

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Batna 2 – Mustapha Ben Boulaid

Faculté de Technologie

Département Génie Industriel



THÈSE

Présentée au

Laboratoire d'Automatique et Productique

En vue de l'obtention du diplôme de doctorat en sciences

Spécialité : Génie industriel

Option : Génie industriel

Par

Khedidja DERDOUR

Magister en Génie Industriel

THEME

Un Système de Reconnaissance de Formes Basé sur une Approche Multi-Classifieurs

Présentée devant le jury composé de

M. Djamel MOUSS	Prof.	Université de Batna 2	Président
L. Hayet MOUSS	Prof.	Université de Batna 2	Rapporteur
Abdelkamel TARI	Prof.	Université de Bejaia	Examineur
Khaled REZEG	Prof.	Université de Biskra	Examineur
Nabil BENOUDJIT	Prof.	Université de Batna 2	Examineur
Okba KAZAR	Prof.	Université de Biskra	Examineur

Soutenue le 14 Janvier 2021

Un Système de Reconnaissance de Formes Basé sur une Approche Multi-Classifieurs

Résumé

Améliorer la performance d'un système de reconnaissance de formes fait l'objet de recherche dans de nombreuses disciplines. On obtient cette amélioration par l'optimisation dans les différents partis du système de reconnaissance de formes : les prétraitements, l'extraction des paramètres caractéristique (primitives), la classification.

Les travaux de recherche présentés dans cette thèse abordent le problème de la reconnaissance des chiffres arabe imprimés et manuscrits. L'objectif principal de ce travail est l'amélioration de la performance en terme de taux de reconnaissance par l'application d'un système multi classificateur (MCS). Différents classifieurs sont utilisés (KPPV, PMC, SVM, LDA, Arbre de décision, Naïve Bayesien, Pseudo inverse), à l'aide de différents types de vecteurs de caractéristiques extraite de l'image. Enfin, comme les performances de MCS dépendent des performances des classifieurs appliquée (les performances individuelles des classifieurs), et comme la performance d'un classifieur dépend ainsi des caractéristiques utilisé l'optimisation de l'extractions des primitives pertinentes est également abordée dans la thèse.

Nous avons réalisé plusieurs simulations pour éprouver les classifications, en introduisant des améliorations dans les paramètres caractéristiques et en faisant des combinaisons de classifieurs. Dans une première partie, nous montrons l'intérêt de l'utilisation des paramètres caractéristique pertinente, à l'aide des classifieur individuelle (séparé, indépendants) comme source d'inspiration pour la conception de nouveaux paramètres. Nous proposons en particulier une amélioration de primitives pour la caractérisation des chiffres. On montre qu'il est possible de développer une solution efficace, à moindre coût en terme de réduction de vecteur caractéristique et transformation géométrique.

Donc, Le système développé s'articule autour de quatre modules distincts. Un module de prétraitements, un d'extraction des paramètres caractéristique, un module de reconnaissance (classification) et un module de combinaison de classifieurs. Ce dernier est chargé de fusionner les sorties (décisions) de chaque classifieur basant sur des méthodes (règles) de fusion.

Les résultats obtenus sur les bases des chiffres imprimés et la base MNIST des chiffres manuscrits sont prometteurs. Cette thèse apporte quelques contributions pour faire avancer notre compréhension dans ce domaine de recherche en pleine expansion.

Mots-clés : combinaison des classifieurs, apprentissage automatique, paramètres caractéristiques.

نظام التعرف على الأشكال بالاعتماد على نهج متعدد المصنفات

الملخص

تحسين أداء نظام التعرف على الأشكال هو موضوع بحث في العديد من التخصصات. يتم الحصول على هذا التحسن من خلال التحسين في المراحل المختلفة لنظام التعرف على الأشكال: المعالجة، واستخراج الخصائص المميزة، والتصنيف.

يتناول البحث المقدم في هذه الرسالة مشكلة التعرف على الأرقام العربية المطبوعة والمكتوبة باليد. الهدف الرئيسي من هذا العمل هو تحسين الأداء من حيث نسبة التعرف على الأرقام من خلال تطبيق نظام متعدد المصنفات. تم استخدام مصنفات مختلفة (ك أقرب جار -KPPV-)، الشبكات العصبية المتعددة الطبقات PMC، شعاع الدعم الآلي SVM، تحليل التمييز الخطي LDA، شجرة القرار، بايز البسيط Naïve Bayesian، التسمية العكسية (Reverse Pseudo)، وذلك باستخدام أنواع مختلفة من الخصائص المستخرجة من الصورة. أخيرًا، نظرًا لأن أداء المصنفات المتعددة يعتمد على أداء المصنفات الفردية (الأداء الفردي للمصنفات)، وبما أن أداء المصنف يعتمد على الخصائص المستخدمة، فإن دراسة الخصائص المناسبة تمت مناقشتها في الأطروحة.

أجرينا العديد من التجارب لاختبار المصنفات من خلال إدخال تحسينات في الخصائص المميزة واستعمال دمج المصنفات. الاهتمام الأول هو استخدام الخصائص المميزة والمناسبة بمساعدة المصنفات الفردية (مصنفات منفصلة ومستقلة) كمصدر إلهام لاستخدام الخصائص الجديدة. على وجه الخصوص، نقترح تحسين الخصائص التي تصف الأرقام. وثبت أنه من الممكن، بتكلفة أقل، إيجاد حل فعال من حيث تقليل حجم جدول الخصائص المميزة والأخذ بعين الاعتبار عدم التأثير بالتحويلات الهندسية.

لذلك، فإن تطوير النظام يدور حول أربع وحدات مختلفة. وحدة المعالجة، استخراج الخصائص المميزة، وحدة التصنيف و وحدة دمج المصنفات. هذا الأخير مسؤول عن دمج مخرجات (قرار) كل مصنف بناءً على استخدام قواعد الدمج.

النتائج التي تم الحصول عليها في التعرف على الأرقام المطبوعة وقاعدة البيانات MNIST للأرقام المكتوبة بخط اليد واعدة. تقدم هذه الرسالة بعض المساهمات لتعزيز فهمنا في هذا المجال الواسع للبحث.

الكلمات المفتاحية: دمج المصنفات، التعلم الآلي، الخصائص المميزة.

Pattern Recognition System Based on a Multi-Classifier Approach

Abstract

Improving the performance of a pattern recognition system is the subject of research in many disciplines. This improvement is obtained by the optimization in the various parties of the pattern recognition system: preprocessing, feature extraction and classification.

The research presented in this thesis addresses the problem of recognition of printed and manuscript Arabic numerals (digits). The main objective of this work is the improvement of the performance in terms of recognition rate by the application of a multi-classifier system (MCS). Different classifiers are used (KPPV, MLP, SVM, LDA, Decision Tree, Naïve Bayesian, Reverse Pseudo), with different types of feature extracted from the image. Finally, since the performance of MCS depends on the performance of the classifiers applied (the individual performances of the classifiers), and as the performance of a classifier thus depends on the features used, the optimization of the pertinent feature extractions is also discussed in the thesis.

We used several simulations to test the classifications by introducing improvements in the feature extraction and making combinations of classifiers. In a first part, we show the interest of using the pertinent features with the help of individual classifiers (separate, independent) as inspiration for the design of new features. In particular, we propose an improvement of features for the characterization of digits. It is shown that it is possible to develop an effective solution, lower cost in terms of reducing the size of characteristic vector and invariance to geometric transformations.

So, the developed system is structured around four distinct modules. A preprocessing module, feature extraction, recognition module (classification) and a classifiers combination module. The latter is responsible for the integration of the output (decision) of each classifier based on fusion rule.

The results obtained on the basis of printed digits and the MNIST database of handwritten digits are promising. This thesis provides some contributions to advance our understanding in this expanding field of research.

Keywords: combination of classifier, machine learning, feature extraction.

Remerciements

Remercier est une tâche ingrate puisqu'elle conduit inévitablement à des oublis. Je vais donc commencer par remercier tous ceux qui m'ont aidé de près ou de loin au cours de mes recherches et ont contribué par un mot, un geste, une attention à ce travail et qui ne seront pas cités par la suite.

Je tiens à exprimer toute ma gratitude à mon encadreur Dr Leila Hayet Mouss, Professeur à l'Université Batna 2, qui m'a introduit dans le monde de la recherche scientifique et pour avoir accepté d'encadrer mes travaux et de diriger ce travail de thèse. Merci pour tous vos commentaires et suggestions pertinentes qui ont enrichi cette thèse. Vous trouvez ici l'expression de ma profonde gratitude, Un grand merci à vous.

Merci à Dr Mouss M. Djamel, Professeur à l'Université Batna 2, qui a accepté d'être président de ce jury de Soutenance.

Je remercie également Dr Benoudjit Nabil, Professeur à l'Université Batna 2, Dr Kazar Okba et Dr Rezeg Khaled tous deux Professeurs à l'Université de Biskra, et Dr Tari Abdelkamel, Professeur à l'Université de Bejaia qui ont acceptés d'examiner cette thèse. Je voudrais également remercier chacun, de l'honneur qu'ils me font en étant membre de mon jury.

Je tiens tout particulièrement à remercier Monsieur Pr Tebbikh Hichem qui m'a accueillie au laboratoire et de m'avoir donnée des conseils précieux. Mes remerciements vont à M. BenSaadi Rafik, qui a construit le premier pas pour cet axe de recherche, et je lui souhaite le meilleur au cours de la recherche scientifique.

Enfin, je voudrais exprimer toute ma reconnaissance à mes proches et à mes amis qui m'ont toujours soutenu et encouragé pendant ce long travail. Tous ceux que je remercie sincèrement le savent bien, et ceci sans avoir à préciser le moindre nom, prénom ou surnom.

Je remercie vivement mes grandes familles et ma petite famille de la patience sur mes sautes d'humeur et négligente tout au long de ce travail, qu'ils me pardonnent de tout ce que je leur ai fait subir.

شكرا للجميع

Merci à vous tous

Table des Matières

Résumé.....	i
Remerciements	iv
Table des Matières.....	v
Liste des Tableaux.....	viii
Liste des Figures.....	x
Liste des Abréviations.....	xii
INTRODUCTION GENERALE.....	1
Chapitre 1 . ETAT DE L'ART SUR LA RECONNAISSANCE DES CHIFFRES.....	8
1.1. Introduction.....	9
1.2. Bref historique.....	9
1.3. Reconnaissance de caractère	11
1.4. Domaines d'application de la reconnaissance des chiffres	13
1.5. Reconnaissance des chiffres imprimés	15
1.6. Reconnaissance des chiffres manuscrits.....	17
1.7. Reconnaissance des chiffres par combinaison des classifieurs.....	23
1.8. Conclusion.....	26
Chapitre 2 . SYSTEME DE RECONNAISSANCE DE FORMES.....	27
2.1. Introduction.....	28
2.2. Concepts de base en RdF.....	29
2.3. Prétraitements	30
2.4. Extraction des paramètres caractéristiques	31
2.4.1. Primitives structurelles	33
2.4.2. Primitives statistiques (global)	33
2.4.3. Sélection des paramètres caractéristiques.....	35
2.5. Classification.....	36
2.5.1. Concepts de base	36
2.5.2. Les approches de classification	37
2.6. Conclusion.....	42
Chapitre 3 . SYSTEME MULTICLASSIFIEURS.....	45
3.1. Introduction.....	46
3.2. Définition d'un classifieur	47
3.2.1. Type classe	47
3.2.2. Type rang.....	48
3.2.3. Type mesure	48
3.2.4. Type ensemble	48

3.2.5.	Type oracle	48
3.3.	Système multi-classifieurs	49
3.4.	Les Stratégies de combinaison.....	51
3.4.1.	Approche séquentielle	51
3.4.2.	Approche parallèle	52
3.4.3.	Approche hybride.....	52
3.5.	Les taxonomies de combinaison	53
3.6.	Les méthodes de combinaison des classifieurs	56
3.6.1.	Fusion des classifieurs	56
3.6.2.	Sélection des classifieurs	59
3.7.	Conclusion.....	60
Chapitre 4 . PRESENTATION DU SYSTEME DEVELOPPE		61
4.1.	Introduction.....	62
4.2.	Base de données et prétraitements	62
4.3.	Les paramètres caractéristiques.....	67
4.3.1.	Les moments géométriques	67
4.3.2.	Les moments Hu et les moments invariants étendus	68
4.3.3.	Les moments de Zernike.....	69
4.3.4.	Cavités.....	70
4.3.5.	HOG	71
4.3.6.	Zonage et multi-zonage.....	71
4.3.7.	Profil.....	72
4.3.8.	Histogramme de projection	73
4.3.9.	Rétine (le codage rétinien)	73
4.4.	La classification	73
4.4.1.	Perceptron Multi Couche	74
4.4.2.	Machine à vecteurs de support	74
4.4.3.	K Plus Proche Voisin (KPPV)	75
4.5.	Evaluation des Performances.....	76
4.6.	Combinaison des classifieurs.....	77
4.6.1.	Vote à la pluralité	78
4.6.2.	Borda count	78
4.6.3.	Les règles fixes	79
4.7.	Conclusion.....	80
Chapitre 5 . ANALYSE ET DISCUSSION DES RESULTATS EXPERIMENTAUX DE L'APPROCHE DEVELOPPEE		82

5.1. Introduction.....	83
5.2. Les chiffres imprimés	83
5.3. Les chiffres manuscrits.....	101
5.4. Conclusion.....	109
CONCLUSION GENERALE & PERSPECTIVES	110
REFERENCES BIBLIOGRAPHIQUES.....	114
ANNEXES.....	132
ANNEXE 1 : Glossaire sur la RdF	132
ANNEXE 2 : Glossaire sur les MCS	139
ANNEXE 3 : Les bases de données et les méthodes implémentées	142
ANNEXE 4 : Les images des chiffres de différentes langues	151

Liste des Tableaux

Tableau 01 : Description de la structure de thèse.	6
---	---

Chapitre 1

Tableau 1.1 : Résultats de référence sur la base USPS dans [7] 2004	18
Tableau 1.2 : Résultats de référence de différents types de CNN sur la base MNIST	21
Tableau 1.3 : Résultats de référence des classifieurs individuels sur la base MNIST	22
Tableau 1.4 : Résultats de référence de la combinaison des classifieurs sur la base MNIST	25

Chapitre 2

Tableau 2.1 : Taxonomie des méthodes d'extraction de caractéristiques selon la représentation de l'image [6, 7, 96]	34
Tableau 2.2 : Approches de la RdF	41

Chapitre 3

Tableau 3.1 : Caractéristiques des méthodes de combinaison parallèle [5]	58
--	----

Chapitre 4

Tableau 4.1 : Nombre et pourcentage d'instances de chaque classe	63
Tableau 4.2 : Moments Hu invariant du chiffre « 5 »	68
Tableau 4.3 : Les 12 ordres des moments de Zernike	69
Tableau 4.4 : Sommaire des paramètres des classifieurs	76
Tableau 4.5 : Exemple de Borda count	79
Tableau 4.6 : Exemple montre les résultats de règles fixe	80

Chapitre 5

Tableau 5.1 : Performance de KPPV utilisant différentes distances	84
Tableau 5.2 : Performance par la hybridation Zonage-Cavité	87
Tableau 5.3 : Performance du classifieur PMC	88
Tableau 5.4 : Performance du PMC par concaténation de paramètres caractéristiques	89
Tableau 5.5 : Performance par Hu moment nombre de hole cavité Est surface et ratio	90
Tableau 5.6 : Description et dimension des vecteurs de caractéristiques	91
Tableau 5.7 : Performance de KPPV	91
Tableau 5.8 : Performance de Naïve Bayes	92
Tableau 5.9 : Performance de l'arbre de décision	92
Tableau 5.10 : Performance de LDA	93
Tableau 5.11 : Performance de SVM	93
Tableau 5.12 : Performance de PMC	93
Tableau 5.13 : Ordre des classifieurs pour chaque type de paramètre caractéristique	94
Tableau 5.14 : Ordre des paramètres caractéristiques pour chaque classifieur	94

Tableau 5.15 : Différentes concaténations des paramètres caractéristiques.....	95
Tableau 5.16 : Combinaison de classifieurs	96
Tableau 5.17 : Combinaison de classifieurs de même type avec différents paramètres caractéristiques	97
Tableau 5.18 : Quelques combinaisons de différents classifieurs avec différents paramètres caractéristiques	98
Tableau 5.19 : Performance des classifieurs sur la base BCI_3	101
Tableau 5.20 : Performance des classifieurs	102
Tableau 5.21 : Performance des classifieurs avec enrichissement par "le nombre de holes"	103
Tableau 5.22 : Combinaison des classifieurs avec différents paramètres caractéristiques.....	103
Tableau 5.23 : Combinaison des classifieurs avec les mêmes paramètres caractéristiques.....	104
Tableau 5.24 : Combinaison de différents classifieurs avec différents paramètres caractéristiques	105
Tableau 5.25 : Résultats de références sur la base MNIST	106

Liste des Figures

Figure1 : Structure de la thèse	5
---------------------------------------	---

Chapitre 1

Figure 1.1 : Classification des types d'écriture	12
Figure 1.2 : Exemple de différents types d'écriture (a) Caractères isolés, (b) Texte continu (c) Script cursif, (d) texte contrainte, (e) texte non contrainte. [6]	13

Chapitre 2

Figure 2.1 : Système de RdF général [186]	29
Figure 2.2 : Système de RdF [189]	30
Figure 2.3 : Etapes de base de la conception d'un système de classification [208]	35
Figure 2.4 : Diagramme de la RdF statistique	39
Figure 2.5 : Diagramme de la RdF syntaxique	40
Figure 2.6 : Approches de la RdF	44

Chapitre 3

Figure 3.1 : Les phases d'un système multi-classifieurs	49
Figure 3.2 : Combinaison séquentielle de classifieurs	51
Figure 3.3 : Combinaison parallèle de classifieurs	52
Figure 3.4 : Combinaison hybride de classifieurs	53
Figure 3.5 : Taxonomie des méthodes de combinaison parallèle de classifieurs	54
Figure 3.6 : Taxonomie dans le contexte du MCS	55

Chapitre 4

Figure 4.1 : Un exemple de chiffres extrait de la base MNIST	63
Figure 4.2 : Image du chiffre "3"	64
Figure 4.3 : Image du chiffre "3" de la référence [242]	65
Figure 4.4 : Etape de localisation	66
Figure 4.5 : Squelettisation	66
Figure 4.6 : Chiffres synthétique sous Matlab avant et après prétraitement	66
Figure 4.7 : Cavités : Est : rouge, Ouest : vert, Nord : bleu, Sud : jaune, et Centrale : rose	70
Figure 4.8 : Blocs et cellules d'une image	71
Figure 4.9 : Paramètres HOG du chiffre "0"	71
Figure 4.10 : Multi-zonage du chiffre "2"	72
Figure 4.11 : Profil horizontale du chiffre "5" et "3"	72
Figure 4.12 : Histogramme de projection horizontale et verticale du chiffre "5"	73
Figure 4.13 : Présentation matricielle du chiffre "1"	73
Figure 4.14 : PMC sous Matlab	74
Figure 4.15 : Architecture proposée pour notre MCS	81

Chapitre 5

Figure 5.1 : Système multi-classifieurs KPPV	84
Figure 5.2 : Centre de gravité et similarité des cavités entre les chiffres	85
Figure 5.3 : Cavités de sous blocs	86
Figure 5.4 : Système multi-classifieurs	86
Figure 5.5 : Système multi-classifieurs PMC	88
Figure 5.6 : Système multi-classifieurs	89
Figure 5.7 : Différence entre hole et cavité centrale	90
Figure 5.8 : Chiffres utilisés dans [77]	100
Figure 5.9 : Quelques chiffre déformés	100
Figure 5.10 : Chiffre "5" déconnecté	101
Figure 5.11 : Architecture du système proposé	102
Figure 5.12 : Chiffres mal classés [131]	107
Figure 5.13 : Quelques chiffres (5 6 8 9 3 8) manuscrits déformés	108

Liste des Abréviations

ACP	Analyse en Composantes Principales
AG	Algorithme Génétique
ANPR	Automatic Numeral-plate recognition.
BCI	Base des Chiffres Imprimés
BKS	Behaviour Knowledge Space
BMP	BitMap
CEDAR	Center of Excellence for Document Analysis and Recognition
CENPARMI	Centre For Pattern Recognition and Machine Intelligence.
CNN	Convolutional Neural Networks (Réseau de Convolution)
DAR	Document Analysis and Recognition
DCNNs	Deep Convolutional Neural Networks
DCSNN	Deep Convolutional Spiking Neural Network
DNN	Deep Neural Networks
DS	Sélection Dynamique
DT	Decision tree
ERIM	Environmental Research Institute of Michigam
FD	Descripteur de Fourier
CPU	Central Procressing unit
GPU	Graphics Processing Unit
HWDR	Hand Writing Digit Recognition
HMM	Hidden Markov Model (MMC Modèle de Markov Caché)
HNN	Réseau de Neurones de Hopfield
HOG	Histogram of Oriented Gradients (Histogramme de Gradient Orienté)
IEEE	Institute of Electrical and Electronics Engineers
INDCENPARMI	Base d'images de chiffres Indiens de CENPARMI.
KMOD	Kernel with Moderate Decreasing
KPPV	K Plus Proche Voisin (K-NN: K Nearest Neighbor)
LDA	Linear Discriminant Analysis (Analyse Discriminante Linéaire)
LDC	Linear Discriminant Classifier
LVQ	Learning Vector Quantization
MCDNN	Multi-column DNN
MCS	Multiple Classification System
MLP	Multi-Layer Perceptron (PMC Perceptron Multi-Couche)
MNIST	Modified ou Mixed National Institute of Standards and Technology
MTL	Multi-task learning
NB	Naïve Bayes

NIN	Network In Network
NIST	National Institute of Standards and Technology
NN	Neural Network
OAA	One Against All ou one versus all
OCR	Optical Character Recognition
OVA	One Versus All
OVO	One Versus One
PCA	Principal Component Analysis
PMC	le Perceptron Multicouche
PNG	Portable Network Graphics
QDC	Quadratic Discriminant Classifier
QNN	Quadratic Neural Network (Réseau neuronal quantique)
RBF	Radial Basis Function (Noyau à base radiale)
RBF	Fonction de Base Radiale
RdF	reconnaissance de formes (Pattern Recognition)
RF	Random forest
RMDL	Random Multimodel Deep Learning
ROC	Receive Operating Characteristic
R-signature	Transformée de Randon
RSS	Random Sub-spaces
R-STDP	Reward-modulated STDP (Spike-Timing Dependent Plasticity)
SD	Sélection Dynamique
SDNN	Réseau de Neurones à Déplacement Spatial
SMCS	Single Classifier-based Multiple Classification Scheme
SNN	Spiking Neural Networks
SOM	Self-Organizing Map (Cartes Auto-Organisées)
SS	Sélection Statique
STDP	Spike-Timing Dependent Plasticity
SVHN	Street View House Numbers
SVM	Machines à Vecteurs de Support
TD	Décision Template
TDNN	Réseau Neuronal à Délai Différé
UDT	Unbalanced Decision Tree
USPS	United States Postal Service.
VHS	Vérification Hors-Ligne Des Signatures Manuscrites
WEKA	Waikato Environment for Knowledge Analysis.
ZIP	Zone Improvement Plan

INTRODUCTION GENERALE

La reconnaissance de formes (RdF) est une tâche qui permet aux machines de reconnaître différentes formes. En général, les formes avec des étiquettes (ou classes) connues sont utilisées pour former des agents appelés classifieurs. Une fois formés, ces classifieurs peuvent classifier de nouvelles formes avec des étiquettes inconnues dans certaines classes. En d'autres termes, les classifieurs sont conçus pour rechercher la relation entre les caractéristiques et les étiquettes de formes.

La lecture automatique de l'écriture manuscrite ou de texte est un vieux et difficile problème de la RdF. Elle existe depuis les années 1980. Elle présente un intérêt indéniable dans l'accomplissement des tâches fastidieuses comme celles que l'on rencontre dans certains domaines comme le tri postal, la lecture de chèques bancaires, la lecture des bordereaux, des bons de commande, des feuilles de déclaration etc.... Elle offre aujourd'hui un surcroît d'intérêt avec le développement de nouvelles méthodes permettant de communiquer directement avec la machine de manière plus naturelle, grâce à l'essor des agendas, des blocs-notes électroniques et des ordinateurs sans clavier.

L'un des sous problème de « la lecture automatique de texte » est la reconnaissance des chiffres qui revêt une grande importance. Il a suscité un vif intérêt de la part des chercheurs. Cela est dû à la grande variété d'applications potentielles, telles que le traitement bancaire des chèques pour vérifier les montants, la reconnaissance des codes postaux pour le tri du courrier postal, les entrées numériques dans des formulaires remplis à la main (par exemple, formulaires fiscaux, formulaires de recensement), la lecture des documents imprimés, la reconnaissance des chiffres en ligne sur des tablettes informatiques, la reconnaissance des matricules de véhicule, la reconnaissance de vitesse sur les panneaux routiers etc..

La variabilité de l'écriture, taille, épaisseur, orientation, position et la qualité faible des documents numérisés permet de confronter les algorithmes de classification et d'apprentissage à des problèmes difficiles. À ce niveau, une différence de traitement est faite suivant la nature de l'écrit. En effet l'imprimé n'est pas traité de la même manière que le manuscrit. L'imprimé est

caractérisé par une grande régularité. Par opposition, la grande variabilité associée à l'écriture manuscrite complique fortement la tâche de classification.

La reconnaissance de chiffres est un sujet de recherche important dans le domaine de la reconnaissance et l'analyse de document (Document Analysis and Recognition –DAR-). L'objectif général du DAR est d'automatiser entièrement le processus de saisie et de compréhension des données imprimées ou manuscrites dans l'ordinateur. Le grand défi est de faire en sorte que les ordinateurs s'approchent de la performance humaine pour comprendre ces documents [1].

Notre objectif dans ce travail est le développement d'une méthode de classification pour reconnaître les chiffres (0 - 9) imprimés et manuscrits fournis dans la base de données MINIST (Modified ou Mixed National Institute of Standards and Technology)

Il existe différents types de classifieurs, tels que les perceptrons multicouches (PMC), les modèles de Markov cachés (HMM - Hidden Markov Model-), les K-plus proches voisins (KPPV) et les machines à vecteurs de support (SVM), entre autres. En raison de la complexité du problème, de la dimension « caractéristique », de la dimension « classe » et du nombre d'échantillons d'apprentissage disponibles, certains algorithmes de classification peuvent donner de meilleurs résultats que d'autres. Lorsque nous considérons la sélection d'un algorithme de classification adéquat pour un problème particulier, l'objectif de base est double : Augmenter la précision dans toute la mesure du possible et réduire au maximum le temps d'apprentissage du classifieur.

Le domaine de la RdF, utilise un grand nombre de classifieurs et de méthodes d'extraction de paramètres caractéristiques. Les nombreux travaux dans le domaine cherchent à optimiser un seul classifieur en choisissant les meilleures caractéristiques pour un problème donné. Cela n'a pas permis de mettre en évidence la supériorité d'une méthode de classification sur une autre ou d'un type de paramètres caractéristiques sur un autre.

En effet, de plus en plus, les chercheurs se rendent compte qu'il n'existe aucune méthode pouvant manifester une supériorité sur les autres dans tous les problèmes et toutes les situations. Toutes les méthodes ont leurs points forts et leurs points faibles. Si aucune d'entre elles ne peut satisfaire entièrement les exigences d'une application envisagée, l'utilisation simultanée de plusieurs méthodes en même temps peut éventuellement permettre d'en cumuler les avantages sans en cumuler les inconvénients.

Cependant actuellement les chercheurs ont trouvé plus intéressant de combiner des méthodes de reconnaissance. Donc en pratique, Il est tout à fait possible que la combinaison des décisions de plusieurs systèmes capables d'améliorer les performances de la classification. Dès lors plusieurs mots sont apparus tels que : Système multi-classifieur (-MCS- multiple classifier system), ensembles de classifieurs, comités, fusion, combinaison, synergie, intégration, concaténation, emerging, etc. Tous font appel plus ou moins au même concept mais sont toutefois ressentis différemment. Ils ont fait l'objet d'études approfondies ces dernières années.

En fait, l'idée principale de la fusion de données est de combiner différentes informations relatives à un problème. Les informations fusionnées pouvant être de natures très variées. Elles peuvent être des données provenant de sources différentes ou bien présentant des caractéristiques différentes, extraites des mêmes données initiales. Pour cette raison, nous trouvons souvent dans la littérature le terme fusion d'informations au lieu de fusion de données. Dans un sens beaucoup plus large que la fusion d'informations, il a été suggéré d'utiliser le terme combinaison d'informations. La combinaison n'est pas définie comme un terme opposé à la fusion, mais elle est simplement plus générale [2]. Elle est utilisée souvent pour décrire des processus et des méthodes d'une manière générale, sans tenir compte des détails. L'utilisation du terme combinaison d'informations a suggéré dans un sens beaucoup plus large que la fusion d'informations.

Une synthèse de l'ensemble des terminologies et des définitions de fusion (ou combinaison), ainsi que les techniques utilisées, sont données dans [3]. Différentes architectures de fusion de données sont présentées dans [4] avec une étude historique et une comparaison qui met en évidence les avantages et les inconvénients de chacune.

Depuis les années 90, la combinaison de classifieurs a donc été une des directions de recherche les plus soutenues dans le domaine de la RdF. Les méthodes de combinaison ont ainsi été appliquées avec succès dans des domaines aussi divers que la reconnaissance de l'écrit, la vérification de signatures, la reconnaissance de la parole, l'identification de formulaires, l'identification de visages ou encore l'analyse d'images médicales. L'amélioration de performances des systèmes de reconnaissance est finalement le principal enjeu des recherches menées ces dernières années sur les systèmes de combinaison [5]. Cette augmentation de performance peut se traduire par plus de fiabilité dans les réponses, moins de rejet, ou bien les deux en même temps. Mais les résultats obtenus restent étroitement dépendants des applications traitées et par conséquent sont difficiles à généraliser en dehors d'un contexte applicatif donné. Pour

L'évaluation des méthodes de combinaison il existe certains travaux qui la teste sur différentes bases de données réelles.

Il existe une autre raison pour l'utilisation d'un comité de classifieurs. Le vecteur de caractéristiques peut être constitué de composants provenant de domaines très divers. Certaines peuvent être des quantités statistiques, telles que des moments, et d'autres des descripteurs structurels discrets, tels que le nombre de points de terminaison, les boucles, etc. En conséquence, il peut ne pas être raisonnable du tout de concaténer toutes les caractéristiques en un seul vecteur de caractéristiques et d'utiliser un seul classifieur. [6]

Trois mécanismes fondamentaux régissent la création des MCS : la génération des classifieurs individuels, la sélection des plus diversifiés et finalement leur fusion pour former des MCS.

Différentes approches permettant de combiner un ensemble de classifieurs comme l'approche parallèle, série et hybride. L'approche parallèle est celle qui a donné lieu aux travaux les plus importants. Sa capacité à exploiter les réponses des classifieurs à combiner, son efficacité prouvée dans de nombreux problèmes de classification et sa simplicité de mise en œuvre expliquent son succès notamment par rapport à l'approche séquentielle qui, elle, nécessite obligatoirement une bonne connaissance du comportement de chacun des classifieurs pour pouvoir obtenir un schéma de coopération efficace.

Dans le présent travail, il s'agit de combiner les réponses des classifieurs en proposant des schémas de combinaison parallèle, d'où le terme « combinaison de classifieurs » utilisé dans le reste de ce travail de recherche.

Le but de ce travail est l'évaluation d'un système multi-classifieur (MCS) dans le problème de la reconnaissance des chiffres imprimés et manuscrits en utilisant différents paramètres caractéristiques (qui prennent une grande part de cette thèse). Dans le cadre des travaux de cette thèse, l'objectif est le développement d'un système multi-classifieurs et multi-paramètres caractéristiques. En d'autres termes, nous cherchons à caractériser expérimentalement l'efficacité des paramètres caractéristiques, les différents classifieurs indépendamment et l'ensemble des classifieurs. Nous nous attacherons à répondre aux points suivants :

- Améliorer la performance de la reconnaissance des chiffres.
- Répondre à l'invariance aux transformations géométriques.
- Prendre en considération la réduction du problème de la grande dimension des paramètres caractéristiques (curse of dimension)

- Proposer de nouveaux paramètres caractéristiques.

Afin de faciliter la compréhension de notre travail, le schéma de la figure 1 présente l'organisation de cette thèse, avec une description détaillée donnée dans le tableau 1 :

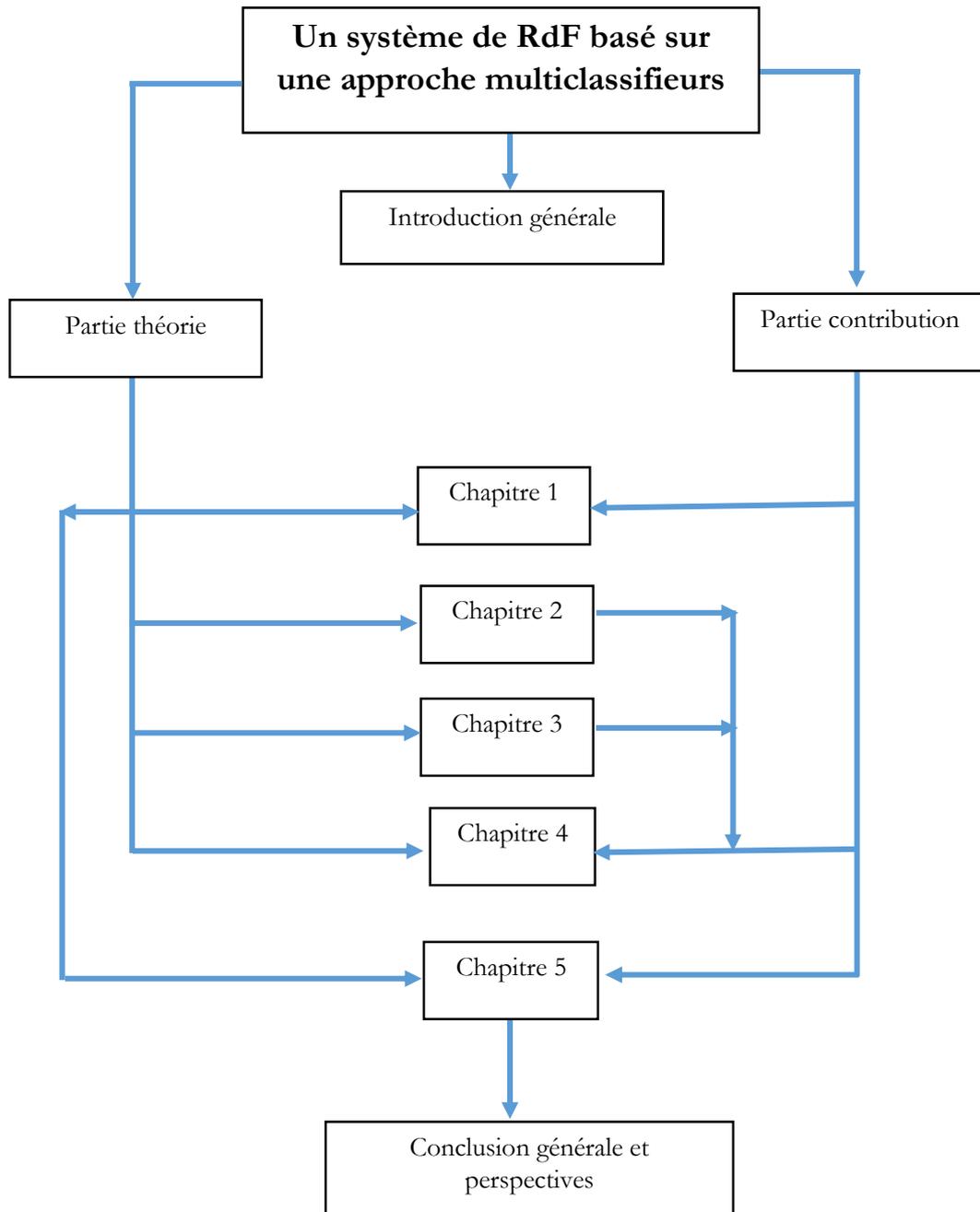


Figure1 : Structure de la thèse

Tableau 01 : Description de la structure de thèse.

Chapitre	Titre	Description
	Introduction générale	Cette introduction est dédiée à la présentation de la problématique et la structure de la thèse
Chapitre 1	Etat de l'art sur la reconnaissance des chiffres	Ce chapitre est réservé à une revue de la littérature sur la reconnaissance des chiffres manuscrits et imprimés, qui sont les deux principaux domaines liés à nos recherches. Après un bref historique de la reconnaissance des caractères, le problème de reconnaissance des chiffres imprimés isolés puis les chiffres manuscrits est abordé. Nous présentons enfin la reconnaissance des chiffres par un MCS.
Chapitre 2	Le système de reconnaissance des formes	Ce chapitre est consacré à la description de système de RdF par un rappel sur les étapes, les notions et les définitions de bases d'un tel système. Les deux approches principales à savoir structurelle et statistique de chaque étape : l'extraction des paramètres caractéristiques et la classification sont développés.
Chapitre 3	Le système multi-classifieurs	La problématique de la combinaison a conduit à la construction de nombreux schémas de combinaison. Les méthodes de combinaison utilisées sont évoquées. Cette démarche vise à établir un système plus performant pour augmenter le taux de reconnaissance des chiffres. Un récapitulatif des principales stratégies de combinaison (approche séquentielle, parallèle et hybride) et les taxonomies d'un MCS ainsi que les différentes méthodes permettant de combiner les décisions des classifieurs sont présentés.

Chapitre	Titre	Description
Chapitre 4	Présentation du système développé	Un prototype de système implémenté est introduit pour l'étude de cas de la classification des chiffres dans ce chapitre. Ainsi, les principes des méthodes utilisées sont présentés. Il s'agit de l'extraction de paramètres caractéristique, et les classifieurs avec les méthodes de combinaison. Les étapes du système sont expliquées dans l'ordre du traitement. Certaines options liées à la mise en œuvre sont abordées.
Chapitre 5	Analyse et discussion des résultats expérimentaux de l'approche développée	Ce chapitre est consacré à la présentation des différentes étapes de notre application avec une analyse des résultats obtenus et une évaluation des méthodes de combinaison utilisées. Nous montrerons comment il est possible de réduire considérablement la complexité du système de reconnaissance mis en œuvre. Nous montrons la difficulté d'établir des résultats généraux indépendamment du domaine applicatif visé.
	Conclusion générale et perspectives	Nous concluons sur les points importants du document, notamment sur l'approche de simulation proposée et les résultats obtenus (la synthèse de nos contributions). Nous parlons des perspectives de ces travaux et les voies de recherche qui semblent prometteuses aussi bien pour améliorer l'approche de simulation proposée que pour la problématique de l'évaluation des performances des méthodes de combinaison.

Chapitre 1 .

ETAT DE L'ART SUR LA RECONNAISSANCE DES CHIFFRES

1.1. Introduction

1.2. Bref historique

1.3. Reconnaissance de caractères

1.4. Domaines d'application de la reconnaissance des chiffres

1.5. Reconnaissance des chiffres imprimés

1.6. Reconnaissance des chiffres manuscrits

1.7. Reconnaissance des chiffres par combinaison des classifieurs

1.8. Conclusion

Ce chapitre présente un aperçu et une vue d'ensemble des divers aspects de la reconnaissance de caractères en général. Il est dédié en particulier à une revue bibliographique sur la reconnaissance hors ligne des chiffres manuscrits et imprimés. La présentation a principalement porté sur l'extraction des paramètres caractéristiques et les classifieurs les plus utilisés par les chercheurs dans la littérature. Les développements récents dans le domaine sont alors détaillés.

1.1. Introduction

Un système de reconnaissance de l'écriture doit idéalement, localiser, reconnaître et interpréter n'importe quel texte ou nombre écrit sur un support de qualité arbitrairement variable tel que des cartes, des formulaires, des agendas, des vieux manuscrits, etc... . Ce type de système aide à convertir l'information contenue dans les vieux manuscrits pour être enregistrée dans des bases de données que tout le monde peut interroger. Néanmoins, on s'est vite aperçu de la difficulté de réaliser des systèmes ad hoc capables de reconnaître tout type de texte et on s'est alors tourné vers des systèmes dédiés, où l'on connaît a priori le style d'écriture que le système doit traiter. Par exemple, il existe des systèmes pour la lecture automatique des chèques bancaires, la reconnaissance du code et de l'adresse postale, la lecture des formulaires d'impôts, etc.... [7]

De nombreuses approches pour résoudre le problème de la reconnaissance des chiffres ont été proposées ces dernières années en raison de ses nombreuses applications possibles. Il est difficile d'établir une taxonomie de ces approches car leurs méthodologies se chevauchent. Toutefois, les recherches dans ce domaine ont essentiellement porté sur : a) les méthodes d'extraction de caractéristiques ; b) méthodes de classification; et c) des architectures de système basées sur différentes stratégies, telles que des combinaisons de plusieurs classifieurs, l'utilisation de plusieurs modèles.

Dans ce chapitre, nous abordons les deux principaux problèmes proposés à savoir la reconnaissance des chiffres manuscrits et imprimés isolés. Dans un premier temps, nous donnons un bref historique sur la reconnaissance des caractères. Nous présentons ensuite les principales recherches qui ont permis d'améliorer les performances de reconnaissance des chiffres manuscrits et imprimés isolés séparément. Certains travaux récents sont brièvement décrits en termes de caractéristiques, de types de classifieurs, de bases de données de tests et de résultats. De nombreux chercheurs ont essayé d'améliorer les performances des systèmes de reconnaissance par exploitation de différentes architectures des systèmes multiples experts (classifieurs).

1.2. Bref historique

La reconnaissance automatique des textes imprimés et manuscrits était considérée comme un objectif important et ambitieux avant même l'avènement des ordinateurs électroniques modernes. Le premier chemin menant aux systèmes modernes a été réalisé par Carey en 1870, quand il a inventé ce qu'il a appelé le scanner de rétine. En 1890, Nipkow développa le premier appareil à balayage séquentiel, et des appareils de lecture [6]

La reconnaissance automatique du texte manuscrit et imprimé est un domaine qui a rapidement intéressé les chercheurs. Selon [8], l'évolution des systèmes de reconnaissance de caractères est passé par trois âges : les premiers âges, les développements et les progrès.

- Les premiers âges (1900-1980) : À la base, l'intérêt a commencé en 1900. Mais, l'âge d'or a débuté avec l'invention du premier ordinateur en 1946. Les premiers dispositifs de reconnaissance de caractères sont apparus au milieu des années 40 avec le développement des ordinateurs numériques. À cette époque, les œuvres étaient concentrées soit sur un texte imprimé à la machine, soit sur un petit ensemble de textes manuscrits ou de symboles bien distingués.

Quelques années plus tard, les premières expériences en reconnaissance de caractères ont pu être réalisées. Des algorithmes efficaces mais contraignants ont été mis en œuvre principalement pour les caractères et les chiffres latins. Les dispositifs de reconnaissance de caractères commerciaux étaient disponibles dans les années cinquante, lorsque les tableaux électroniques capturant les données de coordonnées x-y du mouvement de la pointe du stylo furent introduits pour la première fois. Pendant les années soixante et soixante-dix, les premiers systèmes de lecture automatique du texte imprimé (quelques fontes seulement) ont vu le jour. Dans la même période, un des premiers systèmes de reconnaissance de l'écriture manuscrit a été introduit. Les techniques de classification mises en œuvre durant cette période font appel à des méthodes d'appariement syntaxique (la plupart des méthodes statistiques n'étant pas encore connues).

- Les développements (1980 - 1990) : Avec l'explosion des technologies de l'information, et l'accès facile aux ordinateurs personnels, des approches structurelles ont été lancées dans de nombreux systèmes en plus des méthodes statistiques. La recherche sur la reconnaissance des caractères portait essentiellement sur les techniques de RdF, sans aucune information sémantique. Cela a conduit à une limite du taux de reconnaissance qui n'était pas suffisant dans de nombreuses applications.

Durant la même période, les réseaux de neurones sont abandonnés rapidement, suscitent un vif intérêt de la communauté des chercheurs grâce à l'algorithme de rétro-propagation du gradient. Ainsi, le perceptron multicouche a été rapidement reconnu comme le classifieur par excellence dans beaucoup de problèmes de reconnaissance de caractères. Un peu plus tard, nous avons vu naître aussi d'autres architectures plus complexes ou l'extraction de caractéristiques est intégrée dans le processus d'apprentissage du réseau comme dans Le-net 4 et Le-net 5 [7].

- Les progrès réels dans la reconnaissance des caractères ont été réalisés après 1990. Au début des années 90, le traitement des images et la RdF étaient associés de manière efficace aux techniques de l'intelligence artificielle. Des outils efficaces tels que les réseaux de neurones, les machines à vecteurs de support, les modèles de Markov cachés, le raisonnement flou et le langage naturel associé à des ordinateurs plus puissants et des équipements électroniques plus précis ont donné des résultats assez satisfaisants pour des applications restreintes. Cependant, il reste encore beaucoup à faire pour atteindre le but ultime de la simulation par machine de la lecture humaine, en particulier pour l'écriture manuscrite sur ligne sans contrainte. [8].

Une excellente synthèse [9] résume et compare certaines des méthodes bien connues et utilisées dans toutes les étapes d'un système de RdF, et l'identifie les sujets de recherche et les différentes applications de ce domaine. Comme il existe un riche état de l'art présentant un état d'avancement du domaine de la reconnaissance de caractères [10, 11, 12, 13] et les chiffres [14].

1.3. Reconnaissance de caractère

La reconnaissance des caractères, en général, signifie la reconnaissance de toute forme d'information textuelle ou symbolique soit imprimé ou écrit, y compris, par exemple, des formules mathématiques. La reconnaissance du texte manuscrit est généralement considérée comme une tâche plus exigeante que la reconnaissance du texte imprimé (machine printed) pour laquelle il existe depuis longtemps des produits commerciaux opérationnels.

Les applications de ce domaine sont tous des systèmes capables de localiser, reconnaître et interpréter tous les textes écrits dans n'importe quel document papier ou produit ou autre. La complexité du système peut être considérablement réduite si la tâche est limitée à un domaine précis.

Les utilisateurs potentiels des systèmes de reconnaissance de l'écriture se trouvent principalement dans différents secteurs comme les autorités gouvernementales, telles que les services des postes pour la lecture automatique des adresses et des impôts, et les grandes entreprises privées comme les banque pour le traitement de chèques.

Différents domaines de reconnaissance des caractères se chevauchent. Afin de bien comprendre un cas particulier, il est nécessaire d'avoir une vue d'ensemble du domaine. Les applications de la reconnaissance de caractères peuvent varier selon la nature de l'écriture et son support de saisi.

La reconnaissance des caractères, imprimés ou manuscrits, ou informations symboliques incluant par exemple les formules mathématiques [15], signifie le décodage du texte. La figure 1.1 présente une classification des différents types d'écriture [6, 7, 8] faite selon le média, le support de saisi et l'application considérés. La distinction entre le texte imprimé et le texte manuscrit bien qu'évidente n'est pas toutefois exclusive. Un système qui reconnaît du manuscrit doit être capable de reconnaître de l'imprimé. Aussi, certaines fontes imprimées miment l'écriture manuscrite. Mais, étant donné que les caractères manuscrits sont sujets à plus de bruit que ceux imprimés, la reconnaissance du manuscrit est considérée plus difficile que la reconnaissance de l'imprimé pour lequel des systèmes opérationnels commerciaux existent déjà.

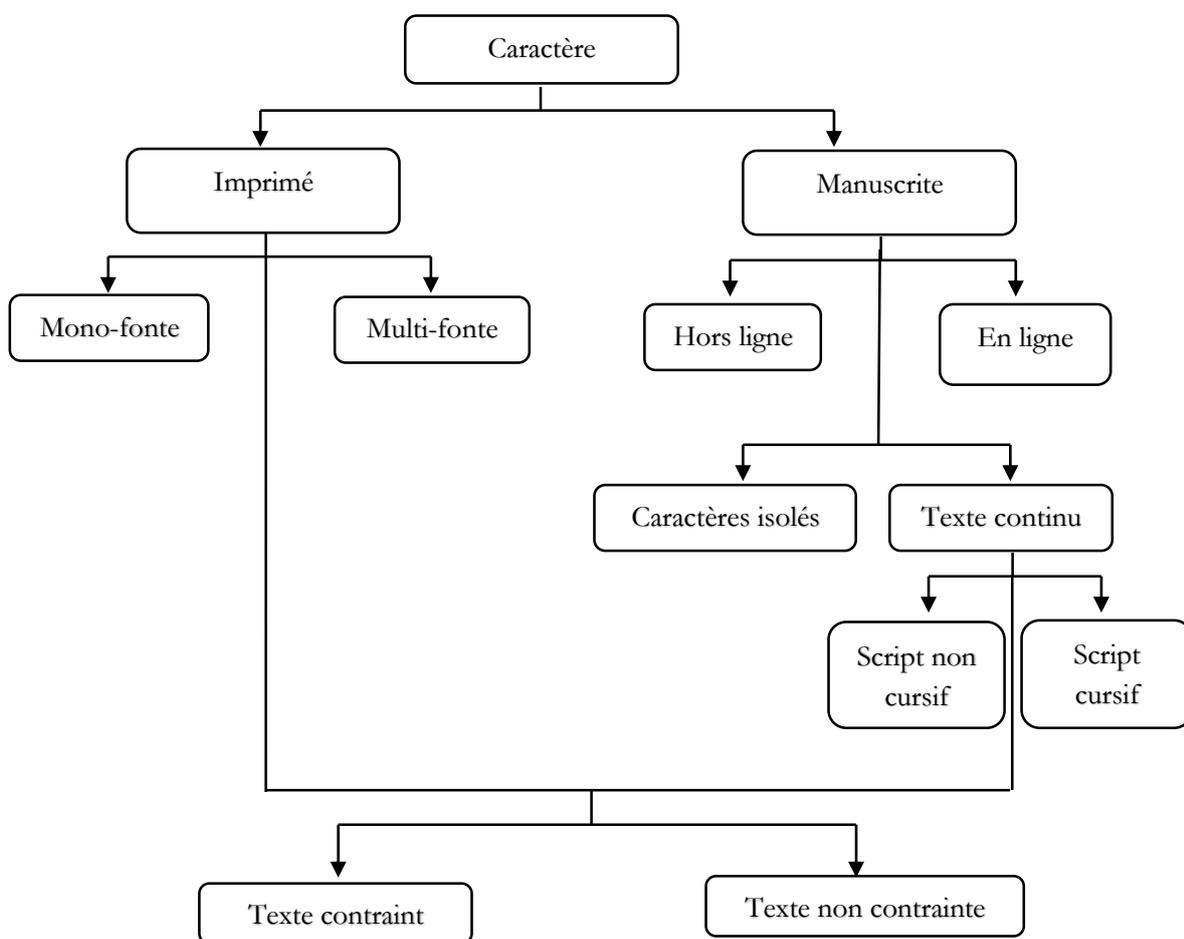


Figure 1.1 : Classification des types d'écriture

La saisie des caractères manuscrits est faite selon deux façons : en ligne ou hors-ligne. Il existe une autre distinction à faire entre les caractères (texte) isolés et le texte continu qui nécessite une phase de segmentation. Le texte continu englobe le script non cursif et le script cursif (Block character) contenant des caractères pouvant se toucher mais n'affectant pas leurs caractères voisins.

Elle dissocie aussi le texte contraint de celui qu'il ne l'est pas. Le premier est le cas quand le scripteur est appelé à écrire dans des zones prédéfinies. Le système dispose alors d'une information a priori lui permettant de pouvoir détecter et extraire l'information plus aisément. La reconnaissance de caractères isolés peut être perçue comme un cas spécial de cette branche. Le script non contraint peut par contre prendre n'importe quelle forme selon n'importe quelle direction et selon n'importe quel style. Ceci ramène au problème de l'analyse de documents qui constitue une branche à part du domaine.

En pratique, le texte est dit non contraint s'il contient plus qu'une ligne et lorsque le nombre de mots par ligne n'est pas connu. Des exemples de textes sous contrainte, sans contrainte, isolé et continu illustrés à la figure 1.2.

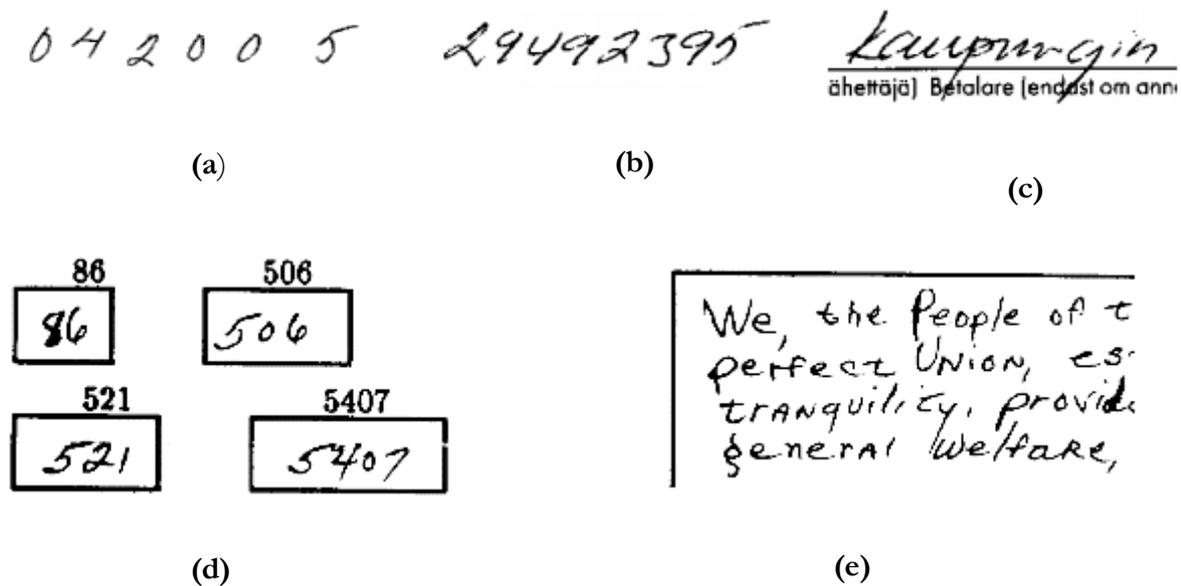


Figure 1.2 : Exemple de différents types d'écriture (a) Caractères isolés, (b) Texte continu (c) Script cursif, (d) texte contraint, (e) texte non contraint. [6]

1.4. Domaines d'application de la reconnaissance des chiffres

Dans notre étude, nous nous intéressons à la reconnaissance hors-ligne de chiffres connus de « 0 » à « 9 » soit imprimé ou manuscrit, parfaitement séparés les uns des autres (isolés). Il s'agit d'un domaine important en raison de ses vastes applications pratiques. Les applications de ce domaine sont très demandées dans le monde réel. A titre d'exemple, nous pouvons citer :

- Le traitement des montants numériques [16] et celui des dates sur les chèques [17, 18, 19].
- Les codes sur les enveloppes pour le traitement ou tri automatique du courrier [20].
- Les systèmes aussi apportent une aide précieuse aux handicapés visuels.

- La reconnaissance des chiffres et des nombres dans des images de scènes naturelles comme les numéros des maisons [21, 22].
- La reconnaissance automatique des plaques d'immatriculation (ANPR –Automatic Numeral -plate recognition-) a de nombreuses applications telles que (perception du péage électronique sur l'autoroute, applications dans le domaine de la police, application de la vitesse moyenne, application de la voie de bus, systèmes de gestion de stationnement, systèmes de temps de parcours, détection de véhicules volés, etc.). [23, 24, 25, 26, 27, 28].

Le système de reconnaissance des chiffres peut être comme un système embarqué dans les systèmes qui détectent les panneaux routiers (la reconnaissance des panneaux de signalisation est un domaine d'application récurrent pour la détection d'objets visuels) et lit les panneaux de limitation de vitesse pour limiter la vitesse de véhicule équipé d'une caméra située en haut du pare-brise à la vitesse lue par la caméra, en appuyant sur un bouton de la commande du limiteur/régulateur de vitesse [29, 30, 31].

Cependant, dans de nombreux cas, les chiffres apparaissent sur des surfaces non planes, telles que la surface sphérique d'une balle dans un jeu interactif homme-machine [32]

Les applications de reconnaissance optique de caractères (OCR) sont presque toujours des systèmes embarqués dans lesquels l'aspect du temps réel est essentiel. Par conséquent, l'équilibre entre la performance de reconnaissance en terme de taux de reconnaissance (la précision) et le temps de calcul disponibles pour chaque tâche de classification doit être soigneusement ajusté. Heureusement, de nombreuses méthodes d'extraction des caractéristiques et de classification statistiques sont assez simples et peuvent être implémentées avec un matériel spécifique.

Dans la littérature nous avons trouvé un autre sujet de recherche important dans le domaine de DAR (Document Analysis and Recognition) la reconnaissance des chaînes numériques (chaînes de chiffres). Comme exemple les dates, code barre, matricule. Vous trouverez un état de l'art sur la segmentation et la reconnaissance de chaîne de chiffres dans [1, 8, 16, 17, 18, 19, 33, 34, 35, 36]. Diverses approches et techniques ont été développées pour résoudre le problème de la segmentation et de la reconnaissance de chaînes de chiffres. Les références [17, 18] présentent un système hybride HMM-MLP pour la segmentation et la reconnaissance de l'écriture manuscrite non-contrainte et plus particulièrement des dates présentes sur les chèques brésiliens.

Vous trouverez aussi dans la littérature des travaux sur la reconnaissance des chiffres on ligne basé sur les mouvements du stylo (pen tip) i.e. la suite ordonnée des déplacements de la plume (les informations de mouvement) sur la surface d'écriture (comme exemple pen-based computer

screen surface) [36]. En général la reconnaissance hors ligne est considéré la tâche la plus difficile, parce qu'il n'est pas possible de construire les informations de mouvement en utilisant seulement l'image.

Plusieurs études de recherche ont été menées pour reconnaître d'autres scripts (Les chiffres d'autre langues) tels que Devanagari (Marathi, Modi) [37, 38, 39, 40, 41, 42]. Dans [43] une étude détaillée de la reconnaissance des caractères et des chiffres manuscrits Urdu. Farsi (Persian) [44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55]. Bangla [56, 57, 58]. Indien (Arabic orienté) [50, 59, 60, 61, 62, 63, 64, 65], Kannada [66, 67, 68, 69, 70], Gurmukhi [71]. Et d'autres scripts : Le Chinois, le Japonais, Oriya, Gujarati, Tamil, Malayalam et Manipuri, etc Un nouveau paradigme proposé dans [72] pour la reconnaissance des chiffres manuscrits Odia basée sur l'effet de supériorité de la configuration gestalt, a donné des résultats semblent prometteurs.

Il y a des travaux traitant différents scripts. [73, 74] qui présentent une étude sur les chiffres Indo-Arabic, Bangla, Devanagari, Roman (Arabic ou anglais), and Telugu. [75] Kannada, Telugu et Devanagari. [76] Kannada, English (Roman ou Arabic).

1.5. Reconnaissance des chiffres imprimés

Il existe de nombreuses recherches sur la reconnaissance des chiffres arabe imprimés, mais sur des bases de données différentes, à cause de l'absence d'une base standard d'une part et qui contient des chiffres multi-font d'une autre part. Nous présentons dans ce qui suit quelques travaux.

Un système de reconnaissance des chiffres imprimés dans une variété de styles et de tailles de polices (10 polices) est conçu dans [77]. La classification par arbre de décision atteint plus de 97% après un apprentissage sur 160 image et 100% après un apprentissage sur 1560 images.

Le travail présenté dans [78] porte sur la reconnaissance des chiffres manuscrits et imprimés (600 chiffres) basée sur une version améliorée de la méthode de caractéristique de Loci (locus) pour extraire les caractéristiques. La reconnaissance est réalisée par KPPV et le PMC, le résultat obtenu est 99%.

Dans [79] une approche invariante au décalage et échelle est implémentée pour la reconnaissance des chiffres. Cette approche consiste à utiliser un facteur de corrélation entre les images de référence et l'image testée. L'approche proposée permet d'atteindre des taux de reconnaissance très élevés. Bien que cette technique de reconnaissance proposée soit appliquée à

la reconnaissance de chiffres, elle peut être généralisée à la reconnaissance de tout chiffre dans n'importe quelle langue et peut être étendue au texte également.

Néanmoins, cette approche n'est pas invariante à la rotation, et les autres travaux ne sont pas invariants aussi au décalage et échelle.

Les auteurs dans [21] classifient des chiffres obtenus à partir de numéros de maison disponible dans la base de données SVHN. La classification est réalisée par les réseaux CNN. La précision obtenue est de 95.10%.

L'auteur dans [80] propose une nouvelle technique de classification hybride permettant de reconnaître les chiffres imprimés (700 échantillons). L'extraction des caractéristiques a été réalisée à l'aide d'une analyse du contour, d'un Descripteur de Fourier (FD) et d'un algorithme basé sur le code chaîne. Un nouvel algorithme basé sur le code chaîne a été proposé pour extraire les caractéristiques de la courbe à partir des images. La reconnaissance a été réalisée à l'aide d'un PMC. Une précision supérieure de 99% a été obtenue.

Dans [81] des techniques d'extraction statistique (multi-zonage), géométrique (distance, angle) et directionnelle (code de chaîne) ont été appliquées sur les chiffres. La classification a été réalisée à l'aide d'un réseau PMC et un classifieur SVM. Le taux de reconnaissance atteint est de 98%. Une petite base des chiffres imprimés et manuscrits (590 chiffres) utilisée dans [52] pour la reconnaissance en utilisant les profils avec les classifieurs KPPV et PMC. Le résultat obtenu est de 86.8%.

Vous trouverez dans [82] l'ensemble des travaux récents de recherche pour la reconnaissance des chiffres de la base SVHN.

Les recherches sur la reconnaissance des chiffres arabe imprimés ont donné des performances élevées en terme de taux de reconnaissance. Néanmoins elles ont utilisé des petites bases de données, ne dépassent pas 2000 échantillons, et différentes avec un nombre limité de fonts (Polices). L'absence d'une base standard nous a conduit à créer une base de chiffres omni-fonts qui contient plus de 6000 chiffres. Cette base nous permet de trouver des résultats plus généraux en utilisant les mêmes méthodes (classifieurs et paramètres caractéristiques) d'une part, et d'appliquer notre système proposé sur une base plus large d'autre part.

Avant les chercheurs se sont concentré sur le script arabe (roman ou anglais) qui est le plus utilisé dans le monde, et quelques autres scripts relatifs au langage d'Asie comme le Chinois, le Coréen, le Japonais. Mais récemment les scripts du langage Indien ont attiré l'attention des chercheurs comme Devanagari, Tamil, Oriya et Bangla.

Dans la littérature, il y a plusieurs travaux de recherche intensifs sur les chiffres imprimés pour certaines langues populaires tel que : le Persan (Farsi) [55, 83, 84], l'Assamese [85], le Bangla [58], l'Odia [86], le Gurumukhi [87], l'Indien [88, 89].

Une revue de la reconnaissance des chiffres : Arabe, Farsi et Urdu est présentée dans [90]. La plupart des articles publiés depuis 2003 et des publications récentes en arabe et en farsi ont été résumés dans cette étude avec les paramètres caractéristiques et les classifieurs utilisés. Une autre revue des travaux récents dans le domaine de la classification et de la reconnaissance des chiffres imprimés et manuscrits est présenté dans [91].

1.6. Reconnaissance des chiffres manuscrits

Le domaine de la reconnaissance des chiffres manuscrits a été intensément étudié pendant de nombreuses années. De nombreux résultats ont été obtenus par les chercheurs durant les dernières années en proposant de nombreux systèmes et algorithmes de classification. Des techniques allant des méthodes statistiques telles que la PCA et l'analyse discriminante de Fisher, les K voisins les plus proches (KPPV) vers l'apprentissage automatique, telles que les réseaux de neurones ou les machines à vecteurs de support SVM dont la performance égale ou dépasse celle des classifieurs classiques.

Les algorithmes génétiques et les réseaux de neurones convolutionnels (CNN) plus sensibles aux propriétés topologiques de l'entrée, ont été appliquées pour résoudre le problème de la reconnaissance de chiffres manuscrits. Les algorithmes d'apprentissage et de reconnaissance des HMM ont été largement appliqués à la reconnaissance de mots manuscrit [92, 93, 94] et à leurs applications à la reconnaissance des chiffres manuscrits ont évolué [36, 95].

Le facteur le plus important pour obtenir des performances élevées est le choix des paramètres caractéristiques. Pour cette raison, une attention considérable a attiré les chercheurs. Une bonne étude sur l'extraction des paramètres caractéristiques peut être trouvée dans [96]. La littérature sur la reconnaissance des chiffres manuscrits nous montre essentiellement deux classes de caractéristiques. Ce sujet fait l'objet d'un chapitre de notre thèse. Vous trouverez une présentation détaillée dans le chapitre 2. La performance de classification est signalée par des publications récentes précisément sur plus de cinq base de données (MNIST, NIST, CENPARMI, CIFAR10, CIFAR100, SVHN, USPS) et d'autres. Nous présentons maintenant quelques travaux vis-à-vis de leurs performances.

Dans [97] les auteurs ont décrit une méthode pour la reconnaissance des chiffres manuscrits sans contraintes. Au début, les tracés du caractère entrée sont convertis en polygones, puis ils

sont traités par un analyseur syntaxique général dont la sortie est une représentation de haut niveau en termes de traits, arcs quadratiques, angles, etc., de la même manière qu'un observateur humain pourrait le faire. Ensuite, un certain nombre d'expressions régulières sont défini et caractérisent chaque classe de symboles. La reconnaissance finale est exécuté par un automate fini. La méthode a été testée sur "IEEE Base de données 1.2.2" avec un taux de réussite d'environ 95%.

L'auteur dans [7] a proposé une méthodologie de sélection de modèle automatique du SVM en minimisant l'erreur quadratique entre les probabilités à posteriori des classes et les probabilités désirées des données. L'approche permet d'optimiser les paramètres de noyaux et de réduire efficacement la complexité du classifieur en minimisant le nombre de vecteurs de support. Ceci s'accompagne d'une réduction drastique de l'erreur de généralisation. L'auteur a utilisé la base des chiffres arabe USPS et la base des chiffres indien INDCENPARMI. Les meilleurs taux d'erreur obtenus sont 4.2%, 4.4% et 5.3% pour les noyaux KMOD, RBF et polynomial respectivement sur USPS. Les résultats sur la base USPS de 1989 à 1998 sont rassemblés dans le tableau 1.1.

Tableau 1.1 : Résultats de référence sur la base USPS dans [7] 2004

Classifieurs	Erreur
Humain	2.5%
Tangent distance	2.6%
Perceptron à 2 couches	5.9%
LeNet1	5.0%
SVM Poly	4.0%
SVM tanh	4.1%
SVM RBF	4.3%
SVM virtuel RBF	3.2%
SVM virtuel Poly	3.2%
Noyau local	3.0%
k-means RBF	6.7%
Réseau RBF	4.9%
PMC	5.9%
SVM kmod 2004	4.2%
SVM RBF 2004	4.4%
SVM poly 2004	5.3%

Pour les performances des systèmes publiés avant 1996, vous pouvez vous reporter aux références [98, 99]. L'auteur dans [8] rassemble certains résultats rapportés dans la littérature de 1996 à 2002 sur CENPARMI, NIST, CEDAR, MNIST. Dans [100] l'auteur a donné les résultats de 1996 à 2004 sur les base NIST, CENPARMI, MNIST, avec différent paramètres caractéristiques. Des résultats de 1994 à 1998 sont présentés dans [1] sur les base CEDAR, NIST ERIM (Environmental Research Institute of Michigam), Chèques bancaires brésiliens et autre.

Il est important de souligner que les taux de reconnaissance ne sont pas directement comparables, car ces résultats sont basés sur différentes bases de données et différentes techniques.

Dans [101] l'auteur utilise des caractéristiques structurelles basées sur la concavité. Elles sont utilisées comme caractéristiques complémentaire pour les caractéristiques de gradient et de codage en chaîne. Deux classifieurs de vecteur de support (support vector classifieur SVC) sont choisis, SVC avec noyau de base radiale (RBF), SVC avec noyau polynomial et KPPV. Les résultats sont obtenus sur les bases de données MNIST, USPS et DIGITS. La méthode a donné les meilleurs résultats dans la majorité des tests.

Une nouvelle méthode de classification est proposée dans [102]. Elle est basée sur des paramètres caractéristiques différents comme : le nombre de contours, le squelette, le nombre de Watershed (cavités), le ratio entre le nombre de pixels de la partie supérieure et la partie inférieure, en utilisant KPPV, SVM et une méthode (classifieur) proposée. Cette méthode a donné de bons résultats sur les bases MNIST, CEDAR, CEPARMI.

La reconnaissance de l'écriture a été l'un des premiers domaines d'application des réseaux de neurones. Ils ont été largement utilisés dans ce domaine et des résultats prometteurs ont été obtenus, notamment en ce qui concerne la reconnaissance des chiffres manuscrits [17]. Jusqu'à très récemment, le perceptron multicouche (PMC) constituait le modèle de prédilection pour traiter diverses tâches d'apprentissage supervisé. Ce modèle est facile à construire et à entraîner, et a été efficacement adopté dans beaucoup de systèmes de reconnaissance de caractères. Nous nous intéressons ici plus particulièrement à la reconnaissance de chiffres manuscrits. Des méthodes neuronales performantes ont déjà été proposées pour réaliser cette tâche [100, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115].

Une étude récente réalisée dans [116], détaille les techniques existantes utilisées pour la reconnaissance des chiffres manuscrits (HandWritten Digit Recognition HWDR). Cette étude est nouvelle dans la mesure où elle se concentre sur HWDR et ne traite que de l'application du

réseau de neurones et de ses algorithmes modifiés. Elle présente une étude détaillée de l'utilisation de NN et de ses variantes pour la reconnaissance des chiffres. Pour chaque travail existant, elle élabore les étapes, la nouveauté, l'utilisation des jeux de données, ainsi que les avantages et les inconvénients. De plus, elle présente une analyse scientométrique de HWDR qui présente les principales revues et journaux dans ce domaine de recherche (Pattern Recognition, International Journal on Pattern Recognition and Artificial Intelligence, Pattern Analysis and Application), et présente également des défis de recherche et des travaux futures potentiels.

L'auteur dans [100] a utilisé le PMC avec les Wavelets et le zonage fixe et adaptatif sur les base NIST, CENPARMI, le meilleur résultat est 99.07%.

Dans [117] l'auteur présente une approche basée sur différentes techniques d'apprentissage automatique. Les classifieurs MLP, SVM, Naïve Bayes (NB), Réseau Bayésien, Random Forest, J48 et Random Tree, ont été utilisées à l'aide de WEKA (Waikato Environment for Knowledge Analysis). Les résultats obtenus sur un ensemble de données fourni par l'Institut autrichien de recherche sur l'intelligence artificielle (contenant 1893 échantillons pour l'apprentissage et 1796 échantillons de test) sont respectivement : 90.37%, 87.97%, 81.85%, 84.35%, 85.75%, 79.51%, 85.6%.

Malgré les bons résultats obtenus par ces travaux de recherches en utilisant les différents classifieurs, ils sont basés sur des paramètres caractéristiques qui ne vérifient pas la propriété de l'indépendance aux transformations géométriques. Les classifieurs ne couvrent pas cette insuffisance quel que soit leurs performances.

Les réseaux de neurones ont donné des résultats remarquables dans ce domaine. Néanmoins la nécessité de performances élevées dans des applications réelles a poussé la recherche vers des modèles de classification de plus en plus complexes comme les CNN (comportant de nombreuses couches ont un grand pouvoir discriminant). CNN peut extraire les propriétés topologique d'une image par la première couche et classifier la forme par la dernière.

Les deep CNN ont révolutionné la vision par ordinateur et ont fait preuve d'une précision remarquable dans de nombreux problèmes de classification des modèles visuels et peuvent désormais dépasser la vision humaine dans de nombreuses tâches de reconnaissance d'objets avec des images naturelles [118, 119]. Au cours des dernières années, les CNN avec leurs différents modèles (RCNN, Maxout, NIN, DSN, SNN, DNN, Dropconnect, Dropout, etc.) ont remporté un grand succès dans de nombreuses tâches de vision par ordinateur y compris la reconnaissance des chiffres manuscrits [3, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133,

134, 135, 136, 137]. Des performances à la pointe sur de nombreuses bases de données des chiffres tels que MNIST (tableau 1.2) CIFAR10, CIFAR100, SVHN) ont été obtenues récemment.

Tableau 1.2 : Résultats de référence de différents types de CNN sur la base MNIST

Référence, Année	Classifieurs	TR
[138], 1998	Le-Net5	99.05%
[139], 2007	SNN	96.5%
[129], 2012	MCDNN	99.77%
[140], 2013	SNN	93.5%,
[141], 2013	SNN	91.6%
[142], 2013	Drop Connect	98.88%
	Dropout	98.80%
[143], 2015	SNN	95%
[144], 2015	CNN	99.31%
[145], 2015	RCNN	99.66%
[146], 2015	Competitive Multi-scale Convolution	99.67%
[147], 2018	SNN	98.4%.
[148], 2018	RMDL	99.76%

En dépit de leurs performances exceptionnelles, la recherche de modèles informatiques inspirés du cerveau se poursuit et attire de plus en plus de chercheurs du monde entier. Poursuivant dans cette voie de recherche, un grand nombre de modèles à bio plausibilité améliorée basés sur des réseaux de neurones à impulsions (SNN Spiking NN) ont émergé. Cependant, les SNN ne sont pas encore compétitifs par rapport aux DCNN en termes de précision de la reconnaissance [118].

Les différents types de CNN en particulier les deep CNN (DCNN) ont une haute performance. Mais puisque Les CNN sont des classifieurs extractible de caractéristique a entraîné, l'exigence d'un grand ensemble de données, ainsi que le coût et la complexité de calcul élevé, limite l'utilisation de CNN. Ils demandent un matériel puissant et des version récente de GPU (Graphics Processing Unit) i.e. ils nécessitent généralement un traitement du processeur graphique au lieu du processeur habituel, ce qui pèse lourdement sur les ressources

informatiques. Ils possèdent des architectures complexes et prennent des heures même des jours pour l'apprentissage.

Nous résumons dans le tableau 1.3 les contributions récentes sur la base des chiffres manuscrits MNIST :

Tableau 1.3 : Résultats de référence des classifieurs individuels sur la base MNIST

Référence, Année	Paramètres Caractéristiques	Classifieurs	TR
[149], 2004	Image	LIRA (Limited Receptive Area)	99.39
[150], 2007	Image	MLP	99
[151], 2010	- Moment Zernike - Caractéristique Gradient	SVM	94.33 99.16
[152], 2011	- GFD (Generic Fourier Descriptor) - R-signature - Zernike - Pixels	SVM-OVA	92.95 79.55 92.25 97.6
[153], 2013	Image	MLP	85.16
[154], 2013	Moment Hu	SVM	63
[155], 2014	HOG	SVM	97.25
[156], 2015	Histogramme de projection	SVM-OVO	99.05
[157], 2017	HOG (Multiple size de cellule)	SVM	99.36
[121], 2018	Image	SNN (Spiking NN) ANN	98.17 98.00
[122], 2018	Image	Dropout Deep Ensemble	99.5 99.3
[158], 2018	- Gradient de Triangle - Ratio	SVM MLP	93.18 96.51
[148], 2018	Image	RDML (Random Multi-model Deep Learning)	79

TR : *taux de reconnaissance en %.*

Certains résultats de la reconnaissance des chiffres manuscrits de la base MNIST De 1998 à 2012 sont disponibles sur la page originale de MNIST [159] sous un tableau, en spécifiant les classifieurs utilisés et les prétraitements appliqués, ces résultats montrent qu'une description des images de chiffre normalisées utilisant le vecteur des pixels binaires (l'image entière) suffit à une reconnaissance supérieure à 95%. Au contraire, notre travail basé sur l'extraction manuelle des caractéristiques et les fournit au classifieur afin de les classifier. Rodrigo Benenson rassemble des résultats supplémentaires des articles plus récents (2002-2016) sur [82].

Afin de réaliser des systèmes plus fiables, de nombreux chercheurs se sont tournés vers la combinaison de caractéristiques statistiques et structurelles [95, 112, 160, 161].

Les contributions présentées ci-dessus et dans les tableaux précédents montrent la variation des performances. Les plus élevées sont obtenues par les deep CNN et/ou par des classifieurs utilisant l'image (un vecteur de grandes dimension). Cela nous a donné l'idée de maintenir les classifieurs les plus utilisables dans le domaine et d'exploiter les paramètres caractéristiques par concaténation et même hybridation, prenant en considération l'invariance aux transformations géométriques.

1.7. Reconnaissance des chiffres par combinaison des classifieurs

La littérature est très riche et diverse dans le domaine de la combinaison des classifieurs. L'idée de combiner les sorties des classifieurs pour créer un système avec une performance élevée n'est pas nouvelle. En effet, plusieurs travaux spécifiques à la combinaison de classifieurs ont été mentionné par Zouari [5]. Récemment, le concept de combinaison de multiple classifieurs dans le domaine de la reconnaissance de caractères, est proposé comme une nouvelle direction pour le développement de haute fiabilité des systèmes de reconnaissances de caractères. Selon [162, 163, 164] la combinaison des classifieurs complémentaire améliore la performance d'un classifieur individuel.

Plusieurs systèmes de combinaison parallèle ont été développées pour différentes applications de reconnaissance de chiffres. Une combinaison des classifieurs KPPV est proposé pour La reconnaissance des chiffres manuscrits [165,166, 167, 168, 169, 170, 171, 172]. Une sélection statique (SS) et dynamique (DS) et dans un ensemble de classifieur est portée dans [173, 174, 175].

L'approche proposée dans [176] est basée sur une stratégie pipeline (série), utilise trois étages de décision. Les deux premiers étages mettent en œuvre une comparaison directe du chiffre à

reconnaître avec l'ensemble des modèles, permettent de classifier 70 à 80% des chiffres avec un taux d'erreur faible et sont capables de générer des décisions sur les classes d'appartenance des chiffres rejetés. Lorsque ces étages ne peuvent pas conclure, ils fournissent une liste d'hypothèses au dernier niveau de décision pour chercher le modèle dans une liste prédéfinie de modèles syntaxiques.

Une vaste expérience sur la combinaison de classifieurs est rapportée et discutée par les auteurs de [270]. Elle inclut à la fois la combinaison de différents classifieurs sur les mêmes paramètres caractéristiques et la combinaison de classifieurs sur paramètres caractéristiques. Diverses règles de combinaison fixes et paramétrique (par apprentissage) sont utilisées. Il est démontré qu'il n'existe pas de règles de combinaison gagnante, et que les mauvais classifieurs ainsi que les mauvais paramètres caractéristiques peuvent contenir des informations précieuses pour l'amélioration des performances par combinaison. Les meilleures performances sont obtenues en combinant les deux, différents ensembles de paramètres caractéristique et différents classifieurs.

Dans [177] une approche de sélection des classifieurs et des caractéristiques basées sur l'algorithme génétique hiérarchique multi-objectifs a été proposée. En effet, l'algorithme génétique est utilisé dans deux étages. Le premier pour la sélection des caractéristiques les mieux adaptées aux classifieurs utilisés. Le deuxième étage pour chercher le meilleur sous-ensemble de classifieurs parmi les bons (du premier niveau). Pour sélectionner le meilleur ensemble de classifieurs, deux critères ont été utilisés : le taux de reconnaissance de l'ensemble (combinaison par la moyenne) et le désaccord entre les classifieurs. En appliquant cette méthode dans le contexte de la reconnaissance de chiffres manuscrits avec trois classifieurs PMC, il est observé que le sous ensemble qui présente le meilleur taux de reconnaissance est aussi composé du nombre minimum de classifieurs. Mais l'auteur a utilisé des caractéristiques qui ne sont pas invariantes aux transformations géométriques. Nous avons appliqué cette approche sur les chiffres imprimé en considérant l'invariance aux transformations géométriques et en utilisant la sélection statique.

Une combinaison hiérarchique entre le classifieur KPPV et SVM est utilisé dans [178] en exploitant trois types de paramètres caractéristique. Statistique (moyenne, variance et entropie). Physique (Moment d'inertie zonale, Moment d'inertie centrale). Analytique (curvature : haut, bas, gauche, droite). Le taux de reconnaissance achevé est 99.26% sur la base MNIST. Sur un ensemble des chiffres des code ZIP une combinaison hybride de MLP-SVM a donné un taux de

98,01 %. Mais, les caractéristiques utilisées ne sont pas invariantes aux transformations géométriques.

Le tableau 1.4 résume quelques contributions récentes de l'application de MCS sur la base des chiffres manuscrits MNIST :

Tableau 1.4 : Résultats de référence de la combinaison des classifieurs sur la base MNIST

Référence, Année	Paramètres Caractéristiques	Classifieur	TR
[179], 2004	- Projections horizontale, verticale et diagonale. - Paramètre de Kirsch	MLP SVM	99.17
[180], 2010	- Projections - Zoning - Edges - Concavities - Gradient	Multiple MLP	99.68
[131], 2012	Image	CNN SVM	99.81
[38], 2016	Histogramme de projection	SVM Optimisé par l'Algorithme Bat	95.30
[181], 2018	Image	- MLP (61%) - C4.5 (Arbre de decision 88%) - NB (81%) - KPPV (97%) - RF (Random Forest 97%)	98
[120], 2018	Image	CNN SVM	98.97
[178], 2018	- Curvature - Moment d'inertie zonal - Variance - Entropie	KPPV (88.98%) KPPV-SVM	99.26
[118], 2018	Image	STDP (spike-timing dependent plasticity) Reward-Modulated STDP DCSNN (Deep Convolutional Spiking NN)	97.2

TR : *taux de reconnaissance de combinaison en %.*

Selon ces travaux de recherches la combinaison des classifieurs améliore le taux de reconnaissance des classifieurs faibles d'une manière remarquable. Pour les classifieurs plus performants, des combinaisons hybrides sont utilisées. Les meilleurs résultats sont obtenus avec les classifieurs utilisant l'image ou des caractéristiques qui ne sont pas invariantes aux transformations géométriques. Pour concrétiser l'effet de la combinaison nous proposons d'utiliser différents types de classifieurs (faible et performant).

1.8. Conclusion

Dans ce chapitre, nous avons présenté un aperçu de la problématique de reconnaissance de chiffres arabe manuscrits et imprimés, application ciblée par ce travail.

En premier lieu, nous avons tenu à présenter un historique sur les étapes du développement de la reconnaissance de l'écriture. Une revue de la littérature sur l'ensembles de classifieurs et les paramètres caractéristiques utilisé pour la reconnaissance des chiffres imprimés et manuscrits a été présenté. Dans la dernière partie du chapitre, nous avons aussi présenté une revue des travaux de recherche sur les systèmes multiple classifieurs utilisés pour augmenter le taux de reconnaissance des chiffres.

A travers les résultats des recherches présentés dans ce chapitre, la question posée est : Quel est le facteur le plus important dans un système de RdF :

- L'extraction des paramètres caractéristiques, les classifieurs ou l'utilisation de multiple classifieurs.
- Dans le cas d'un MCS quelle est la meilleure méthode de combinaison ?

Aussi, notre objectif est d'utiliser différents classifieurs, différents paramètres caractéristiques (considérant l'invariance aux transformations géométriques) et de faire des combinaisons à tous les niveaux à savoir au niveau des paramètres caractéristique (concaténation), au niveau des classifieurs avec les mêmes ou des différents paramètres caractéristiques, et en utilisant des méthodes de combinaisons de différents types.

L'ensemble des paramètres caractéristique et les classifieurs utilisable dans ce domaine ainsi que les différentes stratégies de combinaison de classifieurs, sont présentés en détail dans les chapitres 2 et 3.

Chapitre 2 . SYSTEME DE RECONNAISSANCE DE FORMES

2.1. Introduction

2.2. Concepts de base en RdF

2.3. Prétraitements

2.4. Extraction des paramètres caractéristiques

2.5. Classification

2.6. Conclusion

Les mots reconnaissance et formes sont couramment utilisés. Donc, la reconnaissance des formes (RdF) dans le cas général n'a pas besoin de définition formelle. Elle constitue, dans l'étude des systèmes automatiques ou semi-automatiques, la capacité de reconnaître les formes.

La RdF est caractéristique de tous les organismes vivants. Cependant, différentes créatures reconnaissent différemment [182].

La tâche principale d'un système de RdF est de fournir des décisions basées sur des échantillons donnés, par lequel les nouveaux objets entrants peuvent être classés.

Dans ce chapitre, nous présentons les trois tâches principales d'un système de RdF à savoir : les prétraitements, l'extraction des paramètres caractéristiques et la classification.

Développer un système de RdF non pas pour remplacer l'homme, mais pour l'aider dans les tâches difficiles.

2.1. Introduction

La RdF « pattern recognition » a pour objectif de concevoir et d'analyser des algorithmes de classifications applicables aux images, sons et signaux de toute nature. La RdF ne peut commencer qu'après la transformation de forme en un ensemble de nombre par des capteurs spécifiques (caméra, microphone ...). Selon Milgram [183], les disciplines les plus proches de la RdF sont :

- Les Mathématiques : le calcul des probabilités, la logique, la combinatoire, la géométrie intégrale, le traitement du signal, l'algèbre linéaire ... etc.
- L'informatique : les grammaires formelles, la théorie de la démonstration, les systèmes à base de connaissance, l'intelligence artificielle qui est considérée comme incluant la RdF.

Mais, à partir des années quatre-vingt, la RdF est considérée comme une branche de l'informatique à raison du développement des ordinateurs. La discipline de la RdF selon [184] se trouve au croisement de beaucoup d'autres dont les principales sont : Les statistiques, la linguistique, la recherche opérationnelle la théorie des communications, l'informatique, la biologie, l'optique, l'électronique.

La RdF a été utilisée pour de nombreuses applications. Le domaine le plus souvent utilisé par les chercheurs est la reconnaissance de l'écrit et en particulier de l'écriture manuscrite (cursive). D'autres domaines demandent l'application de la RdF comme l'authentification des visages, la reconnaissance de la parole, le traitement d'images et des signaux biomédicaux, la localisation des éléments artificiels (routes, bâtiments, aérodromes) sur des images satellites, détection et classement des anomalies dans les matériaux et les produits, l'analyse des objets biologiques (chromosomes, fibres...). Nous pouvons citer d'autres applications comme [185] :

- Applications scientifiques : Astronomie, exploration géologie-planétaire, géologie-cartographie et géodésie, analyse de données satellite, détection de la vie et analyse de la date sur des planètes éloignées.
- Applications dans la vie et des sciences du comportement : Anthropologie, archéologie, entomologie., biologie et botanique (microbiologie, l'écologie, et de la zoologie.), psychologie (aspects sociologiques et les aspects criminologiques), cybernétique, systèmes de gestion de l'information, éducation, communication.
- Applications industrielles : La reconnaissance de caractères, contrôle de processus, analyse de signature, analyse de discours, reconnaissance photographique, exploration minérale (analyse du sous-sol), détection de flux interne (rayons X et ultrasons),

amélioration de la photographie commerciale, multimédia et animation. Conceptions des jouets électroniques, Cytologie automatisée.

- Applications médicales : L'examen microscopique et données biomédicale, l'examen de radio-isotopes, examen aux rayons X et tomographie, analyse de l'électrocardiogramme et vectocardiogramme, traçage électroencéphalogramme et traitement du signal neurobiologique, propriétés chromosomiques pour les études génétiques.
- Applications agricoles : Analyse des cultures, évaluation des sols, contrôle des opérations, terre-ressources photographie.
- Applications gouvernementales : Prévisions météorologiques (suivi des nuages, mesures de la température de l'eau), systèmes publics (analyse et contrôle de la circulation), détermination de la croissance urbaine, détection et mesure du smog et de la circulation de l'air, réduction des données radar, des données Terre-ressources et de la télédétection.
- Certaines applications militaires spécifiques : Photographie aérienne et télédétection, détection et classification Sonore, ATR : reconnaissance automatique de cible.

Le schéma général d'un système de RdF est présenté dans la figure 2.1 [186]. Ce schéma est basé sur le point de vue général qui regroupe principalement les méthodes mathématiques proposées pour résoudre les problèmes de RdF en deux approches principales à savoir l'approche de la théorie de décision (ou la discrimination) et l'approche syntaxique ou structurelle [186, 187]

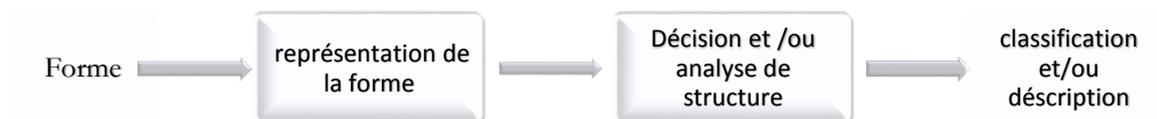


Figure 2.1 : Système de RdF général [186]

Dans ce chapitre nous présentons en détail les trois principales étapes pour construire un système de RdF et ce quel que soit l'approche figure 2.2.

2.2. Concepts de base en RdF

L'objet inspecté pour le processus de reconnaissance est appelé *modèle*. Afin de fournir une description efficace des modèles (formes), un *prétraitement* est souvent nécessaire pour éliminer le bruit et la redondance dans les mesures. Un ensemble de *mesures caractéristiques*, qui peuvent être numériques et/ou non numériques, et les relations entre ces mesures, sont extraites pour la représentation des formes. Une *analyse* (*classification* et/ou *description*) des formes est effectuée sur la base de la représentation.

Habituellement, le nombre de mesures devient élevé. Cette haute dimensionnalité (La malédiction des grandes dimensions -Curse of dimensionality [188]) crée de nombreux problèmes de reconnaissance de formes difficiles. Quand le nombre d'entrées à un classifieur devient plus petit, la conception du classifieur devient plus simple. Afin de profiter de cet avantage, nous devons trouver un moyen pour sélectionner ou extraire des caractéristiques importantes des échantillons observés. Ce problème est appelé *sélection de caractéristiques* qui est un autre thème important de la RdF [185].

Dans la conception d'un système de RdF figure 2.2, i.e. un système qui est en mesure d'obtenir un modèle entrant inconnu et le *classifier* dans un ou plusieurs classes données, en utilisant toutes les informations relatives disponibles. Nous supposons que la classe de certains échantillons disponibles est connue. Ces modèles (échantillons) avec leurs attributs typiques forment un ensemble *d'apprentissage* qui fournit des informations pertinentes de lien entre les données d'entrée et la prise de décision. En utilisant l'ensemble d'apprentissage le système de RdF peut apprendre différents types d'informations comme les paramètres statistiques, les caractéristiques pertinentes, ... etc. [185]

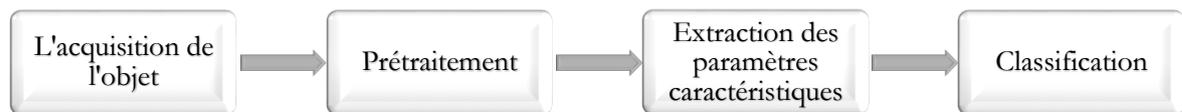


Figure 2.2 : Système de RdF [189]

La construction d'un système de RdF comprend plusieurs étapes distinctes à savoir [189, 190] : Prétraitement qui consiste à isoler et préparer la forme à reconnaître. Le choix des paramètres caractéristiques (primitives) appelé extraction ou sélections des paramètres caractéristiques. La classification par des *classifieurs* qui peut être conçue plus facilement avec une meilleure performance. En effet, si elle est mal conçue, il serait difficile, voire impossible, d'effectuer une reconnaissance efficace.

Ainsi, la RdF, ou dans un sens plus large, *la prise de décision*, peut être considérée comme un problème *de l'estimation des fonctions de densité* dans un espace de grande dimension et divisant l'espace dans des régions de catégories ou classes. Beaucoup d'applications importantes de RdF peuvent être caractérisées comme étant la *classification* de signal ou des figures géométriques [185].

2.3. Prétraitements

Les prétraitements facilitent la caractérisation de la forme à reconnaître soit en nettoyant l'image représentant la forme ou en réduisant la quantité d'information à traiter pour ne garder

que les informations les plus significatives. Donc les prétraitements incluent toutes les opérations et les fonctions effectuées avant l'extraction des caractéristiques pour produire une version « nettoyée » de l'image d'origine afin qu'elle puisse être utilisée directement et efficacement par le système de RdF. Ci-après quelques prétraitements sont présentés :

- **Binarisation** : Il est souvent souhaitable de représenter les images en niveaux de gris ou en couleurs ou sous forme d'images binaires pour réduire le stockage en utilisant les techniques de seuillage global.
- **Détection de contour** : Les contours constituent une information essentielle pour certaines applications de traitement d'image, le contour est les variations significatives en pixels (changement de la valeur de pixel), un contour doit être une chaîne de pixels d'épaisseur 1. La détection des contours peut être réalisée grâce à des filtres dont les coefficients ont été soigneusement choisis [191].
- **Amincissement (Schématisation)** : C'est la suppression de tous les pixels redondants jusqu'à ce que l'épaisseur ne soit plus que d'un point. Sa grande utilisation sera la squelettisation. Différentes méthodes d'amincissements sont présentées dans [192].
- **Segmentation** : le but de la segmentation c'est la détermination le début et la fin d'une sous forme (consiste à isoler la forme à reconnaître) comme les caractères dans les mots. Lorsque les caractères sont déjà séparés les uns des autres (le texte est imprimé), le système de RdF doit écrire les caractères dans des encadrés ou sans les réunir (les dates, les matricules), cela signifie la pré-segmentation.

Pour plus de détails sur les traitements et l'analyse d'image, vous pouvez consulter les ouvrages [190, 191, 193, 194, 195, 196]. Les résultats obtenus lors de cette phase de prétraitement permettront le passage vers la phase d'extraction de caractéristiques et le choix des paramètres caractéristiques limite la nature et la sortie des prétraitements, ce qui explique la dépendance entre les phases du système (figure 2.2).

2.4. Extraction des paramètres caractéristiques

L'objectif de l'extraction des caractéristiques consiste à exprimer les primitives (paramètres) sous une forme numérique (réelles, entières ou binaires). Cette étape permet de décrire les formes appartenant à une même classe tout en les différenciant des autres classes, elle est basée sur une description d'une forme, cette description peut être vue comme une modélisation de l'objet qui permettant un calcul de distance entre deux ou plusieurs modèles. Lorsque nous avons deux ou plusieurs classes, l'extraction de caractéristiques consiste à choisir les caractéristiques qui

sont les plus efficaces pour préserver la séparabilité des classes [185]. Selon la littérature cette phase est la partie la plus importante dans le système de RdF. Elle permet de représenter une image (matrice de pixels) par un ensemble de caractéristiques permettant d'identifier plus facilement ce dernier.

Le choix des attributs dépend énormément du problème posé. Malheureusement il n'y a pas de théorie guidant ce choix. Ils doivent vérifier les propriétés suivantes [184] :

- Discriminabilité (grande variance inter-classe) : un bon attribut doit avoir des valeurs significativement différentes pour des formes appartenant à des classes différentes.
- Fiabilité (faible variance intra-classe) : un bon attribut doit avoir des valeurs très similaires (à faible variabilité) pour les formes d'une même classe.
- Indépendance : dans un ensemble d'attributs choisis pour un problème donné, un attribut quelconque ne doit pas dépendre d'un autre attribut.
- Faible nombre d'attributs : le coût, la complexité et les exigences en temps de calcul d'un système de RdF croissent avec le nombre d'attributs utilisés. Il faut donc choisir les meilleurs attributs pour diminuer ce nombre tout en satisfaisant la probabilité d'erreur acceptable, fixée à l'avance.
- Indépendance au transformation géométrique : les attributs doivent être indépendants de la translation, de la rotation et du facteur d'échelle.

Selon [184] il existe deux méthodes qui permettent de décrire une forme. La première est l'approche *externe (contour)* qui se base sur la description du contour de la forme par deux méthodes *locale* et *globale*. Le but de la méthode locale est de décrire le contour point par point alors que la méthode globale vise à décrire le contour au moyen d'attributs calculés globalement sur tout le contour. La deuxième est l'approche *interne (région)* qui utilise la description de l'intérieur de la forme.

Les primitives sont classées en deux catégories [160, 197]. Les primitives *structurelles* (ou primitives locales) basées sur une représentation linéaire du caractère (intersections avec des droites, décomposition du caractère en segments de droites et courbes, contours du caractère, squelette, cavités, les points d'extrémités, les boucles, les croisements des traits, les extrema). Les primitives *statistiques* (ou primitives globales) basées sur la distribution statistique des points – pixels- (surface, moments, zonage, n-tuples [198, 199], projections, profile, loci [198, 200, 201], ... etc. Dans [8] l'auteur a considéré toute l'image i.e. le niveau gris ou les valeurs binaire de pixels comme une catégorie pour les paramètres caractéristiques

2.4.1. Primitives structurelles

Les primitives structurelles (local) appelées aussi syntaxiques ou linguistiques consistent à donner une description basée sur le contour de la forme. Les valeurs des primitives représentent donc des segments du contour à décrire. La notion de structure recouvre des concepts variés. Une structure doit décrire la décomposition d'un objet complexe en termes d'objets plus simples ainsi que les liens entre eux. Cette description est faite selon un modèle particulier, adapté aux objets considérés [184].

La détermination d'un ensemble de primitives qui permettront de décrire les formes complexes à reconnaître ne suit aucun modèle universel pour la sélection de primitives. Les primitives sont généralement choisies en fonction de deux critères. En effet, elles doivent :

- Favoriser la représentation des formes complexes en terme de relations structurelles ;
- Etre faciles à déterminer ou à classer. Leur extraction n'utilise en règle générale aucune information relative à la description structurelle.

Cette approche est basée sur l'extraction des primitives structurelles. Elle décrit l'organisation topologique (la structure) de la forme à partir de ses composantes élémentaires et nécessite une mesure de la similarité entre deux représentations structurelles. Contrairement à l'approche globale (statistique), l'approche structurelle tente de trouver dans une forme les éléments signifiants (en fonction d'une interprétation précise) et leurs relations [183].

L'extraction des primitives structurelles commence soit par une squelettisation de la forme, soit par une détection des contours intérieurs et extérieurs. Nous présentons dans (l'annexe 1) quelques définitions de primitives structurelles comme l'intersection avec des droites, les contours squelette, les cavités, les points d'extrémité et de jonction, les extrema, code de chaîne, ... etc.

2.4.2. Primitives statistiques (global)

Les primitives statistiques (global) consistent à extraire de la forme à connaître un ensemble de "m" mesures qui constituent les composantes d'un vecteur d'un espace de représentation de dimension "m" (qui peuvent être assez élevé). Elles comprennent principalement des informations topologiques et métriques. Les primitives recherchées doivent présenter une certaine invariance géométrique d'une part, et statistique d'autre part. La première signifie une tolérance vis-à-vis des opérateurs de translation, de rotation, et de changement d'échelle, tandis que la seconde signifie une tolérance vis-à-vis du bruit inévitablement présent sur chaque caractère. Parmi les caractéristiques statistiques nous pouvons citer : le codage rétinien (naturel ou la rétine), la surface, le zonage et la densité des pixels [202], l'histogramme de projection, le profil, la transformée de Hough, l'histogramme de gradient orienté (HOG), les descripteurs de Fourier,

les moments complexes, les moments de Hu, de Zernike, de Tchebichef ... etc. Nous allons présenter les moments géométriques, Hu et Zernike en détail dans le chapitre 4. Quelques définitions sont présentées dans (l'annexe 1).

Nous avons cité les paramètres caractéristiques structurelles et statistiques les plus utilisés dans la littérature. Il existe plusieurs autres comme les moments complexes, les moments de Legendre, les moments invariants étendus [84], la transformée de Karhunen Loève, la transformée de Randon (R-signature) [203,152], les moments invariants uni (united moment invariant) [134], la transformée de Gabor, la transformée de Mojette [74], la transformé de Wavelet, le template matching (correspondance de modèle), le template déformable [204], le codage clairsemé (sparse coding) [205], ...etc. Pour plus de détails le lecteur peut consulter les références [6, 96, 160, 206, 207]. L'ouvrage [208] traite en détail cette phase.

Le problème du choix de méthode d'extraction des paramètres caractéristiques n'est pas évident. En effet la méthode impose la nature et le résultat de l'étape de prétraitement. Il existe différentes méthodes d'extraction de primitives citées dans la littérature. Selon Trier [96], certaines fonctionnent sur des images de niveau gris, d'autres sur des images binaires avec des symboles (forme) solides (non amincis), des squelettes (des symboles amincis) ou sur des contours de la forme. Les différentes méthodes d'extraction de caractéristiques sont conçues selon la représentation des formes (formes binaires pleins, des squelettes de la forme, des contours de caractère ou des sous-images de niveaux de gris -tableau 2.1-).

Tableau 2.1 : Taxonomie des méthodes d'extraction de caractéristiques selon la représentation de l'image [6, 7, 96]

Caractéristiques	Image gris	Image binaire	Contour	Squelette
- Appariement	*	*		*
- Motifs déformables	*			*
- Transformation unitaire	*	*		
- Transformation log-polaire	*	*		
- Moments géométriques	*	*		
- Moments de Zernike	*	*		
- Ondelettes	*	*		
- Algébriques	*	*		
- Histogrammes de projection		*		
- Masques	*	*		
- Profil du contour			*	
- Code de Freeman			*	
- Spline			*	
- Descripteurs de Fourier			*	*
- Description graphique				*
- Discrets				*
- Zonage	*	*	*	*

L'identification de la méthode d'extraction des paramètres caractéristiques appropriée est basée sur le besoin d'une application donnée en termes d'invariance et de capacité de reconstruction. De plus, les primitives extraites doivent correspondre aux exigences de classifieur choisi. Les descriptions de graphes ou les descriptions de caractères basées sur la grammaire conviennent bien aux classifieurs structurels ou syntaxiques. Les vecteurs de valeur réelle sont parfaits pour les classifieurs statistiques [96,160].

Souvent, l'utilisation d'une seule méthode d'extraction de caractéristiques n'est pas suffisante pour obtenir de bonne performance du système de reconnaissance. Cependant pour chaque application donnée de RdF il est nécessaire d'effectuer une évaluation expérimentale de quelques types de paramètres caractéristiques les plus prometteuses. En effet, Ces expériences permettront de faire un choix judicieux des primitives à extraire. La solution évidente est de combiner plusieurs méthodes d'extraction afin de donner une meilleure description de la forme à classifier. Une approche qui intègre des informations (primitives) statistiques et structurelles est proposé dans [95] pour la reconnaissance des chiffres manuscrits.

Dans notre travail nous avons choisi après une évaluation expérimentale (test), des paramètres caractéristiques de chaque catégorie structurelles et statistiques en utilisant les quatre représentations des formes avec et sans intégration (les détails dans le chapitre 4 et 5).

2.4.3. Sélection des paramètres caractéristiques

En pratique, un nombre plus important que nécessaire de paramètres est généré. Le « meilleur » d'entre eux est adopté. L'auteur dans [208] considère « la sélection des caractéristiques » comme une étape de base de la classification (figure 2.3). Il s'agit d'une tâche très importante, qui consiste à choisir le meilleur nombre de paramètres à utiliser.

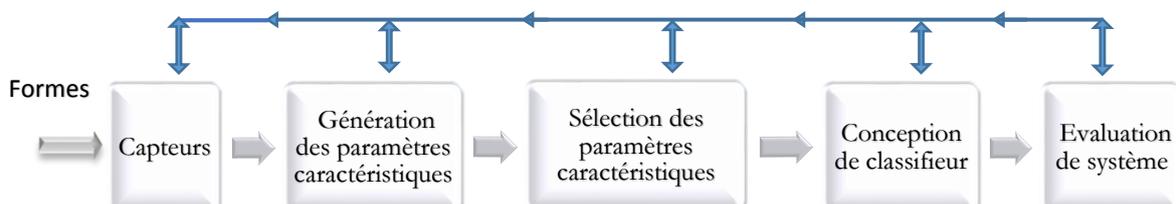


Figure 2.3 : Etapes de base de la conception d'un système de classification [208]

Le processus connu sous le nom de « Sélection de caractéristiques » a pour but de filtrer le vecteur des primitives de base, de manière à en extraire l'information discriminante et à présenter celle-ci au classifieur de manière pertinente. Il est fréquent qu'une partie de celles-ci ne contienne

que des informations redondantes ou inutiles à la classification, rendant ainsi l'apprentissage du système de reconnaissance plus complexe [209]

Aussi, l'objectif principal de la sélection des primitives est la réduction du nombre de caractéristiques utilisées tout en essayant de maintenir ou d'améliorer la performance de la classification du système de RdF. Pour atteindre cet objectif, il suffit de choisir une méthode d'optimisation qui peut être adaptée au problème à résoudre. Parmi ces méthodes nous pouvons citer : l'analyse en composantes principales ACP [191], l'algorithme génétique ...etc. Plusieurs travaux de recherches traitent ce sujet [20, 152, 210]. Les ouvrages [208, 211] présentent plusieurs méthodes de sélection des paramètres caractéristique. Dans [212] vous trouverez une taxonomie des algorithmes de sélection des paramètres caractéristiques.

La représentation des formes à reconnaître par l'extraction de paramètres caractéristiques est une étape importante dans la réalisation d'un système de RdF. L'obtention de bonnes performances, dépend fortement de la qualité de cette représentation. Les autres étapes du système doivent également être optimisées pour obtenir les meilleures performances possibles et ces étapes ne sont pas indépendantes.

2.5. Classification

Après l'établissement de l'espace des attributs, la classification (reconnaissance) consiste à partitionner cet espace en attribuant une région à chaque classe. Cette segmentation consiste à évaluer la valeur des attributs et à prendre une décision.

2.5.1. Concepts de base

L'objectif final de la RdF est la classification d'un objet. De l'information originale que nous recueillons sur l'objet que nous identifions les caractéristiques pertinentes et ensuite nous utilisons un extracteur de caractéristiques pour les mesurer. Ces mesures sont ensuite transmises à un classifieur qui effectue la classification [182]. Le rôle d'un classifieur est de décider à quelle classe appartient un objet donné parmi un ensemble fini de classes.

Un classifieur doit être capable de modéliser au mieux les frontières qui séparent les classes les unes des autres. Cette modélisation fait appel à la notion de fonction discriminante qui permet d'exprimer le critère de classification [209].

En général, il existe deux approches probabilistes pour développer des classifieurs qui reposent sur l'estimation des distributions de chacune des classes de la base d'apprentissage : l'approche *paramétrique* et l'approche *non paramétriques*. La première, dans laquelle on suppose une

connaissance à priori de la distribution des données, présuppose la forme des lois. Elle fait l'hypothèse que les classes étudiées suivent une distribution de probabilité d'une certaine forme connue *à priori*. On applique alors la théorie statistique de décision (Bayes, vraisemblance maximum, ... etc.). Elle exige des bases d'apprentissage assez importantes pour une estimation correcte des paramètres de la distribution supposée. Cette méthode englobe la règle de Bayes, la distance de Mahalanobis, les méthodes neuronales, les chaînes de Markov, le modèle mélange gaussiens multivariés. Dans le cas de l'approche *non paramétrique*, il faut établir des frontières de décision entre les classes parce que les lois de probabilité sont inconnues pour une des classes (aucune connaissance a priori n'est supposée). Les techniques les plus utilisées sont la méthode du k plus proche voisin (KPPV), la méthode de Parzen, Analyse discriminante du noyau (Kernel Discriminant Analysis –KDA-) et la méthode d'appariement de graphes [213, 214].

Les estimations paramétriques ou non paramétriques sont souvent effectuées lors d'une phase dite *d'apprentissage* qui s'effectue de trois manières différentes : *supervisée*, *semi supervisée* ou *non supervisée*. Dans le premier cas les classes sont connues et identifiées à priori i.e. les observations de la base d'apprentissage sont étiquetées. Dans le deuxième cas, une partie des observations est étiquetée alors que dans dernier cas elles sont inconnues (aucune des observations n'est étiquetée, l'étape de la classification identifie automatiquement les formes appartenant à une même classe).

2.5.2. Les approches de classification

Pour construire un classifieur, il existe plusieurs approches. Deux approches ont été présentées dans [96, 160, 215] : l'approche structurelle et l'approche statistique. Ce sont généralement les deux principales approches de la classification. Trois approches présentées par A. Pal and S.K. Pal dans [216] : structurelle, statistique et connexionniste. M. Friedman dans [182] en a mentionné quatre : l'approche statistique, l'approche syntaxique, les réseaux de neurones et l'approche floue. Quatre approches suggérées aussi dans [9, 206], l'approche de correspondance de modèle (Template Matching), l'approche structurelle, l'approche statistique et les réseaux de neurones. K. S. Fu [186, 187] a regroupé les méthodes de reconnaissance en deux principales approches à savoir statistique fondée sur la théorie de décision (ou la discrimination) et syntaxique ou structurelle. Une approche intuitive utilisée dans les premiers développements de la RdF est l'approche par correspondance de modèles (template matching). Nous abordons brièvement dans la suite les quatre approches mentionnées.

a) *L'approche de correspondance de modèles*

Dans l'approche de correspondance (template matching), les formes sont généralement représentées en termes de données brutes. Le processus de prise de décision n'est rien d'autre que

l'adaptation de l'entrée inconnue à chaque modèle, et le critère d'appariement –correspondance- (corrélation, par exemple) utilisé reflète directement la similitude entre les deux [186]. i.e. le degré de similarité entre deux vecteurs (groupes de pixels, formes, courbures, ... etc.) est déterminé dans l'espace des fonctions. Les techniques de correspondance peuvent être regroupées en trois classes : correspondance directe, gabarit déformable et correspondance élastique et correspondance de relaxation [17].

Il faut noter que l'approche par correspondance de modèles peut être considérée comme un cas particulier de l'approche décisionnelle lorsque les formes sont représentées par leurs données brutes ou leurs vecteurs caractéristiques correspondants. D'autre part, l'approche de correspondance de modèle peut également être considérée comme un cas particulier de l'approche syntaxique lorsque les modèles sont également représentés par des chaînes, des arbres ou des graphiques. Le tableau 2.2 résume les principales approches de la reconnaissance des formes du point de vue général [186].

b) L'approche statistique

Cette approche a besoin d'un nombre élevé d'exemples afin de réaliser un apprentissage automatique qui s'appuie sur des bases théoriques des lois de probabilité connues telles que la théorie de la décision bayésienne. Le processus de prise de décision (et/ou d'analyse de structure) est basé sur une mesure de similarité qui, à son tour, est exprimée en termes d'une mesure de distance, d'une fonction de vraisemblance ou d'une fonction discriminante. La figure 2.4 illustre cette approche.

Une revue de cette approche statistique est présentée en détail dans la référence [9]. Une revue des algorithmes de classification d'apprentissage supervisé (également connus comme algorithmes de classification d'induction) les plus utilisés, est présenté dans [217]. Elle contient également une partie référence très riche et couvre les principaux problèmes de ce domaine. Nous décrivons brièvement un ensemble de méthodes statistiques en reconnaissance de formes les plus utilisés dans l'annexe 1.

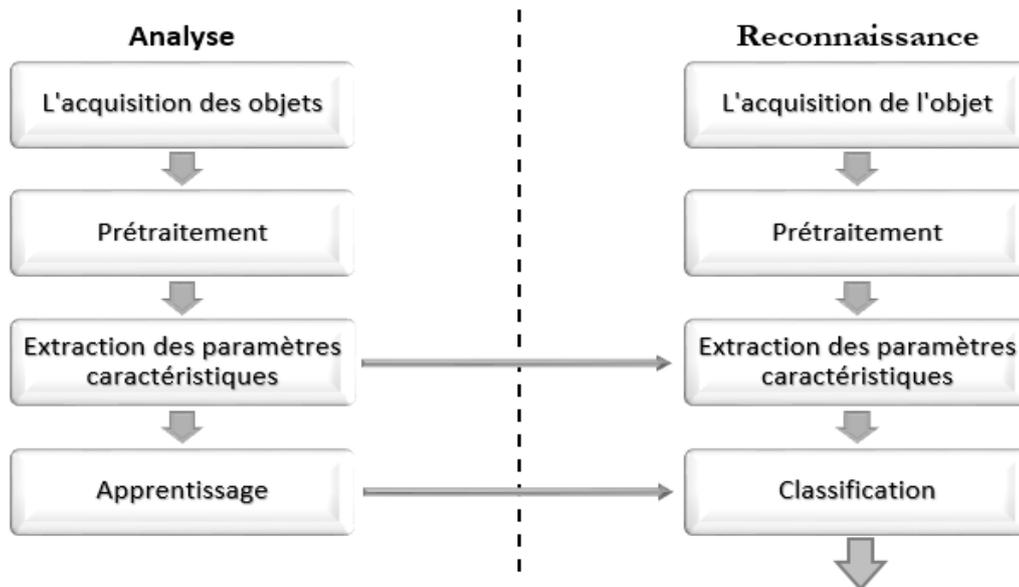


Figure 2.4 : Diagramme de la RdF statistique

c) *L'approche structurale*

Dans l'approche structurale, une classe est décrite par une grammaire est composée d'un ensemble de règles syntaxiques qui déterminent l'ensemble des formes admissibles dans cette classe et présentant en principe des caractéristiques structurales communes. Cet ensemble de formes est appelé *langage engendré par la grammaire*. Les règles syntaxiques d'une grammaire sont déterminées par apprentissage d'un certain nombre d'échantillons jugés représentatif de la classe. Ce problème est connu sous le nom d'inférence grammaticale [184]. Cette approche est illustrée sur la figure 2.5

M. Milgram [183] a distingué deux grandes techniques pour le traitement structuré de la forme. Les structures de graphes et les structures syntaxiques. Ces deux grandes familles du traitement structurale de la forme sont détaillés dans les références [183, 184, 215]. Les techniques syntaxiques ont été largement utilisées dans un certain nombre d'applications [97, 187, 218, 219]. Le traitement général de l'approche syntaxique et une revue de la littérature et des applications peuvent être trouvés dans les livres de Fu [187, 220]. Ce sont de très bonnes références pour comprendre la RdF syntaxique. Un ensemble de méthodes structurales sont présentés dans l'annexe 1.

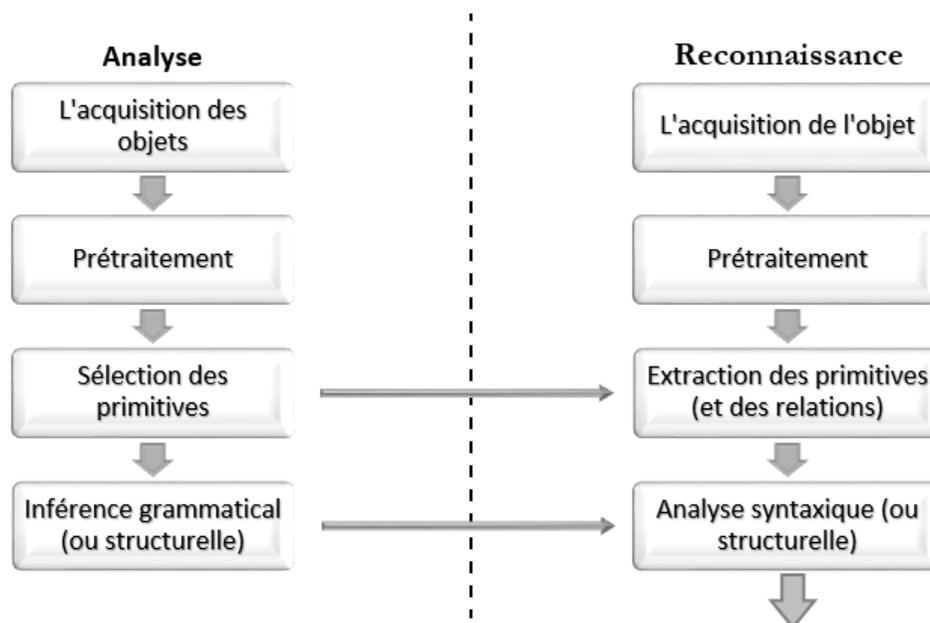


Figure 2.5 : Diagramme de la RdF syntaxique

d) L'approche neuronale

Dans l'approche neuronale (connexionniste), la connaissance réside précisément dans les connexions i.e. des valeurs numériques appelées des *poids* associées aux arcs reliant les unités appelées *neurones* (cellules, automates). Chaque unité possédant une *activité* caractérisée un nombre. Pour définir un réseau connexionniste spécifique, il y a trois éléments à fournir [183] : *l'architecture*, *la dynamique* et *l'apprentissage*.

Les principaux avantages des réseaux de neurones sont : la possibilité de s'entraîner automatiquement à partir d'exemples, les bonnes performances avec des données bruitées, l'implémentation parallèle possible et les outils efficaces pour l'apprentissage de bases de données volumineuses.

Le réseau de neurones a plusieurs architectures. La plus largement étudiée et utilisée est le perceptron multicouche (Multi-Layer Perceptron –MLP-). Parmi les autres architectures nous pouvons citer, le perceptron, l'Adaline, l'Adaline multicouche, le réseau de convolution (CN), les cartes auto-organisées (SOM -Self Organised Map –), la fonction de base radiale (RBF), le réseau de neurones à déplacement spatial (SDNN), le réseau neuronal à délai différé (TDNN), Réseau neuronal quantique (QNN), et le réseau de neurones de Hopfield (HNN), Boltzmann, LVQ (Learning Vector Quantization).

Le tableau 2.2 résume les paramètres caractéristiques et les classifieurs les plus utilisés pour les quatre approches à savoir statistique, structurelle, correspondance de modèle et connexionniste.

Tableau 2.2 : Approches de la RdF

Approche	Paramètres caractéristiques	Classifieur (processus de prise de décision et / ou d'analyse de structure)
Correspondance de modèle	- Donnée brute (Image)	<ul style="list-style-type: none"> - Modèle Flexible - Prototype déformable - Masque ruber - Correspondance directe - ... Etc.
Approche statistique	- Vecteur de caractéristiques statistique N-dimensionnelle,	<ul style="list-style-type: none"> - KPPV, MMC, SVM - Classifieur Bayésien - Fenêtres de Parzen - Classifieur discriminant polynomial - Raisonnement flou - Arbre de décision - ... Etc.
Approche structurelle	<ul style="list-style-type: none"> - Chaîne - Arbre - Graphe 	<ul style="list-style-type: none"> - Distances entre chaîne - Mise en correspondance combinatoire - Relaxation - Mise en coïncidence - Structure syntaxique régulière - Structure syntaxique algébrique - Structure d'arbre - ... Etc.
Approche connexioniste	- Vecteur de caractéristiques statistique N-dimensionnelle	<ul style="list-style-type: none"> - MLP, - RBF, - SDNN, - TDNN - QNN - HNN - LVQ - ... Etc.

K. S. Fu dans [186] a cité d'autres nouvelles approches proposées pour la RdF comme la logique à valeur variable (variable-value logic), la théorie des catégories (category theory), la

théorie des relations (relation theory), la théorie des ensembles flous (fuzzy set theory) et les modèles de résolution de problèmes (problem-solving models).

Toutes les approches statistique, structurelle et connexionniste ont leurs propres points forts et leurs points faibles. Le choix d'une approche peut être lié à des contraintes telles que, le temps de calcul, la taille mémoire et la taille de la base d'apprentissage. Les approches statistiques et structurelles peuvent être combinées aux mêmes domaines d'application comme une solution optimale pour un système de RdF [161, 221]. S. K. Pal et A. Pal dans [216] ont considéré l'intégration de ces approches avec la logique floue et les algorithmes génétiques comme une autre approche baptisée approche hybride, en vue de développer un nouveau paradigme efficace appelé *soft computing*. L'objectif est de développer des paradigmes informatiques offrant des solutions relativement bonnes à faible coût dans des conditions non idéales et ressemblent beaucoup à une prise de décision semblable à celle de l'homme.

2.6. Conclusion

Nous avons présenté dans ce chapitre un aperçu détaillé d'un système de RdF à savoir les prétraitements, l'extraction des paramètres caractéristique et la classification. Deux approches d'extraction des paramètres caractéristique ont été présentés (statistique et structurelle).

Les deux catégories des paramètres caractéristiques structurelles et statistiques semblent être complémentaires en ce sens qu'elles mettent en évidence différentes propriétés. La combinaison des deux approches peut être utilisée pour fournir une description efficace et complète des formes, Afin de couvrir les variabilités intra-classe et inter-classe des classes. Toutefois, plusieurs classifieurs peuvent être utilisés [96], soit en tant que schéma de classification à plusieurs étapes, ou en tant que classifieurs parallèles, lorsqu'une combinaison des résultats de la classification individuelle est utilisée pour décider de la classification finale.

Malheureusement il n'y a pas une théorie unifiée de la RdF. Malgré cette absence, les approches du système de RdF peuvent être répartis en deux catégories : l'approche *statistique* basé sur la théorie statistique de la décision et *structurelle* basée sur des règles syntaxiques (figure 2.6).

L'approche syntaxique qui utilise des caractéristiques structurelles a été omise de cette étude. Elle est intuitivement attrayante car elle semble plus proche de la stratégie de reconnaissance humaine. Malheureusement, elle est généralement difficile à mettre en œuvre de manière rapide, adaptable et robuste sur une grande variété de formes. Dans le cadre de ce travail de recherche, nous nous sommes intéressés, à l'ensemble des étapes de RdF en utilisant l'approche statistique.

Elle a été étudiée et utilisée de manière plus intensive dans la pratique et consiste à caractériser une forme d'une manière statistique.

Les méthodes de classification, n'ont pas permis d'identifier laquelle est la meilleure méthode par rapport aux autres. Ceci montre qu'il y a une complémentarité qui semble très importante de l'exploiter pour augmenter les performances par rapport à un seul classifieur. La même chose pour les paramètres caractéristiques. En effet, en ajoutant une caractéristique supplémentaire (complémentaire), permet à un classifieur de mieux décider. Aussi, une décision d'un autre classifieur permet d'améliorer la décision finale. Cela impose l'idée d'utiliser plusieurs classifieurs simultanément. Le chapitre 3 est dédié à la présentation en détails de la combinaison de classifieurs objet de notre problématique.

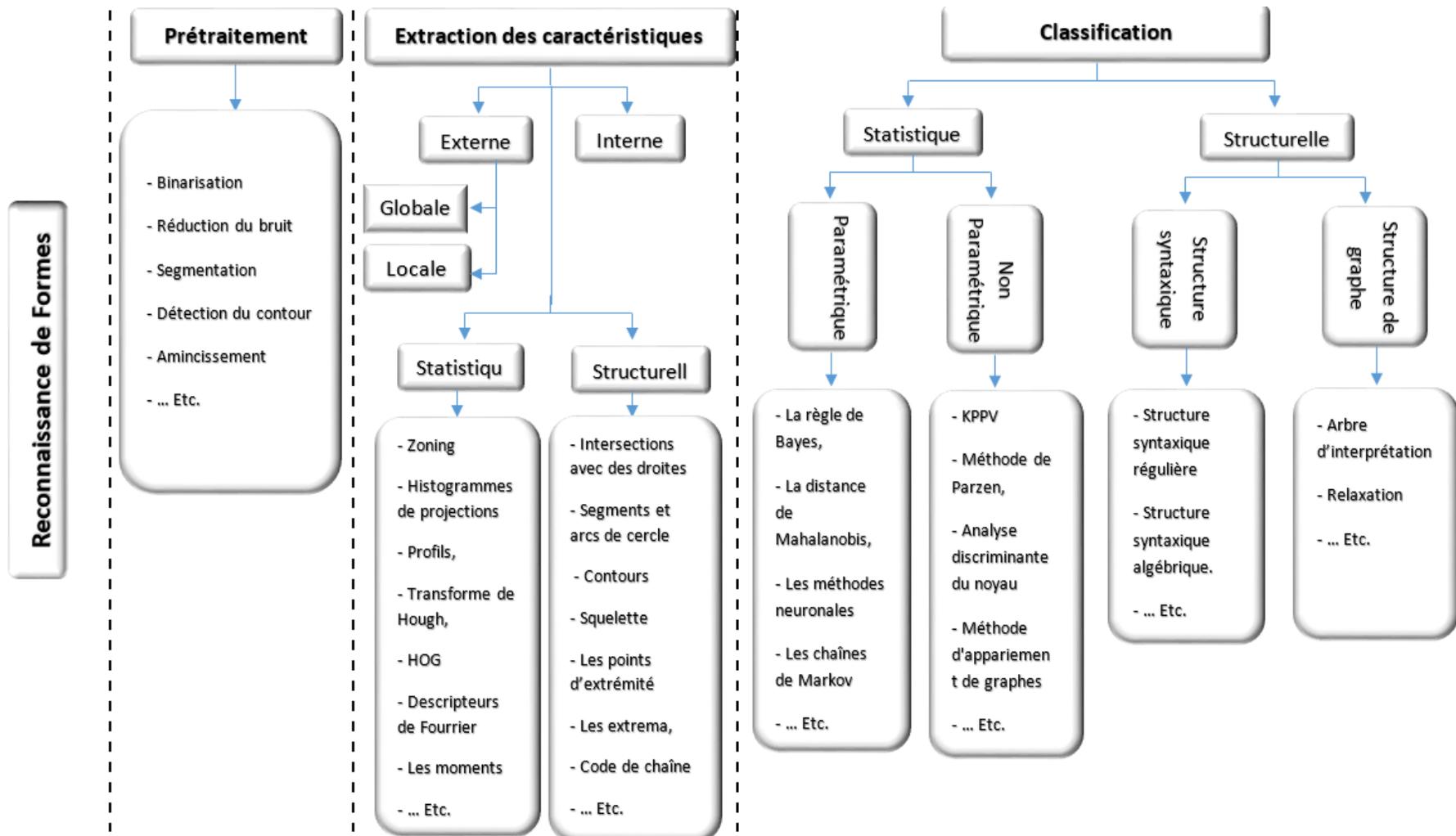


Figure 2.6 : Approches de la RdF

Chapitre 3 .

SYSTEME MULTICLASSIFIEURS

- 3.1. Introduction
- 3.2. Définition d'un classifieur
- 3.3. Système de classifieur multiple
- 3.4. Stratégies de combinaison
- 3.5. Les taxonomies de combinaison
- 3.6. Les méthodes de combinaison des classifieurs
- 3.7. Conclusion

La recherche actuelle en RdF porte sur la combinaison de classifieurs qui permettent d'améliorer la performance d'un système de RdF. En conservant les caractéristiques les mieux adaptés pour un classifieur, il est possible d'intégrer dans un même système l'avis de plusieurs classifieurs.

L'objet de la classification supervisée est la construction d'une règle d'affectation aux classes. On entend par classifieur, une fonction qui associe à une description individuelle, un vecteur de scores d'appartenance aux classes. L'expérience montre qu'il n'existe pas de méthode systématiquement supérieure aux autres. L'idée alternative est dans l'utilisation de plusieurs classifieurs i.e. on associe à chaque description individuelle, plusieurs vecteurs de scores d'appartenance aux classes.

Ce chapitre traite des stratégies, des taxonomies d'un système multi-classifieur et des différentes méthodes combinant ces derniers.

3.1. Introduction

La combinaison de deux ou plusieurs classifieurs confère les avantages de chacun d'eux pour créer un système avec une performance élevée, ce qui conduit à des solutions plus robustes. Il n'est donc pas étonnant que l'hybridation de méthodologies progresse rapidement dans le contexte de la RdF.

Ces dernières années, de nombreux chercheurs ont combiné plusieurs classifieurs afin d'améliorer les résultats de reconnaissance. L'idée n'est pas de compter sur un classifieur unique. Différents schémas de combinaison de classifieurs ont été mis au point et il a été démontré expérimentalement que certains d'entre eux surperforment le meilleur classifieur individuel [222], les méthodes de combinaison des classifieurs exploitent la complémentarité potentielle pour obtenir des performances supérieures à celles d'un seul classifieur.

Une caractéristique supplémentaire apportant une information complémentaire permet à un classifieur de mieux décider. Une décision provenant d'un autre classifieur permet à un système de classifieurs de mieux décider, d'où l'idée de combinaison de classifieurs s'impose progressivement.

Le véritable intérêt pour la combinaison de classifieurs date en fait des années 80 où les chercheurs ont montré l'importance d'avoir des solutions robustes pour les problèmes de reconnaissance de l'écriture manuscrite en particulier. Cependant, ce n'est qu'à partir des années 90 que les systèmes de combinaison de classifieurs ont été concrètement mis en œuvre [5].

La littérature est très riche et diverse concernant la combinaison de classifieurs. Une série d'ateliers internationaux annuels sur les systèmes de classifieur multiples (MCS –Multiple Classifier System), organisés depuis les années 2000, a joué un rôle central dans l'organisation, la systématisation et le développement des connaissances dans le domaine de la combinaison de classifieur [223]. MCS constitue le principal forum international de discussion sur les problèmes liés aux systèmes de classifieur multiples et aux méthodes d'ensemble. L'objectif de l'atelier est de rassembler des chercheurs de diverses communautés concernées par ce sujet, notamment la RdF, l'apprentissage automatique, les réseaux de neurones, l'exploration de données et les statistiques.

Dans ce qui suit, nous présentons l'objectif d'un classifieur et d'un système multi-classifieur, et décrivons les différentes stratégies de combinaison qui se différencient essentiellement par l'agencement des classifieurs. Nous abordons ensuite les taxonomies existantes et les méthodes de combinaison les plus utilisées et les plus reconnues dans la littérature. Enfin, nous concluons

sur la limite atteinte par un MCS et la difficulté de choisir une méthode de combinaison en dehors d'un contexte applicatif donné.

3.2. Définition d'un classifieur

Le classifieur est défini comme étant tout système de traitement de données qui reçoit une forme x et donne des informations à propos de la classe correspondant à cette forme. La mise en œuvre de tout classifieur nécessite le choix d'une représentation pour décrire les données (paramètres caractéristiques), d'un algorithme de décision et d'une base d'apprentissage permettant de fixer les paramètres du classifieur. Lorsqu'il est intégré dans un système de reconnaissance, le classifieur atteint certaines performances. Introduire des modifications dans le classifieur que ce soit au niveau des données qu'il traite ou au niveau de ses paramètres (type de sorties, règles de décision, ...) modifie ses performances [5].

Dans le cadre de la combinaison, Zouari [5] a défini le classifieur comme étant un système de reconnaissance qui travaille dans un certain espace de caractéristiques, utilisant une certaine base pour apprendre ses paramètres, et prend sa décision à partir d'une certaine règle et fournit en sortie un certain type de réponse. D'après cette définition, deux classifieurs qui diffèrent par au moins l'un de ces paramètres (espace de caractéristiques, données d'apprentissage, règle de décision et type de sortie) ne donnent pas les mêmes résultats et sont donc considérés comme différents.

Le classifieur finalement a pour but de fournir en sortie une réponse à propos de la forme à reconnaître. Selon Xu [162], cette réponse est divisée en cinq catégories suivant le niveau d'information apporté par le classifieur : de type classe, rang, mesure, ensemble et oracle. Cette catégorisation est adoptée dans la majorité des travaux.

3.2.1. Type classe

Ce type est nommé aussi type ou niveau abstrait. Par définition, tout classifieur est capable de produire une étiquette pour x , le niveau abstrait est donc le plus universel. [223]

$$e_j(x) = C_i, i \in \{1, \dots, N\} \quad (3.1)$$

C_i est la classe parmi N classes attribuées par le classifieur e_j à x . Dans ce cas, l'avis du classifieur est binaire. On peut alors représenter la réponse du classifieur par un vecteur binaire dans lequel « 1 » indique la classe proposée par le classifieur. Un classifieur peut aussi produire un ensemble de classes. Il considère alors que la forme x appartient à une des classes de cet ensemble sans donner d'autres informations permettant de discriminer les classes.

3.2.2. Type rang

Il s'agit d'un classement sur les classes. Le classifieur indique ce classement en fournissant en sortie un vecteur des rangs de taille N.

$$e_j(x) = (r_{1,j}, r_{2,j}, \dots, r_{N,j}) \quad (3.2)$$

r_{ij} est le rang attribué à la classe C_i par le classifieur e_j .

La classe placée au premier rang de la liste proposée par le classifieur est considérée comme la plus probable pour la forme x et la classe du dernier rang est la moins probable. Ce type est particulièrement adapté aux problèmes avec un grand nombre de classes, par exemple, reconnaissance des caractères, visage, reconnaissance de parole, ... etc. [5].

3.2.3. Type mesure

Elle indique le niveau de confiance du classifieur dans sa proposition. La sortie du classifieur est donc un vecteur de mesures de taille N.

$$e_j(x) = (m_{1,j}, m_{2,j}, \dots, m_{N,j}) \quad (3.3)$$

Où m_{ij} est la mesure attribuée à la classe C_i par le classifieur e_j .

Cette mesure, normalisée ou non, peut-être une distance, une probabilité a posteriori, une valeur de confiance, un score, une fonction de croyance, une possibilité, une crédibilité, une mesure floue, ... etc. L'auteur dans [223] suppose que les sorties contiennent des valeurs comprises entre 0 et 1.

3.2.4. Type ensemble

Le classifieur donne sa réponse sous forme d'un ensemble de classes candidates sans préciser ses préférences [106]

$$e_j(x) = \{C_i / i \leq m\} \quad (3.4)$$

3.2.5. Type oracle

On sait qu'également la sortie du classifieur e_i pour un x donné est soit correcte, soit fausse. Nous ignorons délibérément les informations concernant l'étiquette de classe qui a été attribuée. La sortie d'oracle est artificielle car nous ne pouvons l'appliquer qu'à un jeu de données étiqueté. Pour un ensemble de données, le classifieur e_i produit un vecteur de sortie y_i tel que [223].

$$y_{ij} = \begin{cases} 1, & \text{si } e_j \text{ classifie l'objet } x_i \text{ correctement} \\ 0, & \text{si non} \end{cases} \quad (3.5)$$

Chaque type de sortie (classe, rang, mesure ou oracle) correspond à un niveau d'information différent fourni par le classifieur. La sortie de type classe est la plus simple mais la moins riche en information. La sortie de type rang reflète l'ordre de préférence des propositions fournies par le classifieur. Celle de type mesure est la plus riche en information puisqu'elle reflète le niveau de confiance du classifieur dans ses propositions.

Les sorties de type rang ou de type mesure, peuvent être transformés en type classe (avec perte d'informations). Ceci consiste à tenir compte uniquement de la première solution de la liste proposée par chaque classifieur. Pour le type rang, il suffit de choisir la classe qui est placée au premier rang. Les autres classes ne seront pas prises en compte. Il suffit de choisir la classe ayant la meilleure mesure (valeur minimale ou maximale, selon que la mesure est croissante ou décroissante).

3.3. Système multi-classifieurs

De nombreux chercheurs se sont concentrés sur les MCS et par conséquent, de nombreuses nouvelles solutions ont été dédiées à chacune des trois phases possibles du MCS à savoir : génération, sélection et intégration (figure 3.1), qui seront détaillées dans la section 3.5. Un groupe de classifieurs est généré dans la première phase. Dans la deuxième phase, un ou un sous-ensemble de ces classifieurs est sélectionné, tandis que dans la dernière phase, une décision finale est prise sur la (les) prédiction (s) du (des) classifieur (s) sélectionné (s) [224].

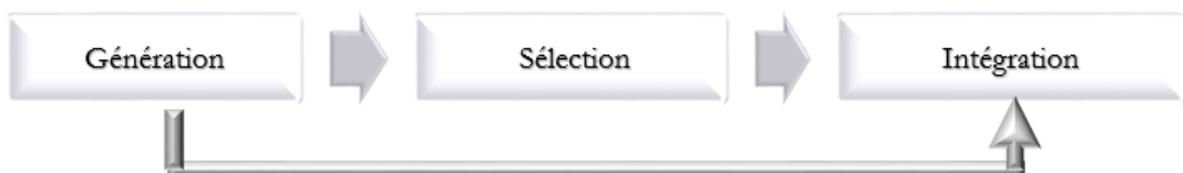


Figure 3.1 : Les phases d'un système multi-classifieurs

La première phase est responsable de la génération d'un groupe de classifieurs de base, prenant en compte une stratégie donnée, afin de créer des experts divers et précis. Un groupe peut être composé de classifieurs homogènes (même classifieurs de base) ou hétérogènes (classifieurs de base différents). Dans les deux cas, une certaine diversité est attendue. L'idée est de générer des classifieurs qui font des erreurs différentes et, par conséquent, montrent un certain degré de complémentarité. Dans les groupes homogènes, la diversité est obtenue en faisant varier les informations utilisées pour construire leurs éléments, telles que la modification des paramètres

initiaux, l'utilisation de différents sous-ensembles de données d'apprentissage (*Bagging, Boosting*) ou de l'utilisation de différents types de paramètres caractéristiques. D'autres part, l'idée de base des groupes hétérogènes est d'obtenir des experts qui diffèrent en termes de propriétés et de concepts sur lesquels ils sont basés.

Concernant la phase de sélection d'un MCS, le type de sélection et la notion de compétence du classifieur sont importants. Le type de sélection peut être statique ou dynamique. Habituellement, la sélection est effectuée en estimant la compétence des classifieurs disponibles sur les paramètres caractéristiques. À cette fin, un processus de partitionnement est couramment utilisé pendant les phases de l'apprentissage ou de test du MCS. Dans ce processus, l'espace paramètres caractéristiques est divisé en différentes partitions et les classifieurs les mieux adaptés à chacune d'elles sont déterminés.

La troisième phase d'un MCS consiste à appliquer les classifieurs sélectionnés pour reconnaître un motif de test donné. Dans les cas où tous les classifieurs sont utilisés (sans sélection) ou lorsqu'un ensemble est sélectionné, une méthode de fusion est nécessaire (voir section 3.6.).

Il est à noter qu'une telle représentation n'est pas unique, car les phases de sélection et d'intégration peuvent être facultatives. Par exemple, on peut trouver des systèmes MCS dans lesquels le groupe entier de classifieurs est utilisé sans aucune sélection ni de systèmes dans lesquels un seul classifieur est sélectionné dans le groupe, ce qui rend la phase d'intégration inutile [224].

En résumé, les contributions récentes à la première phase indiquent que la direction la plus prometteuse consiste à générer un groupe de classifieurs précis et divers. Une condition nécessaire et suffisante pour qu'un ensemble de classifieurs soit plus précis que n'importe quel de ses membres individuels est que les classifieurs soient précis et divers. Un classifieur donne un taux d'erreur inférieur à la prédiction aléatoire sur de nouveaux échantillons, alors que deux classifieurs sont différents s'ils commettent des erreurs différentes sur de nouveaux échantillons. La raison est que les différents classifieurs précis du groupe peuvent se faire concurrence en faisant des erreurs différentes et peut-être complémentaires. Au cours de la phase de sélection, des résultats intéressants ont été obtenus en sélectionnant des classifieurs spécifiques pour chaque motif de test, ce qui caractérise une sélection dynamique de classifieurs, au lieu d'utiliser le même classifieur pour tous (sélection statique). De plus, des contributions supplémentaires ont été observées lorsque des ensembles sont sélectionnés au lieu d'un seul classifieur. Dans un tel cas,

les sorties des classifieurs sélectionnés doivent être combinées et la troisième phase du MCS est nécessaire. Les principales contributions de cette phase ont consisté en différentes stratégies combinant les classifieurs et l'hypothèse selon laquelle le meilleur choix d'intégration dépend généralement du problème [224].

3.4. Les Stratégies de combinaison

Trois stratégies pour la combinaison des classifieurs mises au point traitent les données de manière différentes. Il s'agit de la stratégie parallèle, séquentielle et hybride [5, 225, 226]. D'autres organisations avec bouclage ou avec interaction sont aussi possibles. Vous pouvez faire référence à [5] pour un panorama assez complet de ces approches.

3.4.1. Approche séquentielle

La combinaison séquentielle ou série, appelée aussi combinaison verticale par Rahman et al [225] et pipeline dans [176] ou encore série ou concaténé. Les décisions sont en niveaux successifs. Elle permet de réduire progressivement le nombre de classes possibles. Dans chaque niveau, il existe un seul classifieur qui prend en compte la réponse fournie par le classifieur placé en amont afin de traiter les rejets ou confirmer la décision obtenue sur la forme qui lui est présentée (figure 3.2). Une telle approche peut être vue comme un filtrage progressif des décisions dans la mesure où elle permet de diminuer au fur et à mesure le taux d'erreur globale de la reconnaissance.

Cette approche demeure particulièrement sensible à l'ordre dans lequel sont placés les classifieurs. Les premiers classifieurs doivent être robustes, i.e. qu'en cas de mauvaise décision du premier classifieur, placé en amont de la série des classifieurs utilisés, l'erreur se répandra de manière irréversible. Il faudra donc choisir judicieusement le premier classifieur afin d'éviter - autant que possible- l'apparition d'une telle situation.

Notons que dans cette approche, chaque classifieur est réglé en fonction du classifieur placé en amont de la chaîne. Une simple modification du premier classifieur peut provoquer un ré-paramétrage (ré-apprentissage) des classifieurs suivants (figure 3.2).



Figure 3.2 : Combinaison séquentielle de classifieurs

3.4.2. Approche parallèle

Dans l'approche parallèle, les différents classifieurs opèrent indépendamment les uns des autres puis fusionnent leurs réponses respectives. Cette fusion est faite soit de manière démocratique, dans le sens où elle ne favorise aucun classifieur par rapport à un autre, soit au contraire dirigée et, dans ce cas, on attribue à la réponse de chaque classifieur, un poids en fonction de ses performances. L'ordre d'exécution des classifieurs n'intervient pas dans cette approche. La figure 3.3 présente la combinaison parallèle de classifieurs [227].

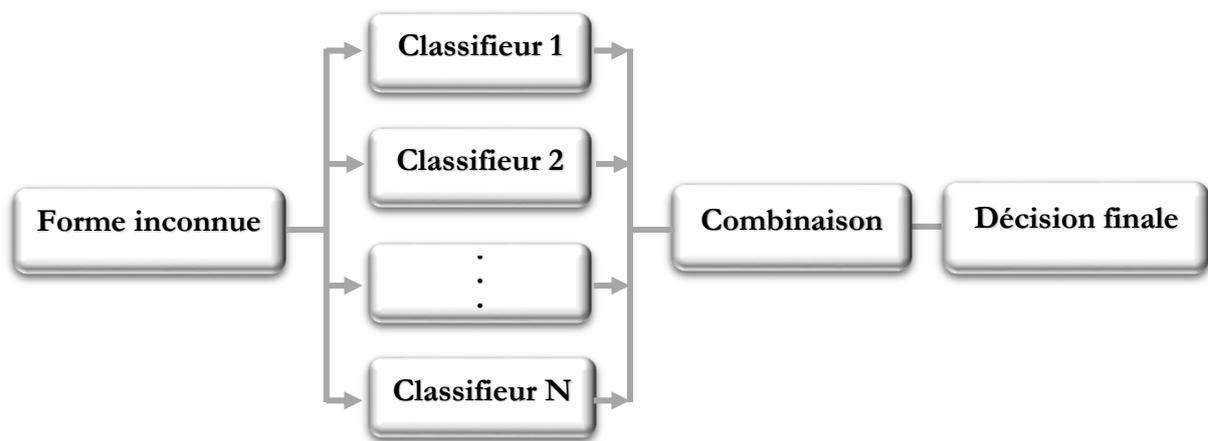


Figure 3.3 : Combinaison parallèle de classifieurs

Plusieurs termes ont été utilisés pour faire allusion à ces systèmes [5], nous pouvons citer *combinaison de classifieurs*, *fusion de classifieurs*, *mixture d'experts*, *comité de classifieurs*, *agrégation de consensus*, *sélection de classifieurs*, *ensemble de classifieurs*, *système multi-experts*, *système modulaire*, *groupe de classifieurs*, ... etc.

3.4.3. Approche hybride

L'approche hybride consiste à combiner à la fois des architectures séquentielles et parallèles afin de tirer pleinement avantage de chacun des classifieurs utilisés. La figure 3.4 illustre une représentation de la combinaison hybride dans laquelle on combine un classifieur série avec deux classifieurs en parallèle. Ce type d'approche permet de générer de nombreux schémas de coopération qui peuvent rapidement devenir complexes à optimiser [5].

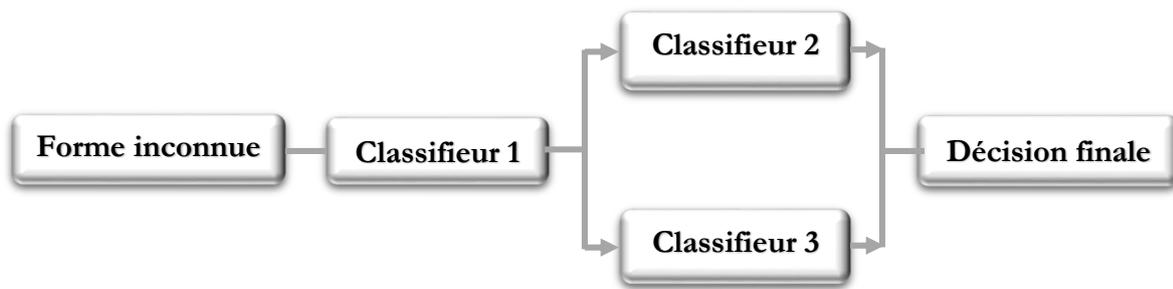


Figure 3.4 : Combinaison hybride de classifieurs

D'autres schémas de combinaison de classifieurs ont été proposés dans la littérature. Une approche de combinaison avec bouclage où le résultat de combinaison peut à son tour être utilisée par les classifieurs et modifie ainsi leurs sorties. Il y'a aussi une combinaison avec interaction entre les classifieurs où les classifieurs interagissent entre eux, et modifient leurs résultats avant que la combinaison soit réalisée. Une étude dans [228] propose un classifieur unique basé sur une classification multiple (Single Classifier-based Multiple Classification Scheme –SMCS-) utilisant un seul classifieur pour générer plusieurs classifications pour un point de test donné. Enfin, nous citons une combinaison utilisant directement la forme d'entrée en même temps que les réponses des classifieurs.

3.5. Les taxonomies de combinaison

Plusieurs méthodes et taxonomies des méthodes ont été proposées dans la littérature selon la façon dont on traite les classifieurs [5, 162, 224, 226, 229]. Une excellente revue de la littérature de ces taxonomies est présentée dans [5]. Ces taxonomies permettent de distinguer les méthodes de combinaison en fonction de différents critères comme le type de sorties des classifieurs, la capacité d'apprentissage de la combinaison, les stratégies de combinaison, la structure des classifieurs. Cependant, le type de sorties des classifieurs est un critère utilisé pratiquement dans toutes les taxonomies présentées dans la littérature.

Zouari [5] a proposé une taxonomie qui réunit de façon synthétique les trois critères les plus importants à savoir les stratégies de combinaison, la capacité d'apprentissage de la combinaison et le type de sorties des classifieurs (figure 3.5).

Cette taxonomie distingue au premier niveau les méthodes de sélection et des méthodes de fusion (agrégation). Les premières utilisent un traitement supplémentaire qui consiste à choisir le ou les meilleur(s) classifieur(s) parmi l'ensemble disponible, il s'agit de l'approche modulaire où le problème à traiter est divisé en sous-problèmes. Chaque classifieur ou module se spécialise dans un sous-problème d'où le nom expert. Concernant les méthodes de fusion, elles tiennent compte de toutes les sorties des classifieurs.

Au deuxième niveau viennent les méthodes de sélection statique et les méthodes de sélection dynamique. Les premières prennent en compte les mêmes classificateurs pour reconnaître toutes les formes. Les secondes sont adaptatives dans la mesure où elles cherchent le meilleur sous-ensemble de classificateurs en fonction des données présentées en entrée.

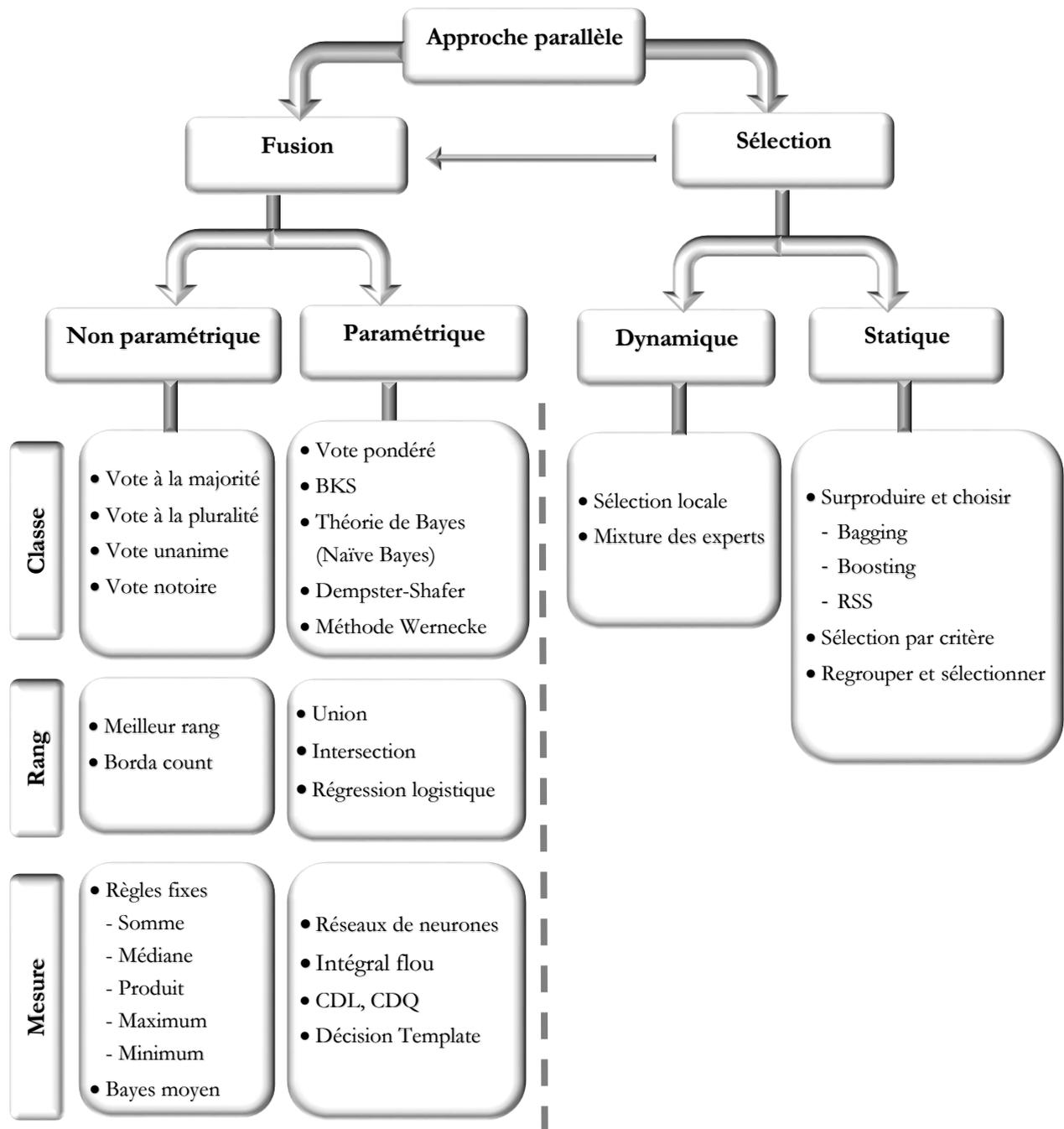


Figure 3.5 : Taxonomie des méthodes de combinaison parallèle de classificateurs

Dans les méthodes de fusion, les classificateurs ne sont pas nécessairement performants pour toutes les données, mais seulement dans leurs régions d'expertise. La fusion suppose la présence de tous les classificateurs dans la prise de décision. Chaque classificateur participe de manière

indépendante sur le même problème et les sorties de tous les classifieurs sont combinées par un certain opérateur de combinaison.

Les notions d'ensemble, groupe et comité, sont très utilisées pour faire référence au système de fusion. On distingue les méthodes dites *figées* ou *non-paramétriques* (les sorties des classifieurs sont combinées dans un schéma dont les paramètres sont invariables). Les méthodes *paramétriques* dites avec apprentissage qui cherchent à apprendre, sur les données disponibles, les paramètres nécessaires à la règle de décision de la combinaison.

Enfin, la complexité de ces méthodes peut varier en fonction du niveau d'information associé aux réponses fournies par les classifieurs à combiner (sortie de types classe, rang ou mesure).

Britto et al. [224] suggèrent une taxonomie présentée dans la figure 3.6, et se sont concentré sur la deuxième phase d'un MCS, en particulier, les approches basées sur la sélection dynamique (DS) par l'évaluation de la compétence des classifieurs. Les mesures de compétence sont organisées en deux types à savoir individuel et groupe.

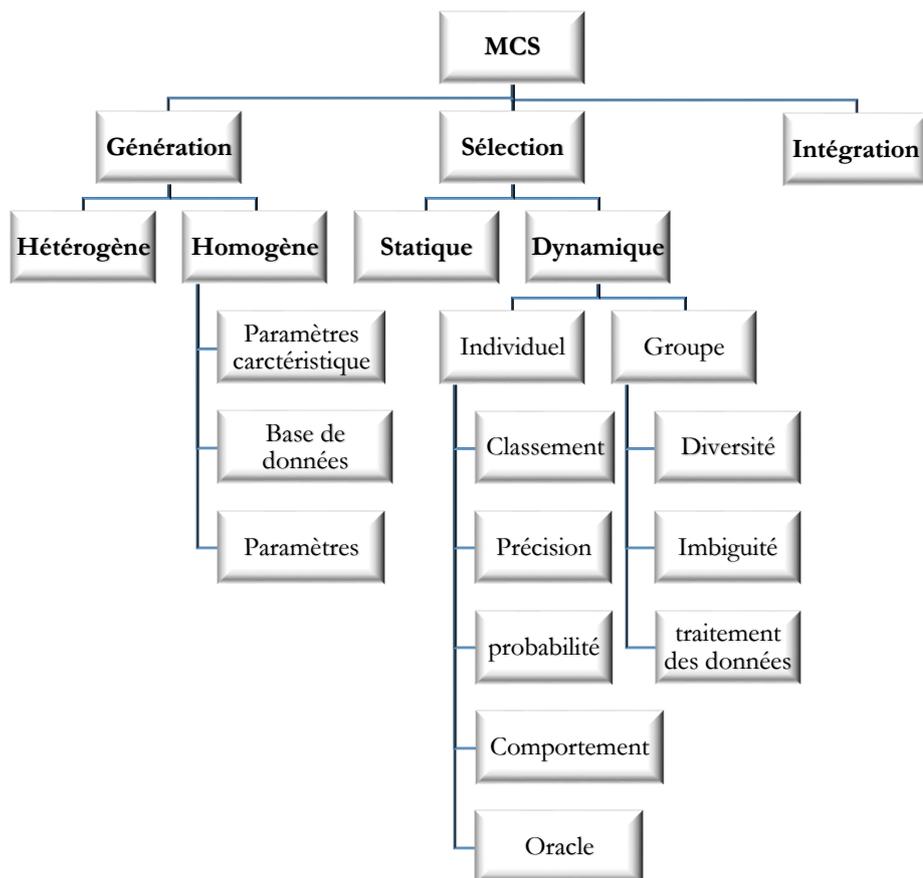


Figure 3.6 : Taxonomie dans le contexte du MCS

La première présente les mesures dans lesquelles, d'une manière ou d'une autre, la performance individuelle de chaque classifieur est la principale source d'information. Les classifieurs sont sélectionnés en fonction de leurs compétences individuelles sur l'ensemble de l'espace caractéristique représenté par la base d'apprentissage ou de validation, ou sur une partie de celui-ci, appelée région locale (la région locale peut être définie à l'avance, pendant la phase d'apprentissage, en utilisant des techniques de partitionnement, ou pendant la phase de test, en utilisant la règle PPV pour définir les K plus proches voisins du modèle inconnu). Cette catégorie a été subdivisée en cinq autres sous-catégories : classement, précision, probabiliste, comportement et Oracle.

La deuxième est composée des mesures qui prennent en compte l'interaction entre les éléments du groupe. Cette catégorie est subdivisée en trois sous-catégories: diversité, ambiguïté et traitement des données.

Les principaux concepts de base liés à la DS et les algorithmes et méthodes clés disponibles dans la littérature sont présentés en détail dans [224]. Certaines applications rapportées sur DS sont résumées dans [173]. La DS est très utilisée récemment dans différents domaines comme la reconnaissance de visage [230], de signature [16].

3.6. Les méthodes de combinaison des classifieurs

Pour l'intégration des sorties du classifieur, différents schémas sont disponibles dans la littérature. Les détails complets concernant les méthodes de combinaison et leur taxonomie sont disponibles dans [5, 9, 222, 224, 231, 232]. Dans ce qui suit, nous présentons parmi les nombreuses règles de combinaison de classifieurs multiples suggérés dans la littérature, les méthodes de combinaison (fusion et sélection) :

3.6.1. Fusion des classifieurs

- **Type classe**

La combinaison de type classe peut être utilisée pour tout type de classifieur (classe, rang ou mesure). Elle est basée sur le principe du vote qui consiste à interpréter chaque sortie d'un classifieur comme un vote pour l'une des classes possibles. La classe ayant un nombre de votes supérieur à un seuil préfixé est retenu comme décision finale. Ces méthodes sont les plus simples à mettre en œuvre. Elles contiennent trois catégories : vote avec seuil, majorité sans conflit et majorité notoire.

$$E(x) = \begin{cases} C_i & \text{si } \sum_{j=1}^L e_{i,j} = \max_{i=1}^N \sum_{j=1}^L e_{i,j} \geq \lambda \cdot L \\ \text{rejet} & \text{si non} \end{cases} \quad (3.6)$$

λ correspond à la proportion de classifieurs.

Si $\lambda=0$, il s'agit du *vote à la pluralité* où la classe qui reçoit le plus de votes est choisie comme classe finale. C'est la méthode la plus simple et est la plus facile à appliquer. Si toutes les classes ont le même nombre de votes (conflit) il y a un rejet.

Si $\lambda=0.5$, il s'agit du *vote à la majorité*. La classe finale celle proposée (décidée) par plus de la moitié des classifieurs. Elle est aussi très simple à appliquer.

Si $\lambda=1$, ce vote appelée *majorité unanime* où la classe finale est choisie si tous les classifieurs proposent cette réponse sinon la réponse finale est le rejet.

Les méthodes de combinaison paramétriques sont plus complexes à mettre en œuvre. Les méthodes les plus utilisées sont principalement le vote pondéré, la théorie de Bayes, la méthode de Dempster-Shafer [162, 233] et la méthode d'espace de connaissance du comportement (Behaviour Knowledge Space ou BKS) et la méthode de Wernecke [234, 235]. Pour plus d'informations sur la combinaison de type classe voir [225].

- **Type rang**

Les méthodes de type rang ont été développées essentiellement pour résoudre les problèmes des méthodes de vote. Elles combinent des listes de propositions de classifieurs ordonnées de manière décroissante selon leur préférence. Les rangs attribués par chaque classifieur peuvent être considérés comme des votes pour toutes les classes. Ayant un nombre N de classes, chaque classifieur attribue un poids N à la classe placée au début de la liste, un poids $N-1$ à la seconde classe, etc. La classe placée à la fin de la liste reçoit un seul vote. Les méthodes non-paramétriques de type rang les plus reconnues sont le Bordacount et le meilleur rang. Ces méthodes ne tiennent pas compte de la différence dans la performance des classifieurs (tous les classifieurs participent de manière égale dans la prise de décision).

Les méthodes paramétriques interviennent pour dépasser cette limite. Ces méthodes intègrent un degré pour chaque classifieur dans la combinaison surtout quand on sait qu'un des classifieurs est plus performant que les autres. Les méthodes de type rang les plus citées dans ce cas sont la somme pondérée, la régression logistique, l'intersection et l'union.

- **Type mesure**

Les méthodes de type mesure combinent des mesures qui reflètent le degré de confiance des classifieurs sur l'appartenance de la forme à reconnaître en chacune des classes. Toutefois,

comme les sorties des classifieurs ne sont pas toujours comparables, une normalisation est souvent nécessaire. Les méthodes de combinaison non-paramétrique les plus utilisées sont les règles fixes (somme, produit, minimum, maximum, médiane) et la moyenne de Bayes [173].

Il existe d'autres méthodes paramétriques pour combiner les classifieurs de type mesure tels que les règles pondérées, les méthodes floues [163, 164], Décision Template (TD) et les réseaux de neurones.

Le choix entre les méthodes paramétriques et non paramétriques est un problème posé. Les méthodes de combinaison non paramétriques sont les plus utilisées par les chercheurs. Cela est dû au fait que ces méthodes sont simples à mettre en œuvre et n'utilisent pas de traitements supplémentaires (apprentissage). Le tableau 3.1 présente les principales caractéristiques des méthodes de combinaison parallèle de classifieurs en fonction du type de sorties des classifieurs et de la capacité d'apprentissage. Lors de la conception d'un système de combinaison de classifieurs le choix des méthodes de combinaison se fait selon ces critères.

Tableau 3.1 : Caractéristiques des méthodes de combinaison parallèle [5]

Méthode de combinaison	Type	Apprentissage	Commentaire
Vote	Classe	Non	-Très simple -Utilisable pour un grand nombre de classifieurs
Somme, Médiane, Moyenne	Mesure	Non	-Robuste et suppose l'indépendance des classifieurs
Produit, Max, Min	Mesure	Non	-Suppose l'indépendance des classifieurs -utilisable avec des classifieurs performants
Borda count	Rang	Non	-Utilisable pour un très grand nombre de classes
Regression logistique	Rang	Oui	-Dépend des poids des pondération
Behaviour Knowledge Space (BKS)	Classe	Oui	-Demande beaucoup de données -N'utilise pas d'hypothèse d'indépendance de classifieurs
Théorie de Bayes	Classe	Oui	-Utilisée avec l'hypothèse d'indépendance des classifieurs

La littérature montre que la combinaison des sorties de type classe a retenu une attention des chercheurs plus grande que la combinaison des autres types de sorties. Elle est très simple à appliquer et à analyser expérimentalement et théoriquement. Les méthodes de type mesure quand on peut les utiliser exploitent toute l'information fournie par le classifieur, Parker [236] a confirmé que cela ne veut pas dire qu'elles sont toujours plus intéressantes vis à vis des autres méthodes de type rang. Il a montré que les méthodes de type rang peuvent être plus performantes que les méthodes de type classe et mesure. Kittler et al. [237] ont montré que, pour les distributions Gaussiennes, la règle de type mesure « Somme » dépasse toujours le résultat du « Vote » et pour les distributions à queue lourde (heavy tail) la règle de « Vote » peut surperformer la règle de « Somme ».

3.6.2. Sélection des classifieurs

Le problème de sélection des classifieurs est traité comme étant un problème d'optimisation de l'espace de décisions des classifieurs. Lorsqu'on dispose d'un grand nombre de classifieurs, la question posée est de savoir combien et quels classifieurs choisir pour améliorer la reconnaissance, i.e. améliorer la performance avec le minimum de classifieurs ? Cependant, sélectionner les classifieurs ayant les performances les plus élevées n'est pas forcément la bonne solution pour aboutir au meilleur résultat [5].

Les méthodes de sélection de classifieurs sont divisées en deux groupes : statique et dynamique (annexe2). La sélection statique consiste à choisir une fois pour toutes le *meilleur* classifieur ou le *meilleur* sous-ensemble de classifieurs, parmi un grand nombre de classifieurs disponibles, qui sera utilisé lors de la combinaison. La sélection dynamique (adaptative) retient les sorties d'un ou de plusieurs classifieurs adaptés à chacune des formes de test à traiter.

Selon Zouari [5] certains chercheurs considèrent que les méthodes de sélection font partie des méthodes de combinaison paramétriques (avec apprentissage) puisque les résultats d'apprentissage des classifieurs sont utilisés dans la procédure de sélection. Mais selon lui les méthodes paramétriques utilisent une phase supplémentaire permettant d'estimer les paramètres nécessaires à la combinaison indépendamment de l'apprentissage des classifieurs.

Avant de conclure, une terminologie de base du domaine, une discussion sur l'efficacité des algorithmes de combinaison, le concept de diversité [73], les méthodes de création d'un ensemble de classifieurs, des approches permettant de combiner les décisions de chaque classifieur, Des études et aussi des orientations futures possibles sont présentés dans [188]. Une étude examine les relations entre différentes méthodes de combinaison et différentes mesures de la diversité

dans [234, 235]. Dans [226] l'auteur présente une étude actualisée sur les MCS du point de vue des systèmes intelligents hybrides. L'article traite des principaux problèmes, tels que la diversité et les méthodes de fusion décisionnelle, offrant une vision sur des applications en cours de développement.

3.7. Conclusion

Dans ce chapitre, nous avons défini ce qu'est un classifieur dans le cadre de la combinaison. Nous avons présenté les stratégies de combinaison possibles qui se différencient essentiellement par l'agencement des classifieurs, les taxonomies et les méthodes de combinaison spécifiques qui diffèrent essentiellement par leur capacité d'apprentissage et le type de sortie des classifieurs.

Les diverses méthodes de combinaison que nous avons présentées se distinguent essentiellement par le niveau d'information en sortie qu'apporte chacun des classifieurs. Au niveau classe, la sortie de chaque classifieur est une étiquette. Au niveau rang, elle est une liste ordonnée de solutions. Au niveau mesure, elle propose en plus des mesures reflétant la confiance qu'il a dans les classes.

Malgré la diversité des travaux sur la combinaison cela n'a pas permis d'avoir une approche générale qui traite le problème de la combinaison de manière satisfaisante dans tous les cas. Il est difficile de prouver que tel classifieur est meilleur que tel autre pour tout problème de classification.

Il n'est pas aisé non plus de prouver que telle méthode de combinaison convient mieux que telle autre pour tout problème de reconnaissance. Le problème actuel n'est alors plus de savoir si les méthodes de combinaison pourront améliorer les résultats de l'application traitée, mais de chercher dans quels cas leur emploi est nécessaire.

Il est évident que les travaux existants, théoriques et expérimentaux, n'ont pas permis de répondre de manière claire à ces questions importantes. Même pour les règles les plus utilisées comme le vote à la majorité et la somme, les travaux ne convergent pas toujours vers les mêmes conclusions.

Dans notre travail nous avons utilisé les trois type (classe, rang et mesure) pour la combinaison des classifieurs, en utilisant l'ensemble des classifieurs ou en sélectionnant les plus performants (meilleur sous-ensemble de classifieurs). Le chapitre quatre traite en détail notre choix et les techniques utilisées.

Chapitre 4 . PRESENTATION DU SYSTEME DEVELOPPE

- 4.1. Introduction
- 4.2. Base de données et prétraitements
- 4.3. Les paramètres caractéristiques
- 4.4. La classification
- 4.5. Evaluation des Performances
- 4.6. La Combinaison des classifieurs
- 4.7. Conclusion

Ce chapitre traite la conception d'un système de RdF adapté à la reconnaissance automatique des chiffres arabe imprimés ou bien manuscrits. Pour la réaliser, il est nécessaire de calculer des paramètres discriminants et de fournir ces paramètres à un classifieur.

Nous présentons en détail dans ce chapitre les outils que nous avons utilisé dans les trois parties du système de RdF. Nous décrivons ainsi les différentes méthodes que nous avons appliquées pour la combinaison des classifieurs afin de maximiser la performance en terme de taux de reconnaissance.

4.1. Introduction

La connaissance des méthodes de prétraitement et d'extraction des caractéristiques est un sujet important dans la conception de toute application de RdF. Cette connaissance est généralement dédiée à une application. Au contraire, les méthodes de classification sont plus universelles et une technique de classification peut en général être appliquée à une grande variété d'applications différentes. Certains facteurs de conception, tels que le type d'entrée et la capacité d'ordinateur disponible, dictent les choix de conception de toutes les parties d'un système de RdF. Les ressources de calcul (informatique) doivent donc être soigneusement réparties entre les étapes d'un système de reconnaissance afin d'obtenir la meilleure performance possible.

Dans ce chapitre, nous allons présenter de manière détaillée les outils et les différentes techniques utilisées pour la réalisation du système de reconnaissance des chiffres arabe imprimés et manuscrits hors-ligne dans les trois grandes phases identifiées à savoir : le prétraitement, l'extraction des paramètres caractéristiques et la classification. Nous allons aussi expliquer les règles de combinaison appliquées.

4.2. Base de données et prétraitements

Pour évaluer le système proposé, nous avons utilisé deux bases d'images de chiffres l'une pour les manuscrits et l'autre pour les imprimés (annexe 4). Pour les chiffres manuscrits, des bases d'images connues comme : MNIST [159], CEDAR [238], USPS [239], DIGITS [240], un ensemble de données fourni par l'Institut autrichien de recherche sur l'intelligence artificielle [214].

Dans notre travail nous avons utilisé la base MNIST (Modified ou Mixed National Institute of Standards and Technology) qui est devenu une base standard de test et très utilisée en apprentissage automatique et en RdF (figure 4.1). Elle regroupe 60.000 images d'apprentissage et 10.000 images de test. C'est un sous ensemble d'une très grande base de donnée, appelée NIST. Les chiffres de MNIST sont des images en niveaux de gris, normalisées et centrées (par le centre de masse – gravité-) dans une image de taille 28*28 pixels.

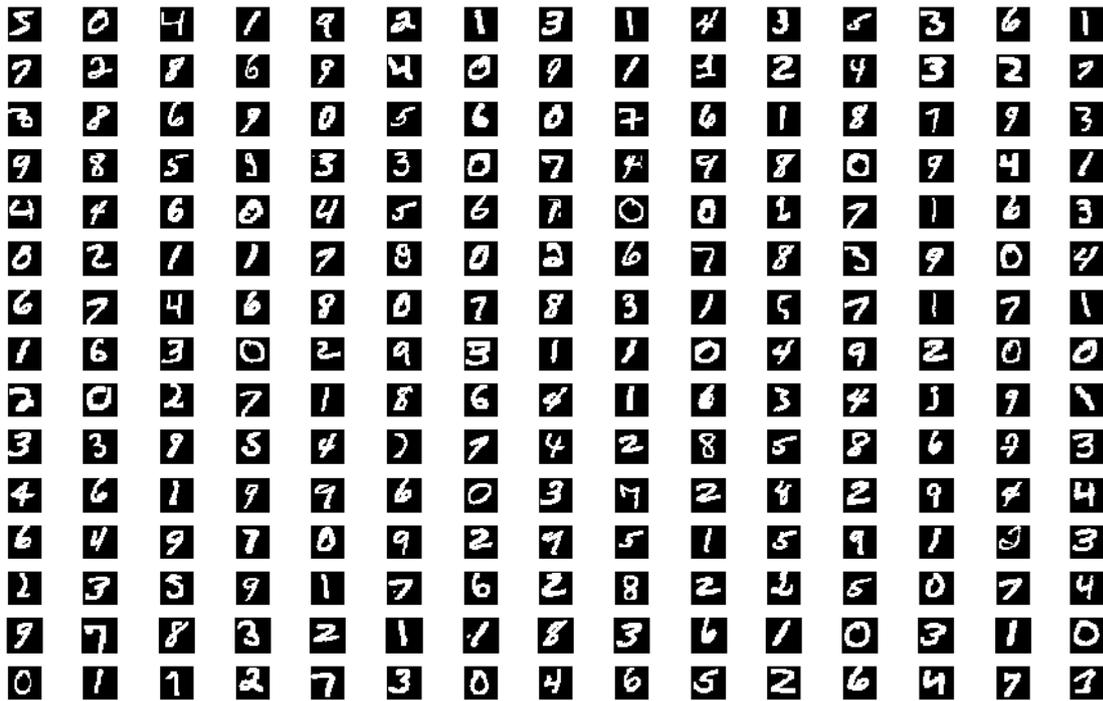


Figure 4.1 : Un exemple de chiffres extrait de la base MNIST

Le nombre d'images pour chaque chiffre n'est pas le même. Le tableau 4.1 présente le nombre d'images pour chaque chiffre et le pourcentage dans la base.

Tableau 4.1 : Nombre et pourcentage d'instances de chaque classe

Classe	Apprentissage		Test		Total
	Nombre	Pourcentage	Nombre	Pourcentage	
0	5923	9.87%	980	9.80%	6903
1	6742	11.24%	1135	11.35%	7877
2	5958	9.93%	1032	10.32%	6990
3	6131	10.22%	1010	10.10%	7141
4	5842	9.74%	982	9.82%	6824
5	5421	9.04%	892	8.92%	6313
6	5918	9.86%	958	9.58%	6876
7	6265	10.44%	1028	10.28%	7293
8	5851	9.75%	974	9.74%	6825
9	5949	9.91%	1009	10.09%	6958

Contrairement au chiffre manuscrit, il n'existe pas de base de données pour les chiffres imprimés, sauf une petite base récente dans le langage Matlab, ou la base SVHN (The Street View

House Numbers) [241]. Elle est obtenue à partir de numéros de maison dans les images de Google Street View. Pour cette raison, nous avons choisi de concevoir une nouvelle base de chiffres arabe imprimés. Nous avons construit une base de données composée de 6240 échantillons de chiffres en dix images (image pour chaque chiffre de 0 à 9). Chaque image contient 624 occurrences d'un chiffre de différentes tailles, polices –fonts- et styles (annexe 3). La figure 4.2 illustre l'image du chiffre 3. Pour l'acquisition des chiffres. Nous les avons saisis via le logiciel de traitement de texte Word en utilisant la majorité des polices (plus de 133 polices) et pour chaque police des tailles différentes (5, 8, 11, 12, 14, 18, 22) en utilisant aussi la mise en forme (style) normale, **gras**, *italique* et **gras italique**. Les images de chiffres ont été stockées au format PNG (Portable Network Graphics) par l'application Paint (logiciel graphique) de Windows. Nous allons utiliser l'abréviation BCI_1 pour cette base (base des chiffres imprimés_1).

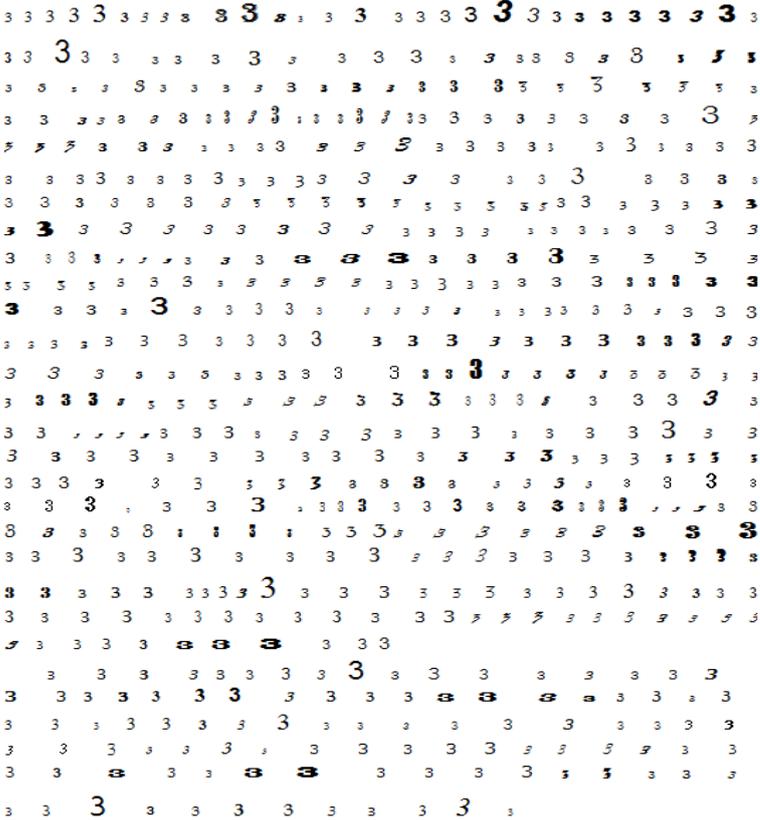


Figure 4.2 : Image du chiffre "3"

Nous avons utilisé aussi la base d'apprentissage et de test, de la référence [242] comme une base de test. Elle est composée de 5550 (4550+ 1000) échantillons de chiffres en dix images (image pour chaque chiffre de 0 à 9). Chaque image contient 555 occurrences d'un chiffre de différentes taille ces images sont sous le format BMP (BitMap) (figure 4.3 présente le chiffre 3). On va nommer cette base par BCI_2 (base des chiffres imprimés_2).

Pour montrer la faisabilité de l'approche une troisième base de chiffre BCI_3 (base des chiffres imprimés_3) est utilisée pour les tests. C'est un ensemble des images de chiffres synthétiques disponible sous Matlab (figure 4.6). Elles contiennent chacune un chiffre entouré d'autres chiffres. Elle comprend 101 images pour chacun des 10 chiffres.

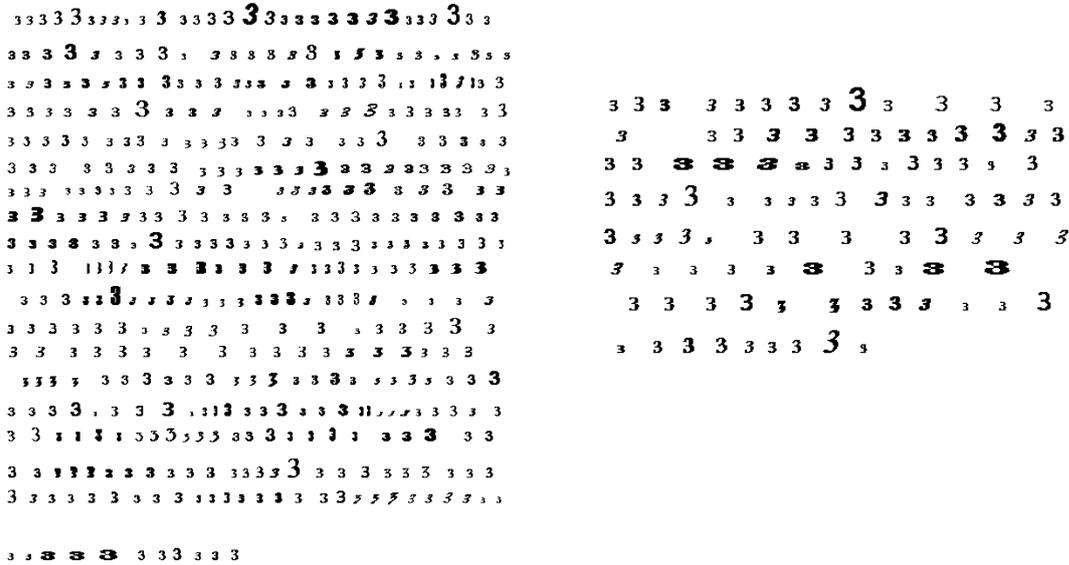


Figure 4.3 : Image du chiffre "3" de la référence [242]

Le prétraitement est une étape intermédiaire précédant la mise en place de la reconnaissance. Nous nous intéresserons ici à la reconnaissance de chiffres imprimés et manuscrits « off line », et sans contrainte d'écriture (une application typique pour les deux types est la reconnaissance de codes postaux). Le problème de la séparation des chiffres connectés ne sera pas étudié.

Dans le cadre de notre étude, l'objet à reconnaître est le chiffre (une image d'un chiffre), nous traiterons donc des chiffres préalablement isolés pour les chiffres manuscrits (MNIST). Les chiffres sont déjà isolés, chaque chiffre dans une image séparée (figure 4.1).

Pour les bases de chiffres imprimés, la première étape de prétraitement et la binarisation, il faut trouver le bon seuil pour ne pas déformer les chiffres, puis la segmentation (localisation) pour séparer les chiffres et les encadrés dans des fenêtre minimale en traçant un rectangle autour du chiffre (contour englobant chaque chiffre –recadrage–), comme le montre la figure 4.4.

Aussi nous nous sommes concentrés sur les prétraitements de la binarisation, l'élimination du bruit (figure 4.6) et la squelettisation (figure 4.5). Dans cette étape nous avons utilisés les fonctions prédéfinies de Matlab « im2bw, bwlabel, regionprops et bwmorph ».

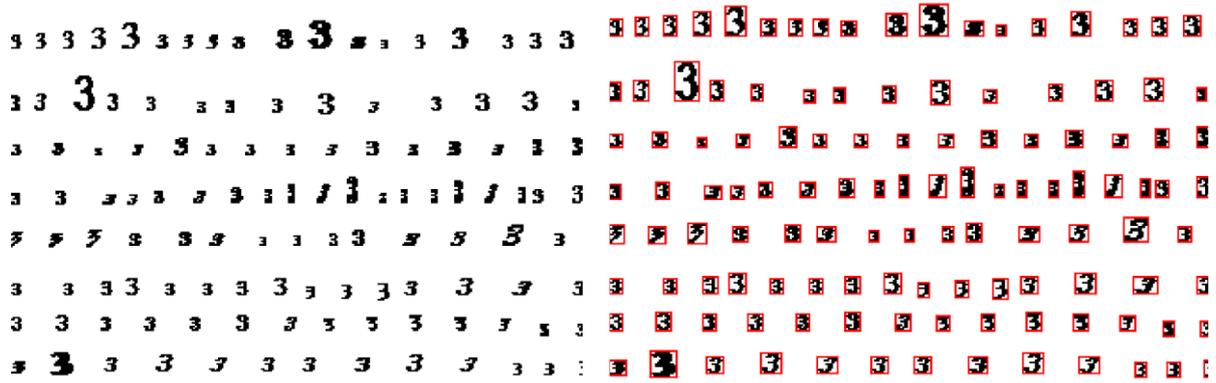


Figure 4.4 : Etape de localisation

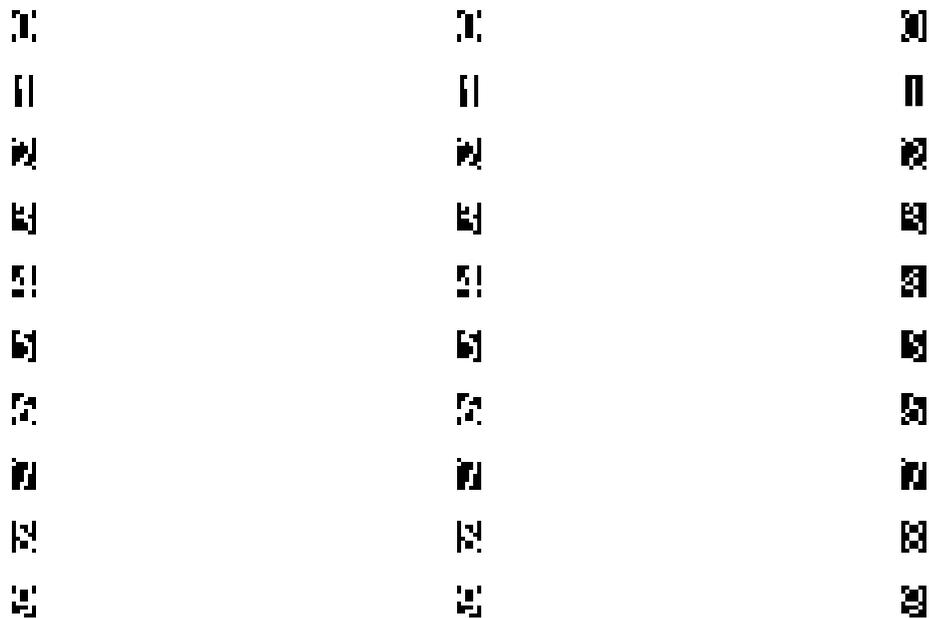


Figure 4.5 : Squelettisation

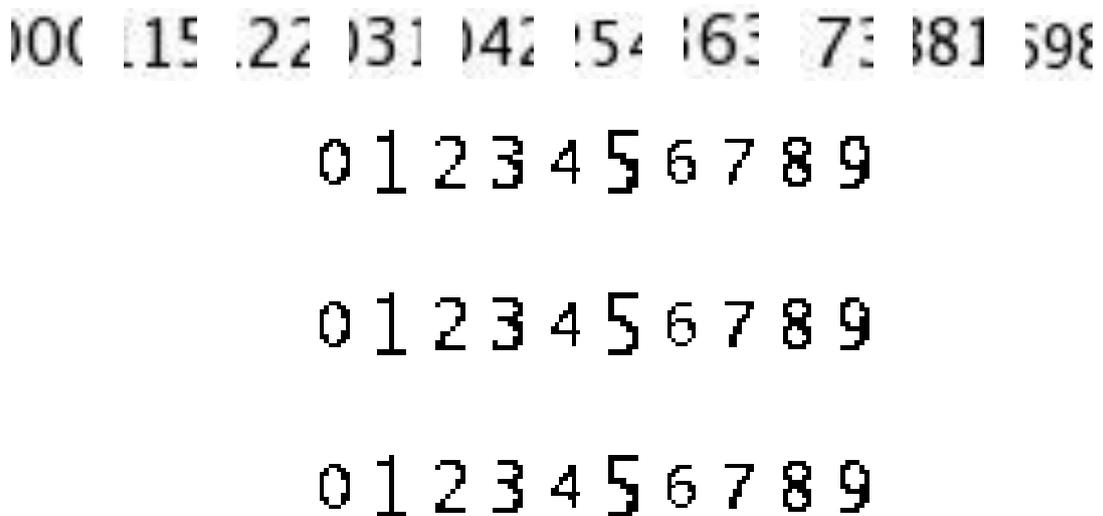


Figure 4.6 : Chiffres synthétique sous Matlab avant et après prétraitement

4.3. Les paramètres caractéristiques

L'extraction des paramètres caractéristiques consiste à représenter la forme par une liste de longueur fixe de "n" variables numériques appelée vecteur caractéristique. L'équilibre entre la longueur du vecteur caractéristique et la dimension de la base d'apprentissage est obligatoire. La classique règle générale indique que le nombre d'échantillon par chaque classe dans la base d'apprentissage devrait être au moins égal de 5 à 10 fois la dimension du vecteur caractéristique [6].

Selon l'information à modélisé Il existe un très grand nombre de descriptions possibles pour une forme. Nous avons choisi les descriptions les plus couramment utilisés dans ce domaine et qui vérifient les propriétés mentionnées dans la section 2.4. Nous avons concentré dans notre travail sur la propriété qui répond à l'indépendance aux transformations géométriques à travers les moments. Nous proposons quelques concaténations entre ces paramètres caractéristiques (chapitre 5). Dans ce qui suit nous les présentons en détail.

4.3.1. Les moments géométriques

Pour une image numérisée et discrète (codage rétinien) de taille $K \times L$, $g(k,l)$ est la fonction décrivant le contenu de l'image en chacun de ses points de coordonnées k, l . Et i et j étant deux entiers naturels quelconques [243] :

$$M_{ij} = \sum_{k=1}^K \sum_{l=1}^L (k)^i (l)^j g(k, l) \quad (4.1)$$

M_{ij} : est la définition du moment géométrique d'ordre $(i+j)$ avec la répétition $|i-j|$.

$g(k, l)$: présente la valeur 0 ou 1, et $(k)^i (l)^j$ est la base.

L'ensemble complet de moments d'ordre n : $\{M_{ij}, i+j \leq n\}$ est $(n+1)(n+2)/2$.

Afin de rendre les moments M_{ij} invariants à la translation, on les défini en choisissant (x_g, y_g) comme origine. On parle alors de moments centrés μ_{ij} dont la définition formelle est donnée par (4.2) et (4.3).

$$\mu_{ij} = \sum_{k=1}^k \sum_{l=1}^l (k - x_g)^i (l - y_g)^j g(k, l) \quad (4.2)$$

$$x_g = \frac{M_{10}}{M_{00}} \quad , y_g = \frac{M_{01}}{M_{00}} \quad (4.3)$$

En divisant chaque moment μ_{ij} par $(\mu_{00})^{(i+j+2)/2}$ nous construisons les moments normalisés η_{ij} qui sont invariants par homothétie (changement d'échelle, grossissement).

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{(i+j+2)/2}} \quad (4.4)$$

En résumé, les moments géométriques fournissent un codage compact et facile à calculer, doté de bonnes propriétés d'invariance théoriques. En fait, il n'y a pas de méthodes permettant de déterminer le nombre idéal de moments pour un problème donné, cet inconvénient est surmonté par les moments de Zernike qui possèdent une propriété d'orthogonalité permettant une reconstruction approchée de la forme à partir des moments [183].

4.3.2. Les moments Hu et les moments invariants étendus

Les sept moments de Hu sont définis à partir des moments normalisés η_{ij} d'ordre 2 et 3, qui sont invariants à la translation, l'échelle et la rotation [183, 208] :

$$\phi_1 = \eta_{20} + \eta_{02} \quad (4.5)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (4.6)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (4.7)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} - \eta_{03})^2 \quad (4.8)$$

$$\begin{aligned} \phi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03}) + (\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (4.9)$$

$$\phi_6 = (\eta_{20} + \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) + 4\eta_{11}(\eta_{03} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (4.10)$$

$$\begin{aligned} \phi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & - (\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (4.11)$$

Le tableau 4.2 donne les moments Hu pour différentes statuts du chiffre « 5 »

Tableau 4.2 : Moments Hu invariant du chiffre « 5 »

	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7
5	0.3595	0.0210	0.0012	0.0002	0.0000	0.0000	-0.0000
5	0.3595	0.0210	0.0012	0.0002	0.0000	0.0000	-0.0000
5	0.3595	0.0210	0.0012	0.0002	0.0000	0.0000	-0.0000
5	0.3595	0.0210	0.0012	0.0002	0.0000	0.0000	0.0000
5	0.3595	0.0210	0.0012	0.0002	0.0000	0.0000	0.0000
5	0.3258	0.0039	0.0002	0.0000	-0.0000	0.0000	0.0000
5	0.3345	0.0111	0.0003	0.0000	0.0000	0.0000	0.0000
5	0.3345	0.0111	0.0003	0.0000	-0.0000	-0.0000	0.0000

Les moments invariants étendus ont été utilisés comme descripteur d'image. Ces moments sont non linéaires, invariants à la translation, rotation, mise à l'échelle et inversion d'image [84]

$$\Phi_8 = \eta_{40} + \eta_{22} + \eta_{02} \quad (4.12)$$

$$\Phi_9 = (\eta_{40} - \eta_{04})^2 + 4(\eta_{31} - \eta_{13})^2 \quad (4.13)$$

$$\Phi_{10} = (\eta_{40} - 6\eta_{22} + \eta_{04})^2 + 16(\eta_{31} - \eta_{13})^2 \quad (4.14)$$

$$\begin{aligned} \Phi_{11} = & (\eta_{40} - 6\eta_{22} + \eta_{04})^2 [(\eta_{40} - \eta_{04})^2 + 4(\eta_{31} - \eta_{13})^2] \\ & + 16(\eta_{40} - \eta_{04}) + (\eta_{31} + \eta_{13})(\eta_{31} - \eta_{13}) \end{aligned} \quad (4.15)$$

$$\begin{aligned} \Phi_{12} = & (\eta_{40} - 6\eta_{22} + \eta_{04})^2 [(\eta_{40} - \eta_{04})^2 + 4(\eta_{31} - \eta_{13})^2] \\ & - 16(\eta_{40} - \eta_{04}) + (\eta_{31} + \eta_{13})(\eta_{31} - \eta_{13}) \end{aligned} \quad (4.16)$$

4.3.3. Les moments de Zernike

Dans l'application pratique, il est beaucoup plus simple pour calculer les moments de Zernike directement qu'à partir de leur définition (annexe 3), i.e. Les moments de Zernike peuvent également être dérivés de moments centrés μ_{ij} comme suit [112]:

$$A_{nl} = \frac{n+1}{\pi} \sum_{k=l}^n \sum_{j=0}^q \sum_{m=0}^l (-i)^m \binom{q}{j} \binom{l}{m} \beta_{nlk} \mu_{k-2j-l+m, 2j+l-m} \quad (4.17)$$

avec : $q=(k-l)/2$, $(l-k)$ et $(n-k)$ pairs.

$$\beta_{nlk} = \frac{(-1)^{(n-k)/2} [(n+k)/2]!}{[(n-k)/2]! [(l+k)/2]! [(k-l)/2]!} \quad (4.18)$$

$\binom{j}{s}$ Étant le nombre de combinaisons de s élément parmi j .

Nous avons utilisé Les moments Zernike invariants jusqu'à l'ordre 12 qui contient 49 variables données dans le tableau 4.3.

Tableau 4.3 : Les 12 ordres des moments de Zernike

Ordre	Moments de Zernike
0	A_{00}
1	A_{11}
2	A_{20}, A_{22}
3	A_{31}, A_{33}
4	A_{40}, A_{42}, A_{44}
5	A_{51}, A_{53}, A_{55}
6	$A_{60}, A_{62}, A_{64}, A_{66}$
7	$A_{71}, A_{73}, A_{75}, A_{77}$
8	$A_{80}, A_{82}, A_{84}, A_{86}, A_{88}$
9	$A_{91}, A_{93}, A_{95}, A_{97}, A_{99}$
10	$A_{10,0}, A_{10,2}, A_{10,4}, A_{10,6}, A_{10,8}, A_{10,10}$
11	$A_{11,1}, A_{11,3}, A_{11,5}, A_{11,7}, A_{11,9}, A_{11,11}$
12	$A_{12,0}, A_{12,2}, A_{12,4}, A_{12,6}, A_{12,8}, A_{12,10}, A_{12,12}$

4.3.4. Cavités

Nous avons utilisé les cavités comme des structures caractéristiques. Il existe cinq types de cavités : Est, Ouest, Nord, Sud, et Centrale. Cela suppose la définition de 4 directions cardinales dans l'image : Ouest vers la gauche, Nord vers le haut etc. Par exemple un point appartient à une cavité ouest si et seulement si les conditions suivantes sont simultanément vérifiées [191]

- Ce point n'appartient pas au tracé.
- A partir de ce point, en se déplaçant en ligne droite vers l'Ouest, on ne croise pas le tracé.
- A partir de ce point, en se déplaçant en ligne droite vers le Nord, le Sud, ou l'Est, on croise le tracé.

Par analogie nous obtenons les définitions des autres cavités. Pour la cavité centrale, un point lui appartient si et seulement si les conditions suivantes sont simultanément vérifiées :

- Ce point n'appartient pas au tracé.
- A partir de ce point, en se déplaçant en ligne droite vers l'Ouest, le Nord, le Sud, ou l'Est, on croise le tracé.

La figure 4.7 illustre quelques exemples de cavités détectées sur un ensemble des chiffres de la base MNIST. Chaque type de cavité est représenté par une couleur.



Figure 4.7 : Cavités : Est : rouge, Ouest : vert, Nord : bleu, Sud : jaune, et Centrale : rose

4.3.5. HOG

La technique calcule des histogrammes locaux de l'orientation du gradient sur des zones (bloque) régulièrement réparties sur l'image comme illustré sur la figure 4.8.

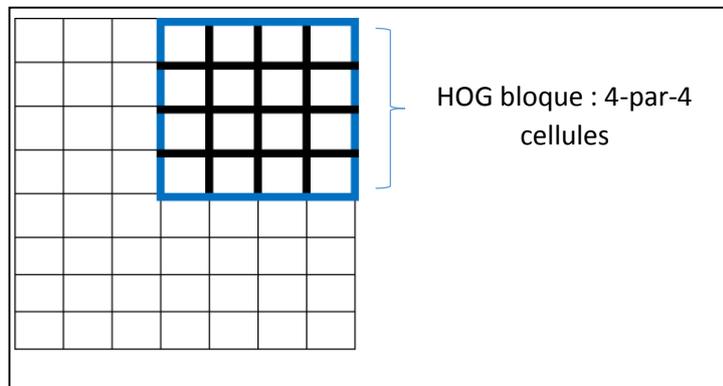


Figure 4.8 : Blocs et cellules d'une image

Avec une taille de bloque élevée, les modifications de l'éclairage local peuvent être perdue en raison du nombre volumineux de pixels dans le bloc. Par contre, Réduire la taille du bloc aide à saisir l'importance des pixels locaux, i.e. peut aider à supprimer les modifications de l'éclairage des fonctions HOG. Et Pour capturer beaucoup d'informations sur la forme (information spatial), il faut réduire la taille de la cellule, mais cela augmente de manière significative la dimensionnalité du vecteur de caractéristiques HOG (figure 4.9).

En pratique, les paramètres HOG doivent être modifiés en fonction de l'apprentissage et de tests répétés du classifieur afin d'identifier les paramètres optimaux. Après plusieurs tests nous avons utilisé un bloque du taille 2*2, et une cellule de taille 2*2 pixels.

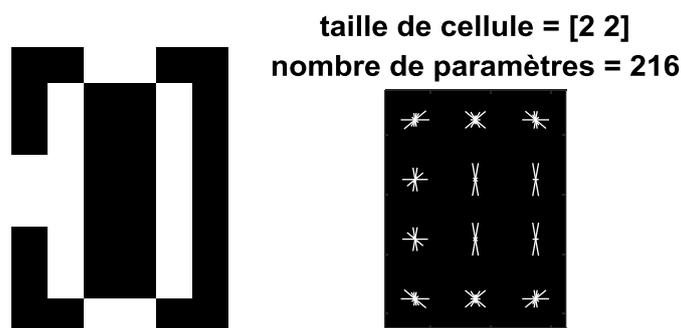


Figure 4.9 : Paramètres HOG du chiffre "0"

4.3.6. Zonage et multi-zonage

C'est la densité de pixels noir calculées dans différentes zones (zonage) de l'image du chiffre. Ces mesures sont ainsi obtenues : on « divise » horizontalement et verticalement le rectangle

englobant le chiffre en zones de taille égale. Le nombre de pixels noir dans chaque zone forme alors les composantes du vecteur de caractéristiques. En découpant par exemple l'image en "n" zones verticales d'égale largeur et "m" zones horizontales d'égale hauteur, nous obtenons un vecteur à n*m composantes. A titre d'exemple nous présentons sur la figures 4.10 trois cas de zonage : 2*2, 1*2, 2*1. L'utilisation de plusieurs zonages c'est le multi-zonage.

Les densités dans chaque zone devront être normalisées (en les divisant par exemple par la surface de la zone -moyenne-) puisque les chiffres ne sont pas tous de même taille.



Figure 4.10 : Multi-zonage du chiffre "2"

4.3.7. Profil

Les caractéristiques à extraire des chiffres sont leurs profils gauche et droit. Nous déterminons sur un certain nombre de lignes, en général, réparties uniformément sur la hauteur du chiffre, la distance entre le bord gauche (respectivement droite) du chiffre et le premier pixel noir rencontré sur cette ligne (figure 4.11). L'ensemble de ces distances définit un profil gauche, respectivement droite, du chiffre. Nous avons utilisé 12 profils et ce après plusieurs essais de différents nombres de profils.

La variation de taille représente un problème difficile pour la reconnaissance, dans ce cas, une méthode de normalisation de la taille devrait être capable de gérer les variations de taille. Dans notre cas Les profils gauche et droit devront être normalisés en les divisant par la largeur de l'image du chiffre traité puisque les chiffres ne sont pas tous de même taille.

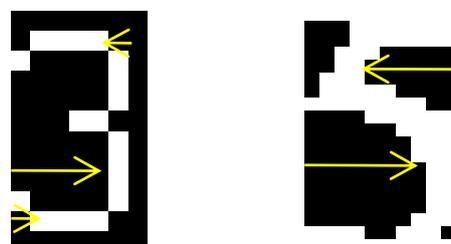


Figure 4.11 : Profil horizontale du chiffre "5" et "3"

4.3.8. Histogramme de projection

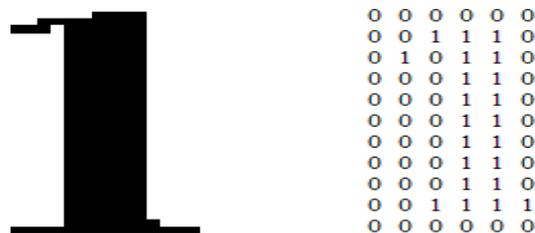
La projection peut être horizontale ou verticale. Elle présente le nombre de pixel de chaque ligne, respectivement colonne, (figure 4.12). Elle peut être indépendante de l'échelle en divisant par le nombre total de pixels de la forme d'image.



Figure 4.12 : Histogramme de projection horizontale et verticale du chiffre "5"

4.3.9. Rétine (le codage rétinien)

Nous prenons les pixels de la matrice (image) comme des paramètres caractéristiques. Cette méthode est très utilisée dans certaines approches neuronales. Mais ne tolère absolument pas les



transformations géométrique (rotation, translation, échelle) (figure 4.13)

Figure 4.13 : Présentation matricielle du chiffre "1"

4.4. La classification

À partir des caractéristiques que nous obtenons sur les chiffres en premier, une transmission à un classifieur est effectuée pour la classification réelle. i.e., détermine à quelle classe existante classifieur le chiffre.

L'objectif final de la RdF est la classification. Réaliser la classification d'une base d'observations (d'apprentissage), consiste à trouver les frontières des classes qui la compose. Le terme de classification regroupe en fait trois cas de figure qui impliquent des stratégies différentes : *supervisée*, *semi supervisée* ou *non supervisée*.

Dans le cas de l'apprentissage supervisé, les échantillons sont étiquetés pour indiquer la classe dans laquelle l'échantillon (la forme) soit rangé. Cette phase d'apprentissage consiste à analyser les

ressemblances entre les éléments d'une même classe et les dissemblances entre les éléments de classes différentes, cette phase nécessite un stockage dans une table d'apprentissage pour que la décision de classifieur se référera ensuite pour classer les nouvelles formes qui lui sont présentées.

Dans notre cas de la reconnaissance des chiffres, la classification est supervisée. Nous avons utilisé les classifieurs les plus utilisés et les plus performants dans le domaine de l'écriture à savoir le perceptron multicouche (PMC), machines à vecteurs de support (SVM), arbre de décision, Naïve Bayésien, KPPV. Nous avons même utilisé deux classifieurs faibles (pseudo inverse et analyse discriminante linéaire –LDA- (annexe 1)) pour voir d'où vient l'efficacité d'un système de RdF, de classifieur ou de paramètres caractéristiques.

4.4.1. Perceptron Multi Couche

Un PMC désigne un réseau connexionniste ayant une architecture particulière, une dynamique feedforward (sans circuit) et une règle d'apprentissage du type rétro propagation. Comme son nom l'indique, les unités de PMC sont réparties en couches (au moins deux), une unité ne recevant des signaux que des unités en amont (couche immédiatement précédente). On distingue les unités de la première couche (entrée), de la dernière couche (sortie) et des couches intermédiaires (cachée) [183].

Nous avons utilisé un PMC de 20 neurones (figure 4.14) et une couche cachée (après essai de différents nombres de neurones et couches) (tableau 4.4).

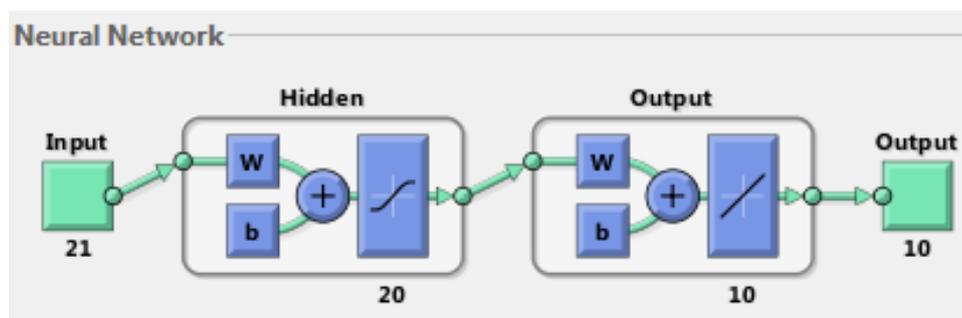


Figure 4.14 : PMC sous Matlab

4.4.2. Machine à vecteurs de support

Dans les SVM original, un problème de classification de n classes est converti en n problèmes à deux classes, et dans le i ème problème à deux classes nous déterminons la fonction de décision optimale qui sépare la classe i des classes restantes. Dans la classification, si l'une des n fonctions de décision classe un échantillon de données inconnues dans une classe définie, il est classé dans cette classe. Dans cette formulation, si plusieurs fonctions de décision classent un échantillon de

données dans des classes définies ou si aucune fonction de décision ne classe l'échantillon de données en classe définie, l'échantillon de données est inclassable [213].

Un autre problème des SVM est l'apprentissage lent. Ces derniers sont formés en résolvant les problèmes d'une programmation quadratique avec le nombre de variables égal au nombre de données d'apprentissage. L'apprentissage est lent pour un grand nombre de données d'apprentissage [213].

De nombreuses solutions ont été proposées pour étendre l'utilisation des SVM aux problèmes multi-classes.

- ✓ **La stratégie « un contre tous »** : l'acronyme connu sous OAA (One Against All ou one versus all), la solution la plus simple pour résoudre un problème multi-classes à l'aide de SVM consiste à le décomposer en un ensemble de sous-problèmes binaires et à construire indépendamment autant de SVM qu'il y a de classes et les entraîner pour séparer une classe du restant. Chaque SVM sépare une classe qui est étiquetées par (+1), de toutes les autres classes qui sont étiquetées par (-1). Chaque SVM est ainsi associé à une classe et sa sortie peut être considérée comme une mesure d'appartenance à la classe. La décision finale consiste donc à attribuer la donnée inconnue à la classe correspondant au SVM ayant la plus grande valeur de sortie.
- ✓ **La stratégie « un contre un »** : Elle consiste à entraîner les SVM afin d'obtenir toutes les frontières de décision séparant les classes une à une. Il en existe $K(K-1)/2$ classifieurs pour un problème à K classes. Cette stratégie classique consiste à construire un SVM pour chaque paire de classes. Pour un problème de n classes soit $n(n-1)/2$ SVM construit. Chaque classifieur est donc sépare les données d'une classe de celles d'une autre classe.

Notons que dans la littérature, cette stratégie peut aussi être connue sous le nom de « pairwise coupling », « all pairs », ou encore « round robin » [244]. La règle de décision utilisée est le vote majoritaire. Chaque SVM vote alors pour une classe et la donnée inconnue est finalement associée à la classe ayant reçu le plus grand nombre de votes.

4.4.3. K Plus Proche Voisin (KPPV)

KPPV est un classifieur non paramétrique très répandu. Le principe est basé sur le

- Choix d'un entier k (souvent $k = \text{nombre d'attributs} + 1$, ou $k = \sqrt{n}$).
- Calcul des distances (ex : distance Euclidienne, -la distance la plus populaire-).

- Choix des k observations pour lesquelles ces distances sont les plus petites (retenir les k plus proches voisins d'un cas de référence).
- Comptage de nombres de fois où ces k observations apparaissent dans chacune des classes (déterminer les classes correspondantes).
- Choix de la classe la plus représentée, celle ayant le plus grand nombre d'apparition.

Nous présentons les autres classifieurs (arbre de décision, pseudo inverse, LDA, Naïve Bayésien) utilisés dans l'annexe 3. Le tableau 4.4 suivant présente un sommaire des paramètres utilisés par les différents classifieurs.

Tableau 4.4 : Sommaire des paramètres des classifieurs

Classifieur	Fonction Matlab	Type et valeur de paramètre
KPPV	fitcknn	- Nombre de voisin K aléatoire - Distance : 'correlation', 'cosine', 'euclidean', 'seuclidean', 'cityblock', 'chebychev', 'hamming', 'jaccard', 'mahalanobis', 'minkowski', 'spearman',
Bayes	fitcnb	- Distribution : 'normal' : Gaussien, 'kernel', 'mn' : multinomial, 'mwmn' : Multivariate multinomial
Arbre de décision	fitctree	- Paramètre par défaut
LDA	fitcdiscr	- Type de Discrimination : 'linear', 'diaglinear', 'pseudolinear', 'quadratic', 'diagquadratic', 'pseudoquadratic'
SVM	fitcecoc	- codage : 'onevsone' 'onevsall'
PMC	feedforwardnet	- nombre de neurones dans la couche cachée : 20, - 'trainlm' - min_grad= 1e-017; - max_fail=770; - epochs = 500 (pour les chiffres imprimés) ; = 150 (pour les chiffre manuscrits)

4.5. Evaluation des Performances

Une qualité importante d'un classifieur est d'être capable de généraliser i.e. de pouvoir fonctionner correctement sur des données qu'il n'a pas apprises. Il est évident que ces données ne doivent pas être les mêmes que celles sur lesquelles l'apprentissage a été effectué. Le besoin de généralisation explique bien pourquoi le fait d'utiliser la même base est à éviter.

L'étape de test est une étape importante, elle permet d'évaluer la performance du classifieur avec les primitives choisies. Pour trouver les primitives pertinentes au problème posé, il faut tester la performance du système à chaque fois par les primitives jugées utiles au problème ou même de combiner les primitives extraites jusqu'à atteindre les performances souhaitées désirées.

Pour estimer la performance d'un classifieur, En effet plusieurs méthodes visées à optimiser l'utilisation de ces données à savoir la validation croisée, tout sauf un (leave one out), resubstitution, Holdout (annexe 3). Nous avons utilisé ces méthodes pour l'apprentissage sur la bases des chiffres imprimé.

La mesure globale classique, la plus utilisée pour évaluer la performance d'un classifieur, est le taux de reconnaissance (TR). Un taux de reconnaissance se calcule sur des décisions de type classe. On compte les bonnes décisions et les mauvaises décisions.

$$TR = \frac{\text{nombre de formes reconnues correctement}}{\text{nombre total de formes}} * 100 \quad (4.19)$$

D'autres mesures globales peuvent être utilisées comme le taux de rejet (TRJ) et le taux de confusion (ou d'erreur -TE-)

$$TE = \frac{\text{nombre de formes mal reconnues}}{\text{nombre total de formes}} * 100 \quad (4.20)$$

$$TRJ = \frac{\text{nombre de formes rejetées}}{\text{nombre total de formes}} * 100 \quad (4.21)$$

Tel que : TR+TE+TRJ=100%

Généralement, le taux de fiabilité (TF) correspond à la formule :

$$TF = \frac{TR}{100-TRJ} \quad (4.22)$$

La performance locale d'un classifieur quelconque peut être représentée par des mesures locales comme la matrice de confusion, qui nous permet de calculer la probabilité que ses éléments soient reconnus comme appartenant à cette classe ou à chacune des autres classes possibles.

4.6. Combinaison des classifieurs

Les recherches se concentrent actuellement sur la combinaison (séquentielle, parallèle ou hybride) de plusieurs systèmes de classifieurs, qui peuvent être construit selon des modèles identiques ou différents, et/ou des approches de création d'ensembles de données. Ces systèmes

fusionnent les informations des décisions de classification à différents niveaux en surmontant les limites des approches traditionnelles basées sur des classificateurs uniques et améliorent nettement les performances du système de reconnaissance par rapport à chacun des classificateurs pris isolément.

La combinaison parallèle de classificateurs a été proposée comme une voie de recherche permettant de fiabiliser la reconnaissance en utilisant la complémentarité qui peut exister entre les classificateurs. Sur ce point, nous avons vu dans les chapitres précédent 1 et 3 que la littérature abonde de travaux présentant une variété de schémas de combinaison qui diffèrent principalement par leur capacité d'apprentissage et le type de sortie des classificateurs. Toutefois, le choix de ces méthodes ne suit a priori aucune règle précise, et dépend étroitement de l'application que l'on veut traiter, de la façon dont on veut la traiter et des outils disponibles (bases de données, classificateurs et règles de combinaison).

Nous présentons maintenant les règles de combinaison que nous avons utilisées à savoir, la règle de vote à la pluralité, Borda count, le maximum, le minimum, la somme, le produit et la moyenne. Ces règles n'utilisent aucunes informations supplémentaires sur le comportement des classificateurs et aucun apprentissage.

4.6.1. Vote à la pluralité

Une classe est choisie comme une classe finale si elle reçoit le plus grand nombre de votes. C'est la méthode la plus simple, la plus facile à appliquer et la plus optimale. Si toutes les classes ont le même nombre de votes (conflit) il y a un rejet (équation 3.6). Donc il est préférable d'utiliser un nombre impair de classificateurs.

Une analyse de la performance de la combinaison des classificateurs par cette règle est discutée dans [33].

4.6.2. Borda count

Borda count est basé sur le calcul du rang total pour chacune des classes proposées, en calculant la somme des nombres de classes placées en dessous par chacun des classificateurs. C'est la méthode la plus convenable pour les problèmes muticlasses où la classe correcte peut apparaître dans les premières rangées mais quelque fois pas la première (top). Les rangements sont intéressants pour éviter le problème de variabilité entre les valeurs obtenues par les différents classificateurs, depuis la normalisation des sorties des classificateurs n'est pas triviale (confiance, probabilité postérieur).

Le tableau 4.5 présente le résultat de classification d'un nouvel objet en utilisant trois classifieurs (C1, C2, C3) et 4 classes (a, b, c, d). Tout simplement la méthode Borda count somme les rangées de chaque classe. Pour la classe a : $1+4+3=8$, la classe b : $3+3+4=10$, la classe c : $4+1+2=7$, la classe d : $2+2+1=5$. Donc la classe de cet objet est la « classe b »

Tableau 4.5 : Exemple de Borda count

Valeur du rang	C1	C2	C3
4	c	a	b
3	b	b	a
2	d	d	c
1	a	c	d

4.6.3. Les règles fixes

Le tableau 4.6 présente les décisions obtenues par 3 classifieurs (Y1, Y2, Y3) pour 4 classes (C1, C2, C3, C4), et en utilisant les règles fixe (maximum, minimum, somme, produit et moyenne).

- **Maximum**

Nous utilisons la règle Maximum pour sélectionner le classificateur produisant la confiance estimée la plus élevée à partir du type de mesure selon la formule 4.23 :

$$\text{Max}(x) = \text{Max}_{k=1}^c \text{Max}_{i=1}^n P(w_k | y_i(x)) \quad (4.23)$$

n : nombre de classifieurs, c : nombre de classes.

Cette règle trouve la valeur maximale pour chaque classe parmi les classifieurs et assigne l'objet à la classe qui a la valeur maximale.

$\text{Max}(x) = \text{Max} \{0.2, 0.5, 0.8, 0.1\} = 0.8$. Donc le résultat est la classe 3.

- **Minimum**

Trouve la valeur minimale de chaque classe parmi les classifieurs est assigne la forme à la classe qui a la valeur maximale.

$\text{Max}(x) = \text{Max} \{0.1, 0.0, 0.2, 0.1\} = 0.2$. Donc le résultat est la classe 3.

- **La somme**

Additionner les résultats des classifieurs et assigne la classe avec la valeur maximale à la forme.

$\text{Max}(x) = \text{Max} \{0.6, 0.8, 1.3, 0.2\} = 1.3$. Donc le résultat est la classe 3.

- **Le produit**

Multiplier les résultats des classifieurs et assigne la classe avec la valeur maximale à la forme.

$\text{Max}(x) = \text{Max} \{0.006, 0.0, 0.048, 0.00\} = 0.048$. Donc le résultat est la classe 3.

- **La moyenne**

Trouve la moyenne de chaque classe parmi les classifieurs et assigne l'objet à la classe avec la valeur maximale.

$\text{Max}(x) = \text{Max} \{0.17, 0.27, 0.43, 0.07\} = 0.43$. Donc le résultat est la classe 3.

Tableau 4.6 : Exemple montre les résultats de règles fixe

	C1	C2	C3	C4	Décision
Y1	0.1	0.5	0.2	0.1	C2
Y2	0.2	0.3	0.3	0.1	C2
Y3	0.2	0.0	0.8	0.0	C3
Min	0.1	0.0	0.2	0.0	C3
Max	0.2	0.5	0.8	0.1	C3
Moyenne	0.17	0.27	0.43	0.07	C3
Somme	0.6	0.8	1.3	0.2	C3
Produit	0.006	0.00	0.048	0.00	C3

4.7. Conclusion

Nous avons présenté dans ce chapitre les techniques les plus utilisées au niveau des paramètres caractéristiques et au niveau de classifieurs pour la réalisation de nos travaux. Comme vous avez pu le remarquer dans la section « 4.4 » plusieurs classifieurs interviennent. Donc pour la prise de la décision finale et pour améliorer leur performance nous avons utilisé sept méthodes de combinaison.

Le chapitre suivant présente les résultats de l'application et l'implémentation de tous ces techniques à travers les étapes du système proposé, avec analyse. L'architecture globale de notre système est illustrée dans la figure 4.15

Notre application est développée en Matlab R2016b. La plateforme est 64 bits, windows 7, le processeur est Intel Core Intel (R) Core (TM) i7-2670QM CPU @ 2.20GHz, avec une mémoire de 6 Go.

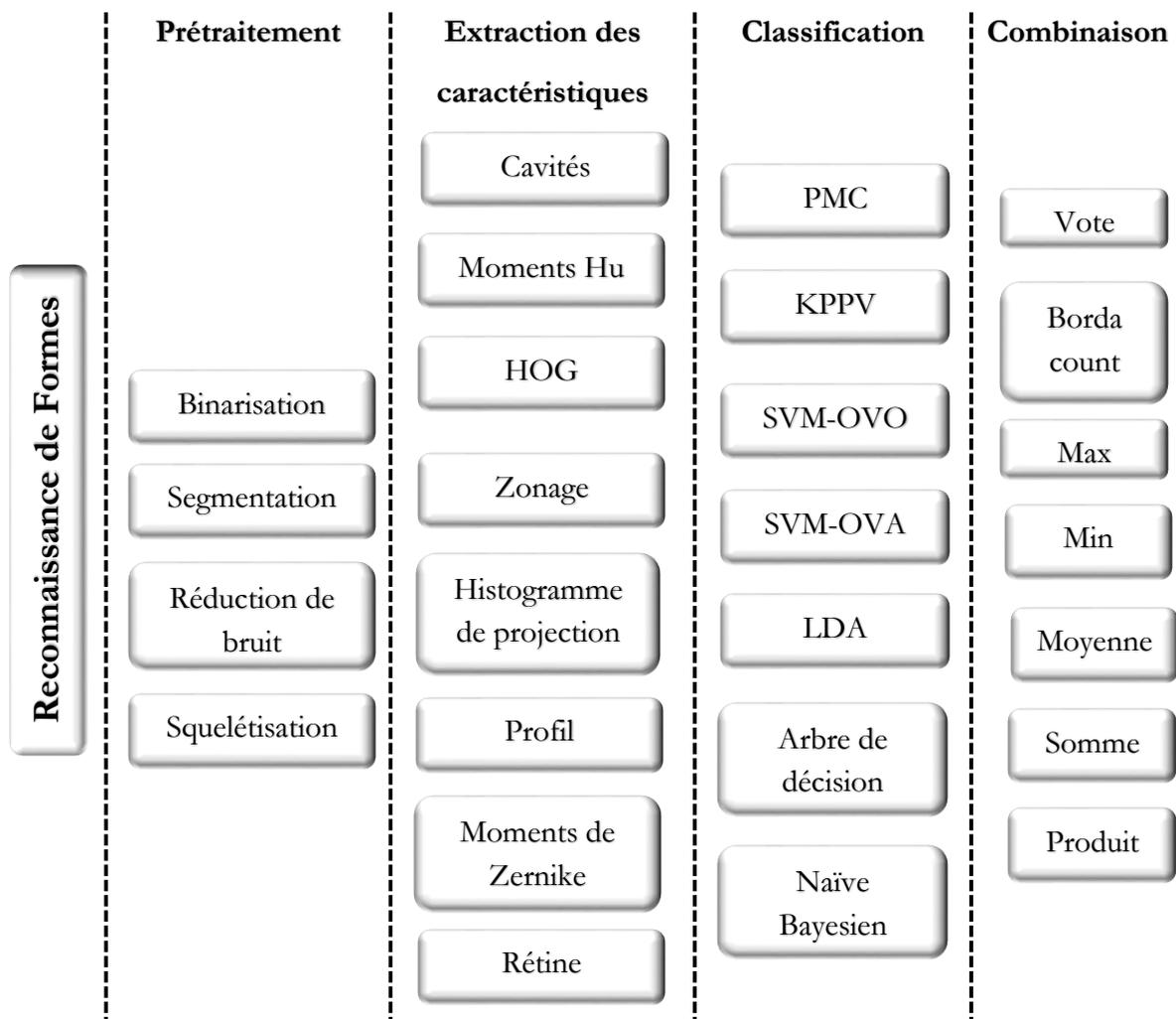


Figure 4.15 : Architecture proposée pour notre MCS

Chapitre 5 .

ANALYSE ET DISCUSSION DES RESULTATS EXPERIMENTAUX DE L'APPROCHE DEVELOPPEE

5.1. Introduction

5.2. Les chiffres imprimés

5.3. Les chiffres manuscrits

5.4. Conclusion

Ce chapitre décrit en détail les expériences entreprises au cours du développement de l'approche proposée.

Les résultats des expériences réalisées avec un ensemble de classifieurs individuels et différents paramètres caractéristiques sont présentés dans ce chapitre. Nous exposons ensuite la démarche suivie pour l'optimisation du système par la concaténation des paramètres caractéristiques et la combinaison des classifieurs en utilisant différents types de règles de combinaison. Nous illustrons les résultats des expériences réalisées avec une analyse et une comparaison.

5.1. Introduction

Nous présentons dans ce chapitre les résultats obtenus sur les trois bases des chiffres imprimés (BCI_1, BCI_2 et BCI_3) présentées dans le chapitre précédent et une base des chiffres manuscrits (MNIST) avec une approche MCS. Le point de départ de notre recherche a été la reconnaissance en utilisant des classifieurs individuels. Pour cette raison, nos premières contributions ont été faites sur l'extraction des paramètres caractéristiques, puis sur la classification.

Les résultats sont donnés au premier lieu avec les classifieurs individuel (KPPV, LDA, arbre de décision, Naïve Bayésien, MLP et SVM) avec les mêmes paramètres caractéristiques (rétine, cavité, profil, histogramme de projection, zonage, les moments Hu, les moments de Zernike et HOG, ...). Des comparaisons entre les performances de ces classifieurs utilisés de manière indépendante ont été réalisés. Au deuxième lieu, les résultats sont donnés en utilisant différents concaténation de paramètres caractéristiques avec les mêmes classifieurs, pour que les performances des classifieurs soient augmentées.

Finalement, la combinaison des classifieurs est utilisée selon sept règles de combinaison parallèle (vote à la pluralité, borda count, Max, Min, moyen, somme, produit). Nous avons aussi comparé le résultat final de notre système avec celui d'autres systèmes de la littérature ayant été testés eux aussi sur la base MNIST.

5.2. Les chiffres imprimés

Nous alimentons les différents classifieurs directement par l'image (rétine) et par les différents types de paramètres caractéristiques susceptibles de lever certaines ambiguïtés résiduelles et ne perdent pas (ou très peu) d'informations.

Des résultats expérimentaux sur la base BCI_1 présentés dans le tableau 5.1, provenant de 7 caractéristiques (5 cavités, nombre de cavité et ratio) en utilisant le classifieur KPPV, sont présentés dans [245]. L'architecture de ce travail est présentée dans la figure 5.1.

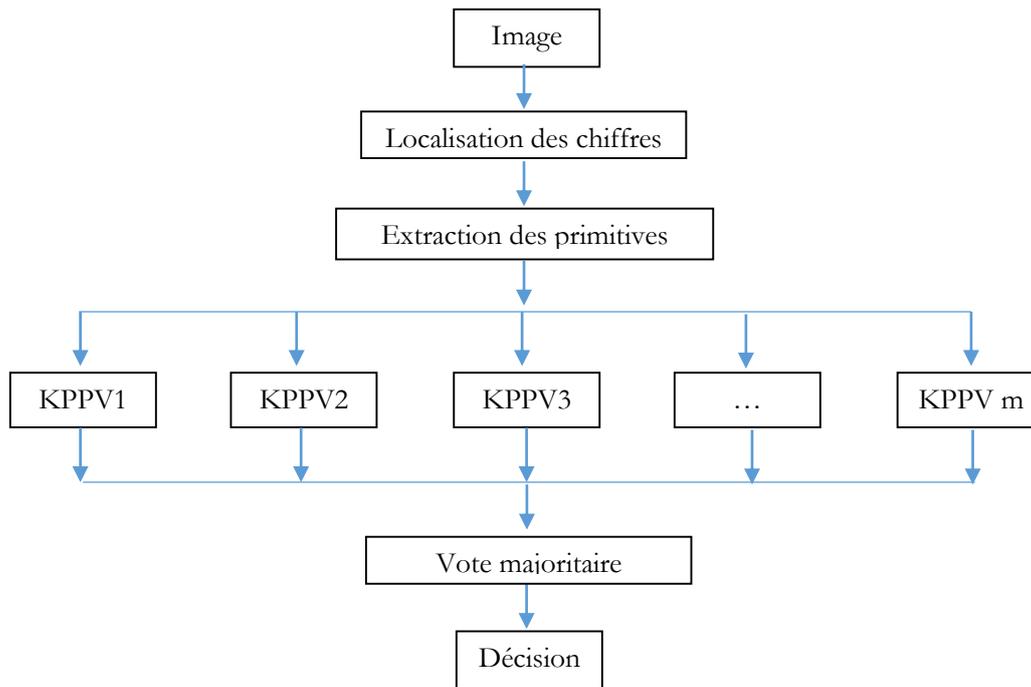


Figure 5.1 : Système multi-classifieurs KPPV

Comme le montre la figure, ce travail utilise le classifieur KPPV (KNN) en fonction de plusieurs distances (Euclidienne, Minkowsky, Manhattan, Chebychev, Mahalanobis, cosinus, corrélation, Euclidien normalisé) –Annexe 3- . Pour l'apprentissage deux techniques sont exploitées: la validation croisée (10-fold cross validation), et la méthode Holdout.

Tableau 5.1 : Performance de KPPV utilisant différentes distances

KPPV	Validation Croisée	Holdout
Cityblock	59.9317 %	60.8483 %
Chebychev	59.3530 %	60.6410 %
Correlation	57.6753 %	58.8124 %
Cosine	54.3425 %	55.1981 %
Euclidien	59.4000 %	61.2721 %
Mahalanobis	59.7066 %	61.7068 %
Minkawski	59.4000 %	61.2721 %
Seuclidien	59.6916 %	61.4874 %
Combinaison	62.0192	61.7521

Le meilleur résultat est obtenu par la distance "cityblock" dans la "validation croisée" et par la distance "Mahalanibis" pour la méthode "Holdout". Nous constatons que la validation croisée n'a pas un grand effet dans l'amélioration des performances. Cela confirme que cette méthode est

efficace pour les bases de données de petites dimensions. D'une autre coté la performance a augmentée à 62.02% par la règle de vote.

Selon les résultats qui ne dépassent pas 62%, nous pouvons dire que les cavités peuvent être insuffisantes pour discriminer les chiffres à cause de la confusion entre les chiffres: "0" "4" et "8", ils ont seulement une seule cavité de type central, "un" en italique "1" et "1" et "7" (cavité Est), "2" et "5" (cavité Est Ouest) comme illustré sur la figure 5.2.

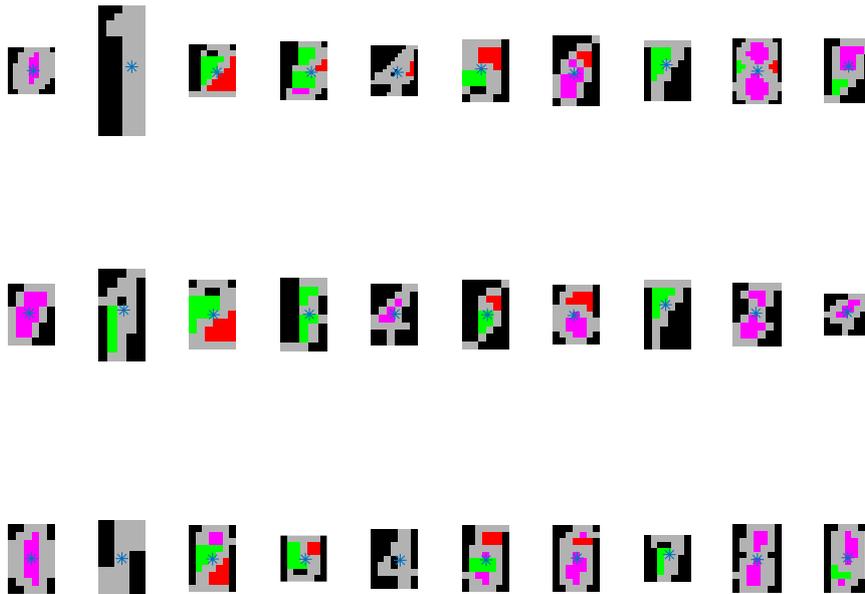


Figure 5.2 : Centre de gravité et similarité des cavités entre les chiffres

Pour éliminer cette confusion, nous avons proposé dans [246] d'extraire les cavités de différents sous bloques en utilisant le zonage. Le zonage utilisé est 2*1, pour éliminer quelques confusions comme la celle entre "0" et "8". Le bloque supérieur de zéro contient une cavité "Sud", et une cavité centrale pour le "8". De même pour les chiffres "2" et "5", une cavité "Est" pour le chiffre "5" dans le bloc supérieur et pour le chiffre "2" dans le bloc inférieur (figure 5.3).

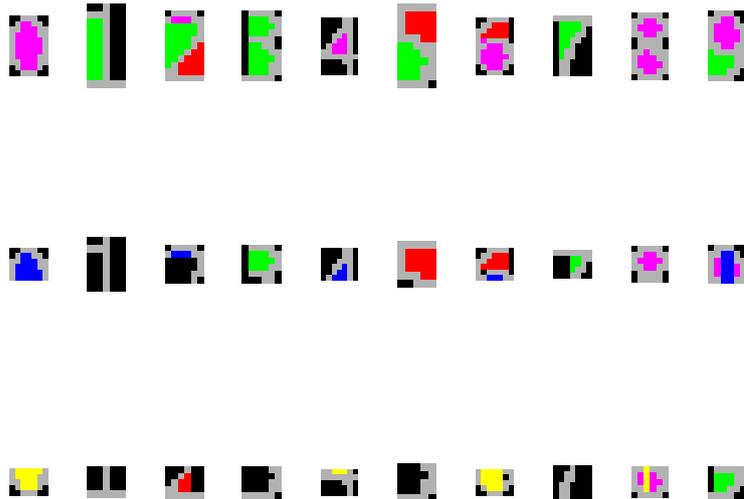


Figure 5.3 : Cavités de sous blocs

Pour voir la performance de cette méthode nous avons utilisé l'architecture proposée dans la figure 5.4:

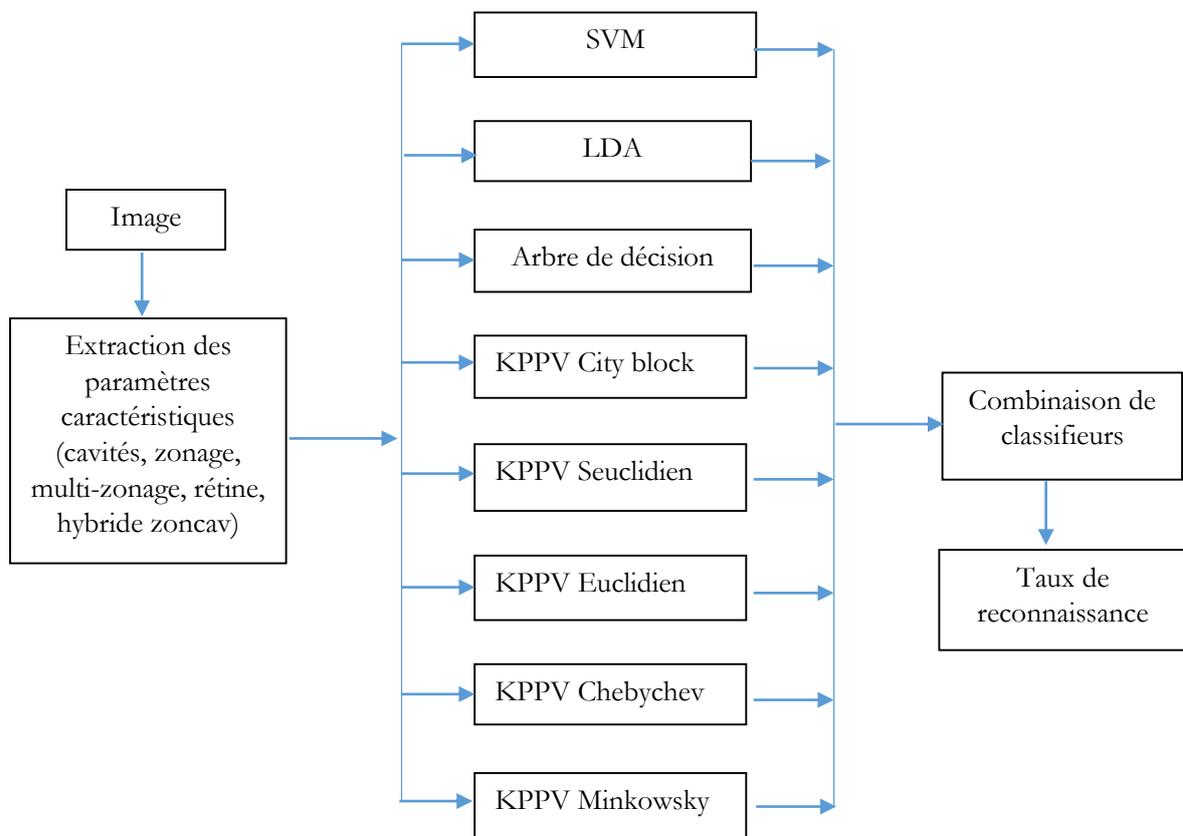


Figure 5.4 : Système multi-classifieurs

Aussi, l'idée proposée est de faire une hybridation entre deux types de paramètres caractéristiques, le zonage et les cavités. Les résultats obtenus sur la base BCI_1, sont présentés dans le tableau 5.2. Nous appelons ce type de paramètres par « hybride zoncav ».

Tableau 5.2 : Performance par la hybridation Zonage-Cavité

Classifieur	Validation Croisée	Holdout
LDA	70.32 %	71.05 %
Arbre deDecision	80.56 %	79.38 %
SVM	93.75 %	93.79 %
KPPV cityblock	80.45 %	81.62 %
KPPV Seuclidien	83.81 %	83.01 %
KPPV Euclidien	81.41 %	80.34 %
KPPV Chebychev	80.13 %	78.21 %
KPPV Minkawski	81.41 %	80.34 %
Combinaison	84.78 %	82.16 %

L'augmentation de la performance est très claire. Nous constatons dans ce tableau que la performance par le classifieur KPPV cityblock a augmenté de 59.93% à 83.81% pour la validation croisée. La performance avec ce type de paramètre arrive jusqu'à 93.79% par le classifieur SVM. Mais la combinaison de classifieur n'a pas amélioré la performance à cause de l'existence d'un classifieur dominant.

Pour vérifier l'efficacité de ce type de paramètres caractéristiques, nous avons utilisé un autre classifieur, le PMC, avec d'autres paramètres pour voir la supériorité de notre vecteur de paramètres proposé, selon la figure 5.5 [247]

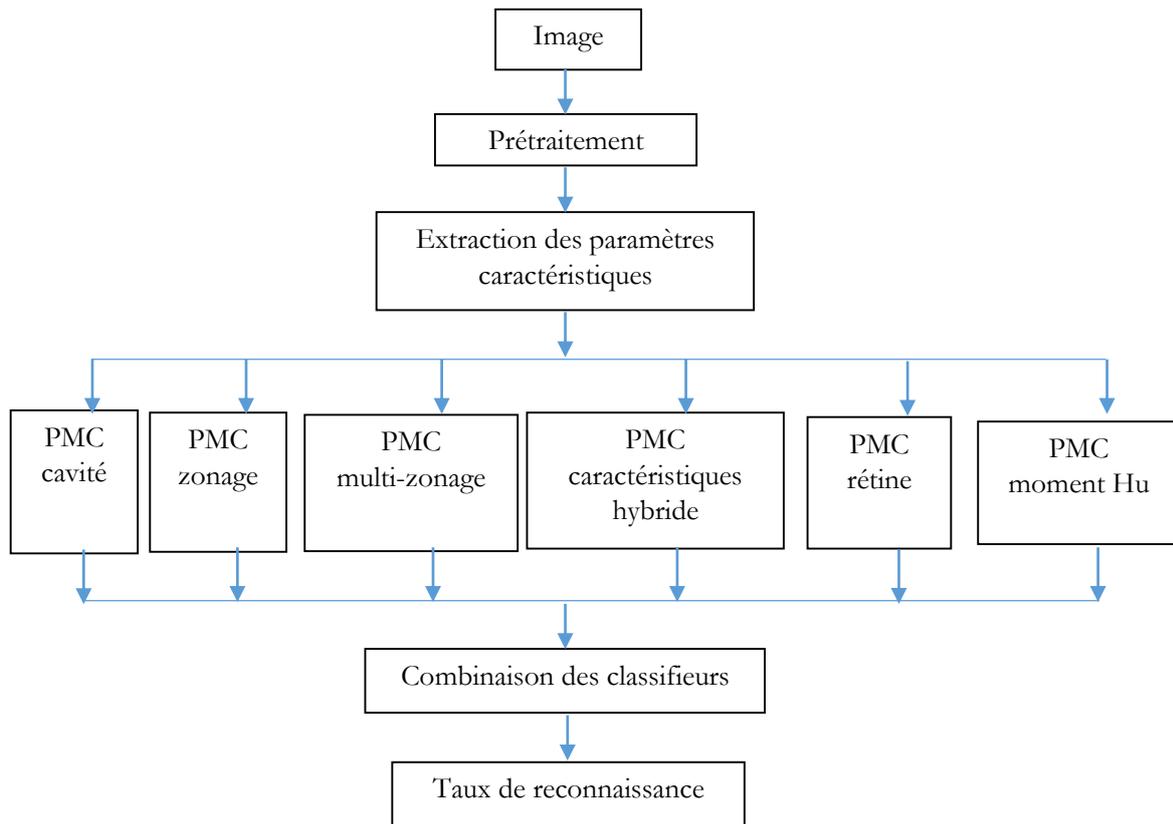


Figure 5.5 : Système multi-classifieurs PMC

Les résultats obtenus sont présentés dans le tableau 5.3. La performance trouvée par le classifieur PMC en utilisant ‘hybride zoncav’ est de 93.75% le même trouvée par le classifieur SVM.

Tableau 5.3 : Performance du classifieur PMC

	Validation croisée	Holdout	Resubstitution
Cavité	66.38 %	68.27 %	67.52 %
Moment Hu	53.46 %	54.06 %	58.45 %
Retine	88.83 %	88.14 %	92.85 %
Zonage	29.65 %	28.10 %	32.18 %
Multizonage	82.90 %	82.08 %	84.10 %
Hybride zoncav	93.75 %	93.38 %	94.44 %

Pour améliorer la performance, nous avons fait des concaténations entre ces différents paramètres caractéristiques. Les meilleures concaténations sont présentées dans le tableau 5.4.

Tableau 5.4 : Performance du PMC par concaténation de paramètres caractéristiques

	Validation roisée	Holdout	Resubstitution
Cavité+ moment Hu	86.30 %	79.81 %	80.93 %
Cavité+ hybrid	79.62 %	87.50 %	87.79 %
Hybrid+ moment Hu	87.37 %	85.68 %	89.10 %
Cavité+hybrid+ moment Hu	90.34 %	91.24 %	93.16 %
Cavité+hybrid+ moment Hu +multizonage	91.73 %	90.71 %	94.33 %
Cavité+hybrid+ moment Hu +multizoning+retine+zonage	93.70 %	93.48 %	98.88 %

La concaténation a donné une amélioration remarquable de la performance soit 98.88% spécialement pour les moment Hu qui nous donne l'avantage de l'invariance aux transformations géométriques. En se basant sur cette propriété de l'invariance aux transformations géométriques, nous avons proposé l'architecture illustrée sur la figure 5.6 [248]

Les résultats obtenus par la concaténation des moments Hu, la cavité Est, nombre de hole, la surface de chiffres et le ratio sont présentés dans le tableau 5.5. Le meilleur résultat est donné par le classifieur KPPV soit 99.89%.

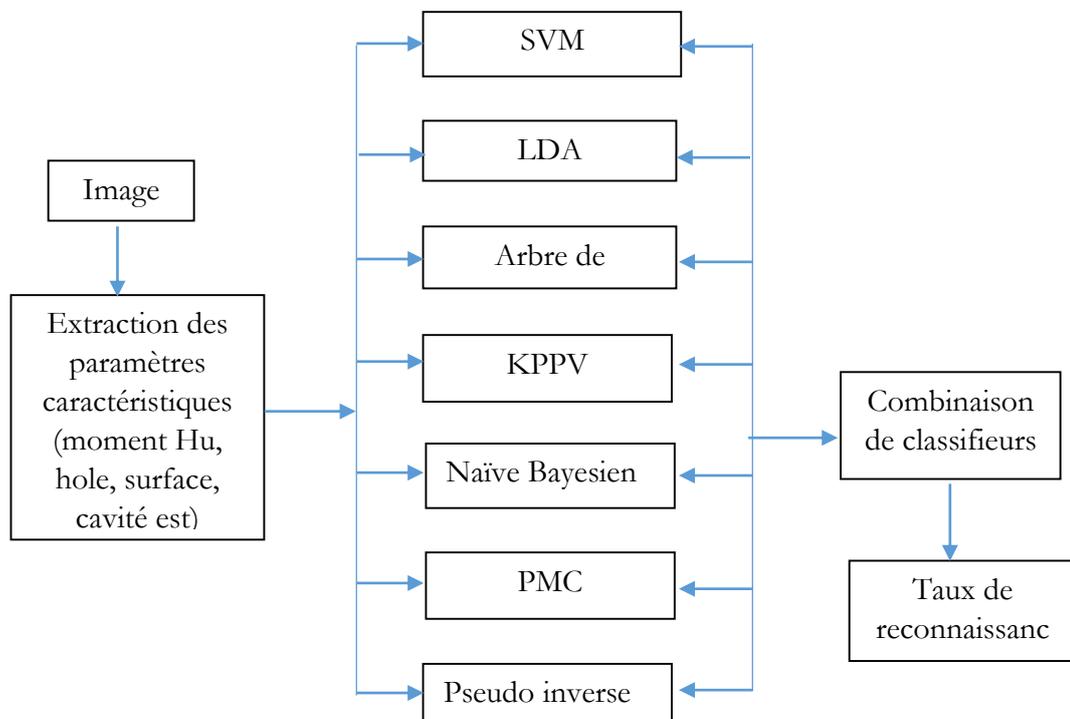


Figure 5.6 : Système multi-classifieurs

Tableau 5.5 : Performance par Hu moment nombre de hole cavité Est surface et ratio

Classifiers	Validation croisée		Holdout		Resubstitution
	Non stratifié	stratifié	Non stratifié	stratifié	
Pseudo-inverse	55.50 %	52.95 %	55.24 %	54.38 %	53.08 %
LDA	52.84 %	50.99 %	54.81 %	49.25 %	54.38 %
N_Bayes	54.96 %	54.66 %	54.91 %	54.70 %	63.78 %
Arbre de Decision	75.93 %	69.46 %	74.15 %	66.88 %	91.14 %
SVM	62.08 %	62.02 %	60.04 %	63.78 %	67.11 %
KPPV	79.97 %	79.65 %	73.93 %	77.35 %	99.89 %
PMC	73.03 %	71.96 %	70.51 %	72.12 %	72.84 %

Nous avons utilisé une seule base d'apprentissage et test en utilisant trois méthodes de la validation croisée. Des résultats satisfaisants sont obtenus. Pour illustrer de mieux nos résultats, nous avons utilisé une autre base de test "BCI_2" (chapitre 4).

Nous commençons par proposer une nouvelle amélioration dans l'extraction des cavités pour lever l'ambiguïté. Elle consiste à une extraction récursive des cavités. Nous répétons l'extraction des cavités pour quatre parties de l'image supérieure, inférieure, droite et gauche. Nous obtenons les quatre parties en divisant l'image dans le point de centre de gravité. Puis enrichir ces paramètres par :

- Le nombre de Hole : il y a une différence entre la cavité centrale et hole (figure 5.7)
- Les points de croisement.
- Le nombre de points finaux.
- Le rapport entre la surface de la partie supérieur et inférieur du chiffre.
- La surface du chiffre par rapport au surface totale (surface de l'ensemble des cavités et de chiffre).

Nous avons obtenu un vecteur qui contient 36 paramètres que nous avons baptisé « cavité récursive ».

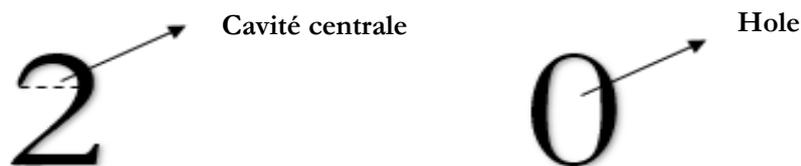


Figure 5.7 : Différence entre hole et cavité centrale

Nous alimentons tous les classifieurs par les paramètres mentionnés dans le chapitre 4 et les deux nouveaux paramètres proposés, sur une nouvelle base de test BCI_2. Le tableau 5.6 présente le nombre de variable dans chaque type de paramètre caractéristique (dimension du vecteur).

Tableau 5.6 : Description et dimension des vecteurs de caractéristiques

	Description	Dimension
Rétine	Nombre de pixels (nombre de ligne * nombre de pixel)	40 (8*5)
Cavité récursive	-	36 (5+5+5+5+5+1+1+1+1+1+1+1+4)
Profil	Profil droite gauche haut et bas	26 (8+8+5+5)
Hybride zoncav	-	63 (5+4+2+2+24+12+12+1+1)
Histogramme	Nombre de pixel pour chaque ligne et colonne	13 (8+5)
Moment de Hu	7 moment de Hu 5 moment invariants étendus	12 (7+5)
Moment de Zernike	Les moments Zernike jusqu'à l'ordre 12	49
HOG	-	108

Les résultats obtenus par le classifieur KPPV sont présentés dans le tableau 5.7, par le classifieur Naïve Bayes dans le tableau 5.8, arbre de décision dans le tableau 5.9, LDA dans le tableau 5.10, SVM dans le tableau 5.11 et MLP dans le tableau 5.12. TR représente la performance sur la base d'apprentissage BCI_1 et TS la performance sur la base de test BCI_2.

Tableau 5.7 : Performance de KPPV

	Distance	TR %	TS %
Rétine	Correlation	99.9198	94.1622
Cavité récursive	Cityblock	99.7298	85.0811
Profil	Cityblock	94.1186	89.5135
Hybride zoncav	Cosine	99.8311	86.4865
Histogramme	Correlation	93.9904	76.7387
Moment de Hu	Mahalanobis	99.8077	51.1532
Moment de Zernike	Mahalanobis	99.8077	65.7838
HOG	Cityblock	99.6795	69.7838

Nous avons utilisé plusieurs valeurs de K (le nombre de voisin). Les meilleures performances sont données par la valeur 1 (le nombre optimal de voisins les plus proches). Nous constatons qu'après la rétine, le profil donne des bons résultats. Vient ensuite les deux méthodes proposées, dans le troisième et quatrième rang (Cavité récursive, Hybride zoncav).

Tableau 5.8 : Performance de Naïve Bayes

	TR %	TS %
Rétine	88.7981	50.0541
Cavité récursive	83.7500	77.1712
Profil	78.7019	84.8649
Hybride zoncav	89.6955	70.8468
Histogramme	68.5417	16.5766
Moment de Hu	79.5994	18.8468
Moment de Zernike	40.9295	13.6577
HOG	93.5096	46.2162

Le classifieur Naïve Bayes utilise plusieurs distributions: Multi-nomial, Multi-variate multinomial, kernel, Normal (Gaussian). Les meilleurs résultats obtenus sont ceux de la distribution "Kernel". Nous voyons que notre méthode proposé donne des bons résultats après le profil.

Tableau 5.9 : Performance de l'arbre de décision

	TR %	TS %
Rétine	95.9615	77.6216
Cavité récursive	94.1506	82.9910
Profil	89.6635	85.0991
Hybride zoncav	91.5224	85.2793
Histogramme	87.6763	15.3153
Moment de Hu	89.2628	51.9279
Moment de Zernike	90.4487	50.8649
HOG	92.7564	53.0090

Après la rétine, le profil donne les meilleurs résultats. Vient ensuite les deux méthodes proposées dans le troisième et quatrième rang.

Tableau 5.10 : Performance de LDA

	TR %	TS %
Rétine	90.8974	92.0360
Cavité récursive	79.0705	79.7477
Profil	73.2051	83.8198
Hybride zoncav	81.2198	85.5495
Histogramme	59.6314	41.0991
Moment de Hu	54.7436	37.0270
Moment de Zernike	64.9679	42.9189
HOG	85.7212	79.4234

Après la rétine, la méthode ‘Hybride zoncav’ donne des bons résultats.

Tableau 5.11 : Performance de SVM

	Codage	TR %	TS %
Rétine	Un contre tous	99.9199	94.3063
Cavité récursive	Un contre un	89.4551	88.6126
Profil	Un contre un	95.7212	89.6757
Hybride zoncav	Un contre un	94.1346	88.8829
Histogramme	Un contre un	95.6891	37.4955
Moment de Hu	Un contre un	86.6506	63.6036
Moment de Zernike	Un contre un	94.9519	72.7387
HOG	Un contre un	99.6795	81.4775

Après la rétine, le profil donne des bons résultats, puis vient les deux méthodes proposées, dans le troisième et le quatrième rang.

Tableau 5.12 : Performance de PMC

	TR %	TS %
Rétine	94.0385	89.3153
Cavité récursive	86.7468	86.7027
Profil	81.1058	87.7477
hybride zoncav	87.1154	87.5676
Histogramme	69.3109	17.9279
Moment de Hu	68.6218	55.8378
Moment de Zernike	74.0064	57.5856
HOG	89.7115	70.5225

D'après ces résultats la meilleure performance pour chaque type de paramètre caractéristique est obtenue par le classifieur SVM sauf pour l'histogramme le meilleur résultat est obtenu par le classifieur KPPV. Nous pouvons donc extraire l'ordre des classifieur (tableau 5.13).

Tableau 5.13 : Ordre des classifieurs pour chaque type de paramètre caractéristique

	Ordre des classifieurs
Rétine	SVM, KPPV, LDA, PMC, Arbre de Décision, NBayes
Cavité récursive	SVM, PMC, KPPV, Arbre de Décision, LDA, NBayes
Profil	SVM, KPPV, PMC, Arbre de Décision, NBayes, LDA
Hybride zoncav	SVM, PMC, KPPV, LDA, Arbre de Décision, NBayes
Histogramme	KPPV, LDA, SVM, PMC, NBayes, Arbre de Décision
Moment de Hu	SVM, KPPV, PMC, Arbre de Décision, NBayes, LDA
Moment de Zernike	SVM, KPPV, PMC, LDA, Arbre de Décision, NBayes
HOG	SVM, PMC, KPPV, LDA, Arbre de Décision, NBayes

Nous constatons que l'ordre selon les paramètres caractéristiques n'est pas le même pour tous les classifieurs. La rétine, le profil donne de bons résultats, puis vient les deux méthode proposées (Cavité récursive, hybride zoncav) dans le deuxième et le troisième rang. Le tableau 5.14 présente l'ordre des paramètres caractéristique selon le classifieurs. Donc les résultats ont montré que les deux méthodes développées et le profil sont qualifiés de classifieur les chiffres imprimés.

Tableau 5.14 : Ordre des paramètres caractéristiques pour chaque classifieur

KPPV	NBayes	Arbre de décision	LDA	SVM	MLP
Rétine	Profil	hybride zoncav	Rétine	Rétine	Rétine
profil	Cavité récursive	Profil	hybride zoncav	profil	profil
hybride zoncav	hybride zoncav	Cavité récursive	profil	hybride zoncav	hybride zoncav
cavité récursive	rétine	rétine	HOG	Cavité récursive	Cavité récursive
Histogramme	HOG	HOG	MZer	HOG	HOG
HOG	MHu	MHu	Histogramme	MZer	MZer
MZer	Histogramme	MZer	MHu	MHu	MHu
MHu	MZer	Histogramme	Cavité récursive	Histogramme	Histogramme

MHu: moment de Hu, MZer: moment de Zernike.

A part la rétine et le profil qui donnent de bon résultats, la confusion survient entre les classes de "2", "3" et "8", en effet la partie supérieure du nombre de "2" est similaire à "3" et "8". De même une similarité entre "1" et "7". Par ailleurs la partie médiane de "2" est similaire à "7" et "8". Pour couvrir ce genre de similarité, nous proposons d'ajouter au profil d'autres paramètres qui couvrent cette similarité comme les cavités et le zonage simple, cela aide à séparer entre "2", "3" et "8".

Nous avons essayé d'améliorer la performance en première approche par la concaténation des paramètres caractéristiques. Nous avons utilisé plusieurs concaténations. Les meilleurs résultats (tableau 5.15) sont obtenus par les concaténations suivantes :

- Concat1 : rétine, profil droite, profil gauche, point final et point de croisement
($8*5+8+8+1+1=58$)
- Concat2 : moments de Zernike (25 moments), nombre de holes, profil droit, profil gauche, rétine ($25+1+8+8+8*5=82$)

Tableau 5.15 : Différentes concaténations des paramètres caractéristiques

	Concat1		Concat2	
	TR	TS	TR	TS
KPPV	99.98	93.71	100.00	92.29
N Bayes	92.80	66.61	88.59	78.39
Arbre de D	96.97	82.27	96.78	81.27
LDA	92.96	93.84	94.09	93.46
SVM	99.98	95.51	100.00	95.19
PMC	95.64	93.36	96.25	90.61

Nous constatons que la première concaténation a donné presque les mêmes performances de la rétine. Par contre la deuxième a donné une amélioration remarquable (de 94.16% et 65.78% à 95.19%) en plus elle conserve la propriété de l'invariance aux transformations géométriques.

L'idée développée a abouti à des résultats satisfaisants. En effet, notre idée s'appuie sur la complémentarité entre les paramètres caractéristiques. Celui-ci pourrait donner plus de performance en exploitant la complémentarité entre les classifieurs.

Pour la combinaison des classifieurs, nous avons développé un programme (à l'aide de la fonction matlab « combnk » pour générer toutes les possibilités) qui fait l'intégration des classifieurs pour toutes les combinaisons possibles C_n^m (n=7, m de 2 classifieurs jusqu'au 7 classifieurs). Donc :

$$\text{Nombre de possibilités} = C_7^2 + C_7^3 + C_7^4 + C_7^5 + C_7^6 + C_7^7$$

Le tableau 5.16 présente les résultats obtenus pour chaque type de paramètres caractéristiques en spécifiant la meilleure performance donnée par un classifieur individuel et la meilleure performance obtenue parmi toutes les possibilités de combinaison de classifieurs (c'est une possibilité parmi le nombre des possibilités de combinaisons).

Tableau 5.16 : Combinaison de classifieurs

Paramètre Caractéristique	Meilleur Classifieur/TR	MCS/TR	Règle de combinaison
Rétine	OVO 94.3063	KPPV, LDA, OVO, OVA, PMC 94.9369	Vote à la pluralité
Cavité récursive	OVO 88.6126	Arbre de décision, LDA, OVO, OVA, PMC 90.5225	Bordacount
Hybride zoncav	OVO 88.8829	KPPV, Arbre de décision, OVO, OVA, PMC 90.5405	Vote à la pluralité
Histogramme	KPPV 76.7387	KPPV, Bayes 76.7387	Règle de maximum
Profile	OVO 89.6757	KPPV, Arbre de décision, OVA, PMC 91.6216	Vote à la pluralité
Moment Zernike	OVO 72.7387	KPPV, Arbre de décision, OVO, OVA 73.6036	Vote à la pluralité
Moment Hu	OVO 72.7387	KPPV, Arbre de décision, OVO, OVA 73.6036	Règle de maximum
HOG	OVO 81.4775	LDA, OVO, OVA 81.7838	Vote à la pluralité
Concat1	OVO 95.5135	KPPV, LDA, OVO, OVA, PMC 95.9099	Vote à la pluralité
Concat2	OVO 95.1892	KPPV, LDA, OVO, OVA, PMC 95.5315	Vote à la pluralité

OVO: one versus one, OVA: one versus all.

Nous constatons que la combinaison de classifieur en utilisant le même type de paramètres caractéristiques a donné une amélioration remarquable mais pas pour tous les types de paramètres caractéristiques, seulement pour "cavité récursive, hybride zoncav, profil, zernike, Hu" (particulièrement pour les deux méthodes proposées) avec la règle de « vote à la pluralité et bordacount ».

Maintenant nous présentons les résultats obtenus par la combinaison des classifieurs de même type alimenté par différents types de paramètres caractéristiques (tableau 5.17). Pour générer toutes les combinaisons possibles de paramètres caractéristiques, nous avons appliqué le même principe utilisé pour générer les combinaison d'intégration des classifieurs.

$$\text{Nombre de possibilités} = C_8^2 + C_8^3 + C_8^4 + C_8^5 + C_8^6 + C_8^7 + C_8^8$$

Tableau 5.17 : Combinaison de classifieurs de même type avec différents paramètres caractéristiques

Classifieur	Meilleur Caractéristique /TR	Différents paramètres caractéristiques/TR	Règle de combinaison
KPPV	Rétine 94.1622	Rétine, Histogramme, MZer, HOG 95.6396	Vote à la pluralité
NBayes	Profil 84.8649	Hybride, hybride zoncav, Profil 88.7568	Règle de maximum
Arbre de Décision	Hybride zoncav 85.2793	Rétine, cavité récursive, hybride zoncav, profil, HOG 92.1441	Vote à la pluralité
LDA	Rétine 92.0360	Rétine, hybride zoncav, histogramme, MZer 93.6577	Règle de maximum
SVM	Rétine 94.3063	Rétine, cavité récursive, hybride zoncav, histogramme, profil, MZer, HOG 96.3063	Bordacount
PMC	Rétine 89.3153	Rétine, cavité récursive, hybride zoncav, profil 95.0811	Bordacount

Nous constatons que la combinaison de classifieurs de même type alimenté par différents paramètres caractéristique fait une amélioration très remarquable de la performance, particulièrement pour le classifieur PMC (de 89.31% à 95.08%) et Arbre de décision (de 85.28% à 92.14%).

Pour la combinaison de différents classifieurs avec différents paramètres caractéristiques, nous avons trouvé une difficulté pour générer toutes les possibilités automatiquement. Cependant, nous avons utilisé l'approche statique pour sélectionner les classifieurs, elle consiste à choisir une fois pour toutes le *meilleur* classifieur ou le *meilleur* sous-ensemble de classifieurs, parmi un grand nombre de classifieurs disponibles. Selon les résultats précédents nous avons choisi les meilleurs classifieurs (SVM, KPPV et MLP) avec les meilleurs paramètres caractéristiques pour générer quelques combinaisons (tableau 5.18). Nous alimentons le SVM par les Mzer (moment de Zernike) pour augmenter l'indépendance aux transformations géométriques.

Tableau 5.18 : Quelques combinaisons de différents classifieurs avec différents paramètres caractéristiques

Combinaison de Classifieurs	Vote	Borda count	Max	Min	Som	Prod	Moy
-SVM MZer -PMC Cavité récursive -Arbre de décision Hybride zoncav -KPPV profile	90.46	92.09	90.56	88.59	90.34	73.13	90.34
-PMC MZer -KPPV profile	83.42	84.77	89.51	89.51	89.51	72.74	89.51
-SVM MZer -PMC profile	82.83	85.09	87.75	87.76	87.75	72.67	87.74
-KPPV Cavité récursive -SVM Hybride zoncav	86.36	85.80	85.08	85.08	85.08	85.15	85.08
-SVM Cavité récursive -PMC Hybride zoncav	87.75	88.41	87.57	87.65	87.57	87.26	87.56
-KPPV Cavité récursive -PMC Hybride zoncav	85.69	86.05	86.25	84.84	86.56	86.09	86.55
-PMC Cavité récursive -KPPV profil	86.03	85.58	90.56	89.17	90.43	89.71	90.43
-KPPV Cavité récursive -SVM profil	84.83	87.81	88.12	88.12	88.12	86.07	88.12
-SVM MZer -KPPV profile -MLP Hybride zoncav	92.00	91.29	90.92	89.20	89.13	73.13	89.13
-SVM MZer -KPPV rétine	86.34	88.19	94.28	94.28	94.28	72.74	94.28
-SVM rétine -PMC MZer	83.78	86.88	89.31	89.29	89.31	72.57	89.31
-SVM profile -KPPV rétine	92.14	93.28	94.11	94.10	94.10	89.67	94.11

Nous remarquons que, la plupart des améliorations sont obtenus par le vote à la pluralité, Bordacount, et la règle de maximum. Parmi ces possibilités, la meilleure combinaison est celle de SVM MZer+KPPV rétime, elle a amélioré la performance de 72.74% pour MZer et 94.16% pour la retine jusqu'à 94.28%. par quatre règles de combinaisons (max, min som, moy). Par contre à la lumière des données présentées ci-dessus, la combinaison des deux classifieurs SVM, MLP n'améliore pas la performance donc ne sont pas complémentaires.

Finalement, selon ces résultats nous avons démontré que l'amélioration de la performance vient en premier lieu par la concaténation des paramètres caractéristiques (cela signifie que les performances d'un classifieur peuvent s'appuyer autant sur les paramètres caractéristiques que sur le classifieur lui-même), puis la combinaison des classifieurs de même type par différents types de paramètres caractéristiques et la combinaison de différents classifieurs par différents types de paramètres caractéristiques. Concernant les règles de combinaisons, la plupart des améliorations sont obtenue par la règle de vote à la pluralité.

La comparaison des performances est difficile car les bases de données (apprentissage et test) sont différentes. De plus la majorité des auteurs élaborent leur système sur la même base de données que celle qui sert à l'évaluation. Donc Les performances annoncées peuvent approcher à 100%, mais il est très difficile de prévoir ce que seraient ces performances sur de nouvelles données.

Nous comparons ces résultats aux résultats obtenus dans [77] qui présente un travail très proche et similaire à cette étude. L'auteur a utilisé 10 multi fonts présentés dans la figure 5.8 (avant-garde, classic, helvetica, bookman, courier, halvetica, narrow, moderne, palatino, times, new century school book) multi taille (12, 14, 18, 24) et multi style (normal, gras, italique, gras italique). Avec les paramètres caractéristiques (point de croisement, croisement verticale, point final, zonage et hole), en utilisant un arbre de décision sur une base de 1560 images. La performance donnée est plus de 97.2% sur une base de test et 100% sur 1560 images. Nous avons achevé un taux de reconnaissance de 99.98% pour la base d'apprentissage et 96.30% pour la base de test. Ces résultats sont très satisfaisants par rapport aux différences dans les fonts utilisés et les tailles.

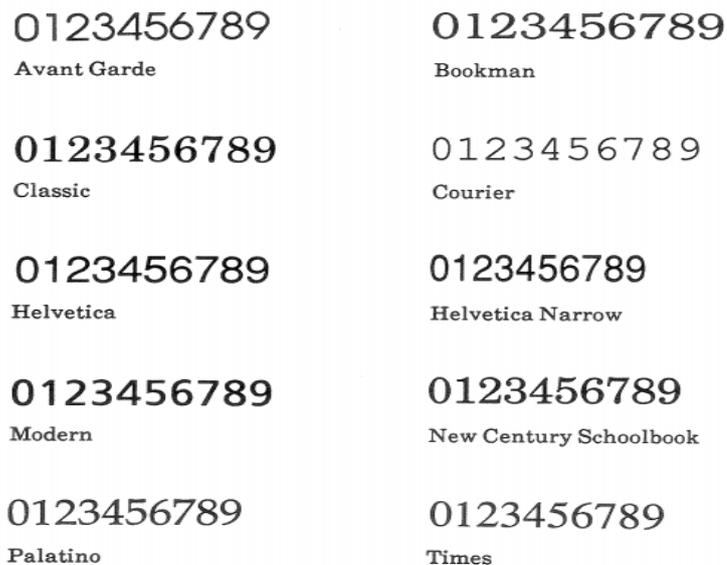


Figure 5.8 : Chiffres utilisés dans [77]

En se basant sur l'analyse des résultats des tableaux précédents, les raisons de la mauvaise classification peuvent être ainsi résumées :

- Confusion naturelle entre les paires des chiffres à cause de la structure et la forme similaire totalement ou partiellement, par exemple : chiffres "1-7", la partie supérieure de "2-3", la partie droite "3-8", la partie supérieur et inférieur de "0-8".
- Utilisation de la petite taille (5 et 8) et le style gras. Les chiffres se transforment en point noir comme le chiffre "0", "8". Certains font déformer le chiffre comme "2" qui prend la forme de "1 8", "5" prend la forme de "0 6", "6" prend la forme de "8" et la forme d'un bloque noir etc... (figure 5.9). Pour voir certains cas similaires, voir l'annexe 3.
- Segmentation ou binarisation qui a diminué la qualité de l'image (perte d'information) par exemple un "5" dont la barre supérieure est déconnectée du reste du chiffre (cassé) (figure 5.10).

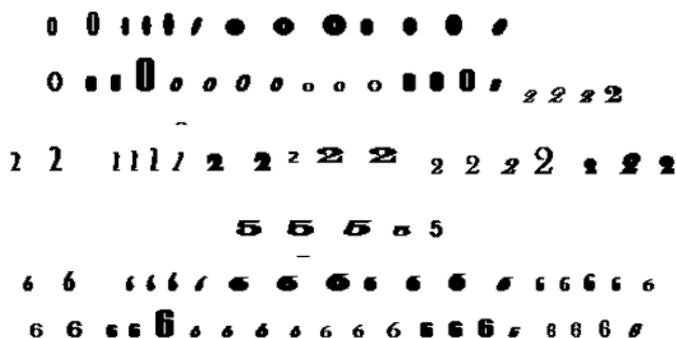


Figure 5.9 : Quelques chiffre déformés



Figure 5.10 : Chiffre "5" déconnecté

Nous utilisons maintenant une autre base de test (base BCI_3) pour démontrer l'efficacité de notre méthode « tableau 5.19 ».

Tableau 5.19 : Performance des classifieurs sur la base BCI_3

	KPPV	NBayes	Arbre de décision	LDA	SVM	PMC
Rétine	100.00	30.00	100.00	99.31	100.00	99.90
Cavité récursive	100	90	100	100	100	100
Profil	100	99.90	100.00	91.48	100	93.76
Hybride zoncav	86.33	87.92	80.10	90.00	100	90
Histogramme	99.40	38.71	21.58	90.19	74.95	36.63
Moment de Hu	79.21	28.32	68.91	78.20	90.99	91.09
Moment de Zernike	94.75	10.00	64.75	84.35	88.61	99.01
HOG	76.24	63.47	86.53	92.57	86.24	79.90

Ce tableau montre que les paramètres développés particulièrement « Cavité récursive » acheminé des performances supérieures sur de nouvelles données i.e. une bonne généralisation.

5.3. Les chiffres manuscrits

L'architecture proposée pour la reconnaissance des chiffres manuscrits isolés est présentée dans la figure 5.11.

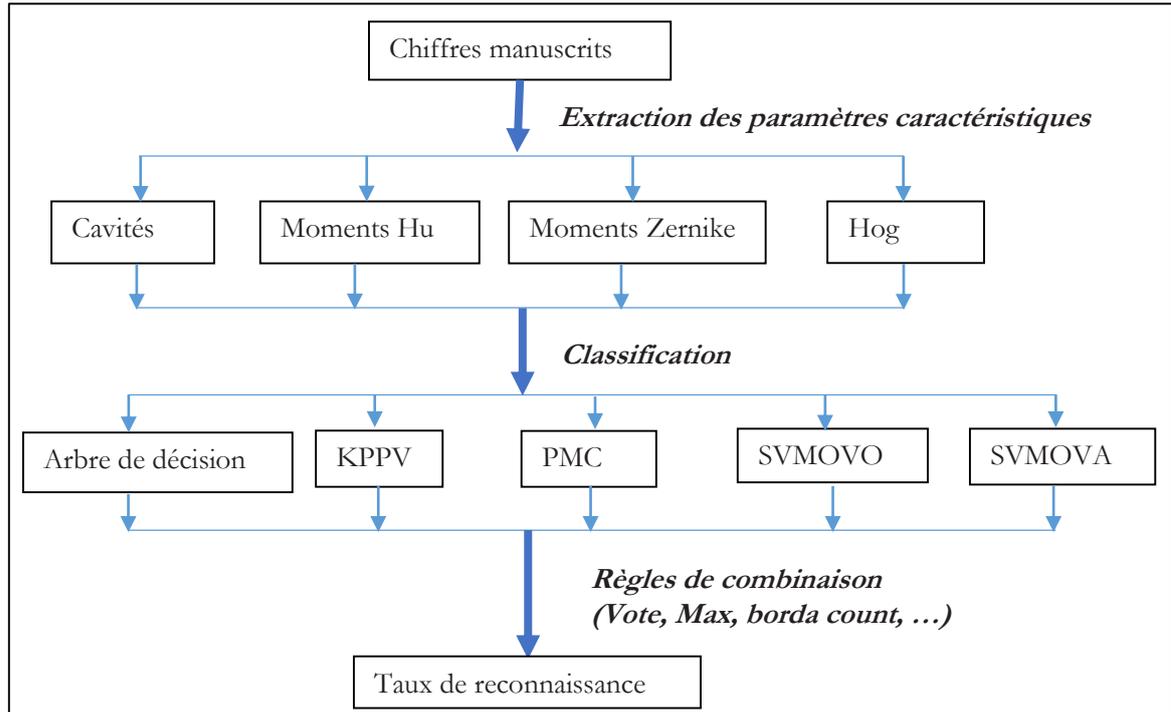


Figure 5.11 : Architecture du système proposé

Le tableau 5.20 présente le taux de reconnaissance de chaque classifieur avec différents paramètres caractéristiques indépendamment. Les résultats illustrent que la meilleure performance varie entre les classifieurs. Avec les cavités, le meilleur résultat est donné par le PMC, les moments Hu et Zernike avec le SVMOVA, HOG avec le KPPV.

Tableau 5.20 : Performance des classifieurs

	KPPV		Arbre de décision		PMC		SVMOVO		SVMOVA	
	Appr	Teste	Appr	Teste	Appr	Teste	Appr	Teste	Appr	Teste
Cavité	42.76	39.87	46.66	44.58	45.58	45.26	43.39	43.40	34.41	34.07
MHu	100	40.57	82.80	53.02	56.91	57.70	37.23	37.81	60.91	59.98
MZernike	100	77.04	92.35	72.66	83.57	83.69	52.88	51.67	91.61	86.97
Hog	100	96.57	97.38	89.12						

Appr : apprentissage.

Nous notons que les vecteurs de caractéristiques, cavité contient cinq cavité et nombre de cavités, Hu contient les sept moments de Hu, Zernike 47 moment de Zernike. Hog est de grande dimension (1296),

Le classifieur SVMOVA a atteint un taux de reconnaissance de 86.97% avec 47 moment de Zernike qui sont invariants aux transformations géométriques. KPPV a donné 96.57% avec

HOG qui a une grande dimension (1296). Les moments de Zernike ont les meilleurs résultats puis les cavités et les moments Hu, mais ne sont pas si bons que les paramètres de HOG. *Il faut noter que HOG a donné des résultats mieux sur les chiffres manuscrits que sur les chiffres imprimés.*

Pour améliorer le taux de reconnaissance un autre paramètre caractéristique supplémentaire est ajouté. Il s'agit du nombre de holes. Le tableau 5.21, indique clairement que cet enrichissement donne de meilleurs résultats pour les cavités et les moments Hu. Un excellent taux de reconnaissance pour HOG avec le classifieur KPPV 97.38%.

Tableau 5.21 : Performance des classifieurs avec enrichissement par "le nombre de holes"

	KPPV		Arbre de décision		PMC		SVMOVO		SVMOVA	
	Appr	Teste	Appr	Test	Appr	Test	Appr	Test	Appr	Test
Cavité	51.89	48.98	55.52	53.87	54.55	54.62	53.24	52.91	39.93	39.76
MHu	100	50.59	86.74	62.86	67.80	67.83	47.42	47.35	70.98	69.12
MZernike	100	79.99	93.83	77.17	84.50	84.50	47.01	46.89	93.17	88.42
Hog	100	96.57	97.38	89.12						

En comparaison avec les résultats obtenus dans [151] pour les moments de Zernike 84.10%, nos résultats illustrent que le classifieur SVMOVA donne un taux de reconnaissance mieux pour les moments de Zernike 88.42% sans aucune normalisation.

Pour augmenter le taux de reconnaissance, nous avons utilisé la combinaison des classifieurs. Nous commençons par la combinaison des classifieurs en utilisant différents paramètres caractéristiques. Dans notre cas, chaque classifieur est entraîné par quatre types de paramètres caractéristiques et trois méthodes de combinaison à savoir vote à la majorité, borda count et la règle de maximum

Le tableau 5.22 montre que la combinaison diminue le meilleur taux de reconnaissance et augmente le taux de reconnaissance le plus bas, à l'exception de classifieur SVMOVO qui augmente le taux de reconnaissance avec les trois règles spécialement avec la règle Bordacount.

Tableau 5.22 : Combinaison des classifieurs avec différents paramètres caractéristiques

	KPPV	Arbre de décision	PMC	SVMOVO	SVMOVA
Vote à la pluralité	65.28	71.81	73.81	49.45	76
Borda count	37.97	78.22	77.93	67.94	67.28
Max	48.98	79.21	81.15	57.34	84.78

Une comparaison expérimentale des différentes combinaisons en utilisant différents types de paramètres caractéristiques est présenté dans le tableau 5.23.

Tableau 5.23 : Combinaison des classifieurs avec les mêmes paramètres caractéristiques

	Cavité	Moments Hu	Moments Zernike	Hog
Vote à la pluralité	54.50	66.74	87.89	92.61
Borda count	53.4500	66.340	86.0300	89.36

Avec les cavités le taux de reconnaissance augmente en comparaison avec le classifieur individuel en exception le classifieur PMC pour le vote majoritaire et à l'exception des classifieurs PMC et l'arbre de décision pour la règle de Borda count mais pas avec une grande différence. Avec les moments Hu les résultats diminuent en comparaison avec les classifieur PMC et SVMOVA.

Le taux de reconnaissance est significativement amélioré avec les moments de Zernike par rapport au classifieur individuel, en particulier l'arbre de décision, KPPV et SVMOVO. En comparaison avec les résultats de SVMOVA, il diminue avec une faible différence. En ce qui concerne HOG, le résultat n'est pas amélioré par rapport au classifieur KPPV. En résumé, lorsque les différents classifieurs sont combinés en utilisant le même type de paramètre caractéristique, les performances ne sont pas améliorées sauf pour les moments de Zernike. Pour les autres paramètres, les performances de combinaison ne sont pas meilleur que du classifieur individuel.

Enfin, nous utilisons différents classifieurs avec différents paramètres caractéristiques. En fonction des résultats précédents, nous choisirons le classifieur qui donne le taux de reconnaissance élevé pour un type de paramètre caractéristique. On obtient donc la combinaison suivante: KPPV-HOG + SVMOVA-Zernike moments+ SVMOVA-Hu moments + PMC-cavities

Les résultats de cette combinaison et d'autres sont présentés dans le tableau 5.24. Un taux de reconnaissance de 94,18% est obtenu avec la combinaison de KPPV-HOG + SVMOVA-moment de Zernike avec la règle de Borda count. Cette valeur se situe entre le taux de reconnaissance le plus élevé obtenu par KPPV-HOG 96,57%, et le taux de reconnaissance obtenu par SVMOVA- moments de Zernike 88,42%.

Tableau 5.24 : Combinaison de différents classifieurs avec différents paramètres caractéristiques

Combinaison de classifieurs	vote à la pluralité	Borda count
KPPV-HOG SVMOVA- Moment Zernike SVMOVA- Moment Hu PMC-cavities	90.23	91.38
KPPV-HOG SVMOVA- Moment Zernike	91.81	94.18
SVMOVA- Moment Zernike SVMOVA- Moment HU	79.43	81.79
SVMOVA- Moment Zernike PMC-cavities	68.78	80.26

En raison des performances plus élevées de SVM en classification, de l'invariance au transformation géométrique des moments de Zernike et de la grande dimension du vecteur HOG utilisée pour le classifieur rapide KPPV, nous pouvons dire que cette combinaison est très satisfaisante.

Notre expérience indique que les travaux proposés ci-dessus sont bons en comparaison avec d'autres systèmes, le tableau 5.25, mais pas suffisants. D'autres améliorations sont demandées, telles que l'enrichissement de l'extraction des caractéristiques avant de les attribuer aux classifieurs, i.e. éventuellement si nous ajoutons des caractéristiques supplémentaires à ces vecteurs, telles que caractéristique concave ou point final, ... etc. Les résultats seraient encore meilleurs. Un taux de reconnaissance élevé peut également être atteint en fonction de l'étape de prétraitement, qui est une séquence d'opérations effectuée sur l'image d'entrée afin d'améliorer la qualité de l'image : suppression du bruit, normalisation de la taille de l'image (emplacement du chiffre).

Tableau 5.25 : Résultats de références sur la base MNIST

Références	Méthodes	Taux de reconnaissance en %
[151]	SVM moment Zernike normalisé	94.33
[38]	Histogramme de projection SVM, algorithme de bat	95.60
[158]	SVM Paramètres triangulaire améliorée	93.18
[158]	PMC Paramètres triangulaire améliorée	96.51
[178]	KPPV SVM	99.26
[178]	KPPV (k=3)	88.98
[116]	Théories Neuronal, correspondance de Graphe élastique (Elastic Graph Matching) Paramètres de Gabor	96.528
[249]	KPPV Nombre de cavité, Réservoirs d'eau dans quatre directions (cavité), Distances de profil maximales dans quatre directions Densité de cavité (trous)	96.94
[131]	CNN + SVM	99.81 sans rejet 94.40 avec rejet
[119]	SVM+Arbre de décision non balancé (UDT Unbalanced Decision Tree)	100

Notre contribution à cette recherche porte principalement sur l'amélioration de la classification en termes de taux de reconnaissance en combinant des classifieurs. Trois méthodes sont utilisées pour la combinaison des classifieurs à savoir vote à la majorité, nombre de Bordacount, règle max. Les expériences ont démontré que la combinaison de KPPV avec les caractéristiques HOG et SVMOVA avec les moments de Zernike par la règle de comptage de Borda a considérablement amélioré le taux de reconnaissance de 86,97% des moments de Zernike à 94,18% et réduit le taux de reconnaissance de 96,57% de la fonctionnalité HOG à 94,18%. Cette diminution de 2% n'est pas prise en compte avec l'augmentation de 8% et confère la propriété d'invariance aux transformations géométriques.

L'efficacité globale obtenue dans le présent travail a été comparée en résumé à celle d'autres références. Notre résultat atteint un taux de reconnaissance de 96.57% (par HOG), ce qui est supérieur à celui des travaux énumérés dans le tableau ci-dessus [38, 39, 116, 131, 151, 158, 178]. Le taux de reconnaissance dans [249] est 96.94% mais sur 5000 image de chiffres. Dans

[178] a atteint 99.26% par une combinaison cascade de KPPV et SVM sans prendre en considération l'invariance aux transformations géométriques dans les paramètres caractéristiques (voir section 1.6). La même chose pour le résultat de [249]. Cela indique la supériorité de notre résultat encourageant.

Dans [131] dix-neuf chiffres ont été mal classés par le modèle hybride CNN-SVM. L'étiquette supérieure sur la figure 5.12 est le numéro d'identification de l'image dans la base de données de test MNIST et l'étiquette inférieure est la prédiction correspondante de la vrai. L'auteur a présenté ainsi un pourcentage de 67.94% de la classification correcte de 25 chiffres (y compris les 19 chiffres) par 70 participants humains. Nous voyons vraiment l'illisibilité de ces chiffres par l'être humain.

Selon l'auteur la raison de la mauvaise reconnaissance par le modèle hybride pourrait être due au manque d'échantillons d'apprentissage avec de telles structures. Pour résoudre ce problème, il a proposé d'importer dans la base de données davantage d'échantillons d'apprentissage non utilisés. Cela signifie une mémorisation des chiffres ou apprentissage par cœur de ce modèle. En plus de la grande dimension du vecteur caractéristique (l'image entier).

Un taux de reconnaissance de 100% a été atteint par [119] en utilisant l'image entier i.e. la propriété de l'invariance aux transformations géométriques n'est pas prise en considération ainsi la dimension de vecteur de caractéristique. L'être humain n'a pas pu reconnaître son écriture, comment la machine a pu le faire. Nous voyons ça comme une contradiction et cela confirme l'apprentissage par cœur de ces modèles.



Figure 5.12 : Chiffres mal classés [131]

Certaines difficultés possibles contribuant à la non reconnaissance de quelques chiffres (mauvaise classification) sont les suivantes :

- Variété des styles d'écriture manuscrite influe sur l'apparence des chiffres.
- Similitude entre les chiffres 1 et 7, 5 et 6, 3 et 8, 9 et 8, 3 et 9 etc.
- Inclinaison de l'écrivain. Elle constitue simplement un facteur contribuant à la variabilité du script.
- Chiffres brisés(cassés) peut être dû à de mauvaises conditions d'écriture ou à une mauvaise qualité des processus de numérisation et de binarisation (chiffres 5 8 9 dans la figure 5.13).
- Mauvaise qualité des images en raison du processus de numérisation, de l'instrument d'écriture ou de la normalisation des tailles. Certaines parties de l'image sont brisées ou des bruits sont générés.
- Cercle rempli provoqué par la vitesse de l'écrivain, comme le cercle inférieur du chiffre huit (chiffre 6 8 dans la figure 5.13)
- Fins se touchent comme le chiffre trois et forme des cercles (figure 5.13).
- Similarité de la structure et la forme de certaines paires de nombres causé par le même mouvement d'écriture : par ex. chiffres 4-9, 0-6, 1-7
- Manque d'échantillons d'apprentissage de certains chiffres écrits dans un style atypique [250]

Une Analyse des erreurs de chiffres manuscrites faites par une multitude de classificateurs est présentée dans [250]. Cependant, selon ces difficultés, il est irréaliste de disposer d'un système de reconnaissance manuscrite des chiffres avec une précision de reconnaissance de 100%. Il n'est pas possible de créer un système de reconnaissance général reconnaissant n'importe quel chiffre dans chaque application.

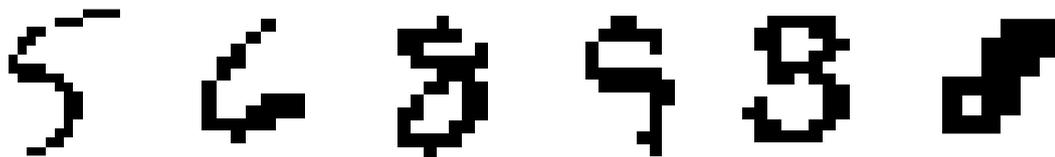


Figure 5.13 : Quelques chiffres (5 6 8 9 3 8) manuscrits déformés

5.4. Conclusion

Un système de reconnaissance de chiffres peut être divisé en plusieurs étapes: prétraitement, extraction de caractéristiques (et sélection), classification. Ce travail se concentre sur l'extraction des caractéristiques et la classification.

Dans ce chapitre nous avons présenté les différentes applications dans lesquelles les travaux de la thèse ont été intégrés. Quatre applications sur les chiffres imprimés, elles ont conduit à la mise au point de deux méthodes d'extraction de paramètres caractéristiques robustes qui ont été validées expérimentalement sur deux bases de chiffres imprimés.

Nous nous sommes ensuite penchés sur le problème de MCS. Les classifieurs présentant des taux de reconnaissance plus élevés sont beaucoup plus importants même si certains échantillons peuvent être classés de manière plus fiable par d'autres. L'approche multi-classifieurs a pour objectif d'accroître le taux de reconnaissance en utilisant les règles de combinaisons. En conséquence, la conclusion tirée est que le MCS qui donne une performance plus élevée c'est celui qui utilise des paramètres caractéristiques différentes soit par un même type de classifieur soit par des classifieurs différents avec la règle de combinaison vote à la majorité.

Nos travaux ont abouti à des résultats satisfaisants sur une base de données largement utilisée, le MNIST. Les meilleurs résultats sont obtenus avec les caractéristiques HOG. Les moments Hu et les cavités apparaissent légèrement pires. Mais les moments de Zernike sont les plus prometteurs compte tenu du fait qu'ils vérifient la propriété de l'invariance aux transformations géométriques.

CONCLUSION GENERALE & PERSPECTIVES

La discipline de RdF traite essentiellement du problème du développement d'algorithmes permettant de mettre en œuvre sur ordinateur de nombreuses tâches de reconnaissance normalement effectuées par l'être humain. La motivation est d'exécuter ces tâches avec plus de précision, ou plus rapidement, et peut-être encore, de manière plus économique que les humains et, dans de nombreux cas, de les libérer des tâches fastidieuses résultant de l'exécution de tâches de reconnaissance de routine de manière répétitive et mécanique. La portée des relations publiques englobe également les tâches pour lesquelles les humains ne sont pas doués, comme la lecture de codes à barres. La recherche sur la RdF vise à concevoir des moyens d'automatiser certains processus de prise de décision menant à la classification et à la reconnaissance [216].

La RdF est un problème de classification où d'objets (patterns) inconnus devront être associés à une classe d'appartenance. Afin de réaliser cette tâche, des classifieurs doivent être sélectionnés. Une approche classique consiste à choisir le classifieur le plus performant sur une base de validation. Une autre approche consiste à choisir et à combiner un ensemble de classifieurs. Il a été montré dans la littérature qu'en général, les MCS généralisent mieux que les classifieurs individuels sur des nouvelles données.

Ces dernières années, les recherches dans le domaine de la RdF ont porté sur l'étude d'algorithmes efficaces permettant de combiner les résultats de nombreux classifieurs simples plutôt que sur l'étude de classifieurs individuels puissants mais extrêmement complexes.

En fait, pour un problème de classification quelconque et en utilisant un ensemble de classifieurs ayant des performances intrinsèques différentes sur une base de données spécifique, nous pouvons trouver une méthode de combinaison qui permet de donner le meilleur résultat. Si on change de problème (par exemple, en changeant de base de données ou en changeant les classifieurs à combiner), nous ne pouvons pas garantir que la méthode choisie reste la meilleure. Par conséquent, les travaux actuels restent difficiles à généraliser en dehors d'un domaine applicatif donné. Ce qui explique bien les limites des systèmes développés du point de vue performance pour traiter d'autres applications plus complexes.

La conception d'un MCS optimale n'est pas toutefois une tâche simple. Cependant, aucune méthode n'est supérieure à toutes les autres de manière consistante. D'autres part, la combinaison des meilleurs classifieurs sélectionnés sur la base de validation ne donne pas obligatoirement une meilleure performance sur la base de test. Le problème général de la sélection des caractéristiques convenable et de la combinaison efficace des résultats de différents classifieurs. La détermination de la meilleure organisation reste un problème ouvert.

La reconnaissance automatique de l'écriture des chiffres manuscrits et imprimés tient une place de plus en plus prépondérante dans les préoccupations industrielles et scientifiques (imprimé au même titre que celle de manuscrit). Ce document a présenté l'ensemble du processus de reconnaissance d'un chiffre manuscrit ou imprimé, depuis l'image jusqu'au résultat final constitué d'une réponse. Les travaux présentés portent sur différents classifieurs appliqués à la reconnaissance des chiffres arabe, basant sur différents types de paramètres caractéristiques soit indépendamment ou par combinaison.

Synthèse du travail effectué

Cette thèse porte sur plusieurs aspects traitant la conception d'un MCS pour la mise en œuvre de systèmes de reconnaissance des chiffres arabe. Des méthodes novatrices sont proposées et analysés pour obtenir des résultats plus performants.

Cette thèse est organisée en en deux parties. La partie théorique et La partie pratique. La première partie contient trois chapitres. Le premier chapitre donne brièvement un historique sur la reconnaissance des caractères et un état de l'art des travaux de reconnaissance des chiffres manuscrits et imprimés. Les deux autres chapitres présentent un rappel sur le système de RdF et le MCS. Le second se propose de redéfinir les différentes étapes de conception d'un système de RdF (prétraitement, extraction des paramètres caractéristique, classification) et donne une description des approches existantes dont les deux principales qui sont l'approche statistique et structurelle. Le troisième chapitre introduit les différentes approches d'un système MCS ainsi que les méthodes de combinaison des classifieurs.

La seconde partie dédiée à l'application et aux résultats expérimentaux. Elle contient deux chapitres. Le premier est consacré aux techniques utilisées dans notre système proposé à savoir les caractéristiques, les classifieurs et les méthodes de combinaison. Le deuxième chapitre expose les différents résultats des applications qui sont ensuite analysés et discutés en essayant de mettre en évidence les avantages et les limitations du système présentés.

Enfin nous terminons par une conclusion et quelques perspectives à ces travaux.

Contributions

Les premiers résultats obtenus sur les paramètres caractéristiques ont été décevants. Donc Nous avons réalisé plusieurs simulations pour éprouver les classifications en introduisant des améliorations dans les paramètres caractéristiques et en faisant des combinaisons de classifieurs.

Quatre applications sur les chiffres imprimés, ont conduit à la mise au point de deux méthodes d'extraction de paramètres caractéristiques robustes qui ont été validées expérimentalement sur deux bases de chiffres imprimés (BCI_1, BCI_2 et BCI_3).

Nous nous sommes ensuite penchés sur le problème le MCS, les classifieurs présentant des taux de reconnaissance plus élevés sont beaucoup plus importants même si certains échantillons peuvent être classés de manière plus fiable par d'autres. L'approche multi-classifieurs a pour objectif d'accroître le taux de reconnaissance i.e. améliorent les performances de chacun pris séparément en utilisant les règles de combinaisons. En conséquence, la conclusion tirée est que le MCS qui donne une performance plus élevée c'est celui qui utilise des paramètres caractéristiques différente soit par un même type de classifieurs ou par des classifieurs différents avec la règle de combinaison vote à la majorité.

Nos travaux ont abouti à des résultats satisfaisants sur une base de données largement utilisée, le MNIST, malgré les difficultés possibles contribuant à la non reconnaissance de quelques chiffres (mauvaise classification). Les meilleurs résultats sont obtenus par la combinaison KPPV-HOG + SVMOVA-moment de Zernike, elle donne un taux de reconnaissance très satisfait. Elle profite de la performance élevée de SVM, de l'invariance au transformation géométrique des moments de Zernike et la rapidité du classifieur KPPV qui réduire la complexité du système causé par la grande dimension du vecteur HOG. Cela indique que notre modèle n'est pas seulement efficace et comparable à l'état de l'art mais aussi répond à la propriété de l'invariance aux transformations géométriques.

Perspectives

Bien entendu, les approches que nous avons proposées sont encore largement perfectibles et plusieurs perspectives de travaux de recherche sont envisageables. Notre système s'appuie comme une base pour un développement ultérieur. Nous pouvons citer

- Mettre le doigt sur le difficile problème que pose la dimension des caractéristiques et proposer des méthodes partielles pour la sélection des caractéristiques principales.
- Utiliser la méthode structurelle en terme de classification.

- Utiliser le vote à la pluralité pondérée en utilisant le taux de reconnaissance de chaque expert.

- Enrichir les caractéristique en les combinant avec d'autres et utiliser de nouvelles techniques d'amélioration d'image lors du prétraitement pour améliorer également les performances d'extraction des paramètres et accroître le taux de reconnaissance.

- Utiliser Les méthodes qui utilisent le concept d'oracle, c'est-à-dire celui qui peut donner de sages conseils [224]

En conséquence, la principale conclusion de ces travaux est que le problème de la conception optimale du MCS reste toujours ouvert. Cependant, nous espérons que les références citées couvriront les principaux problèmes théoriques, guideront le chercheur dans des directions de recherche intéressantes et suggéreront d'éventuelles propositions (combinaisons de biais) qui doivent encore être explorées. Nous espérons aussi que ce travail contribuera à accroître notre connaissance de ce domaine de recherche.

REFERENCES BIBLIOGRAPHIQUES

- [1] Britto Jr., A. (2001). *A two-stage HMM-based method for recognizing handwritten numeral strings*. Thèse de Doctorat, Pontifical Catholic University of Paraná, Brésil.
- [2] Chitoub, S. (2004). Combinaison de classifieurs : une approche pour l'amélioration de la classification d'images multisources multitudes de télédétection. *Télédétection*, 4(3): 289-301.
- [3] Jain, A., Subrahmanyam, G.R.S. & Mishra, D. (2018). Rotation invariant digit recognition using convolutional neural network. *Proceedings of 2nd International Conference on Computer Vision & Image Processing Springer, Singapore*. 91-102.
- [4] Ayed, S.B., Trichili, H. & Alimi, A.M. (2015). Data fusion architectures: A survey and comparison. *15th International Conference on Intelligent Systems Design and Applications*. 277-282
- [5] Zouari, H.K. (2004). *Contribution à l'évaluation des méthodes de combinaison parallèle de classifieurs par simulation*. Thèse de Doctorat, Université de Rouen, France.
- [6] Laaksonen, J. (1997). *Subspace classifiers in recognition of handwritten digits*. Thèse de Doctorat, Helsinki University of Technology, Finland.
- [7] Ayat, N. (2004). *Sélection de modèle automatique des machines à vecteurs de support : application à la reconnaissance d'images de chiffres manuscrits*. Thèse de Doctorat, École Supérieure de Technologie, Quebec.
- [8] Oliveira, L.E.S. (2003). *Automatic recognition of handwritten numerical strings*. Thèse de Doctorat, École Supérieure de Technologie, Quebec.
- [9] Jain, A.K., Duin, R.P.W. & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1): 4-37.
- [10] Harmon, L.D. (1972). Automatic recognition of print and script. *Proceedings of the IEEE*, 60(10): 1165-1176.
- [11] Mantas, J. (1986). An overview of character recognition methodologies. *Pattern Recognition*, 19(6): 425-430.
- [12] Tappert, C.C., Suen, C.Y. & Wakahara, T. (1990). The state of the art in online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8): 787-808.
- [13] Arica, N. & Yarman-Vural, F.T. (2001). An overview of character recognition focused on off-line handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(2): 216-233.
- [14] Suen, C.Y., Kim, J., Kim, K., Xu, Q. & Lam, L. (2000). Handwriting recognition-the last frontiers. *Proceedings 15th International Conference on Pattern Recognition, IEEE*, 4: 1-10.
- [15] Lazrek, A. (2002). *Vers un système de traitement du document scientifique arabe*. Thèse de Doctorat, Université Cadi Ayyad, Marrakech.

- [16] Ayat, N.E., Cheriet, M. & Suen, C.Y. (2000). Un système neuro-flou pour la reconnaissance de montants numériques de chèques arabes. *Colloque International Francophone sur l'Écrit et le Document*. 171-180.
- [17] Morita, M.E. (2003). *Automatic recognition of handwritten dates on brazilian bank cheque*. Thèse de Doctorat, École Supérieure de Technologie, Quebec.
- [18] Xu, Q., Lam, L. & Suen, C.Y. (2003). Automatic Segmentation and recognition system for handwritten dates on Canadian bank chèques. *Proceedings of the Seventh International Conference on Document Analysis and Recognition, IEEE Computer Society*.
- [19] Xu, Q., Kim, J.H., Lam, L. & Suen, C.Y. (2002). Recognition of handwritten month words on bank chèques. *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, IEEE*. 111-116.
- [20] Grandidier, F. (2003). *Un nouvel algorithme de sélection de caractéristiques : application à la lecture automatique de l'écriture manuscrite*. Thèse de Doctorat, École Supérieure de Technologie, Quebec.
- [21] Sermanet, P., Chintala, S. & LeCun, Y. (2012). Convolutional neural networks applied to house numbers digit classification. *Proceedings of the 21st International Conference on Pattern Recognition*, November. 3288-3291.
- [22] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. & Ng, A.Y. (2011). Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.
- [23] Mohamed, B.R., El Kader, H.M.A., Rafaat, H.M. & Sharaf, M.S. (2013). Automatic number plate recognition. *International Journal of Scientific and Research Publications*, 3(12): 59-63.
- [24] Caner, H., Gecim, H.S. & Alkar, A.Z. (2008). Efficient embedded neural-network-based license plate recognition system. *IEEE Transactions on Vehicular Technology*, 57(5): 2675-2683.
- [25] Patel, C., Shah, D. & Patel, A. (2013). Automatic number plate recognition system (anpr): A survey. *International Journal of Computer Applications*, 69(9): 21-33.
- [26] El Rahman, A.A., Hamdy, A. & Zaki, F. (2013). Automatic arabic number plate recognition. *The 2nd International Japan-Egypt Conference on Electronics, Communications and Computers*. 23-28.
- [27] Alginahi, Y.M. (2011). Automatic arabic license plate recognition. *International Journal of Computer and Electrical Engineering*, 3(3): 454-460.
- [28] Shankar, S.K. (2015) Skew and slant corrections for IR images in Indian license plate recognition using HCD & HPR. *International Journal of Research and Scientific Innovation*, 2(11) : 78-84.
- [29] Do, H.N., Vo, M.T., Luong, H.Q., Nguyen, A.H., Trang, K. & Vu, L.T. (2017). Speed limit traffic sign detection and recognition based on support vector machines. *International Conference on Advanced Technologies for Communications, IEEE*. 274-278.
- [30] Greenhalgh, J. & Mirmehdi, M. (2012). Real-time detection and recognition of road traffic signs. *IEEE Transactions on Intelligent Transportation Systems*, 13(4): 1498-1506.

- [31] Sahbani, A. (2018). NARMA-L2 Neuro controller for speed regulation of an intelligent vehicle based on image processing techniques. *The 21st Saudi Computer Society National Computer Conference, IEEE*. 1-6.
- [32] Wang, D. & Kong, S. (2013). Learning class-specific dictionaries for digit recognition from spherical surface of a 3D ball. *Machine Vision and Applications*, 24(6): 1213-1227.
- [33] Lin, X., Yacoub, S., Burns, J. & Simske, S. (2003). Performance analysis of pattern classifier combination by plurality voting. *Pattern Recognition Letters*, 24(12): 1959-1969.
- [34] Ribas, F.C., Oliveira, L.S., Britto, A.S. & Sabourin, R. (2013). Handwritten digit segmentation: a comparative study. *International Journal on Document Analysis and Recognition (IJДАР)*, 16(2): 127-137.
- [35] Oliveira, L.S., Sabourin, R., Bortolozzi, F. & Suen, C.Y. (2003). A methodology for feature selection using multiobjective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(06): 903-929.
- [36] Alpaydin, E., Kaynak, C. & Alimoglu, F. (2000). Cascading multiple classifiers and representations for optical and pen-based handwritten digit recognition. *International Workshop on Frontiers in Handwriting Recognition, Amsterdam, The Netherlands*. 453-462
- [37] Mali, S.M. (2012). Moment and density based handwritten Marathi numeral recognition. *Indian journal of computer science and engineering (IJCSE)*, 3(5): 707-712.
- [38] Tuba, E., Tuba, M. & Simian, D. (2016). Handwritten digit recognition by support vector machine optimized by bat algorithm. *The 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. 369-376.
- [39] Thomas, S. (2018). A Study of representation learning for handwritten numeral recognition of multilingual data set. *Information and Communication Technology for Sustainable Development, Springer, Singapore*. 475-482.
- [40] Holambe, A.N., Holambe, S.N. & Thool, R.C. (2010). Comparative study of Devanagari handwritten and printed character & numerals recognition using Nearest-Neighbor classifiers. *The 3rd International Conference on Computer Science and Information Technology, IEEE*, 1: 426-430.
- [41] Mhetre, S.L. & Patil, M.M. (2013). Comparative study of two methods for Handwritten Devanagari Numeral Recognition. *IOSR Journal of Computer Engineering*, 15: 49-53.
- [42] Mali, S.M. & Patil, C.H. (2015). Marathi handwritten numeral recognition using Zernike moments and Fourier descriptors. *National Conference on Digital Image and Signal Processing*. 32-34.
- [43] Jameel, M. & Kumar, S. (2017). Offline recognition of handwritten Urdu characters using B spline curves: A survey. *International Journal of Computer Applications*, 157(1): 28-34.
- [44] Boukharouba, A. & Bennia, A. (2017). Novel feature extraction technique for the recognition of handwritten digits. *Applied Computing and Informatics*, 13(1): 19-26.

- [45] Alaei, A., Pal, U. & Nagabhushan, P. (2009). Using modified contour features and SVM based classifier for the recognition of Persian/Arabic handwritten numerals. *The 7th International Conference on Advances in Pattern Recognition, IEEE*. 391-394.
- [46] Alaei, A., Nagabhushan, P. & Pal, U. (2009). Fine classification of unconstrained handwritten Persian/Arabic numerals by removing confusion amongst similar classes. *The 10th International Conference on Document Analysis and Recognition, IEEE*. 601-605.
- [47] Parvin, H., Alizadeh, H., Minaei-Bidgoli, B. & Analoui, M. (2008). A scalable method for improving the performance of classifiers in multiclass applications by pairwise classifiers and GA. *The 4th International Conference on Networked Computing and Advanced Information Management, IEEE*. 137-142.
- [48] Parvin, H., Alizadeh, H., Moshki, M., Minaei-Bidgoli, B. & Mozayani, N. (2008). Divide & conquer classification and optimization by genetic algorithm. *The 3rd International Conference on Convergence and Hybrid Information Technology, IEEE*, 2: 858-863.
- [49] Karimi, H., Esfahanimehr, A., Mosleh, M., Salehpour, S. & Medhati, O. (2015). Persian handwritten digit recognition using ensemble classifiers. *Procedia Computer Science*, 73: 416-425.
- [50] Montazer, G.A., Soltanshahi, M.A. & Giveki, D. (2017). Farsi/Arabic handwritten digit recognition using quantum neural networks and bag of visual words method. *Optical Memory and Neural Networks*, 26(2): 117-128.
- [51] Behbahan, N.S. & Mousavinasab, Z. (2013). A method based on decision trees and neural network for recognition of Farsi handwritten digits. *International Journal of Management, IT and Engineering*, 3(7): 455-464.
- [52] Benchaou, S., Nasri, M., El Melhaoui, O. & Bouali, B. (2014). New approach of preprocessing for numeral recognition. *International Journal of Engineering Research and Applications*, 4(7): 26-30.
- [53] Moradi, V., Razzazi, F. & Behrad, A. (2016). Recognition of handwritten Persian two-digit numerals using a novel hybrid SVM/HMM algorithm. *Majlesi Journal of Electrical Engineering*, 10(3): 19-25.
- [54] Azad, R., Davami, F. & Shayegh, H.R. (2014). Recognition of handwritten Persian/Arabic numerals based on robust feature set and K-NN classifier. *arXiv preprint arXiv: 1407.6492*.
- [55] Boveiri, H.R. (2010). Persian printed numeral characters recognition using geometrical central moments and fuzzy min-max neural network. *International Journal of Signal Processing*, 6(2): 76-82.
- [56] Khan, H.A., Al Helal, A. & Ahmed, K.I. (2014). Handwritten bangla digit recognition using sparse representation classifier. *The International Conference on Informatics, Electronics & Vision, IEEE*. 1-6.
- [57] Alom, M.Z., Sidike, P., Taha, T.M. & Asari, V.K. (2017). Handwritten bangla digit recognition using deep learning. *arXiv preprint arXiv: 1705.02680*.
- [58] Majumdar, A. & Chaudhuri, B. B. (2006). A MLP classifier for both printed and handwritten Bangla numeral recognition. In: Kalra P.K. & Peleg S. (eds.). *Computer Vision, Graphics and Image Processing. Vol. 4338. Springer, Berlin, Heidelberg*. 796-804.

- [59] El Hindi, K., Khayyat, M. & Abu Kar, A. (2017). Comparing the machine ability to recognize hand-written Hindu and Arabic digits. *Intelligent Automation & Soft Computing*, 23(2): 295-301.
- [60] Jeiad, H.A. (2018). Indian number handwriting features extraction and classification using multi-class SVM. *Engineering and Technology Journal*, 36(1): 33-40.
- [61] Taqi, A.K. & Korial, A.E. (2018). Comparison between feature based and deep learning recognition systems for handwriting Arabic numbers. *Computer Engineering and Intelligent Systems*, 9(4): 51-66.
- [62] Ashiquzzaman, A. & Tushar, A.K. (2017). Handwritten Arabic numeral recognition using deep learning neural networks. *IEEE International Conference on Imaging, Vision & Pattern Recognition*. 1-4.
- [63] Musleh, D., Halawani, K. & Mahmoud, S. (2015). Fuzzy modeling for handwritten Arabic numeral recognition. *International Arab Journal of Information Technology*, 14(4): 1-10
- [64] Al-Zoubi, H. & Al-Khassaweneh, M. (2008). Offline machine-print Hindi digit recognition using translational motion estimation. *International Conference on Computational Intelligence for Modelling Control & Automation, IEEE*. 1118-1122.
- [65] Das, N., Mollah, A.F., Saha, S. & Haque, S.S. (2010). Handwritten arabic numeral recognition using a multi-layer perceptron. *arXiv preprint arXiv: 1003.1891*.
- [66] Rani, N.S. & Ashwini, P.S. (2014). A standardized frame work for handwritten and printed Kannada numeral recognition and translation using probabilistic neural networks. *International Journal of Innovative Science, Engineering & Technology*, 1(4): 178-182.
- [67] Rajput, G.G., Horakeri, R. & Chandrakant, S. (2010). Printed and handwritten mixed Kannada numerals recognition using SVM. *International Journal on Computer Science and Engineering*, 2(05): 1622-1626.
- [68] Hegadi, R.S. (2012). Recognition of printed Kannada numerals based on zoning method. *National Conference on Advanced Computing and Communications, NCACC*, April. 5-8.
- [69] Mamatha, H.R., Karthik, S. & Srikanta, M.K. (2013). Classifier fusion method to recognize handwritten Kannada numerals. *arXiv preprint arXiv: 1301.0167*.
- [70] Ramappa, M.H. & Krishnamurthy, S. (2013). A comparative study of different feature extraction and classification methods for recognition of handwritten Kannada numerals. *International Journal of Database Theory & Application*, 6(4): 71-90.
- [71] Singh, P. & Budhiraja, S. (2012). Offline handwritten gurmukhi numeral recognition using wavelet transforms. *International Journal of Modern Education and Computer Science*, 4(8): 34-39.
- [72] Dash, K.S., Puhan, N.B. & Panda, G. (2015). Gestalt configural superiority effect: a complexity paradigm for handwritten numeral recognition. *The 8th International Conference on Advances in Pattern Recognition, IEEE*. 1-6.
- [73] Brown, G. (2009). An information theoretic perspective on multiple classifier systems. *International Workshop on Multiple Classifier Systems*. 344-353.

- [74] Singh, P.K., Das, S., Sarkar, R. & Nasipuri, M. (2017). Recognition of handwritten *Indic* script numerals using Mojette transform. *Proceedings of the 1st International Conference on Intelligent Computing and Communication*. 459-466.
- [75] Dhandra, B.V., Benne, R.G. & Hangarge, M. (2011). Kannada, Telugu and Devanagari handwritten numeral recognition with probabilistic neural network: a script independent approach. *International Journal of Computer Applications*, 26(9): 11-16.
- [76] Dhandra, B.V., Mukarambi, G. & Hangarge, M. (2011). Kannada and English numeral recognition system. *International Journal of Computer Applications*, 26(9): 17-22.
- [77] Buchowski, T. J. & Hwang, V.S.S. (1988). *Omnirec : a character recognition system*. Master's thesis, University of Texas at Austin.
- [78] El Melhaoui, O., Maroc, M., El Hitmy, M. & Lekhal, F. (2011). Arabic numerals recognition based on an improved version of the loci characteristic. *International Journal of Computer Applications*, 24(1), pp 36-41.
- [79] Alqudah, A.T., Al-Zoubi, H.R. & Al-Khassaweneh, M.A. (2012). Shift and scale invariant recognition of printed numerals. *Abhath Al-Yarmouk: Basic Sci. & Eng*, 21(1), pp. 41-49.
- [80] Radha, R. & Aparna, R.R. (2014). Digit recognition using hybrid classifier. *World Congress on Computing and Communication Technologies, IEEE*. 34-38.
- [81] Sharma, N., Kumar, B. & Singh, V. (2014). Recognition of off-line hand printed English characters, numerals and special symbols. *The 5th International Conference-Confluence The Next Generation Information Technology Summit, IEEE*. 640-645.
- [82] <http://rodrigob.github.io/are-we-there-yet/build/classification-datasets-results.html>. Consulté en 2018.
- [83] Boveiri, H.R. (2011). Transformation-invariant classification of Persian printed digits. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 4(3): 153-164.
- [84] Boveiri, H.R. (2010). Persian printed numerals classification using extended moment invariants. *World Academy of Science, Engineering and Technology*, 63: 167-174.
- [85] Medhi, K. & Kalita, S. K. (2014). Recognition of Assamese handwritten numerals using mathematical morphology. *International Advance Computing Conference, IEEE*. 1076-1080.
- [86] Mohapatra, R. K., Majhi, B. & Jena, S. K. (2016). Printed Odia digit recognition using finite automaton. *The 3rd international conference on advanced computing, networking and informatics*. 643-650.
- [87] Kaur, I. & Singh, K.J. (2016). Printed Text Recognition System for Multi-Script Image. *International Journal of Signal Processing Systems*, 4(5): 411-416.
- [88] Malallah, F.L., Saeed, M.G., Aziz, M.M., Arigbabu, O.A. & Ahmad, S.M.S. (2016). Off-Line Arabic (Indian) Numbers Recognition Using Expert System. *International Journal of Advanced Computer Science and Applications*, 7(4): 397-406.

- [89] Helali, M., Alneghaimish, A. & Ahmad, I. (2017). Handwritten digit recognition under constrained training conditions. *8th International Conference of Pattern Recognition Systems*.
- [90] Naz, S., Ahmed, S., Ahmad, R. & Razza, M. (2015). Arabic script based digit recognition systems. *International Conference on Recent Advances in Computer Systems*, 67-73.
- [91] Banjare, K., Massey, S. & CSVTU, B. (2016). Handwritten Numeric Digit Classification and Recognition: Recent Advancements. *International Journal of Emerging Technologies in Engineering Research*, 4(6): 228-231.
- [92] Nicolas, S., Kessentini, Y., Paquet, T. & Heutte, L. (2005). Handwritten document segmentation using hidden Markov random fields. *IEEE In Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, 212-216.
- [93] Kessentini, Y., Paquet, T. & Hamadou, A. B. (2010). Off-line handwritten word recognition using multi-stream hidden Markov models. *Pattern Recognition Letters*, 31(1): 60-70.
- [94] Kessentini, Y., Paquet, T. & Ben Hamadou, A. (2009). Reconnaissance de l'écriture manuscrite arabe et latine par des modèles de Markov cachés multi-Flux, *Traitement du Signal*, 26(5): 395-407.
- [95] Cai, J. & Liu, Z. Q. (1999). Integration of structural and statistical information for unconstrained handwritten numeral recognition. *Transactions on Pattern Analysis and Machine Intelligence, IEEE*. 21(3): 263-270.
- [96] Trier, Ø.D., Jain, A.K. & Taxt, T. (1996). Feature extraction methods for character recognition-a survey. *Pattern Recognition*, 29(4) : 641-662.
- [97] Ali, F. & Pavlidis, T. (1977). Syntactic recognition of handwritten numerals. *Transactions on Systems, Man and Cybernetics, IEEE*. 7(7): 537-541.
- [98] Lee, S. W. (1996). Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network. *Transactions on Pattern Analysis and Machine Intelligence, IEEE*. 18(6): 648-652.
- [99] LeCun, Y., Jackel, L. D., Bottou, L., Brunot, A., Cortes, C., Denker, J.S., ... & Simard, P. (1995). Comparison of learning algorithms for handwritten digit recognition. *International conference on artificial neural networks*, 60: 53-60.
- [100] Correia, S. E. N. (2005). *Reconhecimento de caracteres manuscritos usando wavelets*. Thèse de doctorat, Campina Grande, Paraíba, Brazil.
- [101] Karić, M. & Martinović, G. (2013). Improving Offline Handwritten Digit Recognition Using Concavity-based Features. *International Journal of Computers Communications & Control*, 8(2): 220-234.
- [102] Reddy, R.V. K. & Babu, U.R. (2018). Efficient Handwritten Digit Classification using User-defined Classification Algorithm. *International Journal on Advanced Science, Engineering and Information Technology*, 8(3): 970-979.
- [103] Cao, J., Ahmadi, M. & Shridhar, M. (1994, May). Handwritten numeral recognition with multiple features and multistage classifiers. *Proceedings of IEEE, International Symposium on Circuits and Systems*. 6: 323-326.

- [104] Lee, D.S., (1995). *A theory of classifier combination: the neural network approach*. Thèse de doctorat, University of New York.
- [105] Giacinto, G. (1998). *Design of multiple classifier systems*. Thèse de doctorat, University of Salerno, Italy.
- [106] Gasmî, I., Merouani, H. & Souici-Meslati, L. (2005). Combinaison de classifieurs. *The 3rd International Conference: Sciences of Electronic, Technologies of Information and telecommunications, Tunisie*.
- [107] Fitriana, G. F. (2014). Handwriting Digit Recognition using United Moment Invariant Feature Extraction and Self Organizing Maps. *The 3rd ICT International Student Project Conference, IEEE*. 43-46.
- [108] Kan, C. & Srinath, M.D. (2002). Invariant character recognition with Zernike and orthogonal Fourier–Mellin moments. *Pattern recognition*, 35(1), pp. 143-154.
- [109] Knerr, S., Personnaz, L. & Dreyfus, G. (1992). Handwritten digit recognition by neural networks with single-layer training. *Transactions on neural networks, IEEE*. 3(6): 962-968.
- [110] Mukesh V., Mamatha E., Sravan Kumar, M.V. & Dasthagiri R. (2017). Back-Propagation network to hand written digit recognition. *International Journal of Computer & Mathematical Sciences IJCMS*. 6(6): 200-208.
- [111] Liu, C. L., Nakashima, K., Sako, H. & Fujisawa, H. (2002). Handwritten digit recognition using state-of-the-art techniques. *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, IEEE*, 320-325.
- [112] Singh, P. K., Sarkar, R. & Nasipuri, M. (2016). A study of moment based features on handwritten digit recognition. *Applied Computational Intelligence and Soft Computing*. 16:1-17
- [113] Majumder, S., von der Malsburg, C., Richhariya, A. & Bhanot, S. (2018). Handwritten Digit Recognition by Elastic Matching. *Journal of Computers*, 13(9): 1067-1074.
- [114] Seewald, A.K. (2005). Digits-a dataset for handwritten digit recognition. *Austrian Research Institut for Artificial Intelligence Technical Report, Vienna (Austria)*. TR-2005-27
- [115] Hou, J., Zeng, H., Cai, L., Zhu, J., Cao, J. & Hou, J. (2017). Handwritten numeral recognition using multi-task learning. *International Symposium on Intelligent Signal Processing and Communication Systems, IEEE*. 155-158.
- [116] Ramzan, M., Khan, H.U., Akhtar, W., Zamir, A., Awan, S.M., Ilyas, M. & Mahmood, A. (2018). A Survey on using Neural Network based Algorithms for Hand Written Digit Recognition. *Environment*, 9(9): 519-528.
- [117] Shamim, S.M., Miah, M.B.A., Sarker, A., Rana, M. & Al Jobair, A. (2018). Handwritten Digit Recognition Using Machine Learning Algorithms. *Global Journal of Computer Science and Technology*, 18(1): 17-23.
- [118] Mozafari, M., Ganjtabesh, M., Nowzari-Dalini, A., Thorpe, S.J. & Masquelier, T. (2018). Combining STDP and reward-modulated STDP in deep convolutional spiking neural networks for digit recognition. *arXiv preprint arXiv:1804.00227*.

- [119] Gil, A.M., Costa Filho, C.F.F. & Costa, M.G.F. (2014). Handwritten digit recognition using SVM binary classifiers and unbalanced decision trees. *International Conference Image Analysis and Recognition*, 246-255.
- [120] Shima, Y., Nakashima, Y. & Yasuda, M. (2018). Handwritten Digits Recognition by Using CNN Alex-Net Pre-trained for Large-scale Object Image Dataset. *Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing*, 36-40.
- [121] Kulkarni, S.R. & Rajendran, B. (2018). Spiking neural networks for handwritten digit recognition - Supervised learning and network optimization. *Neural Networks*, 103: 118-127.
- [122] Sensoy, M., Kaplan, L. & Kandemir, M. (2018). Evidential deep learning to quantify classification uncertainty. *Advances in Neural Information Processing Systems*, 3179-3189.
- [123] Gal, Y. & Ghahramani, Z. (2015). Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*.
- [124] Kingma, D. P., Salimans, T. & Welling, M. (2015). Variational dropout and the local reparameterization trick. *Advances in Neural Information Processing Systems*, 2575-2583.
- [125] Lakshminarayanan, B., Pritzel, A. & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 6402-6413.
- [126] Molchanov, D., Ashukha, A. & Vetrov, D. (2017). Variational dropout sparsifies deep neural networks. *Proceedings of the 34th International Conference on Machine Learning*, 70: 2498-2507.
- [127] Sato, I., Nishimura, H. & Yokoi, K. (2015). Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*.
- [128] McDonnell, M. D. & Vladusich, T. (2015, July). Enhanced image classification with a fast-learning shallow convolutional neural network. *International Joint Conference on Neural Networks, IEEE*. 1-7.
- [129] CireşAn, D., Meier, U., Masci, J. & Schmidhuber, J. (2012). Multi-column deep neural network for traffic sign classification. *Neural networks*, 32: 333-338.
- [130] Lauer, F., Suen, C.Y. & Bloch, G. (2007). A trainable feature extractor for handwritten digit recognition. *Pattern Recognition*, 40(6): 1816-1824.
- [131] Niu, X.X. & Suen, C.Y. (2012). A novel hybrid CNN-SVM classifier for recognizing handwritten digits. *Pattern Recognition*, 45(4): 1318-1325.
- [132] McDonnell, M.D. & Vladusich, T. (2015). Enhanced image classification with a fast-learning shallow convolutional neural network. *International Joint Conference on Neural Networks, IEEE*. 1-7.
- [133] Lee, C.Y., Gallagher, P.W. & Tu, Z. (2016). Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. *Artificial Intelligence and Statistics*, 464-472.
- [134] Ciresan, D.C., Meier, U., Gambardella, L.M. & Schmidhuber, J. (2010). Deep big simple neural nets excel on handwritten digit recognition. *Neural computation*, 22(12): 3207-3220.

- [135] Chang, J.R. & Chen, Y. S. (2015). Batch-normalized maxout network in network. *arXiv preprint arXiv:1511.02583*.
- [136] Graham, B. (2015). Fractional max-pooling. arxiv: cs. *arXiv preprint arXiv:1412.6071*.
- [137] Liao, Z. & Carneiro, G. (2016). On the importance of normalisation layers in deep learning with piecewise linear activation units. *Winter Conference on Applications of Computer Vision, IEEE*. 1-8.
- [138] LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278-2324.
- [139] Brader, J.M., Senn, W. & Fusi, S. (2007). Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. *Neural computation*, 19(11): 2881-2912.
- [140] Querlioz, D., Bichler, O., Dollfus, P. & Gamrat, C. (2013). Immunity to device variations in a spiking neural network with memristive nanodevices. *Transactions on Nanotechnology, IEEE*. 12(3): 288-295.
- [141] Beyeler, M., Dutt, N.D. & Krichmar, J. L. (2013). Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule. *Neural Networks*, 48, pp. 109-124.
- [142] Wan, L., Zeiler, M., Zhang, S., Le Cun, Y. & Fergus, R. (2013). Regularization of neural networks using dropconnect. *International conference on machine learning*, 1058-1066.
- [143] Diehl, P.U. & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 9: 99.
- [144] Sato, I., Nishimura, H. & Yokoi, K. (2015). Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*.
- [145] Liang, M. & Hu, X. (2015). Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3367-3375.
- [146] Liao, Z. & Carneiro, G. (2015). Competitive multi-scale convolution. *arXiv preprint arXiv:1511.05635*.
- [147] Kheradpisheh, S.R., Ganjtabesh, M., Thorpe, S.J. & Masquelier, T. (2018). STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99: 56-67.
- [148] Kowsari, K., Heidarysafa, M., Brown, D.E., Meimandi, K.J. & Barnes, L.E. (2018). Rmdl: Random multimodel deep learning for classification. *Proceedings of the 2nd International Conference on Information System and Data Mining*, 19-28.
- [149] Kussul, E. & Baidyk, T. (2004). Improved method of handwritten digit recognition tested on MNIST database. *Image and Vision Computing*, 22(12): 971-981.
- [150] Salakhutdinov, R. & Hinton, G. (2007). Learning a nonlinear embedding by preserving class neighbourhood structure. *Artificial Intelligence and Statistics*, 412-419.

- [151] Chmielnicki, W. & Stapor, K. (2010). Investigation of normalization techniques and their impact on a recognition rate in handwritten numeral recognition. *Schedae Informaticae*, 19 : 53-78.
- [152] Chouaib, H. (2011). *Sélection de caractéristiques : méthodes et applications*. Thèse de doctorat, Descartes University, Paris, France.
- [153] Man, Z., Lee, K., Wang, D., Cao, Z. & Khoo, S. (2013). An optimal weight learning machine for handwritten digit image recognition. *Signal Processing*, 93(6): 1624-1638.
- [154] Zekovich, S. & Tuba, M. (2013). Hu moments based handwritten digits recognition algorithm. *Recent Advances in Knowledge Engineering and Systems Science*, pp. 98-103.
- [155] Ebrahimzadeh, R. & Jampour, M. (2014). Efficient handwritten digit recognition based on histogram of oriented gradients and SVM. *International Journal of Computer Applications*, 104(9), pp. 10-13.
- [156] Tuba, E. & Bacanin, N. (2015). An algorithm for handwritten digit recognition using projection histograms and SVM classifier. *The 23rd Telecommunications Forum Telfor, IEEE*. 464-467.
- [157] Khan, H. A. (2017). Mcs hog features and svm based handwritten digit recognition system. *Journal of Intelligent Learning Systems and Applications*, 9(02): 21-33.
- [158] Arbain, N. A., Azmi, M. S., Muda, A. K., Muda, N. A. & Radzid, A. R. (2018). Offline Handwritten Digit Recognition Using Triangle Geometry Properties. *International Journal of Computer Information Systems and Industrial Management Applications*, 10: 87-97.
- [159] LeCun, Y., Cortes, C. & Burges, C. (2010). MNIST handwritten digit database [base de données]. <http://yann.lecun.com/exdb/mnist/>, consulté en 2015.
- [160] Heutte, L., Paquet, T., Moreau, J.V., Lecourtier, Y. & Olivier, C. (1998). A structural/statistical feature based vector for handwritten character recognition. *Pattern recognition letters*.19(7): 629-641.
- [161] Tsai, W.H. & Fu, K. S. (1980). Attributed grammar-a tool for combining syntactic and statistical approaches to pattern recognition. *Transactions on Systems, Man, and Cybernetics, IEEE*. 10(12): 873-885.
- [162] Xu, L., Krzyzak, A. & Suen, C.Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *Transactions on Systems, Man, and Cybernetics*, 22(3): 418-435.
- [163] Cao, J., Shridhar, M. & Ahmadi, M. (1995). Fusion of classifiers with fuzzy integrals. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1: 108-111.
- [164] Beiraghi, S., Ahmadi, M., Shridhar, M. & Ahmed, M.S. (2000). Application of fuzzy integrals in fusion of classifiers for low error rate handwritten numerals recognition. *Proceedings 15th International Conference on Pattern Recognition, IEEE*. 2, pp. 487-490.
- [165] Domeniconi, C. & Yan, B. (2004). Nearest neighbor ensemble. *Proceedings of the 17th International Conference on Pattern Recognition*, 1:228-231.

- [166] Bay, S.D. (1999). Nearest neighbor classification from multiple feature subsets. *Intelligent data analysis*, 3(3): 191-209.
- [167] Prudent, Y. & Ennaji, A. (2004). Les K Plus proches classifieurs. *Conférence Internationale Francophone sur l'Écrit et le Document (CIFED 04)*.
- [168] Shrivastava, S.K. & Mewada, P. (2011). ACO Based feature subset selection for multiple k-nearest neighbor classifiers. *International Journal on Computer Science and Engineering*, 3(5): 1831-1838.
- [169] Bellet, A., Habrard, A., Morvant, E. & Sebban, M. (2013). Vote de majorité a priori contraint pour la classification binaire : spécification au cas des plus proches voisins. *Conférence sur l'Apprentissage Automatique*.
- [170] Bao, Y. & Ishii, N. (2002). Combining multiple K-nearest neighbor classifiers for text classification by reducts. *International Conference on Discovery Science. Springer, Berlin, Heidelberg*, 340-347.
- [171] Bao, Y., Ishii, N. & Du, X. (2004). Combining multiple k-nearest neighbor classifiers using different distance functions. In *International Conference on Intelligent Data Engineering and Automated Learning, Springer, Berlin, Heidelberg*, pp. 634-641.
- [172] Belongie, S. J., Malik, J. & Puzicha, J. (2001). Matching Shapes. The 8th International Conference on Computer Vision (ICCV) IEEE. 1: 454-461.
- [173] Miranda Dos Santos, E. (2008). *Static and dynamic overproduction and selection of classifier ensembles with genetic algorithms*. Thèse de Doctorat, École Supérieure de Technologie, Quebec.
- [174] Ko, A.H.R. (2007). *Static and dynamic selection of ensemble of classifiers*. Thèse de Doctorat, École Supérieure de Technologie, Quebec.
- [175] Cavalin, P.R., Sabourin, R. & Suen, C.Y. (2013). Dynamic selection approaches for multiple classifier systems. *Neural Computing and Applications*, 22(3-4), pp. 673-688.
- [176] Gader, P., Forester, B., Ganzberger, M., Gillies, A., Mitchell, B., Whalen, M. & Yocum, T. (1991). Recognition of handwritten digits using template and model matching. *Pattern Recognition*, 24(5): 421-431.
- [177] Oliveira, L. S., Sabourin, R., Bortolozzi, F. & Suen, C. Y. (2003). Feature selection for ensembles: A hierarchical multi-objective genetic algorithm approach. *The 7th International Conference on Document Analysis and Recognition, (ICDAR) IEEE*. 676-680.
- [178] Prasad, B. K. & Sanyal, G. (2018). Novel features and a cascaded classifier based Arabic numerals recognition system. *Multidimensional Systems and Signal Processing*, 29(1): 321-338.
- [179] Gorgevik, D. and Cakmakov, D. (2004) An efficient three-stage classifier for handwritten digit recognition. *The 17th International Conference on Pattern Recognition*, 4: 507-510.
- [180] Cruz, R. M., Cavalcanti, G. D. & Ren, T. I. (2010). Handwritten digit recognition using multiple feature extraction techniques and classifier ensemble. *The 17th International Conference on Systems, Signals and Image Processing*. 215-218.

- [181] Zhao, H. & Liu, H. (2018). Algebraic fusion of multiple classifiers for handwritten digits recognition. *International Conference on Wavelet Analysis and Pattern Recognition, IEEE*, 250-255.
- [182] Friedman, M. & Kandel, A. (1999). *Introduction to pattern recognition: statistical, structural, neural and fuzzy logic approaches*. World Scientific Publishing Company.
- [183] Milgram, M. (1993). *Reconnaissance des formes : Methodes numeriques et connexionistes*. Ed. Armand Colin.
- [184] Kunt, M., Coray, G., Granlund, G., Haton, J., Ingold, R. & Kocher, M. (2000). *Reconnaissance des formes et analyse de scènes*. Presses Polytechniques et Universitaires Romandes.
- [185] Fukunaga, K. (2013). *Introduction to statistical pattern recognition*. Academic press.
- [186] Fu, K.S. (1980). Recent developments in pattern recognition. *IEEE Transactions on Computers*, 100(10): 845-854.
- [187] Fu, K.S. (1974). *Syntactic methods in pattern recognition*. Academic Press.
- [188] Ponti Jr, M.P. (2011). Combining classifiers: from the creation of ensembles to the decision fusion. *The 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials, IEEE*, 1-10
- [189] Duda, R.O., Hart, P.E. & Stork, D.G. (2012). *Pattern classification*. John Wiley & Sons.
- [190] Bow, S.T. (2002). *Pattern recognition and image preprocessing*. Marcel Dekker.
- [191] Burel, G. (2001). *Introduction au traitement d'images : simulation sous Matlab*. Hermès.
- [192] James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). *An introduction to statistical learning*. springer.
- [193] Rgonzalez, R.C., Woods, R.E. & Eddins, S.L. (2009). *Digital image processing using Matlab*. Gatesmark Publishing.
- [194] Young, I.T., Gerbrands, J.J. & Van Vliet, L.J. (1998). *Fundamentals of image processing*. Delft University of Technology, Netherlands.
- [195] Pratt, W.K. (2001). *Digital image processing*. John Wiley & Sons.
- [196] Acharya, T. & Ray, A.K. (2005). *Image processing: principles and applications*. John Wiley & Sons.
- [197] Govindan, V.K. & Shivaprasad, A.P. (1990). Character recognition-a review. *Pattern Recognition*, 23(7): 671-683.
- [198] Suen, C.Y., Berthod, M. & Mori, S. (1980). Automatic recognition of hand printed characters - the state of the art. *Proceedings of the IEEE*, 68(4): 469-487.
- [199] Bakis, R., Herbst, N.M. & Nagy, G. (1968). An experimental study of machine recognition of hand-printed numerals. *Transactions on systems science and cybernetics, IEEE*. 4(2): 119-132.

- [200] Knoll, A.L. (1969). Experiments with "characteristic loci" for recognition of hand printed characters. *IEEE Transactions on Computers*, 100(4): 366-372.
- [201] Harvey A.C. & Qin C. (1995). Loci Features for Binary Pattern Recognition, *Proceedings Digital Image Computing : Techniques and Applications*, 540 - 546.
- [202] Impedovo, D. & Pirlo, G. (2014). Zoning methods for handwritten character recognition: A survey. *Pattern Recognition*, 47(3): 969-981.,
- [203] Tabbone, S. & Wendling, L. (2003). Binary shape normalization using the Radon transform. In *International Conference on Discrete Geometry for Computer Imagery*, Springer, Berlin, Heidelberg. 184-193.
- [204] Jain, A. K. & Zongker, D. (1997). Representation and recognition of handwritten digits using deformable templates. *Transactions on Pattern Analysis and Machine Intelligence*, IEEE. 19(12): 1386-1390.
- [205] Labusch, K., Barth, E. & Martinetz, T. (2008). Simple method for high-performance digit recognition based on sparse coding. *Transactions on Neural Networks*, IEEE. 19(11): 1985-1989.
- [206] Lu, Y., Schlosser, S. & Janeczko, M. (1993). Fourier descriptors and handwritten digit recognition. *Machine Vision and Applications*, 6(1): 25-34.
- [207] Reiss, T.H. (1991). The revised fundamental theorem of moment invariants. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 8, pp. 830-834.
- [208] Theodoridis, S. & Koutroumbas, K. (2003). Feature selection. *Pattern recognition*, 5 : 261-322.
- [209] Gosselin, B. (1996). Application de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits. Thèse de doctorat, Faculté polytechnique de Mons, Belgique.
- [210] Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3: 1157-1182.
- [211] Theodoridis, S., Pikrakis, A., Koutroumbas, K. & Cavouras, D. (2010). *Introduction to pattern recognition: a Matlab approach*. Academic Press.
- [212] Jain, A. & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *transactions on pattern analysis and machine intelligence*, IEEE. 19(2): 153-158.
- [213] Abe, S. (2005). *Support vector machines for pattern classification*. Springer.
- [214] Arrivault, D. (2006). *Apport des graphes dans la reconnaissance non-contrainte de caractères manuscrits anciens*. Thèse de doctorat, Université de Poitiers, France.
- [215] Miclet, L. & Haton, J. P. (1984). *Méthodes structurelles pour la reconnaissance des formes*. Eyrolles.
- [216] Pal, S.K. & Pal, A. (2001). *Pattern recognition: from classical to modern approaches*. World Scientific.

- [217] Kotsiantis, S.B., Zaharakis, I. & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160 : 3-24.
- [218] You, K.C. & Fu, K.S. (1979). A syntactic approach to shape recognition using attributed grammars. *IEEE transactions on systems, man, and cybernetics*, 9(6): 334-345.
- [219] Fu, K.S. (1979). Recent advances in syntactic pattern recognition. In *Pattern Formation by Dynamic Systems and Pattern Recognition*, 5: 176-185.
- [220] Fu, K.S. (1977). *Syntactic Pattern Recognition, applications*. Springer, Verlag.
- [221] Fu, K.S. (1986). A Step Towards Unification of Syntactic and Statistical Pattern Recognition, *transactions on pattern analysis and machine intelligence, IEEE*. 8(3): 398-404.
- [222] Kittler, J.O.S.E.F., Hatef, M., Duin, R.P. & Matas, J. (1998). On Combining Classifiers. *Trans. Pattern Analysis and Machine Intelligence, IEEE*. 20(3): 226-239.
- [223] Kuncheva, L.I. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.
- [224] Britto Jr, A.S., Sabourin, R. & Oliveira, L.E. (2014). Dynamic selection of classifiers - a comprehensive review. *Pattern Recognition*, 47(11), pp. 3665-3680.
- [225] Rahman, A.F.R. & Fairhurst, M.C. (2003). Multiple classifier decision combination strategies for character recognition: A review. *Document Analysis and Recognition*, 5(4): 166-194.
- [226] Woźniak, M., Graña, M. & Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16: 3-17.
- [227] Rahman, A.F.R. & Fairhurst, M.C. (2000). Multiple expert classification: a new methodology for parallel decision fusion. *International Journal on Document Analysis and Recognition*, 3(1): 40-55.
- [228] Ko, A.H.R. & Sabourin, R. (2013). Single Classifier-based Multiple Classification Scheme for weak classifiers: An experimental comparison. *Expert Systems with Applications*, 40(9): 3606-3622.
- [229] Impedovo, S. & Salzo, A. (2000). A new evaluation method for expert combination in multi-expert system designing. In *International Workshop on Multiple Classifier Systems*. 230-239.
- [230] Connolly, J.F., Granger, E. & Sabourin, R. (2013). Evolving Classifier Ensembles using Dynamic Multi objective Swarm Intelligence. *The 2nd International Conference on Pattern Recognition Applications and Methods, ICPRAM*, Barcelona, Spain. 1: 206-215,
- [231] Kittler, J. (1998). Combining classifiers: A theoretical framework. *Pattern analysis and Applications*, 1(1), pp. 18-27.
- [232] Ruta, D. & Gabrys, B. (2000). An overview of classifier fusion methods. *Computing and Information systems*, 7(1): 1-10.

- [233] Zadeh, L.A. (1986). A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. *Artificial Intelligence Magazine*, 7(2): 85-85.
- [234] Shipp, C.A. & Kuncheva, L.I. (2002). Relationships between combination methods and measures of diversity in combining classifiers. *Information fusion*, 3(2): 135-148.
- [235] Kuncheva, L.I. (2002). Switching between selection and fusion in combining classifiers: An experiment. *Transactions on Systems, Man, and Cybernetics, Part B, IEEE*. 32(2): 146-156.
- [236] Parker, J.R. (2001). Rank and response combination from confusion matrix data. *Information fusion*, 2(2): 113-120.
- [237] Kittler, J. & Alkoot, F. M. (2003). Sum versus vote fusion in multiple classifier systems. *transactions on pattern analysis and machine intelligence, IEEE*. 25(1): 110-115.
- [238] <https://cedar.buffalo.edu/Databases/index.html>. Consulté en 2018.
- [239] <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps>. Consulté en 2018.
- [240] Semeion Handwritten Digit. (2008). Disponible dans : UCI (Université de Californie à Irvine) Machine Learning Repository.
- [241] SVHN (Base de données), <http://ufldl.stanford.edu/housenumbers/>. Consulté en 2018.
- [242] Derdour K. (2010). *Reconnaissance de formes du chiffre arabe imprimé : Application au code à barre d'un produit*. Thèse de Magistère, université de Batna, Algérie.
- [243] Hu, M.K. (1962). Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2) : 179-187.
- [244] Milgram, J. (2007). *Contribution à l'intégration des machines à vecteurs de support au sein des systèmes de reconnaissance de formes : application à la lecture automatique de l'écriture manuscrite*. Thèse de Doctorat, École Supérieure de Technologie, Québec.
- [245] Derdour K. & Mouss, H. (2014)., Combinaison des classifieurs KPPV pour la classification des chiffres imprimés. *Conférence international sur l'intelligence artificielle et les technologies de l'information*.
- [246] Derdour K. & Mouss, H. (2015). Digit recognition using multiple classifiers, *IEEE international symposium on programming and systems*.
- [247] Derdour, K. & Mouss, H. (2015). Printed digits recognition using multiple multi Layer perceptron classifiers and Hu moments. *Symposium on Complex Systems and Intelligent Computing*.
- [248] Derdour K. & Mouss, H. (2019); multiple classifiers and invariant features extraction for digit recognition, *International Journal of Computer and Electrical Engineering*, 11(1): 41-52.
- [249] Babu, U.R., Venkateswarlu, Y. & Chintha, A. K. (2014, February). Handwritten digit recognition using K-nearest neighbour classifier. *World Congress on Computing and Communication Technologies, IEEE*. 60-65.

- [250] Suen, C.Y. & Tan, J. (2005). Analysis of errors of handwritten digits made by a multitude of classifiers. *Pattern Recognition Letters*, 26(3): 369-379.
- [251] Beghdadi, H.A. Snouci, M. (2007). *Réseaux de neurones théorie et pratique*. Office des publications universitaires.
- [252] Slavik, P. & Govindaraju, V. (2001). Equivalence of different methods for slant and skew corrections in word recognition applications. *transactions on pattern analysis and machine intelligence, IEEE*. 23(3): 323-326.
- [253] Nagendraprasad, M.V., Gupta, A. & Feliberti, V. (1995). Correction of slant in handwritten numerals for automated document processing. *Engineering Applications of Artificial Intelligence*, 8(4): 469-472.
- [254] Taira, E., Uchida, S. & Sakoe, H. (2004). Non uniform slant correction for handwritten word recognition. *IEICE TRANSACTIONS on Information and Systems*, 87(5): 1247-1253.
- [255] Blumenstein, M., Cheng, C.K. & Liu, X.Y. (2002). New preprocessing techniques for handwritten word recognition. *Proceedings of the 2nd International Conference on Visualization, Imaging and Image Processing (VIIP), ACTA Press, Calgary*, 480-484.
- [256] Crettez, J.P. & Lorette, G. (1998). *Reconnaissance de l'écriture manuscrite*. Ed. Techniques Ingénieur.
- [257] Oprean, C., Likforman-Sulem, L. & Mokbel, C. (2013). Handwritten word preprocessing for database adaptation. *Document Recognition and Retrieval XX*, 8658: 865808.
- [258] Kimura, F. & Shridhar, M. (1991). Handwritten numerical recognition based on multiple algorithms. *Pattern recognition*, 24(10): 969-983.
- [259] Hart, P.E. & Duda, R.O. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1): pp. 11-15.
- [260] Khorashadizadeh, S. & Latif, A. (2016). Arabic/Farsi Handwritten Digit Recognition using Histogram of Oriented Gradient and Chain Code Histogram. *International Arab Journal of Information Technology*, 13(4): 367-374.
- [261] Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. *IEEE Computer Society In international Conference on computer vision & Pattern Recognition*. 1: 886-893
- [262] Lord, E.A. & Wilson, C.B. (1986). *The mathematical description of shape and form*. John wiley & sons.
- [263] Micó, L. & Oncina, J. (1998). Comparison of fast nearest neighbour classifiers for handwritten character recognition. *Pattern Recognition Letters*, 19(3-4): 351-356.
- [264] Moreno-Seco, F., Micó, L. & Oncina, J. (2003, June). Extending fast nearest neighbour search algorithms for approximate k-NN classification. *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, Berlin, Heidelberg. 589-597.
- [265] Roli, F., Giacinto, G. & Vernazza, G. (2001). Methods for designing multiple classifier systems. In *International Workshop on Multiple Classifier Systems*, 78-87.

- [266] De Stefano, C., Della Cioppa, A. & Marcelli, A. (2003). Exploiting reliability for dynamic selection of classifiers by means of genetic algorithms. *Proceedings of the Seventh International Conference on Document Analysis and Recognition, IEEE*. 671.
- [267] Arrivault, D. (2006). *Apport des graphes dans la reconnaissance non-contrainte de caractères manuscrits anciens*. Thèse de doctorat, Université Poitiers, France.
- [268] Zhang, D. & Lu, G. (2002). Shape-based image retrieval using generic Fourier descriptor. *Signal Processing: Image Communication*, 17(10): 825-848.
- [269] Kim, H.K., Kim, J.D., Sim, D.G. & Oh, D.I. (2000). A modified Zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval. *International Conference on Multimedia and Expo, Proceedings. Latest Advances in the Fast Changing World of Multimedia*, 1: 307-310.
- [270] Duin, R.P. & Tax, D.M. (2000). Experiments with classifier combining rules. *International Workshop on Multiple Classifier Systems*, 16-29.
- [271] Horata, P., Chiewchanwattana, S. & Sunat, K. (2011). A comparative study of pseudo-inverse computing for the extreme learning machine classifier. *IEEE In The 3rd International Conference on Data Mining and Intelligent Information Technology Applications*, 40-45.

ANNEXE 1 : Glossaire sur la RdF

Les prétraitements

- Réduction du bruit : Le bruit est une valeur découlant habituellement de la reproduction, de la numérisation et de la transmission de l'image originale (l'apparition des points noirs isolés-empâtements-, ou des trous –points blancs ou manquement-). Le bruit c'est une erreur aléatoire dans la valeur de pixel qu'il ne peut pas toujours être entièrement supprimé. Pour éliminer le bruit il faut définir les composantes connexes d'une image, si une composante comporte moins d'un nombre prédéfini de pixels (seuil), doit l'effacer [251].
- Correction d'inclinaison et de pente : Il existe des écritures droites, d'autres penchées à gauche et d'autres penchées à droite ou encore un mélange des deux. Le redressement de la pente permet l'uniformisation de l'écriture. Ce prétraitement consiste à évaluer l'angle d'inclinaison locale d'un caractère ou l'angle d'inclinaison moyenne des lettres à l'intérieur d'un mot et à effectuer une rotation en sens inverse [28, 252, 253, 254, 255, 256].
- Normalisation : Les formes, peuvent être placées à différents endroits (à l'intérieur d'une image), ou/et pivotées, ainsi la taille varie énormément (dans les caractères) selon les styles d'écriture. Il est donc souvent nécessaire d'effectuer une opération de normalisation avant d'entamer toute extraction des paramètres caractéristiques afin de standardiser la taille et centrer la forme [251].
- Intensité de pixel [257] : Consiste à décomposer l'image en plusieurs zones. L'intensité moyenne (pourcentage des cellules noirs) de chaque zone est retenue et les densités dans chaque zone devront être normalisées (en les divisant par exemple par la surface de la zone) puisque les formes ne sont pas tous de même taille.
- Amincissement (Schématisation) : C'est la suppression de tous les pixels redondants jusqu'à ce que l'épaisseur ne soit plus que d'un point. L'amincissement est en fait une érosion avec conservation de la connexité. Il peut servir à améliorer les contours d'objets fins, mais sa grande utilisation sera la squelettisation. Les méthodes efficaces d'amincissement enlève essentiellement des couches de pixels du contour de l'image d'origine de la ligne tout en évitant le découpage et la fragmentation des lignes. Différentes méthodes d'amincissements sont présentées dans [192].

- Lissage : Après correction de l'inclinaison, une opération de lissage est effectuée pour régulariser les contours(bordure) de l'image et supprimer les petits bruits parasites [17]. Il consiste à parcourir l'image (la rétine) avec une fenêtre, la valeur d'un pixel est modifiée si elle est contraire à la majorité de celles qui l'entourent. Le lissage rend l'image plus arrondie.

Les paramètres caractéristiques structurelles

- Intersections avec des droites : c'est le nombre et la position des points de l'intersection des droites (ligne) verticale et horizontales avec la forme) [160].
- Segments et arcs de cercle : consiste à diviser successivement le contour (décomposition) en segments de droite de plus en plus petits. Puis à réunir les segments de droites adjacents (composition de traits élémentaires –segments arcs de cercle) juxtaposé ou superposé, et déplacer les sommets de polygone [184].
- Contours : la description numérique du contour fait intervenir deux étapes qui sont respectivement la *quantification* et le *codage*. Parmi les techniques qui permettent de définir la position relative d'un ensemble de points à savoir le code de chaîne [84], cette technique exprime les positions des points(pixels) connexe de connectivité 4 ou 8 par des directions qui relie deux points.
- **Squelette** : Le squelette est un graphe qui caractérise la forme d'un objet binaire et qui peut permettre de le reconstituer (transformation réversible). Il sera très utile pour faire de la reconnaissance d'objets.
- **Cavités** : la définition de 4 directions cardinales dans l'image : Ouest vers la gauche, nord vers le haut, etc., donne 5 type de cavités : Ouest, Est, Nord, Sud et Centrale [191]. Les cavités devront être normalisées (en les divisant par la surface totale des cavités).
- **Les points d'extrémité et de jonctions** : ce sont les point finaux (Points terminaux) d'une forme obtenue par un rupture d'un chainage des points du contour. L'existence de l'intersection donne les points de jonctions.
- **Les extrema** : correspond à la position et au nombre d'extrema. Il y a quatre types d'extrema : haut, bas, est, ouest. [160].
- **Code de chaîne** : exprime la position d'un point par rapport à un autre point connexe de connectivité 4 ou 8 au moyen d'un mot binaire à 2 ou à 3 bits. Le nombre spécifique indiqué par le mot binaire correspond à l'une des directions qui relie les deux points [184].

Les paramètres caractéristiques statistiques

- **Codage rétinien (naturel ou la rétine) :** Les pixels de la matrice (image) sont utilisés comme des paramètres caractéristiques, Cette méthode est très utilisée dans certaines approches neuronales. Mais ne tolère absolument pas les transformations géométrique (rotation, translation, échelle) ou les déformations [183].
- **Surface :** c'est la somme de l'ensemble des pixels appartenant à la forme diviser par le nombre total de pixels de l'image.
- **Zoning et densité des pixels :** Consiste à calculer la densité de pixels dans différentes zones (zoning) de l'image. Les zones sont obtenues en découpant l'image horizontalement en m zones horizontales d'égale hauteur et verticalement en n zones verticales d'égale largeur on obtient des zones de taille égale ; le nombre de pixels noirs dans chaque zone forme alors les composantes du vecteur de caractéristiques qui ne sont pas invariantes par illumination [96]. Une étude détaillée sur les méthodes de zonage est présentée dans l'article [202]
- **Les histogrammes de projections :** Leur principe est de sommer le nombre de pixels noirs de chacune des lignes (respectivement des colonnes) de l'image binaire de la forme. La projection horizontale (respectivement verticale) est le nombre de pixel de chaque ligne (respectivement colonne). Elles peuvent être indépendantes de l'échelle en divisant par le nombre total de pixels imprimés en caractère d'image. Toutefois., Les projections d'histogrammes sont très sensibles à la rotation, et dans une certaine mesure, la variabilité dans le style d'écriture. [96, 160]. La projection peut s'effectue en calculant la somme des pixels selon une direction donnée, particulièrement sur les diagonales 45° et 135° [251].
- **Profils :** L'extraction des profils gauche et droit sont obtenues par détermination sur un certain nombre de lignes –réparties uniformément sur la hauteur de l'image – la distance entre le bord gauche (respectivement droit) de la forme et le premier pixel noir rencontré sur cette ligne ; l'ensemble de ces distances définit un profil gauche (respectivement droit) de la forme [160, 258].
- **Transformé de Hough :** Hough a proposé une procédure intéressante et efficace pour détecter et calculer des lignes dans les images. R. Duda et al [259] proposent une paramétrisation par les coordonnées polaires qui simplifie davantage le calcul. Il montre également comment cette méthode peut être utilisée pour détecter d'autre formes (transformée généralisée de Hough). Cette méthode est détaillée dans [190].
- **Un histogramme de gradient orienté (HOG) :** Présente des informations sur la forme dans une image par la distribution gradienne de densité et de contour (edge). L'idée de

base consiste à diviser l'image à des régions, puis des blocs régulièrement répartis sur l'image sont construits par accumulations des régions [260, 261]. La méthode s'est montrée particulièrement efficace pour la détection de personnes.

- **Descripteurs de Fourier :** Les descripteurs de Fourier correspondent à une suite de nombres dont le but est de caractériser au mieux la forme à décrire. Il existe principalement deux méthodes *angulaire* et *complexe* qui permettent de le définir [184]. Leurs applications dans la description de formes géométriques sont présentées dans la référence [262].
- **Les moments :** Les moments fournissent un codage compact, doté de bonnes propriétés d'invariance théoriques. En fait, il y a plusieurs types de moments : les moments géométriques, les moments complexes, les moments de Hu, de Zernike, de Tchebychev etc.

Les classifieurs statistiques

- **Classifieur par distance euclidienne minimum :** La méthode de décision proposée consiste à attribuer à la forme à reconnaître, la classe dont la distance de la forme au centre de cette classe est minimale. Il faut donc disposer, préalablement à toute reconnaissance, du centre de chacune des classes : c'est l'objet de la phase d'apprentissage.
- **KPPV :** Cette méthode de décision (K plus proches voisins) [183, 208, 192] nécessite de disposer, préalablement à toute reconnaissance, de l'ensemble des points formant la base d'apprentissage pour calculer la distance du point à classer, représentant la forme à reconnaître, à tous les points de la base d'apprentissage et affecter à ce point la classe la plus représentée parmi ses K plus proches voisins. La règle KPPV est un classifieur non paramétrique très répandue, dans laquelle la probabilité a posteriori est estimée à partir de la fréquence des plus proches voisins du modèle inconnu. Le problème avec cette méthode est le coût de calcul élevé lorsque la classification est effectuée. Pour surmonter un tel problème, certains chercheurs ont proposé des méthodes plus rapides de KPPV. Une comparaison des classifieurs PPV rapides pour la reconnaissance de l'écriture est donnée dans [263, 264]
- **Classifieur Bayésienne :** Classifieur Bayésienne [208] est un classifieur paramétrique, il attribue une forme à une classe avec la probabilité a posteriori maximale. Les classes prototypes sont utilisés à la phase d'apprentissage pour estimer la fonction de densité de la probabilité conditionnelle de classe pour un vecteur de caractéristiques.

Il est basé sur l'application du théorème de Bayes avec l'hypothèse de fortes indépendances (naïves) entre les caractéristiques. Le classifieur Naïve Bayésien suppose que la valeur de la caractéristique particulière n'est pas liée à l'absence ou à la présence de toute autre caractéristique, compte tenu de la variable de classe. Le principal avantage du classifieur naïve Bayésien est son temps de calcul court pour l'apprentissage.

- **Pseudo inverse** : Cette méthode nécessite d'abord une phase d'apprentissage pour calculer une matrice W avec des lignes qui sont le nombre des classes et les colonnes le nombre de paramètres caractéristiques. Cette matrice minimise l'erreur quadratique moyenne. [191]

$$W = SE^T(EE^T)^{-1}$$

E : matrice des vecteurs d'entrée, S : la matrice des vecteurs de sortie.

Cette formule fait apparaître $E^T(EE^T)^{-1}$, qui est la pseudo-inverse de E d'où le nom de la méthode [191]. D'autres méthodes pour calculer le pseudo-inverse sont présentées dans [271].

- **Analyse discriminante linéaire (LDA)** : Elle consiste à trouver une combinaison linéaire de caractéristiques séparant ou caractérisant deux ou plusieurs classes d'objets. Le principe consiste à approximer les limites entre les classes dans l'espace des caractéristiques m -dimensionnelles en plaçant des hyperplans, parmi des régions correspondant aux différentes classes, en supposant qu'elles soient séparables linéairement.
- **Fenêtres (Noyaux) de Parzen** : c'est une méthode Bayésienne non paramétrique, c'est une approche classique consiste à estimer la densité de probabilité sur un voisinage du vecteur considéré [214].
- **Classifieur discriminant polynomial** : il assigne une forme à une classe avec la valeur discriminante maximale qui est calculée par un polynôme dans les composantes d'un vecteur de caractéristiques. Les classes modèles sont implicitement représentées par les coefficients dans le polynôme.
- **Modèles de Markov Cachés (MMC)** : sont devenus un outil très populaire en reconnaissance d'écriture manuscrite (Hidden Markov Model (HMM)), est un processus doublement stochastique, avec un processus sous-jacent stochastique non observable (d'où le mot caché), mais pouvant être observé par un autre processus stochastique qui produit la séquence d'observations. Les HMM se sont avérés être l'un des outils les plus

puissants pour modéliser la parole et, plus tard, pour une grande variété d'autres signaux du monde réel. Ces modèles probabilistes offrent de nombreuses propriétés souhaitables pour la modélisation de caractères ou de mots. L'une des propriétés les plus importantes est l'existence d'algorithmes efficaces pour former automatiquement les modèles sans qu'il soit nécessaire d'étiqueter les données prédéfinies [17].

- **Le raisonnement flou** : c'est est une technique qui utilise des éléments d'ensemble flous pour décrire les similitudes entre les caractéristiques. Les éléments d'ensemble flous donnent un résultat plus réaliste lorsqu'il n'y a pas de connaissance a priori des données et que, par conséquent, les probabilités ne peuvent pas être calculées.
- **SVM** : Support Vector Machine (SVM) [208] est basé sur la théorie de l'apprentissage statistique et l'optimisation de la programmation quadratique. Un SVM est de base un classifieur binaire, et plusieurs SVM peuvent être combinés pour former un système permettant une classification multi-classe. Au cours des dernières années, SVM a reçu une attention croissante dans la communauté de l'apprentissage automatique en raison de ses excellentes performances en matière de généralisation.

Les machines à vecteurs de support, leurs variantes et leurs extensions, souvent appelé méthodes basées sur le noyau (ou simplement les méthodes du noyau), ont été étudiés de manière approfondie et appliqués à divers problèmes d'approximation et de classification.

- **Perceptron multicouche PMC** : Les réseaux de neurones ont connu un essor important grâce à l'algorithme de rétro propagation du gradient. Ce classifieur PMC (MLP Multilayer perceptron) a trouvé application dans beaucoup de domaines tels que la reconnaissance de caractères, la reconnaissance de visages, la classification d'expressions de gènes, etc. Il est capable d'inférer n'importe quelle fonction de décision non linéaire moyennant une seule couche de neurones cachées et des fonctions d'activation sigmoïdales [7].
- **Arbre de décision** : chaque nœud correspond à un test sur la valeur d'un ou plusieurs attributs ; et chaque branche partant d'un nœud correspond à une ou plusieurs valeurs de ce test ; à chaque feuille est associée une valeur de l'attribut cible [217].

Elle est basée sur une série de tests sur les caractéristiques. C'est une idée simple, à travers l'ordre de ces tests, la réponse à l'un d'eux indique quel prochain test, que cet objet doit être soumis. Le principe est l'organisation de tous les tests possibles sous forme d'arborescence. Les classes sont notées les feuilles de cet arbre et le test associe un nœud.

Les classifieurs structurelles

- **Distances entre chaîne** : c'est le moyen le plus simple pour définir une classe de formes. Le codage d'une forme par les primitives nous donne des mots ont des longueurs différentes, cela ne nous permis pas de calculer la distance directement. Pour produire une distance raisonnable entre deux mots x et y, il faut faire des transformations élémentaires (supprimer un symbole, insérer un symbole, changer un symbole) permettant de passer du mot x au mot y en affectant un cout à chaque transformation et en totalisant les coûts des transformations utilisées.
- **Mise en correspondance combinatoire** : c'est de mettre en correspondance les primitives d'une forme à reconnaître avec celles d'un modèle ou prototype. Deux stratégies de mise en correspondance combinatoire présentée dans : les graphe d'appariement et les arbres d'interprétation. [183]
- **Relaxation** : c'est une méthode numérique itérative permettant de diminuer progressivement l'ambiguïté d'une correspondance entre deux ensembles finis appelés respectivement ensemble des objets et ensemble des étiquettes. Cette méthode repose sur l'idée d'un appariement entre les éléments de la forme et ceux des modèles.
- **Mise en coïncidence** : elle se présente comme une approche intégrée où segmentation et interprétation sont réalisées simultanément. Deux techniques sont traités dans [183] : les prototypes déformables, la transformation de Hough généralisée.
- **Structure syntaxique régulière** : c'est un outil permettant d'engendrer une famille de phrases sur un alphabet : les grammaire formelles (régulières), ainsi que les automates finis.
- **Structure syntaxique algébrique** : la grammaire algébrique engendre le langage algébrique appelée aussi context-free, libre du contexte, hors contexte, de Chomsky ou BNF (Backus-Naur Form).
- **Structure d'arbre** : c'est un modèle très utile en RdF, leur propriété mathématique autorisent de plus l'utilisation d'algorithmes efficaces pour leur manipulation.

ANNEXE 2 : Glossaire sur les MCS

Sélection Statique

La sélection statique de classifieurs peut être divisé en 3 groupes :

- **Surproduire et choisir** : cette technique consiste à générer des classifieurs différents en se basant sur les méthodes de création d'ensembles (phase de surproduction) et à choisir ensuite le groupe de classifieurs (phase de sélection) dont la combinaison produit le meilleur résultat. Il existe plusieurs méthodes pour créer des ensembles de classifieurs : utiliser des caractéristiques différentes.

La génération aléatoire des sous-ensembles de caractéristiques par la méthode Random subspaces (RSS). Utiliser des sous-bases de données différentes (Bagging, Boosting). Cette technique est le sujet de la thèse [173], elle porte sur l'étude de la phase de sélection utilisant les algorithmes génétiques (AG), qui constitue le défi le plus important dans la stratégie de surproduction et choix. Six méthodes de cette technique sont décrites et comparées dans [265].

- **Regrouper et extraire** : elle consiste à regrouper les classifieurs en sous-ensembles qui diffèrent selon le niveau de corrélation et à extraire ensuite un classifieur représentatif de chaque sous-ensemble tout en essayant d'avoir un compromis entre diversité d'erreur et performance de l'ensemble final. Deux façons de regrouper les classifieurs :
 - La combinaison (par vote à la majorité) des classifieurs du même groupe donne un taux élevé d'erreur. Ceci permet de dire que les classifieurs de chaque groupe ont un degré faible de complémentarité (sont très corrélés).
 - La combinaison de classifieurs de groupes différents permet d'avoir un pourcentage d'erreur faible.
- **Rechercher et sélectionner** : elle se base sur le même principe des méthodes de sélection de caractéristiques utilisées en RdF statistiques. Des méthodes heuristiques fondées sur le principe de la recherche locale sont utilisées pour l'étape de la recherche comme les algorithmes génétiques [266] et la recherche séquentielle (recherche en avant, recherche en arrière, ...). Utilisation d'un algorithme de recherche et un ou plusieurs critères de sélection (comme la performance de la combinaison).

Sélection Dynamique

Zouari [5] a divisé les méthodes de sélection dynamique en deux groupes :

- **Sélection locale** : elle consiste à diviser l'espace de caractéristiques en régions, estimer la performance locale des classifieurs dans chaque région et choisir le meilleur classifieur par région. Si une forme à reconnaître est attribuée à une région, alors le classifieur représentatif de cette même région prend la décision finale.
- **Sélection par pondération** : elle consiste à utiliser un sélecteur (enseignant) généralement de type réseau de neurones dont le rôle est d'attribuer des poids aux classifieurs (le poids le plus élevé au meilleur classifieur). L'approche la plus utilisée est la mixture d'experts. Celle-ci se base sur le principe "diviser et conquérir". "Diviser" est la distribution de la base d'apprentissage sur les classifieurs disponibles. Alors "Conquérir" concerne la compétition entre les classifieurs.

Complémentarité et Indépendance des classifieurs

Dans la combinaison de classifieurs, plusieurs termes comme orthogonalité, complémentarité, non-corrélation, et indépendance ont été souvent utilisés pour décrire la qualité désirée de la relation entre les classifieurs à combiner. Ces termes indiquent la tendance des classifieurs à produire des décisions différentes. Cependant, ils manquent de précision et de définition. Pour différencier entre ces termes, l'analyse bibliographique présentée par Zouari [5] dans le domaine de l'évaluation des performances des méthodes de combinaison nous a permis de constater que :

- ✓ La notion de corrélation et de dépendance ne sont pas identiques. Quand deux variables X et Y sont décorrélées, on peut dire qu'elles n'ont pas de dépendance linéaire, mais on ne peut pas dire qu'elles sont indépendantes puisqu'elles peuvent avoir des dépendances non linéaires. Mathématiquement, deux variables sont indépendantes si et seulement si :

$$p(x, y) = p(x) * p(y)$$

x et y sont non corrélées si et seulement si

$$E(x, y) = E(x) * E(y)$$

Où E() indique l'espérance mathématique.

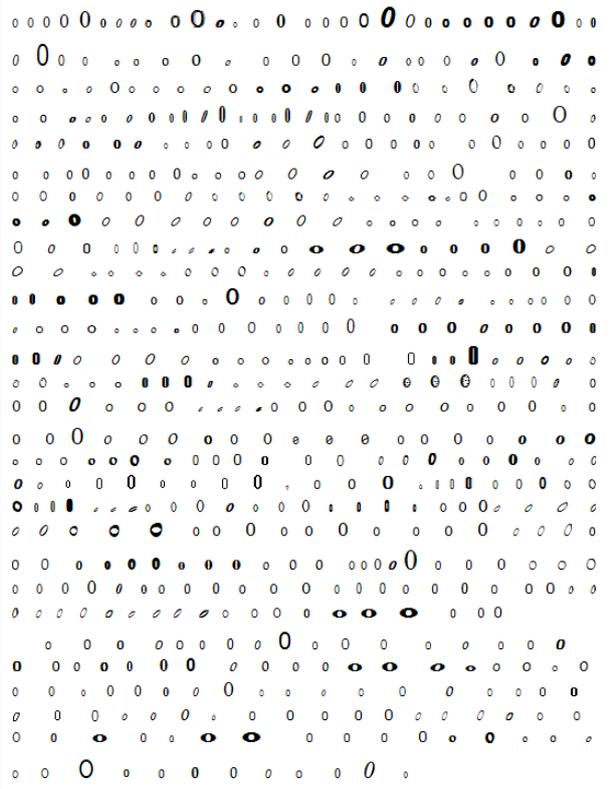
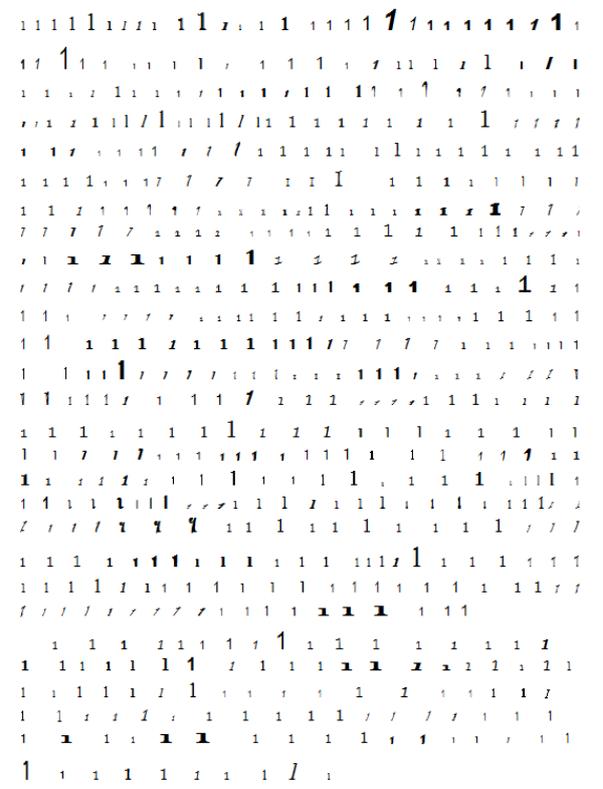
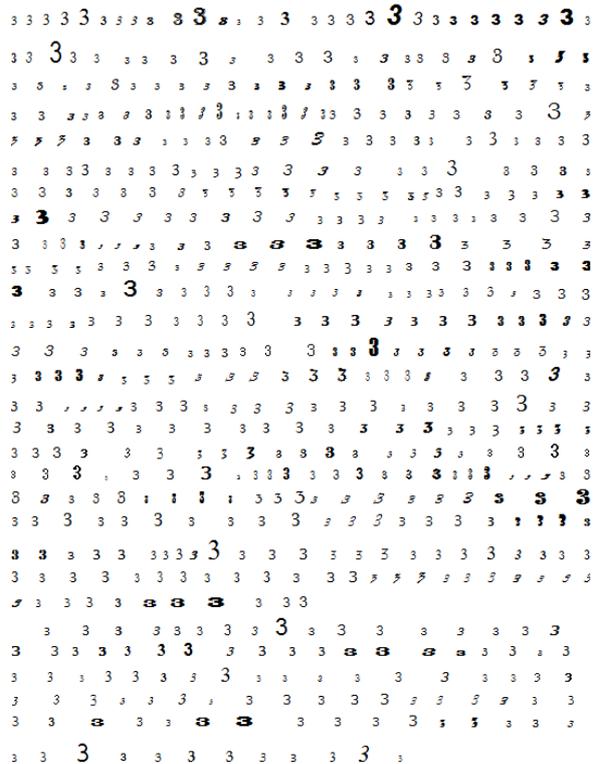
L'indépendance de deux variables implique leur non corrélation, la réciproque n'est pas vraie

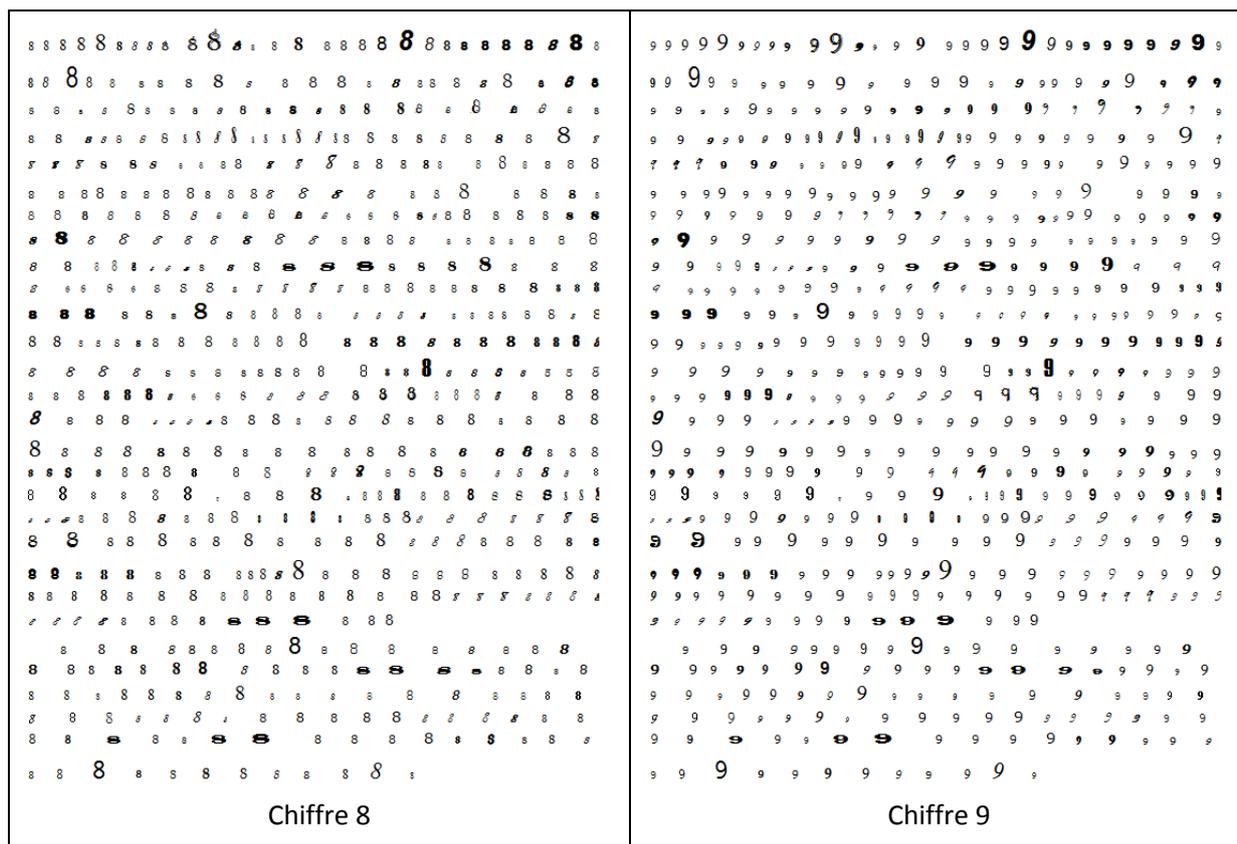
- ✓ Néanmoins en pratique, l'indépendance entre les classifieurs est supposée satisfaite en utilisant par exemple des caractéristiques ou des bases d'apprentissage différentes.

- ✓ L'indépendance des classifieurs n'est pas nécessaire pour obtenir une complémentarité entre ces classifieurs. L'information mutuelle est utilisée pour quantifier la complémentarité qui peut exister entre les classifieurs. L'indépendance des classifieurs n'implique pas une amélioration de performance de leur combinaison. Certains chercheurs ont prouvé qu'avec des classifieurs dépendants, il est possible d'avoir des résultats intéressants et même meilleurs qu'avec des classifieurs indépendants.
- ✓ Une étude présentée dans [33] a démontré que la combinaison de classifieurs indépendants peut entraîner une amélioration considérable de la performance. Cependant, En théorie, les classifieurs indépendants peuvent ne pas être le meilleur choix.
- ✓ Le concept de la dominance d'un classifieur dans l'ensemble est plus important que celui de l'indépendance permettant de décrire les conditions d'amélioration de performance de la combinaison. En effet, pour que la combinaison améliore la performance de chacun des classifieurs, il ne faut pas avoir un classifieur dominant dans l'ensemble à combiner.
- ✓ Les experts du domaine sont maintenant convaincus que la combinaison de classifieurs ne peut être efficace que si les classifieurs individuels sont différents et pas nécessairement indépendants. Ainsi, l'étude du rôle de la diversité, appelée aussi complémentarité ou orthogonalité, dans l'amélioration des performances d'un ensemble de classifieurs est identifiée comme une direction de recherche importante, mais pas nouvelle. Les chercheurs suggèrent que "*il est toujours préférable de forcer les différents classifieurs à utiliser des méthodologies aussi diverses que possible*". Par exemple, si on dispose de trois versions de classifieurs A, B et C alors il est plus intéressant de construire un système comme ABC que AAB, BBA, ACC, CCA, etc. Comme ils ont montré qu'il est préférable de combiner des classifieurs moins performants mais qui font des erreurs différentes plutôt que des classifieurs très performants mais qui présentent des erreurs identiques. En effet, plus les classifieurs sont divers (c'est à dire qu'ils exhibent une "*diversité distincte*" meilleur sera le résultat de leur combinaison.
- ✓ Dans la littérature, Les résultats obtenus ont montré qu'aucune des mesures de diversité ne permet de donner une idée claire sur la relation entre performance du vote à la majorité et diversité entre les classifieurs.
- ✓ Les systèmes de combinaison ne produisent pas toujours de meilleures performances que celles des classifieurs à combiner. Ceci est dû au fait que les membres de tels systèmes sont fortement corrélés et tendent à produire les mêmes erreurs simultanément.

ANNEXE 3 : Les bases de données et les méthodes implémentées

Les images de la base BCI_1

 <p style="text-align: center;">Chiffre 0</p>	 <p style="text-align: center;">Chiffre 1</p>
 <p style="text-align: center;">Chiffre 2</p>	 <p style="text-align: center;">Chiffre 3</p>



Les polices utilisées

ae_Kayrawan	Felix Titling	Microsoft Sans Serif
Algerian	Footlight MT Light	MingLiU
Angsana New	Forte	Mistral
Arial	Franklin Gothic Book	Modern No. 20
Arial black	Franklin Gothic Demi	Monotype Corsiva
Arial narrow	Franklin Gothic Demi Cond	MS Gothic
Arial Rounded MT Bold	Franklin Gothic Heavy	MS Mincho
Arial Unicode MS	Franklin Gothic Medium	MS Reference Sans Serif
Baskerville Old Face	Franklin Gothic Medium Cond	Niagara Engraved
Bauhaus 93	Freestyle Script	Nina
Bell MT	French Script MT	Old English Text MT
Berlin Sans FB	Garamond	Onyx
Berlin Sans FB Demi	Georgia	Palace Script MT
Bernard MT Condensed	Gigi	Perpetua
Blackadder ITC	Gill Sans MT	Perpetua Titling MT
Book Antiqua	Gill Sans MT Condensed	Playbill

Bodoni MT	Gill Sans MT Ext Condensed Bold	Poor Richard
Bradley Hand ITC	Gill Sans Ultra Bold	Pristina
Britannic Bold	Gill Sans Ultra Bold Condensed	Rage Italic
Browallia New	Gloucester MT Extra Condensed	Ravie
Brush Script MT	Goudy Old Style	Rockwell
Calibri	Goudy Stout	Rockwell Condensed
Californian FB	GulimChe	Rockwell Extra Bold
Calisto MT	Haettenschweiler	Script MT Bold
Cambria	Harlow Solid Italic	Segoe UI
Cambria Math	Harrington	Showcard Gothic
Candara	High Tower Text	Stencil
Castellar	Impact	Sylfaen
Centaur	Imprint MT Shadow	Symbol
Century	Informal Roman	Technical
Century Gothic	Jokerman	Tahoma
Century Schoolbook	Juice ITC	Tempus Sans ITC
Chiller	Kartika	Times New Roman
Colonna MT	Kristen ITC	Trebuchet MS
Comic Sans MS	Kunstler Script	Tw Cen MT
Constan	Lucida Bright	Tw Cen MT Condensed
Cooper Black	Lucida Calligraphy	Tw Cen MT Condensed Extra
Copperplate Gothic Bold	Lucida Console	Bold
Copperplate Gothic Light	Lucida Fax	Verdana
Corbel	Lucida Handwriting	Viner Hand ITC
Cordia New	Lucida Sans	Vivaldi
Courier New	Lucida Sans Typewriter	Vladimir Script
Curlz MT	Lucida Sans Unicode	Vrinda
Edwardian Script ITC	Magneto	Wide Latin
Elephant	Maiandra GD	ZWAdobeF
Engravers MT	Matura MT Script Capitals	
Eras Bold ITC		
Eras Demi ITC		

Les moments de Zernike

Les moments de Zernike A_{nm} sont définis comme des polynômes complexes qui forment un ensemble orthogonal complet du cercle unité $x^2+y^2 \leq 1$. Ils correspondent à la projection de la forme $f(x, y)$ sur une base ZP de fonctions orthogonales $V_{nm}(x, y)$ et se défini par [151, 267, 268, 269] :

$$ZP = \{V_{nm}(x, y) \mid x^2 + y^2 \leq 1\}$$

Où le polynôme complexe V_{nm} est défini par :

$$V_{nm}(x, y) = R_{nm}(x, y) e^{(im \arctan(x/y))}$$

Tel que :

$$n - |m| \text{ soit pair et } |m| \leq n, i = \sqrt{-1}, n \geq 0, m : \text{ nombre entier positif ou négatif.}$$

Le rapport polynomial est défini comme suit :

$$R_{nm}(x, y) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2}-s\right)! \left(\frac{n-|m|}{2}-s\right)!} (x^2 + y^2)^{(n-2s)/2}$$

Les moments de Zernike sont les projections de la fonction d'image $f(x,y)$ dans la base orthogonale de fonction $V_{nm}(x,y)$. Le moment Zernike, d'ordre n et de répétition m , est un nombre complexe défini par :

$$A_{nm} = \frac{n+1}{\pi} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) [V_{nm}(x, y)]$$

L'ordre des moments possède de grande influence sur la conservation de l'information angulaire. Plus l'ordre est élevé et plus les variations angulaires décrites sont fines.

Les moments de Zernike sont bien connus pour être invariants à la rotation. Une normalisation d'image est nécessaire pour rendre ces moments invariants à la translation et au facteur d'échelle. En outre les moments de Zernike ont une caractéristique intéressante, l'image original peut être reconstruit à partir de ces moments.

Les moments de Zernike sont liées à l'habitude des moments μ_{ij} (ou les moments normalisés η_{ij}), une fois ces moments sont calculés.

Dans ce qui suit, nous donnons l'expression de neuf moments calculés à partir des moments centrés. Notons que les deux premiers moments sont indépendants (valeurs fixes). On peut donc utiliser 47 variables au lieu de 49 :

$$A_{00} = \mu_{00}/\pi = 1/\pi$$

$$A_{11} = A_{1-1} = 0$$

$$A_{22} = 3(\mu_{02} - \mu_{20} - 2i\mu_{11})/\pi$$

$$A_{20} = 3(2\mu_{20} + 2\mu_{02} - 1)/\pi$$

$$A_{33} = 4[\mu_{03} - 3\mu_{21} - i(\mu_{30} - 3\mu_{12})]/\pi$$

$$A_{31} = 12[\mu_{03} - \mu_{21} - i(\mu_{30} - \mu_{12})]/\pi$$

$$A_{44} = 5[\mu_{40} - 6\mu_{22} - \mu_{04} + 4i(\mu_{31} - \mu_{13})]/\pi$$

$$A_{42} = 5\{4(\mu_{04} - \mu_{40}) + 3(\mu_{20} - \mu_{02}) - 2i[4(\mu_{31} - \mu_{13}) - 3\mu_{11}]\}/\pi$$

$$A_{40} = 5[6(\mu_{40} + 2\mu_{22} + \mu_{04}) - 6(\mu_{20} + \mu_{02}) + 1]/\pi$$

Fonctions de distance

L'utilité des métriques et des distances est de pouvoir comparer les ressemblances et différences entre deux vecteurs. Cette opération est importante, par exemple, dans le domaine de la classification. Il est plus probable que deux vecteurs semblables soient dans une même classe que deux. L'utilité des métriques et des distances est de pouvoir comparer les ressemblances et différences entre deux vecteurs. Cette opération est importante, par exemple, dans le domaine de la classification. Il est plus probable que deux vecteurs semblables soient dans une même classe que deux vecteurs dissemblables. De fait, l'utilisation de métriques pour mesurer des distances est presque toujours une étape essentielle à la classification automatique.

- La distance la plus connue est la distance **Euclidienne**,

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Où x et y sont deux vecteurs (une de la base de donnée et l'autre à classifier), n est le nombre d'attributs dans l'application (dimension de vecteur). La distance Euclidienne est souvent utilisée au carré.

- La distance de **Minkowsky**.

$$d(x, y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

La distance Euclidien, n'est qu'un cas particulier pour $p=2$ de la distance de Minkowsky.

- Pour $p = 1$, on obtient la distance de **Manhattan** (aussi appelée distance « **city-block** » ou métrique absolue).

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

- Pour $p=\infty$, la distance de **Chebychev** (aussi appelée distance « Queen-wise » ou encore métrique maximum).

$$d(x, y) = \max_{i=1}^n |x_i - y_i|$$

- Distance de **Mahalanobis**

$$d(x, y) = \sqrt{(x - y)C^{-1}(x - y)'}$$

Où C, est la matrice de covariance.

- Distance de **cosinus**

$$d(x, y) = 1 - \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i^2}}$$

- Distance de **corrélation**

$$d(x, y) = 1 - \frac{\sum_{i=1}^n (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^n (x_i - \bar{x}_i)^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y}_i)^2}}$$

Où $\bar{x}_i = 1/n \sum_j x_{ij}$, et $\bar{y}_i = 1/n \sum_j y_{ij}$

- Distance **Euclidien normalisé** : c'est la distance euclidienne normalisée en divisant sur la variance de chaque variable.

$$d(x, y) = \sqrt{\sum_{i=1}^n \frac{(x_i - y_i)^2}{s_k^2}}$$

Les méthodes d'évaluation des performances

La méthode de Resubstitution :

Elle consiste à entraîner et tester le classificateur avec la même base de données (toute la base de données). Dans cette méthode le nombre de données d'entraînement (apprentissage) est maximal ce qui permet de mieux construire le classificateur. Dans ce cas l'erreur de classification est optimiste à cause d'un phénomène de sur-apprentissage (en anglais « over fitting ») qui signifie la perte de pouvoir de prédiction sur de nouveaux échantillons.

La méthode du tout-sauf-un « Leave-One-Out » :

Elle consiste à entraîner le classificateur avec toutes les données, sauf une, qui sert au test. Cette méthode est coûteuse en volume de calculs parce qu'il est nécessaire de répéter l'entraînement de classificateur autant de fois qu'il y a d'échantillon dans la base de données, pour calculer le taux d'erreur moyen.

La méthode du « Holdout » :

Le plus souvent, c'est la méthode la plus utilisée, elle consiste à diviser aléatoirement l'ensemble total de données en un ensemble d'entraînement et un ensemble de tests, ne comportant aucun élément en commun. L'inconvénient de cette méthode est que le nombre de données disponibles pour l'apprentissage du classifieur se trouve réduit.

La méthode de Rotation (k validation croisée) :

Il s'agit d'un compromis des deux méthodes précédentes. L'ensemble total de données est divisé aléatoirement en k partitions distinctes, dont une sert à tester le classifieur entraîné à l'aide des (k-1) autres. Une rotation des partitions est ensuite réalisée, et de nouveaux tests effectués. Lorsque chacune des k partitions a été utilisée pour tester le classifieur, le taux d'erreur moyen peut être calculé.

Comportement des méthodes de combinaison

L'évaluation de performances des méthodes de combinaison a eu un intérêt considérable ces dernières années.

Il existe plusieurs facteurs qui peuvent influencer sur le comportement des méthodes de combinaison. Ils sont liés directement aux classifieurs à utiliser et aux problèmes de classification à traiter. Les principaux sont :

- nombre de classes et le nombre de classifieurs
- performances des classifieurs (taux de reconnaissance, d'erreur, de rejet)
- nature des caractéristiques et la dimension de l'espace de caractéristiques
- niveau de diversité entre les classifieurs
- taille des bases d'apprentissage, de test et d'évaluation

C'est surtout en fonction des performances des classifieurs (taux d'erreurs et de reconnaissance en particulier) que les chercheurs ont analysé le comportement des méthodes de combinaison. Une analyse bibliographique sur l'influence de ces facteurs sur le comportement des méthodes de combinaison est présentée dans [5]. Les chercheurs évaluent les méthodes de combinaison dans des conditions différentes (nombre de classifieurs différents, problèmes différents, performances différentes). La comparaison des résultats est difficile à réaliser et reste étroitement dépendante des applications traitées. Donc les travaux existants montrent qu'aucune méthode de combinaison n'est actuellement capable de donner de meilleures performances dans tous les problèmes de reconnaissance.

ANNEXE 4 : Les images des chiffres de différentes langues

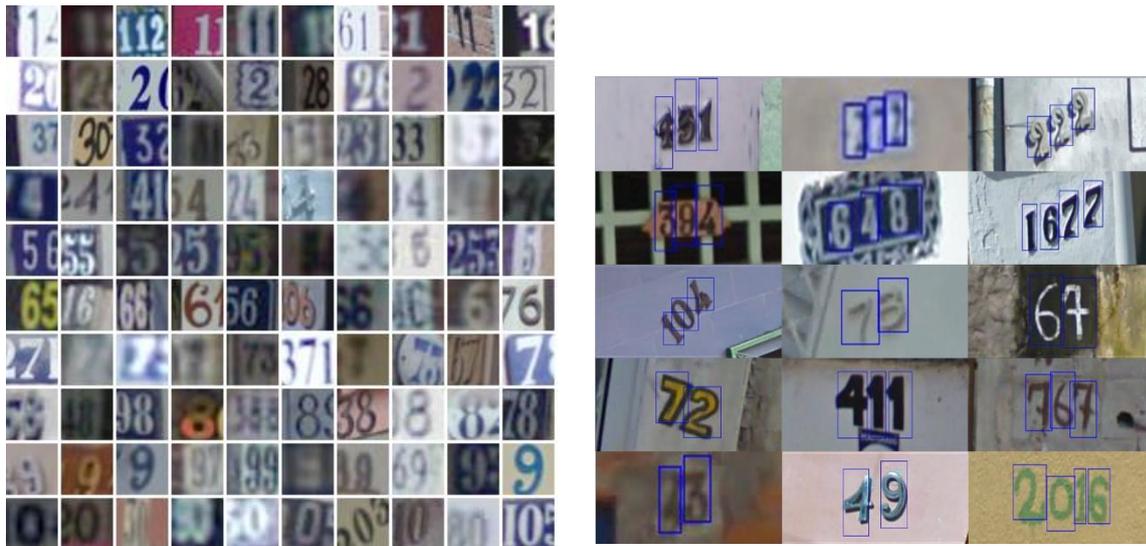
Chiffres de la base CENPARMI

0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

Chiffres de la base CEDAR

0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9

Chiffres de la base SVHN



Chiffres de différentes langues

- Chiffres de la langue Marathi

०	०	०	०	०	०	०	०	०	०	०
१	१	१	१	१	१	१	१	१	१	१
२	२	२	२	२	२	२	२	२	२	२
३	३	३	३	३	३	३	३	३	३	३
४	४	४	४	४	४	४	४	४	४	४
५	५	५	५	५	५	५	५	५	५	५
६	६	६	६	६	६	६	६	६	६	६
७	७	७	७	७	७	७	७	७	७	७
८	८	८	८	८	८	८	८	८	८	८
९	९	९	९	९	९	९	९	९	९	९

- Chiffres de la langue MODI

०	१	२	३	४	५	६	७	८	९
0	1	2	3	4	5	6	7	8	9

- Chiffres de la langue Farsi (Persien, Urdu)

0 1 2 3 4 5 6 7 8 9
 ۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹

- Chiffres de la langue Bangla

One	১	১	১	১
Two	২	২	২	২
Three	৩	৩	৩	৩
Four	৪	৪	৪	৪
Five	৫	৫	৫	৫
Six	৬	৬	৬	৬
Seven	৭	৭	৭	৭
Eight	৮	৮	৮	৮
Nine	৯	৯	৯	৯
Zero	০	০	০	০

- Chiffres de la langue Hindi (arabe orientale, arabic)

۰	۱	۲	۳	۴	۵	۶	۷	۸	۹
0	1	2	3	4	5	6	7	8	9

- Chiffres de la langue Gurmukhi

0	੦	5	੫
1	੧	6	੬
2	੨	7	੭
3	੩	8	੮
4	੪	9	੯

- Chiffres de la langue Assamese

੦ ੧ ੨ ੩ ੪ ੫ ੬ ੭ ੮ ੯

- Chiffres de la langue Devanagari

0	१	२	३	४	५	६	७	८	९
०	१	२	३	४	५	६	७	८	९
०	१	२	३	४	५	६	७	८	९
०	१	२	३	४	५	६	७	८	९
०	१	२	३	४	५	६	७	८	९
०	१	२	३	४	५	६	७	८	९
०	१	२	३	४	५	६	७	८	९
०	१	२	३	४	५	६	७	८	९
०	१	२	३	४	५	६	७	८	९
०	१	२	३	४	५	६	७	८	९

- Chiffres de la langue Telugu

౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯

- Chiffres de la langue Arabic (Roman, anglais)

0 1 2 3 4 5 6 7 8 9

౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯
౦	౧	౨	౩	౪	౫	౬	౭	౮	౯

- Chiffres de la langue Odia

୦ ୧ ୨ ୩ ୪ ୫ ୬ ୭ ୮ ୯
0 1 2 3 4 5 6 7 8 9