
- $o_j$  : réponse de la  $j^{\text{ème}}$  cellule de sortie (0 ou 1).
- $t_j$  : réponse théorique (ou désiré) de la  $j^{\text{ème}}$  cellule de sortie (0 ou 1).
- $w_{ij}^t$  : intensité de la connexion entre la  $i^{\text{ème}}$  cellule d'entrée et la  $j^{\text{ème}}$  cellule de sortie, au temps  $t$  (les valeurs  $w^0$  sont généralement choisies au hasard).
- $\mu$  : Une constante positive.

On montre que les réseaux monocouches du type Perceptron sont limités aux seuls problèmes linéairement séparables. Cette limitation a été principalement mise en avant dans une publication [69] qui a considérablement freiné les recherches sur les réseaux de neurones, jusqu'à ce que la limitation soit levée avec la mise au point des réseaux à couches.

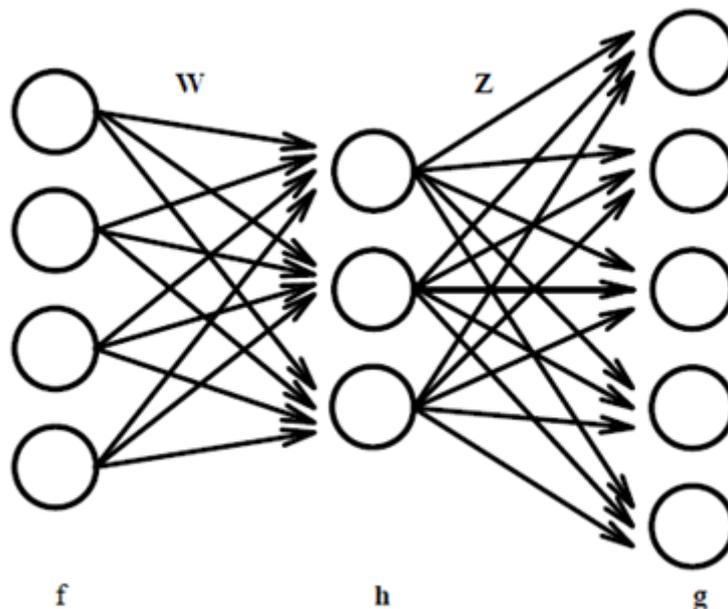
### 2.6.2. Architectures à couches cachées: Rétro-propagation:

Un réseau utilisant la technique de rétro-propagation de l'erreur comporte une couche d'entrée, une couche de sortie, et au moins une couche cachée intermédiaire entre la couche d'entrée et la couche de sortie (Figure.4.6).

L'idée essentielle de la rétro-propagation est remarquablement simple. Les cellules de la couche de sortie calculent l'erreur entre la réponse donnée et la réponse désirée et ajustent l'intensité des connexions en utilisant la règle de Widrow-Hoff. Ensuite, l'erreur est propagée en sens inverse de la couche de sortie à la couche cachée en utilisant les connexions reliant les cellules de la couche cachée à la couche interne. Puis les cellules de la couche cachée

modifient à leur tour l'intensité des connexions les reliant aux cellules de la couche d'entrée de manière à réduire l'erreur.

L'idée essentielle de la rétro-propagation est remarquablement simple. Les cellules de la couche de sortie calculent l'erreur entre la réponse donnée et la réponse désirée et ajustent l'intensité des connexions en utilisant la règle de Widrow-Hoff. Ensuite, l'erreur est propagée en sens inverse de la couche de sortie à la couche cachée en utilisant les connexions reliant les cellules de la couche cachée à la couche interne. Puis les cellules de la couche cachée modifient à leur tour l'intensité des connexions les reliant aux cellules de la couche d'entrée de manière à réduire l'erreur.



**Figure.4.5.** Architecture d'un réseau de neurones avec une couche cachée

De manière plus technique, soit un réseau de neurones avec une couche d'entrée, une couche cachée et une couche de sortie. On note:

- $f_k$  : le vecteur à  $I$  éléments représentant le  $k^{ieme}$  stimulus (la couche d'entrée comporte  $I$  cellules).
- $h_k$  : le vecteur à  $L$  éléments représentant la réponse des  $L$  cellules de la couche cachée lorsque le  $k^{ieme}$  stimulus est présenté en entrée (la couche cachée comporte  $L$  cellules).
- $\hat{g}_k$  : le vecteur à  $J$  éléments représentant la réponse des cellules de la couche de sortie pour le  $k^{ieme}$  stimulus (la couche de sortie comporte  $J$  cellules).
- $g_k$  : le vecteur à  $J$  éléments représentant la réponse désirée (ou théorique) des cellules de la couche de sortie pour le  $k^{ieme}$  stimulus.
- $W$  : la matrice d'ordre  $L*I$  des valeurs des connexions reliant les cellules de la couche d'entrée aux cellules de la couche cachée;  $w_{li}$  donne la valeur de la connexion entre la  $i^{ieme}$  cellule de la couche d'entrée et la  $l^{ieme}$  cellule de la couche cachée.

- $Z$  : la matrice d'ordre  $J*L$  des valeurs des connexions reliant les cellules de la couche d'entrée aux cellules de la couche cachée;  $z_{jl}$  donne la valeur de la connexion entre la  $l^{ieme}$  cellule de la couche cachée et la  $j^{ieme}$  cellule de la couche de sortie.

Pour utiliser la technique de rétro-propagation, la réponse des cellules doit être une fonction non-linéaire de l'état d'activation de la cellule. En notant  $a_n$  l'état d'activation à la cellule  $n$ , la réponse de la cellule (notée  $o_n$ ) sera:

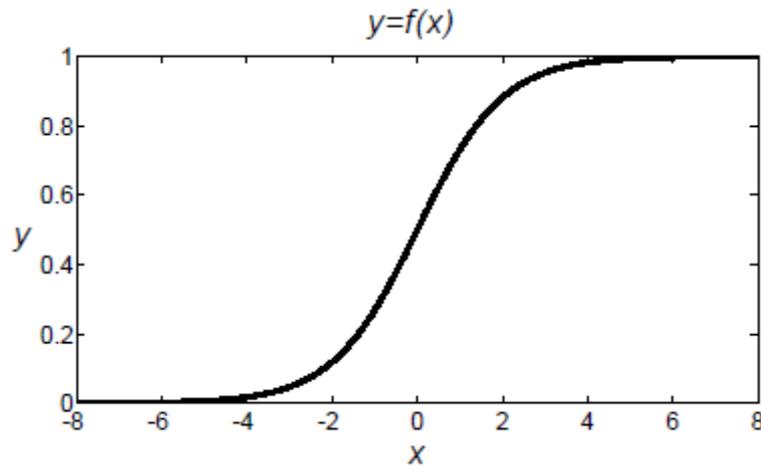
$$o_n = f(a_n) \quad (50)$$

Il existe plusieurs fonctions  $f$  satisfaisantes, une des plus fréquentes et des plus pratiques est la fonction logistique :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (51)$$

Comme le montre la (Figure.4.6), la fonction logistique possède la propriété intéressante de rester dans le domaine  $[0, 1]$ . En outre, sa dérivée, utilisée pour l'apprentissage, est simple à calculer:

$$f'(x) = -\frac{e^{-x}}{(1 + e^{-x})^2} = f(x)[1 - f(x)] \quad (52)$$



**Figure.4.6.** Graphe de la fonction logistique  $y=f(x)$

Pour répondre à un stimulus, le signal est propagé de la couche d'entrée à la couche de sortie en passant par la couche cachée. A chaque étape, la réponse des cellules s'obtient par la fonction logistique. On peut évidemment concevoir des réseaux avec des fonctions de réponse différentes pour chaque couche, mais l'analyse en sera rendue plus difficile. Ainsi lorsque le  $k^{ieme}$  stimulus est présenté en entrée, le vecteur de réponse des cellules de la couche cachée est donné par:

$$h_k = f(Wf_k) \quad (53)$$

Puis la réponse de la cellule de sortie est donnée par le vecteur:

$$\hat{g}_k = f(Zh_k) \quad (54)$$

La technique de rétro-propagation est une technique d'apprentissage supervisée, elle modifie l'intensité des connexions de manière à diminuer l'intensité de l'erreur commise par la cellule pour la réponse considérée. La procédure de prise en compte de l'erreur est la même pour toutes les couches, mais l'estimation du signal d'erreur diffère suivant les couches.

Pour les cellules de la couche de sortie, l'erreur est évaluée en comparant la réponse donnée par la cellule avec la réponse théorique. Le vecteur d'erreur pour le  $k^{ieme}$  stimulus vaut donc:

⊗ Le signal d'erreur prend en compte l'erreur commise par la cellule et l'état d'activation de la cellule. Pour la couche de sortie, il est défini comme:

$$\delta_{sortie,k} = f'(Zh_k) \otimes (e_k) = \hat{g}_k \otimes (1 - \hat{g}_k) \otimes (g_k - \hat{g}_k) \quad (55)$$

avec ⊗ indiquant le produit terme à terme des vecteurs et 1 un vecteur unité.

La procédure d'apprentissage généralise la procédure déjà vue dans le cas linéaire, la matrice de connexions  $Z$  est corrigée par itérations. A l'étape  $t+1$ ,  $Z$  devient:

$$Z_{(t+1)} = Z_{(t)} + \mu \delta_{sortie,k} h_k^T = Z_{(t)} + \Delta_t Z \quad (56)$$

$k$  étant choisi aléatoirement et  $\mu$  étant un nombre réel positif.

Pour les cellules de la couche cachée, le signal d'erreur ne peut être évalué par comparaison avec une valeur idéale. Il est estimé comme une fonction du signal d'erreur en provenance de la couche de sortie et de l'activation des cellules de la couche cachée. Précisément, le vecteur donnant le signal d'erreur pour les cellules de la couche cachée s'obtient comme:

$$\delta_{cache,k} = f'(Wf_k) \otimes (Z^t \delta_{sortie,k}) = h_k \otimes (1 - h_k) \otimes (Z^t \delta_{sortie,k}) \quad (57)$$

Comme on le voit, le signal d'erreur s'obtient en propageant l'erreur de la couche de sortie à la couche cachée, ce qui correspond au sens inverse de la propagation du signal lorsque la mémoire donne une réponse en réponse à une stimulation. L'apprentissage pour la couche cachée se déroule, ensuite, de la manière similaire à celui de la couche de sortie. La matrice de connexions  $W$  est corrigée par itérations. A l'étape  $t+1$ ,  $W$  devient:

$$W_{(t+1)} = W_{(t)} + \mu \delta_{cachée,k} f_k^T = W_{(t)} + \Delta_t W \quad (58)$$

Cette procédure minimise le carré de l'erreur à chaque étape. Elle converge vers un minimum local.

### 2.6.3. Evaluation de la rétro-propagation

La rétro-propagation a probablement été une des causes récentes du regain d'intérêt pour les réseaux de neurones. Une des principales qualités de ces réseaux est de pouvoir

implémenter des classifications de problèmes non-linéaires. En outre, les couches cachées peuvent parfois s'interpréter comme développant une représentation du problème posé, et cette représentation en elle-même peut être objet d'étude.

La rétro-propagation est probablement à la base du plus grand nombre des applications récentes des réseaux de neurones [73], comme par exemple, la reconnaissance de phonèmes, la lecture de textes par ordinateur, la reconnaissance de chiffres écrits, etc.

Néanmoins, la rétro-propagation ne saurait prétendre être une panacée. D'une part, il est difficile de trouver un équivalent neuronal de cette technique. En outre, sa mise en œuvre exige souvent des temps d'apprentissage très long qui rendent son application prohibitive pour de nombreux problèmes de taille raisonnable. Enfin, cette technique ne peut s'appliquer que pour des problèmes de classification où les bonnes réponses sont connues ; elle ne s'applique que pour des apprentissages supervisés.

## **2.7. Champs d'utilisation**

Les réseaux multicouches ont été développés pour diverses applications. Par exemple, il existe un système de prédiction conçu pour détecter les anomalies de fonctionnement d'une colonne à distiller, par comparaison en temps réel des prédictions du modèle et les données effectivement mesurées.

Un autre système est celui qui a été développé pour la société Michelin, pour la prédiction des propriétés physiques des caoutchouc pour pneumatiques.

Les réseaux de ce type sont aussi utilisés pour la reconnaissance de formes. On peut citer par exemple un système de lecture automatique du montant écrit en toutes lettres sur les chèques, ou encore les systèmes de tri adoptés par la Poste qui font appel à des réseaux de neurones pour reconnaître les codes postaux.

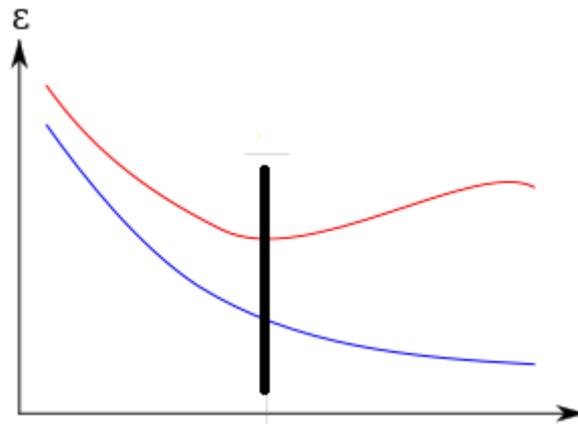
## **2.8. Surapprentissage:**

Le surapprentissage est un problème pouvant survenir dans les méthodes mathématiques et informatiques de classification comme les réseaux de neurones. Une structure dans une situation de surapprentissage aura de la peine à généraliser les caractéristiques des données. Elle se comporte alors comme une table contenant tous les échantillons utilisés lors de l'apprentissage et perd ses pouvoirs de prédiction sur de nouveaux échantillons.

Pour limiter ce genre de problèmes dans le cas des réseaux de neurones, on doit veiller à séparer les données en deux sous-ensembles : l'ensemble d'apprentissage et l'ensemble de validation. L'ensemble d'apprentissage comme son nom l'indique permet de faire évoluer les poids du réseau de neurones avec par exemple une rétropropagation. L'ensemble de validation n'est pas utilisé pour l'apprentissage mais permet de vérifier la pertinence du réseau avec des échantillons qu'il ne connaît pas.

On peut vraisemblablement parler de surapprentissage si l'erreur de prédiction du réseau sur l'ensemble d'apprentissage diminue alors que l'erreur sur la validation augmente de

manière significative (Figure.4.7). Cela signifie que le réseau continue à améliorer ses performances sur les échantillons d'apprentissage mais perd son pouvoir de prédiction sur ceux provenant de la validation.



**Figure.4.7.** Surapprentissage dans un apprentissage supervisé.  
En haut, l'erreur sur l'ensemble de validation. En bas, l'erreur d'apprentissage

Pour avoir un réseau qui généralise bien, on arrête l'apprentissage dès que l'on observe cette divergence entre les deux courbes. On peut aussi diminuer la taille du réseau et recommencer l'apprentissage.

### 3. k - Plus Proches Voisins

k - plus proches voisins (k-PPV) est un algorithme de la reconnaissance des formes qui a prouvé son efficacité face au traitement de données textuelles [74]. Dans un contexte de classification d'une nouvelle observation  $x$ , l'idée fondatrice simple est de faire voter les plus proches voisins de cette observation. La classe de  $x$  est déterminée en fonction de la classe majoritaire parmi les  $k$  plus proches voisins de l'observation  $x$ .

#### 3.1. Algorithme 1-PPV

La méthode du plus proche voisin est une méthode non paramétrique où une nouvelle observation est classée dans la classe d'appartenance de l'observation de l'échantillon d'apprentissage qui lui est la plus proche. La détermination de leur similarité est basée sur des mesures de distance.

Formellement, soit  $L$  l'ensemble de données à disposition ou échantillon d'apprentissage :

$$L = \{(y_i, x_i), i = 1, \dots, n_L\} \quad (59)$$

où  $y_i \in \{1, \dots, c\}$  dénote la classe de l'individu  $i$  et le vecteur  $x_i = (x_{i1}, \dots, x_{ip})$  représente les variables prédictives de l'individu  $i$ . La détermination du plus proche voisin est basée sur une fonction distance arbitraire  $d(.,.)$ . La distance euclidienne entre deux individus caractéristiques par  $p$  covariables est définie par:

$$d((x_1, x_2, \dots, x_p), (u_1, u_2, \dots, u_p)) = \sqrt{(x_1 - u_1)^2 + (x_2 - u_2)^2 + \dots + (x_p - u_p)^2} \quad (60)$$

Ainsi, pour une nouvelle observation  $(y, x)$  le plus proche voisin  $(y_{(1)}, x_{(1)})$  dans l'échantillon d'apprentissage est déterminé par:

$$d(x, x_{(1)}) = \min_i(d(x, x_i)) \quad (61)$$

et  $\hat{y} = y_{(1)}$ , la classe du plus proche voisin est sélectionnée pour la prédiction de  $y$ . Les notations  $x_{(j)}$  et  $y_{(j)}$  représentent respectivement le  $j^{\text{ème}}$  plus proche voisin de  $x$  et sa classe d'appartenance.

Parmi les fonctions distance types, la distance euclidienne est définie comme suit:

$$d(x_i, x_j) = \left( \sum_{s=1}^p (x_{is} - x_{js})^2 \right)^{\frac{1}{2}} \quad (62)$$

et plus généralement la distance de Minkowski:

$$d(x_i, x_j) = \left( \sum_{s=1}^p (x_{is} - x_{js})^q \right)^{\frac{1}{q}} \quad (63)$$

### 3.2. Algorithme k-PPV

Une première extension de cette idée, qui est largement et communément utilisée en pratique, est la méthode des  $k$  plus proches voisins. La plus proche observation n'est plus la seule observation utilisée pour la classification. Nous utilisons désormais les  $k$  plus proches observations. Ainsi la décision est en faveur de la classe majoritairement représentée par les  $k$  voisins. Soit  $k_r$  le nombre d'observations issues du groupe des plus proches voisins appartenant à la classe  $r$ .

$$\sum_{r=1}^c k_r = k \quad (64)$$

Ainsi une nouvelle observation est prédite dans la classe  $l$  avec:

$$l = \max_r(k_r) \quad (65)$$

Ceci évite que la classe prédite ne soit déterminée seulement à partir d'une seule observation. Le degré de localité de cette technique est déterminé par le paramètre  $k$ : pour  $k=1$ , on utilise la méthode du seul plus proche voisin comme technique locale maximale, pour  $k \rightarrow n_1$ , on utilise la classe majoritaire sur l'ensemble intégral des observations (ceci impliquant une prédiction constante pour chaque nouvelle observation à classer).

### 3.3. Quelques règles sur le choix de $k$

Le paramètre  $k$  doit être déterminé par l'utilisateur :  $k \in \mathbb{N}$ . En classification binaire, il est utile de choisir  $k$  impair pour éviter les votes égalitaires. Le meilleur choix de  $k$  dépend du jeu

de donnée. En général, les grandes valeurs de  $k$  réduisent l'effet du bruit sur la classification et donc le risque de sur-apprentissage, mais rendent les frontières entre classes moins distinctes. Il convient donc de faire un choix de compromis entre la variabilité associée à une faible valeur de  $k$  contre un 'oversmoothing' ou surlissage (i.e gommage des détails) pour une forte valeur de  $k$ . Un bon  $k$  peut être sélectionné par diverses techniques heuristiques, par exemple, de validation-croisée. Nous choisirons la valeur de  $k$  qui minimise l'erreur de classification.

#### 4. Bayes Naïve

La classification naïve bayésienne est un type de classification Bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Elle met en œuvre un classifieur bayésien naïf, ou classifieur naïf de Bayes.

En termes simples, un classifieur bayésien naïf suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques. Un fruit peut être considéré comme une pomme s'il est rouge, arrondi, et fait une dizaine de centimètres. Même si ces caractéristiques sont liées dans la réalité, un classifieur bayésien naïf déterminera que le fruit est une pomme en considérant indépendamment ces caractéristiques de couleur, de forme et de taille.

Selon la nature de chaque modèle probabiliste, les classifieurs bayésiens naïfs peuvent être entraînés efficacement dans un contexte d'apprentissage supervisé. Dans beaucoup d'applications pratiques, l'estimation des paramètres pour les modèles bayésiens naïfs repose sur le maximum de vraisemblance. Autrement dit, il est possible de travailler avec le modèle bayésien naïf sans se préoccuper de probabilité bayésienne ou utiliser les méthodes bayésiennes.

Malgré leur modèle de conception naïf et ses hypothèses de base extrêmement simplistes, les classifieurs bayésiens naïfs ont fait preuve d'une efficacité plus que suffisante dans beaucoup de situations réelles complexes. En 2004, un article a montré qu'il existe des raisons théoriques derrière cette efficacité inattendue [75].

L'avantage du classifieur bayésien naïf est qu'il requiert relativement peu de données d'entraînement pour estimer les paramètres nécessaires à la classification, à savoir moyennes et variances des différentes variables. En effet, l'hypothèse d'indépendance des variables permet de se contenter de la variance de chacune d'entre elle pour chaque classe, sans avoir à calculer de matrice de covariance.

##### 4.1. Théorème de Bayes

En théorie des probabilités, le théorème de Bayes énonce des probabilités conditionnelles : étant donné deux événements  $A$  et  $B$ , le théorème de Bayes permet de déterminer la probabilité de  $A$  sachant  $B$ , si l'on connaît les probabilités de  $A$ , de  $B$  et de  $B$  sachant  $A$ .

Ce théorème élémentaire (originellement nommé « de probabilité des causes ») a des applications considérables.

Pour aboutir au théorème de Bayes, on part d'une des définitions de la probabilité conditionnelle :

$$P(A|B)P(B) = P(A \cap B) = P(B|A)P(A) \quad (66)$$

En notant  $P(A \cap B)$  la probabilité que  $A$  et  $B$  aient tous les deux lieu. En divisant de part et d'autre par  $P(B)$ , on obtient:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (67)$$

On améliore parfois le théorème de Bayes en remarquant que:

$$P(B) = P(A \cap B) + P(\bar{A} \cap B) = P(B|A)P(A) + P(B|\bar{A})P(\bar{A}) \quad (68)$$

afin de réécrire le théorème ainsi:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\bar{A})P(\bar{A})} \quad (69)$$

où  $\bar{A}$  est le complémentaire de  $A$ . Plus généralement, si  $\{A_i\}$  est une partition de l'ensemble des possibilités, on obtient:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)} \quad (70)$$

## 4.2. Modèle bayésien naïf

Le modèle probabiliste pour un classifieur est le modèle conditionnel  $P(C|F_1, \dots, F_n)$ , où  $C$  est une variable de classe dépendante dont les instances ou classes sont peu nombreuses, conditionnée par plusieurs variables caractéristiques  $F_1, \dots, F_n$ .

A l'aide du théorème de Bayes, nous écrivons:

$$P(C|F_1, \dots, F_n) = \frac{P(C)P(F_1, \dots, F_n|C)}{P(F_1, \dots, F_n)} \quad (71)$$

En pratique, seul le numérateur nous intéresse, puisque le dénominateur ne dépend pas de  $C$  et les valeurs des caractéristiques  $F_i$  sont données. Le dénominateur est donc en réalité constant. Le numérateur est soumis à la loi de probabilité à plusieurs variables  $p(C, F_1, \dots, F_n)$  et peut être factorisé de la façon suivante, en utilisant plusieurs fois la définition de la probabilité conditionnelle:

$$p(C, F_1, \dots, F_n) = p(C)p(F_1|C)p(F_2|C, F_1)p(F_3|C, F_1, F_2) \dots p(F_n|C, F_1, F_2, \dots, F_{n-1}) \quad (72)$$

C'est là que nous faisons intervenir l'hypothèse naïve: si chaque  $F_i$  est indépendant des autres caractéristiques  $F_{j \neq i}$ , alors  $p(F_i|C, F_j) = p(F_i|C)$  pour tout  $i \neq j$ , par conséquent la probabilité conditionnelle peut s'écrire:

$$p(C, F_1, \dots, F_n) = p(C) \prod_{i=1}^n p(F_i|C) \quad (73)$$

### 4.3. Estimation de la valeur des paramètres

Tous les paramètres du modèle (probabilités a priori des classes et lois de probabilités associées aux différentes caractéristiques) peuvent faire l'objet d'une approximation par rapport aux fréquences relatives des classes et caractéristiques dans l'ensemble des données d'entraînement. Il s'agit d'une estimation du maximum de vraisemblance des probabilités. Les probabilités a priori des classes peuvent par exemple être calculées en se basant sur l'hypothèse que les classes sont équiprobables (chaque antérieure =  $1/(\text{nombre de classes})$ ), ou bien en estimant chaque probabilité de classe sur la base de l'ensemble des données d'entraînement (antérieur de C =  $(\text{nombre d'échantillons de C})/(\text{nombre d'échantillons total})$ ).

Lorsqu'on travaille avec des caractéristiques qui sont des variables aléatoires, on suppose généralement que les lois de probabilités correspondantes sont des lois normales, dont on estimera l'espérance et la variance.

L'espérance,  $\mu$ , se calcule avec:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (74)$$

où N est le nombre d'échantillons et  $x_i$  est la valeur d'un échantillon donné.

La variance  $\sigma^2$  est donnée par:

$$\sigma^2 = \frac{1}{(N-1)} \sum_{i=1}^N (x_i - \mu)^2 \quad (75)$$

En supposant qu'il existe m classes, les probabilités  $P(F_i/C_i)$  pour  $i=1\dots m$ , peuvent être estimées de la façon suivante:

$$P(F_i/C_i) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} \exp\left(-\frac{(x_i - \mu_{ik})^2}{2\sigma_{ik}^2}\right) \quad (76)$$

Si, pour une certaine classe, une certaine caractéristique ne prend jamais une valeur donnée dans l'ensemble de données d'entraînement, alors l'estimation de probabilité basée sur

la fréquence aura pour valeur zéro. Cela pose un problème puisque l'on aboutit à l'apparition d'un facteur nul lorsque les probabilités sont multipliées. Par conséquent, sachant que  $f: x \rightarrow \log(x)$  est une fonction croissante, on corrige les estimations de probabilités en utilisant la formule suivante:

$$\log(P(C, F_1, \dots, F_n)) = \log(P(C)) + \sum_{i=1}^n \log(P(F_i/C)) \quad (77)$$

#### 4.4. Classification à partir d'un modèle de probabilité

Jusqu'à présent nous avons établi le modèle à caractéristiques indépendantes, à savoir le modèle de probabilités bayésien naïf. Le classifieur bayésien naïf couple ce modèle avec une règle de décision. Une règle couramment employée consiste à choisir l'hypothèse la plus probable. Il s'agit de la règle du maximum a posteriori. Le classifieur correspondant à cette règle est la fonction *classifieur* suivante:

$$\text{classifieur}(f_1, \dots, f_n) = \operatorname{argmax}(P(C = c) \prod_{i=1}^n P\left(F_i = \frac{f_i}{C} = c\right)) \quad (78)$$

### 5. Sparse Representation

Dans la littérature, plusieurs techniques de classification sont utilisés pour améliorer la performance du système, l'une des plus récentes est la méthode de Sparse Representation, cette méthode a reçu beaucoup d'attentions ces dernières années, parmi les premières mises en application de cette méthode est sur la reconnaissance du visage [76], après, cette technique a été utilisés dans plusieurs domaines, comme le suivis du trafic routier[77], la reconnaissance de la voie [78], la détection du cancer [79], etc.

#### 5.1. Principe

L'un des problèmes fondamentaux de la reconnaissance optique des objets est d'utiliser les échantillons d'apprentissage issues de  $k$  classes distincts et de déterminer correctement la classe à laquelle un nouvel échantillon de test appartient. Nous organisons les échantillons d'apprentissage de la  $i^{ieme}$  classe sous forme de colonne d'une matrice  $A_i = [v_{i,1}, v_{i,2}, \dots, v_{i,n_i}]$ . La méthode de Sparse Representation suppose que, pour chaque échantillon de test  $y \in R^m$  appartenant à la  $i^{ieme}$  classe, est représenté approximativement de la forme suivante:

$$y = \alpha_{i,1}v_{i,1} + \alpha_{i,2}v_{i,2} + \dots + \alpha_{i,n_i}v_{i,n_i} \quad (79)$$

avec  $\alpha_{i,j} \in R, j = 1, 2, \dots, n_i$ .

Nous définissons ainsi une nouvelle matrice  $A$  des ensembles d'apprentissage par la concaténation des  $n$  ensembles d'apprentissage pour les  $k$  classes:

$$A = [A_1, A_2, \dots, A_k] = [v_{1,1}, v_{1,2}, \dots, v_{k,n_k}] \quad (80)$$

donc la représentation linéaire de  $y$  est comme suit:

$$y = Ax_0 \quad \in R^m \quad (81)$$

avec  $x_0 = [0, \dots, 0, \alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,n_i}, 0, \dots, 0]^T \in R^m$  est un vecteur de coefficients dont les valeurs sont des zéros sauf pour ceux associés à la  $i^{ieme}$ .

Comme les paramètres du vecteur  $x_0$  peuvent être considérés comme représentation de l'échantillon de test  $y \in R^m$ , il est tentant de définir la classe d'appartenance de  $y$  en résolvant le système d'équations  $y = Ax$ , le problème revient à la résolution du système d'équations suivant

$$\widehat{x}_0 = \arg \min \|x\|_0 \quad \text{solution de} \quad Ax = y \quad (82)$$

avec  $\|\cdot\|_0$  désigne la  $l^0$ -norm, qui compte le nombre des éléments non nuls dans le vecteur. Plusieurs travaux de recherche qui s'intéressent à la méthode de Sparse Representation ont démontré que la solution du problème de  $l^0$ -minimisation est équivalent à la résolution du problème de  $l^1$ -minimisation ([80], [81], [82] [83]):

$$\widehat{x}_1 = \arg \min \|x\|_1 \quad \text{solution de} \quad Ax = y \quad (83)$$

## 5.2. Traitement avec un bruit faible

Jusqu'à présent, nous avons supposé le cas parfait où l'échantillon de test  $y$  est représenté exactement par les échantillons d'apprentissage qui correspondent à la classe d'appartenance, cependant, comme les données sont bruitées, l'équation (3) est être modifié comme suit:

$$y = Ax_0 + z \quad (84)$$

avec  $z \in R^m$  représente les termes du bruit avec  $\|z\|_2 < \varepsilon$ , ainsi, la méthode de Sparse Representation revient à la résolution du système d'inéquations linéaires suivant:

$$\widehat{x}_1 = \arg \min \|x\|_1 \quad \text{subject to} \quad \|Ax - y\|_2 < \varepsilon \quad (85)$$

## 5.3. Algorithme

Pour chaque classe  $i$ , on considère la fonction  $\delta_i : R^n \rightarrow R^n$  qui sélectionne les coefficients associés à la  $i^{ieme}$  classe, ainsi, pour  $x \in R^n$ ,  $\delta_i(x) \in R^n$  est un nouveau vecteur dont les éléments sont des zéros sauf pour ceux associés à la  $i^{ieme}$  classe, nous classifions donc  $y$  en fonction de ces approximations en l'affectant à la classe d'objet qui minimise le résidus entre  $y$  et  $\widehat{y}_i = A\delta_i(\widehat{x}_1)$ :

$$\min_i r_i(y) = \|y - A\delta_i(\widehat{x}_1)\|_2 \quad (86)$$

### Algorithme

1) **Input:** la matrice des échantillons d'apprentissage  $A = [A_1, A_2, \dots, A_k] \in R^{m \times n}$  pour les  $k$  classes, un échantillon de test  $y \in R^m$ .

2) Résoudre de problème de  $l^1$ -minimisation  $\widehat{x}_1 = \arg \min_x \|x\|_1$  solution de  $Ax = y$

3) Calculer le résidus  $r_i(y) = \|y - A\delta_i(\widehat{x}_1)\|_2$  for  $i = 1, \dots, k$ .

4) **Output:**  $identify(y) = \operatorname{argmin}_i r_i(y)$

## 6. Conclusion

Dans ce chapitre, on a détaillé les quatre méthodes de classification utilisées dans cette thèse : les réseaux de neurones et les k-plus proches voisins et la méthode de bayes naïve et Sparse Representation. Dans la suite du document, on détaillera les résultats expérimentaux des différentes approches mentionnées dans cette thèse et une comparaison entre les réseaux de neurones, les k-PPV, la méthode de bayes naïve et Sparse Representation.

## Chapitre 5

# Contribution

### 1. Introduction

L'étape de prétraitement est l'une des premières tâches d'un système de reconnaissance après l'étape d'acquisition. Cette étape n'est pas spécifique à la reconnaissance du manuscrit mais fait partie de tout système de reconnaissance de forme visant à améliorer la qualité de l'information pour les étapes de traitements qui vont suivre. Dans le cadre du manuscrit, elle regroupe l'ensemble des processus visant au bon conditionnement du message écrit et qui sont indispensables à son identification.

La squelettisation est un processus important du prétraitement, c'est le nom apporté à la méthode qui consiste à extraire le squelette d'une forme binaire, néanmoins, elle n'est pas indispensable. Plusieurs travaux de recherche ont utilisés la squelettisation dans le processus de reconnaissance des caractères (Figure 5.1.a), d'autres ont choisis d'effectuer le traitement sans passer par la squelettisation (Figure 5.1.b), ce qui rend le chercheur perplexe sur le choix de l'utilisation ou de la non utilisation du squelette, c'est pour cette raison que nous avons proposé une approche qui consiste à effectuer le traitement sur la forme brute et sur le squelette du caractère, la moyenne des deux vecteurs d'extraction est celle utilisée dans la phase de classification (Figure 5.2).

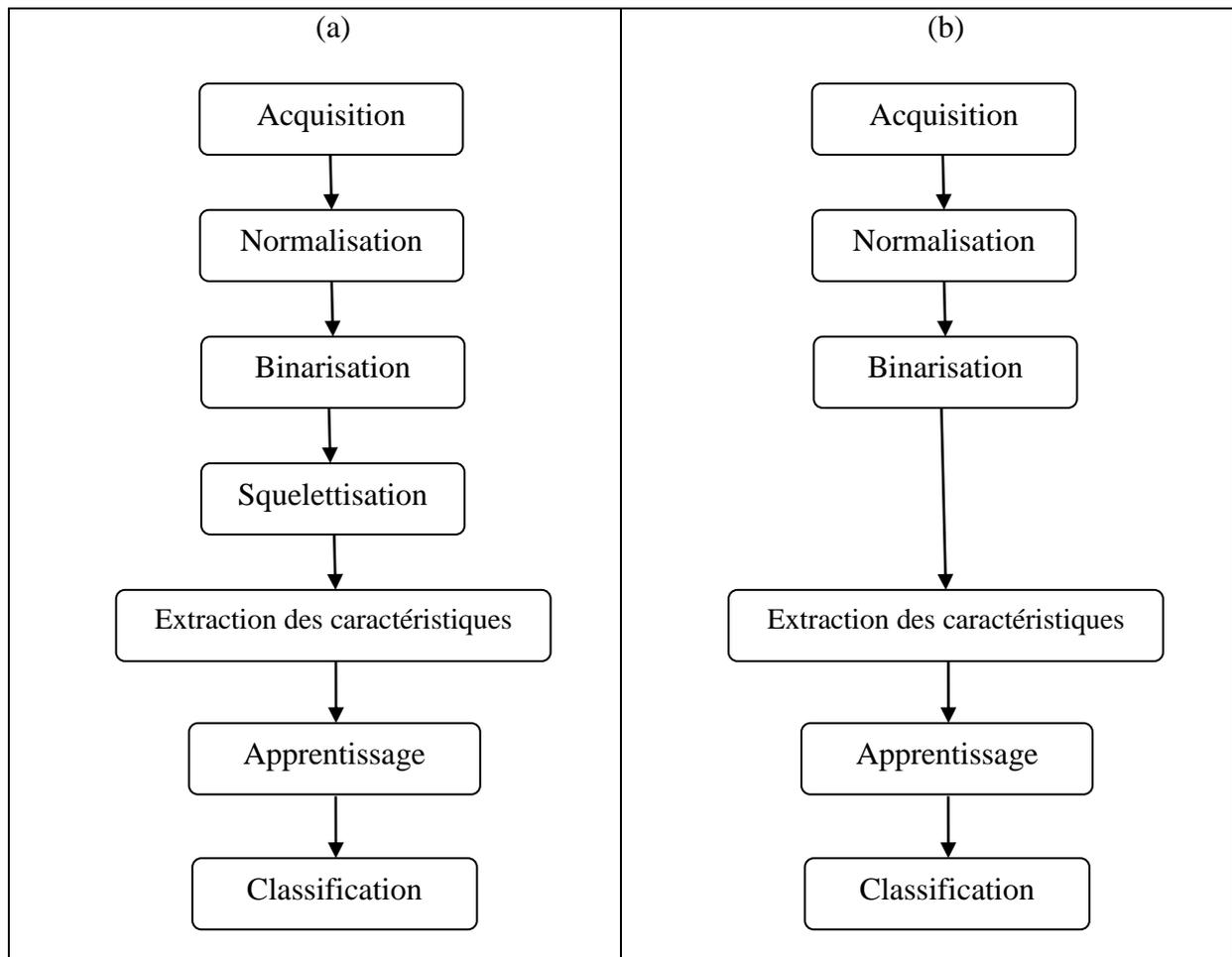
### 2. Prétraitement de l'image

Comme nous l'avons mentionné précédemment, les résultats expérimentaux sont appliqués sur les caractères Gujarati manuscrits et sur la base des caractères MNIST qui est en ligne.

Contrairement aux caractères MNIST qui sont normalisés avec une taille de 28\*28 pixels, la taille de chaque caractère Gujarati dépend de chaque individu, ainsi, une étape de normalisation est indispensable pour la suite du traitement.

Nous avons varié la taille des caractères en fonction de la méthode d'extraction des caractéristiques utilisée, ainsi, pour la méthode de la distance, la taille du caractère est 50\*30 pixels, et pour la méthode de la somme, la taille est 30\*30 pixels.

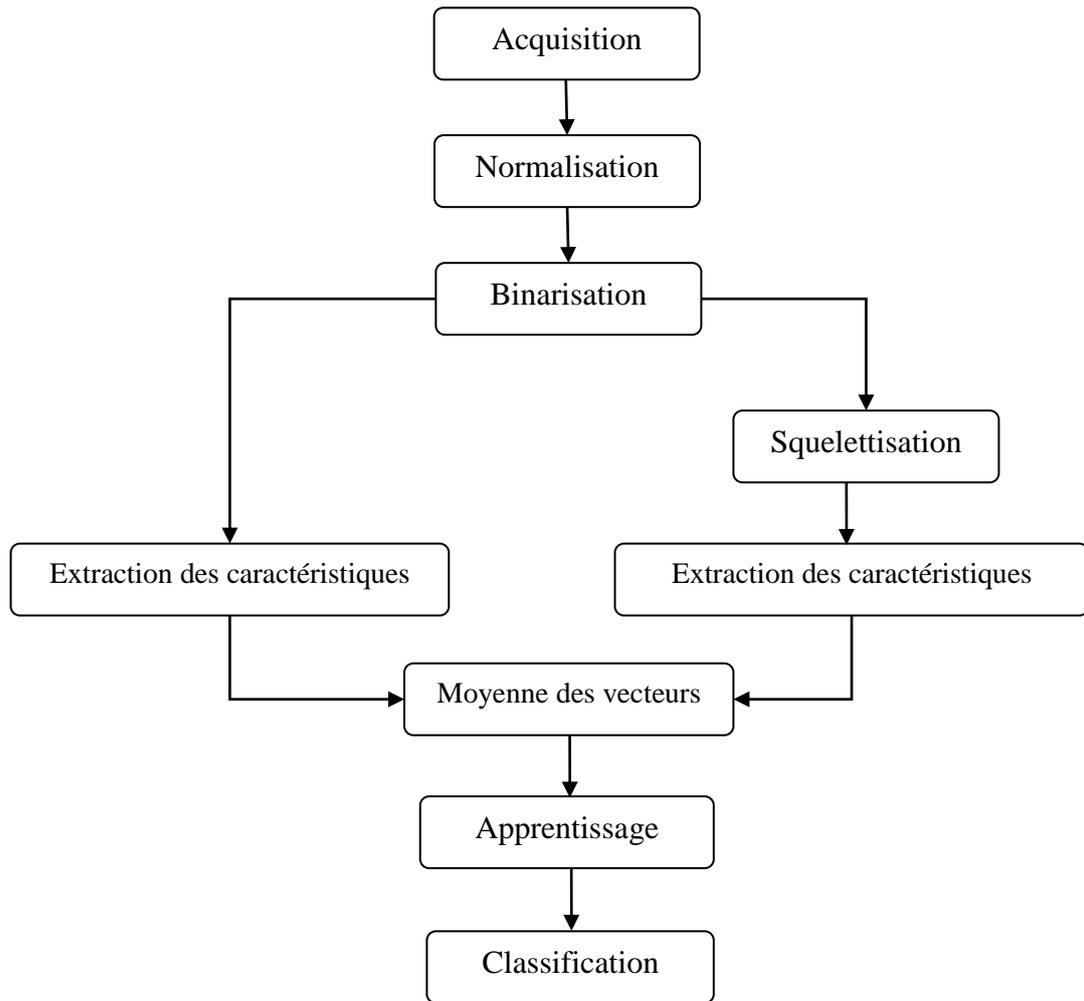
La binarisation et la squelettisation sont les deux étapes les plus importantes dans la phase du prétraitement, une étude comparative a été effectuée pour désigner les algorithmes qui correspondent à notre contexte (voir annexe), ainsi, l'algorithme de Guo\_Hall pour la squelettisation et l'algorithme de Wolf pour la binarisation sont ceux qui ont été choisis. Les figures 5.3 et 5.4 montrent les résultats de l'utilisation de l'algorithme de Wolf sur les caractères Gujarati et MNIST, et les figures 5.5 et 5.6 montrent l'utilisation de l'algorithme de Guo\_Hall.



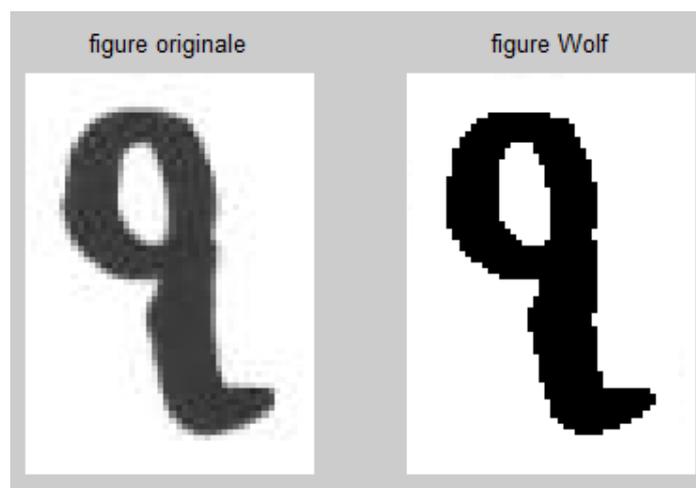
**Figure 5.1.** Processus de Reconnaissance, a: avec squelette, b: sans squelette

### 3. Extraction des Caractéristiques

L'extraction des caractéristiques est la phase la plus importante dans tout processus de reconnaissance des caractères. M.J.Baheti et al [84] ont utilisés les moments invariants, B. El Kessab et al [6] ont choisis d'utiliser l'interpolation mathématique. De notre part, on a choisi d'utiliser les méthodes de la distance et de la somme décrites précédemment.



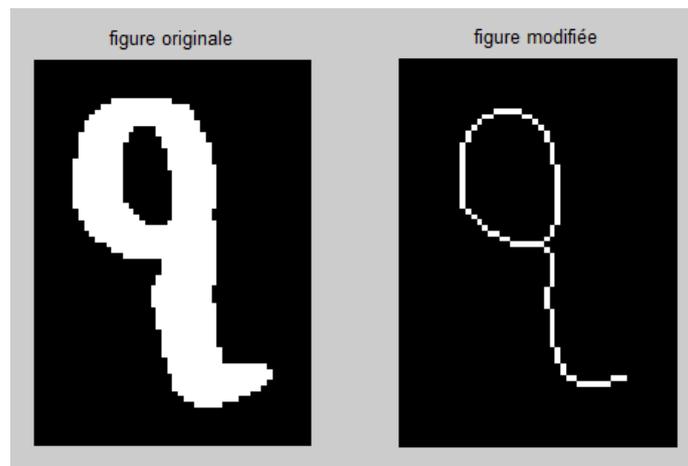
**Figure 5.2.** Contribution



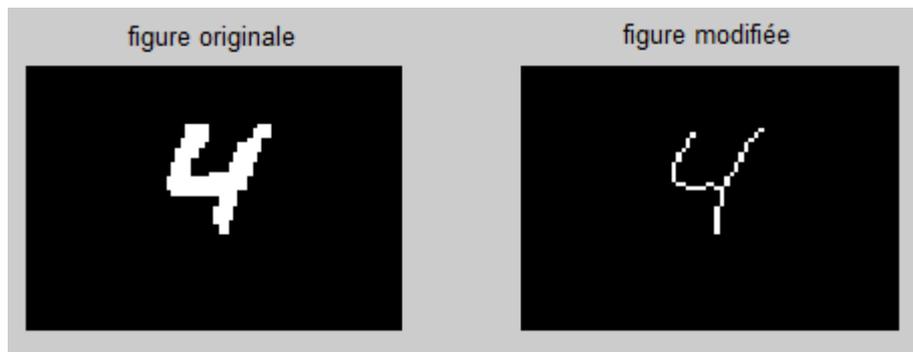
**Figure.5.3.** Application de l'algorithme de Wolf sur le caractère Gujarati



**Figure.5.4.** Application de l'algorithme de Wolf sur le caractère MNIST



**Figure.5.5.** Application de l'algorithme Guo\_Hall sur le caractère Gujarati



**Figure.5.6.** Application de l'algorithme de Guo\_Hall sur le caractère MNIST

Les dimensions du vecteur d'extraction des caractéristiques dépendent de la base des caractères et de la méthode d'extraction utilisée. Concernant la méthode de la somme, si on applique le schéma de la Figure.3.1 sur le caractère MNIST de taille 24\*24 pixels et le caractère Gujarati de taille 30\*30 pixels, on aura deux vecteurs d'extractions de dimensions 142 et 178 respectivement, en effet, pour les caractères MNIST, on a éliminé deux pixels sur les quatre bords de l'image qui ne sont que des pixels noirs, ceci a pour but de diminuer la taille du vecteur.

Pour la méthode de la distance sur les caractères Gujarati, vue que la dimension de chaque image est 50\*30 pixels, nous avons divisé cette dernière en 50 blocs de 5\*6 pixels, en effet, nous avons mis 5 lignes au niveau vertical et 10 ligne au niveau horizontal. Sur les caractères MNIST qui sont de tailles 28\*28 pixels, comme précédemment, on a divisé l'image en 28 blocs de 4\*7 pixels, puis, la même image est divisée une deuxième fois en 28 blocs, mais cette fois-ci, ces derniers ont la taille de 7\*4 pixels, par la suite, les deux vecteurs d'extraction sont rassemblés en un seul pour former un vecteur de 56 éléments.

Le tableau.5.1. résume les dimensions du vecteur d'extraction suivant chaque cas, et le tableau.5.3 donne des échantillons des vecteurs d'extraction.

Base des Caractères	Méthode d'Extraction	Dimension du Vecteur d'Extraction
MNIST	Distance	56
MNIST	Somme	142
Gujarati	Distance	50
Gujarati	Somme	178

**Tableau.5.1.** Dimensions des Vecteurs d'Extraction suivant les cas

#### 4. Réseaux de Neurones

Plusieurs travaux de recherche ont utilisé les réseaux de neurones au niveau de la classification des caractéristiques ([5], [6], [14]). Dans ce mémoire, nous avons choisi de même le perceptron multicouche comme type du réseau pour la classification. La méthode de rétro propagation du gradient est utilisée au niveau de l'apprentissage, la fonction sigmoïde comme fonction d'activation au niveau de la couche d'entrée et cachée,

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (87)$$

avec  $\alpha = 0.1$ , la fonction seuil au niveau de la couche de sortie et nous avons fixé la constante d'apprentissage à  $\gamma = 0,1$ . L'architecture du réseau dépend de chaque cas traité, le tableau.5.2 résume les détails de ces architectures.

Base des caractères	Méthode d'extraction	NB1	NB2	NB3
MNIST	Distance	56	30	10
MNIST	Somme	142	80	10
Gujarati	Distance	50	30	10
Gujarati	Somme	178	50	10

**Tableau.5.2.** Architecture des réseaux de Neurones suivant les cas. NB1: Nombre de neurones de la couche d'entrée; NB2: Nombre de neurones de la couche cachée; NB3: Nombre de neurones de la couche de sortie

Base des Caractères	Méthode d'Extraction	Valeur du vecteur d'extraction														
MNIST	Distance	0	0	0	0	1.3004	0	0	0	0	0	0	0	0	0	
		0	3.5507	1.3580	0	0	0	0	0	0	0.8638	0	0	0	0	
		4.7920	0	0	0	0	0	0	1.8055	0	0	0	0	0	0	
		0	0	0	0	0	0	0	0	0	0	1.9715	0	0	0	0
		0	0	0	2.7686	0	0	0	0.7593	2.2267	0	0	0	0	0	0
		0	3.2751	0	0	0	0	0	2.6693	0	0	0	0	0	0	0
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MNIST	Somme	0	0	0	0	0	1	1	1	1	1	1	1	1	1	
		1	1	1	1	1	1	1	1	0	0	0	0	0	0	
		0	0	0	0	1	2	2	3	2	4	4	0	0	0	
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		0	0	0	0	0	0	0	0	0	1	1	2	2	1	
		2	2	1	2	2	2	0	0	0	0	0	0	0	0	
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		0	0	1	1	1	1	0	1	1	1	1	0	1	2	
		1	1	1	0	1	1	0	1	1	0	1	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0			
Gujarati	Distance	0	0	0	0	0	0	0	0	3.7869	3.0518	0	0	0		
		0	0	3.0293	0	0	0	0	0	3.5101	0	0	0	0		
		0	0	2.6507	1.6151	0	4.5058	0	5.0938	0	1.2459	0	0	0		
		0	4.7798	1.1501	0	5.9658	0	1.2459	0	0	0	0	0	0		
		0	2.6421	4.2454	0	0	3.3885	0	9.5242	0	0	0	0	0		
		6.8275	10.3110	0	0	0	3.4453	0	0	0	0	0	0	0		
Gujarati	Somme	0	0	6	15	18	16	17	17	16	13	13	16	15		
		17	12	5	6	6	5	5	4	5	6	5	6	6		
		0	1	5	8	9	10	12	13	12	10	9	9	9		
		14	17	20	19	16	16	14	12	8	8	6	0	0		
		0	0	0	0	0	0	0	0	2	7	7	8	9		
		12	11	10	5	5	5	5	5	6	6	6	6	7		
		13	12	12	11	8	6	4	4	3	3	4	3	4		
		1	1	0	0	0	0	0	0	0	0	0	0	0		
		0	3	6	7	8	10	11	10	7	7	7	7	7		
		9	8	8	8	10	11	12	17	20	20	16	13	11		
		0	0	0	0	0	0	0	0	0	0	0	0	0		
		0	0	0	0	0	0	0	0	0	0	0	0	0		
		0	0	0	0	0	0	0	0	0	0	0	0	0		
		0	0	0	0	0	0	0	0	0	0	0	0	0		

**Tableau.5.3.** Valeurs des Vecteurs d'Extraction suivant les cas

Pour les caractères de la base MNIST, 1000 caractères sont pris en considération pour l'apprentissage, celui-ci est effectué dans un premier temps sur 10 chiffres, après, nous effectuons l'apprentissage sur 110 caractères, 100 chiffres de la base MNIST en plus des 10 caractères initialement utilisés, les 100 chiffres représentent les 10 classes de la base, 10 chiffres pour chaque classe. De même, nous utilisons encore une fois les 10 chiffres pour un troisième apprentissage en plus de 200 caractères de la base MNIST, 20 caractères pour chaque classe. Après, nous réalisons l'apprentissage sur 300, 500, 800 et 1000 caractères.

De même pour les caractères Gujarati, l'apprentissage est effectué dans un premier temps sur 10 caractères, puis 100 et enfin 300 caractères.

Pour prévenir le sur-apprentissage, un ensemble de caractères de validation est utilisé, ces caractères ont pour but de définir à l’algorithme les meilleures valeurs des poids synaptiques. Les données de validations sont neutres dans la détermination des poids ; elles ne servent qu’à arrêter l’apprentissage à une itération précédent le début du sur-apprentissage. Dans notre cas, nous avons utilisé 200 caractères de validation pour la base MNIST et 100 pour la base Gujarati.

## 5. K-Plus Proches Voisins

L'utilisation des k-plus proches voisins dans la classification des caractères est très répandue, citons par exemple ([84], [85], [86]). C'est une approche très simple et directe. Elle ne nécessite pas d'apprentissage mais simplement le stockage des données d'apprentissage. Son principe est le suivant: une donnée de classe inconnue est comparée à toutes les données stockées. On choisit pour la nouvelle donnée la classe majoritaire parmi les K plus proches voisins au sens d'une distance choisie. Afin de trouver les K plus proches d'une donnée à classer, on choisit la distance euclidienne. Soient deux données représentées par deux vecteurs  $x_i$  et  $x_j$ , la distance entre ces deux données est donnée par:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (88)$$

Pour déterminer le nombre K, un ensemble de caractères de validation est utilisé. De même que pour les réseaux de neurones, on utilise 200 caractères de validation pour la base MNIST et 100 caractères pour la base Gujarati.

## 6. Bayes Naïve

La méthode de bayes naïve est une méthode de classification assez récente et à notre connaissance, peu de travaux ont appliqué cette méthode pour la reconnaissance des caractères, citons par exemple [87] et [88], et plus particulièrement sur les caractères Gujarati.

Son principe consiste à parcourir l’ensemble des caractères de reconnaissance, calculer la probabilité d’appartenance à l’ensemble des classes supposées et par la suite, la classe qui correspond au maximum de probabilité est celle qui est choisie.

De même que les réseaux de neurones et les K-PPV, on choisit 1000 et 300 caractères d’apprentissage, 25000 et 600 caractères de reconnaissance pour les bases MNIST et Gujarati respectivement.

## 7. Sparse Representation

La méthode a reçu beaucoup d'attentions ces dernières années, cette technique a été utilisée dans plusieurs domaines, comme la reconnaissance des visages [76], le suivis des véhicules[77], la reconnaissance de la voie [78], la détection du cancer [79], etc. A notre connaissance, nous sommes les premiers à avoir appliqué cette méthode sur la reconnaissance optique des caractères Gujarati.

En supposant l'organisation des échantillons d'apprentissage de la  $i^{ieme}$  classe sous forme de colonne d'une matrice  $A_i = [v_{i1}, v_{i2}, \dots, v_{in_i}]$ . La méthode de Sparse Representation suppose que, pour chaque échantillon de test  $y \in R^m$  appartenant à la  $i^{ieme}$  classe, est représenté approximativement de la forme suivante:

$$y = \alpha_{i,1}v_{i,1} + \alpha_{i,2}v_{i,2} + \dots + \alpha_{i,n_i}v_{i,n_i} \quad (89)$$

avec  $\alpha_{i,j} \in R, j = 1, 2, \dots, n_i$ .

Après des transformations mathématiques, le problème revient à résoudre un système d'inéquations linéaires avec contraintes comme décrit dans le chapitre précédent, l'application pratique de cette méthode est réalisée sur les caractères Gujarati et MNIST.

## 8. Résultats et Discussions

Nous avons effectué la reconnaissance de trois manières différentes, la première en utilisant la méthode proposée, la deuxième en passant par le squelette du caractère et la troisième sans passer par le squelette, les traitements ont été appliqués pour la base MNIST sur 25000 caractères, soit 2500 caractères pour chaque classe, et 600 caractères pour la base Gujarati, soit 60 caractères pour chaque classe. Les tableaux 5.4, 5.5, 5.6, 5.7 et 5.8 résument les résultats expérimentaux sur les caractères Gujarati et MNIST.

Nombre des caractères	Type de test	Taux de Reconnaissance pour la méthode de la distance (%)			Taux de Reconnaissance pour la méthode de la somme (%)		
		Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
300	Apprentissage	92,67	98,67	94	88,67	100	93
600	Reconnaissance	79,83	82,26	83	76,33	79,67	81,17

**Tableau.5.4.** Taux de Reconnaissance sur les Caractères Gujarati avec les réseaux de neurones

Nombre des caractères	Type de test	Taux de Reconnaissance pour la méthode de la distance (%)			Taux de Reconnaissance pour la méthode de la somme (%)		
		Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
1000	Apprentissage	97,80	98,90	99,00	91,00	97,70	95,10
25000	Reconnaissance	86,24	86,86	87,52	75,43	76,81	77,12

**Tableau.5.5.** Taux de Reconnaissance sur les Caractères MNIST avec les réseaux de neurones

Nombre des caractères	Base des caractères	Taux de Reconnaissance pour la méthode de la distance (%)			Taux de Reconnaissance pour la méthode de la somme (%)		
		Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
600	Gujarati	86,83	78,67	85,83	89,83	91,76	90,33
25000	MNIST	75,26	77,76	78,26	80,98	74,46	81,25

**Tableau.5.6.** Taux de Reconnaissance avec les k-ppv

Nombre des caractères	Base des caractères	Taux de Reconnaissance pour la méthode de la distance (%)			Taux de Reconnaissance pour la méthode de la somme (%)		
		Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
600	Gujarati	85,33	80,5	85,67	87,17	93,50	89,50
25000	MNIST	64,12	64,87	63,91	66,96	81,15	72,88

**Tableau.5.7.** Taux de Reconnaissance avec la méthode de Bayes Naïve

Nombre des caractères	Base des caractères	Taux de Reconnaissance pour la méthode de la distance (%)			Taux de Reconnaissance pour la méthode de la somme (%)		
		Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
600	Gujarati	82,5	80,33	83,17	xxx	xxx	xxx
25000	MNIST	47,18	54,75	47,64	xxx	xxx	xxx

**Tableau.5.8.** Taux de Reconnaissance avec la méthode de Sparse Representation

D'après les tableaux ci-dessus, on constate que la méthode de la distance donne en général de meilleurs résultats par rapport à la méthode de la somme, on remarque aussi l'avantage de l'utilisation de la méthode proposée par rapport au traitement direct sur le caractère, ou bien en passant par son squelette. Nous constatons aussi que l'application de la méthode de Sparse Representation n'aboutit pas à de bons résultats sur la base des caractères MNIST, nous pensons que s'est dû à l'application de la méthode du simplexe pour la résolution d'un système d'inéquations linéaire avec contraintes sur une grande quantité de données.

Les tableaux 5.8 jusqu'à 5.59 donnent plus de détails sur les taux de reconnaissance pour chaque classe sur les caractères Gujarati et MNIST en utilisant les méthodes de la distance et de la somme.

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	100	0	0	1	0	1	0	0	0	0	100
1	0	98	0	1	0	0	0	0	1	0	98
2	0	0	100	5	0	2	0	4	1	0	100
3	0	0	0	98	0	0	1	0	1	0	98
4	0	0	0	0	100	0	0	0	0	0	100
5	0	0	0	0	0	99	0	1	0	0	99
6	5	0	0	0	0	0	92	0	1	2	92
7	0	0	0	0	0	0	0	100	0	0	100
8	2	0	0	4	0	0	0	0	92	2	92
9	0	0	0	1	0	0	0	0	0	99	99

**Tableau.5.9.** Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST d'apprentissage pour la méthode de la distance avec le réseau des neurones

Chiffres	Avec Squelettisation										Succès(%)	
	0	1	2	3	4	5	6	7	8	9		
0	100	0	0	0	0	0	0	0	0	0	0	100
1	0	100	0	0	0	0	0	0	0	0	0	100
2	0	0	99	0	0	1	0	0	0	0	0	99
3	0	0	0	98	1	0	0	0	0	0	1	98
4	0	0	0	0	100	0	0	0	0	0	0	100
5	0	0	0	0	0	100	0	0	0	0	0	100
6	0	0	0	0	0	0	100	0	0	0	0	100
7	0	0	1	0	0	0	0	99	0	0	0	99
8	2	1	0	2	0	0	0	0	95	0	0	95
9	1	0	0	1	0	0	0	0	0	98	0	98

**Tableau.5.10.** Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST d'apprentissage pour la méthode de la distance avec le réseau des neurones

Chiffres	Contribution										Succès(%)	
	0	1	2	3	4	5	6	7	8	9		
0	100	0	0	1	0	1	0	0	0	0	0	100
1	0	100	0	0	0	0	0	0	0	0	0	100
2	0	0	99	0	1	0	0	0	0	0	0	99
3	0	0	0	99	0	1	0	0	0	0	0	99
4	0	0	0	0	100	0	0	0	0	0	0	100
5	0	0	0	0	0	100	0	0	0	0	0	100
6	1	0	0	0	0	0	99	0	0	0	0	99
7	0	0	0	0	1	0	0	99	0	0	0	99
8	0	0	0	0	1	0	0	0	98	1	0	98
9	1	1	0	1	1	0	0	0	0	96	0	96

**Tableau.5.11.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST d'apprentissage pour la méthode de la distance avec le réseau des neurones

Chiffres	Sans Squelettisation										Succès(%)	
	0	1	2	3	4	5	6	7	8	9		
0	2330	0	4	28	0	1	32	2	66	37	0	93.20
1	1	2343	2	55	0	1	20	1	16	61	0	93.72
2	0	2	2077	5	219	59	2	125	11	0	0	83.08
3	72	9	1	1934	1	4	40	2	246	191	0	77.36
4	0	0	28	0	2310	46	2	101	11	2	0	92.40
5	5	0	117	9	147	2025	20	150	26	1	0	81.00
6	100	6	0	10	1	0	2136	25	118	104	0	85.44
7	0	0	21	4	74	89	5	2302	2	3	0	92.08
8	74	22	3	254	0	2	83	1	1962	99	0	78.42
9	58	2	0	90	0	2	19	8	188	2133	0	85.32

**Tableau.5.12.** Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST de test pour la méthode de la distance avec le réseau des neurones

Chiffres	Avec Squelettisation										
	0	1	2	3	4	5	6	7	8	9	Succès(%)
0	2166	3	8	145	0	2	76	8	74	18	86.64
1	0	2360	0	68	0	1	20	1	27	23	94.40
2	22	1	2038	20	163	102	23	127	4	0	81.52
3	39	26	13	2115	1	13	39	28	136	90	84.60
4	2	0	46	1	2363	50	4	33	0	1	94.52
5	3	0	19	51	128	2215	3	66	7	8	88.60
6	57	30	3	29	25	3	2105	74	81	93	84.20
7	1	0	49	3	113	39	2	2289	1	3	91.56
8	29	74	1	240	4	17	95	5	1918	117	76.72
9	56	14	1	80	4	16	35	30	118	2146	85.84

**Tableau.5.13.** Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST de test pour la méthode de la distance avec le réseau des neurones

Chiffres	Contribution										
	0	1	2	3	4	5	6	7	8	9	Succès(%)
0	2243	3	8	27	1	0	90	0	118	10	89.72
1	0	2349	0	72	0	8	17	1	29	24	93.96
2	102	9	2025	3	132	55	30	140	2	2	81.00
3	60	17	2	1884	3	18	71	20	291	134	75.36
4	12	0	45	1	2350	38	20	27	0	7	94.00
5	7	0	37	46	156	2194	3	44	5	8	87.76
6	32	13	0	4	40	0	2269	14	94	34	90.76
7	4	0	29	2	84	56	3	2316	0	6	92.64
8	32	91	1	111	3	8	71	0	2113	70	92.52
9	85	7	1	76	11	14	51	13	105	2137	85.48

**Tableau.5.14.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST de test pour la méthode de la distance avec le réseau des neurones

Chiffres	Sans squelettisation										
	0	1	2	3	4	5	6	7	8	9	Succès(%)
0	96	0	2	1	0	1	0	0	0	0	96
1	0	99	0	0	0	0	0	0	1	0	99
2	2	0	86	5	0	2	0	4	1	0	86
3	1	0	0	90	0	6	1	0	0	2	90
4	0	0	1	0	98	0	1	0	0	0	98
5	0	1	1	2	0	90	1	0	2	3	90
6	2	1	0	0	1	7	88	0	1	0	88
7	2	0	0	2	2	1	0	88	1	4	88
8	0	0	0	4	0	5	0	1	83	7	83
9	0	2	0	4	0	1	0	0	1	92	92

**Tableau.5.15.** Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST d'apprentissage pour la méthode de la somme avec le réseau des neurones

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	100	0	0	0	0	0	0	0	0	0	100
1	0	100	0	0	0	0	0	0	0	0	100
2	0	1	97	0	0	1	1	0	0	0	97
3	0	0	0	96	1	0	1	1	0	1	96
4	0	0	1	0	98	0	1	0	0	0	98
5	0	0	0	0	0	98	1	0	1	0	98
6	0	1	1	0	0	0	97	0	1	0	97
7	0	0	0	0	1	0	0	99	0	0	99
8	0	0	1	2	0	1	1	0	95	0	95
9	1	1	0	0	0	0	0	1	0	97	97

**Tableau.5.16.** Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST d'apprentissage pour la méthode de la somme avec le réseau des neurones

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	98	0	0	0	0	0	1	0	1	0	98
1	0	99	1	0	0	0	0	0	1	0	99
2	1	0	95	0	0	0	1	2	0	1	95
3	0	0	1	96	0	1	1	1	0	0	96
4	0	0	1	0	98	0	0	1	0	0	98
5	2	0	1	3	0	86	3	2	1	2	86
6	0	1	0	1	0	0	97	0	0	1	97
7	0	2	1	0	2	0	0	93	0	2	93
8	1	0	0	1	0	1	0	2	94	1	94
9	2	1	0	1	0	0	1	0	0	95	95

**Tableau.5.17.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST d'apprentissage pour la méthode de la somme avec le réseau des neurones

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	2136	7	122	51	0	55	22	12	17	78	85,44
1	1	2412	16	16	0	4	2	13	32	4	96,48
2	141	40	1729	244	126	34	58	41	61	26	69,19
3	67	22	72	1870	77	178	30	31	63	90	74,80
4	17	12	11	20	2211	21	110	17	17	64	88,44
5	36	114	207	218	76	1452	116	56	171	54	58,08
6	147	93	33	2	280	133	1666	10	82	54	66,64
7	53	25	8	30	108	23	3	2000	44	206	80,00
8	33	81	62	360	64	225	67	20	1384	204	55,66
9	100	11	17	113	63	16	8	99	75	1998	79,92

**Tableau.5.18.** Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST de test pour la méthode de la somme avec le réseau des neurones

Chiffres	Avec Squelettisation										
	0	1	2	3	4	5	6	7	8	9	Succès(%)
0	2163	3	35	72	9	30	97	20	46	25	86,52
1	3	2427	20	11	2	4	9	3	8	13	80,00
2	16	10	1971	61	106	83	44	118	78	13	73,33
3	109	17	114	1618	50	118	137	80	132	125	88,33
4	6	3	25	8	2054	149	63	17	72	103	88,33
5	38	18	91	178	98	1680	151	50	170	26	78,33
6	139	63	15	39	76	83	1799	8	201	77	75,00
7	19	1	36	24	116	46	4	2112	12	130	91,67
8	31	32	51	206	50	221	304	12	1479	114	95,00
9	69	13	17	44	105	66	42	89	156	1899	61,67

**Tableau.5.19.** Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST de test pour la méthode de la somme avec le réseau des neurones

Chiffres	Contribution										
	0	1	2	3	4	5	6	7	8	9	Succès(%)
0	2181	1	25	57	2	7	157	25	27	18	87,24
1	2	2378	24	56	4	7	6	4	9	9	95,16
2	25	11	2039	82	77	21	34	84	101	26	81,56
3	156	9	163	1610	21	39	55	166	126	155	64,40
4	2	6	42	9	2124	12	100	36	27	142	84,96
5	32	10	243	279	64	1404	198	81	166	23	56,16
6	103	44	30	23	136	56	1848	42	88	130	73,92
7	11	5	76	31	185	5	6	2003	26	152	80,12
8	26	13	51	161	75	148	90	69	1703	164	68,12
9	62	20	15	80	147	11	22	66	87	1990	79,60

**Tableau.5.20.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST de test pour la méthode de la somme avec le réseau des neurones

Chiffres	Sans Squelettisation										
	0	1	2	3	4	5	6	7	8	9	Succès(%)
0	30	0	0	0	0	0	0	0	0	0	100
1	0	29	1	0	0	0	0	0	0	0	96,67
2	0	0	29	0	0	1	0	0	0	0	96,67
3	1	0	0	27	0	0	0	2	0	0	90,00
4	0	0	0	0	30	0	0	0	0	0	100
5	0	0	0	0	0	30	0	0	0	0	100
6	0	0	0	1	0	0	28	0	0	1	93,33
7	4	0	0	0	0	0	0	25	0	1	83,33
8	0	0	0	0	1	0	0	0	26	3	86,67
9	3	0	0	0	0	0	0	0	3	24	80,00

**Tableau.5.21.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la distance avec le réseau des neurones

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	29	1	0	0	0	0	0	0	0	0	96,67
1	0	30	0	0	0	0	0	0	0	0	100
2	0	0	30	0	0	0	0	0	0	0	100
3	1	0	0	29	0	0	0	0	0	0	96,67
4	0	0	0	0	29	0	0	0	1	0	96,67
5	0	0	0	0	0	30	0	0	0	0	100
6	0	0	0	0	0	0	30	0	0	0	100
7	0	0	0	0	0	0	0	30	0	0	100
8	0	0	0	0	1	0	0	0	29	0	96,67
9	0	0	0	0	0	0	0	0	0	30	100

**Tableau.5.22.** Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la distance avec le réseau des neurones

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	30	0	0	0	0	0	0	0	0	0	100
1	0	29	1	0	0	0	0	0	0	0	96,67
2	0	0	27	0	0	2	0	0	0	1	90,00
3	1	0	0	27	0	0	0	2	0	0	90,00
4	0	0	0	0	29	0	0	0	1	0	96,67
5	0	0	0	0	0	30	0	0	0	0	100
6	0	0	0	1	0	0	28	0	0	1	93,33
7	0	0	0	0	0	0	0	27	0	3	90,00
8	0	0	0	0	1	0	0	0	26	3	86,67
9	0	0	0	0	0	0	0	0	1	29	96,67

**Tableau.5.23.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati d'apprentissage pour la méthode de la distance avec le réseau des neurones

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	54	0	0	0	0	0	0	4	2	0	90,00
1	0	54	2	0	1	3	0	0	0	0	90,00
2	0	11	41	0	5	2	0	0	1	0	68,33
3	0	0	0	55	0	0	2	3	0	0	91,67
4	0	1	0	0	48	2	0	2	7	0	80,00
5	0	10	6	0	0	44	0	0	0	0	73,33
6	9	0	0	3	0	0	45	1	1	1	75,00
7	3	0	0	2	0	0	0	55	0	0	91,67
8	0	0	0	0	2	0	0	0	52	6	86,67
9	0	0	0	0	3	0	0	0	26	31	51,67

**Tableau.5.24.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati de test pour la méthode de la distance avec le réseau des neurones

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	54	0	0	1	0	0	1	0	3	1	90,00
1	0	53	1	0	2	4	0	0	0	0	88,33
2	0	11	44	1	4	0	0	0	0	0	73,33
3	0	0	0	57	0	0	0	3	0	0	95,00
4	0	3	0	3	47	0	0	0	5	2	78,33
5	0	11	6	1	0	41	1	0	0	0	68,33
6	0	0	0	9	0	2	46	3	0	0	76,67
7	2	0	0	1	1	0	1	54	0	1	90,00
8	0	0	0	0	2	0	0	0	56	2	93,33
9	0	1	0	1	0	0	0	0	14	44	73,33

**Tableau.5.25.** Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati de test pour la méthode de la distance avec le réseau des neurones

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	52	0	0	0	0	0	0	4	3	1	86,67
1	0	53	2	0	2	3	0	0	0	0	88,33
2	0	8	43	0	5	2	0	0	1	1	71,67
3	0	0	0	55	0	0	1	4	0	0	91,67
4	0	1	0	0	49	0	0	1	3	6	81,67
5	0	11	5	0	0	42	2	0	0	0	70,00
6	1	0	0	5	0	0	48	0	0	6	80,00
7	1	0	0	2	0	0	0	57	0	0	95,00
8	0	0	0	0	1	0	0	0	58	1	96,67
9	0	0	0	0	2	0	0	0	17	41	68,33

**Tableau.5.26.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati de test pour la méthode de la distance avec le réseau des neurones

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	29	0	0	0	0	0	0	0	0	1	96,67
1	0	30	0	0	0	0	0	0	0	0	100
2	0	2	26	2	0	0	0	0	0	0	86,67
3	0	2	0	28	0	0	0	0	0	0	93,33
4	0	0	0	0	30	0	0	0	0	0	100
5	0	0	0	0	1	28	1	0	0	0	93,33
6	0	1	0	0	0	3	26	0	0	0	86,67
7	0	0	0	0	0	9	0	21	0	0	70,00
8	1	0	0	0	1	0	0	0	28	0	93,33
9	0	0	0	0	0	8	0	0	2	20	66,67

**Tableau.5.27.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la somme avec le réseau des neurones

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	30	0	0	0	0	0	0	0	0	0	100
1	0	30	0	0	0	0	0	0	0	0	100
2	0	0	30	0	0	0	0	0	0	0	100
3	1	0	0	30	0	0	0	0	0	0	100
4	0	0	0	0	30	0	0	0	0	0	100
5	0	0	0	0	0	30	0	0	0	0	100
6	0	0	0	0	0	0	30	0	0	0	100
7	0	0	0	0	0	0	0	30	0	0	100
8	0	0	0	0	0	0	0	0	30	0	100
9	0	0	0	0	0	0	0	0	0	30	100

**Tableau.5.28.** Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la somme avec le réseau des neurones

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	30	0	0	0	0	0	0	0	0	0	100
1	0	30	0	0	0	0	0	0	0	0	96,67
2	0	2	26	2	0	0	0	0	0	0	90,00
3	0	2	0	28	0	0	0	0	0	0	90,00
4	0	0	0	0	30	0	0	0	0	0	96,67
5	0	0	0	0	1	28	1	0	0	0	100
6	0	0	0	0	0	2	28	0	0	0	93,33
7	0	0	0	0	0	8	0	21	1	0	90,00
8	0	0	0	0	0	0	0	0	30	0	86,67
9	0	0	0	0	0	0	0	0	2	28	96,67

**Tableau.5.29.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la somme avec le réseau des neurones

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	53	0	1	0	0	0	0	1	2	3	88,33
1	0	50	4	0	0	0	1	2	0	3	83,33
2	0	3	47	6	0	0	2	2	0	0	78,33
3	0	10	0	50	0	0	0	0	0	0	83,33
4	0	0	0	0	48	0	0	8	3	1	80,00
5	0	5	0	1	5	30	4	6	0	9	50,00
6	0	10	0	7	1	2	36	0	0	4	60,00
7	0	0	0	0	1	0	0	58	1	0	96,67
8	0	0	0	0	0	0	0	4	56	0	93,33
9	2	0	0	0	1	0	0	16	10	30	50,00

**Tableau.5.30.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati de test pour la méthode de la somme avec le réseau des neurones

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	39	0	0	0	0	0	0	17	1	3	65,00
1	0	48	6	2	0	3	0	1	0	0	80,00
2	0	2	44	6	0	1	5	2	0	0	73,33
3	0	1	3	53	0	0	2	1	0	0	88,33
4	0	0	1	0	53	0	0	4	1	1	88,33
5	0	7	1	0	5	47	0	0	0	0	78,33
6	0	0	0	11	0	0	45	4	0	0	75,00
7	0	0	0	3	0	0	2	55	0	0	91,67
8	0	0	0	0	0	0	0	0	57	3	95,00
9	0	1	0	0	1	0	0	10	11	37	61,67

**Tableau.5.31.** Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati de test pour la méthode de la somme avec le réseau des neurones

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	59	0	0	0	0	0	0	0	1	0	98,33
1	0	47	4	0	0	4	3	2	0	0	78,33
2	0	1	50	5	0	0	1	3	0	0	83,33
3	0	3	0	56	0	0	0	1	0	0	93,33
4	0	0	0	0	50	0	0	8	0	2	83,33
5	0	9	0	0	5	30	7	9	0	0	50,00
6	0	4	0	5	1	3	46	0	0	1	76,67
7	0	0	0	1	4	0	0	54	1	0	90,00
8	0	0	0	0	0	0	0	0	60	0	100
9	0	0	0	0	2	0	0	6	17	35	58,33

**Tableau.5.32.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati de test pour la méthode de la somme avec le réseau des neurones

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	58	0	0	0	0	0	0	1	1	0	96,67
1	0	49	5	0	0	6	0	0	0	0	81,67
2	0	11	47	0	1	1	0	0	0	0	78,33
3	0	0	0	60	0	0	0	0	0	0	100
4	0	0	0	1	56	0	0	0	3	0	93,33
5	0	7	4	0	0	49	0	0	0	0	81,67
6	0	1	1	1	1	3	52	0	1	0	86,67
7	3	0	0	3	0	0	0	54	0	0	90
8	0	0	0	0	0	0	0	0	60	0	100
9	0	0	0	0	0	0	0	1	23	36	60

**Tableau.5.33.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati pour la méthode de la distance et les k-ppv

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	55	0	0	0	0	0	0	1	1	3	91,67
1	0	46	4	0	0	9	0	0	1	0	76,67
2	0	9	45	0	0	0	0	0	6	0	75
3	0	0	1	55	0	1	2	0	1	0	91,67
4	0	1	2	1	51	0	0	1	4	0	85
5	0	8	5	0	1	46	0	0	0	0	76,67
6	0	1	4	8	0	3	43	0	0	1	71,67
7	7	0	0	6	0	0	2	45	0	0	75
8	0	0	1	0	0	0	1	0	56	2	93,33
9	0	0	1	0	0	0	0	0	29	30	50

**Tableau.5.34.** Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati pour la méthode de la distance et les k-ppv

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	57	0	0	0	0	0	0	1	1	1	95
1	0	50	3	0	0	7	0	0	0	0	83,33
2	0	10	47	0	2	1	0	0	0	0	78,33
3	0	0	0	60	0	0	0	0	0	0	100
4	0	0	0	1	53	0	0	0	6	0	88,33
5	0	5	5	0	1	49	0	0	0	0	81,67
6	0	1	2	3	0	0	53	0	1	0	88,33
7	3	0	0	4	0	0	0	52	0	1	86,67
8	0	0	0	0	1	0	0	0	58	1	96,67
9	0	0	0	0	0	0	0	0	24	36	60

**Tableau.5.35.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati pour la méthode de la distance et les k-ppv

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	57	0	0	0	0	0	0	0	3	0	95
1	0	51	2	0	0	7	0	0	0	0	85
2	0	0	59	0	0	1	0	0	0	0	98,33
3	0	0	0	58	0	0	1	1	0	0	96,67
4	0	0	0	0	57	1	0	0	1	1	95
5	0	3	2	0	1	54	0	0	0	0	90
6	0	1	0	4	0	0	54	0	0	1	90
7	0	0	0	2	0	0	0	58	0	0	96,67
8	0	0	0	0	0	0	0	0	60	0	100
9	0	0	0	0	0	0	0	0	26	34	56,67

**Tableau.5.36.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati pour la méthode de la somme et les k-ppv

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	60	0	0	0	0	0	0	0	0	0	100
1	0	56	4	0	0	0	0	0	0	0	93,33
2	0	3	55	0	0	0	0	0	2	0	91,67
3	0	0	0	59	0	0	1	0	0	0	98,33
4	0	0	0	0	60	0	0	0	0	0	100
5	0	1	0	0	0	59	0	0	0	0	98,33
6	0	0	0	0	0	0	58	0	0	2	96,67
7	0	0	0	0	0	0	1	59	0	0	98,33
8	0	0	0	0	0	0	0	0	60	0	100
9	0	0	0	0	0	0	0	0	36	24	40

**Tableau.5.37.** Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati pour la méthode de la somme et les k-ppv

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	59	0	0	0	0	0	0	0	1	0	98,33
1	0	52	4	0	0	4	0	0	0	0	86,67
2	0	3	54	1	1	1	0	0	0	0	90
3	0	0	0	59	0	0	1	0	0	0	98,33
4	0	0	0	0	56	1	0	0	2	1	93,33
5	0	3	3	0	0	54	0	0	0	0	90
6	0	1	0	4	0	0	55	0	0	0	91,67
7	0	0	0	3	0	0	0	57	0	0	95
8	0	0	0	0	0	0	0	0	60	0	100
9	0	0	0	0	0	0	0	0	24	36	60

**Tableau.5.38.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati pour la méthode de la somme et les k-ppv

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	2222	15	19	35	11	32	52	5	95	14	88,88
1	1	2463	7	7	0	1	4	11	3	3	98,52
2	50	200	1938	57	91	9	59	55	28	13	77,52
3	44	158	79	1870	27	55	11	107	81	68	74,80
4	8	127	26	1	1793	6	89	56	14	380	71,72
5	52	273	8	354	128	1398	98	56	70	63	55,92
6	47	87	60	2	46	29	2215	0	13	1	88,60
7	2	124	10	3	59	10	4	2100	0	188	84
8	160	443	37	123	87	76	69	56	1352	97	54,08
9	12	100	9	36	447	3	12	400	16	1465	58,60

**Tableau.5.39.** Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST pour la méthode de la distance et les k-ppv

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	2411	14	2	11	6	15	13	20	6	2	96,44
1	0	2470	4	6	0	0	1	19	0	0	98,80
2	109	240	1953	22	40	10	34	68	5	19	78,12
3	75	167	50	1937	15	66	12	107	30	41	77,48
4	32	179	48	3	1848	1	55	91	4	239	73,92
5	54	188	3	358	30	1716	52	45	18	36	68,64
6	91	161	13	4	9	28	2189	0	2	3	87,56
7	13	122	12	3	38	4	0	2250	1	57	90
8	240	242	29	93	148	138	84	51	1381	94	55,24
9	29	120	8	35	426	6	8	571	8	1289	51,56

**Tableau.5.40.** Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST pour la méthode de la distance et les k-ppv

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	2293	14	14	22	14	30	35	10	64	4	91,72
1	0	2466	8	9	0	1	3	9	2	2	98,64
2	64	194	1982	48	58	10	44	61	26	13	79,28
3	44	143	69	1916	22	51	8	113	63	71	76,64
4	9	149	30	0	1887	7	79	38	11	290	75,48
5	25	243	7	346	97	1531	87	63	39	62	61,24
6	45	76	34	2	35	25	2273	0	9	1	90,92
7	2	121	11	3	50	8	0	2185	2	118	87,40
8	171	362	27	108	91	72	56	43	1476	94	59,04
9	16	98	7	34	372	6	8	393	10	1556	62,24

**Tableau.5.41.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST pour la méthode de la distance et les k-ppv

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	2393	36	2	2	1	4	27	0	32	3	95,72
1	3	2471	10	0	0	0	0	3	8	5	98,84
2	63	214	1996	18	42	5	14	76	50	22	79,84
3	39	218	96	1836	8	63	6	109	87	38	73,44
4	12	188	1	0	1951	1	11	13	5	318	78,04
5	53	504	12	176	56	1506	104	19	41	29	60,24
6	28	85	7	1	21	11	2335	0	11	1	93,40
7	3	159	13	11	64	6	3	2086	4	151	83,44
8	82	426	45	64	51	63	46	18	1659	49	66,36
9	17	135	6	24	160	10	3	124	7	2014	80,56

**Tableau.5.42.** Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST pour la méthode de la somme et les k-ppv

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	2457	23	1	2	0	0	10	2	2	3	98,28
1	0	2493	5	0	0	0	0	2	0	0	99,72
2	41	810	1514	6	3	3	13	101	3	6	60,56
3	15	723	82	1283	3	77	7	294	5	11	51,32
4	18	425	1	1	1873	0	8	40	0	134	74,92
5	9	948	14	42	16	1387	47	22	2	13	55,48
6	27	351	5	0	7	12	2097	0	1	0	83,88
7	1	214	5	2	21	1	0	2223	0	33	88,92
8	329	266	24	58	74	69	72	25	1475	108	59
9	30	221	2	23	125	12	0	274	0	1813	72,52

**Tableau.5.43.** Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST pour la méthode de la somme et les k-ppv

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	2412	28	2	2	2	0	27	1	22	4	96,48
1	1	2480	8	2	0	0	0	3	2	4	99,20
2	71	296	1946	15	41	7	12	67	33	12	77,84
3	35	304	113	1750	12	74	8	101	65	38	70
4	11	199	4	0	1990	0	8	12	3	273	79,60
5	20	575	14	146	30	1556	87	16	28	28	62,24
6	33	94	8	1	13	9	2336	0	5	1	93,44
7	6	161	13	9	48	3	3	2146	1	110	85,84
8	76	378	42	48	58	50	54	16	1717	61	68,68
9	16	146	1	31	173	14	3	132	4	1980	79,20

**Tableau.5.44.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST pour la méthode de la somme et les k-ppv

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	55	0	0	0	0	0	0	3	0	2	91,67
1	0	54	1	0	0	5	0	0	0	0	90
2	0	8	47	1	1	0	0	2	1	0	78,33
3	0	1	0	59	0	0	0	0	0	0	98,33
4	0	3	0	0	53	0	0	1	2	1	88,33
5	0	5	5	0	2	48	0	0	0	0	80
6	0	3	1	2	0	0	49	3	0	2	81,67
7	0	1	0	4	0	0	0	55	0	0	91,67
8	0	0	0	0	1	0	0	0	59	0	98,33
9	0	0	0	0	1	0	0	9	17	33	55

**Tableau.5.45.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati pour la méthode de la distance et la classification naïve bayésienne

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	56	0	0	0	0	0	0	2	2	0	93,33
1	0	42	2	0	0	14	2	0	0	0	70
2	0	12	43	0	2	0	1	1	1	0	71,67
3	0	1	0	58	0	0	1	0	0	0	96,67
4	0	3	1	1	49	1	0	1	4	0	81,67
5	0	5	3	0	1	51	0	0	0	0	85
6	0	2	0	3	0	3	50	0	1	1	83,33
7	2	0	1	4	0	0	0	53	0	0	88,33
8	2	0	0	0	2	0	1	0	54	1	90
9	0	0	2	0	1	0	0	8	22	27	45

**Tableau.5.46.** Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati pour la méthode de la distance et la classification naïve bayésienne

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	56	0	0	0	0	0	0	3	0	1	93,33
1	0	51	2	0	0	7	0	0	0	0	85
2	0	4	52	0	0	0	0	2	2	0	86,67
3	0	1	0	58	0	0	1	0	0	0	96,67
4	0	2	0	0	55	0	0	1	1	1	91,67
5	0	7	4	1	2	46	0	0	0	0	76,67
6	0	3	1	1	0	0	51	3	0	1	85
7	0	1	0	4	0	0	0	55	0	0	91,67
8	1	0	0	0	3	0	0	0	56	0	93,33
9	0	0	0	0	2	0	0	13	11	34	56,67

**Tableau.5.47.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati pour la méthode de la distance et la classification naïve bayésienne

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	51	0	1	0	0	0	0	6	2	0	85
1	0	47	4	0	0	9	0	0	0	0	78,33
2	0	1	55	0	0	2	0	2	0	0	91,67
3	0	0	2	49	0	1	2	6	0	0	81,67
4	0	0	0	0	55	2	0	2	0	1	91,67
5	0	3	4	0	2	51	0	0	0	0	85
6	0	3	0	0	0	0	46	6	0	5	76,67
7	0	2	0	0	0	0	0	58	0	0	96,67
8	0	0	0	0	0	0	0	0	59	1	98,33
9	1	0	0	0	0	0	0	3	4	52	86,67

**Tableau.5.48.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati pour la méthode de la somme et la classification naïve bayésienne

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	59	0	0	0	0	0	0	1	0	0	98,33
1	0	59	1	0	0	0	0	0	0	0	98,33
2	0	6	52	0	0	1	0	1	0	0	86,67
3	0	0	0	57	0	0	2	1	0	0	95
4	0	0	0	0	58	1	0	0	0	1	96,67
5	0	2	0	0	0	58	0	0	0	0	96,67
6	0	1	0	2	0	0	53	4	0	0	88,33
7	0	0	0	0	0	0	0	60	0	0	100
8	0	0	0	0	0	0	0	0	58	2	96,67
9	0	0	0	0	0	0	1	10	2	47	78,33

**Tableau.5.49.** Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati pour la méthode de la somme et la classification naïve bayésienne

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	51	0	1	0	0	0	0	6	1	1	85
1	0	51	4	0	0	5	0	0	0	0	85
2	0	2	54	0	0	2	0	2	0	0	90
3	0	0	1	52	0	0	3	4	0	0	86,67
4	0	0	0	0	56	1	0	1	0	2	93,33
5	0	3	3	0	2	52	0	0	0	0	86,67
6	0	3	1	1	0	0	49	5	0	1	81,67
7	0	0	0	1	0	0	0	59	0	0	98,33
8	0	0	0	0	0	0	0	0	59	1	98,33
9	0	0	0	0	0	0	0	3	3	54	90

**Tableau.5.50.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati pour la méthode de la somme et la classification naïve bayésienne

Chiffres	Sans Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	1724	15	39	13	52	256	77	2	310	12	68,96
1	0	2236	120	10	3	20	6	35	45	25	89,44
2	100	68	1671	121	95	67	204	53	45	25	66,84
3	69	103	308	1296	43	302	16	103	189	71	51,84
4	57	79	20	1	1255	113	165	144	162	504	50,20
5	94	191	50	193	188	1450	49	100	121	64	58
6	131	292	69	0	28	81	1817	1	69	12	72,68
7	9	126	22	25	95	31	16	1972	87	117	78,88
8	117	165	64	103	33	380	33	43	1474	88	58,96
9	36	132	5	26	330	82	21	480	252	1136	45,44

**Tableau.5.51.** Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST pour la méthode de la distance et la classification naïve bayésienne

Chiffres	Avec Squelettisation										
	0	1	2	3	4	5	6	7	8	9	Succès(%)
0	1364	4	12	12	18	317	80	58	540	95	54,56
1	0	2230	162	13	7	16	4	27	5	36	89,20
2	35	54	1784	44	74	68	155	144	84	58	71,36
3	31	77	260	1404	19	188	10	225	128	158	56,16
4	34	23	57	0	948	178	258	118	229	655	37,92
5	56	144	33	143	62	1421	56	240	87	258	56,84
6	102	188	27	1	15	122	1934	0	94	17	77,36
7	10	40	34	3	9	9	4	2233	18	140	89,32
8	51	128	73	25	18	288	73	73	1578	193	63,12
9	17	51	11	21	109	46	22	756	146	1321	52,84

**Tableau.5.52.** Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST pour la méthode de la distance et la classification naïve bayésienne

Chiffres	Contribution										
	0	1	2	3	4	5	6	7	8	9	Succès(%)
0	1500	4	38	12	43	342	111	10	425	15	60
1	0	2032	327	13	3	18	7	67	16	17	81,28
2	42	48	1820	49	65	58	227	80	89	22	72,80
3	27	70	310	1356	46	265	15	169	151	91	54,24
4	27	42	58	0	1094	89	244	223	139	584	43,76
5	39	152	41	201	198	1414	58	227	81	89	56,56
6	84	201	64	2	22	112	1921	2	74	18	76,84
7	6	61	32	5	60	13	18	2133	31	141	85,32
8	59	112	95	71	24	345	50	60	1564	120	62,56
9	22	49	10	25	257	66	23	716	188	1144	45,76

**Tableau.5.53.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST pour la méthode de la distance et la classification naïve bayésienne

Chiffres	Sans Squelettisation										
	0	1	2	3	4	5	6	7	8	9	Succès(%)
0	2226	28	11	1	0	125	32	7	69	1	89,04
1	10	1868	24	12	0	286	23	35	197	45	74,72
2	288	35	1737	34	12	67	82	68	167	10	69,48
3	244	50	139	1331	5	215	18	145	330	23	53,24
4	61	129	10	9	1170	319	177	26	340	259	46,80
5	145	53	14	63	12	1883	75	41	189	25	75,32
6	148	90	12	1	1	78	2109	2	59	0	84,36
7	65	66	59	134	36	234	22	1457	250	177	58,28
8	527	72	32	39	3	330	28	24	1427	18	57,08
9	104	94	3	23	84	143	10	112	393	1534	61,36

**Tableau.5.54.** Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST pour la méthode de la somme et la classification naïve bayésienne

Chiffres	Avec Squelettisation										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	2208	2	0	0	1	5	71	0	195	18	88,32
1	0	2395	21	2	0	36	17	18	0	11	95,80
2	17	14	1979	86	9	63	175	69	32	56	79,16
3	6	36	120	1819	0	172	68	191	55	33	72,76
4	22	8	0	4	1312	363	192	21	56	522	52,48
5	4	41	15	137	6	2037	168	15	42	35	81,48
6	2	51	3	1	0	25	2403	0	14	1	96,12
7	15	18	16	30	20	71	7	2029	5	289	81,16
8	70	8	6	9	0	217	73	9	2037	71	81,48
9	15	13	0	36	25	160	4	131	47	2069	82,76

**Tableau.5.55.** Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST pour la méthode de la somme et la classification naïve bayésienne

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	2166	11	4	7	0	116	56	8	130	2	86,64
1	1	2026	22	6	1	265	18	42	87	32	81,04
2	80	14	1886	55	12	89	82	58	201	23	75,44
3	70	24	170	1414	3	277	23	168	329	22	56,56
4	18	49	8	4	1282	403	196	32	228	280	51,28
5	28	27	14	79	10	2035	78	24	184	21	81,40
6	33	33	5	2	1	107	2267	1	51	0	90,68
7	20	22	39	94	40	250	16	1709	138	172	68,36
8	232	17	22	41	3	146	37	11	1769	22	70,76
9	38	34	0	30	76	242	11	115	288	1666	66,64

**Tableau.5.56.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST pour la méthode de la somme et la classification naïve bayésienne

Chiffres	Sans Squelette										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	49	0	0	1	0	0	0	2	5	3	81,67
1	0	49	4	1	0	4	1	0	1	0	81,67
2	0	2	50	1	3	0	0	1	1	2	83,33
3	0	0	0	60	0	0	0	0	0	0	100
4	0	1	0	0	49	6	0	0	4	0	81,67
5	0	6	4	0	1	49	0	0	0	0	81,67
6	0	1	0	3	0	0	54	0	0	2	90
7	0	1	0	4	0	0	4	50	1	0	83,33
8	0	0	0	0	1	0	0	0	59	0	98,33
9	0	0	0	0	0	0	0	1	33	26	43,33

**Tableau.5.57.** Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati pour la méthode de la distance et la classification Sparse Representation

Chiffres	Avec Squelette										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	50	0	0	0	0	0	0	1	7	2	83,33
1	0	42	5	0	1	10	1	0	1	0	70
2	1	4	52	1	0	0	0	0	2	0	86,67
3	0	0	0	59	0	0	0	1	0	0	98,33
4	0	0	1	1	50	0	0	1	7	0	83,33
5	0	1	7	0	1	50	1	0	0	0	83,33
6	0	1	0	16	0	1	41	0	1	0	63,33
7	2	0	0	7	0	0	1	47	3	0	78,33
8	0	0	0	0	0	0	0	0	60	0	100
9	1	0	0	1	0	0	0	0	27	31	51,67

**Tableau.5.58.** Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati pour la méthode de la distance et la classification Sparse Representation

Chiffres	Contribution										Succès(%)
	0	1	2	3	4	5	6	7	8	9	
0	50	0	0	0	0	0	0	3	4	3	83,33
1	0	49	4	1	0	4	1	0	1	0	81,67
2	0	5	49	2	1	0	0	1	2	0	81,67
3	0	0	0	59	0	0	0	1	0	0	98,33
4	0	0	0	1	52	4	0	0	3	0	86,67
5	0	6	2	0	1	50	1	0	0	0	83,33
6	0	1	0	4	0	0	53	0	0	2	83,33
7	0	0	0	5	0	0	3	51	1	0	85
8	0	0	0	0	1	0	0	0	59	0	98,33
9	0	0	0	0	0	0	0	0	33	27	45

**Tableau.5.59.** Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati pour la méthode de la distance et la classification Sparse Representation

On constate d'après les tableaux ci-dessus que, pour les caractères de la base MNIST, le chiffre '8' est confondu avec le chiffre 3, et le '1' est faiblement confondu avec les autres caractères, ceci concerne les trois approches. De même pour les caractères Gujarati, le chiffre '9' est fortement confondu avec le '8', et le '7' est faiblement confondu avec les autres caractères en utilisant les trois approches. Les tableaux 5.60, 5.61, 5.62, 5.63, 5.64 et 5.65 résument les confusions entre les caractères MNIST et Gujarati pour les trois approches. On considère que les caractères sont confondus si le taux d'erreur dépasse 10%.

Caractère Traité	Confusion des caractères avec la méthode de la distance			Confusion des caractères avec la méthode de la somme		
	Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
0	Aucune	3	8	2	Aucune	Aucune
1	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
2	4, 7	4, 5, 7	0, 4, 7	0, 3, 4	4, 7	8
3	8, 9	8	8, 9	5	0, 2, 5, 9	0, 2, 7, 8, 9
4	7	Aucune	5	6	5, 6, 8, 9	6, 9
5	2, 4, 7	4	4	1, 2, 3, 6, 8	3, 6, 8	2, 3, 6, 8
6	0, 8, 9	Aucune	Aucune	0, 4, 5	0, 8	0, 4, 9
7	Aucune	4	Aucune	4, 9	4, 9	4, 9
8	3	3	3	3, 5, 9	3, 5, 6, 9	3, 5, 9
9	8	8	8	0, 3	4, 8	4

**Tableau.5.60.** Confusion des caractères MNIST avec les réseaux de neurones

Caractère Traité	Confusion des caractères avec la méthode de la distance			Confusion des caractères avec la méthode de la somme		
	Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
0	Aucune	Aucune	Aucune	Aucune	7	Aucune
1	Aucune	Aucune	Aucune	Aucune	2	Aucune
2	1	1	1	3	3	Aucune
3	Aucune	Aucune	Aucune	1	Aucune	Aucune
4	8	Aucune	9	7	Aucune	7
5	1,2	1,2	1	7,9	1	1,6,7
6	0	3	9	1,3	3	Aucune
7	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
8	9	Aucune	Aucune	Aucune	Aucune	Aucune
9	8	8	8	7,8	8	8

**Tableau.5.61.** Confusion des caractères Gujarati avec les réseaux de neurones

Caractère Traité	Confusion des caractères avec la méthode de la distance			Confusion des caractères avec la méthode de la somme		
	Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
0	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
1	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
2	Aucune	Aucune	Aucune	1	Aucune	1
3	Aucune	Aucune	Aucune	1,7	Aucune	1
4	Aucune	9	9	1	9	9
5	3	1,3	3	1	1	1
6	Aucune	Aucune	Aucune	1	Aucune	Aucune
7	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
8	Aucune	1	1	0,1	1	1
9	3,7	4,7	4,7	7	Aucune	Aucune

**Tableau.5.62.** Confusion des caractères MNIST avec les k-ppv

Caractère Traité	Confusion des caractères avec la méthode de la distance			Confusion des caractères avec la méthode de la somme		
	Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
0	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
1	5	5	5	5	5	5
2	1	1,8	1	1	1,8	1
3	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
4	Aucune	Aucune	8	Aucune	Aucune	8
5	1	1	Aucune	1	1	Aucune
6	Aucune	3	Aucune	Aucune	3	Aucune
7	Aucune	0,3	Aucune	Aucune	0,3	Aucune
8	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
9	8	8	8	8	8	8

**Tableau.5.63.** Confusion des caractères Gujarati avec les k-ppv

Caractère Traité	Confusion des caractères avec la méthode de la distance			Confusion des caractères avec la méthode de la somme		
	Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
0	Aucune	Aucune	Aucune	Aucune	7	7
1	5	Aucune	5	Aucune	5	Aucune
2	1	1	Aucune	Aucune	Aucune	Aucune
3	Aucune	Aucune	Aucune	Aucune	7	Aucune
4	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
5	Aucune	Aucune	1	Aucune	Aucune	Aucune
6	Aucune	Aucune	Aucune	Aucune	7	Aucune
7	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
8	Aucune	Aucune	Aucune	Aucune	Aucune	Aucune
9	7,8	7,8	7,8	7	Aucune	Aucune

**Tableau.5.64.** Confusion des caractères Gujarati avec la méthode de bayes naïve

Caractère Traité	Confusion des caractères avec la méthode de la distance			Confusion des caractères avec la méthode de la somme		
	Sans Squelette	Avec Squelette	Contribution	Sans Squelette	Avec Squelette	Contribution
0	5,8	5,8	5,8	Aucune	Aucune	Aucune
1	Aucune	Aucune	2	Aucune	5	5
2	Aucune	Aucune	Aucune	Aucune	0	Aucune
3	2	2,5	2,5	Aucune	Aucune	5,8
4	6,9	9	9	5,9	5,8,9	5,9
5	9	Aucune	Aucune	Aucune	Aucune	Aucune
6	Aucune	1	Aucune	Aucune	Aucune	Aucune
7	Aucune	Aucune	Aucune	9	8	5
8	5	5	5	Aucune	0,5	Aucune
9	7	4,7,8	4,7	Aucune	8	8

**Tableau.5.65.** Confusion des caractères MNIST avec la méthode de bayes naïve

Caractère Traité	Confusion des caractères avec la méthode de la distance		
	Sans Squelette	Avec Squelette	Contribution
0	Aucune	8	Aucune
1	Aucune	5	Aucune
2	Aucune	Aucune	Aucune
3	Aucune	Aucune	Aucune
4	5	8	Aucune
5	1	2	1
6	Aucune	3	Aucune
7	Aucune	Aucune	Aucune
8	Aucune	Aucune	Aucune
9	8	8	8

**Tableau.5.66.** Confusion des caractères Gujarati avec la méthode de Sparse Representation

Les tableaux 5.67 et 5.68 montrent des échantillons sur les caractères MNIST et Gujarati confondus.

Chiffre Traité	Chiffres Confondus		
0	૨		
૧	૨ ૨	૨ ૫	૨ ૬
૨	૧ ૫	૧ ૫	૫ ૫
૩	૪ ૬	૬	૬

**Tableau.5.67.** Echantillons sur des caractères Gujarati confondus

૪	૧ ૮	૫ ૮	૭ ૮
૫		૭	
૬		૨	
૭	૧ ૩	૧ ૫	૨ ૮
૮	૨ ૮	૪ ૮	૮ ૮
૯	૦ ૨	૦ ૪	૨ ૮

Tableau.5.67. Echantillons sur des caractères Gujarati confondus (suite)

Chiffre Traité	Chiffres Confondus		
0	૨ ૭	૩ ૫	4 ૫

Tableau.5.68. Echantillons sur des caractères MNIST confondus

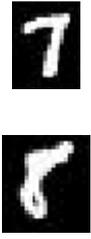
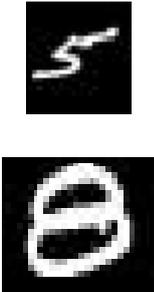
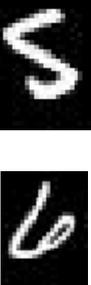
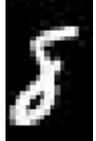
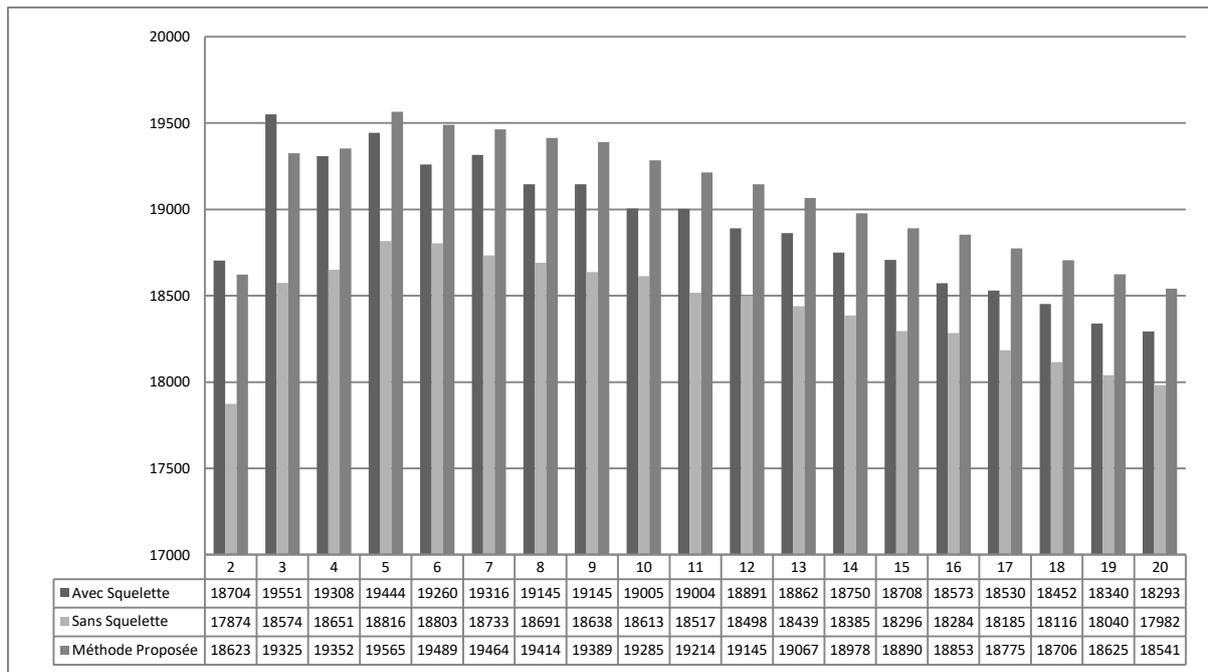
<p style="text-align: center; font-size: 2em; font-weight: bold;">1</p>			
<p style="text-align: center; font-size: 2em; font-weight: bold;">2</p>			
<p style="text-align: center; font-size: 2em; font-weight: bold;">3</p>			
<p style="text-align: center; font-size: 2em; font-weight: bold;">4</p>			
<p style="text-align: center; font-size: 2em; font-weight: bold;">5</p>			

Tableau.5.68. Echantillons sur des caractères MNIST confondus (suite)

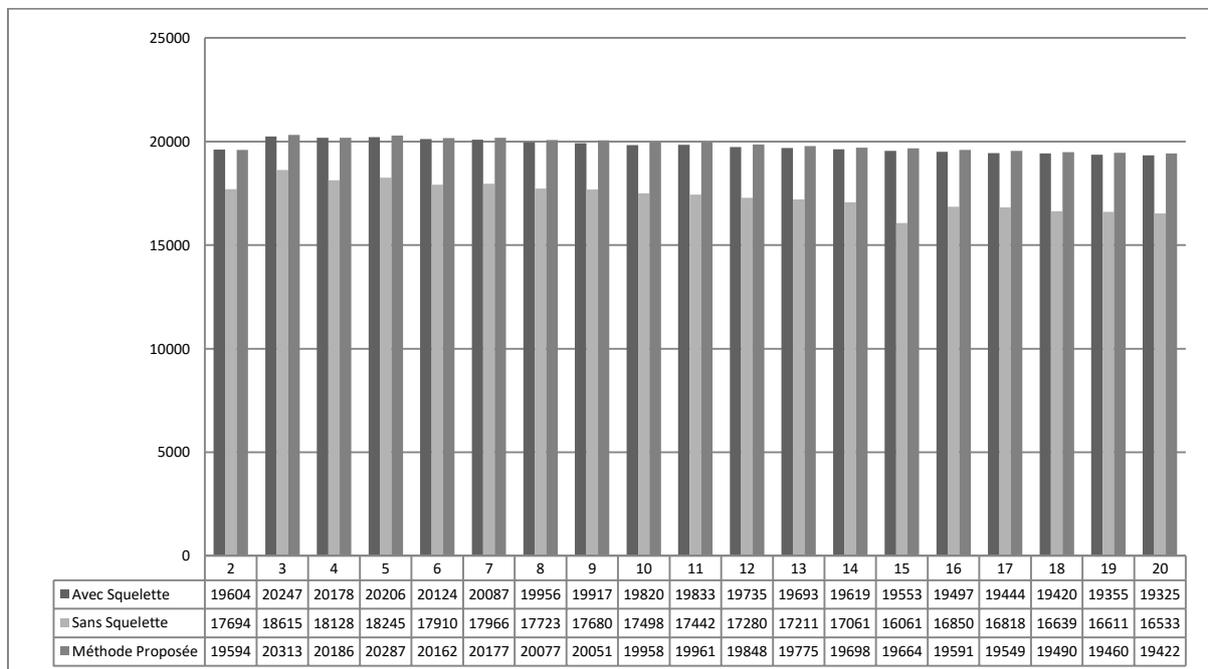
6	 	 	 
7	 	 	 
8	 	 	 
9	 	 	 

**Tableau.5.68.** Echantillons sur des caractères MNIST confondus (suite)

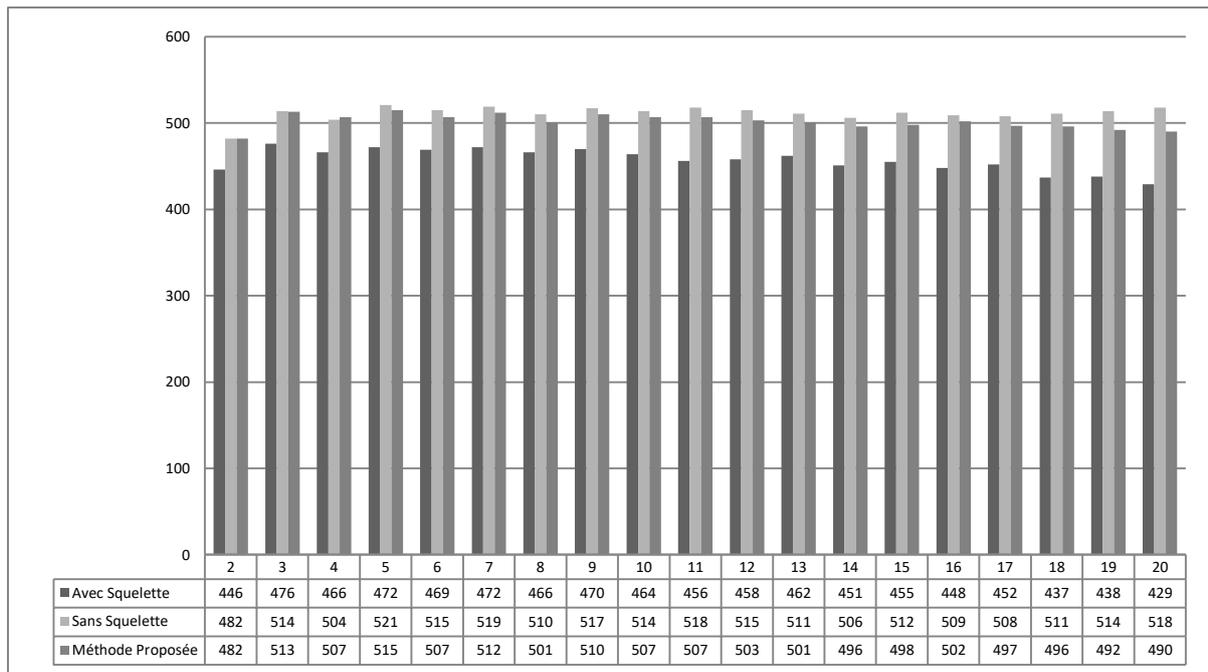
Les Figures 5.7, 5.8, 5.9 et 5.10 montrent l'évolution des taux de reconnaissance en variant le paramètres 'k' de la méthode des K-PPV, et le tableau.5.69 détermine le 'k' optimum pour cette méthode.



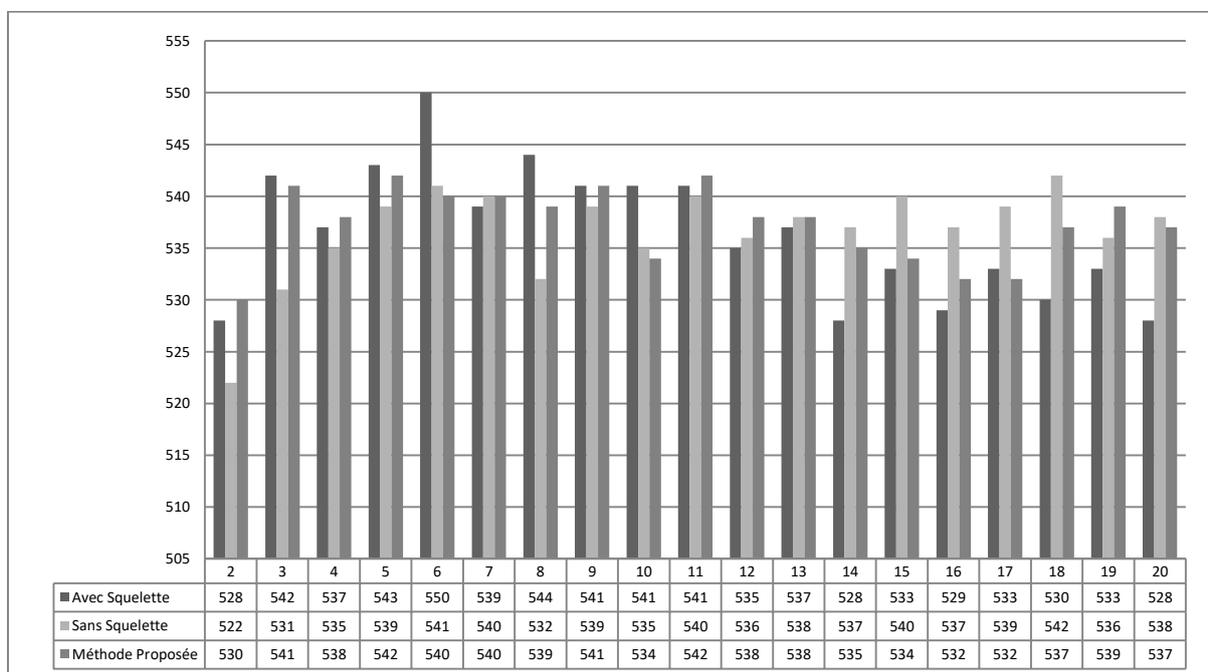
**Figure.5.7.** Evolution du taux de reconnaissance en variant le paramètre 'k' de la méthode des K-PPV avec la méthode de la distance sur les caractères MNIST



**Figure.5.8.** Evolution du taux de reconnaissance en variant le paramètre 'k' de la méthode des K-PPV avec la méthode de la somme sur les caractères MNIST



**Figure.5.9.** Evolution du taux de reconnaissance en variant le paramètre ‘k’ de la méthode des K-PPV avec la méthode de la distance sur les caractères Gujarati

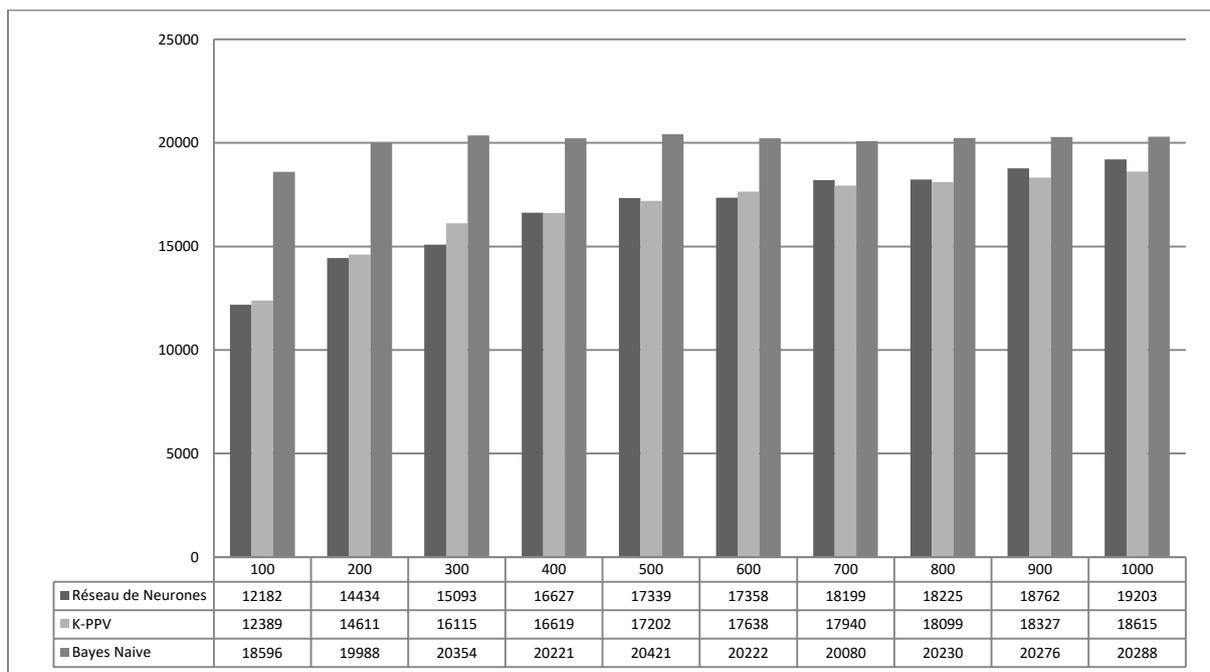


**Figure.5.10.** Evolution du taux de reconnaissance en variant le paramètre ‘k’ de la méthode des K-PPV avec la méthode de la somme sur les caractères Gujarati

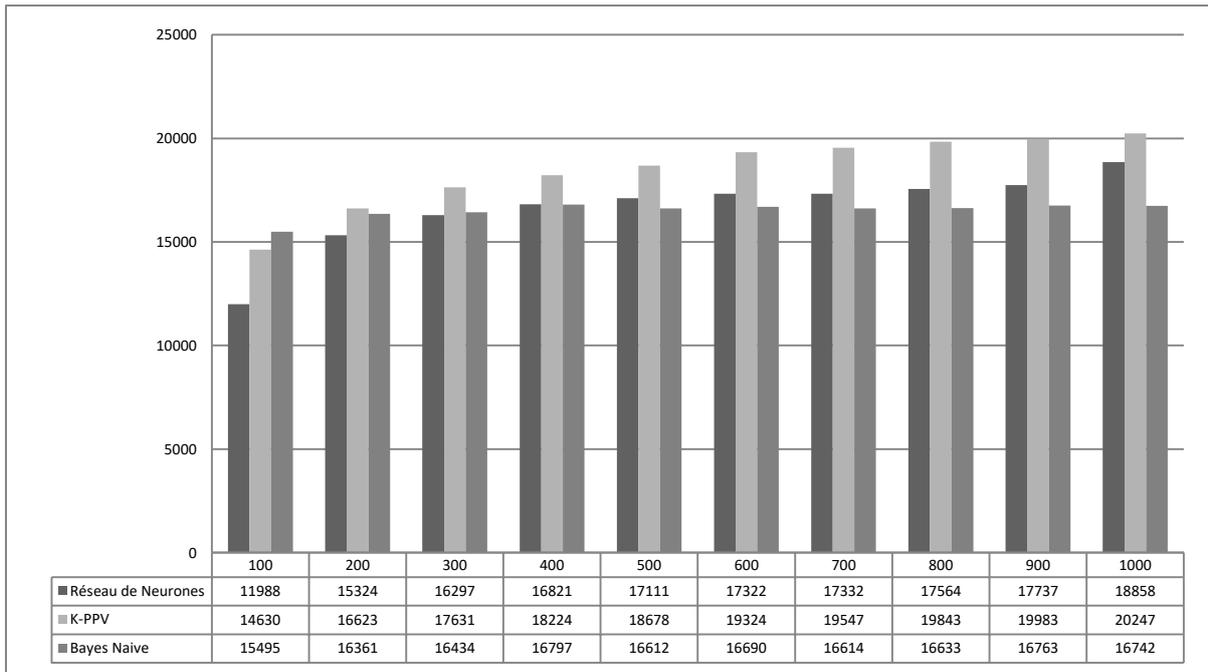
Base des Caractères	Méthode d'Extraction	Utilisation du Squelette	K-Optimum
MNIST	Distance	Sans Squelette	4
MNIST	Distance	Avec Squelette	3
MNIST	Distance	Méthode Proposée	5
MNIST	Somme	Sans Squelette	3
MNIST	Somme	Avec Squelette	3
MNIST	Somme	Méthode Proposée	3
Gujarati	Distance	Sans Squelette	7
Gujarati	Distance	Avec Squelette	3
Gujarati	Distance	Méthode Proposée	5
Gujarati	Somme	Sans Squelette	18
Gujarati	Somme	Avec Squelette	6
Gujarati	Somme	Méthode Proposée	5

**Tableau.5.69.** Détermination du k-optimum pour la méthode des K-PPV

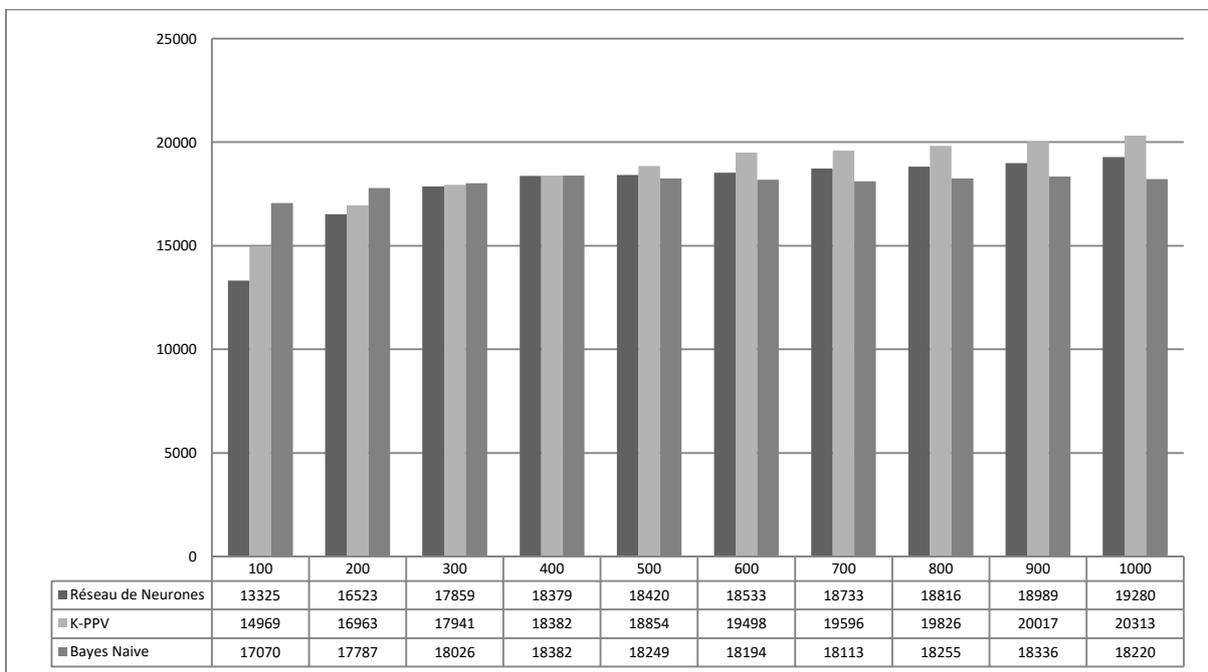
Les Figures 5.11 jusqu'à 5.22 montrent l'évolution des taux de reconnaissance en variant le nombre des caractères d'apprentissage.



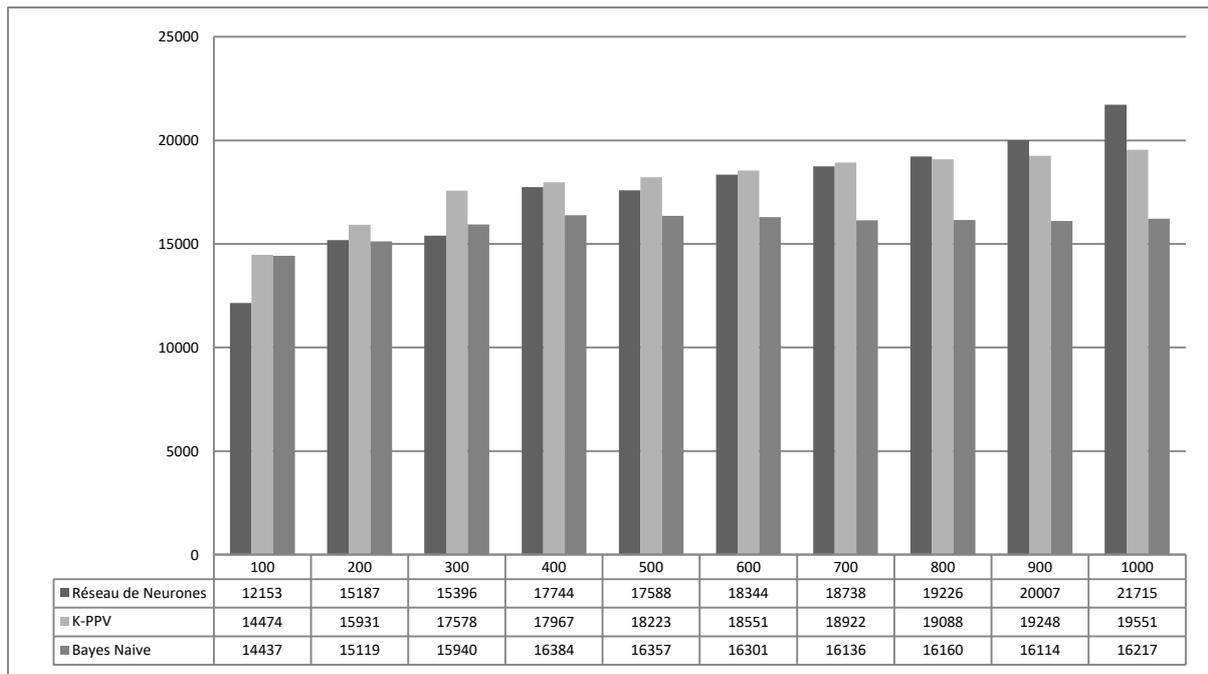
**Figure.5.11.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base MNIST avec la méthode de la somme et la squelettisation



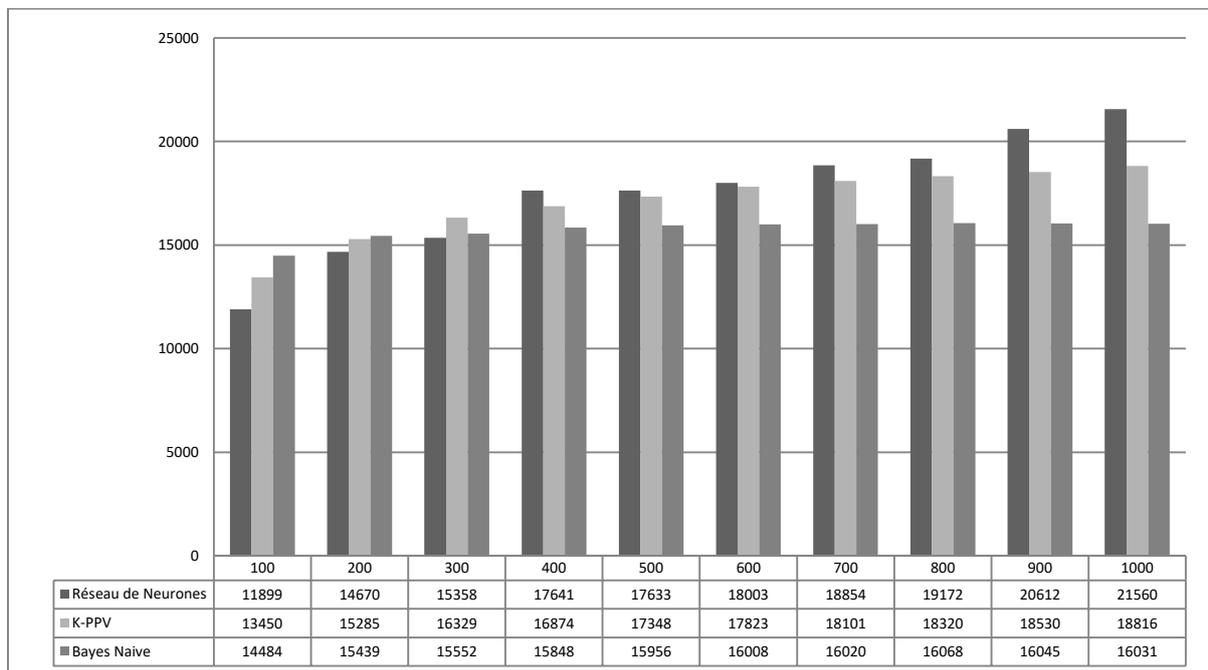
**Figure.5.12.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base MNIST avec la méthode de la somme et sans squelettisation



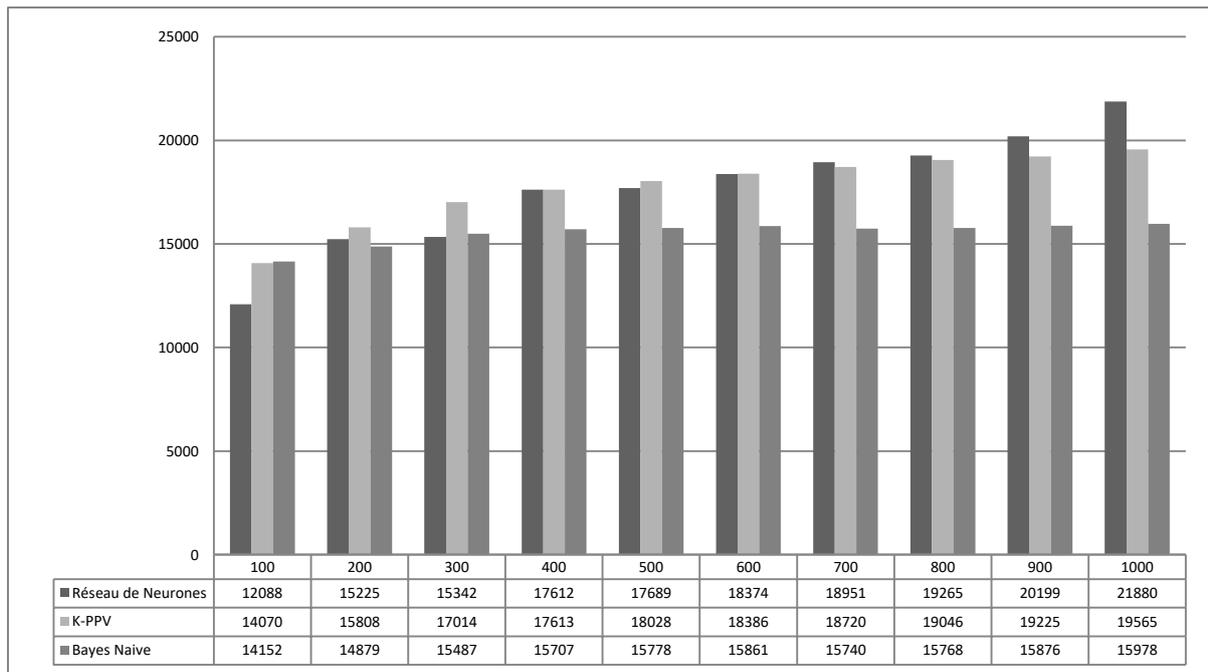
**Figure.5.13.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base MNIST avec la méthode de la somme et la méthode proposée



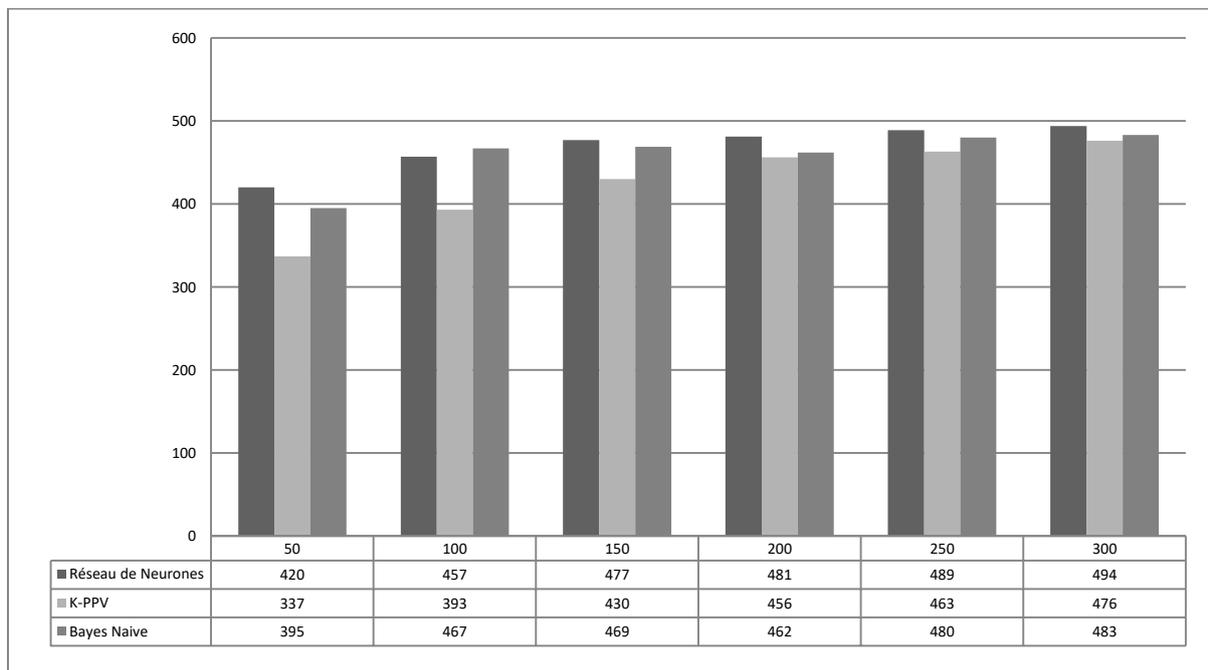
**Figure.5.14.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base MNIST avec la méthode de la distance et la squelettisation



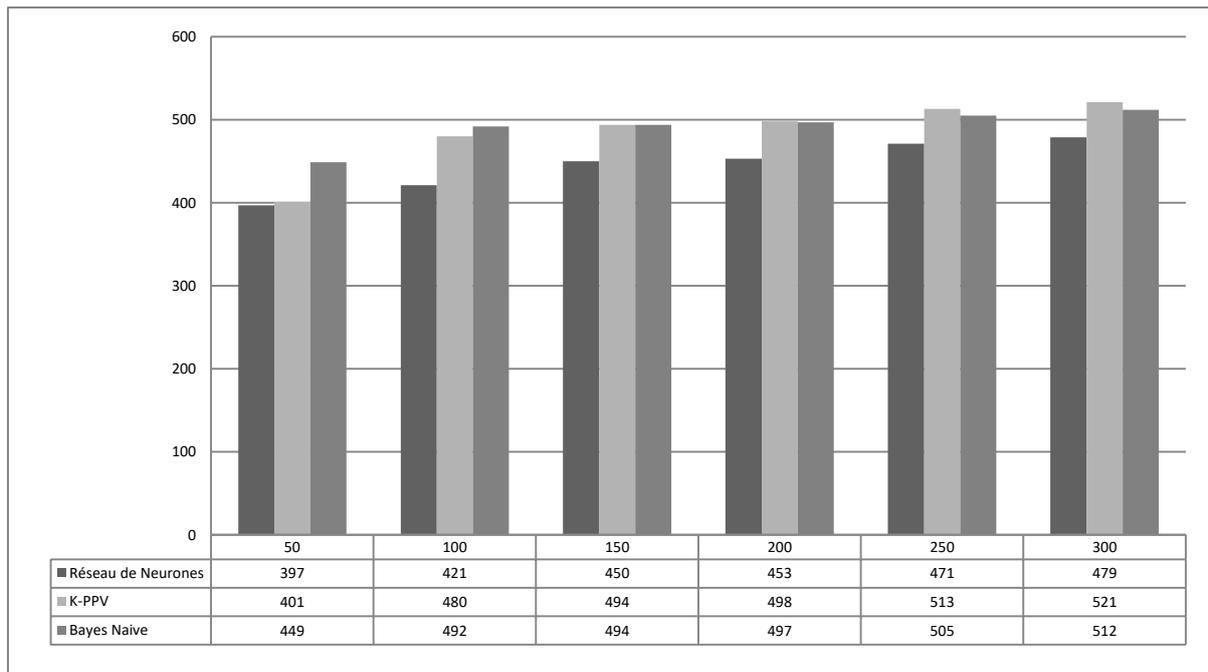
**Figure.5.15.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base MNIST avec la méthode de la distance et sans squelettisation



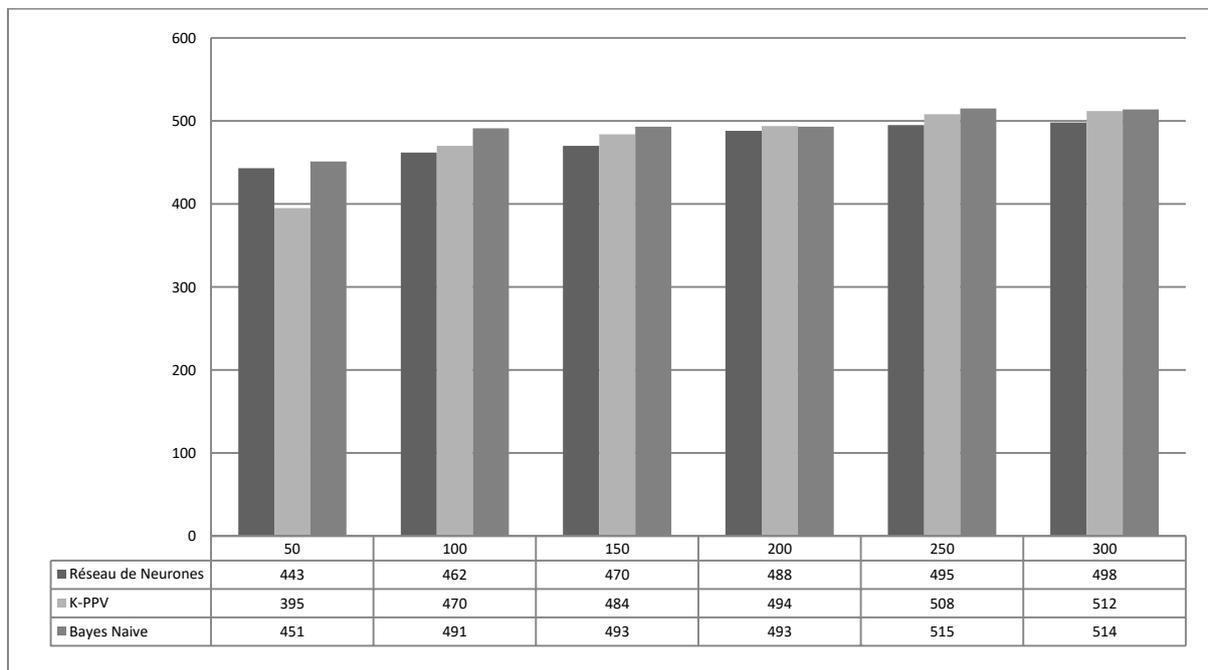
**Figure.5.16.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base MNIST avec la méthode de la distance et la méthode proposée



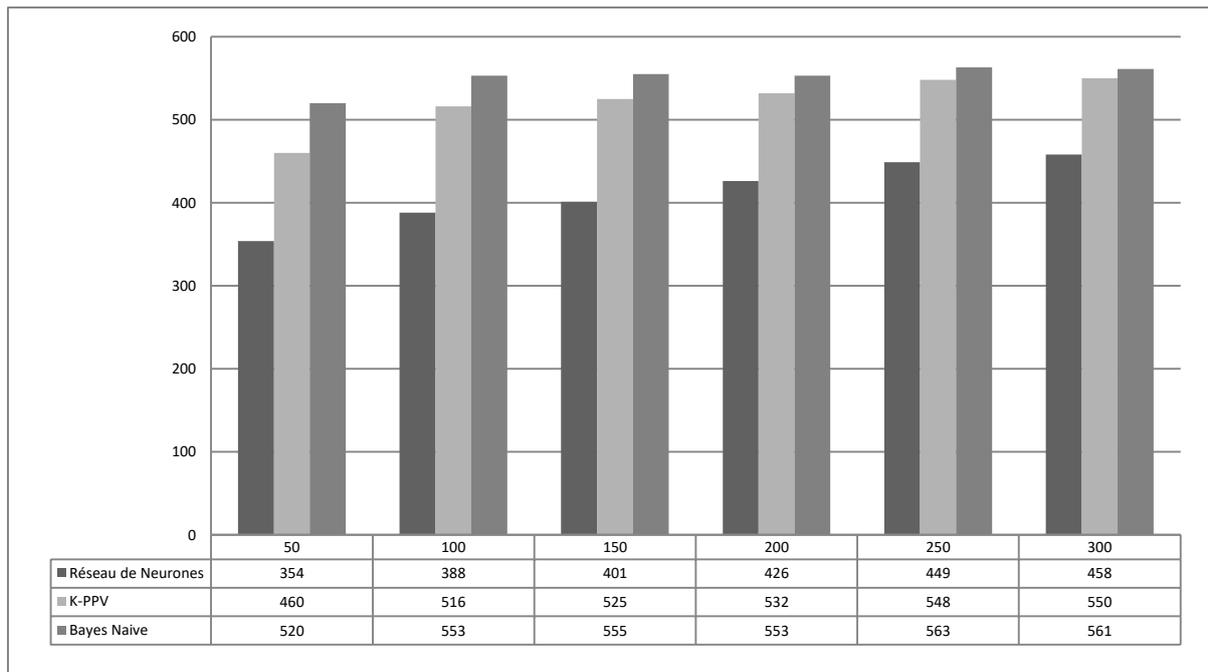
**Figure.5.17.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base Gujarati avec la méthode de la distance et la squelettisation



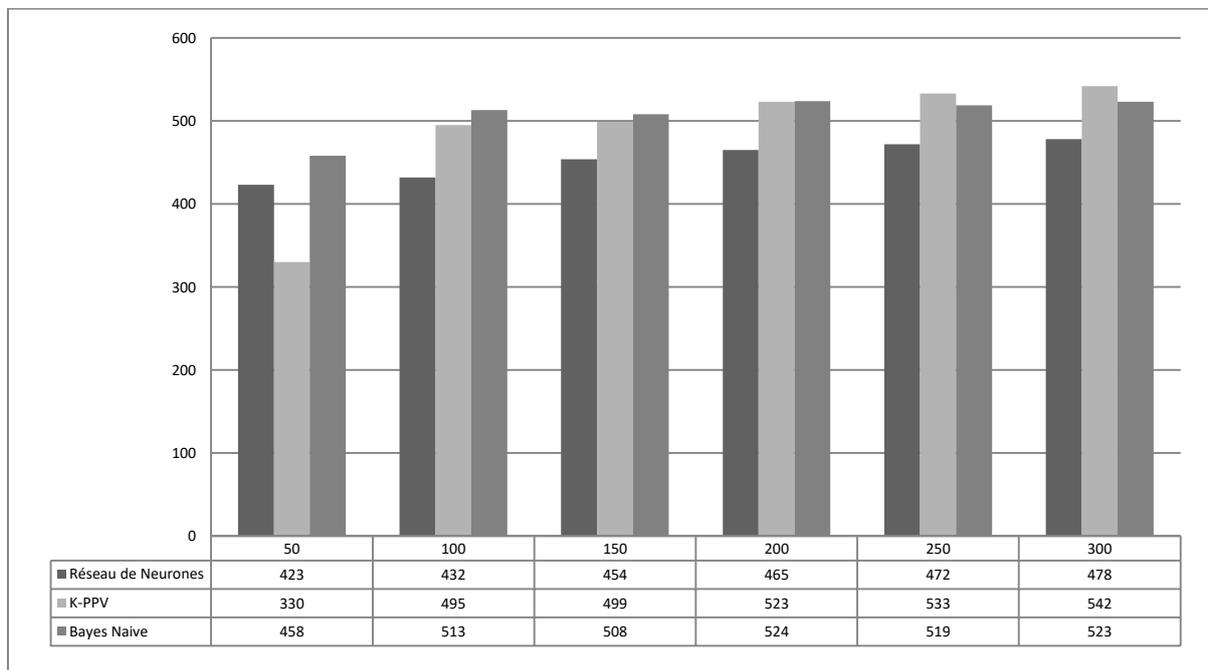
**Figure.5.18.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base Gujarati avec la méthode de la distance et sans squelettisation



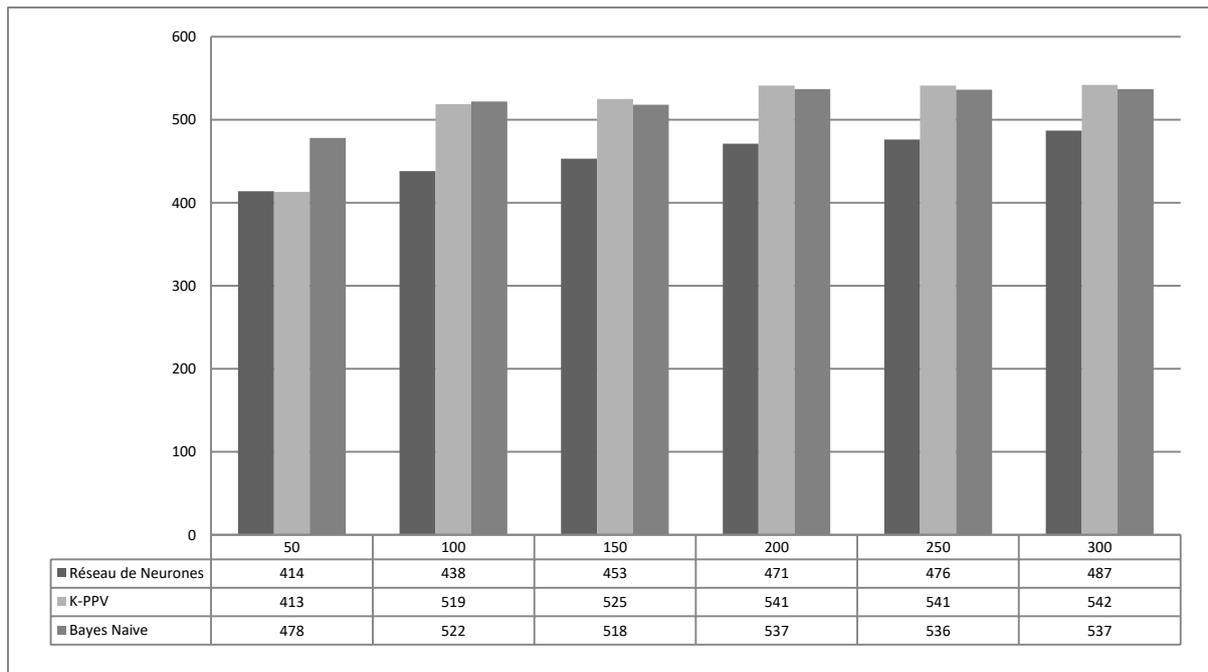
**Figure.5.19.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base Gujarati avec la méthode de la distance et la méthode proposée



**Figure.5.20.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base Gujarati avec la méthode de la somme et la squelettisation



**Figure.5.21.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base Gujarati avec la méthode de la somme et sans squelettisation



**Figure.5.22.** Evolution du taux de reconnaissance en variant le nombre des caractères d'apprentissage sur la base Gujarati avec la méthode de la somme et la méthode proposée

## 9. Conclusion

Dans ce chapitre, on a présenté les différents résultats du processus de reconnaissance décrit précédemment. Dans la phase de prétraitement, on a utilisé l'algorithme de Wolf pour la binarisation et l'algorithme de Guo-Hall pour la squelettisation. Aussi, on a appliqué deux méthodes d'extraction des caractéristiques qu'on les a nommées: la méthode de la distance et la méthode de la somme. Après, on a passé à la phase de classification, trois méthodes largement utilisées par la communauté scientifique ont été mises en application: les réseaux de neurones, les K-PPV et la méthode de bayes naïve. nous avons aussi appliqué la méthode de sparse representation qui est une méthode de classification récente. Enfin, on a réalisé une étude comparative entre les différentes approches. Les résultats expérimentaux ont été appliqués sur les caractères Gujarati et la base des caractères MNIST.

## Conclusion Générale

Dans cette thèse, on a utilisé un processus de reconnaissance basé sur la forme du caractère et sur le squelette. Les résultats expérimentaux ont été appliqués sur la base des caractères Gujarati manuscrits et sur la base MNIST qui est en ligne.

La binarisation constitue une étape importante dans la phase de prétraitement, on a présenté cinq algorithmes de binarisation utilisant l'approche locale, et par la suite, on a utilisé l'algorithme de Wolf qui donne des résultats satisfaisants sur les caractères.

Vingt et un algorithmes de squelettisation ont été décrits et comparés, ces algorithmes donnent des résultats plus ou moins satisfaisants. L'algorithme de Guo-Hall est celui utilisé dans les résultats expérimentaux.

L'extraction des caractéristiques est la phase la plus importante dans tout processus de reconnaissance des caractères, l'augmentation ou la diminution du taux de reconnaissance dépend fortement de cette étape. Dans ce mémoire, on a adopté deux méthodes d'extraction des caractéristiques nommées: la méthode de la distance et celle de la somme.

Les réseaux de neurones, K-PPV et bayes naïve s'avèrent des outils robustes pour résoudre les problèmes de classification des caractères. la rétro propagation du gradient dans une architecture de réseaux de neurones multicouches est celle utilisée dans la phase de classification avec les K-PPV et bayes naïve. On a aussi appliqué la méthode de classification Sparse Representation dans la phase de classification.

L'utilisation de l'approche proposée et décrite dans ce mémoire montre l'avantage par rapport au taux de reconnaissance sur les caractères Gujarati manuscrits et la base MNIST en effectuant une comparaison avec d'autre approches classiques, à savoir le traitement sur la forme brute ou bien sur le squelette, que ce soit en se servant des méthodes d'extraction des caractéristiques: la distance ou la somme.

Pour perspectives, on prévoit de se servir d'autres méthodes de classification (les réseaux bayésiens, la logique floue,...) et l'utilisation d'autres méthodes d'extraction des caractéristiques pour essayer d'augmenter le taux de reconnaissance.

## Annexe

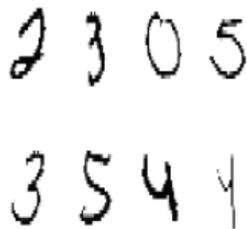
### Base de données

Un système de reconnaissance des caractères est généralement validé sur un ensemble de caractères indépendants de la base d'apprentissage. Pour que ces tests soient concluants, les ensembles de validation doivent inclure un assez grand nombre d'échantillons qui reflètent des styles différents et qui se trouvent dans la vie courante. Dans ce travail, nous avons suivi ce principe de base. Nous avons utilisé la base MNIST qui est en ligne et la base Gujarati manuscrite rédigée par les étudiants de l'université Hassan Premier.

#### 1. Base MNIST

Considéré comme un point de référence pour les essais des algorithmes de classification, la base de données MNIST a été employée couramment pour concevoir des systèmes de reconnaissance des chiffres manuscrits.

La base des caractères MNIST, disponible dans [82], est composée de 60000 caractères d'apprentissage et 10000 caractères de test. C'est un sous-ensemble de la base NIST qui est plus vaste. Les chiffres ont été normalisés et centrés dans une image de taille fixe (28\*28). C'est une base de données convenable pour les personnes qui souhaitent essayer des techniques de reconnaissance des caractères avec un minimum d'effort de normalisation et de prétraitement. La Figure.A.1 montre un échantillon des caractères MNIST.



**Figure.A.1.** Un échantillon des caractères MNIST

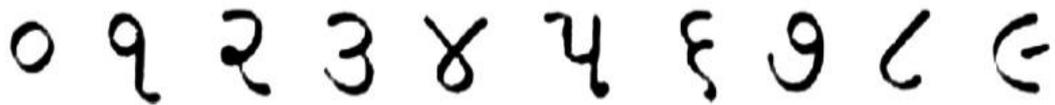
Plusieurs travaux de recherches ont utilisés la base des caractères MNIST, on cite par exemple [76,83,84].

#### 2. Caractères Gujarati

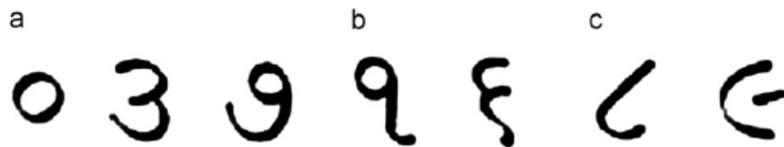
Le gujarati est une langue indienne parlée entre autres au Gujarat, un État de l'ouest de l'Inde, elle est parlée par 50 millions de personnes dans cet état. Il s'agit d'un alphabet très similaire à la Devanagari utilisée pour écrire le hindi, mais sans la ligne supérieure caractéristique de cette écriture et qui diffère par quelques caractères. Comme la quasi-totalité

des écritures indiennes, le gujarati descend de l'écriture brahmi , plus précisément de l'écriture gupta. Le plus ancien document connu écrit en gujarati date de 1592.

Comme il est visible à la Figure.A.2, les chiffres Gujarati sont très particuliers par nature. Seulement deux chiffres, le 1 et le 5 contiennent des lignes droites, ce qui rend l'identification des caractères Gujarati un peu plus difficile et entraîne des erreurs de classification. La Figure.A.3 montre la ressemblance de quelques chiffres Gujarati.



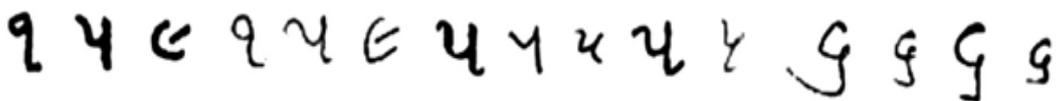
**Figure.A.2.** Chiffres Gujarati 0-9



**Figure.A.3.** Confusion des caractères Gujarati

Comme il est clair dans la Figure.A.3, les chiffres (0, 3 et 7) et les chiffres (1 et 6) et les chiffres (8 et 9) sont similaires.

Pour développer un système pour l'identification des chiffres Gujarati manuscrits, nous avons collectés 1000 chiffres représentant les 10 classes 0-9 écrits par 200 étudiants de l'université Hassan Premier, vu l'inexistence d'une base standard pour les chiffres Gujarati. Ces chiffres ont été balayés par un scanner plat et ils ont été séparés dans des cases 90\*70 chacune. Etant donné que notre problème est d'identifier des chiffres manuscrits, la première chose nécessaire est de normaliser les chiffres. Ceci est nécessaire parce que quand un individu écrit, il peut utiliser différents types de stylos, papiers, il peut même écrire avec des styles différents etc. La Figure.A.4 montre la nécessité du prétraitement des images avant la classification.



**Figure.A.4.** Caractères scannés de différentes natures

Très peu de travaux de recherches ont utilisés la base des caractères Gujarati [7], on site par exemple [85,86,87].

## Références

- [1] Hanmandly, M., Ramana Murthy, O.V, Fuzzy model based recognition of handwritten numerals, Pattern recognition 40, pp: 1840-1854, 2007
- [2] Hanmandlu, M., Murali Mohan, K. R., Chakraborty, S., Goyal, S., Roy Choudhury, D., Unconstrained handwritten character recognition based on fuzzy logic, Pattern Recognition 36 (3), pp: 603-623, 2003
- [3] Hanmandlu, M., Yusof, M. H. M., Vamsi Krishna, M., Off-line signature verification and forgery detection using fuzzy modelling, Pattern Recognition 38 (3), pp: 341-356, 2005
- [4] M.Modi, F.Macwan, R.Prajapati. Gujarati character Identification : A Survery. International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering. Vol 2 Issue 2, Febuary 2014.
- [5] Amrouch, M., Rachidi, A., El Yassa, M., Mammas, D., Apprentissage Markovien et Neuronal: cas des caractères amazighes imprimés, SITACAM, pp: 58-67, 2009
- [6] El Kessab, B., Bouikhalene, B., Fakir, M., Safi, S., Reconnaissance des caractères Tifinaghe par l'utilisation des réseaux de neurones multicouches, SITACAM, pp: 68-83, 2009
- [7] Desai, A., Gujarati handwritten numeral optical character reorganization through neural network, Pattern Recognition 43, pp: 2582-2589, 2010
- [8] K.S.Siddharth, M. Jangid, R.Dhir, R.Rani. Handritten Gurmukhi Character Recognition Using Statistical and Background Directional Ditrubution Feature. International Journal of Computer Science and Engineering. Vol 3 N 6. Juin 2011.
- [9] T. Hildebrandt, W. Liu. Optical Recognition of Handwritten Chinese Characters : advances since 1980. Pattern Recognition, pp: 205-225. 1993
- [10] S . R. Ramesh. A Generalized Character Recognition Algorithm: A Graphical Approach. Pattern Recognition, 22, pp: 347-350. 1989
- [11] G. Baptista, K. M. Kulkarani. A High Curacy Algorithm for Recognition of Handwritten Numerals. Pattern Recognition. pp: 287-291. 1988
- [12] H.S. Baird. Application of Multi-dimensional Search to Structural Feature Identification. Workshop on Syntactical and Structural Pattern Recognition. pp:21-24. 1987
- [13] F. Lebourgeois. Approche Mixte pour la Reconnaissance des Documents Imprimés. Ph.D thesis, Institut National des Sciences Appliquées de Lyon. 1991

- [14] P. Dargenton. Contribution à la Segmentation et à la Reconnaissance de l'Écriture Manuscrite par l'Ordinateur. Ph.D thesis, INSA de Lyon. 1994
- [15] A. Belaid, Y. Belaid. Reconnaissance des Formes : Méthodes et Applications. InterEdition. 1992
- [16] P. Devijver. Reconnaissance des Formes par la Méthode des plus Proches Voisins. Workshop on Syntactical and Structural Pattern Recognition. pp : 21-24. 1987
- [17] L. Beheim. Coopération entre Segmentation et Reconnaissance des Caractères Imprimés Dégradés. Ph.D thesis, Université Pierre et Marie Curie. 2001
- [18] J. T. Tou, R. C. Gonzales. Pattern Recognition. Assion Wesley. 1974
- [19] A. Curnuéjols, L. Miclet. Apprentissage artificiel: concepts et algorithms. Eyrolles. 2002
- [20] V. Vapnik. Statistical Learning Theory. J. Willey. 1998
- [21] L. R. Rabiner, B. H. Juang. An introduction to Hidden Markov Models. Recent Advances in speech Understanding and Dialog Systems, 3. pp: 4-16. 1986
- [22] F. Grandidier, C. R. Suen, M. Gilloux. Une nouvelle stratégie pour l'amélioration des jeux de primitives d'un système de reconnaissance de l'écriture. In Colloque International Francophone sur l'Écrit et le Document (CIFED'2000). pp : 111-120. 2000
- [23] A.J. Elms. The representation and recognition of text using hidden Markov models. Ph.D. thesis, University of Surrey Guildford. 1996
- [24] M. Gilloux. Reconnaissance de chiffres manuscrits par modèles de Markov pseudo 2D. CNED. pp: 11-17. 1994
- [25] J. Pearl. Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference. Morgan Kaufman, deuxième édition. 1988
- [26] S. Souafi. Contribution à la reconnaissance des structures des documents écrits: Approche probabiliste. Ph.D. thesis, INSA de LYON. 2002
- [27] C. CHOW, C. LIU, Approximating Discrete Probability Distributions with Dependence Trees, IEEE Trans. Information Theory, vol.14, p. 462-467, 1968.
- [28] X. Xiao, G. Leedham. Signature verification using a modified Bayesian network. Pattern Recognition, 35. pp: 983-995. 2002

- [29] G. Zweig. Speech recognition with Dynamic Bayesian networks. Ph.D. thesis, University of California, Berkeley. 1998
- [30] K. Daoudi, D. Fohr, C. Antoine. Réseaux bayésiens pour la reconnaissance de l'écriture manuscrite par l'ordinateur. Ph.D. thesis, INSA de LYON. 1994
- [31] G. Leedham, S. Varma, A. Patankar, and V. Govindaraju. Separating text and background in degraded documents images - A comparison of global thresholding techniques for multi-stage thresholding. Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition, p. 244–249, Canada, August 2002
- [32] Abderrahmane Kefali, Toufik Sari et Mokhtar Sellami. Evaluation de plusieurs Techniques de seuillage d'images de documents arabes anciens. IMAGE'09 Biskra
- [33] J. Sauvola, M. Pietikainen. Adaptive document image binarization. Pattern Recognition, 33(2), p. 225–236, 2000
- [34] Kefali.A, Toufik.S, Sellami.M. Evaluation de plusieurs techniques de seuillage d'images de documents arabes anciens. Image'09 Biskra. Pp : 123, 134, 2009.
- [35] K. Khurshid, I. Siddiqi, C. Faure, N. Vincent. Comparison of Niblack inspired Binarization methods for ancient documents. 16th International conference on Document Recognition and Retrieval, USA, 2009
- [36] J. Bernsen, « Dynamic thresholding of grey-level images », in Proc. 8th International Conference on Pattern Recognition, p. 1251-1255, Paris, France, 1986
- [37] W. Niblack, « An Introduction to Digital Image Processing », Ed. Prentice Hall, Englewood Cliffs, 1986
- [38] J. Sauvola, T. Seppänen, S. Haapakoski, M. Pietikänen, « Adaptive document binarisation », Fourth International Conference Document Analysis and Recognition (ICDAR), p. 147-152, Ulm, Germany, August 1997
- [39] C. Wolf, J. M. Jolion, and F. Chassaing. « Extraction de texte dans des vidéos : le cas de la binarisation », In 13ème Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle, p. 145-152, 2002
- [40] H. BLUM, “A transformation for extracting new descriptions of shape”, *In Models for the Perception of Speech and Visual Form*, MIT Press, pp: 362–380, 1967
- [41] D. RUTOVITZ : Pattern recognition. *Journal of Royal Statistical Society*, 129:504–530, 1966
- [42] C.J. HILDITCH : Linear skeletons from square cupboards. *Machine Intelligence*, 4:403–420, 1969
- [43] T.Y. Zhang and C.Y. Suen. A fast parallel algorithm for digital patterns. *Comm. ACM*, 27(3) :236–239, March 1984

- [44] H.E. Lü and P.S.P. Wang. A comment on "a fast parallel algorithm for digital patterns". *Comm. ACM*, 29(3) :239–242, March 1986
- [45] Z. Guo and R.W. Hall. Parallel thinning with two-subiteration algorithms. *Comm. ACM*, 32(3) :359–373, March 1989
- [46] A. Rosenfeld. The fuzzy geometry of image subsets. *Pattern Recognition Letters*, 2 :311–317, 1984
- [47] C.M. Holt, A. Stewart, M. Clint, and R.H. Perrott. An improved parallel thinning algorithm. *Comm. ACM*, 30(2) :156–160, February 1987
- [48] R.W. Hall. Fast parallel thinning algorithms : Parallel speed and connectivity preservation. *Comm. ACM*, 32(1) :124–131, January 1989
- [49] Z. Guo and R.W. Hall. Fast fully parallel thinning algorithms. *CVGIP : Image Understanding*, 55(3) :317–328, May 1992
- [50] R.T. Chin, H.-K. Wan, D.L. Stover, and R.D. Iverson. A one-pass thinning algorithm and its parallel implementation. *Computer Vision, Graphics and Image Processing*, (40) :30–40, 1987
- [51] R.-Y. Wu and W.-H. Tsai. A new one-pass parallel thinning algorithm for binary images. *Pattern Recognition Letters*, 13 :715–723, October 1992
- [52] A. Manzanera, T. M. Bernard, F. Prêteux, and B. Longuet. Medial faces from a concise 3D thinning algorithm. In *Proceedings ICCV*, 1999
- [53] R.C. GONZALEZ et R.E. WOODS : *Digital Image Processing*. Addison Wesley, 1992
- [54] E.S. DEUTSCH : Thinning algorithms on rectangular, hexagonal, and triangular arrays. *Communications of the ACM*, 15(9):827–837, septembre 1972
- [55] Nemeth.G, Palagyi.K. 2D Parallel Thinning Algorithms Based on Isthmus-Preservation. 7<sup>th</sup> International Symposium on Image and Signal Processing and Analysis. (ISPA 2011) , Dubrovnik, September 4-6, Croatia. 2011.
- [56] P. Dargenton. Contribution à la segmentation et à la reconnaissance de l'écriture manuscrite. Thèse de doctorat de l'institut de science appliquée de lion, 1994
- [57] Fluser.J, Suk.T, Zitova.B. Moments and Moment Invariants in Pattern Recognition. (2009).
- [58] Robert M. Haralick and Lhda G. Shapiro. Computer and Robot Vision, volume 1. Addison- Wesley Publishing Company, 1992
- [59] Robert M. Haralick and Linda G. Shapiro. Computer and Robot Vision, volume 2. Addison- Wesley Publishing Company, 1993

- [60] Waî-Hong Wong, Wan-Chi Siu, and Kin-Man Lam. Genemtion of Moment Invariants and Their Uses for Character Recognition. *Pattern Recognition Letters*, 16315-123, february 1995
- [61] F. P. Kuhl and C. R. Giardina. Elliptic fourier feature of a closed contour. *Computer Vision, Graphics and Image Processing*, volume 18, pp: 236–258, 1982
- [62] Christian Brechbuhler and Sean Ho. Reconstruction of a closed curve from its elliptic descriptor
- [63] J. J. Oliveira, J. de Carvalho, C. Freitas, and R. Sabourin. Feature sets evaluation for handwritten word recognition, 2002
- [64] M. Hanmandlu, O.V. Ramana Murthy. Fuzzy model based recognition of handwritten numerals. *Pattern Recognition* 40, p: 1840-1852. 2007
- [65] M. Hanmandlu, K.R. Murali Mohan, S. Chakraborty, S Goyal, D. Roy Choudhury. Unconstrained handwritten character recognition based on fuzzy logic. *Pattern Recognition* 36 (3), p: 603-623. 2003
- [66] M. Hanmandlu, M.H.M. Yusof, M. Vamsi Krishna. Off line signature verification and forgery detection using fuzzy modeling. *Pattern Recognition* 38 (3), p: 341-356. 2005
- [67] W.S McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 1943
- [68] Y. Le Cun. Une procedure d'apprentissage pour réseau à seuil assymétrique. In *Cognitiva 85 : A la Frontière de l'Intelligence Artificielle des Sciences de la Connaissance des Neurosciences*. pp : 599-604, Paris, 1985
- [69] D.E. Rumelhart, G.E. Hinton, R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533-536, 1986
- [70] P. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Phd thesis, Havard University, 1974
- [71] B. Widrow and M. E. Hoff "Adaptive switching circuits", *WESCON Conv. Record*, pp.96 -104. 1960
- [72] F. Rosenblatt. Principles of neurodynamics. Washington, DC: Spartan Books.1962
- [73] A. Maren, C. Harston, R. Pap Handbook of Neural Computing Applications. Academic Press, Inc., San Diego, 1990
- [74] Y. Yang, J. Pederson. A Comparative Study on Feature Selection in Text Categorization. 14th International Conference on Machine Learning. pp. 412-420. 1997
- [75] Zhang.H. The Optimality of Naïve Bayes. FLAIRS2004 Conference.

- [76] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.
- [77] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 11, pp. 2259–2272, 2011.
- [78] J. M. K. Kua, E. Ambikairajah, J. Epps, and R. Togneri, "Speaker verification using sparse representation classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE*, 2011, pp. 4548–4551.
- [79] A. A. Helal, K. I. Ahmed, M. S. Rahman, and S. K. Alam, "Breast cancer classification from ultrasonic images based on sparse representation by exploiting redundancy," in *16th International Conference on Computer and Information Technology, ICCIT 2013, Mar. 2014*.
- [80] E. Amaldi and V. Kann, "On the Approximability of Minimizing Nonzero Variables or Unsatisfied Relations in Linear Systems," *Theoretical Computer Science*, vol. 209, pp. 237-260, 1998.
- [81] D. Donoho, "For Most Large Underdetermined Systems of Linear Equations the Minimal  $l_1$ -Norm Solution Is Also the Sparsest Solution," *Comm. Pure and Applied Math.*, vol. 59, no. 6, pp. 797- 829, 2006.
- [82] E. Cande's, J. Romberg, and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements," *Comm. Pure and Applied Math.*, vol. 59, no. 8, pp. 1207-1223, 2006.
- [83] E. Cande's and T. Tao, "Near-Optimal Signal Recovery from Random Projections: Universal Encoding Strategies?" *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5406-5425, 2006.
- [84] Baheti.M.J, Kale.K.V. Gujarati Numeral Recognition: Affine Invariant Moments Approach. 1<sup>st</sup> International Conference on Recent Trends in Engineering & Technology. Mar 2012.
- [85] G.G. Rajput, Anita.H.B. Handwritten Script Recognition using DCT and Wavelet Features at Block Level. IJACA Special Issue on Recent Trends in Image Processing and Pattern Recognition, 2010
- [86] K.V.Kale, P.D.Deshmukh, S.V.Chavan, M.M.Kazi. Zernike Moment Feature Extraction for Handwritten Devanagari (Marathi) Compound Character Recognition. *International Journal of Advanced Reseach in Artificial Intelligence*, Vol 3 N 1, 2014.
- [87] D. Pavolov, R. Balasubramanyan, B.Dom, S.Kapur, J.Parikh. Document Processing for naïve Bayes Classification and Clustering with mixture of multinomials. *Proceeding of the tenth ACM SIGKDD International Conference on knowledge discovery and data mining*. Pp: 829-834, 2004.

[88] B.R. Shambhavi, P. Ramakanth Kumar. Current State of the Art Pos Tagging for Indian Languages- A Study. International Journal of Computer Engineering and Technology. Vol 1 Num 1. Pp: 250-260. May-June 2010.

[89] <http://yann.lecun.com/exdb/mnist/>

[90] K.T.Abou-Moustafa, M. Cheriet, and C. Suen. Classification of time-series data using a generative/discriminative hybrid. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pages 51 - 56, Oct,2004

[91] C.L Liu, K. Nakashima, H. Sako, H. Fujisawa. Handwritten digit recognition: benchmarking of state-of-the-art techniques. *Pattern Recognition* 36 (10) pp: 2271-2285. 2003

[92] S.N.S. Rajasekaran, B.L. Deekshatulu. Recognition of printed Telugu characters, *Computer Graphics Image Processing*. pp: 335–360. 1977

[93] J. Dholakia, A. Negi, S. Ram Mohan. Zone identification in the printed Gujarati text. *Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR05)*, 2005

[94] M.K. Jindal, R.K. Sharma, G.S. Lehal. Segmentation of horizontally overlapping lines in printed Indian scripts. *International Journal of Computational Intelligence Research* 3 (4) pp:277–286. 2007

## Liste des Figures

<b>Figure.2.1.</b> Comparaison des algorithmes de binarisation avec une fenêtre $r=9*9$ sur le caractère Gujarati	<b>19</b>
<b>Figure.2.2.</b> Comparaison des algorithmes de binarisation avec une fenêtre $r=9*9$ sur une photo réelle	<b>19</b>
<b>Figure.2.3.</b> Comparaison des algorithmes de binarisation avec une fenêtre $r=9*9$ sur un caractère MNIST	<b>20</b>
<b>Figure.2.4.</b> exemple de squelette	<b>20</b>
<b>Figure.2.5.</b> Préservation de la géométrie	<b>21</b>
<b>Figure.2.6.</b> Epaisseur nulle	<b>21</b>
<b>Figure.2.7.</b> Préservation de la topologie	<b>21</b>
<b>Figure.2.8.</b> Invariance aux translations affines	<b>22</b>
<b>Figure.2.9.</b> Réversibilité	<b>22</b>
<b>Figure.2.10.</b> Continuité	<b>22</b>
<b>Figure.2.11.</b> Un point P et ses voisinages	<b>22</b>
<b>Figure.2.12.</b> Principe des 4(-8) voisinages	<b>23</b>
<b>Figure.2.13.</b> Chemin 4-connexe	<b>23</b>
<b>Figure.2.14.</b> Boules maximale	<b>23</b>
<b>Figure.2.15.</b> Points 8-(-4) supprimables	<b>24</b>
<b>Figure.2.16.</b> Condition 4 de l'algorithme GH_AFP2	<b>30</b>
<b>Figure.2.17.</b> Condition a)-4 de l'algorithme GH_AFP3	<b>31</b>
<b>Figure.2.18.</b> Condition b) de l'algorithme GH_AFP3	<b>31</b>
<b>Figure.2.19.</b> Conditions de l'algorithme CWSI	<b>32</b>
<b>Figure.2.20.</b> Conditions de l'algorithme WT	<b>33</b>
<b>Figure.2.21.</b> Conditions de l'algorithme MB_2	<b>34</b>
<b>Figure.2.22.</b> Conditions de la non suppression des points marqués 'x'	<b>37</b>

<b>Figure.2.23.</b> Notation des voisinages 3*3 d'un point $P$	<b>37</b>
<b>Figure.2.24.</b> Comparaison des algorithmes de squelettisation sur le caractère Gujarati 'l'	<b>39</b>
<b>Figure.2.25.</b> Continuité de l'Algorithme de Guo-Hall	<b>40</b>
<b>Figure.2.26.</b> Comparaison des algorithmes de squelettisation sur le caractère '4' de la base MNIST	<b>41</b>
<b>Figure.3.1.</b> Somme des pixels d'une matrice 3*3 pixels	<b>46</b>
<b>Figure.3.2.</b> Forme 3*3 pixels	<b>47</b>
<b>Figure.3.3.</b> Division de l'image en blocks et le calcul des distances	<b>47</b>
<b>Figure.4.1</b> Un neurone biologique et ses principaux composants	<b>50</b>
<b>Figure.4.2.</b> Quelques fonctions d'activation	<b>52</b>
<b>Figure.4.3.</b> Différentes configurations de réseaux	<b>53</b>
<b>Figure.4.4.</b> Un exemple de perceptron	<b>55</b>
<b>Figure.4.5.</b> Architecture d'un réseau de neurones avec une couche cachée	<b>57</b>
<b>Figure.4.6.</b> Graphe de la fonction logistique $y=f(x)$	<b>58</b>
<b>Figure.4.7.</b> Surapprentissage dans un apprentissage supervisé.	<b>61</b>
<b>Figure 5.1.</b> Processus de Reconnaissance, a: avec squelette, b: sans squelette	<b>70</b>
<b>Figure 5.2.</b> Processus de Reconnaissance avec la méthode proposée	<b>71</b>
<b>Figure.5.3.</b> Application de l'algorithme de Wolf sur le caractère Gujarati	<b>71</b>
<b>Figure.5.4.</b> Application de l'algorithme Guo_Hall sur le caractère Gujarati	<b>72</b>
<b>Figure.5.5.</b> Application de l'algorithme de Wolf sur le caractère MNIST	<b>72</b>
<b>Figure.5.6.</b> Application de l'algorithme de Guo_Hall sur le caractère MNIST	<b>72</b>
<b>Figure.5.7.</b> Evolution du taux de reconnaissance en variant le paramètre 'k' de la méthode des K-PPV avec la méthode de la distance sur les caractères MNIST	<b>101</b>
<b>Figure.5.8.</b> Evolution du taux de reconnaissance en variant le paramètre 'k' de la méthode des K-PPV avec la méthode de la somme sur les caractères MNIST	<b>101</b>
<b>Figure.5.9.</b> Evolution du taux de reconnaissance en variant le paramètre 'k' de la méthode des K-PPV avec la méthode de la distance sur les caractères Gujarati	<b>102</b>

<b>Figure.5.10.</b> Evolution du taux de reconnaissance en variant le paramètre ‘k’ de la méthode des K-PPV avec la méthode de la somme sur les caractères Gujarati	<b>102</b>
<b>Figure.5.11.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base MNIST avec la méthode de la somme et la squelettisation	<b>103</b>
<b>Figure.5.12.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base MNIST avec la méthode de la somme et la sans squelettisation	<b>104</b>
<b>Figure.5.13.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base MNIST avec la méthode de la somme et la méthode proposée	<b>104</b>
<b>Figure.5.14.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base MNIST avec la méthode de la distance et la squelettisation	<b>105</b>
<b>Figure.5.15.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base MNIST avec la méthode de la distance et sans squelettisation	<b>105</b>
<b>Figure.5.16.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base MNIST avec la méthode de la distance et la méthode proposée	<b>106</b>
<b>Figure.5.17.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base Gujarati avec la méthode de la distance et la squelettisation	<b>106</b>
<b>Figure.5.18.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base Gujarati avec la méthode de la distance et sans squelettisation	<b>107</b>
<b>Figure.5.19.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base Gujarati avec la méthode de la distance et la méthode proposée	<b>107</b>
<b>Figure.5.20.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base Gujarati avec la méthode de la somme et la squelettisation	<b>108</b>
<b>Figure.5.21.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base Gujarati avec la méthode de la somme et sans squelettisation	<b>108</b>
<b>Figure.5.22.</b> Evolution du taux de reconnaissance en variant le nombre des caractères d’apprentissage sur la base Gujarati avec la méthode de la somme et la méthode proposée	<b>109</b>
<b>Figure.A.1.</b> Un échantillon des caractères MNIST	<b>111</b>
<b>Figure.A.2.</b> Chiffres Gujarati 0-9	<b>112</b>

<b>Figure.A.3.</b> Confusion des caractères Gujarati	<b>112</b>
<b>Figure.A.4.</b> Caractères scannés de différentes natures	<b>112</b>

## Liste des Tableaux

<b>Tableau.3.1.</b> Vecteurs d'Extraction	<b>48</b>
<b>Tableau.5.1.</b> Dimensions des Vecteurs d'Extraction suivant les cas	<b>73</b>
<b>Tableau.5.2.</b> Architecture des réseaux de Neurones suivant les cas	<b>73</b>
<b>Tableau.5.3.</b> Valeurs des Vecteurs d'Extraction suivant les cas	<b>74</b>
<b>Tableau.5.4.</b> Taux de Reconnaissance sur les Caractères Gujarati avec les réseaux de neurones	<b>76</b>
<b>Tableau.5.5.</b> Taux de Reconnaissance sur les Caractères MNIST avec les réseaux de neurones	<b>76</b>
<b>Tableau.5.6.</b> Taux de Reconnaissance avec k-ppv	<b>76</b>
<b>Tableau.5.7.</b> Taux de Reconnaissance avec la méthode de Bayes Naïve	<b>77</b>
<b>Tableau.5.8.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST d'apprentissage pour la méthode de la distance avec le réseau des neurones	<b>77</b>
<b>Tableau.5.9.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST d'apprentissage pour la méthode de la distance avec le réseau des neurones	<b>77</b>
<b>Tableau.5.10.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST d'apprentissage pour la méthode de la distance avec le réseau des neurones	<b>78</b>
<b>Tableau.5.11.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST de reconnaissance pour la méthode de la distance avec le réseau des neurones	<b>78</b>
<b>Tableau.5.12.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST de reconnaissance pour la méthode de la distance avec le réseau des neurones	<b>78</b>
<b>Tableau.5.13.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST de reconnaissance pour la méthode de la distance avec le réseau des neurones	<b>79</b>
<b>Tableau.5.14.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST d'apprentissage pour la méthode de la somme avec le réseau des neurones	<b>79</b>

<b>Tableau.5.15.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST d'apprentissage pour la méthode de la somme avec le réseau des neurones	<b>79</b>
<b>Tableau.5.16.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST d'apprentissage pour la méthode de la somme avec le réseau des neurones	<b>80</b>
<b>Tableau.5.17.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST de reconnaissance pour la méthode de la somme avec le réseau des neurones	<b>80</b>
<b>Tableau.5.18.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST de reconnaissance pour la méthode de la somme avec le réseau des neurones	<b>80</b>
<b>Tableau.5.19.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST de reconnaissance pour la méthode de la somme avec le réseau des neurones	<b>81</b>
<b>Tableau.5.20.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la distance avec le réseau des neurones	<b>81</b>
<b>Tableau.5.21.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la distance avec le réseau des neurones	<b>81</b>
<b>Tableau.5.22.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati d'apprentissage pour la méthode de la distance avec le réseau des neurones	<b>82</b>
<b>Tableau.5.23.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati de reconnaissance pour la méthode de la distance avec le réseau des neurones	<b>82</b>
<b>Tableau.5.24.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati de reconnaissance pour la méthode de la distance avec le réseau des neurones	<b>82</b>
<b>Tableau.5.25.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati de reconnaissance pour la méthode de la distance avec le réseau des neurones	<b>83</b>
<b>Tableau.5.26.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la somme avec le réseau des neurones	<b>83</b>

<b>Tableau.5.27.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la somme avec le réseau des neurones	<b>83</b>
<b>Tableau.5.28.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati d'apprentissage pour la méthode de la somme avec le réseau des neurones	<b>84</b>
<b>Tableau.5.29.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati de reconnaissance pour la méthode de la somme avec le réseau des neurones	<b>84</b>
<b>Tableau.5.30.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati de reconnaissance pour la méthode de la somme avec le réseau des neurones	<b>84</b>
<b>Tableau.5.31.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati de reconnaissance pour la méthode de la somme avec le réseau des neurones	<b>85</b>
<b>Tableau.5.32.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati pour la méthode de la distance et les k-ppv	<b>85</b>
<b>Tableau.5.33.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati pour la méthode de la distance et les k-ppv	<b>85</b>
<b>Tableau.5.34.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati pour la méthode de la distance et les k-ppv	<b>86</b>
<b>Tableau.5.35.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati pour la méthode de la somme et les k-ppv	<b>86</b>
<b>Tableau.5.36.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati pour la méthode de la somme et les k-ppv	<b>86</b>
<b>Tableau.5.37.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati pour la méthode de la somme et les k-ppv	<b>87</b>
<b>Tableau.5.38.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST pour la méthode de la distance et les k-ppv	<b>87</b>
<b>Tableau.5.39.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST pour la méthode de la distance et les k-ppv	<b>87</b>
<b>Tableau.5.40.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST pour la méthode de la distance et les k-ppv	<b>88</b>
<b>Tableau.5.41.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST pour la méthode de la somme et les k-ppv	<b>88</b>

<b>Tableau.5.42.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST pour la méthode de la somme et les k-ppv	<b>88</b>
<b>Tableau.5.43.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST pour la méthode de la somme et les k-ppv	<b>89</b>
<b>Tableau.5.44.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati pour la méthode de la distance et la classification naïve bayésienne	<b>89</b>
<b>Tableau.5.45.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati pour la méthode de la distance et la classification naïve bayésienne	<b>89</b>
<b>Tableau.5.46.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati pour la méthode de la distance et la classification naïve bayésienne	<b>90</b>
<b>Tableau.5.47.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères Gujarati pour la méthode de la somme et la classification naïve bayésienne	<b>90</b>
<b>Tableau.5.48.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères Gujarati pour la méthode de la somme et la classification naïve bayésienne	<b>90</b>
<b>Tableau.5.49.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères Gujarati pour la méthode de la somme et la classification naïve bayésienne	<b>91</b>
<b>Tableau.5.50.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST pour la méthode de la distance et la classification naïve bayésienne	<b>91</b>
<b>Tableau.5.51.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST pour la méthode de la distance et la classification naïve bayésienne	<b>91</b>
<b>Tableau.5.52.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST pour la méthode de la distance et la classification naïve bayésienne	<b>92</b>
<b>Tableau.5.53.</b> Performance du processus de reconnaissance sans squelettisation sur les caractères MNIST pour la méthode de la somme et la classification naïve bayésienne	<b>92</b>
<b>Tableau.5.54.</b> Performance du processus de reconnaissance avec squelettisation sur les caractères MNIST pour la méthode de la somme et la classification naïve bayésienne	<b>92</b>
<b>Tableau.5.55.</b> Performance du processus de reconnaissance avec la méthode proposée sur les caractères MNIST pour la méthode de la somme et la classification naïve bayésienne	<b>93</b>
<b>Tableau.5.56.</b> Confusion des caractères MNIST avec les réseaux de neurones	<b>93</b>
<b>Tableau.5.57.</b> Confusion des caractères Gujarati avec les réseaux de neurones	<b>93</b>

<b>Tableau.5.58.</b> Confusion des caractères MNIST avec les k-ppv	<b>94</b>
<b>Tableau.5.59.</b> Confusion des caractères Gujarati avec les k-ppv	<b>94</b>
<b>Tableau.5.60.</b> Confusion des caractères Gujarati avec la méthode de bayes naïve	<b>95</b>
<b>Tableau.5.61.</b> Confusion des caractères MNIST avec la méthode de bayes naïve	<b>95</b>
<b>Tableau.5.62.</b> Echantillons sur des caractères Gujarati confondus	<b>95</b>
<b>Tableau.5.63.</b> Echantillons sur des caractères MNIST confondus	<b>96</b>
<b>Tableau.5.64.</b> Détermination du k-optimum pour la méthode des K-PPV	<b>96</b>