



THESE

Présentée à la Faculté des Sciences et Techniques, pour
obtenir le Grade de **Docteur en Informatique**.

Systemes de Reconnaissance des Formes à Base de la Géométrie Riemannienne : Application à la Reconnaissance des Caractères Tifinagh et des Objets 3D

Par

Omar BENCHAREF

Encadrant: Pr. Mohamed FAKIR
Co-Encadrant: Pr. Brahim MINAOUI
Co-Encadrant: Pr. Belaid BOUIKHALENE

Soutenue le 01/12/2012, devant les membres de jury :

- Président :** Professeur Ahmed ELKHADIMI de l'INPT de Rabat.
Rapporteur : Professeur Abdelaziz KREOUILE de l'ENSIAS de Rabat.
Rapporteur : Professeur Saïd SAFI, de la FP de Beni Mellal.
Examineur : Professeur Brahim MINAOUI, de la FST de Beni Mellal.
Examineur : Professeur Mohamed FAKIR de la FST de Beni Mellal.
Examineur : Professeur Belaid BOUIKHALENE de la FP de Beni Mellal.



Université Sultan Moulay Slimane
Faculté des Sciences et Techniques
Département informatique
Beni Mellal



THÈSE

Présentée à la Faculté des Sciences et Techniques,
pour l'obtention du Grade de Docteur en Informatique.

Systemes de Reconnaissance des Formes à Base de la Géométrie Riemannienne : Application à la Reconnaissance des Caractères Tifinagh et des Objets 3D

Par
Omar BENCHAREF

Encadrant: M. FAKIR

Co-encadrant : B. MINAOUI

Co-encadrant : B. BOUIKHALENE

Résumé:

La présente thèse se consacre au sujet de la reconnaissance automatique des caractères et des objets réels. L'objectif principal étant de développer des algorithmes, ou plus précisément, des systèmes de reconnaissance de formes. La difficulté de ce travail réside particulièrement dans le chevauchement des apparences multiples qu'un seul objet pourrait prendre.

Les systèmes que nous proposons mettent en avant trois nouvelles méthodes d'extraction inspirées essentiellement de la géométrie riemannienne (les descripteurs métriques, les descripteurs géodésiques et les descripteurs de courbures). Ces systèmes intègrent une multitude de classificateurs hybrides basés sur les réseaux de neurones, les réseaux neuroflous et les SVM. Nous avons également établi une étude comparative entre les méthodes d'extraction couramment utilisées (Zernike, Hu, Fourier, Radon, etc.) et nous avons étudié l'influence des paramètres primaires (à savoir : le changement de la base de couleur, la luminance, l'existence de bruit et des déformations spatiales) sur le rendement des réseaux de neurones.

Mots clés: Reconnaissance des formes, Extraction, Géométrie riemannienne, SVM, Réseaux Neuro-Flou, Réseaux de neurones.

Abstract

This thesis focuses on the topic of automatic recognition of characters and real objects. The main objective is to develop algorithms, or more precisely, shape recognition systems. The difficulty of this task lies particularly in the overlap of multiple appearances a single object could have.

The systems we are proposing put forward three new methods of extraction inspired essentially from the Riemannian geometry (metric descriptors, geodesic descriptors and curvature descriptors). These systems integrate several hybrid classifiers based on neural networks, neuro-fuzzy networks and SVM. We also established a comparative study of the extraction methods commonly used (Zernik, Hu, Fourier, Radon, etc.) and we studied the influence of primary parameters (change of color base, luminosity, existence of noise and space deformations) on neural networks performance.

ملخص:

أصبحت الرؤية بواسطة الحاسوب واحدة من مجالات البحث الأكثر جاذبية، نظرا لأهمية وفائدة التطبيقات التي تنتجها. هذه الرسالة تهدف إلى تطوير خوارزميات ونظم التعرف على الأحرف والأجسام الحقيقية. التحدي الأبرز في هذا العمل يكمن في المظاهر المتعددة التي يمكن أن يتخذها شكل واحد. بشكل عام، يتكون نظام التعرف من ثلاث مراحل: أولا التحضير، تليها مرحلة استخراج السمات وأخيرا مرحلة التصنيف. النظم المقترحة تقدم ثلاث طرق جديدة لاستخراج السمات، مستوحاة أساسا من هندسة ريمان (واصف متري، واصف الانحرافات و واصف جيوديسي)، وهذه النظم تدمج العديد من المصنفات الهجينة المعتمدة أساسا على الشبكات العصبية الاصطناعية، وشبكات neurofuzzy و SVM. كما أنجزنا مقارنة بين الطرق الكلاسيكية لاستخراج السمات (زيرنيك، هو، فورييه، الرادون (...). كما قمنا بدراسة تأثير كل من تغيير اللون، الإنارة ووجود التشوهات على أداء المصنفات المدروسة.

Je dédis ce travail

À MES TRÈS CHERS PARENTS

Les mots sont vains pour vous exprimer mon amour et mon affection et rien ne saurait vous remercier de tant d'années de sacrifices et d'abnégation.

Que Dieu vous garde longtemps parmi nous.

A MES TRES CHERS FRERES ET A MA SŒUR

NADIRA, AMINE ,SAAD

En témoignage de ma grande affection et de ma reconnaissance pour votre soutien qui n'a jamais fait défaut.

Ce travail est aussi le vôtre.

À TOUTE MA FAMILLE

Particulièrement à ma femme SELOUA et à mon fils AALAE qui m'ont soutenu durant mes années d'études.

Sans oublier ma tante Mina et mes grands parents.

Que ce travail soit le témoignage de l'amour et de la tendresse que je vous porte.

À LA MEMOIRE DE MA TANTE MALIKA

Qui a été pour moi une seconde mère et qui m'a poussé avec toute sa générosité, sa force et son courage à aller en avant.

Remerciements

À NOTRE DIRECTEUR DE THESE

LE PROFESSEUR MOHAMED FAKIR

Vous nous avez fait l'honneur de nous confier ce travail tout en nous guidons de vos conseils précieux.

Vous nous avez toujours accueillis avec une grande amabilité et avec simplicité.

Qu'il nous soit permis de vous témoigner notre vive reconnaissance et notre grand respect.

A NOTRE ENCADRANT DE THESE

LE PROFESSEUR BRAHIM MINAOUI

Nous vous remercions de l'amabilité de votre accueil et de l'honneur que vous nous faites en acceptant d'encadrer mon travail.

Votre compétence, votre sagesse et vos grandes qualités humaines nous ont toujours impressionnés.

Veillez trouver ici le témoignage de notre profonde gratitude.

À NOTRE ENCADRANT DE THESE

LE PROFESSEUR BELAID BOUIKHALENE

L'amabilité et la gentillesse avec laquelle vous nous avez reçus nous ont beaucoup touchés.

Qu'il nous soit permis de vous témoigner de notre profond respect et de notre grande Reconnaissance

À MESSIEURS LES MEMBRES DE JURY

Vous nous faites grand honneur en acceptant de juger notre travail.

Nous avons toujours admiré vos qualités humaines et professionnelles.

Veillez en trouver ici un hommage respectueux.

À

Saad, Amine, Jawad, housna et Seloua, qui m'ont aidé dans la rédaction de cette thèse, je vous remercie de tout mon cœur.

Table des matières

| | |
|--|----------------------|
| LISTE DES FIGURES | 4 |
| LISTE DES TABLEAUX | 6 |
| INTRODUCTION..... | 7 |
| CHAPITRE1 : | ÉTAT DE L'ART |
| 1. STRUCTURE D'UN SYSTEME DE RECONNAISSANCE DE FORMES (RF) | 10 |
| 1.1. PHASE D'ACQUISITION..... | 10 |
| 1.2. PHASE DE PRETRAITEMENT..... | 11 |
| 1.3. PHASE D'EXTRACTION DES CARACTERISTIQUES..... | 12 |
| 1.4. CLASSIFICATION | 13 |
| 1.4.1. <i>Apprentissage</i> :..... | 13 |
| 1.4.2. <i>Reconnaissance et décision</i> : | 14 |
| 2. RECONNAISSANCE DES CARACTERES..... | 17 |
| 2.1. ASPECT GENERAL D'UN OCR | 17 |
| 2.2. RECONNAISSANCE DES CARACTERES TIFINAGH..... | 19 |
| 2.2.1. <i>Caractères Tifinagh</i> | 19 |
| 2.2.2. <i>État de l'art</i> | 19 |
| 2.2.3. <i>Principales Contributions</i> : | 21 |
| 3. RECONNAISSANCE D'OBJETS REELS..... | 22 |
| 3.1.1. <i>Approches de Reconnaissance</i> | 22 |
| 3.1.2. <i>État de l'art</i> | 23 |
| 3.1.3. <i>Principales contributions</i> | 26 |
| 4. CONCLUSION..... | 26 |
| 1. INTRODUCTION..... | 28 |
| 2. DESCRIPTEURS DE BASE | 28 |
| 2.1. HISTOGRAMMES DE PROJECTION | 28 |
| 2.2. ZONING (DENSITES) | 29 |
| 3. DESCRIPTEURS INVARIANTS..... | 29 |
| 3.1. MOMENTS GEOMETRIQUES..... | 29 |
| 3.2. MOMENTS DE ZERNIKE | 31 |
| 3.3. DESCRIPTEURS DE FOURIER GENERIQUES | 32 |
| 3.4. TRANSFORMEE DE RADON, | 33 |
| 4. DESCRIPTEURS DE COULEUR | 34 |
| 5. EXTRACTION DES SEGMENTS DE CONTOURS | 37 |
| 5.1. MODELES DE CONTOURS | 37 |
| 5.2. APPROCHES DE DETECTION | 38 |
| 5.3. EXTRACTION DE CONTOURS. OPERATEURS DERIVATIFS DU 1 ^o ORDRE | 38 |
| 6. CONCLUSION..... | 40 |

| | |
|--|-----------|
| 1. INTRODUCTION | 41 |
| 2. RESEAUX DE NEURONES | 41 |
| 2.1. HISTORIQUE..... | 41 |
| 2.2. PRINCIPES..... | 43 |
| 2.3. ARCHITECTURE D'UN RESEAU DE NEURONES | 45 |
| 2.4. PERCEPTRON..... | 46 |
| 2.5. RESEAUX MULTICOUCHE (MULTILAYER NEURAL NETWORKS) | 47 |
| 2.5.1. <i>Apprentissage des RNA</i> | 48 |
| 3. RESEAU NEURO-FLOU (FUZZY-NEURAL) | 51 |
| 3.1. DE LA LOGIQUE FLOUE AUX RESEAUX NEURO-FLOU..... | 51 |
| 3.2. ARCHITECTURE DES RESEAUX NEURO-FLOUS..... | 52 |
| 3.3. ALGORITHME D'APPRENTISSAGE:..... | 53 |
| 4. SEPARATEURS A VASTE MARGE(SVM) | 56 |
| 4.1. NOTIONS DE BASE : HYPERPLAN, MARGE ET SUPPORT VECTEUR | 56 |
| 4.2. LINEARITE ET NON-LINEARITE..... | 57 |
| 4.3. PROBLEME D'APPRENTISSAGE | 58 |
| 4.4. CLASSIFICATION A VALEURS REELLES..... | 58 |
| 4.4.1. <i>Transformation des entrées</i> | 59 |
| 4.4.2. <i>Maximisation de la marge</i> | 59 |
| 4.4.3. <i>Problème primal</i> | 60 |
| 4.4.4. <i>Problème dual</i> | 60 |
| 4.5. NON-LINEARITE (CAS NON SEPARABLE/ MARGE MOLLE) | 61 |
| 4.5.1. <i>Fonction noyau</i> | 61 |
| 5. LES ARBRES DE DECISIONS | 62 |
| 5.1. NOTIONS DE BASE | 62 |
| 5.2. UN EXEMPLE INTRODUCTIF..... | 63 |
| 5.3. REGLE DE CLASSIFICATION | 65 |
| 6. CONCLUSION | 66 |
| 1. NOTIONS DE BASE DE LA GEOMETRIE RIEMANNIENNE | 67 |
| 1.1. VARIETES C^∞ -DIFFERENTIABLES..... | 68 |
| 1.2. VARIETE RIEMANNIENNE..... | 68 |
| 1.3. CHAMPS DE VECTEURS ET FORMES EXTERIEURES..... | 70 |
| 1.3.1. <i>Longueur sur une variété riemannienne</i> | 70 |
| 1.3.2. <i>Connexions et Courbures</i> | 71 |
| 1.4. TORSION | 73 |
| 1.5. CARACTERISTIQUES DES VARIETES RIEMANNIENNES..... | 74 |
| 1.5.1. <i>Connexion de Levi – Civita</i> | 74 |
| 1.5.2. <i>Tenseur de Riemann</i> | 76 |
| 1.5.3. <i>Courbure de Ricci</i> | 79 |
| 1.5.4. <i>Courbure scalaire</i> | 80 |
| 2. DESCRIPTEURS METRIQUES | 81 |
| 2.1. PRETRAITEMENT | 81 |
| 2.2. CALCUL DES DESCRIPTEURS METRIQUES | 83 |

| | | |
|-----------|--|------------|
| 3. | LES DESCRIPTEURS GEODESIQUES..... | 86 |
| 3.1. | COURBES GEODESIQUES..... | 86 |
| 3.2. | TRAITEMENT PRELIMINAIRE..... | 88 |
| 3.3. | CALCUL DES DESCRIPTEURS GEODESIQUES..... | 89 |
| 4. | DESCRIPTEURS DE COURBURE..... | 91 |
| 5. | CONCLUSION..... | 95 |
| 1. | INTRODUCTION..... | 97 |
| 2. | RECONNAISSANCE DES CARACTERES..... | 98 |
| 2.1. | DESCRIPTEURS METRIQUES AVEC RNA ET SVM..... | 98 |
| 2.1.1. | <i>Segmentation.....</i> | 98 |
| 2.1.2. | <i>Base d'apprentissage.....</i> | 98 |
| 2.1.3. | <i>Apprentissage et reconnaissance.....</i> | 99 |
| 2.2. | DESCRIPTEURS DE COURBURES AVEC HYBRIDATION DES RNA ET DES FUZZY-NEURAL..... | 101 |
| 2.3. | DESCRIPTEURS GEODESIQUES TESTES PAR LES RNA ET LES ARBRES DE DECISIONS..... | 104 |
| 2.4. | COMPARAISON DE METHODES PROPOSEES..... | 106 |
| 3. | RECONNAISSANCE D'OBJETS COULEUR..... | 108 |
| 3.1. | PRETRAITEMENT..... | 108 |
| 3.1.1. | <i>Acquisition des images (Coil-100 Data Base).....</i> | 108 |
| 3.1.2. | <i>Segmentation.....</i> | 108 |
| 3.2. | METHODE CLASSIQUE DE LA RECONNAISSANCE D'OBJETS COULEUR..... | 111 |
| 3.2.1. | <i>Reconnaissance des images binaires.....</i> | 111 |
| 3.2.2. | <i>Calcul de la fonction de couleur.....</i> | 112 |
| 3.2.3. | <i>Inconvénients de la méthode classique.....</i> | 112 |
| 3.3. | METHODES PROPOSEES..... | 113 |
| 3.3.1. | <i>Choix des descripteurs et de la topologie interne du RNA.....</i> | 113 |
| 3.3.1. | <i>Première méthode.....</i> | 113 |
| 3.3.2. | <i>Deuxième méthode : Influence de la topologie externe des RNA.....</i> | 117 |
| 3.3.3. | <i>Troisième méthode (Intégration des descripteurs Géodésiques centralisés).....</i> | 119 |
| 3.4. | INFLUENCE DES SYSTEMES DE COULEUR SUR LE RENDEMENT DES RESEAUX DE NEURONES [6]..... | 122 |
| 3.4.1. | <i>Espaces de couleurs.....</i> | 122 |
| 3.4.2. | <i>Système de couleurs proposées.....</i> | 123 |
| 3.4.3. | <i>Rendement des espaces de couleurs.....</i> | 124 |
| 4. | CONCLUSION..... | 125 |
| | Conclusion générale..... | 127 |
| | Références..... | 130 |

LISTE DES FIGURES

| | |
|---|----|
| Figure 1.1: Schéma général d'un système de reconnaissance des caractères. | 10 |
| Figure 1.2: Caractères Tifinagh. | 20 |
| Figure 1.3: Exemples d'objets de la base de données Coil-100. | 23 |
| Figure 1.4: Objet de la base Coil pour différents vues. | 23 |
| Figure 2.1: Histogramme horizontal et vertical du chiffre 5 (Matlab Lib). | 29 |
| Figure 2.2: Photographies et histogrammes associés (les quatre composantes R, V, B, L). | 36 |
| Figure 2.3: Quelques profils de contours. | 37 |
| Figure 2.4: Approche gradient et Laplacien pour la détection de contour. | 38 |
| Figure 2.5: Méthodes dérivatives par différences finies | 39 |
| Figure 3.1: Historique des RNA entre 1940 et 1990. | 42 |
| Figure 3.2: Schéma d'un neurone artificiel | 44 |
| Figure 3.3: Architecture d'un réseau de neurones (avec deux couches cachées). | 46 |
| Figure 3.4: Exemple de la PMC auto-associatif. | 50 |
| Figure 3.5: (De la Bibliothèque Matlab). | 54 |
| Figure 3.6: Séparation par SVM. | 56 |
| Figure 3.7: Exemple d'augmentation de l'espace de représentation . | 57 |
| Figure 3.8: Arbre de décision correspondant au tableau 3.3. | 64 |
| Figure 4.1: Exemple de variété géométrique (sphère, tore, plan). | 67 |
| Figure 4.2: Exemple de l'effet de la fonction Meshgrid (0 :1 :7): Matrice (s) | 81 |
| Figure 4.3: Exemple de la projection des résultats obtenus par le programme 1, de deux caractères Tifinagh. | 82 |
| Figure 4.4 : Exemple de détection de contour. | 89 |
| Figure 4.5 : Exemple des extrémités des trois caractères Tifinagh. | 89 |
| Figure 4.6 : Exemple de chemins Géodésiques. | 90 |
| Figure 4.7 : Organigramme de l'algorithme de calcul des descripteurs de courbure. | 92 |
| Figure 5.1 : Segmentation des caractères Tifinagh en utilisant l'algorithme des régions isolées. | 98 |
| Figure 5.2 : Exemple d'un caractère Tifinagh de la base de 'Y.Ouguengay' pour différents | 99 |

styles et tailles.

| | |
|--|-----|
| Figure 5.3 : Processus de classification utilisé avec les descripteurs de Courbures. | 102 |
| Figure 5.4 : Caractères reconnus directement par les descripteurs Géodésiques. | 104 |
| Figure 5.5 : Caractères composés. | 104 |
| Figure 5.6 : Caractères circulaire. | 104 |
| Figure 5.7 : Schéma du processus de classification utilisé avec les descripteurs géodésiques. | 105 |
| Figure 5.8 : Objet de la base pour différentes vues. | 108 |
| Figure 5.9: Base de données Coil – 100 | 109 |
| Figure 5.10: Etape d'étiquetage et de segmentation. | 110 |
| Figure 5.11: Représentation binaire de deux objets de la base Coil-100 | 111 |
| Figure 5.12 : Moments de Zernike sur les images binaires de la figure 5.11. | 111 |
| Figure 5.13 : Conversion en binaire donne une représentation incomplète de l'objet. | 112 |
| Figure 5.14: Perte d'information. | 112 |
| Figure 5.15: Séparation des tris niveaux de gris : quelques valeurs des moments de Zernike pour les trois niveaux gris. | 113 |
| Figure 5.16: Hu & Zernike en série pour les tris niveaux de gris. | 114 |
| Figure 5.17 : Avancement de l'apprentissage du réseau de neurone. | 114 |
| Figure 5.18 : Exemple d'image composée. | 115 |
| Figure 5.19 : Exemple d'image Bruitée. | 115 |
| Figure 5.20: Zernike & Hu avec tris RNA et une fonction de sortie globale $G(x)$. | 117 |
| Figure 5.21 : Hu & Zernike en série avec RNA multi sorties avec couleur et formes détection. | 118 |
| Figure 5.22: Exemples de distances géodésiques entre le centre de l'objet et ses extrémités. | 120 |
| Figure 5.23: Moments de Zernike et Hu et descripteurs géodésiques en série avec un RNA entraîné par la rétropropagation du gradient. | 121 |

Liste des Tableaux

| | |
|--|-----|
| Tableau 1.1: Quelques méthodes de reconnaissances proposées entre 2009-2011. | 20 |
| Tableau 3.1: Analogie entre les neurones biologiques et artificiels | 43 |
| Tableau 3.2: Exemple de fonction d'activation | 45 |
| Tableau 3.3: Exemple de Tableau de Vérité | 63 |
| Tableau 4.1: Descripteurs métriques pour quelques caractères Tifinagh | 85 |
| Tableau 4.2: Descripteurs métriques pour les deux caractères Yass et Yarr. | 86 |
| Tableau 4.3: Descripteurs métriques pour les deux caractères Yass et Yarr. | 86 |
| Tableau 4.4: Résultats DF géodésiques obtenus pour quelques caractères Tifinagh. | 90 |
| Tableau 4.5: Descripteurs géodésiques relatifs aux caractères "Yars" et "Yass" | 91 |
| Tableau 4.6: Descripteurs géodésiques calculés pour différentes tailles du caractère "Yass" | 91 |
| Tableau 4.7: Descripteurs de courbures pour quelques caractères Tifinagh. | 95 |
| Tableau 5.1: Évaluation de taux de reconnaissance en fonction du nombre de caractères imprimés à identifier. | 100 |
| Tableau 5.2: Évaluation de taux de reconnaissance en fonction du nombre de caractères manuscrits. | 100 |
| Tableau. 5.3 : Taux de reconnaissance en fonction du nombre de caractères à identifier. | 102 |
| Tableau.5.4 : Matrice de confusion du système SVM et Descripteurs métriques | 103 |
| Tableau 5.5: Taux de reconnaissance en fonction du nombre de caractères à identifier. | 105 |
| Tableau 5.6: Matrice de confusion obtenue par le système amélioré appliqué sur la base des caractères amazighe. | 107 |
| Tableau 5.7 : Tableau récapitulatif des résultats obtenus | 106 |
| Tableau 5.8: Taux de reconnaissances obtenus par les Réseaux de neurones et les SVM pour différentes altérations. | 115 |
| Tableau 5.9 : Bilan globale des résultats obtenus pour un PC (Core-Duo 2Ghz, 1Go de RAM) . | 116 |
| Tableau 5.10 : Taux de reconnaissances obtenus par les différents DF testés. | 116 |
| Tableau 5.11 : Exemples de sorties d'un réseau de neurones à 15 sorties. | 118 |
| Tableau 5.12: La sorties O_d avant et après l'application des règles de calcul. | 119 |
| Tableau5.13: Taux de reconnaissances obtenus en appliquant Hu & Zerinke comme vecteur d'entrée pour différentes topologies de RNA. | 119 |
| Tableau 5.14 : Exemples de descripteurs Géodésiques centralises | 120 |

| | |
|--|-----|
| Tableau 5.15: Taux de reconnaissance obtenus en utilisant Hu, Zernike et leur combinaison avec les descripteurs géodésiques, (RNA avec 2 couches cachées). | 121 |
| Tableau 5.16 : Taux de reconnaissances obtenus par application des RNA et des moments de Zernike sur les différents espaces de couleurs. | 125 |

Introduction

La reconnaissance des formes est définie comme étant un ensemble de techniques et méthodes visant à identifier des motifs informatiques à nature visuelle, à partir de données brutes (des images) afin de prendre une décision en fonction de la catégorie attribuée à ce motif [1].

Un système de reconnaissance est généralement composé de trois phases :

- Une phase de prétraitement qui dépend de la nature et de la complexité de la forme traitée.
- Une phase d'extraction d'attributs qui vise la réduction de la taille des informations traitées est l'extraction de signatures capables d'identifier l'objet.
- Une phase de classification qui consiste à attribuer une classe ou catégorie à chaque objet (ou forme) à classer, en se basant sur les descripteurs calculés dans la phase d'extraction.

Dans nos travaux de recherches, nous nous sommes intéressés à développer des systèmes de reconnaissance automatique de formes et nous avons étalé nos plans de recherches sur deux lignes : la reconnaissance des caractères et la reconnaissance des objets réels. Ces deux lignes de recherche partagent le même schéma général, mais elles utilisent des méthodes plus au moins différentes selon la complexité des objets traités et des algorithmes adoptés.

Cette thèse regroupe l'ensemble des travaux de recherche que nous avons réalisés entre 2009 et 2012 [3,4,5,6,7,8], faisant appel à de nouvelles méthodes d'extraction avec intégration d'une multitude de systèmes de reconnaissance hybrides basés essentiellement sur les réseaux de neurones. Dans ce cadre nous avons deux objectifs, premièrement l'étude des descripteurs classiques et de leurs comportements hybrides et deuxièmement l'exploitation des outils proposés par la géométrie riemannienne (la notion de la métrique riemannienne, la notion de la courbure et de la distance géodésique) pour le développement de nouveaux descripteurs du contenu visuel des images. Également, Nous avons établi des études comparatives entre les paramètres qui peuvent influencer les taux de reconnaissances, à savoir : les espaces de couleurs, les classificateurs et les bases de données.

Cette thèse est organisée comme suite :

Le chapitre 1 donne une brève présentation du processus standard de reconnaissance et de ses différentes étapes. Il présente en deux parties les deux voies de recherches sur lesquelles nous nous sommes penchés : la reconnaissance des caractères et la reconnaissance d'objets réels. Nous avons conclu chaque partie avec un état d'art suivi par les principales contributions.

Le chapitre 2 est entièrement consacré à l'étape d'extraction. Nous avons essayé de présenter les descripteurs de formes couramment utilisés : les moments de Zernike, les moments de Hu, les descripteurs de Fourier, les descripteurs morphologiques et les descripteurs basés sur les histogrammes de couleurs.

Le chapitre 3 est consacré à la phase de classification. Il présente les principes théoriques des réseaux de neurones (RNA) et les topologies des réseaux que nous avons intégrés : les réseaux multicouches et les réseaux Neuro-Flou et il met en relief les mécanismes de fonctionnements des SVM (Séparateurs à Vaste Marge), que nous avons choisis pour comparer le rendement des RNA et il se termine par une petite introduction sur les Arbres de décision.

Le chapitre 4 présente les descripteurs de formes (DF) proposés et qui sont essentiellement basés sur les principes de la géométrie riemannienne. Ce chapitre est subdivisé en deux parties : la première présente les notions de base de la géométrie riemannienne et la deuxième décrit les méthodes d'extractions que nous avons proposées.

Le chapitre 5 met en évidence les systèmes de reconnaissances proposés et les résultats expérimentaux obtenus. Ce chapitre est divisé en deux parties : la première est dédiée aux systèmes que nous avons développés pour la reconnaissance des caractères et la deuxième est consacrée aux systèmes de reconnaissance d'objets couleur. Nous présentons à la fin de chaque partie les résultats obtenus par ces systèmes, tout en fournissant des interprétations.

Chapitre 1

État de l'art

1. Structure d'un système de Reconnaissance de Formes (RF)

Récemment, des progrès considérables ont été réalisés dans la mise au point de systèmes de reconnaissance, d'une part grâce aux nombreux travaux effectués dans ce domaine et, d'autre part, grâce à la production jugée à bas prix des micro-ordinateurs et des systèmes d'acquisition (caméra numérique, scanner, tablette à digitaliser... etc.). Nous présentons dans cette partie les différentes étapes de reconnaissances (Figure. 1.1).

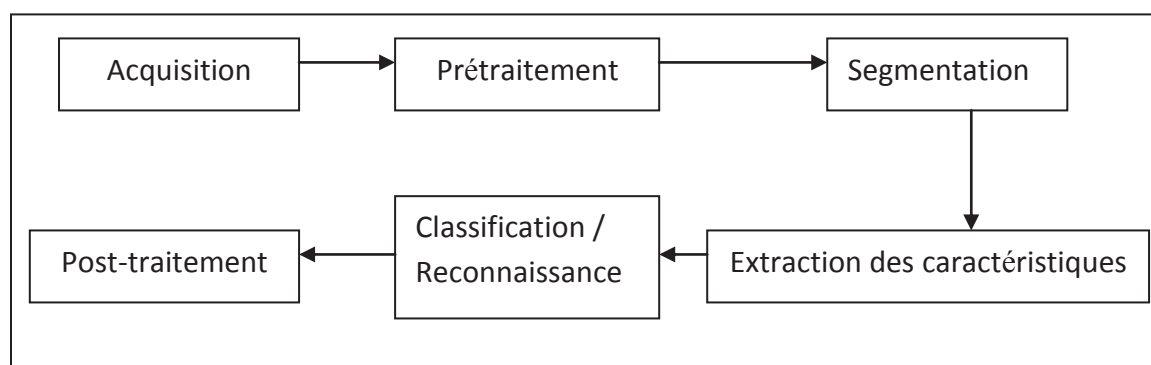


Figure 1.1 : Schéma général d'un système de reconnaissance de formes.

Un système de reconnaissance fait appel généralement aux phases suivantes : acquisition, prétraitement, segmentation, extraction des caractéristiques, classification et une phase de post-traitement en cas de besoin.

1.1.Phase d'acquisition

L'acquisition permet la conversion de l'objet en image numérique. Cette étape délicate dépend fortement du choix et du paramétrage du matériel à utiliser (Appareil photo, Scanner) ainsi que du format de stockage des images.

La numérisation des documents est effectuée par balayage optique. Le résultat est rangé dans un fichier de points, appelés pixels, dont la taille dépend de la résolution [9].

La technicité des matériels d'acquisition (Appareil photo) a remarquablement progressé ces dernières années, on retrouve aujourd'hui des capteurs HD, 3D et 2,5D. Ces technologies ont permis d'introduire de nouvelles techniques et des possibilités de recherche vaste et efficace.

1.2.Phase de prétraitement

Le prétraitement consiste à préparer les données issues du capteur pour permettre davantage d'efficacité et de précision dans les phases de reconnaissance ultérieures. Il s'agit essentiellement de réduire le bruit superposé aux données et essayer de ne garder que l'information significative de la forme représentée. Le bruit est généralement dû aux conditions d'acquisition (éclairage, mise incorrecte du document, résolution du capteur, etc.).

Parmi les opérations de prétraitement généralement utilisées, on peut citer : l'extraction des composantes connexes, le lissage, la normalisation et la squelettisation.

- a. **La binarisation** : est le passage d'une image en couleur ou définie par plusieurs niveaux de gris en image binaire (composée de deux valeurs 0 et 1) qui permet une classification entre le fond et la forme.
- b. **L'extraction des composantes connexes** : Une composante connexe est un ensemble de points dans le plan liés par rapport à leurs voisinages pour composer un objet ou une fraction d'objet. Cette technique est utilisée pour détecter le nombre d'objets ou de caractères séparables [10].
- c. **Le Lissage** : L'image des objets peut être entachée de bruit dû aux outils d'acquisition, qui conduit soit à une absence de points soit à une surcharge de points. Les techniques de lissage permettent de résoudre ces problèmes par des opérations locales qu'on appelle opérations de bouchage et de nettoyage [10]. L'opération de nettoyage permet de supprimer les petits creux et tâches de la forme. Pour le bouchage il s'agit d'égaliser les contours et de boucher les trous internes à la forme du caractère en lui ajoutant des points noirs.

1.3.Phase d'extraction des caractéristiques

C'est l'une des étapes les plus délicates et les plus importantes en RF. La reconnaissance d'un objet passe d'abord par l'analyse de sa forme et l'extraction de ses traits caractéristiques (primitifs) qui seront exploités pour son identification.

Les types de caractéristiques peuvent être classés en quatre groupes principaux : caractéristiques structurelles, caractéristiques statistiques, transformations globales, et superposition des modèles et corrélation [11,12,13].

a. Caractéristiques structurelles :

Les caractéristiques structurelles décrivent une forme en termes de sa topologie et sa géométrie en donnant ses propriétés globales et locales [13]. Parmi ces caractéristiques on peut citer :

- Les traits et les anses dans les différentes directions ainsi que leurs tailles. Les points terminaux [35].
- Les points d'intersection [35].
- Les boucles [36].
- Le nombre de points diacritiques et leur position par rapport à la ligne de base. (le cas des caractères).
- La hauteur et la largeur de l'objet.
- La catégorie de la forme (partie primaire ou secondaire.) [4].
- Plusieurs autres caractéristiques peuvent être tirées, suivant qu'ils soient extraits d'une courbe, d'un trait ou d'un segment de contour.

b. Les caractéristiques statistiques :

Les caractéristiques statistiques décrivent une forme en extrayant un ensemble de mesures de celle ci. Par exemple : le zonage (zoning), les caractéristiques de lieu géométrique et les moments [14].

- Le zonage : consiste à superposer une grille $N \times M$ sur l'image et pour chacune des régions résultantes on calcule la moyenne ou le pourcentage de points en niveaux de gris, donnant ainsi une matrice de taille $n \times m$ de caractéristiques [13].

- La méthode Loci : est basée sur le calcul du nombre de segments blancs et de segments noirs le long d'une ligne verticale traversant la forme, ainsi que leurs longueurs [11].
- La méthode des moments : les moments d'une forme par rapport à son centre de gravité sont invariants par rapport à la translation et à la rotation. Ils sont aussi indépendants de l'échelle. Ces caractéristiques peuvent être facilement et rapidement extraites d'une image et peuvent tolérer modérément les bruits et les variations [12,13].

c. Les transformations globales :

La transformation consiste à convertir la représentation en pixels en une représentation plus abstraite pour réduire la dimension des caractères, tout en conservant le maximum d'informations sur la forme à reconnaître.

Une des transformations les plus simples est celle qui représente le squelette ou le contour d'un caractère sous forme d'une chaîne de codes de directions [14]. La chaîne de code obtenue est souvent simplifiée pour réduire les redondances et les changements brusques de direction.

d. Superposition des modèles (Template Matching) et corrélation :

La méthode de 'Template Matching' appliquée à une image binaire (en niveaux de gris ou squelettes), consiste à utiliser l'image de la forme comme vecteur de caractéristiques qui sera comparé à un modèle (Template) pixel par pixel pour mesurer la similarité [15].

1.4. Classification

La classification dans un système de reconnaissance regroupe deux tâches : l'apprentissage et la reconnaissance. Dans cette étape, les caractéristiques de l'extraction sont utilisées pour identifier l'objet et l'attribuer à un modèle de référence.

1.4.1. Apprentissage :

Il s'agit, lors de cette étape, d'apprendre au système les propriétés pertinentes du vocabulaire utilisé et de l'organiser en modèles de références.

L'idéal serait d'apprendre au système autant d'échantillons que de types d'objets, mais cela est impossible à cause de la grande variabilité des formes qui conduirait à une explosion combinatoire de modèles de représentation. La tendance consiste alors à remplacer le nombre par une meilleure qualité des traits caractéristiques.

L'apprentissage consiste en deux concepts différents : l'entraînement et l'adaptation. L'entraînement consiste à enseigner au système la description des caractères tandis que l'adaptation sert à améliorer les performances du système en profitant des expériences précédentes. Certains systèmes permettent à l'utilisateur d'identifier un objet lorsqu'ils échouent à le reconnaître et ils utilisent l'entrée de l'utilisateur à chaque fois que l'objet est rencontré.

Les procédés d'apprentissage sont différents selon qu'il s'agisse de reconnaissance de caractères ou d'objets plus complexes. D'une manière générale, on distingue deux types de techniques d'apprentissage : supervisé [19] et non supervisé [20].

1.4.2. Reconnaissance et décision :

La décision est l'ultime étape de reconnaissance. À partir de la description en paramètres de l'objet traité, le module de reconnaissance cherche parmi les modèles de référence présents, ceux qui lui sont les plus proches.

La reconnaissance peut conduire à un succès si la réponse est unique (un seul modèle répond à la description de la forme du caractère). Elle peut conduire à une confusion si la réponse est multiple (plusieurs modèles correspondent à la description). Enfin, elle peut conduire à un rejet de la forme si aucun modèle ne correspond à sa description. Dans les deux premiers cas, la décision peut être accompagnée d'une mesure de vraisemblance, appelée aussi score ou taux de reconnaissance.

Les approches de reconnaissance peuvent être classées en trois principaux groupes : l'approche statistique, l'approche structurelle et l'approche stochastique.

a. Approche statistique :

L'approche statistique est fondée sur l'étude statistique des mesures effectuées sur les formes à reconnaître. L'étude de répartition des dites formes dans un espace métrique et la

caractérisation statistique des classes, permettent de prendre une décision de reconnaissance du type « plus forte probabilité d'appartenance à une classe ».

Les approches statistiques bénéficient des méthodes d'apprentissage automatique s'appuyant sur des bases théoriques fondées, telles que la théorie de la décision bayésienne et les méthodes de classification non supervisées. Le problème de reconnaissance revient à affecter une forme inconnue à l'une des classes obtenues pendant l'apprentissage. Nous pouvons citer trois méthodes statistiques couramment utilisées :

- Méthode du plus proche voisin :

L'algorithme KNN (K Nearest Neighbors) affecte une forme inconnue à la classe de son plus proche voisin en le comparant aux formes stockées dans une classe de références nommées prototypes. Il renvoie les K formes les plus proches de la forme à reconnaître suivant un critère de similarité. Une stratégie de décision permet d'affecter des valeurs de confiance à chacune des classes en compétition et d'attribuer la classe la plus vraisemblable (au sens de la métrique choisie) à la forme inconnue [19].

Cette méthode présente l'avantage d'être facile à mettre en œuvre et fournit de bons résultats. Son principal inconvénient est lié à la faible vitesse de classification due au nombre important de distances à calculer.

- Réseaux de neurones :

Un réseau de neurones est un graphe orienté pondéré. Les nœuds de ce graphe sont des automates simples appelés neurones formels. Les neurones sont dotés d'un état interne, l'activation, par lequel ils influencent les autres neurones du réseau. Cette activité se propage dans le graphe le long d'arcs pondérés appelés liens synaptiques [19].

En RF, les primitives extraites d'une image d'un objet (ou de l'entité choisie) constituent les entrées du réseau. La sortie activée du réseau correspond à l'objet reconnu. Le choix de l'architecture du réseau est un compromis entre la complexité des calculs et le taux de reconnaissance [20].

b. Approche structurelle :

Ces approches reposent sur les méthodes structurelles d'extractions précitées dans ce chapitre. Elles cherchent à trouver des éléments simples ou primitifs et à décrire leurs

relations. Les primitives sont de type topologique telles que : une boucle, un arc, un segment de droite, etc, et une relation peut être la position relative d'une primitive par rapport à une autre. Parmi les méthodes structurelles, nous pouvons citer :

- Méthodes de tests :

Elles consistent à appliquer sur chaque caractère traité des tests de plus en plus fins sur la présence ou l'absence de primitives, de manière à répartir les échantillons en classes. Le processus le plus usuel consiste à diviser à chaque test l'ensemble des choix en deux jusqu'à n'obtenir qu'une seule forme correspondant au caractère entré. Ce choix dichotomique est très rapide et très simple à mettre en œuvre, mais il est très sensible aux variations du tracé.

- Comparaison de chaînes :

Les caractères sont représentés par des chaînes de primitives. La comparaison du caractère traité avec le modèle de référence consiste à mesurer la ressemblance entre les deux chaînes. La mesure de ressemblance peut se faire par calcul de distance ou par examen de l'inclusion de toute ou une partie d'une chaîne dans l'autre [9].

c. Approche stochastique :

Contrairement aux méthodes précédemment décrites, l'approche stochastique utilise un modèle pour la reconnaissance, prenant en compte la grande variabilité de la forme. La distance communément utilisée dans les techniques de « comparaison dynamique » est remplacée par des probabilités calculées de manière plus fine par apprentissage. La forme est considérée comme un signal continu observable dans le temps à différents endroits constituant des états «d'observation ». Le modèle décrit ces états à l'aide de probabilités de transitions d'états et de probabilités d'observation par état. La comparaison consiste à chercher dans ce graphe d'états, le chemin de probabilité forte correspondant à une suite d'éléments observés dans la chaîne d'entrée. Ces méthodes sont robustes et fiables du fait de l'existence d'algorithmes d'apprentissage efficaces. Si l'apprentissage est lent, la reconnaissance est par contre très rapide, car les modèles comprennent généralement peu d'états, et le calcul est relativement immédiat. Les méthodes les plus répandues dans le cadre de cette approche sont les méthodes utilisant les Modèles de Markov Cachés (MMC).

2. Reconnaissance des caractères

2.1.Aspect général d'un OCR

Il n'existe pas de système universel d'OCR (Optical Character Recognition) qui permet de reconnaître n'importe quel caractère dans n'importe quelle fonte. Tout dépend du type de données traitées et bien évidemment de l'application visée. Il existe plusieurs modes de classification des systèmes OCR [21,22,26] parmi lesquels on peut citer :

- Les systèmes qualifiés de « en-ligne » ou « hors-ligne » suivant le mode d'acquisition.
- Les approches diffèrent selon si on parle de l'imprimé ou du manuscrit.
- Les approches statistiques, structurelles ou stochastiques relatives aux traits caractéristiques extraits des formes considérées.

a. Reconnaissance En-Ligne et Hors-Ligne :

Ce sont deux modes différents d'OCR, ayant chacun ses outils propres d'acquisition et ses algorithmes correspondants :

• *Reconnaissance En-Ligne (On-Line) :*

Ce mode de reconnaissance s'opère en temps réel (pendant l'écriture). Les symboles sont reconnus au fur et à mesure qu'ils sont écrits à la main. Il est réservé généralement à l'écriture manuscrite. C'est une approche «signal» où la reconnaissance est effectuée sur des données à une dimension. L'écriture est représentée comme un ensemble de points dont les coordonnées sont en fonction du temps. La reconnaissance en-ligne présente un avantage majeur, c'est la possibilité de correction et de modification de l'écriture de manière interactive vu la réponse en continu du système [9].

• *Reconnaissance Hors-Ligne (Off-Line) :*

Elle démarre après l'acquisition, elle convient aux documents imprimés et les manuscrits déjà rédigés. Ce mode peut être considéré comme le cas le plus général de la reconnaissance de l'écriture. Il se rapproche du mode de la reconnaissance visuelle. L'interprétation de l'information est indépendante de la source de génération.

b. Reconnaissance de l'imprimé ou du manuscrit :

Les approches diffèrent selon qu'il s'agit de reconnaissance de caractères imprimés ou manuscrits. Les caractères imprimés sont dans le cas général alignés horizontalement et séparés verticalement, ce qui simplifie la phase de lecture [23]. La forme des caractères est définie par un style calligraphique (fonte) qui constitue un modèle pour l'identification. Dans le cas du manuscrit, les caractères sont souvent ligaturés et leur graphisme est inégalement proportionné provenant de la variabilité intra et inter scripteurs. Cela nécessite généralement l'emploi de techniques de délimitation spécifiques et souvent des connaissances contextuelles pour guider la lecture.

○ *Caractères imprimés :*

Dans le cas de l'imprimé, la reconnaissance peut être monofonte, multifonte ou omnifonte. Un système est dit monofonte s'il ne peut reconnaître qu'une seule fonte à la fois c'est-à-dire qu'il ne connaît de graphisme que d'une fonte unique. C'est le cas le plus simple de reconnaissance de caractères imprimés. Un système est dit multifonte s'il est capable de reconnaître divers types de fontes parmi un ensemble de fontes préalablement apprises [9]. Et un système omnifonte est capable de reconnaître n'importe quelle fonte, généralement sans apprentissage préalable. Cependant, ceci est quasiment impossible, car il existe des milliers de fontes dont certaines illisibles, par l'homme (sauf bien sûr pour celui qui l'a conçue) et avec un logiciel de création de fonte n'importe qui peut concevoir des fontes à sa guise.

○ *Caractères manuscrits :*

La reconnaissance de l'écriture manuscrite se distingue par plusieurs spécificités par rapport à la reconnaissance optique des caractères (OCR : Optical Character Recognition) qui concerne les caractères imprimés ou dactylographiés. Il est nécessaire de distinguer la reconnaissance en ligne (on-line) de l'écriture manuscrite, qui relève plutôt de l'interfaçage entre l'homme et l'ordinateur (un stylo spécial est connecté à la machine et ne fonctionne que sur une tablette sensible), de la reconnaissance hors ligne (off-line) où l'entrée est une image numérique de l'écriture. Seule la reconnaissance hors ligne sera considérée dans ce travail. Dans le cas du manuscrit, la reconnaissance peut être monoscripteur, multiscripteur ou omniscripteur :

- Un système est dit Mono-scripteur (propres au scripteur) lorsqu'il ne peut reconnaître qu'une seule écriture. Tous ces éléments influencent sur la forme des lettres (écriture

penchée, bouclée, arrondie, linéaire, etc.) et bien sûr sur la forme des ligatures, compromettant parfois le repérage des limites entre lettres.

- Un système est dit Multi-scripteurs (propres à l'écriture manuscrite) lorsqu'il peut identifier et reconnaître l'écriture pour un certain nombre de scripteurs.
- Un système est dit Omni-scripteurs (propres à n'importe quelle écriture manuscrite) lorsqu'il réduit l'information contenue dans l'image au minimum nécessaire pour modéliser précisément la structure des caractères.

2.2.Reconnaissance des caractères Tifinagh

2.2.1. Caractères Tifinagh

Historiquement, les caractères Tifinagh étaient connus chez les Foqha (théologiens) marocains sous le nom de « Caractères des Sables (Khath Rraml) », c'est à dire les écritures des caravaniers du désert. Ils étaient utilisés autrefois pour échanger des messages entre les nomades, qui traçaient des signes sur les routes sahariennes, étant donnée l'importance de la communication et la complexité des itinéraires des voyages. Ces caractères ont été précieusement conservés par les sahariens et représentent aujourd'hui l'écriture ancestrale des Touarègues. Les chercheurs ont découvert des textes en Tifinagh ayant différentes formes : humanoïdes, géométriques et d'autres formes symbolisant la sacralité, ceci tout en notant des ressemblances avec d'autres caractères appartenant à diverses civilisations (phénicienne, russe, araméenne). En Tamazight, le mot Tifinagh se compose de deux mots (Tifi : découverte et Nagh : de soi), ce qui voudrait dire : découverte de soi.

2.2.2. État de l'art

Actuellement, et sur l'échelle internationale, différentes équipes de recherches se penchent sur le traitement automatique de la culture amazighe, en l'occurrence l'équipe TIT de la FST de Béni Mellal qui se spécialise dans la reconnaissance des caractères et documents Tifinagh.

L'Institut Royal de la Culture Amazighe (IRCAM) propose une standardisation des caractères Tifinagh composée de 33 caractères (Figure 1.2).



Figure 1.2 : Caractères Tifinagh adoptés par l'IRCAM.

De nos jours, les travaux de recherches liées à l'informatisation de l'Amazigh sont concentrés au Maroc et en Algérie. Le Maroc compte trois pôles de recherche l'IRCAM, l'université Sultan Moulay Sliman de Beni Mellal et l'université IBN ZOHR d'Agadir:

- L'IRCAM se présente comme le pilote national de l'informatisation de Tifinagh par son soutien de toutes les manifestations scientifiques liées au domaine et par une quantité importante de livres et de publications de qualité [24,25].
- L'université Ibn Zohr présente annuellement des publications de valeur par ses nombreux chercheurs (Ex. Pr. Ali Rachidi et Pr. Driss Mammass) [26,27,28,29].
- L'équipe TIT de l'université Sultan Moulay Sliman a produit des dizaines de publications sur la reconnaissance des caractères Tifinagh lors des trois dernières années [3, 4, 5, 6, 7, 8, 9, 22, 30].

Le tableau 1.1 présente les articles qui ont obtenu les meilleurs taux de reconnaissances

| Auteurs | Méthodes proposées | Remarques |
|---------------------|---|--|
| Djematen et al [60] | une méthode statistique de reconnaissance de caractères berbères manuscrits basée sur la position des points caractéristiques dans le rectangle-enveloppe de l'image du caractère. | Taux de confusion élevé. |
| Fakir et al [61]. | Ils ont montré une comparaison entre deux algorithmes : les cartes auto-organisatrices de Kohonen et les k-plus proches voisins flou. La technique du codage rétinien est employée dans l'extraction d'attributs. | Une méthode appliquée sur une base de données partielle. |

| | | |
|-----------------------|---|--|
| El kessab et al [62]. | Ils ont développé un système de reconnaissance des caractères Tifinaghe manuscrits, basé sur l'utilisation des réseaux de neurones et le modèle de Markov caché. La morphologie mathématique est adoptée dans la phase d'extraction. | Un taux de reconnaissance de 93.80% a été obtenu pour une base de test de 1275 images. |
| Bazdouz et al [63]. | Ils ont utilisé les MMC pour la reconnaissance des caractères Tifinaghe isolés. La taille de base de données est de 42 caractères. | Un taux de reconnaissance de 87% a été obtenu. |
| Ait Ouguengay [25]. | Un réseau de neurones artificiels (libre JavaNNS simulateur). Avec comme descripteurs les projections horizontales et verticales, les centres de gravité en x et en y, le périmètre, l'aire, la compacité et les moments centraux d'ordre 2 | -simple implémentation -Taux de reconnaissance insuffisant |
| Amrouch et al [64]. | Modèle de Markov cachée (HMM), et la transformé de Haugh | Un modèle de Markov par caractère ce qui complique l'intégration de ce système. |
| El Ayachi et al [65]. | La programmation dynamique avec les moments invariants et la transformée de Walsh | Avantage des moments invariants par rapport à la transformée de Walsh. |
| El Ayachi et al [66]. | RNA et les moments invariants. | Amélioration des taux de reconnaissances. Confusions encore présentes concernant la reconnaissance des caractères Tifinagh |
| Essaadi et al [29]. | une approche syntaxique avec les automates à état finis | Difficultés d'intégration pour 33 caractères |
| Essaadi et al [67]. | Un système de reconnaissance d'écriture amazighe basée sur la ligne centrale horizontale de l'écriture et les RNA. | La taille du Réseau de neurones est très grande. 92 neurones dans la première couche seulement |

Le tableau 1.1. Quelques méthodes de reconnaissances proposées entre 2009-2011.

2.2.3. Principales Contributions :

- Concernant le processus d'extraction, nous proposons pour la première fois trois nouveaux descripteurs de forme basés sur les principes de la géométrie Riemannienne :
 - Deux types de descripteurs globaux que nous avons appelés les « descripteurs métriques » et les « descripteurs de Courbure » [3,5].

- Un descripteur d'aspect local que nous avons appelé « descripteurs Géodésiques » [4].

Les descripteurs proposés ont fait preuve d'une fiabilité considérable et des taux de reconnaissance intéressants avec un nombre de paramètres réduit.

- Au niveau de la Classification, on a proposé plusieurs systèmes de reconnaissance de caractères Amazigh :
 - Un système intégrant les réseaux de neurones et les descripteurs métriques.
 - Un système composé des Séparateurs à Vaste Marge (SVM) et des descripteurs métriques.
 - Un système composé des descripteurs Géodésiques et l'hybridation des réseaux de neurones et des arbres de décisions.
 - Un système basé sur les descripteurs de courbures et l'hybridation des réseaux de neurones et des réseaux Neuro-Flous.

Progressivement à travers ces systèmes, nous avons réussi à améliorer les différents taux de reconnaissance.

3. Reconnaissance d'objets réels

Malgré l'évolution des acquis technologiques, reconnaître un objet 3D en couleur reste une tâche délicate qui nécessite un traitement personnalisé selon la complexité de l'objet. Actuellement, on peut compter deux groupes majeurs de théories qui divergent sur le format de la représentation selon si celui-ci est dépendant ou non des vues de l'objet à représenter.

3.1.1. Approches de Reconnaissance

Le premier groupe de ces théories considère que la représentation d'un objet est conçue comme un ensemble de caractéristiques (invariants) indépendantes des vues de ce même objet [31]. Il s'agit d'une description structurelle de l'objet. L'une des approches les plus intéressantes et celle de Biederman : «Reconnaissance Par les Composantes » (RPC) [32] qui consiste à représenter l'objet en le décomposant en des structures (primitives). Le RPC prédit que la reconnaissance ne dépendra pas des points de vues tant que l'extraction de la même description structurelle de plusieurs vues différentes de l'objet reste possible.

Le second groupe de ces théories considère que la représentation d'un objet est liée à des vues spécifiques à l'objet et que n'importe quelle autre vue pourra être déduite à l'aide de ces vues [34]. Les modèles de ce type de représentation considèrent une vue comme une collection de caractéristiques (informations 2D information 3D,...). La reconnaissance est exprimée comme une fonction des images déjà vues. Plusieurs modèles computationnels utilisés notamment par Edelman et Poggio, basé sur des expériences en neurophysiologie et des données psychophysiques, ont été testés en employant des réseaux de neurones pour l'apprentissage et la reconnaissance d'objets par les vues.



Figure 1.3 Exemples d'objets de la base de données Coil-100



Figure 1.4 : Objet de la base Coil-100 pour différentes vues

3.1.2. État de l'art

a. Méthodes géométriques

Les premières méthodes de reconnaissance d'instances d'objets sont des méthodes géométriques [37,38]. Elles consistent à représenter les objets de référence par leurs contours. On distingue les méthodes à base d'alignement modèle-objet et les méthodes à base de tables de hachage. Les méthodes d'alignement géométrique disposent d'un modèle 3D de l'objet recherché. Les primitives (par exemple des droites) composant ce modèle 3D sont alignées sur les primitives détectées dans l'image, et la qualité de l'alignement détermine si l'objet est présent ou non [39]. Le calcul des correspondances entre les primitives du modèle et de l'image est un processus NP-complet, des heuristiques sont donc nécessaires pour réduire la complexité du problème. On peut citer les arbres d'interprétation [40], qui sont des arbres de

tous les alignements modèle-image possibles, élagués par l'application d'heuristiques. Par ailleurs, Fischler et al [41] appliquent la méthode RANSAC (RANdom SAmple Consensus) pour obtenir un consensus d'alignement à partir d'un sous ensemble des primitives de l'image, et décident de la validité de l'alignement avec les primitives de l'ensemble complémentaire.

Lamdan et al [42] utilisent des tables de hachage pour la reconnaissance. Les primitives géométriques des différents modèles d'objets sont stockées dans une table de hachage, et lors de la reconnaissance, les primitives observées dans l'image sont recherchées dans la table de hachage, et des votes sont accumulés pour les différents modèles. Un objet est reconnu si le modèle correspondant obtient une quantité minimale de votes.

En ne considérant que le contour des objets, et donc en ignorant l'intérieur de ceux-ci, les méthodes géométriques bénéficient d'une grande robustesse aux changements d'illumination et de texture des objets. De plus, l'utilisation d'un modèle 3D permet de traiter les occultations et le fond chargé de manière très subtile. Ces méthodes souffrent cependant de deux défauts. Premièrement, certaines catégories d'objets, comme les arbres, ne peuvent être définies uniquement par leurs contours. Deuxièmement, ces méthodes ne sont pas robustes à une détection imprécise des contours, ce qui est souvent le cas dans des conditions d'acquisition d'images réelles.

b. Méthodes de reconnaissance globale

Les méthodes de reconnaissance globale [43,44,45] calculent une signature de l'image prise dans sa globalité. Cette signature peut être, par exemple, la distribution des couleurs dans l'image. Une telle signature est calculée pour un ensemble d'images représentatives d'une instance d'objet. Lors de la reconnaissance, la signature de l'image inconnue est calculée, puis est comparée aux images de références. L'objet prédit est l'objet de référence le plus similaire à la signature de l'image inconnue. La signature de l'image peut être constituée de l'ensemble des pixels mis les uns à la suite des autres dans un vecteur colonne. La comparaison avec les images de référence peut se faire avec une analyse en composante principale, ce qui est le cas des Eigenfaces (Visages propres) de Turk et al [46].

Les méthodes globales sont très simples, et elles sont robustes aux modifications d'illumination et de contraste à condition que la base d'images de référence contienne une grande quantité d'images aux conditions d'illumination différentes. Pour cette raison, les

méthodes globales ont l'inconvénient de nécessiter une grande quantité de données. De plus, la signature intégrant l'influence de tous les pixels de l'image, ces méthodes obtiennent de mauvais résultats en cas d'occultation et de présence de fond chargé.

c. Méthodes locales

Les méthodes locales [47,48] pallient les principaux défauts des méthodes globales. Elles ne considèrent plus les images comme un tout monolithique, mais comme une collection de régions locales, qui sont le plus souvent des parties d'images carrées ou rectangulaires, pouvant contenir une centaine de pixels ou bien plus. Ces régions locales s'affranchissent des difficultés rencontrées sur des images entières, car elles peuvent être ramenées à une apparence standard. Les modifications d'illumination sont localement uniformes, et la luminosité et le contraste peuvent donc être normalisés. Ces régions locales correspondent généralement à des surfaces planes, donc leur orientation et leur échelle peuvent être ramenée à des valeurs standard. Quand ces régions sont découpées à l'intérieur des objets, elles ne subissent pas d'influence du fond chargé, et en sont donc indépendantes. Enfin, quand ces régions sont de petites tailles, elles sont le plus souvent présentées intégralement ou bien totalement occultées, mais elles sont rarement partiellement occultées, et donc leur apparence ne varie pas à cause des occultations. Le principe de ces méthodes est de représenter les images de référence par des régions locales et de stocker les descripteurs de ces régions locales (par exemple leurs niveaux de gris) dans une base de données. Lors de la reconnaissance, les descripteurs des régions locales sélectionnées dans l'image sont recherchés dans la base de données, ce qui permet de voter pour les images de référence qui contiennent des descripteurs similaires. Schmid et al [80] utilisent des détecteurs de points d'intérêt pour obtenir des régions locales d'une orientation et échelle normalisées. Des contraintes géométriques sur les positions relatives des régions locales permettent de réduire le taux de fausse alarme. Lowe [81] propose un système temps réel et robuste pour rechercher les objets similaires dans une base de données, et introduit le fameux descripteur SIFT(Scale-Invariant feature transform), qui encode la distribution de l'orientation du gradient dans une région locale. Dans ce cas aussi, la présence des objets est vérifiée à l'aide de contraintes géométriques sur les positions relatives des régions locales. En raison de leur rapidité, de leur bonne gestion des occultations, du fond chargé et des modifications de point de vue, les méthodes locales ont aujourd'hui la préférence de la communauté pour la reconnaissance visuelle d'instances d'objets.

3.1.3. Principales contributions

Nous avons travaillé sur les trois facteurs qui peuvent influencer le rendement des réseaux de neurones :

- **La forme de base de l'image :**
 - En ce qui concerne la représentation des objets 3D, on a choisi une approche 2D/3D où chaque objet est représenté par une série d'images 2D qui représentent ce que l'on appelle les vues caractéristiques.
 - Concernant les bases de couleurs : depuis longtemps, la base de couleurs RGB est utilisée comme un standard dans le domaine de la reconnaissance ; vu que la quasi-totalité des bases d'apprentissage est en RGB (Read, Green, Blue). D'après nos recherches il n'y avait aucune études sur l'influence de la base de couleur sur les taux de reconnaissance, à part celle de l'usage de la base HSV (Hue, Saturation, Value or rightness) dans la détection de la peau humaine. À ce sujet, nous avons réalisé une étude comparative entre les différentes bases de couleurs existantes et Nous avons proposé une base de couleurs que nous avons appelée Expanded-LAB : une base qui a permis une amélioration de 4% des résultats obtenus par la base RGB mais avec un temps d'apprentissage qui a augmenté de 24%.
- **Les descripteurs de formes**

Pour ce qui est des descripteurs de formes, nous avons essayé d'intégrer le maximum de descripteurs existants et de les combiner afin de trouver l'entrée idéale pour les réseaux de neurones intégrés.
- **Structure du classificateur**

Concernant la classification, nous avons travaillé sur les différentes architectures et topologies des réseaux de neurones existants pour arriver au concept d'un réseau de neurones multicouches, multi-sorties ; qui permet de détecter, en parallèle, la couleur de l'objet, la forme générale et l'objet lui-même.

4. Conclusion

Nous avons présenté dans ce chapitre un état de l'art sur la reconnaissance automatique de formes. En effet, ce chapitre décrit le principe, l'architecture d'un système de reconnaissance et les différentes techniques utilisées dans le domaine de RF. Nous avons

analysé chaque composante du système de reconnaissance et nous avons présenté les différentes méthodes et techniques qui y sont développées et utilisées. La dernière partie de ce chapitre présente, séparément, l'état de l'art et les principales contributions dans la reconnaissance des caractères et dans la reconnaissance des objets réels.

Chapitre 2 :

Descripteurs Classiques

1. Introduction

L'extraction des paramètres caractéristiques « features extraction » est une étape déterminante dans le processus de reconnaissance. Une extraction mal conçue rendra difficile, voire impossible, d'effectuer une reconnaissance efficace.

Dans ce chapitre, nous allons présenter les principales méthodes d'extraction couramment utilisées aussi bien dans la reconnaissance des caractères que dans la reconnaissance d'objets complexes.

Notations : Le terme « descripteur » désigne donc l'outil qui permet de décrire une image. Un nombre N d'images est disponible. Chacune est notée I_i , avec $i \in [1, \dots, N]$ le numéro de l'image. Un descripteur fournit alors pour une image I_i un vecteur de caractéristiques notées V_i de taille K . chaque composante k de ce vecteur est donc une caractéristique de l'image I_i – ou « *feature* » en anglais - qui est notée $V_i(k)$ avec $k \in [1, \dots, K]$.

2. Descripteurs de Base

Parmi les méthodes classiques souvent utilisées, on trouve les histogrammes de projection et le zonage.

2.1. Histogrammes de projection

Les histogrammes de projection sont depuis très longtemps utilisés dans le domaine de la reconnaissance de formes. Leur principe est de sommer le nombre de pixels noirs de chacune des lignes (respectivement des colonnes) de l'image binaire de la forme (figure 2.1). Les projections des histogrammes ont été introduites en 1956 par Glauber [49], cette technique est principalement utilisée pour la segmentation des caractères, des mots et de lignes de texte, ou pour détecter si une image d'un texte numérisée est tournée. La projection horizontale (respectivement verticale) est le nombre de pixels 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 [50].

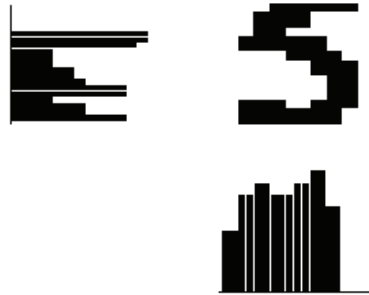


Figure 2.1. Histogramme horizontal et vertical du chiffre 5 (Matlab Lib).

2.2.Zoning (densités)

Le Zoning c'est la densité de pixels noirs calculés dans différentes zones (zoning) de l'image. Pour obtenir ces mesures, on "découpe" horizontalement et verticalement le rectangle englobant le chiffre en zones de taille égale; le nombre de pixels noirs 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), on obtient une matrice à NxM composantes [50].

Malgré leur caractère descriptif, le Zoning et les histogrammes de projection ne permettent pas une description fiable des formes complexes (perte d'informations, sensibilité à la rotation, etc....)

3. Descripteurs invariants

3.1.Moments géométriques

Les moments sont souvent utilisés en physique pour décrire la répartition des masses dans un corps. En associant le niveau de gris d'un point de l'image à la masse d'un corps en chaque point, on peut reprendre le même formalisme pour décrire la répartition des niveaux de gris dans un objet. Dans le cas d'un objet binaire où $g(k,l)$ prend la valeur « 1 » à l'intérieur (si le point appartient à l'objet) et « 0 » à l'extérieur, les différents moments fournissent des informations importantes concernant l'arrangement spatial des points de l'objet [51,52].

Donc on peut définir, comme en mécanique et en statistique, une double suite de coefficients appelés les moments géométriques. Le principe d'utilisation des moments est de sélectionner un sous-ensemble de M_{ij} qui contienne suffisamment d'informations pour caractériser l'image de façon unique, pour une application donnée.

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 coordonnée k,l ; i et j étant deux entiers naturels quelconques :

Soit $I(x, y)$ est le niveau de gris d'un pixel de l'image I , le moment [16] de $(p + q)$ pour $(p, q) > 0$ d'une image I est notée:

$$m_{p,q} = \int_{R^2} x^p y^q I(x, y) dx dy \quad (2.1)$$

Le barycentre de la fonction I est donné par $(x_0 = m_{1,0} / m_{0,0}, y_0 = m_{0,1} / m_{0,0})$. L'image centrée $I_T(x, y) = I(x + x_0, y + y_0)$ est invariant par translation.

Le moment central d'ordre $(p + q)$ de la fonction I est donné par:

$$v_{p,q} = \int_{R^2} x^p y^q I(x + x_0, y + y_0) dx dy \quad (2.2)$$

Les moments centraux sont invariants. Le moment normalisé est défini comme suit:

$$\mu_{p,q} = \frac{v_{p,q}}{v_{0,0}^\gamma} \quad (2.3)$$

$$\text{Où : } \gamma = 1 + (p + q) / 2$$

Ces moments sont invariants par translation, rotation et homothétie. Les moments de Hu sont calculés au moyen des moments normalisés:

$$\begin{aligned} \phi_1 &= \mu_{2,0} + \mu_{0,2} \\ \phi_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \\ \phi_3 &= (\mu_{3,0} - 3\mu_{1,2})^2 + (3\mu_{2,1} - \mu_{0,3})^2 \\ \phi_4 &= (\mu_{3,0} + \mu_{1,2})^2 (\mu_{2,1} + \mu_{0,3})^2 \\ \phi_5 &= (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2})[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2] + (3\mu_{2,1} - \mu_{0,3})^2. \end{aligned} \quad (2.4)$$

3.2. Moments de Zernike

Les moments géométriques fournissent un codage compact et facile à calculer, doté de bonnes propriétés d'invariance théorique. En fait, ce codage reste grossier si on conserve peu de termes et on n'a pas de méthodes permettant de déterminer le nombre idéal de moments pour un problème donné. Ce dernier inconvénient est surmonté par les moments de Zernike qui possèdent une propriété d'orthogonalité permettant la reconstruction (en général approchée) de la forme à partir des moments [53,54].

L'intérêt du calcul des moments de Zernike repose sur l'invariance par translation, par changement d'échelle et par rotation. Ils constituent un espace vectoriel dans lequel l'image de la forme est projetée. Cela permet de s'affranchir du problème de normalisation rencontré dans le cas des moments de Hu, mais aussi d'aboutir à une description plus précise des formes, par utilisation d'ordres plus élevés, en maintenant les propriétés d'invariance.

Les polynômes de Zernike furent proposés en 1934 par Zernike [55]. Dérivés de ces polynômes, les moments ont été utilisés par de nombreux auteurs en reconnaissance de caractères. Plusieurs études montrent également la supériorité de ces descriptions par rapport à d'autres approches [53,54], car elles réduisent le bruit, les redondances et peuvent être reconstruites.

Par définition, le moment géométrique d'une fonction $f(x, y)$ est la projection de cette fonction sur l'espace des polynômes engendrés par $x^p y^q$, $(p, q) \in \mathbb{N}^2$. Zernike introduit un ensemble de polynômes complexes qui forment une base orthonormale définie à l'intérieur du cercle unitaire, c'est-à-dire tel que $x^2 + y^2 \leq 1$. La forme de ces polynômes est la suivante :

$$V_{nm}^*(x, y) = V_{nm}(\rho, \theta) = R_{nm}(\rho) \cdot \exp(jm\theta) \quad (2.5)$$

Où :

n : un entier positif ou nul

m : un entier tel que $|m| \leq n$

r : longueur du vecteur entre l'origine et le pixel (x, y)

θ : angle entre le vecteur p et x

R_{nm} : polynôme radial

$V^*(x, y)$: polynôme complexe projection de $f(x, y)$ sur l'espace des polynômes complexes.

De tels polynômes sont bien orthogonaux puisque :

$$\int_{x^2+y^2 \leq 1} \int [V_{nm}^*(x, y)] V_{pq}(x, y) dx dy = 0 \quad \text{ou} \quad 1 \quad (2.6)$$

Le moment géométrique de Zernike est donc la projection d'une fonction $f(x, y)$ décrivant une image sur un espace de polynômes orthogonaux engendré par $V_{nm}^*(\rho, \theta)$:

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y f(x, y) \cdot V_{nm}^*(\rho, \theta) dx dy \quad (2.7)$$

Et pour l'identification de l'image, on utilise les valeurs absolues des Moments de Zernike :

$$|A_{nm}| = \sqrt{\text{Re}^2(A_{nm}) + \text{Im}^2(A_{nm})} \quad (2.8)$$

3.3.Descripteurs de Fourier génériques

La transformée de Fourier a été largement utilisée dans le traitement et l'analyse d'images. L'avantage d'analyser une image dans le domaine spectral sur l'analyse de la forme dans le domaine spatial est qu'il est facile de pallier au problème du bruit qui est fréquent dans les images numériques [56]. Comme première étape, l'image d'entrée $f(x,y)$ est transformée en une image polaire $f(r, \theta)$

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}, \quad (2.9)$$

Où (x_c, y_c) sont les coordonnées du centre de gravité. Ensuite, une transformation de l'image polaire échantillonnée en trame dans le système des coordonnées cartésiennes est réalisée. Autrement dit, ces images sont créées par un échantillonnage circulaire de l'objet/surface dans l'image. Alors, une Transformation de Fourier 2-D sur l'image transformée est calculée :

$$\text{TF}_2(\rho, \phi) = \sum_r \sum_i f(r, \theta_i) \exp \left[-j2\pi \left(\frac{r}{R} \rho + \frac{i}{T} \phi \right) \right] \quad (2.10)$$

Où $0 \leq \rho < R$ et $\theta_i = i(2\pi/T) (0 \leq i < T)$; $0 \leq \rho < R, 0 \leq \phi < T$. R et T sont respectivement, les résolutions radiales et angulaires.

Enfin, les Descripteurs de Fourier Génériques (DFG) [52] sont calculés à partir de FP_2 normalisés et sont donnés comme suit :

$$DFG = \left\{ \frac{|TF_2(0,0)|}{\pi \times r_{\max}^2}, \frac{|TF_2(0,1)|}{|TF_2(0,0)|}, \dots, \frac{|TF_2(0,n)|}{|TF_2(0,0)|}, \dots, \frac{|TF_2(m,0)|}{|TF_2(0,0)|}, \dots, \frac{|TF_2(m,n)|}{|TF_2(0,0)|} \right\} \quad (2.11)$$

Où m et n sont respectivement, le nombre maximal des fréquences radiales et angulaires sélectionnées et r_{\max} est le rayon maximal de l'objet.

3.4. Transformée de Radon,

Étant donnée une fonction f à valeurs réelles dans \mathfrak{R}^2 , on définit la transformée de Radon par la fonctionnelle [57]:

$$f^\vee(p, \phi) = \int_L f(x, y) ds, \quad L \in L4 \quad (2.12)$$

Où L est une ligne spécifiée par des paramètres réels p et ϕ tels que $L = \{(x, y) \in \mathfrak{R}^2 : x \cos \phi + y \sin \phi = p\}$ et ds est une différentielle de la longueur le long de L de l'ensemble des lignes $L4$ sur \mathfrak{R}^2 .

Nous obtenons toutes les lignes possibles en prenant $p \geq 0$, $-\pi < \phi \leq \pi$ ou en prenant $-\infty < p < \infty$, $0 \leq \phi \leq \pi$. En d'autres termes, l'expression de la transformée de Radon d'une fonction f est donnée par

$$f^\vee(p, \phi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \phi + y \sin \phi - p) dx dy \quad (2.13)$$

Où $\delta(\cdot)$ est la fonction delta de Dirac ($\delta(x) = 1$ si $x = 0$ et $\delta(x) = 0$ si $x \neq 0$).

Maintenant, supposons qu'un nouveau système d'axes est introduit où les axes de x et de y ont pivoté d'un angle ϕ . Si les nouveaux axes sont marqués par p et s , alors

$$\begin{cases} x = p \cos \phi - s \sin \phi \\ y = p \sin \phi + s \cos \phi \end{cases} \quad (2.14)$$

Et par conséquent, nous avons une forme plus explicitée de la transformée :

$$f^\vee(p, \phi) = \int_{-\infty}^{\infty} f(p \cos \phi - s \sin \phi, p \sin \phi + s \cos \phi) ds \quad (2.15)$$

Dans le domaine de la reconnaissance de forme, f représente une forme plane binaire. Ainsi, il est d'un intérêt particulier de considérer le cas dans lequel la fonction générale f est remplacée par une fonction particulière f_D où

$$f_D(x, y) = \begin{cases} 1 & \text{si } (x, y) \in D \\ 0 & \text{ailleurs} \end{cases} \quad (2.16)$$

Où D est le domaine de la forme binaire.

À partir des définitions, les propriétés de la périodicité et de la symétrie peuvent être aisément obtenues :

$$f^\vee(p, \phi) = f^\vee(p, \phi \pm 2k\pi) \quad (2.17)$$

$$f^\vee(p, \phi) = f^\vee(-p, \phi \pm (2k+1)\pi) \quad (2.18)$$

Où k étant un entier relatif. Puisque la transformée de Radon est linéaire par définition, des propriétés géométriques telles que les lignes droites ou des courbes peuvent être mise en évidence explicitement par la transformée de Radon qui concentre les énergies à partir de l'image dans quelques coefficients à valeurs élevées dans le domaine transformé. En effet, chaque pixel (x, y) dans f_D est balayé par N_ϕ lignes à partir de plusieurs directions, ce qui garantit que la transformée est non seulement sans perte, mais aussi riche en informations.

Pour surmonter les problèmes de rotations, de translation et de changement d'échelle, une première adaptation originale de la transformée de Radon a été proposé dans [43], dont la définition est donnée par :

$$\mathfrak{R}_f(\phi) = \int_{-\infty}^{\infty} (f^\vee(p, \phi))^2 dp \quad (2.19)$$

Cette fonction est appelée la \mathfrak{R} -transformée ou \mathfrak{R} -signature.

4. Descripteurs de Couleur

Parmi tous les descripteurs, La couleur est la plus représentative des caractéristiques visuelles. L'histogramme couleur est fréquemment utilisé pour caractériser la couleur des images. Cependant, un espace couleur doit être choisi pour la représenter. Souvent il s'agit de l'espace HSV (Hue, Saturation, Value) ou YCbCr. En effet, il est connu que les différentes couleurs de l'espace HSV coïncident approximativement avec celle de la perception humaine des différentes couleurs. L'espace YCbCr est l'espace le plus utilisé dans les formats d'images et se comporte mieux que l'espace RGB : c'est donc souvent un espace de compromis. La comparaison entre deux histogrammes de deux images est réalisée par un

calcul de distance. Les différentes méthodes sont issues du choix de la distance et de la façon de représenter un histogramme [58].

En général, les histogrammes de couleur quantifiés subissent une transformation pour avoir un descripteur plus compact. Par exemple, en appliquant la transformée de Haar sur l'histogramme couleur suivie d'une quantification binaire des coefficients de la transformée. Le descripteur résultant est un vecteur caractéristique de 63 bits [59]. Les principales descripteurs de couleur sont :

- ***Le descripteur SCD (Scalable Color Descriptor)***

Le SCD est fixé dans l'espace HSV avec une quantification uniforme de l'espace HSV. Les valeurs des pixels subissent une quantification non linéaire pour réaliser un encodage efficace. La transformée de Haar [59] est ensuite appliquée sur ces valeurs pour rendre le descripteur compact et avoir une représentation multiéchelle de l'histogramme. L'intérêt de cette multirésolution est le passage du grossier au raffinement. Ce descripteur ne tient pas compte de la structure locale de la couleur. Cette information pertinente fait l'objet du descripteur CSD.

- ***Le descripteur CSD (Color Structure Descriptor)***

Ce descripteur exprime la structure locale de la couleur dans une image en utilisant un élément structurant 8×8 . On compte le nombre de blocs possédant une couleur particulière. L'histogramme $h(m)$ représente le nombre d'éléments structurant 8×8 contenant un ou plusieurs pixels avec la couleur c_m . Comme l'élément structurant est gardé fixe à 64, l'étendue spatiale de l'élément structurant est à l'échelle de l'image originale [58].

- ***Le descripteur DCD (Dominant Color Descriptor)***

Il permet d'obtenir les couleurs dominantes dans une région d'intérêt. Les couleurs dans l'image ou dans une région donnée sont d'abord regroupées en un petit nombre de couleurs représentatives. Le pourcentage de chacune des couleurs est ensuite calculé ainsi qu'une mesure de cohérence spatiale qui différencie les couleurs étendues sur toute l'image des couleurs localisées. La cohérence spatiale est un simple nombre qui représente l'homogénéité spatiale globale des couleurs dominantes dans l'image. Le nombre de couleurs par région est borné à 8 couleurs dominantes. L'extraction des couleurs dominantes repose sur l'algorithme généralisé de Lloyd [58].

- **Le descripteur CLD (Color Layout Descriptor)**

Ce descripteur renseigne sur la distribution spatiale de la couleur dans l'image. Il constitue un descripteur efficace pour la recherche basée sur le croquis. Ce descripteur utilise une grille 8*8 qui permet d'extraire les couleurs dominantes selon la disposition spatiale. Le procédé d'extraction des caractéristiques s'effectue en 2 étapes : la première est de sélectionner la couleur représentative de chaque cas de la grille 8*8 et ensuite d'effectuer la transformée en cosinus discret DCT (Discret Cosinus Transform) sur la grille résultante. Il est d'ailleurs recommandé de prendre la couleur moyenne pour chaque bloc. L'espace de couleur choisi est le YCbCr [58]:

- **Spatial Structure Histograms (SSH) :**

Ce descripteur est un mélange de plusieurs mesures. Dans un premier temps, l'algorithme cherche les couleurs dominantes de l'image (algorithme des K-moyens) puis à partir de ces couleurs dominantes, il calcule un histogramme de position pour ces différentes couleurs, d'autres notions sont également incluses, comme l'orientation, mais aussi la taille des taches de couleur.

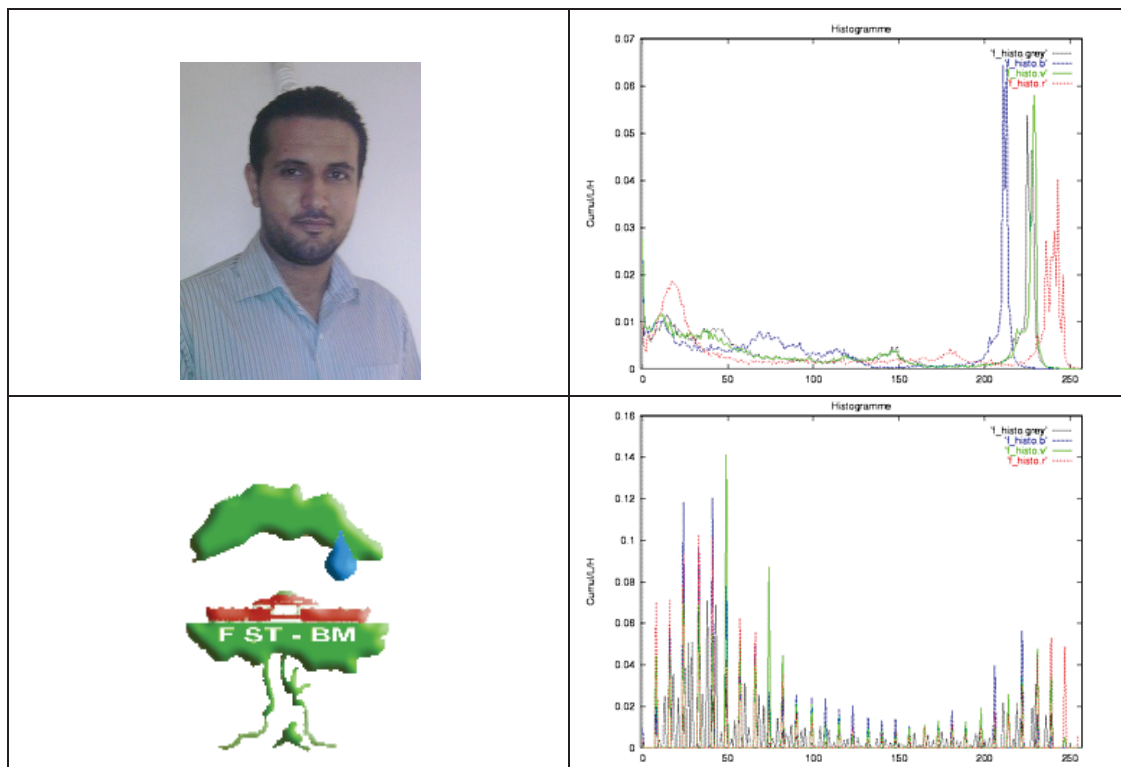


Figure 2.2 : Photographies et histogrammes associés (les quatre composantes R, V, B, L).

5. Extraction des segments de contours

Intuitivement, dans une image numérique, les contours se situent entre les pixels appartenant à des régions ayant des intensités moyennes différentes ; il s'agit de contour de type « saut d'amplitude » [59].

Une manière d'aborder le problème de l'extraction de contours est de considérer que l'image numérique résulte de l'échantillonnage d'une fonction scalaire $A(x,y)$ à support borné et dérivable en tout point. Globalement, on peut dire que :

Les approches dérivatives sont les plus immédiates pour détecter et localiser les variations du signal. Les contours sont assimilés aux points de fort gradient ou de dérivée seconde nulle

Dans les approches surfaciques, l'image des intensités est considérée comme une surface. La transition entre deux régions est modélisée par un gabarit utilisable à deux fins :

- Un contour est présent quand la mise en correspondance entre le gabarit et la zone de l'image est bonne.
- L'approximation par facettes de la surface fournit une équation analytique locale qui permet de calculer de manière très précise (sub-pixel) la position du contour et ses caractéristiques.
- Toutes ces approches sont sensibles au bruit, c'est pourquoi l'image est souvent prétraitée avant la segmentation à proprement dite par un filtre de Lissage.

5.1.Modèles de Contours

Le contour est une frontière de variation d'intensité. Plusieurs types de contours existent dans une image (Figure 2.3). Des bruits divers sont présents : bruit convolutif, bruit additif, bruit blanc, etc. La plupart des méthodes de détection de contours se focalisent sur les contours de type marche (discontinuité d'ordre 0), d'amplitude A , avec un bruit blanc $n(x)$ additif de moyenne nulle et de variance n_0^2 . Le signal d'entrée correspondant au contour est alors représenté par : $I(x) = Au_{-1}(x) + n(x)$ avec $u_{-1}(x)$ la fonction échelon.

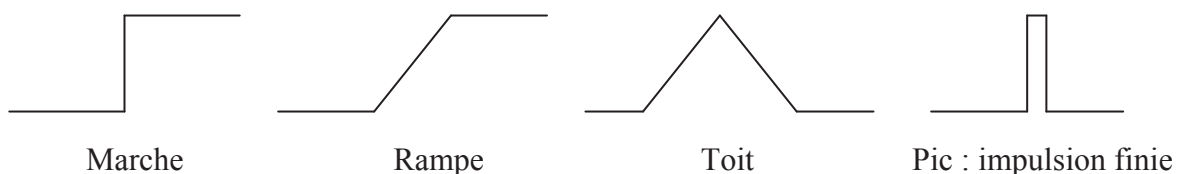


Figure 2.3: Quelques profils de contours

5.2.Approches de Détection

Comme il a été mentionné auparavant, un contour est une variation d'intensité. Il peut être détecté classiquement de deux manières (Figure 2.4) :

1. Calcul du gradient et extraction des maxima locaux de la norme du gradient
2. Calcul du Laplacien et détermination des passages par zéros.



Figure 2.4: Approche gradient et Laplacien pour la détection de contour

Le *gradient* d'une image bidimensionnelle $I(x,y)$ est donné par :

$$G_x(x, y) = \partial I(x, y) / \partial x \qquad G_y(x, y) = \partial I(x, y) / \partial y \qquad (2.20)$$

En chaque pixel (x,y) de l'image, le module et l'angle du vecteur du gradient se calculent comme suit :

$$G = \sqrt{G_x^2 + G_y^2} \approx \max\{G_x, G_y\} \approx |G_x| + |G_y| \text{ et } \alpha = \arctan(G_y / G_x) \in [0, \pi[\quad (2.21)$$

Le gradient ne donne que des contours larges dont il ne faut garder que les maxima dans la direction du gradient.

$$\text{Le Laplacien est calculé par : } L(x,y) = \partial^2 I(x, y) / \partial x^2 + \partial^2 I(x, y) / \partial y^2 \qquad (2.22)$$

Le laplacien est un scalaire, il ne fournit donc pas la direction du contour.

La détection des maxima locaux du gradient revient à détecter les passages par zéro du Laplacien.

5.3.Extraction de Contours. Opérateurs dérivatifs du 1° Ordre

- *Transformation de type Gradient :*

Le contour d'orientation θ au point (x,y) est détecté par un maximum de la dérivée première directionnelle dans la direction Φ du gradient $\vec{\nabla}f(x,y)$ [1].

Pour les orientations O_x, O_y :

$$\|\vec{\nabla}f(x,y)\| = \sqrt{(\Delta x f^2 + \Delta y f^2)} \approx \max(|\Delta x f|, |\Delta y f|) \tag{2.23}$$

• **Opérateur de Roberts :**

Il s'agit d'une approximation du gradient par convolution de l'image avec 2 masques 2*2 suivant des axes à 45°.

$$F_x = f * H_1 \quad \text{et} \quad F_y = f * H_2$$

L'image obtenue possède des contours accentués.

$$F_a = \|\vec{\nabla}f(x,y)\| = \sqrt{f_x^2 + f_y^2} \tag{2.24}$$

$$\text{et } \theta = \pi/4 + \arctan((\partial f / \partial y) / (\partial f / \partial x)) \tag{2.25}$$

• **Opérateur de Prewitt et Sobel :**

Il y a approximation des dérivées directionnelles par convolution de l'image par 2 masques 3*3 suivant O_x et O_y . Soit I_a l'image accentuée approximant la norme du gradient,

$$I_a = \|\vec{\nabla}f(x,y)\| \quad \text{et} \quad \theta = \arctan((\partial f / \partial y) / (\partial f / \partial x)) \tag{2.26}$$

La différence entre Prewitt et Sobel réside dans les valeurs des coefficients des masques. Les contours obtenus sont épais d'où l'utilisation d'une suppression des non maxima locaux suivis d'une hystérésis.

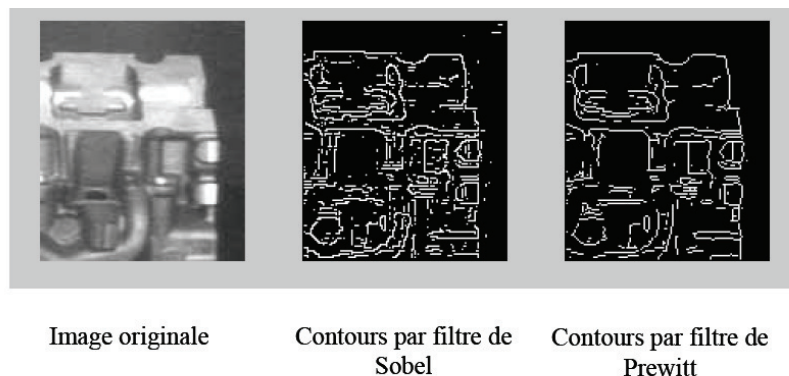


Figure. 2.5 Méthodes dérivatives par différences finies

6. Conclusion

Chacune des méthodes déjà citées possède des avantages et présente des inconvénients. Le choix du descripteur dépend principalement de la nature et de la complexité de l'image. L'extraction des descripteurs reste un champ de recherche très active. Dans les chapitres qui suivent, on va présenter trois nouveaux descripteurs basés sur la fameuse géométrie riemannienne.

Classification

1. Introduction

Le choix du classificateur est primordial. Il constitue l'élément de décision dans un système de reconnaissance des formes. Le but du deuxième chapitre, était d'extraire des informations caractérisant des objets pour les présenter sous la forme la plus condensée possible au classificateur. Dans ce chapitre nous allons essayer de mettre l'accent sur la classification supervisée en présentant quatre classificateurs : Les Réseaux de neurones, les réseaux Neuro-Flou, les Séparateurs à Vaste Marge et les Arbres de décision. La classification est un processus de prise de décision où l'utilisateur connaît les informations concernant l'appartenance aux classes des vecteurs d'apprentissage. Nous disposons donc d'un ensemble de vecteurs étiquetés qui vont être présentés au classificateur afin que celui-ci puisse en déduire une règle de décision. Les sections suivantes présentent plusieurs modèles de réseaux de neurones artificiels, ainsi qu'une analyse de leurs aptitudes à agir en tant que classificateur. Nous allons achever ce chapitre par une présentation des SVM que nous avons choisis comme classificateurs témoin des résultats obtenus par les réseaux de neurones.

2. Réseaux de neurones

2.1. Historique

Dupliquer l'intelligence humaine, constitue sans aucun doute le rêve ultime des chercheurs dans le domaine de l'intelligence artificielle. Les premiers travaux sur les Réseaux de Neurones Artificiels (RNA) ont été menés par Mc Culloch et Pitts en 1943 [71]. L'intégration de la loi de Hebb qui décrit la manière dont les cellules apprennent à modifier l'intensité des connexions a permis le succès du perceptron de Rosenblatt [79]. Le triomphe des RNA a connu une chute dans les années 70, mais au début des années 80, les travaux de plusieurs chercheurs (Figure 3.1), ont permis de relancer les travaux de recherche dans ce domaine.

Actuellement l'utilisation des réseaux de neurones devient une réalité et les recherches continuent. Plusieurs articles ont été publiés sur les apports réciproques des R.N [70,71,72 etc.].

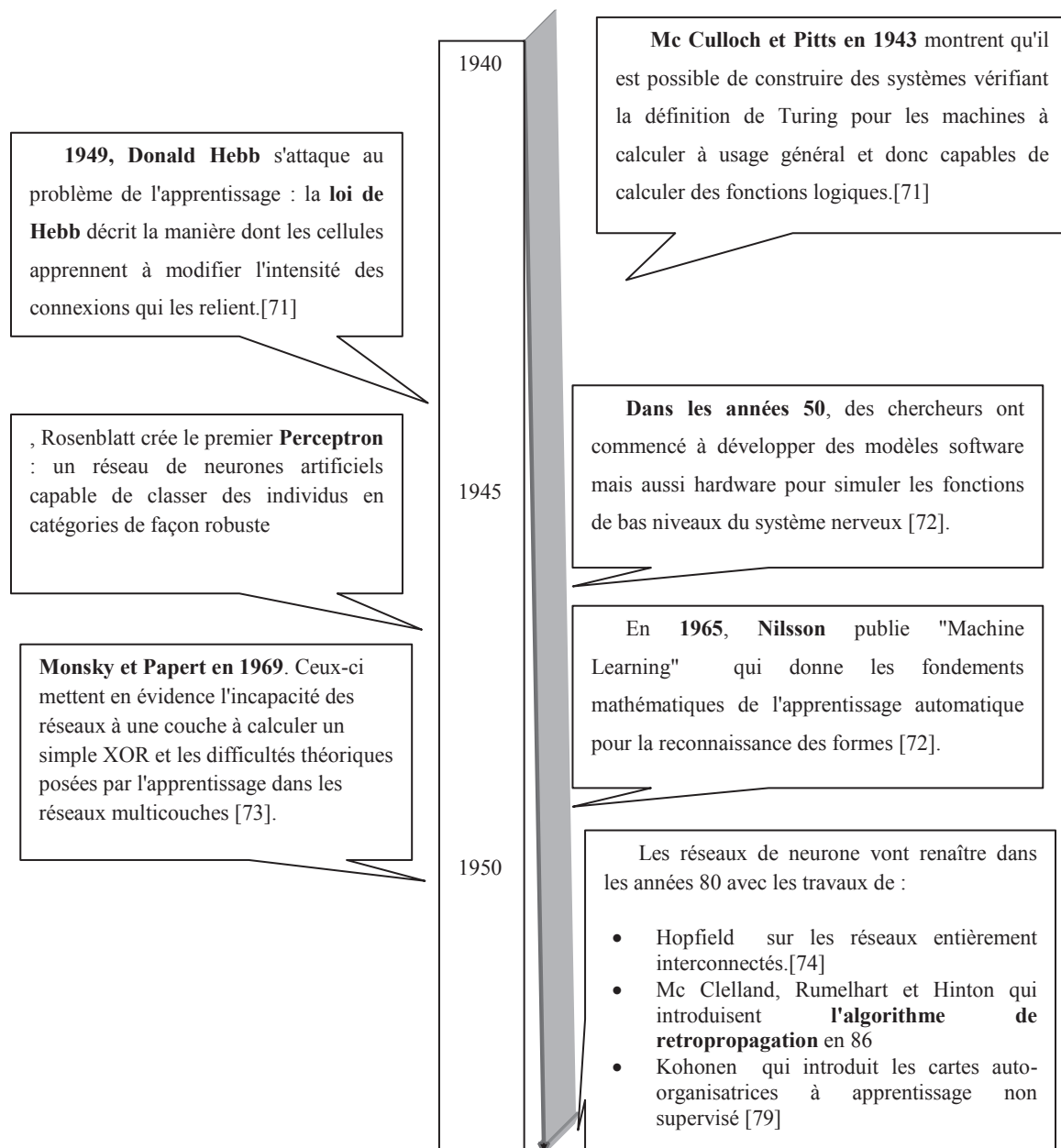


Figure 3.1 : Evolution des RNA entre 1940 et 1990.

2.2.Principes

La structure d'un neurone artificiel est inspirée de la structure des neurones biologiques. Les principales structures biologiques des neurones ont toutes leurs équivalents artificiels (voir plus bas). Le but est de reproduire le fonctionnement du neurone biologique de la meilleure façon possible (d'une manière logique, simple et facilement représentable sur ordinateur). Un neurone est une minuscule structure qui traite les influx nerveux qui arrivent (*inputs*), chacun selon son importance relative, et qui émet un (et un seul) signal de sortie (*output*). Les neurones artificiels reproduisent le même procédé, recevant chaque signal d'entrée (*input*) tempéré par un poids (*weight*). Ces poids sont aussi appelés poids synaptiques, par analogie. Les intrants pondérés, habituellement sommés sont ensuite comparés à un seuil d'activation θ et passés dans la fonction du neurone (habituellement une fonction très simple), qui produit l'extrait (*output*) désiré (Tableau 3.1). Dans les neurones biologiques, les intrants et extrants sont des influx électriques, représentés artificiellement par des valeurs numériques. Habituellement, les neurones sont reliés ensemble en réseaux, les neurones d'un niveau (les réseaux de neurones sont habituellement bâtis en niveaux hiérarchiques) passant leurs extrants aux neurones du niveau suivant (devenant les intrants des neurones suivants). D'une manière générale, un neurone constitue un minuscule processeur, capable de gérer une fonction simple.

| Neurone biologique | Neurone artificiel | Représentation symbolique |
|--------------------|---------------------------|---------------------------|
| Soma | Neurone | |
| Dendrite | Intrant (<i>Input</i>) | x_i , X (intrant net) |
| Axone | Extrant (<i>Output</i>) | Y |
| Synapse | Poids | w_i |

Tableau 3.1 : Analogie entre les neurones biologiques et artificiels

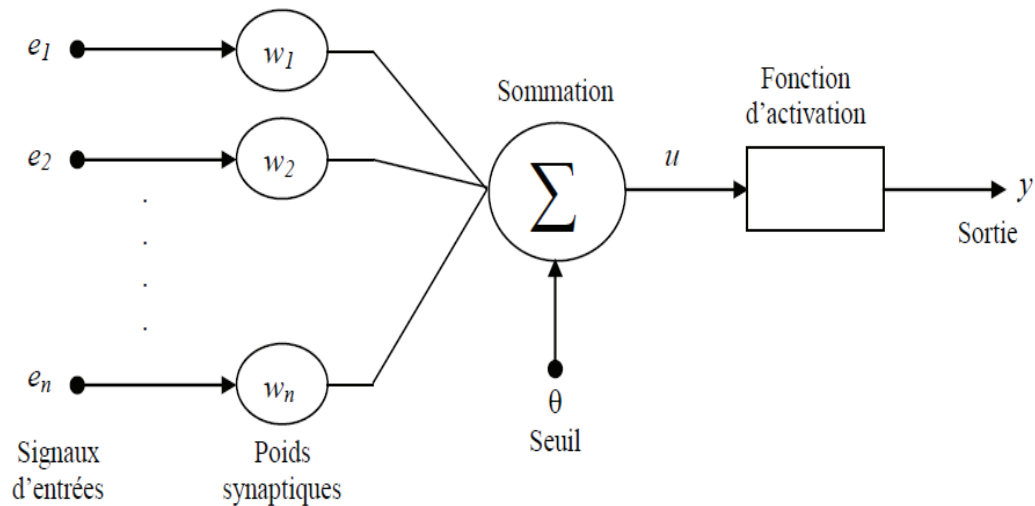


Figure 3.2 : Schéma d'un neurone artificiel

Dans le texte qui suit, des symboles et des variables sont utilisées pour représenter les composantes des réseaux de neurones. Voici les définitions de ce symbolisme.

- S : Système que représente le réseau de neurones, de l'extérieur
- X : Vecteur d'intrant du réseau, ou d'une manière locale, vecteur d'intrant d'une couche du réseau.
- x_i : i-ème élément du vecteur d'intrant **X**
- Y : Vecteur de sortie du réseau, ou d'une manière local, vecteur de sortie produit par une couche du réseau.
- y_i : i-ème élément du vecteur de sortie **Y**
- w_i : Poids du i-ème intrant du neurone
- w_{ij} : Poids sur le lien entre le neurone i et le neurone j
- W_i : Vecteur de poids associé au neurone i
- W : Matrice de poids, associée à une couche.
- $f()$: Fonction d'activation d'un neurone
- Θ : Seuil d'activation d'un neurone

Fonctions d'activation

La fonction d'activation ou bien la fonction de transfert est une fonction qui renvoie un réel appartenant généralement à l'intervalle $[0,1]$. Plusieurs fonctions peuvent être utilisées pour l'activation [79] ; le tableau 3.2 présente quelques formes courantes de fonction d'activation.

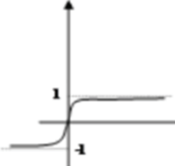
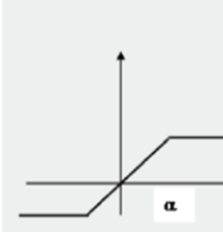
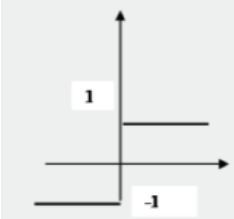
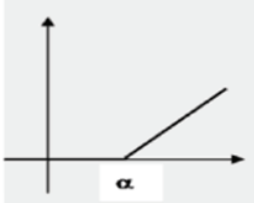
| | | |
|---------------------------|---|---|
| Sigmoïde |  | $\frac{e^{kNET} - 1}{e^{kNET} + 1}$ |
| Fonction linéaire bornée |  | $- \alpha \text{ si } NET < - \alpha$ $OUT = NET \text{ si } NET \in [-\alpha, \alpha]$ $\alpha \text{ si } NET > \alpha$ |
| Fonction signe |  | $OUT = \text{signe}(NET)$ |
| Fonction positive à seuil |  | $OUT = NET$ $- \alpha \text{ si } NET > \alpha$ |

Tableau. 3.2 : Exemple de fonctions d'activation.

2.3. Architecture d'un Réseau de Neurones

D'une manière générale, un réseau de neurones se comporte, d'un point de vue extérieur, comme une fonction S qui traite des données (*inputs*) et produit une réponse correspondante (*output*). Les données entrées peuvent être de n'importe quel type représentable de manière binaire ou numérique. Ces données peuvent également être vues comme des vecteurs, et le

réseau de neurones est une application vectorielle [75]. Il y a deux catégories de RNA statiques. La première concerne les réseaux entraînés, c'est-à-dire les réseaux à apprentissage supervisé. La seconde catégorie concerne les réseaux non évolutifs, comme les réseaux de mémoire. Les poids de ces réseaux sont choisis (et non pas modifiés par un entraînement), pour donner l'effet escompté [72].

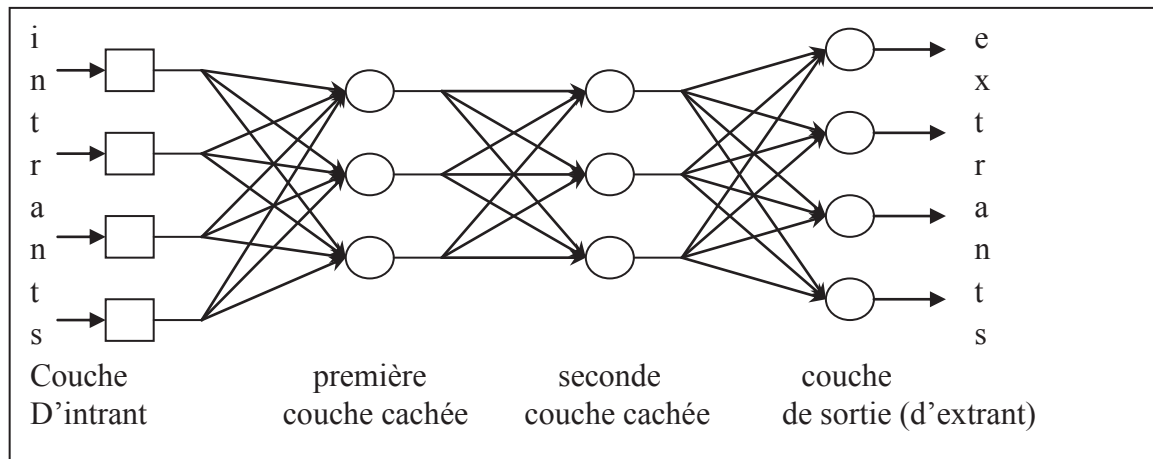


Figure 3.3 : Architecture d'un réseau de neurones (avec deux couches cachées).

2.4. Perceptron

Pour bien comprendre le comportement collectif d'un ensemble de neurones, nous allons présenter le Perceptron (un seul neurone) en phase d'utilisation. L'apprentissage ayant été réalisé,

On suppose que l'apprentissage est réalisé et les poids sont fixes. Le neurone de la figure 3.2 réalise une simple somme pondérée de ses entrées, compare une valeur de seuil, et fournit une réponse binaire en sortie. Par exemple, on peut interpréter sa décision comme classe 1 si la valeur de x est $+1$ et classe 2 si la valeur de x est -1 .

Algorithme de traitement

a. Initialisation

-Mettre les poids initiaux w_1, w_2, \dots, w_n ainsi que le seuil θ à des valeurs aléatoires de l'intervalle $[-0,5, 0,5]$.

- Mettre le taux d'apprentissage α à une petite valeur positive.

b- Activation

-Activer le perceptron en appliquant les intrants $x_1(p)$, $x_2(p)$, $x_n(p)$ et l'extrant désiré $Y_d(p)$.

-Calculer l'extrant actuel à l'itération $p = 1$ tel que :

$$Y(p) = \text{étage} \left[\sum_{i=1}^n x_i(p)w_i(p) - \theta \right] \quad (3.1)$$

Où n est le nombre de signaux intrants, et étage est la fonction d'activation par étage.

c- Entraînement des poids

-Mettre à jour les poids du perceptron

$$w_i(p+1) = w_i + \Delta w_i \quad (3.2)$$

Où $\Delta w_i(p)$ est la correction de poids à l'itération p .

-La correction de poids est calculée par la loi delta :

$$\Delta w_i = \alpha \times x_i(p) \times e(p) \quad (3.3)$$

où α est le taux d'apprentissage et $e(p)$ l'erreur à l'itération p (la différence entre l'extrant désiré et l'extrant actuel du perceptron)

d- Itération

-Augmenter p de 1,

- Retourner à l'étape 2

- Répéter le procédé jusqu'à convergence.

Cependant, les perceptrons à un neurone ne peuvent apprendre que des fonctions séparables linéairement. Dans le cas présenté ci-haut, la fonction apprise est séparable par la droite $x_1w_1 + x_2w_2 = \theta$; [71].

2.5. Réseaux Multicouche (Multilayer Neural Networks)

Les réseaux de neurones multicouches sont habituellement bâtis selon le modèle « normalisé » et comprennent 3 ou 4 couches en tout (donc 1 ou 2 couches cachées). S'il est

théoriquement possible de construire des réseaux avec un très grand nombre de couches cachées, les réseaux comprenant plus de couches cachées sont très rares, étant donné que chaque nouvelle couche augmente la quantité de calculs d'une manière exponentielle. La plupart des réseaux de neurones multicouches sont, dans la pratique, des Perceptrons MultiCouches (PMC).

2.5.1. Apprentissage des RNA

La méthode de modification des poids est très simple avec l'algorithme de Rosenblatt, mais il implique quelques limitations d'apprentissage [64]. Dans le cas de perceptrons multicouches, puisqu'on ne connaît pas les sorties désirées des couches cachées, mais seulement de la dernière couche, il faut propager la responsabilité des erreurs de la dernière couche à la première, dans le sens contraire de l'exécution de réseau, d'où le nom rétro-propagation. De plus, les perceptrons multicouches utilisent des neurones munis de la fonction d'activation sigmoïde, celle-ci permettant les nuances nécessaires à une bonne utilisation de la rétro-propagation. Voici les principales lignes de la méthode de rétro-propagation :

- Calcul de l'erreur :

Elle est donnée par : $e_k(p) = y_{d,k} - y_k$: l'erreur du neurone k est la différence entre la valeur de l'extrant désiré et la valeur actuelle de l'extrant du neurone k .

- Correction de poids (par la loi delta):

Elle est donnée par : $\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p)$: la correction du poids du lien du neurone j au neurone k , où $\delta_k(p)$ est le gradient d'erreur du neurone k à l'itération p .

- Gradient d'erreur (pour les neurones de la couche de sortie) :

$$\delta_k(p) = F'[X_k(p)] \times e_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p) \quad (3.4)$$

La deuxième partie est obtenue par la dérivée de la fonction sigmoïde appliquée à l'intrant net du neurone k à l'itération p .

- Gradient d'erreur (pour les neurones d'une couche cachée) :

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^l \delta_k(p) w_{j,k}(p) \quad (3.5)$$

Où l est le nombre de neurones sur la couche suivante.

Algorithme de traitement

a. Initialisation

-Mettre tous les poids et les seuils d'activation du réseau à des valeurs aléatoires uniformément distribuées dans un petit intervalle :

$\left(-\frac{2.4}{F_i}, \frac{2.4}{F_i}\right)$ ([50], où F_i est le nombre total de signaux intrants du neurone i dans le réseau. Cette initialisation est faite un neurone à la fois.

-Mettre la valeur du taux d'apprentissage à une petite valeur positive.

b- Activation

-Activer le réseau de neurones en appliquant les signaux intrants et les extrants désirés

-Calculer les signaux sortant des neurones des couches successives, de la première couche cachée à la couche de sortie.

c- Entraînement des poids

Mettre à jour les poids du réseau en propageant les erreurs dans le sens inverse :

-Calculer les gradients d'erreurs pour les neurones des couches successives (ordre inverse), de la couche de sortie à la première couche cachée.

-Calculer les corrections de poids pour chaque lien et mettre à jour les poids.

d- Itération

-Incrémenter p , retourner à l'étape 2 et répéter le procédé jusqu'à ce que le critère d'erreur soit atteint. La méthode de rétro-propagation peut mettre énormément de temps avant que la sommation des erreurs carrées ne soit réduite à zéro (ceci peut être impossible selon les cas). C'est pourquoi on utilise plutôt un critère d'erreur, soit une certaine valeur critique, qui doit être atteinte par la sommation des erreurs carrées (habituellement entre 0,001 et 0,1). La loi delta, utilisée par la méthode de rétro-propagation, a pour objectif de réduire la sommation des erreurs carrées produites par le réseau. Si on représente la courbe de la fonction de la somme des erreurs carrées par rapport à un poids du réseau, on peut observer que la variation idéale dépend de la pente de la courbe de la fonction, d'où l'utilisation de la dérivée dans le calcul du gradient d'erreur (et par le fait même, de la variation de poids).

L'entraînement d'un perceptron multicouche avec l'algorithme de rétro-propagation du gradient se fait d'une manière similaire à l'entraînement d'un perceptron à neurone unique, mais il faut procéder neurone par neurone et poids par poids, en commençant par la dernière couche. Il faut calculer le gradient d'erreur pour chaque neurone, puis calculer la variation de

pois pour les liens rattachés à ce neurone. Ensuite on passe à la couche précédente. Cette méthode devient très rapidement lourde en calculs [79].

La grande majorité des PMC sont des réseaux avec apprentissage supervisé, mais il en existe quelques-uns qui peuvent apprendre sans professeur pour les guider. Une classe de ces réseaux non-supervisés est celle des réseaux auto-associatifs. Par défaut, un PMC est un réseau hétéro-associatif, ce qui veut dire que la sortie est différente de l'entrée. Dans le cas particulier où l'entrée doit être la même que la sortie, ces réseaux étant appelés auto-associatifs, on n'a pas besoin des valeurs de sortie que devraient produire le PMC, celles-ci étant les mêmes que l'entrée (Figure 3.3) [74].

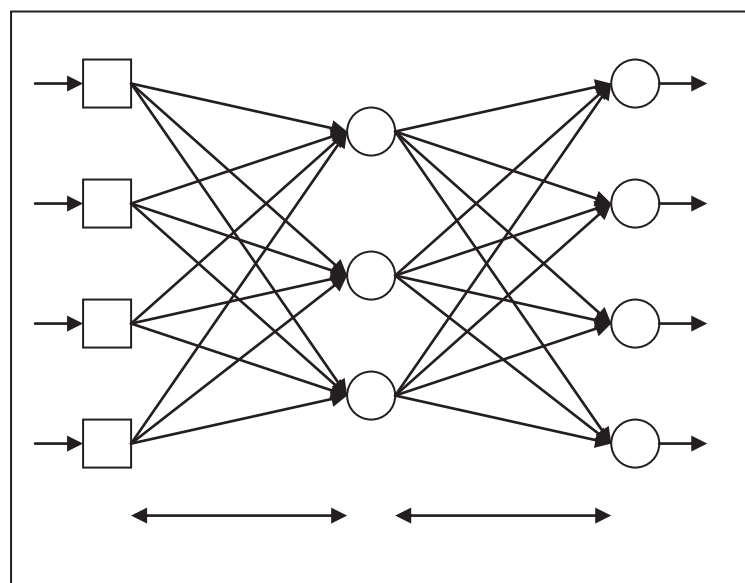


Figure 3.3 : Exemple de PMC auto-associatif

Quelques améliorations ont été proposées pour augmenter la vitesse d'apprentissage des PMC :

a) L'utilisation de la fonction tangente hyperbolique donnée par :

$$Y^{\tan h} = \frac{2a}{1 + e^{-bX}} - a \quad (3.6)$$

Où a et b sont constants [71]. L'avantage de cette fonction par rapport à la fonction sigmoïde de base a été testé empiriquement.

b) L'utilisation d'un terme de momentum (*momentum term*) dans l'équation de la loi delta exprimée par: $\Delta w_{jk}(p) = \beta \times \Delta w_{jk}(p-1) + \alpha \times y_j(p) \times \delta_k(p)$ (3.7)

Avec $0 \leq \beta < 1$. Cette nouvelle équation porte le nom de loi delta généralisée. L'ajout du terme de momentum permet de stabiliser l'apprentissage, tout en augmentant la descente de la courbe de la somme des erreurs carrées selon les itérations.

c) L'utilisation d'un taux d'apprentissage variable plutôt que constant. Si le changement dans la somme des erreurs carrées porte le même signe algébrique, le taux d'apprentissage devrait être augmenté. Si le signe algébrique du changement de la somme des erreurs carrées alterne d'une itération à l'autre, le taux d'apprentissage devrait être décrétementé.

3. Réseau Neuro-flou (Fuzzy-Neural)

Un réseau Neuro-flou peut être défini comme un système flou formé avec un algorithme dérivé de la théorie des réseaux de neurone. L'intégration des réseaux de neurones et des systèmes flous vise la génération d'un système robuste, efficace et facilement interprétable où les avantages de chaque modèle sont maintenus et leurs inconvénients possibles sont supprimés. Certains modèles de réseaux neurones [79,81] ont été appliqués avec succès à la formation de réseaux Neuro-flous.

3.1. De la logique floue aux réseaux Neuro-Flou

La logique floue n'est pas une nouvelle découverte. En effet, c'est en 1937 que le philosophe Max Black avait pressenti ce concept qui a été réellement introduit en 1965 par Lotfi Zadeh [83]. Par contre, le concept n'a pas connu un succès rapide. À vrai dire, ce concept n'est arrivé à maturité qu'en 1990. Au départ la technologie floue n'a été exploitée qu'en Europe et au Japon. De nos jours, l'usage de la logique floue est de plus en plus populaire. Il y a des machines à laver qui utilisent la logique floue, ce qui leur permet de déterminer, par principe d'automatisation, la durée du cycle par la quantité de linge qu'elle détecte elle-même. Cette machine met elle-même la quantité de savon, ce qui permet une économie de savon en plus d'une économie d'électricité et d'eau. Il y a aussi des trains haute vitesse qui utilisent ce type de programme pour permettre un freinage plus adéquat et plus doux tout en améliorant la durabilité des freins. Si on combine les RNA et la logique flou, nous obtenons la science de la neuro-floue. Cette science est l'avenir. À vrai dire, elle constitue la base de l'intelligence artificielle moderne [84].

3.2. Architecture des réseaux Neuro-flous

Le premier type d'utilisation, le plus répandu, est représenté par le cas où un système d'inférence floue mis sous la forme d'un réseau multicouche, dans lequel généralement les poids correspondent aux paramètres du système; une telle architecture est appelée architecture pré-neuronale. Par exemple, pour des règles de la forme : si V1 est Ai1 et V2 est Ai2 alors W est wi. On fait appel à un réseau de neurones qui admet pour entrées les valeurs x1 et x2 prises par les variables V1 et V2 et dont les deux couches cachées correspondent respectivement au calcul de la valeur des fonctions d'appartenances Ai1 pour x1 et Ai2 pour x2, et à celui de la valeur prise par la conjonction des conditions de chaque règle utilisant un opérateur de conjonction adéquate.

Les fonctions d'appartenance intervenant dans les règles sont considérées comme des paramètres ajustés par l'intermédiaire des poids entrant dans la première couche cachée. Les conclusions des règles sont également des paramètres ajustables par l'intermédiaire des poids associés à la dernière couche. Les données vont subir trois étapes de calcul :

➤ **La première couche** : s'occupe du calcul des degrés d'appartenance de chaque entrée. Les paramètres de cette couche vont caractériser la fonction d'appartenance. Le calcul des degrés d'appartenance s'effectue par des neurones spécialisés, dont la fonction d'activation est la fonction d'appartenance. La fonction la plus utilisée est la fonction gaussienne ; elle est donnée par :

$$y = \exp(-(ax + b)^2) \quad (3.8)$$

Avec : $a = \frac{\sqrt{-\log_{10}(\epsilon)}}{dx}$ et $b = -a * m$

➤ **La deuxième couche cachée**: évalue en parallèle les règles de la couche. Les paramètres de cette couche définissent dans ce cas l'opérateur de conjonction ET.

Généralement on utilise celle qui a été proposée par Lukasiewicz [85]. Cette dernière présente une certaine facilité d'implémentation neuronale moyennement satisfaisante. Elle est définie par :

$$ET[fA_1(x_1), fA_2(x_2)] = \text{Max}(0, fA_1(x_1) + fA_2(x_2) - 1) \quad (3.9)$$

Si on pose :

$$u = f_{A_1}(x_1) + f_{A_2}(x_2) - 1 \quad (3.10)$$

On aura :

$$ET[f_{A_1}(x_1), f_{A_2}(x_2)] = \text{Max}(0, u) \quad (3.11)$$

La fonction $\text{Max}(0, u)$ peut être évaluée approximativement par une fonction continue sigmoïde, elle est donnée par :

$$f(u) = \frac{1}{1 + \exp\left(\frac{u-0.5}{0.227}\right)} \quad (3.12)$$

➤ **La dernière couche** : réalise l'opération d'inférence. Ayant calculé les valeurs de vérité produites par les opérateurs de conjonction, on doit maintenant établir une procédure permettant de déduire une sortie en fonction de ces valeurs de vérité. Cette procédure est réalisée soit en choisissant parmi ces valeurs, celle qui est la plus représentative, c'est à dire celle qui a le plus grand degré de vérité, soit en calculant le barycentre de toutes les valeurs. Une telle approche peut être effectuée par un seul neurone, où les valeurs de vérité sont pondérées par les poids synaptiques de ce neurone.

Les poids synaptiques du neurone qui calcule le barycentre (qui fait la défuzzification) sont d'une grande importance, car ce sont eux qui pondèrent les résultats des règles. Une retro propagation serait indispensable à appliquer sur cette couche (c'est-à-dire sur les poids synaptiques du neurone chargé de la défuzzification).

3.3. Algorithme d'apprentissage:

Cet algorithme est basé sur l'algorithme de la retropropagation du gradient descendant. Il est le même que celui qui a été utilisé pour l'apprentissage des réseaux de neurones type PMC. La seule différence qui existe est que les entrées du neurone chargé de la défuzzification seront normalisées.

Cette normalisation est nécessaire, car elle permet de manipuler des valeurs inférieures ou égales à l'unité. Une telle approche est réalisée par la division de chaque entrée de ce neurone, sur la somme de toutes ses entrées.

Les systèmes Neuro-flous modernes sont généralement représentés comme éléments spéciaux multicouches (par exemple les modèles comme des ANFIS [86], Fune [87], Fuzzy RuleNet [88]). Dans notre travail on a employé l'ANFIS méthode.

Le ANFIS (Adaptive Network based Fuzzy Inference System) où le Réseau adaptatif à système d'inférence floue, utilise un algorithme d'apprentissage hybride pour identifier les paramètres de la fonction d'appartenance de la sortie unique, des systèmes de type Sugeno d'Inférence Floue (SIF) [86]. Une combinaison d'au-carré et de rétro-propagation des méthodes de descente de gradient sont utilisées pour les paramètres de la formation des membres de la FIS et des fonctions pour modéliser un ensemble donné d'entrées / sorties de données. Le programme ANFIS est disponible avec la boîte à outils Matlab floue (Figure 3.5).

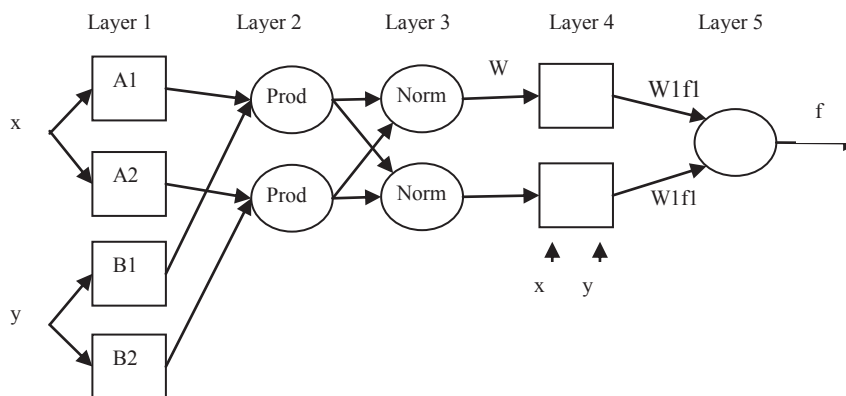


Figure 3.5 : Architecture ANFIS (De la Bibliothèque Matlab).

Algorithme ANFIS

Les règles du modèle de Sugeno:

- **Règle1:** Si x est A₁ et y est B₁, alors $f_1 = p_1x + q_1y + r_1$
- **Règle2:** Si x est A₂ et y est B₂, alors $f_2 = p_2x + q_2y + r_2$

Couche 1

$O_{l,i}$ est la sortie du nœud i de la couche L.

$$O_{1,i} = \mu_{A_i}(x) \text{ pour } i = 1, 2 \text{ ou}$$

$$O_{1,i} = \mu_{B_{i-2}}(x) \text{ pour } i = 3, 4$$

Avec:

X (ou Y) est le nœud d'entrée i et A_i (ou B_{i-2}) est un label linguistique associé à ce nœud.

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \quad (3.13)$$

μ est la fonction d'appartenance typique avec : a_i, b_i, c_i est l'ensemble des paramètres.

Les paramètres sont appelés paramètres prémisse.

Par conséquent $O_{1,i}$ est le grade d'appartenance à un ensemble flou (A_1, A_2, B_1, B_2).

Couche 2

- La sortie est le produit de tous les signaux entrants.
- $O_{2,i} = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y), i = 1, 2$

Couche 3

Les sorties de cette couche sont appelées forces de feu normalisé

- Le nœud calcul le rapport de force tir du de chaque i^{eme} neurone à la somme de tous les tirs.

$$O_{3,i} = \bar{w}_i = \frac{\omega_i}{\omega_1 + \omega_2}, i = 1,2 \quad (3.14)$$

Couche 4

- Chaque nœud i dans cette couche est un nœud avec une fonction adaptative:

$$O_{4,1} = w_i f_i = w_i (p_x + q_{iy} + r_i)$$

- w_i est la force normalisée de tir de la couche 3.
- $\{p_i, q_i, r_i\}$ est le jeu de paramètres de ce nœud.
- Le nœud unique ($O_{5,1}$) dans cette couche est une somme qui calcule la somme de tous les signaux entrants:

$$O_{5,1} = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (3.15)$$

4. Séparateurs à Vaste Marge(SVM)

Parmi les méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik [89]. Les SVM sont des classificateurs binaires et statiques. C'est-à-dire qu'ils permettent de créer une surface de décision entre deux classes définies dans un même espace. Pour cela, ils construisent une frontière de décision par projection des caractéristiques provenant d'un espace d'origine dans un espace de caractéristiques de dimension supérieure (voire infini) dans le but de rendre les classes linéairement séparable.

4.1. Notions de base : Hyperplan, marge et support vecteur

Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan.

Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points.

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

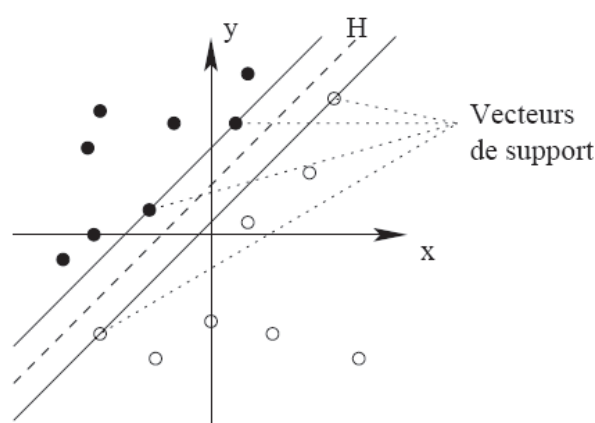


Figure 3.6 : Séparation par SVM

Il est évident qu'il existe une multitude d'hyperplans valide, mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les

hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le « plus sûr ». L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de *séparateurs à vaste marge*.

4.2. Linéarité et non-linéarité

Parmi les modèles des SVM, on constate les cas linéairement séparables et les cas non linéairement séparables. Les premiers sont les plus simples, car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels, il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé, car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables.

Pour surmonter les inconvénients des cas non linéairement séparables, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de dimension. Cette nouvelle dimension est appelée « espace de re-description ». En effet, intuitivement, plus la dimension de l'espace de re-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée. Ceci est illustré à la figure 3.7.

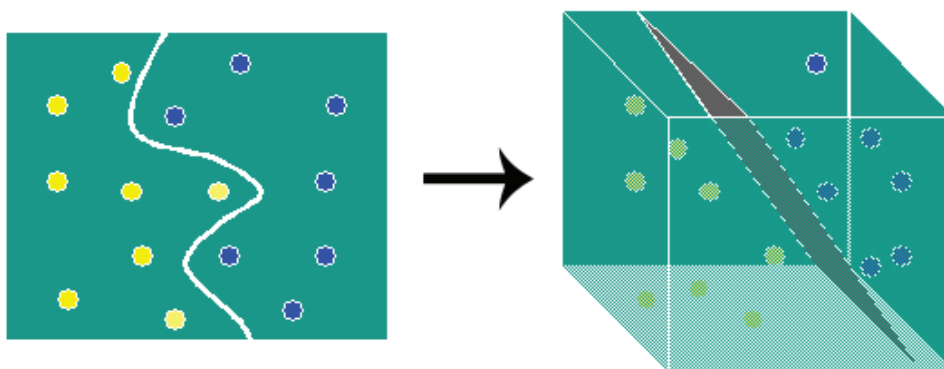


Figure 3.7 : Exemple d'Augmentation de l'espace de représentation [89].

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de re-description de plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau. En pratique, quelques familles de fonctions noyaux paramétrables sont connues et il revient à

l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde et laplacien.

4.3. Problème d'apprentissage

On s'intéresse à un phénomène f (éventuellement non déterministe) qui, à partir d'un certain jeu d'entrées x , produit une sortie $y = f(x)$.

Le but est de retrouver cette fonction f à partir de la seule observation d'un certain nombre de couples entrée-sortie $\{(x_i; y_i) : i = 1, \dots, n\}$ afin de « prédire » d'autres événements.

On considère un couple (X, Y) de variables aléatoires à valeurs dans $X \times Y$.

Seul le cas $Y = \{-1, 1\}$ (classification) nous intéresse ici (on peut facilement étendre au cas $\text{card}(Y) = m > 2$ et au cas $Y = \mathbb{R}$). La distribution jointe de (X, Y) est inconnue.

Sachant qu'on observe un échantillon $S = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ de n copies indépendantes de (X, Y) , on veut: construire une fonction $h : X \rightarrow Y$ telle que $P(h(X) \neq Y)$ soit minimale.

4.4. Classification à valeurs réelles

Plutôt que de construire directement $h : X \rightarrow \{-1, 1\}$, on construit :

$f : X \rightarrow \mathbb{R}$ (ensemble des réels).

La classe est donnée par le signe de f ;

$$h = \text{signe}(f).$$

L'erreur se calcule avec $P(h(X) \neq Y) = P(Yf(X) \leq 0)$. Ceci donne une certaine idée de la confiance dans la classification. Idéalement, $|Yf(X)|$ est proportionnel à $P(Y|X)$.

$Yf(X)$ représente la marge de f en (X, Y) .

Le but à atteindre est la construction de f et donc h .

4.4.1. Transformation des entrées

Il paraît donc nécessaire de transformer les entrées dans le but de les traiter plus facilement.

X est un espace quelconque d'objets.

On transforme les entrées en vecteurs dans un espace F (feature space) par une fonction :

$$\Phi : X \rightarrow F$$

F n'est pas nécessairement de dimension finie, mais dispose d'un produit scalaire (espace de Hilbert). L'espace de Hilbert est une généralisation de l'espace euclidien qui peut avoir un nombre infini de dimensions.

La non-linéarité est traitée dans cette transformation, on peut donc choisir une séparation linéaire (on verra plus loin comment on arrive à ramener un problème non linéaire en un problème linéaire classique).

Dès lors, il s'agit de choisir l'hyperplan optimal qui classe correctement les données (lorsque c'est possible) et qui se trouve le plus loin possible de tous les points à classer. Mais l'hyperplan séparateur choisi devra avoir une marge maximale.

4.4.2. Maximisation de la marge

La marge est la distance du point le plus proche à l'hyperplan.

Dans un modèle linéaire, on a $f(x) = w \cdot x + b$. L'hyperplan séparateur (frontière de décision) a donc pour équation $w \cdot x + b = 0$.

La distance d'un point au plan est donnée par $d(x) = |w \cdot x + b| / \|w\|$

L'hyperplan optimal est celui pour lequel la distance aux points les plus proches (**marge**) est maximale. Soient x_1 et x_2 deux points de classes différentes :

$$f(x_1) = +1 \text{ et } f(x_2) = -1$$

$$(w \cdot x_1) + b = +1 \text{ et } (w \cdot x_2) + b = -1 \text{ donc } (w \cdot (x_1 - x_2)) = 2$$

$$\text{D'où : } (w / \|w\| \cdot (x_1 - x_2)) = 2 / \|w\|.$$

On peut donc en déduire que maximiser la marge revient à minimiser $\|w\|$ sous certaines contraintes que nous verrons dans les paragraphes suivants.

4.4.3. Problème primal

Un point $(x; y)$ est bien classé si et seulement si $y.f(x) > 0$

Comme le couple (w, b) est défini à un coefficient multiplicatif près, on s'impose $y.f(x) \geq 1$

On en déduit (en s'appuyant également sur le paragraphe précédent), le problème de minimisation sous contraintes suivantes :

$$\begin{cases} \min \frac{1}{2} \|\omega\|^2 \\ \forall i, y_i(\omega \cdot x_i + b) \geq 1 \end{cases} \quad (3.16)$$

Il peut être en effet plus aisé de minimiser $\|\omega\|^2$ plutôt que directement $\|\omega\|$.

4.4.4. Problème dual

On passe du problème primal au problème dual en introduisant des multiplicateurs de Lagrange pour chaque contrainte.

Ici on a une contrainte par exemple d'apprentissage

$$\begin{cases} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j & ; \forall i, \alpha_i \geq 0 \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (3.17)$$

C'est un problème de programmation quadratique de dimension n (nombre d'exemples).

On définit ainsi la matrice suivante appelée « **matrice Hessienne** » : $(x_i \cdot x_j)_{i,j}$ qui représente la matrice des produits des entrées X (la notation matricielle permettant de résoudre plus facilement le problème en informatique).

On montre que si les α_i^* sont solutions de ce problème alors on a :

$$w^* = \sum_{i=1}^n \alpha_i^* y_i x_i \quad (3.18)$$

Seuls les α_i^* correspondant aux points les plus proches sont non-nuls. On parle de *vecteurs de support*.

La fonction de décision associée est donc :

$$f(x) = \sum_{i=1}^n \alpha_i^* y_i x_i \cdot x + b \quad (3.19)$$

Il existe néanmoins des cas où on ne peut pas classer les entrées de façon linéaire.

4.5. Non-linéarité (cas non séparable/ marge molle)

On part du problème primal linéaire et on introduit des variables « ressort » pour assouplir les contraintes.

$$\begin{cases} \min \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n \xi_i \\ \forall i, y_i (\omega \cdot x_i + b) \geq 1 - \xi_i \end{cases} \quad (3.15)$$

On pénalise par le dépassement de la contrainte.

On en déduit le problème dual qui a la même forme que dans le cas séparable:

$$\begin{cases} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j & ; \forall i, 0 \leq \alpha_i \leq C \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (3.16)$$

La seule différence est la borne supérieure C sur les α .

4.5.1. Fonction noyau

Dans le cas linéaire, on pouvait transformer les données dans un espace où la classification serait plus aisée. Dans ce cas, l'espace de redescription utilisé le plus souvent est \mathbf{R} (ensemble des nombres réels). Il se trouve que pour des cas non linéaires, cet espace ne suffit pas pour classer les entrées. On passe donc dans un espace de grande dimension.

$$\begin{aligned} \Phi : \mathbb{R}^d &\rightarrow \mathcal{F} \\ x &\rightarrow \Phi(x) \end{aligned}$$

Avec $\text{card}(\mathcal{F}) > d$.

Le passage dans $\mathcal{F} = \mathbb{R}^3$ rend possible la séparation linéaire des données.

On doit donc résoudre :

$$\begin{cases} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) & ; \forall i, 0 \leq \alpha_i \leq C \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (3.17)$$

Et la solution a la forme :

$$f(x) = \sum_{i=1}^n \alpha_i^* y_i \Phi(x_i) \cdot \Phi(x) + b \quad (3.18)$$

Le problème et sa solution ne dépendent que du produit scalaire $\Phi(x) \cdot \Phi(x')$.

Plutôt que de choisir la transformation non-linéaire $\Phi : X \rightarrow F$, on choisit une fonction $k : X \times X \rightarrow \mathbf{R}$ (nombres réels) appelée *fonction noyau*.

Elle représente un produit scalaire dans l'espace de représentation intermédiaire. Du coup k est linéaire (ce qui nous permet de faire le rapprochement avec le cas linéaire des paragraphes précédents). Cette fonction traduit donc la répartition des exemples dans cet espace $k(x, x') = \Phi(x) \cdot \Phi(x')$.

Lorsque k est bien choisi, on n'a pas besoin de calculer la représentation des exemples dans cet espace pour calculer Φ .

5. Les Arbres de décisions

5.1. Notions de base

Les arbres de décision sont une représentation commode de *fonctions de classifications*, moins puissantes que les règles de production, mais plus faciles à utiliser.

Un arbre de décision permet de classer un objet à l'aide de questions : chaque nœud de l'arbre représente une question, chaque lien est une réponse à la question, et chaque feuille est une classe.

Un arbre est composé de :

- Nœuds: attributs
- Branches: valeurs
- Feuilles: décisions (classes)

Les avantages des arbres de décision sont :

- Facile à comprendre et à utiliser.
- Nombre de tests limités par le nombre d'attributs (de question).
- Construction efficace (mais technique) à l'aide d'apprentissage par optimisation

Algorithme d'implémentation (cas binaire)

Nous supposons que toutes les questions ont comme réponse *oui* ou *non*. L'algorithme d'implémentation est le suivant :

Procédure Construit_arbre(X)

Début

Si tous les points de X sont dans la même classe alors

Créer une feuille de cette classe

Si non

choisir le meilleur *sélecteur* (la meilleure question)

pour créer un noeud. Séparer X suivant ce sélecteur en X_d et X_g .

Construit_arbre(X_d)

Construit_arbre(X_g)

Fin si

Fin

5.2. Un exemple introductif

Pour mieux comprendre le fonctionnement des arbres de décisions l'exemple standard de la question [40] : Un écolier peut-il aller jouer dehors ? Voilà ce qu'il a observé :

| Devoir finis ? | Mère de bonne humeur ? | Fait beau ? | Gouter pris ? | Réponse |
|----------------|------------------------|-------------|---------------|---------|
| Oui | Non | Oui | Non | Oui |
| Non | Oui | Non | Oui | Oui |
| Oui | Oui | Oui | Non | Oui |
| Oui | Non | Oui | Oui | Oui |
| Non | Oui | Oui | Oui | Non |
| Non | Oui | Non | Non | Non |
| oui | Non | Non | Oui | Non |
| oui | oui | non | non | non |

Tableau 3.3 : Exemple de tableau de vérité

*

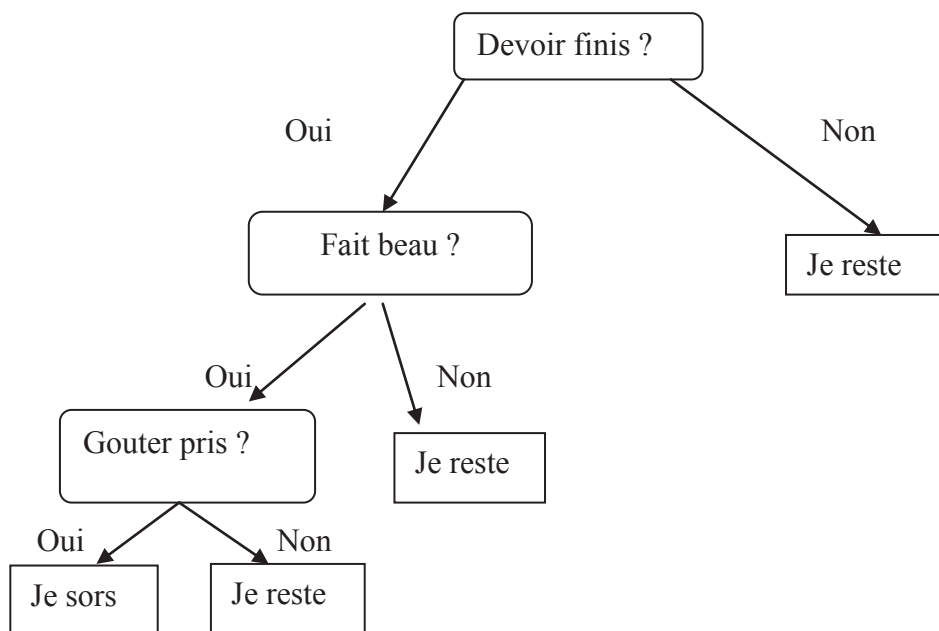


Figure 3.8 : Exemple d'Arbre de décision.

Un arbre de décision est la représentation graphique d'une procédure de classification. En effet, à toute description complète est associée une seule feuille de l'arbre de décision. Cette association est définie en commençant à la racine de l'arbre et en descendant dans l'arbre selon les réponses aux tests qui étiquettent les nœuds internes.

- **Choix de la question**

On dispose de 4 questions (DF, MBH, FB, GP). Chaque question sépare les réponses positives (4 en tout) et négatives (4 en tout) en deux ensembles.

Par exemple, avec la question « Devoirs finis ? », on trouve trois réponses positives et deux réponses négatives pour OUI, et une réponse négative et deux réponses positives pour NON.

L'idée est de maximiser la séparation globale. Les formules utilisées dérivent de la théorie de l'information : on cherche à maximiser le gain d'information.

- **Une mesure pour choisir l'attribut.**

L'entropie d'une variable ω pouvant prendre les valeurs $\{\omega_i\}_i$ avec des probabilités $p(\omega_i)$ est définie comme :

$$H(\omega) = -\sum_{i \in I} \mathcal{P}(\omega_i) \log(\mathcal{P}(\omega_i)) \quad (3.19)$$

L'entropie note l'*incertitude* sur la valeur de la variable. Par exemple, initialement l'entropie de "Je peux aller jouer" est (en considérant le logarithme à base 2), $H(c) = 1$. Si on sait que les devoirs ne sont pas finis, elle devient $H(c) = 0,8$. La *décroissance* de l'entropie note une diminution de l'incertitude, donc une augmentation de l'information.

Calcul des entropies conditionnées pour chaque question :

1. Pour DF :

$$H = -\frac{3}{8} * \log \frac{3}{5} - \frac{2}{8} * \log \frac{2}{5} - \frac{1}{8} * \log \frac{1}{3} - \frac{2}{8} * \log \frac{2}{3} = 0.95$$

2. Pour FB :

$$H = -2 * \frac{3}{8} * \log \frac{3}{4} - 2 * \frac{1}{8} * \log \frac{1}{4} = 0.81$$

3. Pour MBH :

$$H = -\frac{2}{8} * \log \frac{2}{4} - \frac{2}{8} * \log \frac{2}{4} - \frac{3}{8} * \log \frac{3}{4} - \frac{1}{8} * \log \frac{1}{4} = 0.91$$

4. Pour GP : $H=1$

Pour construire un nœud dans l'arbre, on cherche parmi les attributs celui qui a la meilleure entropie croisée avec la distribution des points d'apprentissage sur les classes.

5.3.Règle de classification

L'Idée centrale est de Diviser récursivement les exemples de l'ensemble d'apprentissage par des tests définis à l'aide des attributs jusqu'à ce que l'on obtienne des sous-ensembles d'exemples ne contenant que des exemples appartenant tous à une même classe.

Dans toutes les méthodes, on trouve les trois opérateurs suivants :

1. **Décider si un nœud est terminal**, c'est-à-dire décider si un nœud doit être étiqueté comme une feuille. Par exemple : tous les exemples sont dans la même classe, il y a moins d'un certain nombre d'erreurs, ...
2. **Sélectionner un test à associer à un nœud**. Par exemple : aléatoirement, utiliser des critères statistiques, ...
3. **Affecter une classe à une feuille**. On attribue la classe majoritaire sauf dans le cas où l'on utilise des fonctions coût ou risque.

Les méthodes vont différer par les choix effectués pour ces différents opérateurs, c'est-à-dire sur le choix d'un test (par exemple, utilisation du gain et de la fonction entropie) et le critère d'arrêt (quand arrêter la croissance de l'arbre, soit quand décider si un noeud est terminal). Le schéma général des algorithmes est le suivant :

Algorithme d'apprentissage générique

entrée : langage de description ; échantillon S

Début

Initialiser à l'arbre vide ; la racine est le noeud courant

Répéter

Décider si le noeud courant est terminal

Si le noeud est terminal **alors**

Affecter une classe

Sinon

Sélectionner un test et créer le sous-arbre

FinSi

Passer au noeud suivant non exploré s'il en existe

Jusqu'à obtenir un arbre de décision

Fin

6. Conclusion

Dans ce chapitre, nous nous sommes intéressés à quatre classificateurs connus pour leur robustesse et leurs caractéristiques uniques: les SVM, les RNA, les réseaux Neuro-Flou et les Arbres de décision. Nous avons présenté les principaux algorithmes et théories utilisés par les quatre classificateurs, les qualités des classificateurs présentés nous permettront une évaluation efficace des descripteurs de formes que nous allons présenter dans les chapitres qui suivent.

Chapitre 4:

Développement de nouveaux descripteurs

Dans ce chapitre, nous proposons trois nouvelles approches d'extraction basées sur les principes de la géométrie riemannienne à savoir: les descripteurs métriques, les descripteurs géodésiques et les descripteurs de courbure. Nous allons d'abord présenter les fondements théoriques sur lesquels sont basés ces descripteurs.

1. Notions de base de la géométrie riemannienne

Les variétés munies d'une structure riemannienne (dites aussi variétés riemanniennes) sont des outils très puissants de la géométrie différentielle. Elles figurent dans de nombreux contextes afin de compenser l'insuffisance de la géométrie euclidienne. On les utilise par exemple dans le calcul de la longueur d'un chemin s'inscrivant sur une sphère, l'estimation d'un volume, la génération de maillage adaptatif ou la représentation d'objets 3D. Dans une vue d'exploiter la richesse et les multiples possibilités qu'offrent ces outils, nous avons essayé de les adapter à la reconnaissance des caractères Tifinagh.

La notion de variété illustre le cas où l'espace est courbé ou a une topologie compliquée, mais qui reste localement assimilable à un espace \mathbb{R}^n . L'assimilation ne signifie pas que la métrique est la même, mais que les notions de base de l'analyse (ensembles ouverts, fonctions et coordonnées) sont les mêmes. La variété complète est alors reconstituée en raccordant sans discontinuité toutes ces régions locales, à savoir la sphère, le tore et le plan (Figure 4.1) [91].

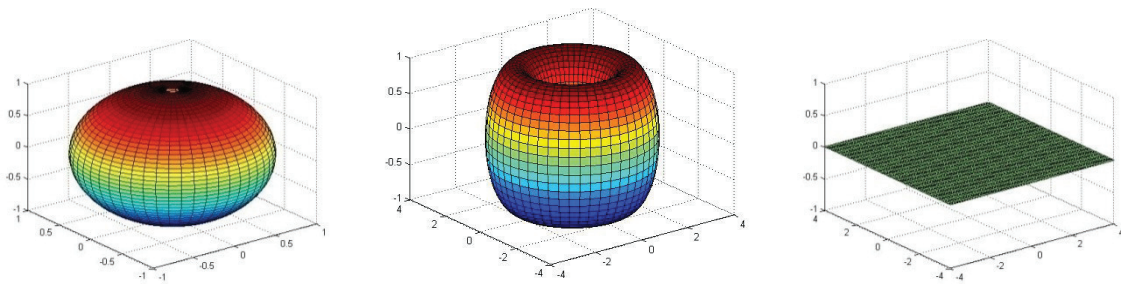


Figure 4.1 : Exemple de variété géométrique (sphère, tore, plan).

Dans ce qui suit, nous présentons les définitions de base relatives à la géométrie riemannienne [91].

1.1. Variétés C^∞ -différentiables

- **Définition 1**

Soit un recouvrement ouvert de M . Une C^∞ partition de l'unité de M

$\{f_\beta : M \rightarrow \mathbb{R}, f_\beta(x) \geq 0\}_{\beta \in J}$ est dite subordonnée à $\{U_\alpha\}_{\alpha \in I}$ si $\forall \beta \in J, \exists \alpha \in I$ tel que

$\text{supp } f_\beta \subset U_\alpha$.

- **Définition 2**

On appelle Métrique riemannienne la donnée d'un produit scalaire défini sur l'espace tangent à une variété C^∞ -différentiable M en chaque point.

i.e $g : T_p M * T_p M \rightarrow \mathbb{R}$

$$(X, Y) \rightarrow g(X, Y) \quad (4.1)$$

Où $g(X, Y)$ est le produit scalaire de vecteurs X et Y au point p .

1.2. Variété riemannienne

- **Définition 3 :**

Une variété C^∞ différentiable munie d'une métrique riemannienne g est appelée Variété riemannienne [91].

Remarque : Le produit scalaire étant une forme bilinéaire symétrique définie positive, la métrique riemannienne est donc un tenseur d'ordre 2 appelé aussi tenseur métrique.

Dans un système de coordonnées locales $(x^1 \dots x^n)$ on a en écriture d'Einstein

$$X = X^i \frac{\partial}{\partial x^i} \text{ et } Y = Y^j \frac{\partial}{\partial x^j} \quad (4.2)$$

D'où :

$$g(X, Y) = g\left(X^i \frac{\partial}{\partial x^i}, Y^j \frac{\partial}{\partial x^j}\right) = g\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right) X^i Y^j \quad (4.3)$$

$$g(X, Y) = g_{ij} X^i Y^j = g_{ij} dx^i \otimes dx^j(X, Y) \quad (4.4)$$

Où :

$$g_{ij} = g\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right) \quad (4.5)$$

Nous tirons :
$$g = g_{ij} dx^i \otimes dx^j \quad (4.6)$$

Dans la plupart des cas on impose également à g d'être non dégénéré c.-à-d que le déterminant de la matrice (g_{ij}) est supposé non nul en tout point p . Et on peut inverser la matrice pour obtenir g^{ij} avec $g^{ij}g_{jk} = \delta_k^i$ cela définit un produit scalaire sur l'espace cotangente qu'on pourra noter g^{-1} et même parfois g s'il n'y a pas d'ambiguïté :

$$g = g^{ij} \frac{\partial}{\partial x^i} \otimes \frac{\partial}{\partial x^j} \quad (4.7)$$

Remarque :

La métrique g est quelques fois notées $ds^2 = g_{ij} dx^i dx^j$ est appelée élément de longueur ou tenseur métrique.

En général, une forme bilinéaire symétrique est caractérisée par signature (p, q) le nombre des signes $+(p)$ et des signes $-(q)$, obtenu lorsqu'on la diagonalise. Si on tient à préciser que la signature est du type $(p, 0)$ ou $(0, q)$ on dira que la variété est proprement riemannienne. Si on tient à préciser que la signature est du type $(p, 1)$ ou $(1, q)$ on dira que la variété est Lorentzien.

• **Théorème 4**

Sur toute variété différentiable M , il existe une métrique riemannienne.

Preuve :

Soit $\{V_\alpha\}$, $\alpha \in I$ un recouvrement de M par des voisinages coordonnés. Alors $\{V_\alpha\}, \alpha \in I$ est un recouvrement localement fini.

Soit $\{f_\alpha\}$ une partition différentiable de l'unité subordonnée à $\{V_\alpha\}$. Pour $X, Y \in T_p M$ et $\alpha \in I$ avec $p \in V_\alpha$.

Posons :
$$g(X, Y) = \sum_{\alpha} f_{\alpha}(p) g^{\alpha}(X, Y) \quad \forall p \in M \quad X, Y \in T_p M$$

g ainsi défini est une métrique riemannienne sur M .

Où g^{α} est le produit scalaire défini sur V_{α} .

1.3.Champs de vecteurs et formes extérieures

- **Définition 5**

Soit M une variété riemannienne. Un champ de vecteurs sur M est une correspondance qui associe à chaque point $p \in M$ un vecteur tangent $X_p \in T_p M$ où $T_p M$ est l'espace tangent à M en p . En termes d'application, un champ de vecteurs est une application X de la variété riemannienne M dans le fibre tangent.

$$X : M \rightarrow TM \tag{4.8}$$

$$p \rightarrow (p, X_p) \quad \text{tel que } \circ X = \text{id}_M$$

$$\text{Où } TM = \bigcup_{p \in M} T_p M = \{(p, X_p) : p \in M, X_p \in T_p M\} \tag{4.9}$$

Et $\pi : TM \rightarrow M$ la projection canonique sur M .

L'ensemble des champs de vecteurs sur M sera noté par $\mathcal{X}(M)$.

Tout champ de vecteurs $X \in \mathcal{X}(M)$ détermine une application [93].

$$X : C^\infty(M) \rightarrow C^\infty(M)$$

$$f \rightarrow X(f)$$

$$\text{Telle que } X(f) : M \rightarrow \mathbb{R}$$

$$p \rightarrow X(f)(p) \stackrel{\Delta}{=} X_p(f) = \sum_{i=1}^n v^i \frac{\partial f}{\partial x^i}(p) \tag{4.10}$$

1.3.1. Longueur sur une variété riemannienne

Soit $\gamma : I \rightarrow (M, g)$ une courbe sur une variété riemannienne (M, g) . Un champ de vecteurs X le long de γ est une application qui associe à toute $t \in I$ un vecteur tangent $X(t) \in T_{\gamma(t)} M$.

L'image d'un champ de vecteurs $\frac{d}{dt}$ sur I par la différentielle $d\gamma$ de γ est le champ de vecteurs $d\gamma\left(\frac{d}{dt}\right)$ noté $\frac{d\gamma}{dt}$ et qu'on appelle le champ de vecteurs de γ .

Soit $[a, b] \subset I$ on peut restreindre γ sur $[a, b]$ et appelé un segment noté $\gamma_{[a, b]}$.

La longueur d'un segment est définie par :

$$L(\gamma) = \int_a^b \left\| \frac{d\gamma}{dt} \right\| = \int_a^b \left[g\left(\frac{d\gamma}{dt}, \frac{d\gamma}{dt}\right) \right]^{\frac{1}{2}} dt \quad (4.11)$$

1.3.2. Connexions et Courbures

Dans \mathbb{R}^n pour dériver le champ de vecteur X au point p dans la direction du vecteur v , nous calculons la limite :

$$\lim_{t \rightarrow 0} \frac{X(p + tv) - X(p)}{t} \quad (4.12)$$

Cependant dans cette limite, l'opération compliquée est la soustraction, car les champs des vecteurs $X(p+tv)$ et $X(p)$ que nous comparons ne sont pas dans le même espace vectoriel. En effet, $X(p+tv)$ est dans l'espace tangent $T_{p+tv} \mathbb{R}^n$ et $X(p)$ dans l'espace tangent $T_p \mathbb{R}^n$. Effectuer une telle opération dans \mathbb{R}^n est possible, car pour chaque point $p \in \mathbb{R}^n$, $T_p \mathbb{R}^n$ peut être canoniquement identifié avec $T_0 \mathbb{R}^n \approx \mathbb{R}^n$

Par contre, pour une variété différentielle M , il n'est pas possible de faire cette identification entre espaces tangents en des points différents de la variété.

Considérons une courbe $\gamma : I \subseteq \mathbb{R} \rightarrow M$ et $X : \gamma \rightarrow TM$ un champ de vecteur le long de γ tangent à M . En général, le vecteur $\frac{dX}{dt}(t)$ n'est pas tangent à M , et n'appartient donc pas à $T\gamma(t)M$. Donc, la fibre tangente n'est pas stable pour la différentiation ordinaire d'un champ de vecteurs dépendant d'un paramètre. Puisque la fibre tangente n'est pas un espace vectoriel et la différence de deux champs de vecteurs en des points différents de M n'est pas dans la fibre tangente, à plus forte raison leur dérivée.

Pour contourner cette difficulté, nous considérons sur l'espace tangent $T\gamma(t)M$ la projection orthogonale de $\frac{dX}{dt}$ que nous appelons dérivée covariante de X notée $\frac{DX}{dt}$.

Une connexion est donc une façon de prendre la dérivée d'un champ de vecteurs dans la direction du vecteur tangent à la variété en ce point conformément à la dérivation ordinaire.

• **Définition 6**

Une connexion ∇ sur M , plus généralement sur la fibre tangente est une application :

$$\nabla : \mathcal{X}(M) \times \mathcal{X}(M) \rightarrow \mathcal{X}(M)$$

$$(X, Y) \rightarrow \nabla_X Y$$

On lit dérivée de Y suivant X qui satisfait les conditions suivantes :

- 1) $\nabla_X(Y + Z) = \nabla_X Y + \nabla_X Z$
- 2) $\nabla_{fX+gY} Z = f \nabla_X Z + g \nabla_Y Z$
- 3) $\nabla_X(fY) = f \nabla_X Y + X(f)Y$ [94]

Il faut noter que, pour une fonction différentiable f , nous avons

$$\nabla_X f = d f = X(f)$$

Si nous choisissons d'écrire X et Y en fonction de leurs composantes dans un système de coordonnées locales (x^1, \dots, x^n) :

$$X = X^i \frac{\partial}{\partial x^i} \quad \text{et} \quad Y = Y^j \frac{\partial}{\partial x^j} \quad (4.13)$$

Alors, nous avons :

$$\nabla_X Y = \nabla_{X^i \frac{\partial}{\partial x^i}} Y^j \frac{\partial}{\partial x^j} = (\nabla_{X^i \frac{\partial}{\partial x^i}} Y^j) \frac{\partial}{\partial x^j} + Y^j (\nabla_{X^i \frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^j}) \quad (4.14)$$

Ou encore :

$$\nabla_X Y = X^i \frac{\partial Y^j}{\partial x^i} \frac{\partial}{\partial x^j} + Y^j X^i \nabla_{X^i \frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^j} \quad (4.15)$$

La relation (2) nous donne l'expression locale de $\nabla_X Y$

Posons :

$$\nabla_{X^i \frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^j} = \Gamma_{ij}^K X^K \quad (4.16)$$

Alors, la relation (4.14) devient :

$$\nabla_X Y = X^i \frac{\partial Y^j}{\partial x^i} \frac{\partial}{\partial x^j} + Y^j X^i \Gamma_{ij}^K X^K \quad (3.17)$$

Les symboles, Γ_{ij}^K sont dits symboles de Christoffel.

1.4. Torsion

- **Définition 7**

Étant donné une connexion ∇ , on appelle torsion de ∇ , l'application T défini par :

$$T : \mathcal{X}(M) \times \mathcal{X}(M) \rightarrow \mathcal{X}(M)$$

$$(X, Y) \rightarrow T(X, Y) = \nabla_X Y - \nabla_Y X - [X, Y]$$

Remarque :

Étant donnés $X = X^i \frac{\partial}{\partial x^i}$ et $Y = Y^j \frac{\partial}{\partial x^j}$ dans un système de coordonnées locales (x_1, \dots, x_n) , alors nous avons :

$$\nabla_X Y = X^i \frac{\partial y^j}{\partial x^i} \frac{\partial}{\partial x^j} + Y^j X^i \nabla_{\frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^j} \quad (4.18)$$

$$\nabla_Y X = Y^j \frac{\partial x^i}{\partial x^j} \frac{\partial}{\partial x^i} + X^i Y^j \nabla_{\frac{\partial}{\partial x^j}} \frac{\partial}{\partial x^i} \quad (4.19)$$

$$[X, Y] = XY - YX = X^i \frac{\partial y^j}{\partial x^i} \frac{\partial}{\partial x^j} - Y^j \frac{\partial x^i}{\partial x^j} \frac{\partial}{\partial x^i} \quad (4.20)$$

D'où l'expression locale de T est donnée par :

$$T = Y^j X^i \nabla_{\frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^j} - X^i Y^j \nabla_{\frac{\partial}{\partial x^j}} \frac{\partial}{\partial x^i} \quad (4.21)$$

- **Définition 3.8**

On appelle connexion symétrique ou encore connexion sans torsion une connexion qui est telle que $T(x, y) = 0$.

Remarque : Pour une connexion sans torsion, nous avons :

$$\nabla_{\frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^j} = \nabla_{\frac{\partial}{\partial x^j}} \frac{\partial}{\partial x^i} \quad (4.22)$$

Ou encore :
$$\Gamma_{ij}^K = \Gamma_{ji}^K \quad (4.23)$$

1.5.Caractéristiques des Variétés riemanniennes

Théorème 8

Sur une variété riemannienne (M, g), il existe une et une seule connexion symétrique (sans torsion) compatible avec métrique g. Elle est donnée par la formule :

$$g(\nabla_X Y, Z) = \frac{1}{2} \{ Xg(Y, Z) - Zg(X, Y) + Yg(Z, X) - g(X, [Y, Z]) + g(Z, [X, Y]) + g(Y, [Z, X]) \} \quad (4.24)$$

1.5.1. Connexion de Levi – Civita

La connexion définie dans le théorème (4.8) est appelée connexion, de Levi – Civita [95].

Dans la suite, sauf mention expresse du contraire, ∇ désignera toujours la connexion de Levi – Civita de (M, g).

Écrivons la connexion de Levi – Civita en coordonnées locales :

$$\nabla_{X_i} X_j = \Gamma_{ij}^K X_K \quad (4.25)$$

Où
$$\frac{\partial}{\partial x^i} = X_i \quad (4.26)$$

Et posons $X = X_i, Y = X_j$ et $Z = X_k,$

Alors, nous avons $[X_i, X_j] = [X_j, X_k] = [X_i, X_k] = 0$ et la relation (4.24) devient :

$$g(X_k, \nabla_{X_i} X_j) = \frac{1}{2} \{ X_i g(X_j, X_k) - X_k g(X_i, X_j) + X_j g(X_k, X_i) \} \quad (4.27)$$

Ou encore

$$g(X_k, \Gamma_{ij}^l X_l) = \frac{1}{2} \left\{ \frac{\partial g_{jK}}{\partial x^i} + \frac{\partial g_{Ki}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^k} \right\} \quad (4.28)$$

$$\Gamma_{ij}^l g_{kl} = \frac{1}{2} \left\{ \frac{\partial g_{jK}}{\partial x^i} + \frac{\partial g_{Ki}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^K} \right\} \quad (4.29)$$

En multipliant les deux membres par g^{km} , nous avons

$$\Gamma_{ij}^l g_{kl} g^{Km} = \frac{1}{2} \left\{ \frac{\partial g_{jK}}{\partial x^i} + \frac{\partial g_{Ki}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^K} \right\} g^{Km} \quad (4.30)$$

$$\Gamma_{ij}^m = \frac{1}{2} \left\{ \frac{\partial g_{jK}}{\partial x^i} + \frac{\partial g_{Ki}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^K} \right\} g^{Km} \quad (4.33)$$

- **Définition 9**

Les symboles Γ_{ijK} et Γ_{ij}^K sont données par

$$\Gamma_{ijK} = \frac{1}{2} \left\{ \frac{\partial g_{jK}}{\partial x^i} + \frac{\partial g_{Ki}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^K} \right\} \quad (4.34)$$

$$\Gamma_{ij}^K = g^{Km} \Gamma_{ijK} = \frac{1}{2} g^{Km} \left\{ \frac{\partial g_{jK}}{\partial x^i} + \frac{\partial g_{Ki}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^K} \right\} \quad (4.35)$$

Ils sont appelés respectivement symboles de Christoffel de premier et de second espèces [96].

- **Proposition 10**

$$(i) \quad \Gamma_{ijK} = \Gamma_{jiK} \quad (4.36)$$

$$(ii) \quad \Gamma_{ij}^m = \Gamma_{ji}^m \quad (4.37)$$

$$(iii) \quad \Gamma_{ij}^m = g^{Km} \Gamma_{ijK} \quad (4.38)$$

$$(iv) \quad \frac{\partial g_{pq}}{\partial x^m} = \Gamma_{pmq} + \Gamma_{qmp} \quad (3.39)$$

$$(v) \quad \frac{\partial g_{pq}}{\partial x^m} = -g^{pn} \Gamma_{mn}^q - g^{qn} \Gamma_{mn}^p \quad (4.40)$$

$$(vi) \quad \Gamma_{pq}^p = \frac{\partial}{\partial x^q} \ln \sqrt{g} \quad (4.41)$$

$$(vii) \quad \Gamma_{pq,r} = g_{rs} \Gamma_{pq}^s \quad (4.42)$$

Évaluation des symboles de Christoffel (a) de première espèce (b) de seconde espèce, pour les espaces où $g_{pq} = 0$ si $p \neq q$.

$$(a) \quad \text{Si } p = q = r \quad \Gamma_{pqr} = \Gamma_{ppp} \frac{1}{2} \left(\frac{\partial g_{pp}}{\partial x^p} + \frac{\partial g_{pp}}{\partial x^p} - \frac{\partial g_{pp}}{\partial x^p} \right) = \frac{1}{2} \frac{\partial g_{pp}}{\partial x^p} \quad (4.43)$$

$$\text{Si } p = q \neq r, \quad \Gamma_{pqr} = \Gamma_{ppr} \frac{1}{2} \left(\frac{\partial g_{pr}}{\partial x^p} + \frac{\partial g_{pr}}{\partial x^p} - \frac{\partial g_{pp}}{\partial x^r} \right) = \frac{1}{2} \frac{\partial g_{pp}}{\partial x^r} \quad (4.44)$$

$$\text{Si } p = r \neq q \quad \Gamma_{pqr} = \Gamma_{rqr} \frac{1}{2} \left(\frac{\partial g_{pp}}{\partial x^q} + \frac{\partial g_{pq}}{\partial x^p} - \frac{\partial g_{pq}}{\partial x^p} \right) = \frac{1}{2} \frac{\partial g_{pp}}{\partial x^q} \quad (4.45)$$

Si p, q, r sont distincts $\Gamma_{pqr} = 0$

$$(b) \quad \Gamma_{pq}^s = g^{rs} \Gamma_{pqr} = 0 \quad \text{si } r \neq s \quad (4.46)$$

$$= g^{ss} \Gamma_{pqs} \frac{\Gamma_{pqs}}{g_{ss}} \quad \text{si } r = s \quad (4.47)$$

D'après (a)

$$\text{Si } p = q = s \quad \Gamma_{pq}^s = \Gamma_{pp}^p = \frac{\Gamma_{ppp}}{g_{pp}} = \frac{1}{2g_{pp}} \frac{\partial g_{pp}}{\partial x^p} = \frac{1}{2} \frac{\partial}{\partial x^p} (\ln g_{pp}) \quad (4.48)$$

$$\text{Si } p = q \neq s \quad \Gamma_{pq}^s = \Gamma_{pp}^s = \frac{\Gamma_{pps}}{g_{ss}} = -\frac{1}{2g_{ss}} \frac{\partial g_{pp}}{\partial x^s} \quad (4.49)$$

$$\text{Si } p = s \neq q \quad \Gamma_{pq}^s = \Gamma_{pq}^p = \frac{\Gamma_{pqp}}{g_{pp}} = \frac{1}{2g_{pp}} \frac{\partial g_{pp}}{\partial x^q} = \frac{1}{2} \frac{\partial}{\partial x^q} (\ln g_{pp}) \quad (4.50)$$

Si p, q, s sont distincts, $\Gamma_{pq}^s = 0$

1.5.2. Tenseur de Riemann.

- **Définition 11**

Soit (M, g, ∇) une variété de riemannienne munie de la connexion de Levi – Civita. La courbure de Riemann est l'application qui associe à chaque paire X, Y de champs de vecteurs un opérateur R

$$R(X, Y) : \mathcal{X}(M) \rightarrow \mathcal{X}(M)$$

$$Z \rightarrow R(X, Y) Z \quad (4.51)$$

Elle est donné par

$$R(X, Y) Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X, Y]} Z \quad (4.52)$$

Remarque : Nous pouvons, en utilisant la métrique g , associer à tout quadruplet de vecteurs X, Y, Z, W la fonction $g(R(X, X) Z, W)$ définie par :

$$g(R(X, Y) Z, W) = R(X, Y, Z, W) \quad (4.53)$$

Appelé **tenseur de courbure**.

$$\text{En posant } X = X_i ; Y = X_j ; Z = X_k \quad \text{où } X_i = \frac{\partial}{\partial x^i}, \quad (4.54)$$

On a :

$$R(X_i, X_j) X_k = \nabla_i \nabla_j X_k - \nabla_j \nabla_i X_k \quad \text{où } \nabla_i = \nabla \frac{\partial}{\partial x^i} \quad (4.55)$$

$$\begin{aligned} R_{ijk}^l X_l &= \nabla_i (\Gamma_{jK}^l X_l) - \nabla_j (\Gamma_{iK}^l X_l) \\ &= (X_i \Gamma_{jK}^l) X_l + \Gamma_{jK}^l \nabla_i X_l - (X_j \Gamma_{iK}^l) X_l + (X_i \Gamma_{jK}^l) X_l \\ &= \Gamma_{jK}^l \Gamma_{il}^m X_m - \Gamma_{iK}^l \nabla_j X_l - (X_j \Gamma_{iK}^l) X_l + (X_i \Gamma_{jK}^l) X_l \end{aligned} \quad (4.56)$$

D'où

$$R_{ijk}^m = \Gamma_{jK}^l \Gamma_{il}^m - \Gamma_{iK}^l \Gamma_{jl}^m + \frac{\partial \Gamma_{jK}^m}{\partial x^i} - \frac{\partial \Gamma_{iK}^m}{\partial x^j} \quad (4.57)$$

$$\text{Et } R_{ijkm} = g_{lm} R_{ijk}^l \quad (4.58)$$

$$\text{Et } R_{ijkm} = g_{lm} R_{ijk}^l = g_{lm} (\Gamma_{jK}^l \Gamma_{il}^m - \Gamma_{iK}^l \Gamma_{jl}^m + \frac{\partial \Gamma_{jK}^m}{\partial x^i} - \frac{\partial \Gamma_{iK}^m}{\partial x^j}) \quad (4.59)$$

De cette expression, on déduit les propriétés de symétrie suivantes :

• **Proposition 12** [98]

- $R_{ijkl} = -R_{jikl} = -R_{ijlk}$
- $R_{ijkl} = -R_{klij}$

Ce qui signifie que le tenseur est symétrique sur chaque paire d'indices ij et km et il est symétrique par rapport à la transposition de ces deux paires.

On vérifie aisément que la somme cyclique des composantes de R_{ijkl} formée sur trois indices quelconque est égale à zéro.

$$R(X, Y) Z + R(Y, Z) X + R(Z, X) Y = 0$$

- $R_{ijkl} + R_{jikl} + R_{ijlk} = 0$

En effet,

$$R(X, Y) Z + R(Y, Z) X + R(Z, X) Y = \nabla_x \nabla_y Z + \nabla_y \nabla_z X - \nabla_z \nabla_y Z + \nabla_z \nabla_x Y - \nabla_x \nabla_z Y = 0$$

Puisque $\nabla_Y Z = \nabla_Z Y$; car ∇ est symétrique.

Démontrons l'identité de Bianchi :

$$\frac{\partial R_{ikl}^n}{\partial x^m} + \frac{\partial R_{imk}^n}{\partial x^l} + \frac{\partial R_{ilm}^n}{\partial x^k} = 0 \quad (4.60)$$

On vérifie cette identité en utilisant le système de coordonnées géodésiques en vertu de son caractère tensoriel, la relation ci-dessus sera vraie dans n'importe quel autre système. En dérivant R_{ijk}^l et en y posant $\Gamma_{ik}^l = 0$, on trouve qu'au point considéré :

$$\frac{\partial R_{ijk}^l}{\partial x^m} = \frac{\partial^2 \Gamma_{jk}^m}{\partial x^m \partial x^l} - \frac{\partial^2 \Gamma_{ik}^m}{\partial x^m \partial x^j} = 0 \quad (4.61)$$

Cette expression permet de s'assurer que l'identité de Bianchi est vraie [91].

• **Définition 13**

Étant donné un point p d'une variété riemannienne (M, g) . Soit σ un plan de l'espace tangent $T_p M$. On appelle courbure sectionnelle de σ sur M au point p , le nombre réel défini par :

$$K(X, Y) = \frac{R(X, Y, X, Y)}{|X|^2 |Y|^2 - (g(X, Y))^2} \quad (4.62)$$

En coordonnées locales

$$X = \varepsilon^i \frac{\partial}{\partial x^i} \quad \text{et} \quad Y = \eta^j \frac{\partial}{\partial x^j} \quad \text{on a :}$$

$$R(X, Y, X, Y) = g(R(X, Y) X, Y) = g\left(R\left(\varepsilon^i \frac{\partial}{\partial x^i}, \eta^j \frac{\partial}{\partial x^j}\right) \varepsilon^K \frac{\partial}{\partial x^K}, \eta^l \frac{\partial}{\partial x^l}\right) = R_{ijkl} \varepsilon^i \eta^j \varepsilon^K \eta^l \quad (4.63)$$

$$|X|^2 = g(X, X) = g\left(\varepsilon^i \frac{\partial}{\partial x^i}, \varepsilon^K \frac{\partial}{\partial x^K}\right) = \varepsilon^i \varepsilon^K g_{iK} \quad (4.64)$$

$$|Y|^2 = g(Y, Y) = g\left(\eta^j \frac{\partial}{\partial x^j}, \eta^l \frac{\partial}{\partial x^l}\right) = \eta^j \eta^l g_{jl} \quad (4.65)$$

$$\begin{aligned} (g(X, Y))^2 &= g(X, Y)g(X, Y) = g\left(\varepsilon^i \frac{\partial}{\partial x^i}, \eta^j \frac{\partial}{\partial x^j}\right) g\left(\varepsilon^K \frac{\partial}{\partial x^K}, \eta^l \frac{\partial}{\partial x^l}\right) \\ &= \varepsilon^i \varepsilon^K \eta^j \eta^l g_{ij} g_{kl} \end{aligned} \quad (4.66)$$

$$\text{Et } K(X, Y) = \frac{R_{ijkl} \varepsilon^i \eta^j \varepsilon^K \eta^l}{(g_{iK} \cdot g_{Kl} - g_{ij} g_{Kl}) \varepsilon^i \eta^j \varepsilon^K \eta^l} \quad (4.67)$$

La courbure sectionnelle détermine donc complètement le tenseur de courbure de Riemann.

1.5.3. Courbure de Ricci.

- **Définition 14**

Le tenseur de courbure de Ricci, noté Ric, d'une variété riemannienne est le tenseur d'ordre 2.

$$\text{Ric}(X, Y) = \text{Tr}(A \rightarrow R(X, A) Y) \quad (4.68)$$

Où Tr désigne la trace de l'application linéaire $A \rightarrow R(X, A) Y$

Noter que le tenseur de courbure de Ricci est symétrique.

En effet soit l'application

$$A : T_p M \rightarrow T_p M \quad \text{auto - adjointe}$$

Et $B : T_p M \times T_p M \rightarrow \mathbb{R}$ l'application bilinéaire définie par $B(X, Y) = g(A(X), Y)$

Si on munit T_pM d'une base orthonormée (X_i) , alors la trace de A est donnée par :

$$T_r A = \sum_i B(X_i, X_i) = g(A(X_i, X_i)) \quad (4.69)$$

En coordonnées

$$R(X, Y) = \sum_i g(R(X, X_i)Y, X_i) = \sum_i R(X, X_i, Y, X_i) \quad (4.70)$$

Il résulte donc des symétries de R que $Ric(X, Y) = Ric(Y, X)$.

1.5.4. Courbure scalaire

- **Définition 3.15**

La courbure scalaire ou le tenseur de courbure scalaire au point $p \in M$ est la trace de l'opérateur de Ricci définie par :

$$Ric(X, Y) = g(Ric(X), Y) \quad (4.71)$$

Si nous la désignons par $scal$, alors :

$$scal = \sum_i Ric(X_i, X_i) = \sum_{i,j} R(X_j, X_i, X_j, X_i) = \sum_{i \neq j} K(X_j, X_i) \quad (4.72)$$

Où K désigne la courbure sectionnelle du plan engendré par $\{X_j, X_i\}$.

On peut obtenir la dernière égalité en observant que

$$R(X, Y, X, Y) = K(X, Y)(|X|^2|Y|^2 - (g(X, Y))^2)$$

$$R(X_j, X_i, X_j, X_i) = K(X_j, X_i)(|X|^2|Y|^2 - (g(X, Y))^2)$$

$$R(X_j, X_i, X_j, X_i) = \begin{cases} K(X_j, X_i) & si \ i \neq j \\ 0 & si \ i = j \end{cases} \quad (4.73)$$

2. Descripteurs métriques

2.1. Prétraitement

La métrique riemannienne est calculée à partir des dérivées partielles en chaque point. Pour cette raison, simuler une image contenant des lettres manuscrites (souvent représentées en noir et blanc) par une variété et calculer sa métrique ne donne pas des résultats exploitables. (La dérivée partielle en chaque point est nulle et par conséquent la métrique est nulle en ce point).

Pour remédier à cette situation, nous avons enrichi le contenu de l'image en la superposant séparément avec deux matrices maillées (figure 4.2). Le maillage est une technique souvent utilisée dans le traitement des objets 3D, soit pour améliorer leur affichage ou pour relier leur squelette (figure 3.4). Pour la mise en œuvre de cette méthode, on utilise le programme ci-après:

Programme N°1 (Matlab)

% nous allons créer deux matrices x et y vide qui recevrons d'une manière uniforme et croissante les valeurs entre 0 et 127

Fonction [x, y, z, ds, dt] = mail()

smin = 127;

smax = 0;

ns = 64;

ds = (smax-smin)/ns;

% t d'une manière ascendante. (Du haut vers le bas)

tmin = 127.0;

tmax = 0.0;

nt = 64;

dt = (tmax-tmin)/nt;

% pour le maillage on peut utiliser la fonction meshgrid() de Matlab

[s,t] = meshgrid(smin:ds:smax, tmin:dt:tmax);

% lecture d'image contenant le caractère

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Figure 4.2 : Exemple de l'effet de la fonction Meshgrid (0 :1 :7): Matrice (s)

% on divise les valeurs de ses pixels par 2

```
i=double (imread('z8.bmp'))/2;
```

%% on applique le changement du paramètre pour obtenir les coordonnées 3D

```
x = s+i;
```

```
y = t+i;
```

```
z = s-s;
```

```
surf(x,y,z); % projection des résultats.
```

La figure 4.3 représente un exemple des résultats projetés du premier algorithme :

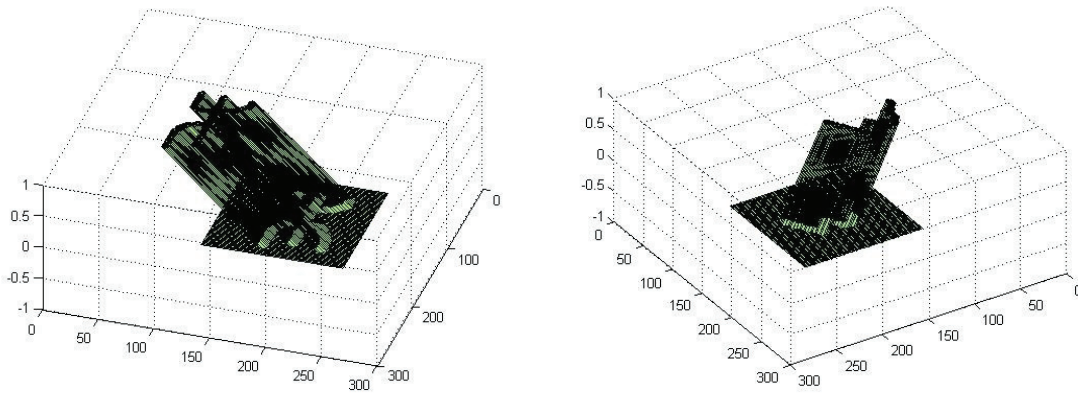


Figure 4.3 : Exemple de la projection des résultats obtenus par le programme 1, de deux caractères Tifinagh.

- *Calcul de la Métrique*

Par la suite chaque caractère est considéré comme une surface particulière pour lequel on calcule la métrique en chaque point. La métrique est représentée par un tenseur (une matrice définie positive) souvent appelé G donné par [94]:

$$G_{ij} = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \quad (4.74)$$

Pour les vecteurs tangents t et s les coefficients métriques sont calculés comme suit :

$$\begin{aligned} g_{11} &= \langle DD_s, DD_s \rangle ; g_{12} = \langle DD_s, DD_t \rangle ; \\ g_{21} &= \langle DD_t, DD_s \rangle ; g_{22} = \langle DD_t, DD_t \rangle ; \end{aligned} \quad (4.75)$$

$$\text{Avec : } DD_s = \left[\frac{\partial x}{\partial s}, \frac{\partial y}{\partial s}, \frac{\partial z}{\partial s} \right] \text{ et } DD_t = \left[\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}, \frac{\partial z}{\partial t} \right] \quad (4.76)$$

2.2. Calcul des descripteurs métriques

- Le premier descripteur métrique est la somme de la déterminante de G en chaque point de la surface (chaque pixel de l'image) :

$$D1 = \sum_{i=0,n} \sum_{j=0,m} G_{ij} \quad (4.77)$$

- Avec n et m : le nombre de lignes et colonnes,

On note :

- CG_j la somme de la déterminante de G en chaque pixel du j^{ème} colonne :

$$CG_j = \sum_{i=0,n} G_{ij}$$

- Et LG_i la somme de la déterminante de G en chaque pixel du i^{ème} ligne :

$$LG_i = \sum_{j=0,m} G_{ij}$$

- On note successivement le deuxième et le troisième descripteur métrique la somme des CG_j allant de 0 à n/2 et de n/2+1 à n :

$$D2 = \sum_{j=0, \frac{n}{2}} CG_j \quad (4.78)$$

$$D3 = \sum_{j=\frac{n}{2}+1, n} CG_j \quad (4.79)$$

- Les quatrième et cinquième descripteurs métriques sont respectivement la somme des LG_i allant de 0 à m/2 et de m/2+1 à m :

$$D4 = \sum_{i=0, \frac{m}{2}} LG_i \quad (4.80)$$

$$D5 = \sum_{i=\frac{m}{2}+1, m} LG_i \quad (4.81)$$

Le Tableau 4.1 illustre des résultats obtenus par les cinq descripteurs décrits dans cette section en utilisant l'algorithme suivant :

Algorithme descripteurs métriques

```

fonction [d1,d2,d3,d4,d5]=des_Metric()
[x,y,z,ds,dt] = algorithme_pre_traitement();
[ns,nt]= la taille de(x);
% Calule du gradient en chaque point
[DxDs,DxDt] = gradient(x,ds,dt);
[DyDs,DyDt] = gradient(y,ds,dt);
[DzDs,DzDt] = gradient(z,ds,dt);

DDs1 = [0 0 0]; DDt1 = [0 0 0];
% Allocation des espaces mémoires pour les vecteurs de taille (ns x
nt).
DDs = zeros(ns, nt, 3); % zeros permet de crée une
DDt = zeros(ns, nt, 3); % matrice de zéros

Nhat = zeros(ns, nt, 3);

% Allocation d'espaces pour les tenseurs métriques.
g = zeros(ns,nt,2,2);
invg = zeros(ns,nt,2,2);
detg=zeros(ns,nt);

for ss = 1:ns
    for tt = 1:nt
        DDs1(1) = DxDs(ss,tt); DDt1(1) = DxDt(ss,tt);
        DDs1(2) = DyDs(ss,tt); DDt1(2) = DyDt(ss,tt);
        DDs1(3) = DzDs(ss,tt); DDt1(3) = DzDt(ss,tt);
        % les coefficient métriques en chaque point(a, b, c, d)
        a = DDs1 * DDs1'; b = DDs1 * DDt1';
        c = DDt1 * DDs1'; d = DDt1 * DDt1';

        g(ss,tt,1,1) = a; g(ss,tt,1,2) = b;
        g(ss,tt,2,1) = c; g(ss,tt,2,2) = d;

        % Calcule de la déterminante des coefficients métriques en chaque
point
        gdet = a*d - b*c;
        detg(ss,tt)=gdet;
    end
end
% Calcul des descripteurs Métriques
d1=detg;

for ss=1:65
    for tt=1:32
        d2=detg(ss,tt)
    end
end
for ss=1:65
    for t=33:65
        d3=detg(ss,tt)
    end
end
for ss=1:32

```

```

for tt=1:65
    d4=detg(ss,tt)
end
end
for ss=33:65
    for tt=1:65
        d5=detg(ss,tt)
    end
end
end

```

| | D1 | D2 | D3 | D4 | D5 |
|---|--------|--------|--------|--------|--------|
| ⴰ | 7.1017 | 4.5116 | 2.5901 | 3.5621 | 3.5396 |
| ⴱ | 3.9435 | 2.5255 | 1.4181 | 2.4242 | 1.5193 |
| ⴲ | 3.3036 | 2.0339 | 1.2697 | 1.8063 | 1.4974 |
| ⴳ | 4.6660 | 2.7093 | 1.9566 | 3.3298 | 1.3362 |
| ⴴ | 5.1201 | 3.4737 | 1.6464 | 2.4777 | 2.6423 |

Tableau 4.1 : les descripteurs métriques pour quelques caractères Tifinagh

On remarque que le premier descripteur permet de distinguer les caractères qui ont une morphologie distincte, tandis que le reste des descripteurs, après une normalisation par rapport à leur somme, permettent :

- de distinguer entre les caractères géométriquement proches (Tableau 4.2 : exemple des caractères Yarr et Yass) ;
- une résistance au changement d'échelle (Tableau 4.3).



| | D1 | D2 | D3 | D4 | D5 |
|---|-------|--------|--------|--------|--------|
|  | 0.333 | 0.1538 | 0.1796 | 0.2070 | 0.1264 |
|  | 0.333 | 0.1937 | 0.1396 | 0.1879 | 0.1455 |

Tableau 4.2 : Descripteurs métriques pour les deux caractères Yass et Yarr.




| | D2 | D3 | D4 | D5 |
|---|--------|--------|--------|--------|
|  | 0.1443 | 0.1890 | 0.2250 | 0.1083 |
|  | 0.1356 | 0.1978 | 0.2062 | 0.1271 |
|  | 0.1402 | 0.1931 | 0.2175 | 0.1158 |

Tableau 4.3 : Descripteurs métriques du caractère Yass calculés pour différentes tailles.

3. Les descripteurs géodésiques

3.1. Courbes Géodésiques

Les géodésiques sont une généralisation de ce qu'une droite ou un segment de droite sont dans le plan. Une géodésique est le chemin le plus court reliant deux points suffisamment proches l'un de l'autre. Les géodésiques dans les variétés riemanniennes peuvent être assimilées à des segments de droites pour les variétés affines. Dans le plan, les géodésiques sont des droites, tandis que sur la sphère, les géodésiques prennent la forme de cercles [96].

Définition 3.16 : Courbes géodésiques

Soit une variété différentiable (M, ∇) munie d'une connexion.

Une courbe paramétrée $\gamma : I \rightarrow M$ est appelée une *géodésique* en un point $t_0 \in I$ si

$$\nabla \frac{d\gamma}{dt} = 0 \quad \text{pour } t = t_0 \quad (4.82)$$

$$\text{Où} \quad \nabla_{\dot{\gamma}} \dot{\gamma} = 0 \quad (4.83)$$

La courbe γ est une géodésique sur I si elle est une géodésique pour tout point $t \in I$. En coordonnées locales (x_1, \dots, x_n) , nous avons :

$$\begin{aligned} \gamma(t) &= (x_1(t), \dots, x_n(t)) \\ \dot{\gamma} &= (\dot{x}_1, \dots, \dot{x}_n) = \sum_i \dot{x}_i \frac{\partial}{\partial x_i} \end{aligned} \quad (4.84)$$

Il vient alors :

$$\begin{aligned} \nabla_{\dot{\gamma}} \dot{\gamma} &= \nabla_{\dot{\gamma}} \left(\sum_i \dot{x}_i \frac{\partial}{\partial x_i} \right) \\ &= \sum_i \nabla_{\dot{\gamma}} (\dot{x}_i) \frac{\partial}{\partial x_i} + \sum_i \dot{x}_i \nabla_{\dot{\gamma}} \left(\frac{\partial}{\partial x_i} \right) \\ &= \sum_i \left(\ddot{\gamma}(x_i) \frac{\partial}{\partial x_i} + \dot{x}_i \nabla_{\dot{\gamma}} \left(\frac{\partial}{\partial x_i} \right) \right) \\ &= \sum_i \left(\frac{d\dot{\gamma}}{dt}(x_i) \frac{\partial}{\partial x_i} + \dot{x}_i \nabla_{\dot{x}_j} \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_i} \right) \\ &= \sum_i \left(\frac{d}{dt}(\dot{x}_i) \frac{\partial}{\partial x_i} + \dot{x}_i \sum_j \dot{x}_j \nabla_{\dot{x}_j} \frac{\partial}{\partial x_j} \frac{\partial}{\partial x_i} \right) \\ &= \sum_i \left(\ddot{x}_i \frac{\partial}{\partial x_i} + \sum_j \dot{x}_i \dot{x}_j \left(\sum_k \Gamma_{ij}^k \nabla_{\dot{x}_k} \frac{\partial}{\partial x_k} \right) \right) \\ &= \sum_k \left(\ddot{x}_k + \sum_{i,j} \Gamma_{ij}^k \dot{x}_i \dot{x}_j \right) \frac{\partial}{\partial x_k} \end{aligned} \quad (4.85)$$

La courbe γ est une géodésique si $\nabla_{\dot{\gamma}} \dot{\gamma} = 0$ c'est-à-dire :

$$\frac{d^2 x_k}{dt^2} + \sum_{i,j} \Gamma_{ij}^k \frac{dx_i}{dt} \frac{dx_j}{dt} = 0 \quad k=1, \dots, n \quad (4.86)$$

Il s'agit des équations différentielles de second ordre d'une géodésique $\gamma(t) = ((x_1(t), \dots, x_n(t)))$ d'une connexion dont les coefficients sont les symboles de Christoffel Γ_{ij}^k [92].

Introduisons maintenant une métrique riemannienne g sur (M, ∇) qui soit compatible avec ∇

$$Xg(Y, Z) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z)$$

Si $\gamma : I \rightarrow M$ est une géodésique alors :

$$\nabla_{\dot{\gamma}} g(\dot{\gamma}, \dot{\gamma}) = g(\nabla_{\dot{\gamma}} \dot{\gamma}, \dot{\gamma}) + g(\dot{\gamma}, \nabla_{\dot{\gamma}} \dot{\gamma}) = 2g(\nabla_{\dot{\gamma}} \dot{\gamma}) = 0 \quad (4.87)$$

Ainsi, la longueur du vecteur vitesse $\frac{d\gamma}{dt}$ est une constante par rapport au temps.

De même, l'accélération est toujours nulle sur une droite d'un espace euclidien, de même la géodésique peut être aussi définie comme une courbe à accélération nulle.

Posons $\left\| \frac{d\gamma}{dt} \right\| = Cte = c$ et prenons $t_0 = 0$ l'origine des temps, alors la longueur de l'arc géodésique entre 0 et t est égale à :

$$(\text{long } \gamma) = \int_a^b \left[g\left(\frac{d\gamma}{dt}, \frac{d\gamma}{dt}\right) \right]^{\frac{1}{2}} dt = \int_0^t \left\| \frac{d\gamma}{dt} \right\| dt = ct \quad (4.88)$$

$$s(t) = \int_0^t \left\| \frac{d\gamma}{dt} \right\| dt = ct \quad (4.89)$$

Il en découle que le paramètre d'une géodésique est proportionnel à la longueur de l'arc géodésique.

La méthode d'extraction proposée se base sur le calcul des distances géodésiques (Avec simplification de Shamfer [103]) entre les quatre extrémités du caractère Tifinagh en question.

3.2. Traitement préliminaire

Le traitement préliminaire que nous considérons se compose de deux processus standards (i) l'élimination de bruit et (ii) la détection des contours (Figure 4.4).

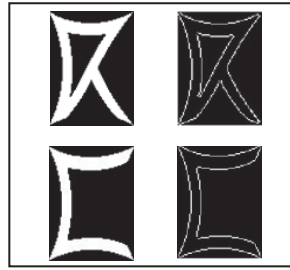


Figure 4.4 Exemple de détection de contour

- **Détection des extrémités**

Afin de repérer les extrémités, nous utilisons un algorithme qui parcourt le contour du caractère et détecte les points les plus proches des angles de l'image.

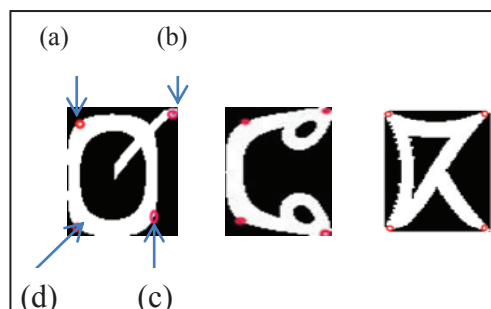


Figure 4.5 Exemple des extrémités des trois caractères Tifinagh

3.3. Calcul des descripteurs géodésiques

Nous avons appelé « descripteurs géodésiques » les distances entre les quatre extrémités du caractère divisées par leurs distances euclidiennes. On pose :

- $Dl_M(xy)$: la distance géodésique entre x et y ;
- d_{xy} : la distance euclidienne entre x et y ; et
- a,b,c et d les extrémités de chaque caractère (figure 4.6).

On appelle :

- 1^{er} descripteur métrique : $D1 = Dl_M(ab) / d_{ab}$
- 2^{ème} descripteur métrique : $D2 = Dl_M(ac) / d_{ac}$
- 3^{ème} descripteur métrique : $D3 = Dl_M(ad) / d_{ad}$
- 4^{ème} descripteur métrique : $D4 = Dl_M(bc) / d_{bc}$
- 5^{ème} descripteur métrique : $D5 = Dl_M(bd) / d_{bd}$

- 6^{ème} descripteur métrique $D6 = DLM(cd) / dcd$

Afin de calculer les distances géodésiques sur une image binaire, nous avons calculé la métrique en chaque pixel et on a utilisé l'équation (4.88) pour déterminer le chemin le plus court. Pour assurer une résistance aux changements d'échelle des descripteurs proposés, nous avons divisé les distances géodésiques de chaque chemin par les distances euclidiennes correspondantes.

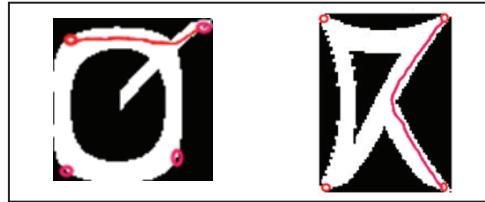


Figure 4.6 Exemple de chemins Géodésiques

Le tableau 4.4 ci-dessous présente les résultats obtenus pour les six descripteurs géodésiques présentés:

| | D1 | D2 | D3 | D4 | D5 | D6 |
|--|------|------|------|------|------|------|
| | 2.12 | 1.31 | 1.29 | 1.21 | 1.23 | 1.02 |
| | 1.10 | 1.62 | 1.13 | 1.38 | 1.11 | 1.40 |
| | 2.02 | 1.41 | 1.71 | 1.70 | 1.43 | 2.00 |
| | 1.07 | 1.04 | 1.03 | 1.41 | 1.12 | 1.73 |

Tableau 4.4. Résultats obtenus pour quelques caractères Tifinagh

Notons que les descripteurs proposés ont permis :

- Une distinction claire entre les caractères testés.
- Une distinction entre les caractères ayant des géométries proches et qui peuvent être obtenus par rotation les uns des autres, par exemples les caractères “Yars” et “Yass” dans le Tableau 4.5).



| | | | | | | |
|---|------|------|------|------|------|------|
|  | 1.13 | 1.48 | 1.11 | 1.22 | 1.40 | 1.20 |
|  | 1.23 | 1.39 | 1.12 | 1.20 | 1.5 | 1.12 |

Table 4.5. Descripteurs géodésiques relatifs aux caractères ‘Yars’ et ‘Yass’

Les descripteurs proposés demeurent également fiables vis-à-vis les changements d'échelle. (Tableau 4.6)




| | D1 | D2 | D3 | D4 | D5 | D6 |
|--|------|------|------|------|------|------|
|  | 1.19 | 1.44 | 1.07 | 1.15 | 1.41 | 1.05 |
|  | 1.22 | 1.39 | 1.10 | 1.18 | 1.49 | 1.11 |
|  | 1.23 | 1.39 | 1.12 | 1.20 | 1.5 | 1.12 |

Table 4.6 Descripteurs Géodésiques calculés pour différentes tailles du caractère ‘Yass’

4. Descripteurs de Courbure

Pour calculer le scalaire de courbure, nous devons d'abord calculer la métrique riemannienne en chaque point (pixel) de la surface ainsi que les symboles Christoffel [96].

La Métrique riemannienne est une matrice symétrique définie. À cause de la bilinéarité on trouve:

$$g_{ij} = g\left(\frac{\partial}{\partial s_i}, \frac{\partial}{\partial s_j}\right) \quad (4.90)$$

Pour les surfaces:

$$g_{11} = g\left(\frac{\partial}{\partial s}, \frac{\partial}{\partial s}\right) \quad (4.91)$$

$$g_{21} = g\left(\frac{\partial}{\partial t}, \frac{\partial}{\partial s}\right) \quad (4.92)$$

$$g_{12} = g\left(\frac{\partial}{\partial s}, \frac{\partial}{\partial t}\right) \quad (4.93)$$

$$g_{22} = g\left(\frac{\partial}{\partial t}, \frac{\partial}{\partial t}\right) \quad (4.94)$$

Pour les surfaces, huit symboles de Christoffel sont prévus, avec $i, j, k = 1, 2$:

$$\Gamma_{ij}^k = \frac{1}{2} g^{km} \left(\frac{\partial g_{jm}}{\partial i} + \frac{\partial g_{im}}{\partial j} - \frac{\partial g_{ij}}{\partial m} \right) \quad (4.95)$$

On utilise ici la sommation d'Einstein :

$$\Gamma_{ij}^k = \sum_{m=1}^2 \frac{1}{2} g^{km} \left(\frac{\partial g_{jm}}{\partial i} + \frac{\partial g_{im}}{\partial j} - \frac{\partial g_{ij}}{\partial m} \right) \quad (4.96)$$

Les 1,3 Tenseur de courbure:

$$R_{ijk}^l = \frac{\partial}{\partial i} \Gamma_{jk}^l + \Gamma_{ik}^m \Gamma_{jm}^l - \frac{\partial}{\partial j} \Gamma_{ik}^l - \Gamma_{ik}^l \Gamma_{im}^l \quad (4.97)$$

Le 0,4 Tenseur de courbure:

$$R_{ijkl} = R_{ijk}^m R_{lm} \quad (4.98)$$

Le 0,2 Tenseur de courbure:

$$R_{ij} = g^{kl} R_{kijl} \quad (4.99)$$

La scalaire courbure: $S = g^{ij} R_{ij} \quad (4.100)$

La figure 4.7 présente l'organigramme proposé pour le calcul des descripteurs de courbure.

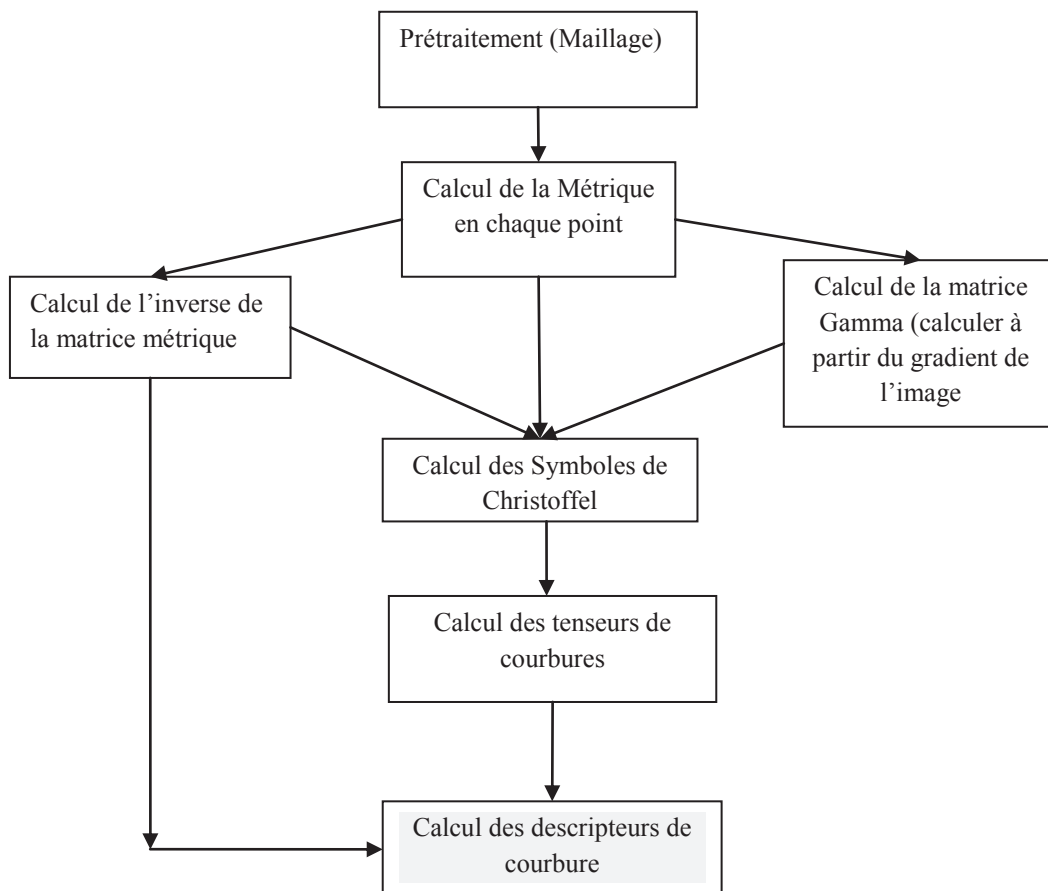


Figure 4.7 : Organigramme de l'algorithme de calcul des descripteurs de courbures

Nous nous sommes basé sur les algorithmes des géométries différentielles proposées par J.KERL [96] pour réaliser l'algorithme des descripteurs de courbures illustré ci-dessous.

Algorithme descripteurs de courbures

```

% on calcule l'inverse du tenseur métrique en chaque point
gdet = a*d - b*c;
invg(ss,tt,1,1) = d/gdet; invg(ss,tt,1,2) = -b/gdet;
invg(ss,tt,2,1) = -c/gdet; invg(ss,tt,2,2) = a/gdet;

% L'unité normale
N1 = cross(DDs1, DDt1);
N1hat = N1 / norm(N1);

% Enregistrement de la base tangente et des unités normales pour
plus tard
DDs (ss,tt,:) = DDs1;
DDt (ss,tt,:) = DDt1;
Nhat(ss,tt,:) = N1hat;

end
end

% Calculer des dérivés des coefficients métriques,
DD_g = zeros(ns,nt,2,2,2);
for j = 1:2
    for k = 1:2
        [DD_g(:,:,1,j,k), DD_g(:,:,2,j,k)] =
gradient(g(:,:,j,k),ds,dt);
    end
end

%% -- Calcul des symboles de Christoffel
Gamma = zeros(ns,nt,2,2,2);
for i = 1:2
    for j = 1:2
        for k = 1:2
            for m = 1:2
                Gamma(:,:,i,j,k) = Gamma(:,:,i,j,k) ...
+ 0.5 * invg(:,:,k,m) .* ( ...
DD_g(:,:,i,j,m) + ...
DD_g(:,:,j,i,m) - ...
DD_g(:,:,m,i,j) ...
);
            end
        end
    end
end

% Calcul de la matrice Gamma.
DD_Gamma = zeros(ns,nt,2,2,2,2);

```

```

for j = 1:2
    for k = 1:2
        for l = 1:2
            [DD_Gamma(:, :, 1, j, k, l), DD_Gamma(:, :, 2, j, k, l)] = ...
                gradient(Gamma(:, :, j, k, l), ds, dt);
        end
    end
end

% Les 1,3 tenseures de courbures
R13 = zeros(ns, nt, 2, 2, 2, 2);
for i = 1:2
    for j = 1:2
        for k = 1:2
            for l = 1:2
                for m = 1:2
                    R13(:, :, i, j, k, l) = R13(:, :, i, j, k, l) ...
                        + DD_Gamma(:, :, i, j, k, l) ...
                        - DD_Gamma(:, :, j, i, k, l) ...
                        + Gamma(:, :, i, k, m) .* Gamma(:, :, j, m, l) ...
                        - Gamma(:, :, j, k, m) .* Gamma(:, :, i, m, l);
                end
            end
        end
    end
end

% les 0,4 tenseurs de courbures:
R04 = zeros(ns, nt, 2, 2, 2, 2);
for i = 1:2
    for j = 1:2
        for k = 1:2
            for l = 1:2
                for m = 1:2
                    R04(:, :, i, j, k, l) = R04(:, :, i, j, k, l) ...
                        + R13(:, :, i, j, k, m) .* g(:, :, l, m);
                end
            end
        end
    end
end

- end
end

% les 0,2 tenseurs de courbure:
R02 = zeros(ns, nt, 2, 2);
for i = 1:2
    for j = 1:2
        for k = 1:2
            for l = 1:2
                R02(:, :, i, j) = R02(:, :, i, j) ...
                    + invg(:, :, k, l) .* R04(:, :, k, i, j, l);
            end
        end
    end
end
end

```

```
% Calcul des descripteurs de courbures
S = zeros(ns,nt);
for i = 1:2
    for j = 1:2
        S(:,j) = S(:,j) + invg(:,j,i,j) .* R02(:,j,i,j);
    end
end
ss=S;
```

Les descripteurs de courbures constituent une extension des descripteurs métriques, mais avec un nombre de paramètres plus grand et avec prise en charge des changements des courbures de l'image.

Le tableau 4.7 illustre les valeurs calculées des descripteurs de courbures (Curvature descriptors) pour des caractères Tifinagh.

| | | | | | | | | |
|--|--------|--------|--------|--------|--------|--------|--------|--------|
| | 0.0112 | 0.0181 | 0.0052 | 0.0026 | 0.0023 | 8.0321 | 0.0222 | 5.3365 |
| | 5.8580 | 0.0252 | 8.7285 | 0.0030 | 0.0014 | 0.0000 | 0.0038 | 0.0074 |
| | 0.0088 | 0.0072 | 0.0115 | 0.0124 | 0.0109 | 0.0085 | 0.0236 | 0.0017 |
| | 0.0010 | 0.0306 | 0.0011 | 0.0107 | 0.0260 | 0.0149 | 0.0146 | .8354 |
| | 5.8444 | 0.0436 | 0.0021 | 0.0008 | 0.0150 | 0.0410 | 5.8350 | 0.0381 |
| | 5.2946 | 0.0245 | 0.0005 | 0.0014 | 0.0417 | 5.3199 | 0.0036 | .0072 |
| | 5.8552 | 0.0334 | 0.0117 | 0.0057 | 0.0107 | 0.0001 | 0 | 0 |
| | 0 | 0 | 0.0002 | 0.0089 | 0.0223 | 5.3336 | 0 | 0.0000 |

Tableau 4.7 descripteurs de courbures pour quelques caractères Tifinagh.

5. Conclusion

La complexité de la géométrie riemannienne a rendu ce champ de recherche restreint au mathématicien ce n'est que dernièrement avec les travaux de plusieurs chercheurs comme Servastas, J.KERL, X. Penneec, etc.. [98,99] que l'informatisation de la géométrie riemannienne a commencé. Inspiré de ces travaux on est arrivé à développer trois descripteurs de formes simples à extraire et faciles à employer :

- Les descripteurs métriques.
- Les descripteurs géodésiques
- Les descripteurs de courbure

Chacun de ces descripteurs présente des avantages et des inconvénients que nous allons discuter dans la partie expérimentale.

Chapitre 5 :

Systemes de reconnaissances réalisés & Résultats expérimentaux

1. Introduction

Ce chapitre regroupe l'ensemble des méthodes de reconnaissances développées ou renouvelées dans le cadre de cette thèse. Il est divisé en deux parties : la première partie est entièrement dédiée à la reconnaissance des caractères, la deuxième partie présente les méthodes et les approches proposées, permettant d'améliorer la reconnaissance des objets 3D couleur moyennant une approche 2D/3D.

Concernant la reconnaissance des caractères, nous présenterons trois systèmes de reconnaissance de caractères Tifinagh basés sur les descripteurs riemanniens proposés dans le chapitre 4 à savoir :

- Les descripteurs métriques testés par les SVM et les réseaux de neurones pour la reconnaissance des caractères Tifinagh ;
- Les descripteurs géodésiques testés par les arbres de décision et les réseaux de neurones pour la reconnaissance des caractères Tifinagh.
- Les descripteurs de courbure testés par les réseaux de neurones et les réseaux Neuro-flou pour la reconnaissance des caractères Tifinagh.

Pour ce qui est de la reconnaissance d'objets 3D couleur, nous avons étudié l'influence de plusieurs paramètres sur les taux de reconnaissances et nous avons :

- Réalisé une étude comparative entre les RNA en séries et les RNA en parallèle avec le RNA multi-sorti que nous avons proposé ;
- Étudié l'influence des descripteurs de formes et leurs combinaisons sur les taux de reconnaissances et nous avons proposé une entrée hybride composée des moments de Zernike, Hu et des distances géodésiques ;
- Teste l'influence des espaces de couleurs sur les taux de reconnaissances et nous avons proposé d'élargir l'espace Lab en Lab-étendu.

2. Reconnaissance des caractères

Dans le cadre du processus de reconnaissance des caractères, nous avons proposé trois méthodes de reconnaissances : Une première basée sur les descripteurs métriques et testée par les RNA et les SVM [3], une deuxième qui intègre les descripteurs géodésiques avec une hybridation des RNA et des Arbres de décision [4] et une troisième avec les descripteurs de Courbures et l'hybridation des RNA et des Fuzzy-Neural [5].

2.1.Descripteurs métriques avec RNA et SVM

2.1.1. Segmentation

La segmentation est définie, comme étant l'opération qui cherche à décomposer une image de texte en pseudo-images de symboles individuels. Le résultat de cette opération est une forme isolée à partir d'une image et qui pourrait être un caractère ou non.

L'écriture amazighe n'est pas cursive. Cela facilite l'opération de segmentation d'une ligne de texte en caractères. Nous avons utilisé les techniques de détection des régions isolées pour segmenter chaque ligne de texte en caractères (Figure 5.1).

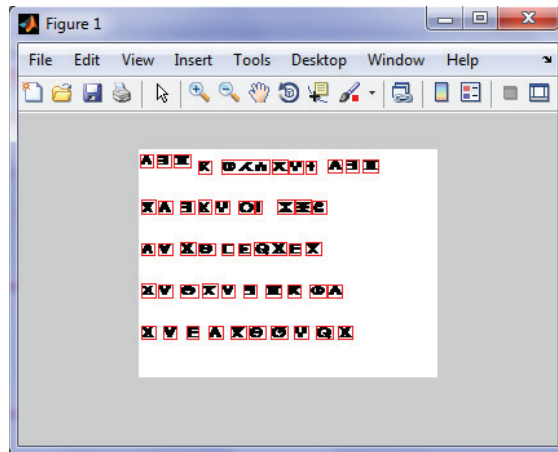


Figure 5.1 : Segmentation des caractères Tifinagh.

2.1.2. Base d'apprentissage

Pour l'apprentissage et les tests, nous avons utilisé deux bases de données, la première de (Y. Ouguengay, M. Taalabi 2009) [24] est composée de 2175 caractères Tifinagh imprimés avec différents styles et tailles (Figure 5.2), la deuxième est une base de données locale composée de 330 caractères manuscrits.



Figure 5.2 : Exemple d'un caractère Tifinagh de la base de 'Y.Ouguengay' pour différents styles et tailles.

2.1.3. Apprentissage et reconnaissance

Nous avons présenté dans le chapitre 4 les descripteurs métriques comme une nouvelle approche d'extraction d'attributs. Pour les mettre en évidence, nous avons choisi de les tester à l'aide de deux classificateurs connus par leur robustesse et leurs caractéristiques uniques : Les réseaux de neurones et les SVM (les séparateurs à vaste marge).

Nous avons testé notre approche sur un grand nombre de caractères de la base d'apprentissage. Nous avons utilisé un réseau de neurones multicouches à apprentissage supervisé, entraîné par la rétro propagation du gradient. La topologie des RNAs utilisée est la suivante :

- La couche d'entrée possède 5 neurones qui correspondent à l'ensemble des caractéristiques extraites ;
- La couche de sortie est composée de 33 neurones qui correspondent aux nombres de classes ;
- Pour la couche cachée, il n'y a pas de règle absolue qui permet de déterminer avec exactitude le nombre de neurones à utiliser dans cette couche. En effet, le nombre de nœuds cachés varie selon les applications, de ce fait, ce nombre doit être déterminé expérimentalement. Dans notre cas nous avons utilisé un nombre de neurones égal à la moyenne des neurones de la couche d'entrée et de la couche de sortie ($N_c = (N_s + N_e) / 2$), et nous l'avons modifié expérimentalement selon les tendances du taux d'apprentissage.

Les paramètres d'apprentissage pour les RNA que nous avons utilisés sont les suivants :

- L'initialisation du RNA se fait avec des poids aléatoires ayant des valeurs comprises entre -1.0 et +1.0. La fonction de transfert que nous avons utilisée est la fonction logistique standard (la sigmoïde) ;

- La valeur du pas τ de l'algorithme de la descente du gradient définie dans le chapitre 3 est fixée à 0.1. Il est à noter que le choix d'une valeur trop grande du pas risque de faire diverger le processus de détection du minimum optimal, et inversement, une valeur du pas trop petite augmente les temps de calcul ;
- Le réseau est formé par le taux d'apprentissage 0,05 ;
- Le nombre d'itérations est fixé à 1000.

Pour les SVM nous avons travaillé avec le Toolbox Matlab de (Dr Gavin & C. Cawley) qui met en œuvre la méthode de Vapnik proposée pour les séparateurs à vaste marge mono class avec un noyaux polynomial [101].

Les Tableaux 5.1 et 5.2 représentent respectivement les taux de reconnaissances obtenus pour les caractères imprimés et manuscrit.

| Nombre de caractères imprimés à identifier | SVM | Réseaux de neurones |
|--|-----|---------------------|
| 10 | 99% | 99% |
| 20 | 88% | 67% |
| 25 | 83% | 0% |
| 33 | 78% | 0% |

Tableau 5.1 : Évaluation de taux de reconnaissance en fonction du nombre de caractères imprimés à identifier.

| Nombre de caractères manuscrit à identifier | SVM | Réseaux de neurones |
|---|-----|---------------------|
| 10 | 97% | 98% |
| 20 | 84% | 67% |
| 25 | 81% | 0% |
| 33 | 71% | 0% |

Tableau 5.2 : Évaluation de taux de reconnaissance en fonction du nombre de caractères manuscrits.

On peut noter que malgré la taille de la base d'apprentissage qui n'a pas dépassé 15 échantillons par caractères, les descripteurs proposés ont donné de très bons résultats pour un nombre de caractères inférieurs à 20 (pour les deux classificateurs), et des résultats acceptables pour la totalité des caractères Tifinagh dans le cas des SVM. Il est clair que le

nombre de paramètres réduit des descripteurs métriques (cinq paramètres) est insuffisant pour entraîner un réseau de neurones de 33 sorties. Les taux de reconnaissance obtenus par les descripteurs métriques restent loin des résultats escomptés.

2.2.Descripteurs de Courbures avec hybridation des RNA et des Fuzzy-Neural

Nous avons mis en œuvre dans les sections précédentes un système de reconnaissance de l'écriture amazighe basé sur les descripteurs métriques et qui donne des résultats satisfaisants. Nous proposons dans cette section une amélioration à ce système de reconnaissance en utilisant les descripteurs de courbures et un classificateur hybride

Les descripteurs de courbures constituent une extension des descripteurs métriques. Ils nous ont permis d'augmenter la taille du vecteur d'entrée des classificateurs. Leur complexité nous a incités à chercher un classificateur plus souple que les SVM pour le combiner avec les RNA.

Pour éliminer toute confusion qui peut être due à la partie secondaire des caractères composés ou aux caractères circulaires (Fig.5.3), nous avons choisi de travailler avec un classificateur hybride composé d'un Réseau Neuro-Flous (RNF) et de trois réseaux de neurones classiques. D'une part, le classificateur RNF avec ses caractéristiques souple et discriminatoire nous a permis de séparer efficacement les caractères en trois classes. D'autre part, les réseaux de neurones, avec leur capacité de détecter implicitement les relations linéaires et non linéaires complexes ainsi que toutes les interactions possibles entre les variables prédictives, permettent la reconnaissance des caractères.

En pratique, nous avons utilisé un réseau de neurones multicouche à apprentissage supervisé, entraîné par la rétro-propagation du gradient avec les descripteurs de courbures comme entrées.

Pour le classificateur RNF, nous avons utilisé l'algorithme ANFIS et les données d'entrée qui suivent:

- Le nombre de composantes.
- La distance entre les centres des deux composantes, si elles existent.
- Existence d'une géométrie circulaire.

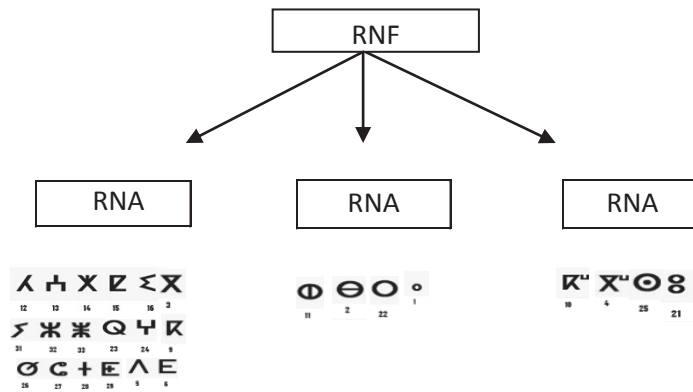


Fig. 5.3 : Processus de classification utilisé avec les descripteurs de Courbures

Nous avons testé cette approche avec la même base de données de Y.Ouguengay [26].

| Nombre de caractères à identifier | Descripteurs Métriques | Descripteurs de Courbures | |
|-----------------------------------|------------------------|---------------------------|----------|
| | SVM | (RNA) | RNA& RNF |
| 15 | 97% | 98% | 99% |
| 23 | 84% | 93% | 97% |
| 29 | 81% | 84% | 96% |
| 33 | 71% | 81% | 95,5% |

Table. 5.3 : Taux de reconnaissance en fonction du nombre de caractères à identifier

A noter que malgré la taille de la base, qui est de 15 échantillons par caractère, les descripteurs suggérés ont abouti à des résultats efficaces en utilisant les réseaux de neurones 81%. L'intégration des RNF a permis d'augmenter les taux de reconnaissances d'environ 14%.

Le Tableau 5.4 illustre la matrice de confusion obtenue par l'intégration des RNA et des (RNF) appliqués sur les descripteurs de courbures. En analysant cette matrice de confusion, nous constatons qu'il n'existe pas un type d'erreurs pertinent. Les erreurs proviennent principalement des styles d'écriture s'avérant parfois très complexe.

| 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | C | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|----|----|----|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 1 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 2 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 3 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 4 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 5 | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 0 | 0 | 0 | 0 | 0 | 6 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 7 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 11 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | |
| 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 19 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 22 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 2 | 0 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 23 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 24 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 25 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 26 |
| 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 |
| 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 28 |
| 0 | 0 | 0 | 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 29 | |
| 0 | 0 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 | |
| 0 | 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 31 | |
| 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 32 | |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 33 | |

Tableau 5.4. Matrice de confusion obtenue par l'intégration des RNA et des Fuzzy-Neural classificateurs appliqués sur les descripteurs de courbures.

2.3.Descripteurs Géodésiques testés par les RNA et les arbres de décisions.

L'idée de base consiste à intégrer une information qui permet de remédier aux problèmes de classifications issus de la ressemblance entre plusieurs caractères. Nous avons proposé dans le chapitre 4 les descripteurs géodésiques qui se basent sur le calcul des distances géodésiques entre les extrémités du caractère. À première vue les descripteurs géodésiques nous ont permis une distinction claire seulement pour les caractères de la figure.5.4. Cependant, la confusion subsiste quand il s'agit de caractères composés (figure 5.5) ou d'autres caractères qui ont une forme circulaire (figure 5.6).



Figure 5.4. Caractères reconnus directement par les descripteurs Géodésiques



Figure 5.5 : Caractères composés



Figure .5.6 : Caractères circulaire

Pour faire face aux particularités des caractères Tifinagh. Nous avons choisi de travailler avec un classificateur hybride, composé des arbres de décision et des réseaux de neurones.

D'une part, les arbres de décision ont une caractéristique discriminatoire qui nous a permis de séparer les caractères en quatre classes (figure 5.7). D'autre part, les réseaux de neurones permettent la reconnaissance des caractères, grâce à leur capacité de détecter implicitement les relations non linéaires complexes entre les variables dépendantes et indépendantes et à détecter toutes les interactions possibles entre les variables prédictives [17][18].

En pratique, nous avons utilisé un réseau de neurones multicouches (deux couches) avec apprentissage supervisé, entraîné par la rétro-propagation du gradient.

Pour l'arbre de décision, nous avons utilisé les règles suivantes :

- R1 : Après détection du nombre de motifs N dans l'image. Si $N > 1$, alors: R22, si non: R21.
- R22 : si la taille du premier motif est deux fois plus grande que la taille du second motif, alors: N3, si non: N4.
- R21 : si le ratio des distances géodésiques ($D1/D3$) et ($D2/D6$) sont compris entre 0,8 et 1,2 alors: N2, si non: N1.

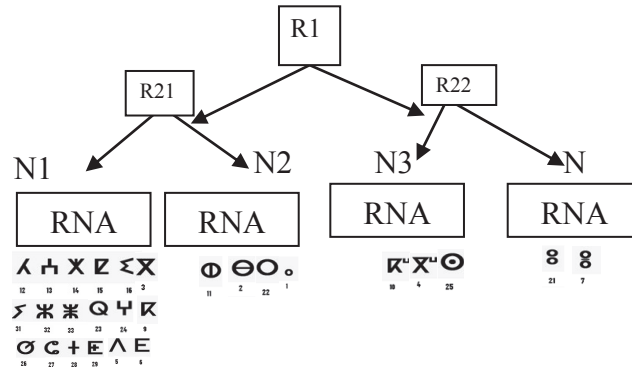


Figure 5.7. Schéma du processus de classification utilisé avec les descripteurs géodésiques

Pour la phase de test de ce travail, nous avons utilisé la même base d'apprentissage utilisée précédemment (Figure 5.2).

Chacun des caractères sera déterminé en utilisant les descripteurs géodésiques pour la description et sera identifié premièrement par les réseaux de neurones et aussi par la combinaison de ces derniers avec les arbres de décision. Nous avons testé notre approche sur différents caractères de la base de données. Le tableau 5.5 donne un aperçu sur les taux de reconnaissance des objets de la base de données.

| Nombre de Caractères à identifier | Réseau de Neurones (RNA) | RNA&Arbres de décisions |
|-----------------------------------|--------------------------|-------------------------|
| 10 | 98% | 99% |
| 20 | 93% | 96% |
| 25 | 81% | 95% |
| 33 | 71% | 94% |

Tableau 5.5. Taux de reconnaissance en fonction du nombre de caractères à identifier

Il est à noter que malgré la taille de la base qui est de 15 échantillons de chaque caractère, les descripteurs géodésiques ont fait preuve d'une grande efficacité en utilisant

les réseaux de neurones. L'intégration des arbres de décision a rendu les taux de reconnaissance remarquablement plus élevés.

Le Tableau 5.6 illustre la matrice de confusion obtenue par le système amélioré sur la base des patterns de la graphie amazighe. En analysant cette matrice de confusion, nous constatons que les erreurs remarquées parviennent principalement :

- Des styles d'écriture testés qui sont des fois assez embrouillés (Figure 5.2) et présentent des ressemblances géométriques issues des similarités morphologiques entre certains caractères : yan, yar , ya , yazz et yab.
- L'existence des caractères composés. Sans oublier la taille de la base d'apprentissage qui reste assez minime.

2.4.Comparaison de méthodes proposées

Nous avons proposé trois nouveaux descripteurs de formes et nous avons testé leurs performances à l'aide de plusieurs classificateurs pour but de trouver le meilleur système de reconnaissance des caractères Tifinagh. Le tableau suivant présente la totalité des résultats obtenus au cours de nos recherches ainsi que les taux de reconnaissances obtenus pour les moments de Zernike.

| Nombre de caractères à identifier | Descripteurs Métriques | Descripteurs de Courbures | | Descripteurs Géodésiques | | Moments de Zernike | |
|---|------------------------|---------------------------|----------------|--------------------------|---------------------------|--------------------|-----|
| | SVM | RNA | RNA & Fuzzy NN | (RNA) | RNA & Arbres de décisions | RNA | SVM |
| 15 | 97% | 98% | 99% | 98% | 99% | 99% | 99% |
| 23 | 84% | 93% | 97% | 93% | 96% | 93% | 94% |
| 29 | 81% | 84% | 96% | 81% | 95% | 91% | 92% |
| 33 | 71% | 81% | 95,5% | 71% | 94% | 84% | 87% |
| Complexité Nb d'opération par pixel pour chaque descripteur | 4op/pixel | 11op/pixel | | 4.5 op/pixel | | 5 op/pixel | |
| Nb de paramètres de chaque descripteur | 5 | 16 | | 6 | | 16 | |

Tableau 5.7 : Tableau récapitulatif des résultats obtenu

Il est clair que les descripteurs métriques sont les moins performants, mais avec un temps d'exécution minime ; les descripteurs de courbures présentent le meilleur taux de reconnaissances quand ils sont utilisés avec les ANN et les Fuzzy NN, mais avec un nombre

| 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | C | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|----|----|----|----|----|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 14 | 1 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 11 | 0 | 2 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 16 | 0 | 0 | 3 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 4 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 5 | | | |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 0 | 0 | 0 | 0 | 0 | 6 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 11 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | | |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | | |
| 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 19 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 4 | 22 | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 2 | 0 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 23 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 24 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 25 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 26 | |
| 0 | 0 | 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 | |
| 0 | 0 | 0 | 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 28 | |
| 0 | 0 | 0 | 0 | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 29 | |
| 0 | 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 | |
| 0 | 0 | 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 31 | |
| 1 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 32 | |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 33 |

Figure 5.6 Matrice de confusion obtenue par le système amélioré sur la base des patterns de la graphie amazighe

d'opérations énorme. Les descripteurs géodésiques présentent le meilleur rapport (taux de reconnaissance/complexité) avec 94% de taux de reconnaissance en seulement 4.5 op/pixel.

On peut aussi, noter l'influence positive des classificateurs hybrides sur les taux de reconnaissances.

3. Reconnaissance d'objets couleur

La reconnaissance d'un objet couleur est bien plus complexe que celle d'un caractère : (i) la détection d'un objet dans une scène est bien plus complexe que la détection d'un caractère dans un texte, (ii) l'angle de vue influence sensiblement la représentation de l'objet et (iii) les descripteurs de forme s'avèrent moins efficaces dès que la complexité de l'objet augmente. Dans cette partie, nous allons présenter les différents systèmes de reconnaissance d'objets couleurs que nous avons proposés [6].

3.1. Prétraitement

3.1.1. Acquisition des images (Coil-100 Data Base)

Nous avons testé notre approche de détection et de reconnaissance d'objets couleur sur la base de données COIL-100 [102]. Cette base de données contient des images couleur de 100 objets avec 72 vues différentes (vues caractéristiques) pour chaque objet. La figure 5.8 présente un objet de la base pour différentes vues, alors que la figure 5.9, présente quelques exemples d'objets de la base de données.



Fig.5.8 : Un objet de la base pour différentes vues.

3.1.2. Segmentation

La segmentation des images consiste à regrouper les pixels des images qui partagent une même propriété pour former des régions connexes. Il existe deux familles d'approches que l'on peut faire coopérer : les approches « contours » et les approches « régions » [2,5]. La



Figure 5.9: La base de données coil-100 de l'université de Columbia

combinaison de la méthode de segmentation couleur et l'étiquetage en composantes connexes permettent une détection efficace des objets dans une scène même avec un fond faiblement texturé. Dans notre cas, après avoir extrait l'image binaire, on calcule le centre de chaque objet et on extrait la partie de l'image couleur qui le contient (Fig.5.8).

Au niveau de l'étiquetage nous avons utilisé le processus standard de numérotation des régions isolés avec l'algorithme suivant :

| Algorithme Etiquetage |
|--|
| <p>L'image à étiqueter étant codée par la matrice A de taille $N \times M$, on construit la suite (A_i) d'images ($i=0, \dots, n$) avec $A_0 = A$ (image initiale) et $A_n =$ image étiquetée finale. L'algorithme est le suivant: Pour Tous les points de A Faire en parallèle Si $A(i,j) = 1$ Alors $A(i,j) = (i-1)*N + j$ Fin si Fin pour $k = 1$ Répéter Pour Tous les points de A_{k-1} Faire en parallèle Si $A_{k-1}(i,j) \neq 0$ Alors $A_k(i,j) =$ minimum sur le voisinage immédiat de $A_{k-1}(i,j)$ Sinon $A_k(i,j) = 0$ Fin si Fin pour Jusqu'à $A_{k-1} = A_k$</p> |

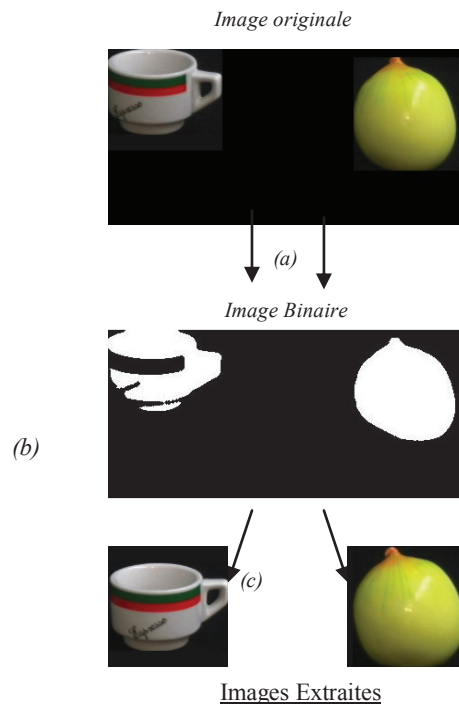


Figure 5.10 : Etape d'étiquetage et de segmentation :

Avec :

- a) : Segmentation.
- b) : étiquetage et détection des objets qui existent dans l'image,
- c) : Extraction (on détecte le centre de chaque objet et on extrait le rectangle qui le contient dans l'image d'origine

3.2.Méthode classique de la reconnaissance d'objets couleur

La méthode classique consiste à détecter la forme globale de l'objet à l'aide de sa représentation binaire, puis à séparer les objets de la même forme à l'aide d'une fonction de couleur ou à l'aide d'histogrammes de couleur.

3.2.1. Reconnaissance des images binaires

Pour identifier les objets binaires, on calcule les moments de Zernike sur des images binaires où la fonction $f(x, y)$ de l'image binaire est égale à 1 ou à 0 (Figure.5.11).



Figure 5.11: Représentation binaire de deux objets de la base Coil-100

La figure 5.12 représente les moments de Zernike (d'ordre 1 à 8) appliqués sur les images binaires des objets présentés dans la figure 5.12. On peut noter que pour les deux objets, les valeurs des moments Zernike sont comparables.

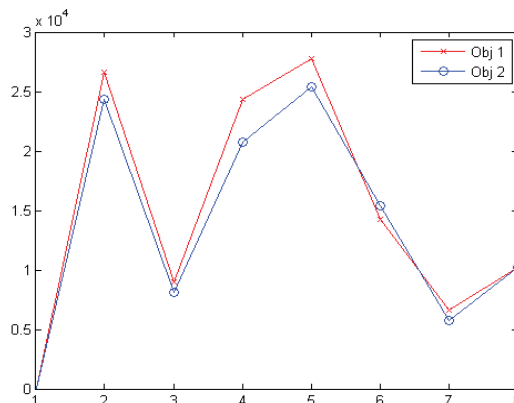


Fig.5.12 : Les Moments de Zernike sur les images binaires de la figure 5.11

3.2.2. Calcul de la fonction de couleur

Pour différencier des objets ayant la même forme, la méthode classique consiste à calculer les descripteurs de formes sélectionnés sur des images couleur où la fonction $f(x, y)$ calcule l'apport des trois niveaux de couleurs sur la matrice traitée. Souvent on utilise la formule:

$$f(x, y) = 0.3R + 0.6V + 0.1B \quad (5.1)$$

3.2.3. Inconvénients de la méthode classique

En pratique, cette méthode présente deux inconvénients majeurs pour la reconnaissance des objets 3D (décrits dans la Fig.5.13 et Fig.5.14). La représentation d'un objet par différents angles de vues, crée des confusions entre les objets présentés en binaire. Nous remarquons dans la Fig.5.13 deux objets différents qui ont pratiquement la même représentation binaire.

Tandis que même avec les meilleurs algorithmes de segmentation et de détection, la présence d'ombre et de déformation spatiale provoque des pertes d'informations (Fig.5.14).

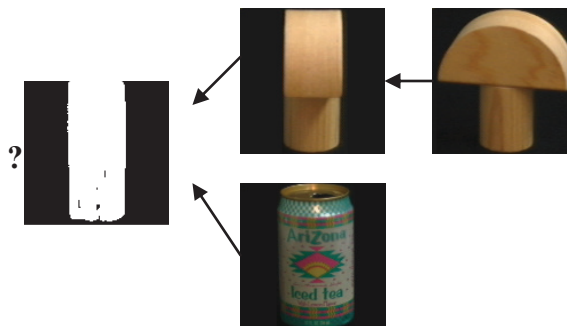


Fig.5.13 : La conversion en binaire donne une représentation incomplète de l'objet



Fig. 5.14: Perte d'information

3.3.Méthodes proposées

Les méthodes de reconnaissance basées sur les descripteurs de forme sont dans la pratique appliquées à des images binaires ainsi qu'aux photos en niveaux de gris. Les méthodes proposées consistent à calculer les DF en parallèle sur les trois niveaux de couleurs RVB, ce qui nous permettra de maintenir le maximum d'informations concernant les couleurs et d'augmenter le nombre de paramètres significatifs présenté aux RNAs (Fig.5.15). Nous avons joué sur la topologie externe des RNA et sur la combinassions des descripteurs de formes pour maximisé le rendement des systèmes de reconnaissances d'objets couleurs.

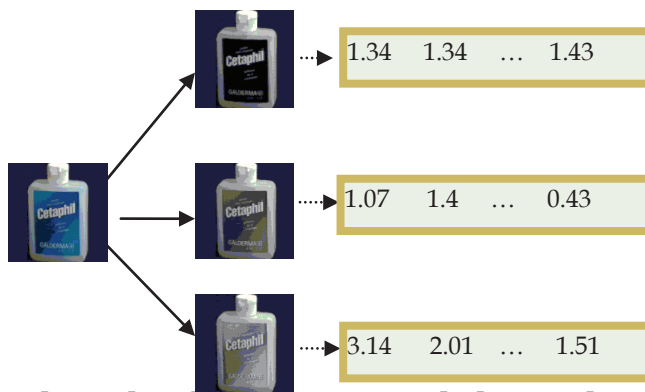


Figure 5.15: Séparation des trois niveaux de gris : quelques valeurs des moments de Zernike pour les trois niveaux gris.

3.3.1. Première méthode

Pour avoir une idée sur les caractéristiques et la complexité de l'algorithme de rétro-propagation du gradient et pour pouvoir détecter les paramètres de départ (nombre de neurones cachés, les meilleurs descripteurs de formes à intégrer), nous avons commencé par une série d'expériences et de tests.

Pour la première expérience, nous avons calculé les moments de Zernike pour chaque niveau de gris et nous les avons regroupés dans un seul vecteur de façon à ce qu'ils représentent une entrée unique du réseau de neurones (Fig. 5.16).

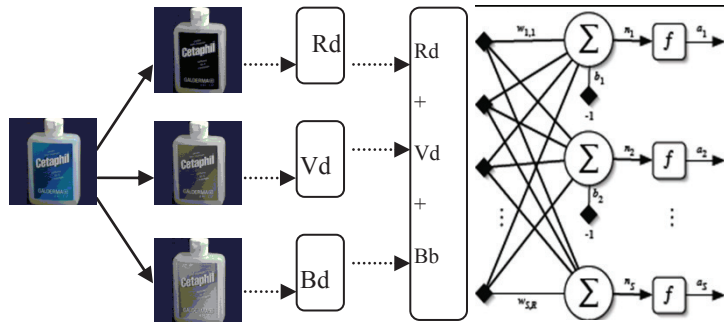


Fig. 5.16: Hu & Zernike en série pour les trois niveaux de gris

3.3.1.1. Expérience N1

Nous avons testé notre approche de détection et de reconnaissance d'objets couleur sur la base de données COIL-100 [8]. Pour notre première expérience, nous avons choisi comme données d'entrée du réseau de neurones les 15 premiers moments de Zernike de chaque image. Pour le nombre de neurones de la couche cachée, le meilleur taux d'apprentissage est obtenu pour un nombre de neurones égal à 10. La figure 5.17 illustre l'avancement de l'apprentissage du réseau de neurones.

Les paramètres d'apprentissage pour les RNAs que nous avons utilisés sont les suivants:

- L'initialisation du RNA se fait avec des poids aléatoires ayant des valeurs comprises entre -1.0 et +1.0. La fonction de transfert que nous avons utilisée est la fonction sigmoïde ;
- La valeur du pas de l'algorithme de la descente du gradient définie dans le chapitre 4 a été fixée expérimentalement à 0.1 ;
- Le taux d'erreur à atteindre est de 0,001 ; et
- Le nombre d'itérations est limité à 5000

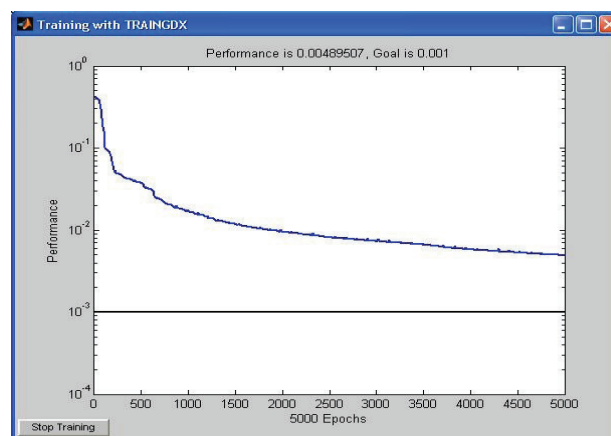


Figure 5.17 : Avancement de l'apprentissage du réseau de neurone

L'erreur quadratique moyenne (Mean Squared Error: MSE) est la moyenne des carrés de la différence entre les sorties (normalisées) et les cibles. Pour tous les tests effectués, nous avons remarqué un taux d'erreur inférieur à 10^{-2} .

Le tableau 5.8 représente les résultats obtenus dans cette expérience pour une base d'apprentissage constituée des 72 vues de chaque objet et d'une base de test constituée de 50 vues pour chaque altération. La robustesse obtenue est excellente vis-à-vis de la présence de bruit et la variation de luminance. La robustesse est correcte dans le cas d'un fond texturé.

Nous avons testé notre approche pour différentes tailles de la base d'apprentissage.

| Types d'image | IA | INA | Lu | FT | IC | IB |
|---|-------|------|-----|-----|------|-----|
| Taux de reconnaissance par réseaux de neurones | 99% | 91% | 85% | 81% | 91% | 99% |
| Taux de reconnaissance par vecteur support machine [15] | 99,6% | ---- | 74% | 60% | ---- | 99% |

Tableau 5.8 : Taux de reconnaissances obtenus par les Réseaux de neurones et les SVM pour différentes altérations

Avec :

IA : Images (Angle de vue) présentées en phase d'apprentissage.

INA : Images (Angle de vue) non présentées en phase d'apprentissage.

IC : Images composées (Fig 5.18)

IB 10%: Images avec un fond Bruité (Fig 5.19)

Lu : Luminance.

FT : Fond texturé.



Fig.5.18 : Image composée



Fig.5.19 : Image Bruitée

Le tableau 5.9 présente les taux de reconnaissances, les temps d'apprentissages et de reconnaissances en fonction de la taille de la base d'apprentissage.

| Taille de la base | Taux de reconnaissance | Nb Itération ~ Objectif 10^{-2} | MSE~i (Erreur quadratique) | Temps D'apprentissage | Temps Reconnaissance |
|-------------------|------------------------|-----------------------------------|----------------------------|-----------------------|----------------------|
| 700 | ~99% | ~500 | 10^{-3} | < 2 min | < 1 s |
| 1100 | ~97% | ~850 | $3 \cdot 10^{-2}$ | <3 min | < 1 s |
| 1600 | ~91% | ~1600 | $7 \cdot 10^{-2}$ | <5 min | < 2 s |
| 2100 | ~82% | ~3000 | $7 \cdot 10^{-2}$ | <6 min | < 2 s |
| 3000 | ~72% | ~5000 | $9 \cdot 10^{-2}$ | <8 min | <3 s |

Tableau 5.9 : Bilan global des résultats obtenus pour un PC (Core-Duo 2Ghz, 1Go de RAM)

On peut noter que le taux de reconnaissance diminue quand le nombre d'objets à identifier augmente, ce qui montre l'insuffisance de ce système dans le cas des bases de données de grande taille.

3.3.1.2. Expérience N2

Pour trouver l'entée adéquate du RNA, nous avons établi une étude comparative entre les descripteurs de formes couramment utilisés : Zernike, Hu, Fourier, Radon, puis nous avons essayé de les combiner afin de trouver un descripteur hybride permettant d'améliorer les taux de reconnaissances (Tableau 5.10).

| Descripteurs de formes sur des images en Gris | Taux de Reconnaissance (Nb. images/Nb. Classes) | | |
|---|---|---------|----------|
| | 720/10 | 1080/15 | 1440 /20 |
| Moments de Zernike (11 premiers) | 99% | 93% | 89% |
| Transformer de Fourier | 77% | 58% | 53% |
| Transformer de Radon | 87% | 83% | 71% |
| Moments de Hu (7 premiers) | 99% | 86% | 81% |
| Moments de Hu+Zernike en séries | 99% | 98% | 94% |

Tableau 5.10 : Taux de reconnaissances obtenus par les différents DF testés

Le tableau 5.10 confirme la complémentarité des moments de Hu et de Zernike, leur intégration a permis d'améliorer notablement les taux de reconnaissances. Dans le reste de nos expériences les objets à identifier seront représentés par des vecteurs composé de la combinassions des moments de Hu et de Zernike.

3.3.2. Deuxième méthode : Influence de la topologie externe des RNA

3.3.2.1. Expérience N1: Réseaux de neurones en parallèle

La première méthode que nous avons adoptée est basée sur l'intégration de trois Réseaux de neurones, un RNA pour chaque niveau de gris (Fig.5.20). Les RNA utilisés sont basés sur l'algorithme de rétro-propagation du gradient. Les entrées des réseaux de neurones sont les 9 premiers moments de Zernike complétés par les trois premiers moments de Hu.

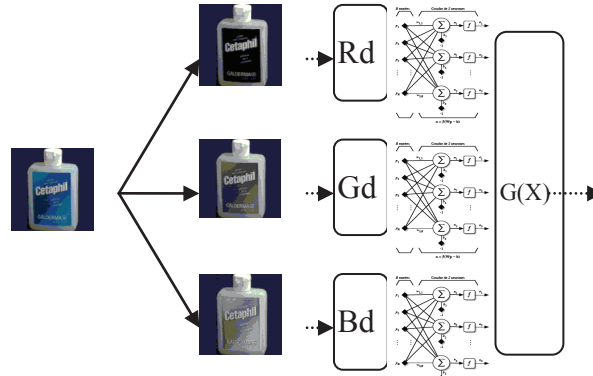


Fig.5.20: Zernike & Hu avec trois RN et une fonction de sortie globale G(x)

Avec :

- R_d : Les moments de Hu et Zernike sur le niveau Rouge
- G_d : Les moments de Hu et Zernike sur le niveau Vert
- B_d Les moments de Hu et Zernike sur le niveau Bleu

Les sorties des trois réseaux de neurones sont regroupées par la fonction G donnée par $G(X) = \max(X_i)$ et $X_i = x_{i1} + x_{i2} + x_{i3}$ pour $i = 1$ à n

Avec :

- x_{i1} est la $i^{\text{ème}}$ sortie du premier RNA.
- x_{i2} est la $i^{\text{ème}}$ sortie du second RNA.
- x_{i3} est la $i^{\text{ème}}$ sortie du troisième RNA.

3.3.2.2. Expérience N°2: RNA multi-sortis avec fonction de validation

Il arrive que les réseaux de neurones fournissent plusieurs sorties qui dépassent le seuil de reconnaissance, c'est à dire que le réseau donne plusieurs réponses (Tableau 5.11). Pour remédier à ce problème, nous proposons un système qui reproduit le même mécanisme que le système précédent, sauf que celui-ci est entraîné pour fournir des résultats supplémentaires : au lieu de détecter seulement la classe à laquelle appartient l'objet, le réseau est entraîné pour fournir la couleur dominante et la forme géométrique de l'objet. De cette manière, les

paramètres supplémentaires vont conduire à une meilleure décision en cas de confusion (Fig. 5.21).

| Numéro de la sortie | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----------------------------|---|---|------|------|---|---|---|------|---|------|------|----|----|----|-----|
| Exemple de sortie idéal | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0.04 | 0 | 0 | 0.94 | 0 | 0 | 0 | 0.1 |
| Exemple de sortie ambiguës | 0 | 0 | 0 | 0.35 | 0 | 0 | 0 | 0.37 | 0 | 0.06 | 0 | 0 | 0 | 0 | 0 |

Tableau 5.11 : Exemples de sorties d'un réseau de neurones à 15 sorties

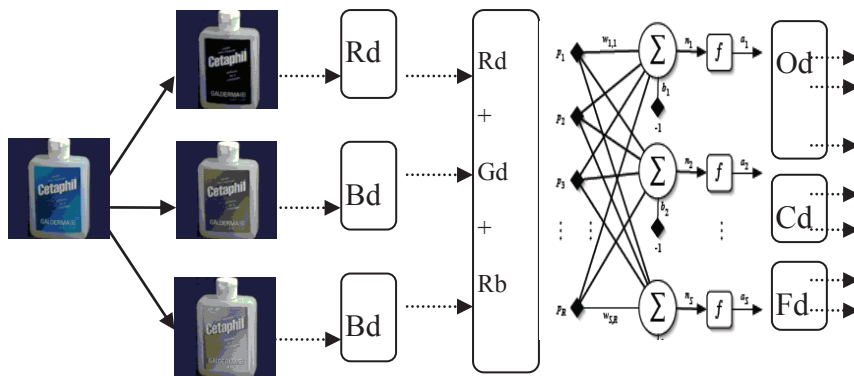


Fig. 5.21 :Hu & Zernike en série avec RN multi sorties avec couleur et formes détection.

Avec :

O_d : Les sorties relatives à la reconnaissance d'objet.

C_d : Les sorties relatives à la détection de couleur.

F_d : Les sorties relatives à la détection de la forme géométrique.

Pour remédier au problème d'ambiguïté déjà mentionné nous avons ajouté une étape de poste traitement qui intègre les règles de décision qui suivent :

- Augmentez de 15% la valeur des sorties O_d de tous les objets qui possèdent la couleur détectée (C_d).
- Augmentez de 10% la valeur des sorties O_d de tous les objets qui possèdent la forme géométrique détectée (S_d).

Le système proposé a permis la résolution du problème lié à l'objet ambigu du Tableau 5.11. Le tableau 5.12 représente les résultats obtenus par application des règles de calcul.

| | | | | | | | | | | | | | | | |
|-----------------------------|-----|------|---|------|-----|---|---|------|---|------|----|------|----|----|-----|
| Numéro de sortis | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Exemple de sorties ambiguës | 0 | 0 | 0 | 0.35 | 0 | 0 | 0 | 0.37 | 0 | 0.06 | 0 | 0 | 0 | 0 | 0 |
| Sortis après traitement | 0.1 | 0.15 | 0 | 0.35 | 0.1 | 0 | 0 | 0.62 | 0 | 0.06 | 0 | 0.15 | 0 | 0 | 0.1 |

Tableau 5.12: Sorties O_d avant et après application des règles de calcul.

3.3.2.3. Résultats

Dans le reste de nos tests, nous avons utilisé les trois premiers moments de H_u tandis que le nombre des moments de Zernike est compris entre 9 et 20 (dépend du nombre de données d'entrée et de sortie du réseau de neurones). Ces entrées sont présentées pour les différentes topologies de RNA arborées dans ce chapitre. Nous avons testé différentes tailles du vecteur d'entrée, le Tableau 5.13 présente les meilleures performances obtenues pour chacune des topologies des RNA.

| Topologie du RNA | Taux de reconnaissances | | | |
|--|------------------------------|------------|------------|------------|
| | (Num. Pictures/Num. Classes) | | | |
| | 1440/20 | 2880/40 | 4320/60 | 5760/80 |
| Réseau de neurones simple avec fonction de couleur | 99% | 94% | 87% | 40% |
| 3 R.N en parallèle pour chaque niveau de couleur | 100% | 99% | 90% | 45% |
| Un seul RNA pour les 3 Niveaux de couleurs | 99% | 98% | 92% | 82% |
| RNA avec couleur et forme détection | 99% | 98% | 95% | 89% |

Tableau 5.13: Taux de reconnaissances obtenus en appliquant H_u & Zernike comme vecteur d'entrée pour différentes topologies de RNA.

3.3.3. Troisième méthode (Intégration des descripteurs Géodésiques centralisés)

La procédure d'extraction proposée dans cette expérience est basée sur le calcul des distances géodésiques entre le centre de l'objet et une série de points situés dans le contour de l'objet.

3.3.3.1. Prétraitement

Un objet peut être facilement détecté dans une image s'il a un contraste suffisamment différent de celui de l'arrière-plan. Les changements dans le contraste peuvent être détectés à l'aide des opérateurs qui calculent le gradient d'une image. L'image de gradient peut être calculée et un seuil peut être appliqué pour créer un masque binaire contenant l'objet segmenté. D'abord, on utilise la détection des extrémités et ensuite l'opérateur de Sobel qui permet le calcul de la valeur de seuil.

3.3.3.2. Descripteurs géodésiques centralisés

Après la détection du centre de l'objet, nous localisons sur la frontière de la forme un nombre déterminé de points d'extrémités (Nous avons travaillé avec 40 points). La Figure 5.22 donne un exemple des distances géodésiques entre le centre et les points d'extrémités. Nous avons appelé descripteurs géodésiques centralisés le vecteur composé par les différentes distances géodésiques entre le centre et les points d'extrémités de l'objet (Tableau 5.14). Nous avons travaillé avec les mêmes algorithmes utilisés pour le calcul des distances géodésiques dans le chapitre 4.

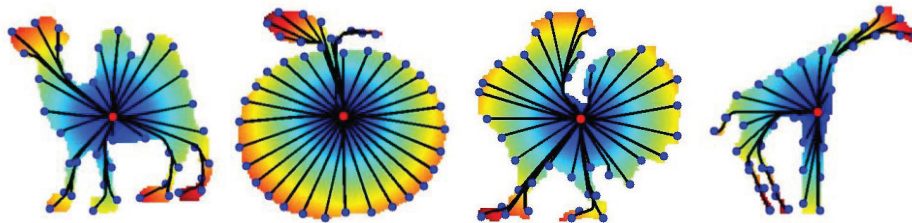


Figure 5.22: Exemples de distances géodésiques entre le centre de l'objet et ses extrémités

Le Tableau 5.14 donne un exemple des valeurs extraites calculées de deux images binaires



| | |
|---|--|
|  | 0.32 0.31 0.34 0.36 0.39 0.40 0.37 0.47 0.55 0.60 0.71 0.80 1 0.41 0.44 0.57 0.66 0.70 0.65 |
|  | 0.10 0.11 0.12 0.23 0.30 0.50 0.63 0.70 0.62 0.53 0.70 0.71 0.72 0.55 0.54 0.51 0.42 0.41 0.8 |

Tableau 5.14: Exemple de descripteurs Géodésiques centralisés

3.3.3.3. Système proposé et taux de reconnaissances

Le vecteur d'entrée du système est composé des moments de Zernike & Hu de chaque niveau de couleur et des descripteurs géodésiques sur l'image binaire de l'objet.

Nous avons utilisé les trois premiers moments de Hu et les neuf premiers moments de Zernike de chaque niveau de couleur ainsi que les descripteurs géodésiques centralisés et nous les avons mis ensemble pour former une entrée unique du réseau de neurones.

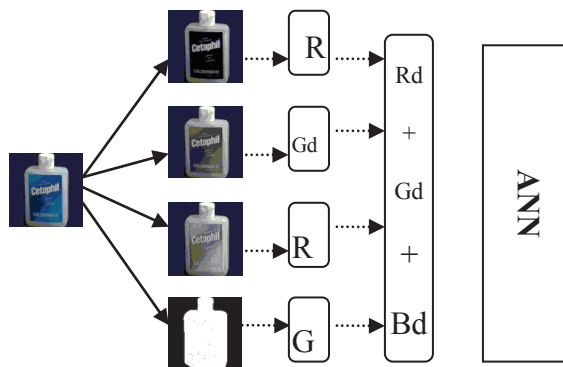


Figure. 5.23: Moments de Zernike et Hu et descripteurs géodésiques en série avec un RN entraîné par la rétropropagation du gradient.

Où:

- Bd: les moments de Zernike et Hu du niveau de gris bleu
- Rd: les moments de Zernike et Hu du niveau de gris rouge.
- Gd: les moments de Zernike et Hu du niveau de gris vert.
- Ge: les descripteurs géodésiques sur l'image binaire.

Nous avons testé différentes tailles des vecteurs d'entrées pour le processus d'apprentissage. Le Tableau 5.15 présente des résultats comparatifs en utilisant les moments de Zernike, Hu et leur combinaison avec des descripteurs géodésiques en tant que vecteurs d'entrée.

| Descripteurs de formes | Taux de reconnaissances (Nb. Images/Nb. Classes) | | | |
|------------------------|--|----------|----------|---------|
| | 1080/15 | 1440 /20 | 2880 /40 | 5760/80 |
| Les moments de Zernike | 99% | 91% | 85% | 63% |
| Les moments de Hu | 99% | 83% | 61% | 0% |
| Hu +Zernike | 99% | 98% | 93% | 83% |
| Le système proposé | 99% | 99% | 98% | 94% |

Tableau 5.15: Taux de reconnaissance obtenus en utilisant Hu, Zernike et leur combinaison avec les descripteurs géodésiques, (RNA avec 2 couches cachées).

Nous remarquons d'abord que l'approche de combinaison proposée permet d'obtenir les meilleurs résultats avec un gain de 11% en termes de taux de reconnaissances par rapport aux autres systèmes. Plus intéressant encore, les améliorations les plus importantes sont obtenues pour une base de données de moyenne taille (40 classes représenté par 2880 images) pour laquelle les taux de reconnaissances restent autour de 98%.

3.4. Influence des systèmes de couleur sur le rendement des réseaux de neurones [6]

Selon nos recherches, aucune étude ciblée sur l'influence de la base de couleur utilisée sur les taux de reconnaissances des différents classificateurs n'a été menée à ce jour. Dans ce travail, nous proposons une étude comparative entre les bases de couleurs adoptées par l'ICE (International Commission on Illumination) et leur influence sur les taux de reconnaissances d'un réseau de neurones. Nous proposons ainsi une nouvelle base de couleurs destinée aux systèmes de reconnaissances.

3.4.1. Espaces de couleurs

Un espace de couleurs est un modèle mathématique abstrait décrivant la façon dont les couleurs peuvent être représentées.

- L'Espace Colorimétrique Absolu:

- Le principe de l'ICE est de représenter mathématiquement un graphique en trois dimensions appelé l'espace CIE XYZ. Chaque couleur est alors représentée par trois coordonnées: XYZ (Z est la profondeur et la luminosité de chaque couleur).
- L'espace $L^*a^*b^*$ de couleur est un espace absolu comme CIE XYZ, mais représenté différemment. Les deux représentent toutes les couleurs, mais pas de la même manière. L pour le niveau de luminosité (de 0 à 100), 'a' est la couleur du rouge au vert et 'b' pour la couleur du bleu au jaune (- 128 à 127). Enfin, ces espaces sont neutres parce qu'ils ont été calculés manuellement.

-Espace de couleur calorimétrique

- Espace Lab et CIE-XYZ n'est pas assez pratique lorsqu'il s'agit de travailler avec des logiciels de traitement d'image. Par exemple, dans Photoshop, on utilise l'espace de

couleurs basé sur RVB (Rouge, Vert et Bleu). Les couleurs sont représentées par les valeurs RVB et non pas par les valeurs XYZ ou Lab ésotériques,

- HSV (teinte, saturation, la valeur ou luminosité) est couramment utilisé par les artistes, car il est souvent plus naturel de penser à une couleur en termes de teinte et de saturation qu'en termes de composantes de couleur additives ou soustractives.
- TSL (teinte, saturation, luminosité / luminance) est assez similaire au HSV, avec "légèreté" remplaçant "luminosité". La différence est que la luminosité d'une couleur pure est égale à la luminosité du blanc, tandis que la légèreté d'une couleur pure est égale à la légèreté d'un gris moyen.

3.4.2. Système de couleurs proposées

Le système de couleurs proposé est basé sur l'espace Lab. L'intégration de deux couleurs distinctes comme le rouge et le vert ne donne pas une représentation des couleurs exploitable à cent pour cent. Nous suggérons alors d'élargir l'espace Lab sur cinq dimensions L * R * V * B * J : Luminosité, Rouge, Vert, Bleu et Jaune. De cette manière, chaque couleur sera représenté par sa propre matrice.

La transformation de l'espace absolu CIE XYZ vers le Lab est donnée par:

$$L^* = 116f(Y/Y_n) - 16 \quad (5.2)$$

$$a^* = 500[f(X/X_n) - f(Y/Y_n)] \quad (5.3)$$

$$b^* = 500[f(Y/Y_n) - f(X/X_n)] \quad (5.4)$$

Avec :

$$f(t) = t^{1/3} \quad \text{si } t > \left(\frac{6}{29}\right)^3$$

$$f(t) = \frac{1}{3}\left(\frac{29}{6}\right)^2 t + \frac{4}{29} \quad (5.55)$$

Ici X_n, Y_n et Z_n sont les valeurs CIE XYZ du point blanc de référence (l'indice n indique «normalisée»).

La division de la fonction f(t) dans deux domaines a été faite pour empêcher une pente infinie à t = 0. F (t) est supposée linéaire ci-dessous pour t = t₀, et supposé correspondre à la t^(1/3) partie de la fonction à t₀ en valeur et en pente. En d'autres termes:

$$t_0^{1/3} = at_0 + b \quad (5.6)$$

Donc :
$$\frac{1}{3}t_0^{-2/3} = a \quad (5.7)$$

La pente a été choisie pour être $b = 16 / 116 = 4 / 29$. Les deux équations ci-dessus peuvent être résolues pour a et t_0 : $a = \frac{1}{3}\delta^{-2} = 7.787037\dots$

$$t_0 = \delta^3 = 0.008856\dots \quad \text{Avec } \delta = 6/29.$$

Une fois que les composantes Lab sont calculées, la transformation vers le Lab-étendu est donnée par l'algorithme suivant :

```
L,a,b : les matrices de couleurs de l'espace Lab
L',R ,G,B,Y :les Matrices couleurs de l'espace Lab-
étendu(Luminance, Rouge, Vert, Bleu, Jaune.

Pour i=1 à n
Pour j=1 à m
L'[i,j]=L[i,j];
si (a[i,j]<=127) alors      R[i,j]=a[i,j];
                           V[i,j]=0;
Si non                     V[i,j]=a[i,j];
                           R[i,j]=0;
Fin si
si (b[i,j]<=127) alors      B[i,j]=b[i,j];
                           J[i,j]=0;
Si non                     J[i,j]=b[i,j];
                           B[i,j]=0;
Fin si
```

3.4.3. Rendement des espaces de couleurs

Nous avons calculé les 9 premiers moments de Zernike pour chaque niveau de couleurs et nous les avons intégrés en série (Figure 5.15) pour former une entrée unique du réseau de neurones. Le tableau suivant présente les résultats obtenus pour les bases de couleurs testées.

| Espaces de couleurs Taille de la base (Nb class/Nb picotures) | RVB | LAB | HSV | HSB | LAB- étendu |
|---|-------|-------|-----|-----|----------------|
| 1440/20 | 99% | 99% | 63% | 64% | 99% |
| 2880/40 | 96.2% | 98.1% | 0% | 0% | 99% |
| 4320/60 | 94% | 93% | 0% | 0% | 97.3% |
| Temps d'apprentissage pour 60 classes | 2h'24 | 2h'27 | ** | ** | 3h'13 |

Tableau 5.16 : Taux de reconnaissances obtenus par application des RNA et des moments de Zernike sur les différents espaces de couleurs

On peut conclure :

- Que l'espace RVB aboutit à de meilleurs résultats. Nous pouvons dire la même chose pour l'espace Lab. Le système proposé (Lab-étendu) a permis d'améliorer les résultats obtenus de 4% ; et
- Que la structure des espaces HSV et HSB n'a pas permis une extraction efficace des données couleur.

4. Conclusion

Nous avons divisé ce chapitre en deux parties, la première est dédiée à la reconnaissance des caractères et la deuxième à la reconnaissance d'objets en couleur.

Concernant la reconnaissance des caractères, nous avons proposé trois méthodes de reconnaissances :

- Les descripteurs métriques avec un nombre de caractéristiques réduit (seulement 5) : Ces descripteurs permettent un traitement de grande vitesse. Sur un nombre réduit de caractères (15), les descripteurs métriques ont donné de très bons résultats.
- Les descripteurs géodésiques combinés avec les RNA et les arbres de décisions présentent un système de reconnaissance robuste. Malgré la taille de la base d'apprentissage (15 échantillons par caractère), nous avons retrouvé un taux de reconnaissance de 94%.

- Les descripteurs de courbures combinés avec RNA et les RNF présentent les meilleurs taux de reconnaissances. Cependant, les temps de reconnaissance ont tendance à être plus longs.

En ce qui concerne la reconnaissance des objets réels, nous avons agi sur trois paramètres pour améliorer les taux de reconnaissances :

- Au niveau du prétraitement, nous avons établi une étude comparative entre les modèles de couleurs et leur influence sur le rendement des systèmes de reconnaissances basés sur les RNA et les DF invariants (Hu et Zernike). L'étude confirme l'influence positive des modèles RVB et Lab sur les taux de reconnaissances. Dans ce cadre, on a proposé un modèle de couleurs Lab-étendu qui a permis une amélioration de 4% des taux de reconnaissances par rapport au RVB.
- Au niveau de la phase d'extraction et après une série de tests sur plusieurs types de descripteurs, nous avons préconisé les moments de Hu et de Zernike comme entrées des classificateurs intégrés.
- En ce qui concerne la phase de classification, nous avons comparé plusieurs topologies de RNA. Nous avons déduit que l'usage des DF et des RNA en parallèle est favorable dans le cas des petites bases de données, alors que l'usage des DF et des RNA en série permet le traitement des données de grande taille. Les RNA multis sortis que nous avons proposés on permet la résolution des problèmes de confusion dus au grand nombre de neurones de la de sortie. Finalement, L'intégration du RNA avec les descripteurs géodésiques de forme et l'exploitation des niveaux couleur a permis une amélioration des taux de reconnaissances de 3%.

Conclusion générale

En guise de conclusion, cette thèse s'est concentrée sur la reconnaissance des caractères et des objets réels. S'agissant de présenter des algorithmes ou plus précisément des systèmes de reconnaissances.

Les dits systèmes permettent de prendre en charge une image en entrée, pour donner comme résultat la catégorie de l'objet contenu dans celle-ci en sortie. Le fait expliquant le défi de ce travail réside dans le chevauchement des apparences multiples qu'un seul objet peut prendre.

Pour cette raison, les systèmes de reconnaissance que nous avons développés se basent principalement sur des classificateurs robustes comme les réseaux de neurones, les réseaux Neuro-flou, les SVM et les arbres de décisions.

Nous sommes arrivés à développer de nouveaux descripteurs basés sur la géométrie riemannienne.

Aussi, nous avons établi une étude comparative entre les méthodes de classification et d'extraction couramment utilisées avant d'entamer enfin l'étude d'influence des paramètres primaires sur le rendement des réseaux de neurones à savoir : le changement de la base de couleur, la luminance, l'existence de bruit et des déformations spatiales.

L'apport de notre travail peut être résumé comme suit :

Au niveau de la reconnaissance d'objets réels.

Nous avons travaillé avec une approche 3D/2D basée sur les vues caractéristiques. Au niveau de la phase d'extraction nous avons testé une multitude de descripteurs (Zernike, Hu, Fourier, Radon). Après une série de testes nous avons choisi la combinaison des moments de Zernike et de Hu comme meilleur vecteur représentatif.

En ce qui concerne la phase de classification, nous avons profité des informations supplémentaires présentées par les niveaux de couleurs pour augmenter le nombre de paramètres significatifs. Ensuite, nous avons testé des RNA en séries, en parallèle, et on a proposé un RNA multi-sorties qui permet la reconnaissance de la couleur, de la forme globale et de l'objet lui-même. Le réseau proposé a donné de très bons résultats (95% pour 4800 images séparées sur 60 classes).

Une méthode locale a été proposée pour gagner en robustesse, elle considère seulement la partie de l'image qui contient l'objet, elle se base sur le calcul des distances géodésiques entre le centre de l'objet et son contour. Les résultats obtenus semblent prometteurs.

Au niveau de la reconnaissance des caractères

L'un des objectifs de cette thèse est de réaliser un OCR tout aussi fiable que rapide. Pour cela, l'objectif était de développer des descripteurs suffisamment informatifs, mais peu coûteux en temps de calcul. Tout d'abord, nous avons implémenté ce qu'on a appelé descripteurs métriques. Ce sont des descripteurs qui demandent un temps de calcul minimal avec un nombre de paramètres réduits (5 paramètres). Ils ont donné de très bons résultats lors de leur intégration pour la reconnaissance des chiffres manuscrits. Mais le nombre de ces paramètres réduit n'a pas permis une distinction efficace entre un nombre de classes qui dépasse 15. Nous l'avons testé avec les SVM et les RNA pour la reconnaissance des caractères Tifinagh, mais il n'a donné que 78% comme taux de reconnaissance. Ensuite on a présenté un système de qualité basé sur les descripteurs géodésiques et la combinaison des RNA et des arbres de décisions. Ce système rapide a permis un taux de reconnaissance de 94%. Enfin, on a proposé un système composé des descripteurs de courbures et des RNA combinés avec les réseaux neuroflous. C'est un système gourmand au niveau du temps de calcul que le dernier, mais qui a permis un taux de reconnaissance de 95,5%.

Perspectives

Bien que les résultats expérimentaux, ont montré l'efficacité des systèmes proposés, le temps d'exécution assez lourd peut être amélioré on jouant sur plusieurs axes :

- L'optimisation des classificateurs utilisés et l'exploitation d'autres classificateurs.
- L'amélioration des descripteurs proposés.
- Refaire les mêmes expériences avec un autre langage de programmation plus rapide tel que JAVA

Aussi, nous envisageons L'intégration des systèmes de reconnaissances proposés dans un processus de notation des images sur le web (avec arborescence).

Finalement, nous avons déjà commencé à rassembler un ToolBox Extraction qui rassemble l'ensemble des descripteurs de formes existant (Zernike, Hu, Hough, Legendre, Radon, Fourier, Gist, SURF, ACP, etc.)

Références

- [1] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern classification*, Wiley-interscience, 2001 (ISBN 0-471-05669-3)
- [2] I. Biederman. *An Invitation to Cognitive Science, Vol. 2: Visual Cognition*. MIT press 1995. 17.
- [3] M.Fakir,O,Bencharef,B.Bouikhalene,B.Minaoui , Tifinagh Character Recognition Using Riemannian Metric, SVM & Neural Networks, *International Journal of Advances in Science and Technology*, Vol. 2, No.6, 2011,pp 1-9.
- [4] Omar Bencharef , Mohamed Fakir, Brahim Minaoui, Belaid Bouikhalene ‘Tifinagh Character Recognition Using Geodesic Distances, Decision Trees & Neural Networks’ (IJACSA) *International Journal of Advanced Computer Science and Applications*,Special Issue on Artificial Intelligence, june 2011, pp: 51,55.
- [5] Omar Bencharef , Mohamed Fakir, Brahim Minaoui, Belaid Bouikhalene “A Geometric Approach for Tifinagh Character Recognition Using Fuzzy NN & ANN, *International Journal of Computer Information Systems*, Vol. 3, No. 4, 2011. ISSN 2229-5208. pp:26-31.
- [6] O.Bencharef et. al. “3D Color Objects Recognition System Using an Artificial Neural Network,”/ *International Journal of Engineering Science and Technology*. 7.2011
- [7] Omar Bencharef , Mohamed Fakir, Brahim Minaoui, Belaid Bouikhalene ,” The influence of color models on the object recognition process “ISI.conference 1-5 juin 2011.
- [8] Omar Bencharef, Mohamed Fakir 1, Brahim Minaoui1, Belaid Bouikhalen ” Application de la géométrie riemannienne à la reconnaissance des caractères Tifinagh ”, SITACAM’11, Agadir-Maroc, 06-07 Mai 2011.
- [9] N. Ben Amara, "Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée", Thèse de doctorat, Université des sciences, des Techniques et de médecine de Tunis II, 1999.
- [10] L. Di Stefano, A. Bulgarelli, “A simple and efficient connected components labeling algorithm”. *Image Analysis and Processing Conference*, 27-29 Sept. 1999 pp : 322 – 327.
- [11] F.L. Alt, , *J. ACM* 9(2) (1962) 240–258. [1] F.L. Alt, “Digital pattern recognition by moments,” *J. ACM* February 1962 , pp. 240–258.
- [12] Jan Flusser, *Moment Invariants in Image Analysis* ,*Transaction on engineering, computing and technology* February ,2006 ,pp:1305-5313.
- [13] L.D. Cohen. *Multiple Contour Finding and Perceptual Grouping Using Minimal Paths*. *Journal of Mathematical Imaging and Vision*, 14(3), 2001 . Presented at VLISM01.

- [14] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- [15] Quoc Le, Morgan Quigley and Andrew Y. Ng. *Visual Servoing by Template matching*. (Unpub-lished)
- [16] Beis, J., and Lowe, D.G “Shape indexing using approximate nearest-neighbour search in high-dimensional spaces”, *Conference on Computer Vision and Pattern Recognition*, Puerto Rico, 1997, pp. 1000–1006.
- [17] Lowe, D.G., *Local feature view clustering for 3D object recognition*. *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001, pp. 682-688.
- [18] Bayes, T. *An essay towards solving a problem in the doctrine of chances*. (1763).
- [19] LI LiLi, ZHANG YanXia and ZHAO YongHeng, “K-Nearest Neighbors for automated classification of celestial objects,” *Science in China Series G-Phys Mech Astron*, Vol.51, no.7, July 2008, pp: 916-922.
- [20] Marc Parizeau « RESEAUX DE NEURONES » GIF-21140 et GIF-64326 .Université de LAVAL 2006.
- [21] S. Macé, E. Anquetil, "Eager interpretation of on-line hand-drawn structured documents: The DALI methodology", *Pattern Recognition*, vol. 42, , 2009, pp: 3202-3214.
- [22] Rachid EL Ayachi, Mohamed Fakir and Belaid Bouikhalene, "Recognition of Tifinaghe Characters Using a Multilayer Neural Network", *International Journal Of Image Processing (IJIP)*, V. 5, Issue 2, 2011.pp: 109-118.
- [23] G. Saon, "Modèles markoviens uni- et bidimensionnels pour la reconnaissance de l'écriture manuscrite hors-ligne", *Thèse de Doctorat*, Université Henri Poincaré - Nancy1, 1997.
- [24] Y. Ait Ouguengay, M. Taalabi, "Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe: Phase d'apprentissage", *Systèmes intelligents-Théories et applications*, Paris: Europia, cop. 2009 (impr. au Maroc), ISBN-102909-285553, 2009.
- [25] Institut royal de la culture amazighe, "Proposition de codification du tifinaghe dans Unicode", *Centre de l'aménagement linguistique*, Rabat, avril 2004.
- [26] D. Mammass, "Approches prétopologique et multi-échelles appliquées à l'extraction de l'écriture du fond de chèques bancaires, à la détection de contours et à la compression d'images", *Thèse de Doctorat d'Etat*, Univ Ibn Zohr, 1999.
- [27] Y. Es Saady, B. Bakkass, A. Rachidi, M. El Yassa, D. Mammass : "Développement d'un Éditeur de Texte Amazighe: État d'avancement et Perspectives", 3ème atelier international sur le thème "Les Technologies de l'information: statuts et opportunités pour l'amazighe", IRCAM, Rabat– Maroc, 24-25 novembre 2008.

- [28] Y. Es Saady, Y. Ait Ouguengay, A. Rachidi, M. Elyassa, D. Mammass, "Adaptation d'un correcteur orthographique existant à la langue Amazighe: cas du correcteur Hunspell", Actes du 1er symposium international sur le traitement automatique de la culture amazighe, SITACAM, Agadir, 2009, pp :149-158.
- [29] Y. Es Saady, A. Rachidi, M. El Yassa, D. Mammass, "Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata", ICGST-GVIP Journal, Vol.10, Issue 2, 2010, pp.1-8.
- [30] M. Fakir, B. Bouikhalene, K. Moro, "Skeletonization Methods Evaluation for the Recognition of Printed Tifinaghe characters", SITACAM'09, Agadir 2009.
- [31] Serre, T., Kouh, M., Cadieu, C., Knoblich, U., Kreiman, G., Poggio, T., "A Theory of Object Recognition: Computations and Circuits in the Feedforward Path of the Ventral Stream in Primate Visual Cortex", Computer Science and Artificial Intelligence Laboratory Technical Report, December 19, 2005 MIT-CSAIL-TR-2005-082.
- [32] Beis, J., and Lowe, D.G "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces", Conference on Computer Vision and Pattern Recognition, Puerto Rico, 1997, pp. 1000–1006.
- [33] Lowe, D.G., Local feature view clustering for 3D object recognition. IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, 2001, pp. 682-688.
- [34] C.MAAOUI " Reconnaissance et détection robuste d'objets couleur " 20th colloque GRETSI, 2005, pp : 727-730.
- [35] Seethalakshmi et al." Optical Character Recognition for printed Tamil text using Unicode" / J Zhejiang Univ SCI 2005 6A(11), pp:1297-1305.
- [36] Hamami, L. and D. Berkani, 2002. Recognition System for Printed Multi-Font and Multi-Size Arabic Characters. Arabian J. Sci. Eng., 27(1B) pp: 57-72.
- [37] H Murase and S Nayar. Visual learning and recognition of 3d objects from appearance. IJCV, 14(1):5–24, 1995. pp: 25- 27.
- [38] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In CVPR, 2007. 31. pp 1–8.
- [39] D.P. Huttenlocher and S Ullman. Recognizing solid objects by alignment with an image. IJCV, 5(2):195–212, 1990. 24, 25 [46] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. Object recognition by affine invariant matching. In CVPR,1988, pp 335–344.
- [40] W.E.L. Grimson and T. Lozano-Pérez. Localizing overlapping parts by searching the interpretation tree. PAMI, 1987, 9(4) pp: 469–482.

- [41] M.A. Fischler and R.C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. ACM, 24(6) pp: 381–395, 1981.
- [42] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. Object recognition by affine invariant matching. In CVPR, 1988, pp 335–344.
- [43] M. Turk and A. Pentland. Face recognition using eigenfaces. In CVPR, 1991, pp: 586–591.
- [44] E.C. Yiu: "Image classification using color cues and texture orientation", Master thesis, MIT, US of America, 1996.
- [45] H.-K. Kim, J.-D. Kim, D.-G. Sim, and D.-I. Oh, "A modified Zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval," in Proc. IEEE Int. Conf. Multimed. Expo, Jul. 2000, vol. 1, pp. 307–310.
- [46] M. Turk and A. Pentland. Face recognition using eigenfaces. In CVPR, 1991, pp 586–591.
- [47] H.J. Wolfson and I Rigoutsos. Geometric hashing: An overview. Computational Science and Engineering, 4(4) 1997, pp: 10–21.
- [48] D. Lowe. Object recognition from local scale-invariant features. In CVPR,1999, pp: 1150–1157.
- [49] Breslin, A.J., Glauberman, H. "The Generation of A Solid Sub-Micron Test Aerosol" American Industrial Hygiene Association Quarterly, Vol. 17, December 1956, pp. 401 - 404.
- [50] D. Lowe. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 2004, pp: 91–110.
- [51] Jan Flusser, Moment Invariants in Image Analysis ,Transaction on engineering, computing and technology February ,2006 , pp: 1305-5313.
- [52] F.L. Alt, "Digital pattern recognition by moments," J. ACM February 1962 , pp: 240–258.
- [53] H.-K. Kim, J.-D. Kim, D.-G. Sim, and D.-I. Oh, "A modified Zernike moment shape descriptor invariant to translation, rotation and scale for similarity-based image retrieval," in Proc. IEEE Int. Conf. Multimed. Expo, Jul. 2000, vol. 1, pp. 307–310.
- [54] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," IEEE Trans. Pattern Anal. Mach. intell, vol. 12, no. 5, May 1990,pp. 489–497.
- [55] F.Zernike. Beugungstheorie des Schneidenverfahrens und seiner verbesserten Form, der Phasenkontrastmethode (Diffraction theory of the cut procedure and its improved form, the phase contrast method). *Physica*, 1, 1934, pp: 689-704.
- [56] D. Zhang and G. Lu. Shape-based image retrieval using generic Fourier descriptor, Signal Processing : Image Communication, 17, 2002, pp: 825–848.

- [57] Chen G.Y., Bui T.D., Krzyzak A., « Invariant pattern recognition using Radon, dual-tree complex wavelet and Fourier transforms», *Pattern Recognition*, vol. 42, n° 9, 2009, pp: 2013-2019.
- [58] MPEG-7 Visual Experimentation Model (XM), Version 10.0, ISO/IEC/JTC1 /SC29 /WG11, Doc. N4063, Mar. 2001.
- [59] D. Comaniciu and P. Meer, "Robust analysis of featurespaces: color image segmentation", *Proceedings of Computer Vision and Pattern Recognition*, 1997, pp: 750-755.
- [60] A. Djematen, B. Taconet., A. Zahour, « Une méthode statistique pour la reconnaissance de caractères berbères manuscrits », *Actes de CIFED'98*, pp: 170-178.
- [61] M. Fakir, B. Bouikhalene et K. Moro, « Skeletonization Methods Evaluation for the Recognition of Printed Tifinaghe characters », *Symposium SITACAM 2009 Agadir 12-13 Décembre Maroc*.
- [62] B. El Kessab, C. Daoui, B. Bouikhalene et M. Fakir, « Utilisation des réseaux de neurones et le modèle de Markov pour la reconnaissance des caractères Tifinagh manuscrits », *Agadir-Maroc, 06-07 Mai 2011*. pp : 31-40.
- [63] B. Bazdouz, B. Bouikhalene, M. Fakir et R. El Ayachi, «Reconnaissance Automatiques d'un Document Amazighs par les Modèles de Markov cachés », *4ème atelier international sur l'amazighe et les TIC, 2011, IRCAM, Maroc*.
- [64] M. Amrouch, Y. Es Saady, A. Rachidi, M. Elyassa et D. Mammass, « Printed Amazigh Character Recognition by a Hybrid Approach Based on Hidden Markov Models and The Hough Transform », *ICMCS'09, Ouarzazate-Maroc, 2009*.
- [65] R. El Ayachi, K. Moro, M. Fakir et B. Bouikhalene, « On The Recognition Of Tifinaghe Scripts ». *JATIT*, vol. 20, No. 2, 2010, pp: 61-66.
- [66] S. GOUNANE, M. FAKIR, B. BOUIKHALENE, « Recognition of TIFINAGHE Characters using Self Organizing Map and Fuzzy k-Nearest Neighbor », *Global Journal of Computer Science and Technology*, Vol 11, No 15 (2011), pp:29-33
- [67] Y. Es Saady, A. Rachidi, M. El Yassa et D. Mammass, « Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character », *International Journal of Advanced Science and Technology* Vol. 33, August, 2011, pp.33-50.
- [68] A. Rosenfeld, "The fuzzy geometry of image subsets", *Pattern Recognition Letters*, 2,1984, pp: 311-317.
- [69] M. Ziaratban, K. Faez, F. Bagheri, "FHT: An Unconstraint Farsi Handwritten Text Database", *ICDAR'09*, pp: 281-285.
- [70] H. Zouari, "Contribution à l'évaluation des méthodes de combinaison parallèle de classifieurs par simulation", *Thèse de Doctorat, Université de Rouen, 2004*.
- [71] Ripley B. D., *Pattern Recognition and Neural Networks*, Cambridge, Cambridge University Press, 1996.

- [72] Riviere D., Mangin J., Papadopoulos D., Martinez J., Frouin V. et Regis J., Automatic Recognition of Cortical Sulci of the Human Brain Using a Congregation of Neural Networks, *Medical Image Analysis*, 6, 2, Jun 2002, pp: 77-92.
- [73] O. Lezoray. Segmentation d'images par morphologie mathématique et classification de données par réseaux de neurones : Application à la classification de cellules en cytologie des séreuses. Thèse de doctorat, Université de Caen (2000).
- [74] D. Rumelhart, G. Hinton & R. Williams, "Learning internal representations by errorpropagation," *Parallel Distributed Processing*, Vol. 1., D. Rumelhart and J. McClelland Eds. Cambridge: MIT Press, 1986 , pp:318-362.
- [75] DARPA, DARPA Neural Network Study, Chap. 8. AFCEA International Press, 1988. M. Minsky and S. Papert, *Perceptrons*, Expanded Edition. MIT Press, 1988.
- [76] Le Cerveau, Bibliothèque de la revue Pour la Science, ISBN 2-902918-24-7, 1982. R. Masland, "L'architecture fonctionnelle de la rétine," revue Pour la Science, février 1987 pp :94-104.
- [77] J. P. Changeux, *L'homme neuronal*, Collection Pluriel, Fayard, ISBN 2-01-009635-5, 1983.
- [78] SIMARD P., STEINKRAUS D., PLATT J. C.(2005), Best Practices for Onvolutional Neural Networks Applied to Visual Document Analysis, *ICDAR*, pp: 958-962
- [79] S. Haykin, "Neural Networks: a Comprehensive Foundation", Prentice-Hall, New Jersey, USA, 1999.
- [80] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *PAMI*, 19(5) pp:530–534, 1997. 26.
- [81] D.G. Lowe. The viewpoint consistency constraint. *IJCV*, 1(1): pp57–72, 1987. pp: 20-25.
- [82] A. Rosenfeld, "The fuzzy geometry of image subsets", *Pattern Recognition Letters*, 1984, 2:3 pp:11–317.
- [83] L. Zadeh "Fuzzy sets and systems". In: Fox J, editor. *System Theory*. Brooklyn, NY: Polytechnic Press, 1965, pp: 29–39.
- [84] "IEEE Computer Society Magazine Honors Artificial Intelligence Leaders". *DigitalJournal.com*. August 24, 2011 (2011-08-24). <http://www.digitaljournal.com/pr/399442>. Retrieved September 18, 2011 (2011-09-18). Press release source: PRWeb.
- [85] Zhou, R. W., & Quek, C. (1996). "POPFNN: A Pseudo Outer-product Based Fuzzy Neural Network". *Neural Networks*, 9(9), 1569-1581.
- [86] J.-S.R. Jang, ANFIS: Adaptive-Network-Based Fuzzy Inference Systems, *IEEE Trans. Systems, Man & Cybernetics* 23 (1993), pp: 665—685.

- [87] S. K. Halgamuge and M. Glesner, Neural networks in designing fuzzy systems for real world applications, *Fuzzy Sets and Systems* 65 (1994) pp: 1-12.
- [88] N. Tschichold-Gürman, RuleNet “ A New Knowledge--Based Artificial Neural Network Model with Application Examples in Robotics”, PhD thesis (ETH Zürich, 1996).
- [89] V.Vapnik, “Support-Vector Network”, *Machine Learning*, vol. 20, issue 3, September 1995, pp: 273-397.
- [90] Christopher J.C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, *Data Mining and Knowledge Discovery*, 2,1998, 121–167.
- [91] S. Gallot, D. Hulin, et J. Lafontaine. *Riemannian Geometry*. Universitext. Springer Verlag, New York, 1990.
- [92] Kimmel, R. & Sethian, J. A. (1998). Computing geodesic paths on manifolds, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 95, No. 15, pp: 8431-8435
- [93] John M. Lee, *Introduction to smooth manifolds*, *Graduate Texts in Mathematics*, vol. 218, Springer-Verlag, New York, 2003. MR1930091 (2003k :58001).
- [94] X.Gu ,(2004) Genus Zero Surface conformal apping,IEEE TANSCTIONS ON MEDICAL IMAGING,VOL.23 NO.8.
- [95] A. Fuster, L. Astola and L. Florack, A Riemannian Scalar Measure for Diffusion Tensor Images, *Lecture notes in Computer Science*, 5702 (2009), pp. 419–426.
- [96] J.KERL (2008), Numerical differential geometry in Matlab, Graduate student Colloquium, university of Arizona.
- [97] J.A. Sethian et R. Kimmel. Computing Geodesic Paths on Manifolds. *Proc. Natl. Acad. Sci.*, 95(15), 1998, pp:8431–8435.
- [98] X. Pennec, P. Fillard and N. Ayache, A Riemannian Framework for Tensor Computing, *Int. J. Computer Vision*, 66(1) (2006), pp: 41–66.
- [99] Kimmel, R. & Sethian, J. A. (1998). Computing geodesic paths on manifolds, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 95, No. 15, pp:8431-8435
- [100] J.M. Lee, *Riemannian manifolds: an introduction to curvature*, Springer Verlag, 1997.
- [101] Gavin C. Cawley, Gareth J. Janacek, and Nicola L. C. Talbot. Generalised kernel machines. In *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks (IJCNN-2007)*, pages 1732-1737, Orlando, Florida, USA, August 12-17 2007.

[102] Univ of Columbia ,USA DATA BASE (Coil-1000 : <http://www1.cs.columbia.edu/CAVE/research/softlib/coil-100.html>).

[103] David Legland, david.legland@grignon.inra.fr, Copyright INRA - Cepia Software Platform.