Université Sidi Mohammed Ben Abdellah
Faculté des Sciences Dhar El Mahraz-
Centre d'Études Doctorales
"Sciences et Technologies"

Formation Doctorale : STIC

Spécialité : Informatique

Laboratoire : LIIAN

THÉSE DE DOCTORAT

Présentée par

AHARRANE Nabil

Contributions à la reconnaissance automatique hors ligne de l'écriture Amazighe imprimée et manuscrite en Tifinagh

Soutenue le 06/12 /2018 devant le jury composé de :

Pr. Hamid Tairi	Faculté des Sciences Dhar Mahraz – Fès	Président
Pr. Khalid SATORI	Faculté des Sciences Dhar Mahraz – Fès	Directeur de thèse
Pr. Samir MBARKI	Faculté des sciences, Université Ibn Tofail – Kenitra	Rapporteur
Pr. Hassan SATORI	Faculté des Sciences Dhar Mahraz – Fès	Rapporteur
Pr. Brahim AKSASSE	Faculté des Sciences et Techniques – Er-Rachidia	Rapporteur
Pr. Chakir LOQMAN	Faculté des Sciences Dhar Mahraz – Fès	Membre
Pr. Abderrahim SAAIDI	Faculté polydisciplinaire – Taza	Membre
Pr. Akram HALLI	Faculté des sciences juridiques économiques et sociales – Meknès	Membre
Pr. Karim EL MOUTAOUAKIL	école Nationale des sciences Appliquées – Al-Hoceima	Co-directeur de thèse

Année universitaire : 2017-2018

Résumé

Cette thèse se focalise sur le sujet de la reconnaissance automatique hors ligne de l'écriture Amazighe imprimée et manuscrite à partir des images. Dans le cadre de l'écriture manuscrite, un système de reconnaissance automatique a été conçu en proposant un nouveau descripteur basé sur des caractéristiques statistiques. Ce descripteur se base sur la décomposition de l'image du caractère isolé en plusieurs zones chevauchées qui prennent en compte la structure et la forme des caractères, ensuite, pour chaque zone, des caractéristiques relatives aux longueurs des traits et de densité sont extraites. Ce système a été testé sur une base d'images de références pour les caractères Amazighs manuscrits appelée AMHCD et les résultats obtenus ont été très satisfaisants. Pour l'écriture imprimée, une base d'images de référence APWID pour les mots Amazighs imprimés multi-polices, multi-tailles et multi-styles a été proposer pour l'évaluation et la comparaison des systèmes de reconnaissance (OCR) des caractères Amazighs imprimés. La base APWID a servi pour l'évaluation d'un système OCR multi-polices et multi-styles des caractères amazighs pour répondre à la variété des styles utilisés pour les documents Amazighs imprimés mais aussi pour reconnaitre le texte diffusé dans les images Web ou de scènes naturelles. Pour ce genre d'image le texte en Tifinagh peut exister dans un environnement multilingue et dans des arrière-plans complexes. Pour ceci, nous avons proposé un système de bout en bout capable de localiser le texte dans l'image, identifier son script et procéder à la reconnaissance de celui en Tifinagh. La phase d'identification du script s'appuie sur un réseau de neurones convolutifs qui a montré sa performance face aux trois scripts utilisés au Maroc (arabe, latin, Tifinagh).

Mots clés : Reconnaissance optique des caractères, Langue Amazighe, Alphabet Tifinagh, vision par ordinateur, Extraction des caractéristiques, apprentissage automatique, Système OCR, Écriture imprimée, Écriture manuscrite.

Abstract

This thesis focuses on the automatic off-line recognition of printed and handwritten Amazigh writing from images. As part of handwriting, an automatic recognition system was developed by proposing a new descriptor based on statistical characteristics. This descriptor is based on the decomposition of the image of the isolated character into several overlapping zones that take into consideration the structure and shape of the characters, then, for each zone, characteristics relating to stroke lengths and density are extracted. This system has been tested on a benchmark database of handwritten Amazigh characters called AMHCD and the results obtained have been very satisfactory. For printed writing, we provide a new Amazigh Printed Word Images database (APWID) for a wide-scale benchmarking of Amazigh character recognition systems (OCR). The proposed database is multifonts, multisizes and multi styles ensuring a large data variability. This database served to train and test the multi-font and multi-style proposed printed Amazigh OCR system. This system was created to respond to the variety of styles used for printed Amazigh documents but also to recognize the broadcast text in Web images or natural scenes. For this kind of image Tifinagh text can exist in a multilingual environment and in complex backgrounds. For this, we proposed an end-to-end system capable of locating the text in the image, identifying its script and proceeding to the recognition of the one in Tifinagh. The script identification phase is based on a convolutional neural network that has shown its performance against the three scripts used in Morocco (Arabic, Latin and Tifinagh).

Key words : Optical character recognition, Amazigh language, Tifinagh script, Computer vision, Features extraction, apprentissage automatique, OCR System, Printed writtiing, Handwritting.

Remerciements

Le travail présenté dans ce mémoire de thèse a été réalisé, au sein du Laboratoire d'Informatique, Imagerie et Analyse Numérique LIIAN, sous la direction scientifique de M. Khalid SATORI, professeur à La Faculté des Sciences Dhar El Mahraz de Fès, à qui je témoigne toute ma gratitude pour sa précieuse aide et pour le soutien qu'il m'a accordé.

J'adresse mes profonds remerciements à M. Brahim AKSASSE, professeur à la faculté des sciences et techniques d'Errachidia, M. Samir MBARKI, professeur à la faculté des sciences au Kénitra, et à M. Hassan SATORI, professeur à la faculté des sciences Dhar Mahraz à Fès, qui m'ont fait l'honneur d'être rapporteurs de ma thèse. Leurs commentaires et leurs questions m'ont notamment permis d'améliorer ce manuscrit.

J'exprime ma profonde gratitude à M. Hamid TAIRI, professeur à la faculté des sciences Dhar Mahraz à Fès, pour l'honneur qu'il m'a fait en présidant mon jury de thèse. Mes remerciements vont aussi à l'égard de M. Abderrahim SAAIDI, professeur à la faculté polydisciplinaire de Taza, et M. Akram HALLI, professeur à la faculté des sciences juridiques économiques et sociales à Meknès, et M. Chakir LOQMAN, professeur à la faculté des sciences Dhar Mahraz à Fès, pour avoir accepté de participer à mon jury de thèse.

Mes remerciements vont également à M. Karim EL MOUTAOUAKIL, professeur à l'École Nationale des Sciences Appliquées à Al-Houcima, pour tout son aide et soutien fourni tout au long de mon parcours doctoral.

Un grand merci pour ma famille pour leur soutien et leur encouragement au cours de ces années des études doctorales et à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicace

À ma Mère

Liste des figures

Liste des figures

Liste des tables

Liste des abréviations

AMHCD AMazigh Handwritten Characters Database

ANFIS Artificial Neural Network Fuzzy Inference System

APWID Amazigh Printed Words Images Database

AUC Area Under the Curve

CART Classification And Regression Trees

CNN Convolutional Neural Network

DCCI Directional Cubic Convolution Interpolation

DCT Discrete Cosine Transform

EGII Edge Guided Image Interpolation

EM Expectation Maximization

GMM Gaussian Mixture Model

HMM Hidden Markov Models

HOG Histogram of Oriented Gradients

ICBI Iterative Curve Based Interpolation

ICDAR International Conference on Document Analysis and Recognition

ID3 Iterative Dichotomizer 3

IRCAM Institut Royal de la Culture AMazighe

IWFHR International Workshop on Frontiers in Handwritten Recognition

KNN K-Nearest Neighbours

MSER Maximally Stable Extremal Regions

MRF Markov Random Fields

OCR Optical Character Recognition

PMC Perceptron MultiCouches

RB Réseau Bayésien

RBF Radial Basis Function

RF Random Forest

ROC Receiver Operating Characteristic curve

SURF Speed Up Robust Features

SVM Support Vector Machine

SWT Stroke Width Transform

Sommaire

Introduction générale

La langue Amazighe est considérée comme un élément incontournable de la culture marocaine. Son patrimoine culturel, de tradition orale comme la majorité des langues africaines, ne se transmet de génération en génération qu'à travers la mémoire et les traditions. Cependant, dans cette ère de mondialisation devenue de plus en plus numérique, cette langue et son patrimoine culturel risquent d'être marginalisés vu la prédominance des langues des pays industrialisés. En effet, les mouvements culturels Amazighs ont vite constaté ce danger et ont revendiqué la conservation et la sauvegarde de cet héritage pour perpétuer les aspects de la culture et des traditions Amazighes auprès des futures générations. Ces revendications ont abouti à la création de l'institut royal de la culture Amazighe (IRCAM) et à la constitutionnalisation et la standardisation de cette langue et son alphabet Tifinagh. Par conséquent, un processus d'aménagement linguistique de la langue Amazighe a été lancé afin de la promouvoir et la revitaliser. Dans ces deux dernières décennies, beaucoup d'efforts ont été fournis pour adapter cette langue aux nouvelles technologies de l'information et de la communication (TIC), qui sont devenues un support social, afin d'assurer sa pérennité et son partage et permettre son traitement automatique. La reconnaissance automatique des caractères (OCR) constitue une opération clé dans ce processus d'aménagement linguistique permettant ainsi d'accélérer la conversion des archives (image et documents) en forme textuelle exploitable.

L'OCR consiste à convertir le texte contenu dans n'importe quel type d'images numériques en une forme complètement compréhensible et éditable par un ordinateur. Ceci implique l'intersection de trois domaines de recherche, à savoir, la reconnaissance des formes, l'intelligence artificielle et la vision par ordinateur [33]. Deux approches de reconnaissance coexistent : l'approche holistique et analytique. L'approche holistique élimine le problème de segmentation des mots en caractères en utilisant une technique descendante pour reconnaitre le mot en entier, cette approche repose sur un vocabulaire limité et exige l'utilisation d'un lexique ou d'un corpus [16]. D'une autre part, l'approche analytique repose sur une technique ascendante commençant au niveau des traits ou caractères jusqu'à la production d'un significatif texte. Dans cette approche, une segmentation implicite ou explicite des mots en caractères est nécessaire, ajoutant ainsi une complexité additionnelle au problème et introduisant des erreurs supplémentaires de segmentation au système. Néanmoins, avec la coopération de la phase de segmentation, le problème est réduit à la reconnaissance des caractères simples et isolés ce qui est bien adapté à des vocabulaires illimités [16]. La reconnaissance automatique de l'écriture peut être en ligne ou hors ligne. La reconnaissance en ligne est effectuée quand l'ordinateur reconnait les caractères au fur et à mesure qu'ils sont dessinés, tandis que la reconnaissance hors ligne est exécutée après la numérisation d'un document déjà rédigé [32].

La langue amazighe et son alphabet Tifinagh, souvent écartés, ont cumulé un retard considérable dans le domaine de la reconnaissance automatique de l'écriture par rapport aux autres langues telles que le latin, l'arabe et le chinois. Les deux dernières décennies ont connu une expansion de la langue Amazighe et son alphabet Tifinagh, et les recherches dans ce domaine sont devenues intenses afin de combler ce retard. Ces recherches ont abouti au développement de quelques systèmes OCR, cependant leurs performances sont loin d'atteindre la perfection (voir Section II. 3. du chapitre 2).

Cette thèse présente des contributions variées au problème de la reconnaissance automatique hors ligne de l'écriture Amazighe imprimée et manuscrite. En effet, en premier lieu, nous nous somme focalisés sur l'écriture Amazigh manuscrite en Tifinagh. En remarquant le manque existant au niveau des systèmes OCR performants et dans la quasi-impossibilité d'utiliser l'approche holistique, notre but était de concevoir un descripteur robuste visant à extraire des caractéristiques discriminantes de chaque caractère Amazigh. Ce descripteur se base sur la décomposition de l'image du caractère isolé en plusieurs zones chevauchées qui prennent en compte la structure et la forme des caractères, ensuite, pour chaque zone, des caractéristiques statistiques sont extraites. Dans la phase de reconnaissance, nous avons mis en place un perceptron multicouches afin de décider de la classe d'un caractère d'entrée. Le taux de reconnaissance obtenu par ce système a été jugé assez satisfaisant en comparaison aux résultats reportés dans la littérature. Pour augmenter la performance de ce système, nous avons apporté des améliorations sur plusieurs niveaux. D'abord, nous avons modifié le descripteur mentionné ci-dessus pour extraire des caractéristiques plus pertinentes. Ensuite, nous avons combiné ce descripteur amélioré avec plusieurs algorithmes de classification. Pour chaque classifieur, nous avons testé différentes configurations permettant la construction du modèle de classification pour choisir la configuration la plus adaptée au problème. En outre, nous avons combiné parallèlement les modèles construits pour concevoir un classifieur plus précis. Enfin, nous avons intégré l'option du rejet comme un post-traitement pour rejeter les caractères susceptibles d'être mal classés et diminuer ainsi le taux d'erreurs. Ces améliorations ont abouti à un système très performant et d'une forte capacité de généralisation. Il faut noter que tous les tests ont été menés sur la base d'images de référence des caractères Amazighs manuscrits AMHCD [44].

Dans le cas de l'écriture imprimée, la pertinence de notre descripteur développé pour les caractères manuscrits nous a poussé à le tester face à des caractères imprimés en différentes polices, tailles et styles. Mais, malheureusement, aucune base d'images publique pour tester et comparer les différents systèmes OCR, développés pour l'écriture imprimée, n'a été mentionnée dans la littérature. Ceci nous a conduit à créer une nouvelle base d'images, nommée APWID, contenant chacune un mot amazigh en Tifinagh. Les mots de notre base APWID ont été rendus avec une procédure automatique en utilisant, pour chaque mot, différentes combinaisons de polices, tailles et styles. Par conséquent, nous offrons une large variabilité des données dans la base, permettant ainsi l'évaluation et la comparaison à grande échelle des systèmes de reconnaissance de l'écriture imprimée. La base APWID est publique et accessible gratuitement sur internet. Après la création de cette base, nous avons conçu un système OCR automatique hors ligne des caractères Amazighs imprimés. Notre système, basé sur le même descripteur mentionné précédemment, est capable de reconnaitre les documents Amazighs dont les caractères sont imprimés avec différents styles, tailles et polices. Ce système a été testé sur la base APWID et a montré sa robustesse face aux différentes polices utilisées. Cependant, ce système est incapable de traiter les documents multilingues et les textes écrits sur des arrière-plans complexes tels que les images Web, les illustrations et les images de scènes naturelles. Actuellement au Maroc, le script Tifinagh cohabite avec les deux scripts arabe et latin, et il est désormais de plus en plus utilisé dans les différents domaines. Pour surmonter ces nouveaux défis, nous avons injecté au système précédent deux opérations de prétraitements visant d'abord à localiser et extraire le texte dans les images et de procéder, ensuite, à l'identification de son script. Pour identifier le script du texte, nous avons conçu un système capable de différencier entre les trois scripts les plus utilisés au Maroc (Arabe, Latin et Tifinagh) mais aussi les

nombres. Notre système se base sur un réseau de neurones convolutifs (CNN), un des algorithmes les plus populaires dans l'apprentissage profond. Les tests effectués sur une base d'images locale de 28800 mots ont donné des résultats très satisfaisants.

Cette thèse se constitue de deux parties. La première partie, divisée en deux chapitres, reporte l'état de l'art et l'architecture des systèmes OCR et présente un tour d'horizon de la langue Amazighe et son alphabet Tifinagh. La deuxième partie, constituée de deux chapitres, présente les différentes contributions apportées à la reconnaissance automatique hors ligne de l'écriture Amazighe imprimée et manuscrite.

Chapitre 1 : présente les concepts et l'architecture générale des systèmes OCR et les principales étapes qui les constituent. Ce chapitre donne un bref historique et survole les méthodes et les techniques les plus utilisées dans les différentes étapes constituantes d'un système OCR (prétraitements, extraction des caractéristiques, reconnaissance et post-traitement).

Chapitre 2 : est consacré à la langue Amazighe et son alphabet Tifinagh. En effet, un aperçu général de cette langue ainsi que l'histoire et l'origine de son alphabet Tifinagh seront présentées. Ensuite, le chapitre reporte les efforts fournis pour l'implantation de cet alphabet dans le domaine informatique et les différents styles de polices proposés. Enfin, il survole les travaux réalisés dans le contexte de la reconnaissance automatique de l'écriture Amazighe imprimée et manuscrite aux niveaux des ressources linguistiques, base de données développées et systèmes de reconnaissance automatique proposés.

Chapitre 3 : s'intéresse aux travaux que nous avons réalisé pour concevoir un système de reconnaissance automatique hors ligne de l'écriture Amazighe manuscrite. Il détaille notre nouveau descripteur proposé extrait des images des caractères Amazighs isolés à base des méthodes statistique et les résultats obtenus en le combinant avec un perceptron multicouches lors de la reconnaissance. Par la suite, il présente nos différentes améliorations effectuées pour augmenter la performance de ce système. Ces améliorations se résument par un changement dans le descripteur, l'utilisation et la combinaison de plusieurs algorithmes de classification ainsi que l'emploi de l'option de rejet comme post-traitement. Les résultats satisfaisants obtenus par ces améliorations seront présentés aussi.

Chapitre 4 : présente nos contributions apportées au domaine de la reconnaissance automatique hors ligne de l'écriture Amazighe imprimée. Il décrit en premier lieu la base proposée des images des mots Amazighs imprimés multi-polices, appelée APWID, ainsi, nous avons présenté certains détails sur la collecte de ses données, sa procédure de génération, son stockage et quelques protocoles de son utilisation. Ensuite, dans le même contexte, nous avons initié le problème de la reconnaissance automatique du texte en Tifinagh dans un environnement multilingue à partir des images Web ou de scènes naturelles et nous avons présenté une solution de bout en bout pour traiter ce genre de texte.

Pour conclure, nous abordons les points discutés dans ce mémoire de thèse et les perspectives envisagées pour les futurs travaux.

Partie. 1 : Architecture des systèmes de reconnaissance automatique des caractères et relation avec la langue Amazighe

Chapitre. 1 : Généralités sur les systèmes de reconnaissance automatique des caractères

Introduction

Depuis notre enfance, nous apprenons à lire et à écrire pendant nos premières années d'études. Au fur et à mesure, nous acquérons des très bonnes compétences en lecture qui nous permettent de lire la plupart des textes, qu'ils soient imprimés dans des polices, des tailles et des styles différents, ou même manuscrits soigneusement ou négligemment. La majorité écrasante devient capables, sans aucun problème, de lire : les impressions légères, lourdes ou renversées ; Publicités avec des polices décoratifs ; des caractères drôlement écrits, bruités ou incomplets ; mots mal orthographiés ou de styles artistiques et figuratifs. Parfois, les caractères et les mots peuvent sembler plutôt déformés et pourtant, par expérience et par contexte, la plupart des gens peuvent encore les comprendre. Au contraire, malgré les recherches intensives pendant les cinq dernières décennies, les compétences de lecture de l'ordinateur sont encore loin des celles des êtres humains. La plupart des systèmes de reconnaissance optique des caractères (systèmes OCR) ont toujours des problèmes concernant la lecture des documents texte dégradés, les textes manuscrits et les textes capturés dans les scènes naturelles.

L'objectif des systèmes OCR est de transformer automatiquement le texte (imprimé ou manuscrit) contenu dans une image en une représentation compréhensible par une machine et facilement reproductible et éditable par un système informatique. Ce domaine constitue un sujet de recherche très actif en raison de ses diverses applications dont les principales sont la numérisation et la reproduction des documents, l'archivage et l'indexation des documents, le tri postal, le traitement des formulaires, la lecture des plaques d'immatriculation, la compréhension des scènes, la géolocalisation, l'automatisation industrielle, la navigation robotique, etc.

En effet L'histoire de l'OCR peut être retracée dès 1900, lorsque le scientifique russe Tyuring a tenté de développer une aide pour les handicapés visuels [101]. Les premiers reconnaisseurs de caractères sont apparus au début des années 1940 avec le développement des ordinateurs numériques [42]. Les premiers travaux sur la reconnaissance automatique des caractères ont été concentrés sur un texte imprimé à la machine ou sur un petit ensemble de textes ou de symboles manuscrits bien distingués. Les systèmes OCR imprimés à la machine dans cette période utilisaient généralement un modèle de correspondance dans lequel une image est comparée à une bibliothèque d'images. Pour le texte manuscrit, des techniques de traitement d'image de bas niveau ont été utilisées sur l'image binaire pour extraire des vecteurs caractéristiques qui sont ensuite introduits dans des classifieurs statistiques. Des algorithmes réussis mais limités ont été implémentés principalement pour les caractères et les chiffres latins. Cependant, quelques études sur des caractères et des chiffres japonais, chinois, hébreux, indiens,

cyrilliques, grecs et arabes dans des cas imprimés à la machine et manuscrits ont également été initiées [31].

En 1950, la révolution technologique progressait à grande vitesse et le traitement électronique des données devenait un domaine important [101]. Les dispositifs de reconnaissance de caractères commerciaux disponibles dans les années 1950 ont été introduits pour la première fois où des tablettes électroniques capturaient les données de coordonnées x-y du mouvement de la pointe du stylet. Cette innovation a permis aux chercheurs de travailler sur le problème de la reconnaissance de l'écriture manuscrite en ligne [31]. Au milieu des années 1950, les machines OCR sont devenues disponibles dans le marché [138], et la première machine de lecture OCR [139] a été installée chez Reader's Digest en 1954.

Les systèmes OCR commerciaux apparaissant de 1960 à 1965 étaient souvent appelés OCR de première génération [104]. Les machines OCR de cette génération étaient principalement caractérisées par des formes de lettres contraintes. Les symboles ont été spécialement conçus pour la lecture de machines.

Au milieu des années 1960 et au début des années 1970, les machines à lire de la deuxième génération sont apparues [104]. Ces systèmes étaient capables de reconnaître les caractères imprimés à la machine et avaient également des capacités de reconnaissance des caractères manuscrits. Le premier et célèbre système de ce type était IBM 1287 en 1965. Pendant cette période, Toshiba a développé la première machine de tri automatique de lettres à base des numéros du code postal. Hitachi a également fabriqué la première machine OCR pour des performances élevées et un faible coût. Pendant cette période, des travaux importants ont été réalisés dans le domaine de la normalisation. En 1966, une étude approfondie des exigences OCR a été achevée et un jeu de caractères OCR standard américain a été défini comme OCR-A Figure 1-a. Cette police était hautement stylisée et conçue pour faciliter la reconnaissance optique. Une police européenne a également été conçue comme OCR-B qui avait plus d'apparence naturelle que la norme américaine Figure 1-b. Des tentatives ont été faites pour fusionner deux polices dans un standard à travers des machines capables de lire les deux standards.

(a) (b)

Figure 1. Jeu de caractères proposés pour la reconnaissance automatique

(a) Standard américain OCR-A

(b) Standard européen OCR-B

Les défis de la troisième génération étaient les documents de mauvaise qualité et la reconnaissance de l'écriture manuscrite, mais aussi la rapidité et la performance des systèmes [42]. Les exigences en matière de reconnaissance de l'écriture manuscrite ont augmenté, car beaucoup de données (les adresses postales, les montants sur les chèques, les noms, les adresses, les numéros d'identité et les valeurs en dollars écrites sur les factures et les formulaires) ont été écrites et elles devaient être traitées par l'ordinateur. Cependant, les techniques traditionnelles

d'OCR étaient principalement basées sur l'appariement des modèles des caractères, certaines caractéristiques géométriques, les contours et leurs dérivés, de telles techniques étaient incapables de traiter les données manuscrites. Pour en faire face les organisations de normalisation internationales ont conçu des modèles de caractères manuscrits à utiliser pour diminuer la variation du style de l'écriture manuscrite, et parfois, on demande aux scripteurs de suivre certaines consignes (écrire grand, fermer les boucles, utiliser des formes simples, ne pas lier les caractères, etc.) pour faciliter la tâche de la reconnaissance.

Au fur et à mesure que les années de recherche intensives se sont déroulées, et avec la naissance de plusieurs nouvelles conférences et ateliers tels que IWFHR (Atelier international sur les frontières dans la reconnaissance de l'écriture manuscrite), ICDAR (Conférence internationale sur l'analyse et la reconnaissance des documents) et d'autres, les chercheurs ont commencé à donner de l'attention à d'autres langues (arabe, chinois, indien, farsi, etc) et les techniques de reconnaissance ont gagné beaucoup d'élan de manière qu'on peut utiliser différents polices et styles pour l'écriture imprimée, mais aussi, les gens pouvaient écrire comme ils le faisaient normalement, et les caractères n'ont pas besoin d'être écrit comme des modèles spécifiés.

De nos jours, vu le développement technologique et la variété des supports d'acquisition (scanners, caméras, téléphones et tablettes mobiles,…), les images textuelles peuvent apparaitre sous différents types qu'on peut les catégoriser selon le type de l'appareil d'acquisition ou la nature de l'image (Document texte, scène naturelle ou image web). La Figure 2 illustre la classification des images textuelles.

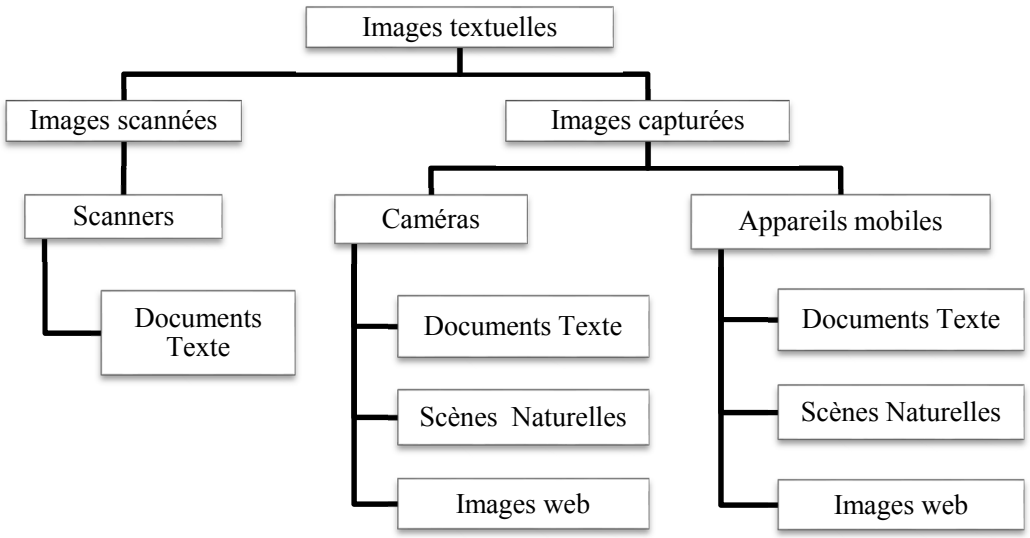


Figure 2. Classification des images textuelles

Les documents texte incluent les livres, les journaux, les formulaires, les magazines et les documents texte manuscrits (notes, lettres, etc.). Pour les scènes naturelles, le texte peut être présent sur les enseignes, les panneaux publicitaires ou signalétiques, les publicités murales, etc. Tandis que sur Internet, les images peuvent être des affiches publicitaires ou des scènes naturelles contenant du texte comme légende. La Figure 3 montre quelques exemples des images textuelles sous différentes formes.

(a) (b) (c)

Figure 3. Exemples des images textuelles

(a) Document texte scanné

(b) Scène naturelle contenant du texte

(c) Image Web d'une scène naturelle contenant du texte comme légende

Une fois qu'une image textuelle est capturée optiquement par un scanner ou par d'autres moyens optiques, l'image numérique est passée à un système de reconnaissance automatique constitué de quatre étapes principales :

- Les prétraitements qui améliore la qualité de l'image d'entrée et effectue les opérations nécessaires (détection du texte, extraction, segmentation, etc.) pour préparer les caractères ou les mots du texte pour la phase de l'extraction des caractéristiques ;
- L'extraction des caractéristiques qui vise à capturer les caractéristiques les plus distinctives des unités de reconnaissance (caractères ou mots) pour faciliter leur reconnaissance ;
- La reconnaissance qui se base sur des techniques de classification développées pour traiter les vecteurs de caractéristiques identifiant les caractères ou les mots ;
- Le Post-traitement qui effectue la correction orthographique des erreurs commises lors de la phase de reconnaissance.

I. Prétraitements et préparations du texte

En tant que première étape importante, les prétraitements subis par l'image servent à extraire les régions d'intérêt, à corriger les lacunes apparues pendant le processus d'acquisition de données et à améliorer et à nettoyer les images afin qu'elles puissent être traitées directement et efficacement durant l'étape de l'extraction des caractéristiques.

En effet, plusieurs facteurs peuvent affecter la qualité de l'image d'entrée rendant difficile son traitement et diminuant ainsi la performance du système OCR. Ces facteurs varient selon le type de l'image textuelle et incluent :

- Qualité du scanner ou de l'appareil d'acquisition (bruit, flou, …);
- Qualité physique du document à numériser;
- Basse résolution;

- Effets d'illumination non uniforme et conditions d'éclairage;
- Arrière-plans complexes ;
- Documents texte contenant des images ou graphiques ;
- Documents multilingues ;
- Inclinaison du texte ;
- Police, taille et style utilisé ;
- Documents manuscrits mal écrits ou cursifs.

Afin de remédier à certains des lacunes causées par les facteurs cités ci-dessus, des opérations de prétraitement sont nécessaires pour normaliser et éliminer les variations qui compliquent la classification et réduisent le taux de reconnaissance. La Figure 4 montre les principales opérations utilisées lors des prétraitements, le choix des opérations à utiliser varie selon la complexité du problème en face. Cette section décrit les principales opérations utilisées en prétraitements.

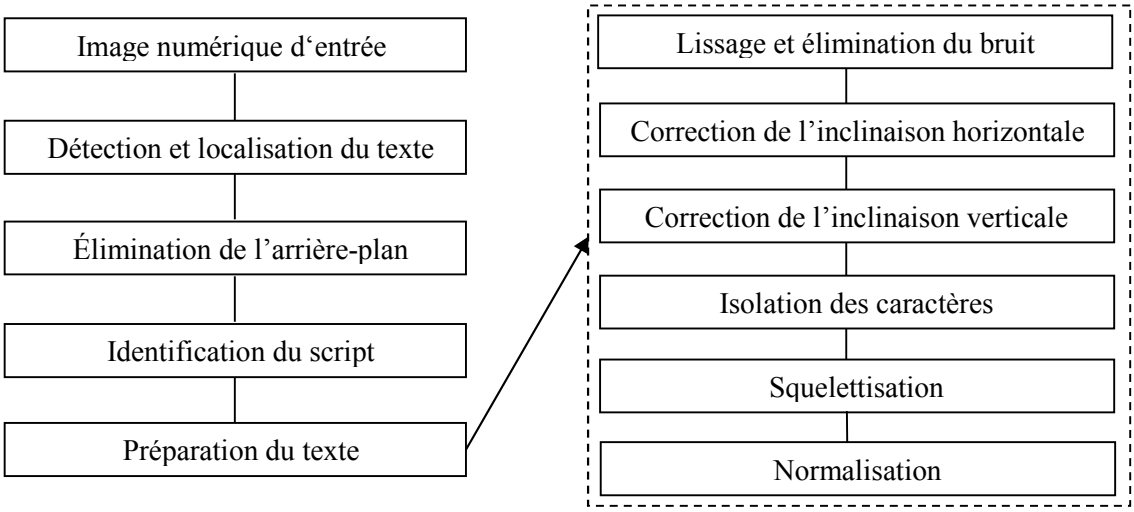


Figure 4. Les différentes étapes de prétraitement

1. Détection et localisation du texte

Le texte est l'un des moyens de communication les plus expressifs, il est présent tout autour de nous sous différentes formes (imprimé et manuscrit) et contient des informations assez importantes. D'une autre part, en plus des scanners, la disponibilité croissante de dispositifs mobiles de haute performance avec une capacité d'imagerie et de calcul a créé une opportunité pour l'acquisition et le traitement d'images à tout moment et n'importe où. Ceci a rendu la localisation et l'extraction du texte contenu dans les images un domaine qui génère beaucoup d'intérêt pour la recherche de nos jours.

En effet, avec les nouvelles tendances de la technologie, le marché des caméras digitales ou intégrées dans les supports mobiles a connu un grand essor par rapport à celui des scanners vu leur utilisation dans différents applications. L'extraction et l'analyse du texte présent sur ces images capturées reste un des principaux domaines d'application, ce texte peut être exploité dans plusieurs applications comme la compréhension des scènes, l'automatisation industrielle, la géolocalisation automatique, la navigation robotique, la lecture des plaques d'immatriculation, l'indexation des documents, etc.

Cependant, la localisation et l'extraction du texte présente plusieurs défis nécessitant l'utilisation des techniques avancées de traitement d'images et de vision par ordinateur. En effet, plusieurs travaux ont été menés pour l'extraction du texte dans les documents texte scannés (livres, magazines, journaux, etc.), ces travaux atteignent des taux de performance satisfaisants, quelques récents travaux sont présentés dans [86, 90, 145]. Le traitement des documents texte capturés par des caméras numériques ou les images de scènes naturelles contenant du texte présente des nouveaux défis où les méthodes traditionnelles semblent incapables de les surmonter et échouent face à ce nouveau et prometteur mode d'acquisition. La majorité des travaux récents ont été consacrés à la localisation et l'extraction du texte à partir des images de scènes naturelles.

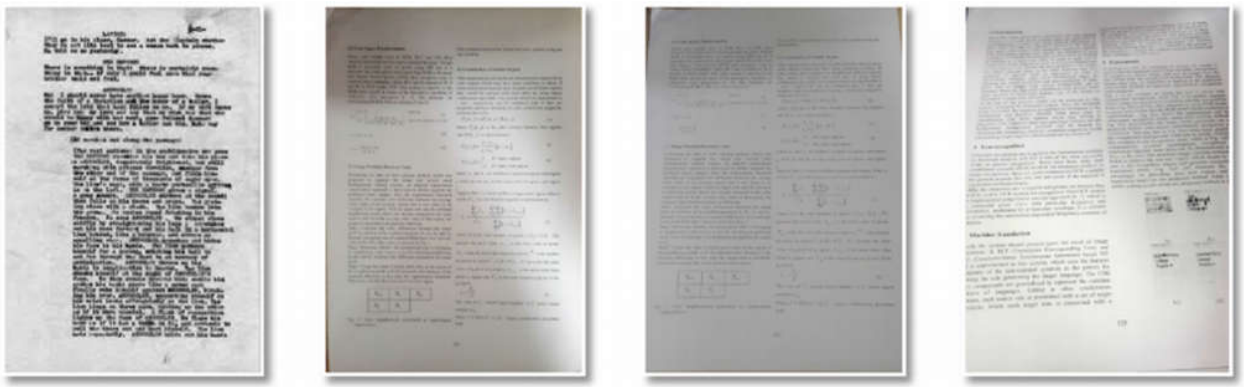
Dans cette partie, nous allons survoler les principaux problèmes rencontrés lors du traitement des images textuelles et survoler les travaux réalisés pour en remédier.

1. 1. **Problèmes majeurs**

La complexité des environnements, les modes d'acquisition d'images et la variation du contenu posent différents défis lors de l'extraction du texte à partir des images capturées ou scènes naturelles [78].

En effet, les documents texte capturés par des caméras (digitales ou intégrées dans des mobiles) présentent des nouveaux problèmes photométriques (conditions d'éclairage, mode d'acquisition, qualité du support d'acquisition,…) et géométriques liés à la forme de l'image textuelle (Figure 5) :

- Flou et bruit : le mode et les conditions d'acquisition d'image peuvent causer une dégradation de la qualité de l'image. Le flou et le bruit sont parmi les distorsions possibles ;



(a) (b) (c) (d)

Figure 5. Problèmes rencontrés pour les documents texte capturés par des caméras [78]

(a) Flou et bruit
(b) Illumination non uniforme
(c) Basse résolution
(d) Distorsion de la perspective et déformation 3D

- Illumination non uniforme : les conditions d'éclairage (ombre et surfaces réfléchissantes), dues à l'environnement physique, sont non contrôlées lors de l'acquisition de l'image causant ainsi la distorsion des couleurs et la détérioration des caractéristiques visuelles ;
- Basse résolution : les images capturées à l'aide de des caméras digitales ou intégrées dans les téléphone ou tablettes portables souffrent d'une faible résolution par rapport aux scanners haute résolution ;
- Distorsion de la perspective et déformation 3D : se produisent lorsque l'axe optique de la caméra n'est pas perpendiculaire au plan du texte. Par conséquent, Les lignes de texte ne restent plus parallèles entre elles et la forme rectangulaire limitant le texte est perdue.

En plus de ces problèmes, les images de scènes naturelles souffrent encore de nouveaux problèmes liés à la complexité des arrière-plans et à la variation du contenu (Figure 6) :

Figure 6. Problèmes rencontrés pour le texte contenu dans les images de scènes naturelles

- Variation de style du texte : le texte diffusé dans les scènes naturelles doit être attirant et accrocheur. Par conséquence, ce texte peut être dans n'importe quelle police, taille, couleur ou mise en page ;
- Orientation et variation de taille du texte : même si le plan du texte présent dans les images de scènes naturelles reste parallèle au plan de l'image, il se peut que le texte lui-même soit orienté ou courbé dans un certain sens, en outre, les ratios d'aspect peuvent être différents vu la variation de la taille du texte. Par conséquent, les simples techniques ne peuvent être appliquées ;
- Arrière-plans complexes : dans les environnements naturels, les images contiennent trop d'informations qui peuvent être textuelles ou non, l'existence des objets artificiels ayant des apparences semblables au texte rend la séparation du texte du non-texte une tâche difficile ;
- Textes multilingues : aujourd'hui, on constate que les scènes naturelles peuvent contenir du texte écrit dans différents scripts, surtout dans les pays où plusieurs langues cohabitent.

1. 2. **Travaux réalisés**

La détection et localisation du texte à partir des images est une étape très importante pour assurer une bonne performance des systèmes OCR, elle vise à localiser les blocs de texte contenu dans les images pour les préparer à la phase de reconnaissance (Figure 7).

En réalité, les défis de l'extraction s'accroissent qu'on est face à des images de scènes naturelles ou des images Web, où le texte est présent sur des arrière-plans complexes et sous différents styles et différentes tailles. Dans cette partie, nous allons nous intéresser à la détection et localisation du texte diffusé dans les scènes naturelles ou les images Web et survoler les principales techniques utilisées dans ce domaine.

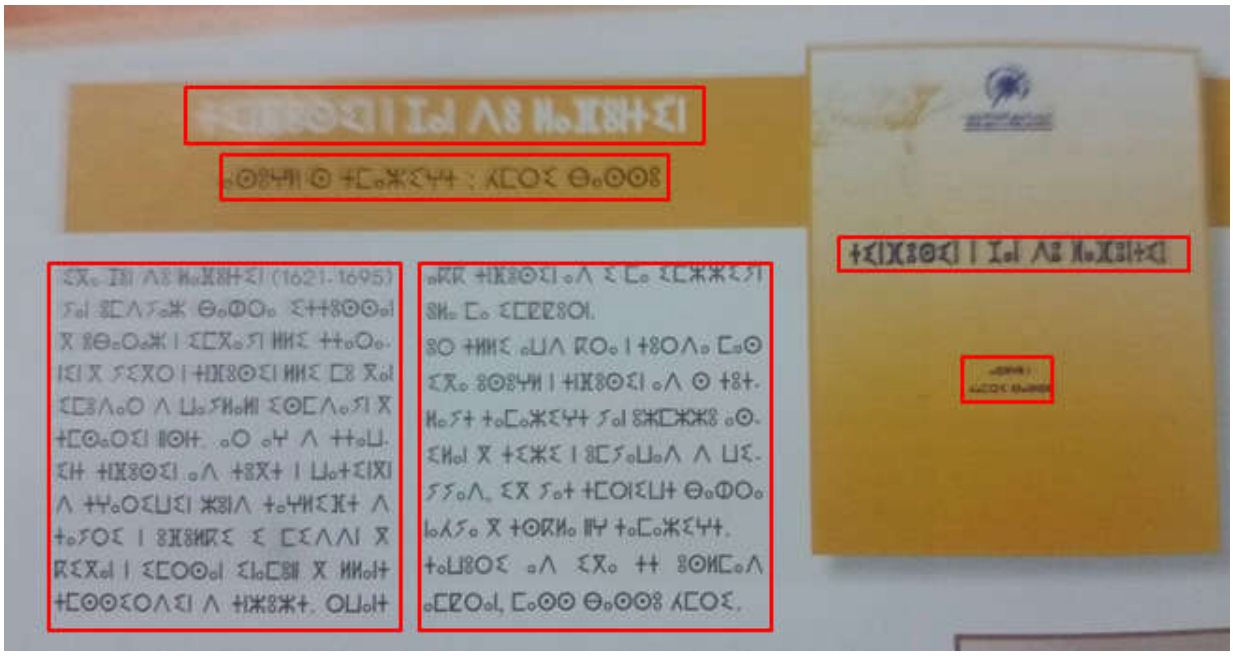


Figure 7. Détection des blocs de texte dans les images

Au début des années quatre-vingt-dix, les chercheurs ont réalisé l'importance du texte présent dans les images, de scènes naturelles ou web, et ont effectué certains travaux visant à le localiser et l'extraire, ces travaux ont déclenché la recherche dans ce domaine. Ensuite, les travaux ont progressé à un rythme assez rapide et la recherche a rapporté différentes techniques et méthodologies. En parcourant la littérature, on voit que les méthodes utilisées pour la détection et la localisation peuvent être classées selon les caractéristiques du texte utilisées (Texture, composants connectés ou contour) [78].

Les méthodes basées sur la texture traitent les textes comme un type de texture particulier et utilisent leurs propriétés texturales pour faire la distinction entre les zones textuelles et non-textuelle dans les images. Ces méthodes peuvent être statistiques (matrice de co-occurrences, modèles de Markov, voisinage, etc.) ou spatio-fréquentielles (Fourier, Gabor, ondelettes, DCT, etc.). Les techniques basées sur la texture sont adaptées aux images présentant du texte sur un arrière-plan complexe. Cependant, ces techniques sont généralement coûteuses en termes de calcul. En outre, ces méthodes traitent principalement des textes horizontaux et sont sensibles à la rotation et au changement d'échelle [168].

Pour les méthodes basées sur les régions, elles sont généralement basées sur l'analyse des propriétés géométriques des composants connexes CC pour choisir les régions textes candidates qui sont regroupées ensuite en deux groupes textes et non-textes [161]. La largeur du trait est aussi considérée comme une excellente caractéristique pour extraire le texte de l'arrière-plan, l'operateur SWT et ses variantes sont souvent utilisés dans ce contexte [66]. Une autre technique qui a attiré beaucoup d'attention, dans ces dernières années, est l'extraction des régions extrêmes maximalement stables MSER pour identifier les régions textes candidates, un classifieur est utilisé pour décider des régions textes finales [162].

Les techniques basées sur les contours considèrent que la présence d'un contraste élevé entre le texte et l'arrière-plan fait du contour une caractéristique clé du texte contenu dans l'image. En effet, L'épaisseur du contour, sa densité, le contraste entre le texte est l'arrière-plan sont les éléments qui distinguent le texte dans un document. Les contours de texte sont détectés et regroupés, puis les régions non textuelles sont filtrées sur la base de plusieurs heuristiques [95, 97].

Certains travaux de recherche ont établi une approche hybride en combinant les méthodes basées sur la texture, les régions et les contours pour surmonter les limites correspondantes à ces dernières [142, 157, 166].

Une fois le texte est détecté et localisé dans l'image, une étape d'élimination de l'arrière-plan est nécessaire.

2. Extraction du texte et élimination de l'arrière-plan

Après une détection et une localisation efficaces du texte dans une image, une séparation du texte de l'arrière-plan est nécessaire. Cette tâche dépend de la complexité de l'arrière-plan considéré et peut être traitée à l'aide de trois types de méthode : les méthodes basées sur des seuils ; les méthodes basées sur les couleurs et les méthodes basées sur des modèles statistiques [78]. La Figure 8 montre quelques résultats des différentes méthodes utilisées pour extraire le texte.

(a)

(b)

(c)

Figure 8. Résultats des différentes techniques d'extraction de texte

(a) Techniques basées sur le seuillage

(b) Techniques basées sur le regroupement par les k-moyennes

(c) Techniques basées sur la segmentation

Les méthodes basées sur le seuillage consiste à binariser l'image en niveaux de gris et la représenter en deux couleurs seulement, en noir et blanc. La binarisation permet d'extraire les pixels du texte de l'image en laissant derrière les pixels de l'arrière-plan. De manière générale, les techniques de binarisation sont de deux types : le seuillage global et le seuillage local [20]. Dans le seuillage global, un seuil global est choisi sur toute l'image et les pixels de l'image sont séparés selon ce seuil. Le seuillage d'Otsu [21], par entropie maximale [22], par moments [23] sont des exemples du seuillage global. Tandis que pour le seuillage local, l'image est divisée en nombreuses sous-images fixes, puis, des valeurs des seuils locaux sont calculés dans chaque sous-image. Les techniques les plus populaires dans cette catégorie sont le seuillage de Niblack [24], Sauvola [25] et Gatos [26]. Les techniques de seuillage sont adaptés à des images textuelles dont l'arrière-plan est moins complexe, les images de scènes naturelles souffrent principalement d'un éclairage inégal et d'un arrière-plan complexe de sorte que ces techniques ne conviennent pas pour de telles images. Les méthodes à seuils locaux peuvent produire de bons résultats lorsque le texte est bien localisé dans l'image.

Un autre type d'élimination de l'arrière-plan exploite les informations de couleur présentes dans l'image et utilise des algorithmes de regroupement tels que les K-moyennes ou ceux basés sur des métriques sélectives pour segmenter le texte de l'arrière-plan [147][100]. Ces techniques peuvent fonctionner dans un espace de couleur ou utiliser de nombreux espaces de couleur, elles sont mieux adaptées aux images de scènes naturelles.

Des travaux récents dans le domaine de la segmentation ont mis en évidence l'utilisation de divers modèles statistiques pour effectuer l'extraction de texte à partir d'images complexes ayant un contexte difficile. Ces modèles incluent le champs aléatoire de Markov (MRF), les modèles basés sur les graphes et les modèles de mélange gaussien (GMM) [103].

Dans les images prises dans un environnement multilingue, après la localisation et l'extraction du texte, on procède à l'identification des scripts des textes présents avant leur reconnaissance.

3. Identification du script

Un script est la forme graphique du système d'écriture d'une langue particulière. Il existe six systèmes d'écriture importants autour du monde [56] :

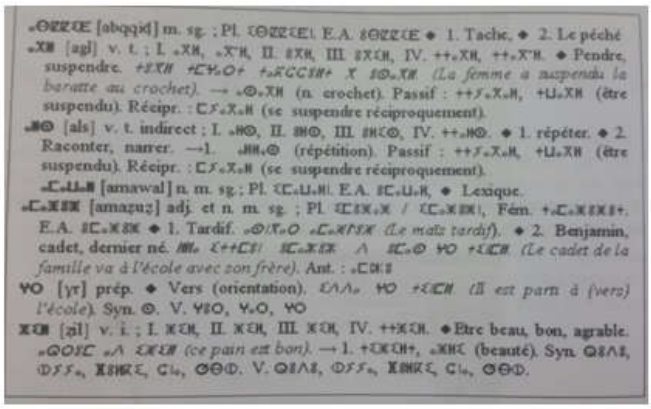
- **Logographique** : Un logogramme, également appelé idéogramme, se réfère à un symbole qui représente graphiquement un mot complet. En conséquence, le nombre de caractères dans un système d'écriture idéographique s'étend généralement à des milliers. Un exemple de script logographique est le Han, qui est principalement associé au chinois.
- **Syllabique** : Dans un système syllabique, chaque symbole écrit représente un son phonétique ou une syllabe. Le Kanas utilisé par l'écriture japonaise est un exemple du système syllabique.
- **Alphabétique** : Un alphabet est un ensemble de caractères représentant des phonèmes d'une langue parlée. Les exemples de scripts qui suivent ce système sont le grec, le latin, le cyrillique et l'arménien. Le script latin, également appelé script romain, est utilisé par de nombreuses langues à travers le monde avec des degrés divers de modification d'une langue à l'autre. Par rapport aux autres scripts, les caractères latins classiques sont de structure simple, principalement composés de quelques lignes et arcs. L'autre script majeur sous le système alphabétique est le cyrillique. Ce script est utilisé par certaines langues d'Europe de l'Est, d'Asie et de régions slaves. Il faut noter que le système d'écriture (Tifinagh) adopté pour l'Amazigh est de nature alphabétique.
- **Abjad** : Le système d'écriture Abjad est semblable au système alphabétique, mais a des symboles que pour les sons consonantiques. Contrairement à la plupart des autres scripts dans le monde, Abjad est écrit de droite à gauche dans une ligne de texte. Deux scripts importants dans cette catégorie sont l'arabe et l'hébreu.
- **Abugida** : Abugida est un autre système d'écriture alphabétique utilisé par la famille de scripts brahmique issu de l'ancien script Brahmi indien et comprend presque tous les scripts de l'Inde et de l'Asie du Sud-est.
- **Système à trait :** Le dernier système d'écriture est le système à trait dans lequel les symboles ou les caractères ne sont pas arbitraires mais codent les caractéristiques phonologiques des phonèmes qu'ils représentent. Un script éminent de ce genre est le Hangul coréen.

La plupart des systèmes OCR sont spécifiques au script auquel ils ont été développés, ceci signifie qu'un système OCR conçu pour un script ne fonctionnera pas pour un autre. Cependant, dans un environnement multilingue où plusieurs langues et scripts cohabitent, les systèmes OCR

doivent être capables de reconnaître les caractères indépendamment du script dans lequel ils ont été écrits. En général, la reconnaissance des différents caractères multi-scripts dans un seul module OCR reste une tâche difficile vue les différences dans le style, l'orientation et la structure des scripts en question. Une autre solution consiste à utiliser une banque des systèmes OCR correspondants à tous les scripts différents qui devraient être reconnus. Les caractères d'un document d'entrée peuvent ensuite être reconnus de manière fiable en sélectionnant le système OCR approprié pour chaque script dans la banque des systèmes OCR. Néanmoins, cela nécessitera de connaître a priori le script dans lequel le texte est écrit. Malheureusement, ces informations peuvent ne pas être facilement disponibles. En outre, l'identification manuelle des scripts peut être très fastidieuse. Par conséquent, les techniques d'identification automatique des scripts sont nécessaires pour identifier les scripts dans le document d'entrée, puis rediriger le texte à reconnaitre au système OCR approprié.

Les chercheurs ont constaté que l'identification du script est l'un des composants clés de la reconnaissance de caractères dans un environnement multilingue vue ses diverses applications comme : le traitement automatique de la langue ; la traduction automatique ; la compréhension des scènes ; la navigation mobile ; le tri, l'indexation et l'analyse des documents ; etc. En effet, il y a eu un intérêt croissant pour l'identification du script au cours des dernières années. Les travaux réalisés ont traité ce problème à différents niveaux, à savoir, page, bloc de texte, ligne ou niveau mot [111]. Le traitement au niveau mot est préconisé lorsque plusieurs scripts sont mélangés dans le même paragraphe tandis que les autres sont utilisés face aux documents présentant des lignes, des paragraphes ou des blocks de texte du même script (Figure 9).

L'identification des scripts repose sur le fait que chaque script possède une structure et distribution spatiale unique et des attributs visuels qui permettent de le distinguer par rapport aux autres scripts. Ainsi, la tâche de base impliquée dans l'identification de script est de concevoir une technique pour découvrir ces caractéristiques à partir d'un document donné, puis classifier le script en conséquence. Sur la base de la nature de l'approche et des caractéristiques utilisées, ces méthodes peuvent être divisées en deux grandes catégories : les méthodes basées sur la structure et celles basées sur l'apparence visuelle [23]. Pour les méthodes structurelles, elles supposent que chaque script possède des caractères qui ont leurs propres et spécifiques formes, géométries, orientations, occupations spatiales, etc. Ces techniques se basent généralement sur l'analyse des composants connexes et des caractéristiques structurelles et topologiques [25]. Les méthodes basées sur l'apparence visuelle utilisent généralement l'analyse de la texture et les profiles des projections pour identifier le script du texte, Les caractéristiques utilisées peuvent être extraites de la matrice des co-occurrences [26] ou basées sur certaines transformées (ondelettes, Gabor, etc.) [27]. Des techniques récemment utilisées pour identifier les scripts se basent sur des réseaux de neurones artificiels tels que les réseaux de neurones convolutifs [28] et récurrents [29].



(a)

(b)

(c)

Figure 9. Les différents niveaux d'identification des scripts

(a) Traitement au niveau mot

(b) Traitement au niveau ligne

(c) Traitement au niveau paragraphe ou bloc

Après l'identification du script du texte, l'image du texte est redirigée au système OCR correspondant. Cette image subit certaines opérations pour préparer les caractères ou les mots du texte pour la phase de l'extraction des caractéristiques.

4. Préparation du texte

4. 1. Lissage et élimination du bruit

Le bruit est toujours présent dans les images numériques depuis l'acquisition jusqu'au traitement indiquant ainsi des informations indésirables à éliminer (artefacts, flou, contraste, bords irréalistes, lignes invisibles, arrière-plans perturbés, etc.). En effet, l'élimination du bruit est un problème historique et persistant et toujours inclus dans les étapes de prétraitement de la plus part des applications de traitement automatique des images.

En OCR, Farahmand et al [47] ont regroupé les différents types de bruit pouvant apparaitre dans les documents scannés et ont discuté quelques méthodes pour éliminer chaque type. Tandis que pour les images de scènes naturelles ou images Web, Joshi et al [73] présente une étude comparative des différentes méthodes pour restaurer et éliminer le bruit et le flou de ce genre d'images.

Actuellement, la littérature contient de très nombreuses techniques qu'on peut les regroupées, selon Saba et al [134], dans deux classes : techniques de filtrage spatial et technique de filtrage dans un domaine transformé comme le montre la Figure 10.

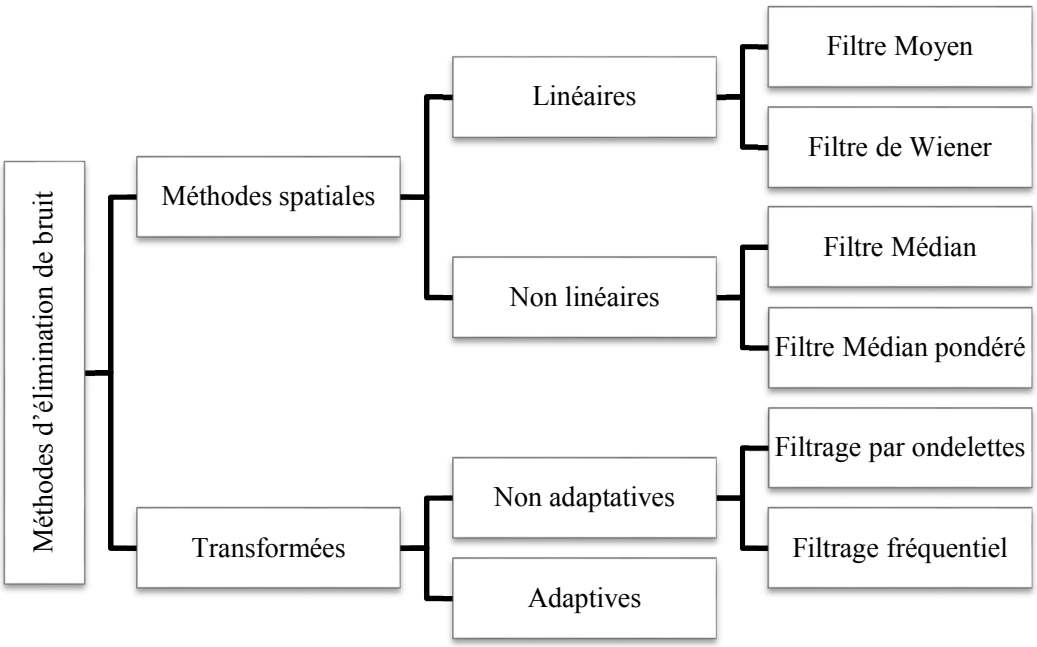


Figure 10. Classification des techniques d'élimination du bruit

Le filtrage spatial est une technique traditionnelle rapide pour éliminer le bruit, il existe deux types de filtrage spatial : linéaire et non-linéaire. Les filtres linéaires ont tendance à éliminer les détails des contours, détruire les lignes et lisser l'image. Le filtrage moyen et celui de Wiener sont des exemples de filtrage spatial. Pour les filtres non linéaires, les opérations de morphologie (érosion et dilatation) et le filtre médian sont les plus connus. Ils sont utiles quand le type du bruit est non identifié et sont capables de préserver les informations structurelles des contours.

Le filtrage dans un domaine transformé a été aussi largement utilisé pour l'élimination du bruit, le filtrage fréquentiel de Fourier et le filtrage par ondelettes sont les plus populaires dans cette catégorie.

4. 2. **Détection et correction de l'inclinaison du texte**

Dans certains cas, le texte à reconnaitre dans l'image d'entée peut présenter une certaine inclinaison due à une légère rotation du document lors de la numérisation ou à l'angle de capture du texte par les caméras ou même due aux scripteurs lors de l'écriture du document. Cette inclinaison doit être éliminée car elle peut réduire dramatiquement le taux de reconnaissance. En réalité, il existe deux types d'inclinaison possibles, une inclinaison horizontale appelée *SKEW* et une verticale appelée *SLANT*.

4. 2. 1. **Inclinaison horizontale**

L'inclinaison horizontale du texte signifie une déviation des lignes du texte par rapport à l'axe horizontal. Cette inclinaison peut nuire à la performance du système OCR et devrait donc être détectée et corrigée. Comme le montre la Figure 11, ce phénomène peut apparaitre dans les documents numérisés ou capturés par des caméras mais aussi dans le texte capturé dans des images de scènes naturelles.

(a) (b)

(c)

Figure 11. Inclinaison du texte dans les différents types d'images
(a) Texte imprimé incliné dans un extrait d'un document scanné
(b) Texte incliné dans une image de scène naturelle
(c) Texte manuscrit incliné dans un extrait d'un document scanné

La solution consiste à estimer l'angle de déviation et la corriger par une rotation du texte jusqu'à ce que la ligne de base du texte soit parallèle à l'axe horizontal (Figure 12) [70].

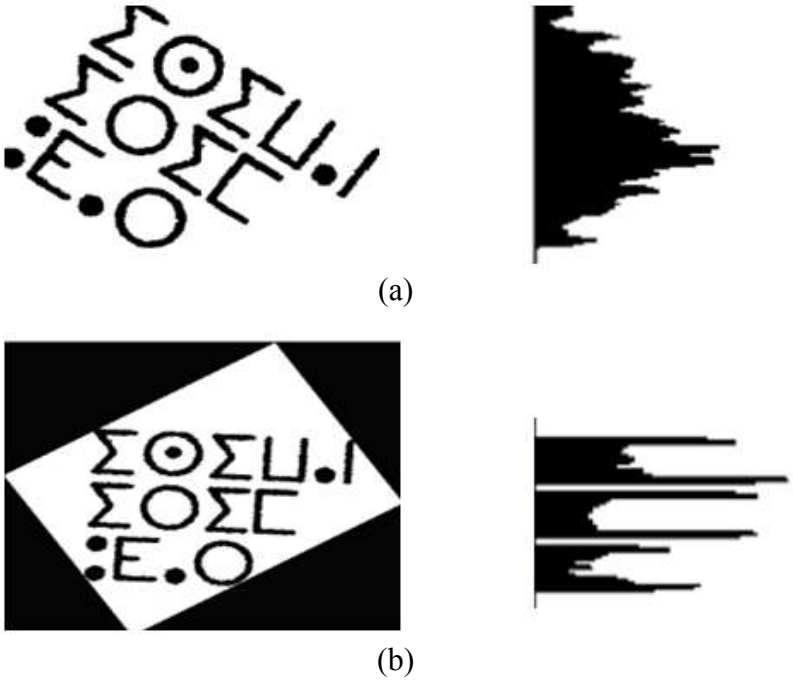


Figure 12. Texte avant et après la correction de l'inclinaison horizontale

(a) Texte incliné et son histogramme de projection horizontale

(b) Texte après correction de l'inclinaison et son histogramme de projection horizontale

Jusqu'à présent, de nombreuses méthodes de détection et correction de l'inclinaison horizontale du texte ont été proposées. Ces méthodes comprennent, entre autres, l'analyse du profil de projection [118], la transformée de Hough [15], le regroupement par plus proches voisins [98], la morphologie mathématique [110], etc.

4. 2. 2. **Inclinaison verticale**

L'inclinaison verticale des caractères apparait souvent lors de l'écriture manuscrite ou lors de l'utilisation des polices italiques dans l'écriture imprimée (Figure 13). C'est la déviation ou l'angle entre l'axe vertical et la direction dominante des traits verticaux.

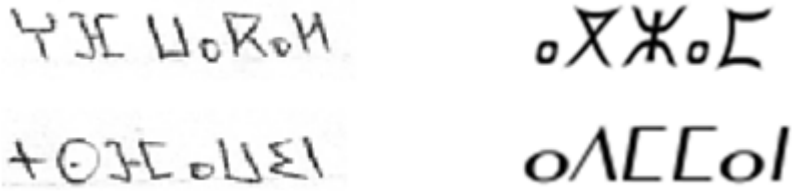


Figure 13. Texte présentant une inclinaison verticale (Slant)

(a) Texte en Tifinagh manuscrit

(b) Texte en Tifinagh imprimé en italique

La correction de cette inclinaison est une opération importante lors du prétraitement qui permet de normaliser tous les caractères dans une forme standard, réduire les variations entre les

symboles du même caractère, améliorer la qualité de segmentation des mots en caractères et obtenir plus de précision dans la phase de reconnaissance (Figure 14).

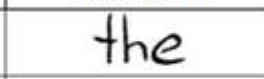
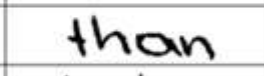
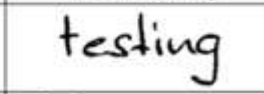
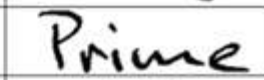
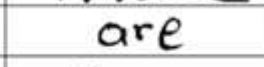
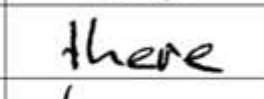
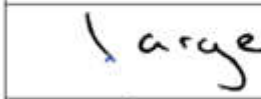
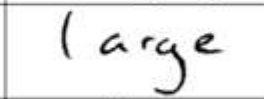
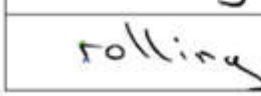
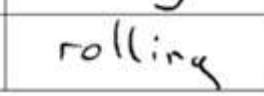
Image originale	Après correction du SLANT
	
	
	
	
	
	
	
	

Figure 14. Texte avant et après la correction de l'inclinaison verticale [131]

Dans la littérature, plusieurs méthodes ont été proposées pour résoudre ce problème, la plupart de ces méthodes ont été basées principalement sur l'analyse du profile de projection verticale [131].

4. 3. **Segmentation du texte**

La segmentation des blocs de texte contenu dans les images est considérée comme l'une des étapes les plus importantes du prétraitement. Elle consiste à diviser les blocs de texte extraits des images ou documents en petites unités de reconnaissance de base qui pourraient être des caractères, des traits, des mots partiels ou des ligatures en fonction du script étudié et de la méthodologie de reconnaissance utilisée. En effet, les chercheurs ont reconnu l'importance de l'étape de segmentation dans le système de reconnaissance des caractères et confirment qu'une bonne segmentation peut conduire à des taux de reconnaissance améliorés et vice versa. Cependant, le processus de segmentation du texte, imprimé ou manuscrit, n'est pas trivial et comporte plusieurs problèmes et défis. Effectivement, des problèmes dus aux : variété des polices, styles et résolutions du texte imprimé ; cursivité et styles de l'écriture manuscrite qui varie d'une personne à autre ; mauvaise qualité des documents et bruit introduit lors de la numérisation ; etc. peuvent engendrer des phénomènes tels que la fusion des lignes, des caractères touchés, des caractères chevauchés et des caractères cassés rendant ainsi la segmentation du texte une tâche très complexe (Figure 15).

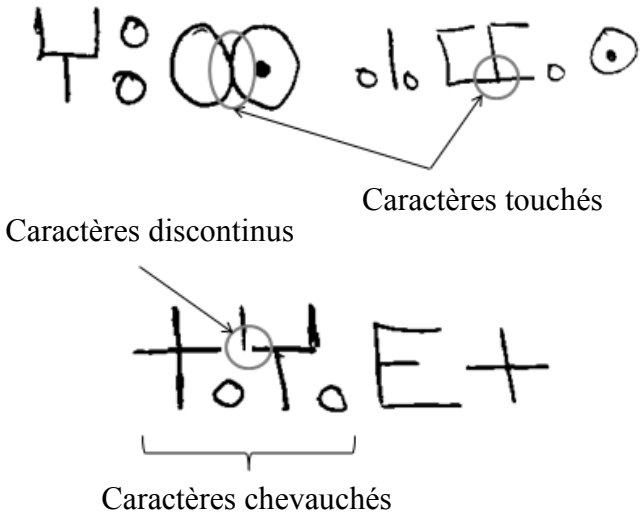


Figure 15. Problèmes rencontrés lors de la segmentation du texte

Après l'extraction des blocs de texte, une pré-segmentation du texte est nécessaire pour découper les blocs en lignes [96] et ensuite les lignes en mots [133]. Selon le script étudié, ceci est généralement effectué en se basant sur la transformée de Hough, l'analyse des profils des histogrammes horizontaux et verticaux et l'analyse des composants connectés. Une fois les mots du texte à reconnaitre sont localisés, ils peuvent être directement reconnus en utilisant l'approche holistique qui consiste à reconnaitre le mot en entier en se basant sur le vocabulaire du script en question. Dans l'absence de ce dernier, la segmentation des mots en caractères ou en unités de reconnaissance (graphèmes, ligatures, etc.) devient inévitable, les caractères sont ensuite regroupés après leur reconnaissance pour former les mots de départ. La segmentation en caractères est une tâche complexe qui, selon le script étudié, présente plusieurs défis et nécessite des techniques développées pour l'accomplir.

De nombreuses techniques ont été développées pour la segmentation du texte en caractères ou graphèmes [135][77] et la plupart d'entre elles sont spécifiques au script sous considération et peuvent ne pas fonctionner pour d'autres scripts. Ces techniques peuvent être explicites ou implicites [130].

Dans les techniques explicites, la segmentation se fait en se basant sur la structure des caractères à côté des points de segmentation sans aucune connaissance a priori de l'alphabet. La reconnaissance du caractère se fait dans une étape appart en utilisant des techniques d'appareillement ou d'apprentissage. Les techniques explicites, souvent appelées techniques de dissection, se basent généralement sur l'analyse du profile de projection verticale et l'analyse des composants connectés [28].

Les techniques implicites reposent sur la reconnaissance pour valider les hypothèses de segmentation en se basant sur l'alphabet du script étudié, par conséquence, la segmentation et la reconnaissance sont réalisées en même temps. La segmentation implicite fournit tous les segments possibles pour segmenter le mot et laisse à la reconnaissance la tâche de choisir la meilleure segmentation où chaque segment doit forcément appartenir à un des caractères de

l'alphabet en question. Les techniques implicites se basent généralement sur le fenêtrage, les graphes et les modèles de Markov [28].

Il est difficile de comparer entre les deux techniques et l'utilisation d'une technique ou d'une autre dépend de la complexité du script étudié (imprimé ou manuscrit, cursif ou non, présence des caractères touchées, chevauchés ou découpés, etc.).

4. 4. **Squelettisation**

La squelettisation fournit une représentation efficace et compacte des objets de l'image en réduisant leur dimensionnalité à un axe médian ou squelette tout en préservant leurs propriétés topologiques et géométriques (Figure 16). Le squelette obtenu lors de cette opération doit obligatoirement satisfaire les conditions suivantes [137] :

- Être le plus fin possible (1 pixel d'épaisseur) ;
- Maintenir la connectivité et la forme de l'objet;
- Être centré dans l'objet qu'il représente.

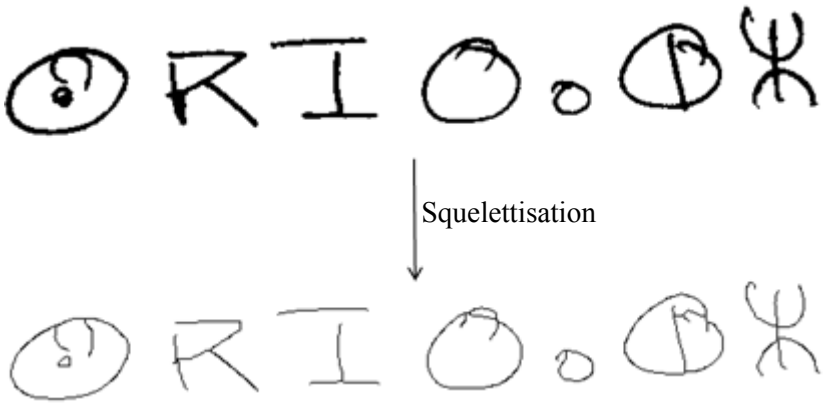


Figure 16. Squelettisation d'un mot Amazigh en Tifinagh

Dans la littérature, on trouve deux types d'approches pour la squelettisation, itérative et non-itérative [137]. Dans l'approche itérative, les techniques utilisées peuvent être séquentielles ou parallèles, ces techniques sont semblables dans la détermination des pixels à supprimer et différent dans le temps de suppression. En séquentiel le pixel est supprimé dès qu'il est jugé indésirable [3] alors qu'en parallèle la suppression s'effectue après la détermination de tous les pixels indésirables [40]. Pour l'approche non-itérative, le squelette est extrait directement sans examiner chaque pixel individuellement mais ces techniques sont lentes, difficiles à mettre en œuvre et généralement basées sur les réseaux de neurones [7], les diagrammes de Voronoi [99] et la transformation en ondelettes [163].

4. 5. **Normalisation des caractères**

Le résultat de l'étape de segmentation des caractères produit des images des caractères isolés de différentes tailles. Avant de les passer à l'étape de l'extraction des caractéristiques et pour assurer l'uniformité des données, les caractères isolés sont normalisés dans une taille

spécifique décidée empiriquement en fonction de l'application et des techniques d'extraction des caractéristiques ou de classification utilisées (Figure 17).

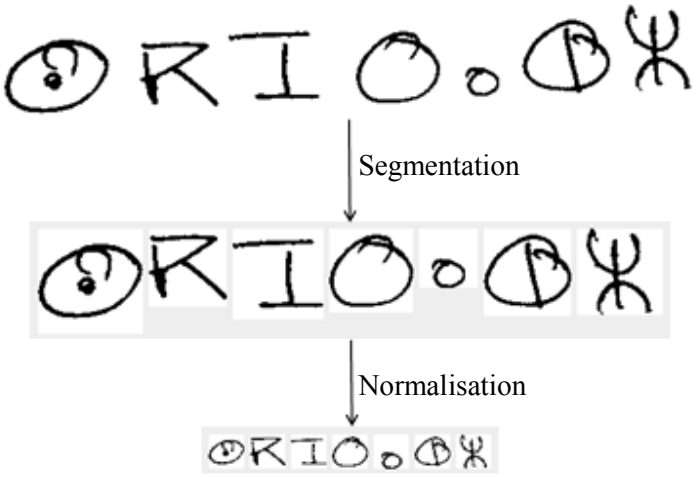


Figure 17. Normalisation des caractères après la segmentation

Le processus de redimensionnement de l'image produit une nouvelle version de l'image de taille différente mais avec une certaine perte d'informations et peut ajouter des informations indésirables (flou, bruit, etc.). On parle de sur-échantillonnage lorsqu'on agrandit l'image et de sous-échantillonnage lorsque la taille de l'image d'entrée est réduite (Figure 18) [151].

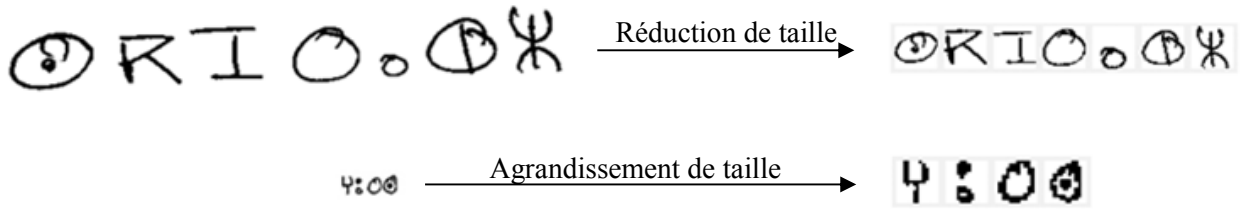


Figure 18. Effets de la réduction et l'agrandissement de la taille des images

Les algorithmes de redimensionnement des images les plus simples sont ceux basés sur l'interpolation linéaire [24] comme l'interpolation bilinéaire, bi-cubique et Lanczos. Ils sont rapides mais génèrent des effets indésirables sur l'image de destination tels que le flou, la pixellisation et le sur-lissage.

Les algorithmes d'interpolation d'image dirigée vers le contour sont plus complexes et utilisent des informations antérieures sur les images. Ces algorithmes corrigent les lacunes de l'interpolation linéaire mais peuvent corrompre les textures de l'image. L'interpolation d'image guidée par le contour (EGII), l'interpolation itérative à base de courbure (ICBI) et les algorithmes d'interpolation de convolution cubique directionnelle (DCCI) sont les plus populaires dans ce genre d'algorithmes [164][129].

Les algorithmes basés sur la régularisation [88] considèrent le problème de redimensionnement d'image comme un problème inverse Az = u, où z est l'image destination recherchée, u est l'image source d'entrée et A est l'opérateur de redimensionnement constitué généralement d'un filtrage passe-bas. Ce problème est généralement résolu par des méthodes itératives en utilisant la régularisation Ces algorithmes sont très coûteux en termes de temps.

La normalisation de la taille des caractères est une étape de prétraitement très importante est peut être obligatoire dans certains cas où l'extraction des caractéristiques dépend de la taille de l'image ou aussi pour certains classifieurs qui prennent en entrée des vecteurs de taille précise.

II. Extraction des caractéristiques

L'extraction des caractéristiques est une étape clé dans un système de reconnaissance des caractères. C'est lors de cette étape que les caractères, constituant le texte à reconnaitre, prennent des représentations qui leur permettent de se distinguer par rapport aux autres, ces représentations sont les caractéristiques extraites, à partir de l'image du caractère, regroupées dans des vecteurs descripteurs. La performance du système de reconnaissance dépend largement de la pertinence des caractéristiques extraites. Nombreuses techniques ont été proposées dans la littérature qu'on peut les regrouper selon leur type local ou global, ou selon la technique d'extraction utilisée (statistique, structurelle, ou transformée globale).

En effet, la représentation des images des caractères joue l'un des rôles les plus importants dans un système de reconnaissance. Dans le cas le plus simple, des images des caractères en niveaux de gris ou binaires sont envoyées directement au classifieur pour décider la classe du caractère. Cependant, dans la plupart des systèmes de reconnaissance, afin d'éviter une complexité supplémentaire et augmenter la précision des algorithmes, une représentation plus compacte est requise [16]. À cette fin, un ensemble approprié de caractéristiques capables de bien représenter les caractères de l'alphabet en question est extrait.

L'extraction des caractéristiques est une étape primordiale qui vise à extraire, à partir de l'image brute du caractère, les caractéristiques les plus pertinents, représentatives et discriminantes pour la classification. De préférence et selon l'application, ces caractéristiques devraient être :

- **Discriminantes** : minimisent la variance intra-classe tout en maximisant la variance inter-classe.
- **De dimension raisonnable** : un grand nombre de caractéristiques permet de mieux représenter les différentes classes mais compliquera la phase de classification. Au contraire, un faible nombre accélérera la reconnaissance avec une séparation des classes moins fiable.
- **Invariantes** : à la rotation, translation et changement d'échelle, et de préférence au bruit et distorsion.

Nombreuses sont les techniques et les méthodes développées dans ce contexte, généralement, elles sont regrouper en trois principales catégories, à savoir, les caractéristiques statistiques, les caractéristiques structurelles et les transformations globales [154].

1. Caractéristiques statistiques

Les caractéristiques statistiques sont des mesures numériques dérivées des distributions statistiques des pixels et facilement calculées sur des images ou des régions d'images, Ces caractéristiques présentent une certaine tolérance au bruit, distorsions et variations des styles, et permettent la réduction de la dimension des données de l'image d'entrée en fournissant une

grande vitesse et une faible complexité. Cependant, ce type de caractéristiques ne permet pas la reconstruction de l'image [16].

Plusieurs techniques sont utilisées pour l'extraction de caractéristiques statistiques; certains d'entre eux sont : le zonage, les profiles des projections, les croisements et les distances.

1. 1. **Zonage**

La technique de zonage est l'une des techniques les plus populaires dans le domaine de la reconnaissance des caractères. C'est une technique très simple qui consiste à diviser le cadre englobant le caractère en plusieurs zones, chevauchées ou pas, pour extraire des informations locales relatives à ces zones [67]. Dans la littérature, le problème du zonage a été principalement lié à la conception de la topologie du zonage à utiliser, qui définit la manière dont une image de caractère doit être segmentée afin d'extraire autant d'informations discriminantes que possible [122]. Généralement, on trouve deux types de zonage : zonage statique et zonage dynamique.

Les zones dans les techniques statiques sont conçues sans utiliser des informations a priori sur la distribution topologique des caractères d'entrée. Dans ce cas, le zonage est effectué selon les expérimentations ou sur la base de l'intuition ou l'expérience du concepteur [67]. On trouve dans cette catégorie : le zonage par grille uniforme [155], le zonage en tranches [121] et le zonage hiérarchique [119] (Figure 19).

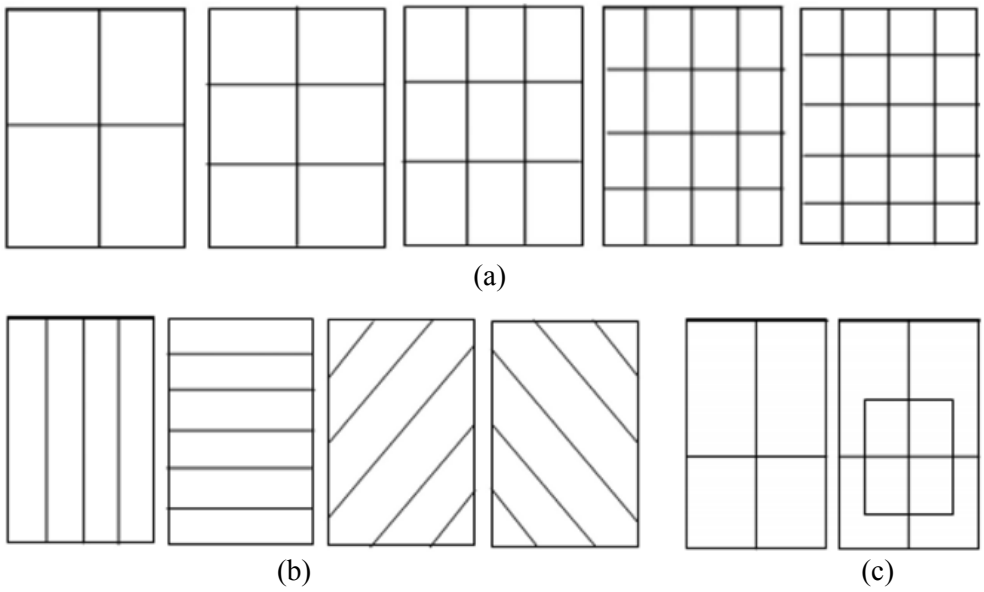


Figure 19. Zonage par les techniques statiques [67]

(a) Grilles uniformes

(b) Zonage en tranches

(c) Zonage hiérarchique

Les topologies dynamiques de zonage sont non uniformes et peuvent être obtenues manuellement en se basant sur la perception [50] ou automatiquement sur la base des résultats des procédures d'optimisation. Dans ce cas, une variété d'informations peut être utilisée pour concevoir la topologie la plus rentable pour un problème de classification spécifique. Dans les topologies automatiques, le choix des zones optimales peut être effectué selon leur capacité

discriminante [37] ou selon leur performance durant la classification, ces dernières se basent sur des gabarits [125], le maillage [159] et les diagrammes de Voronoi [68].

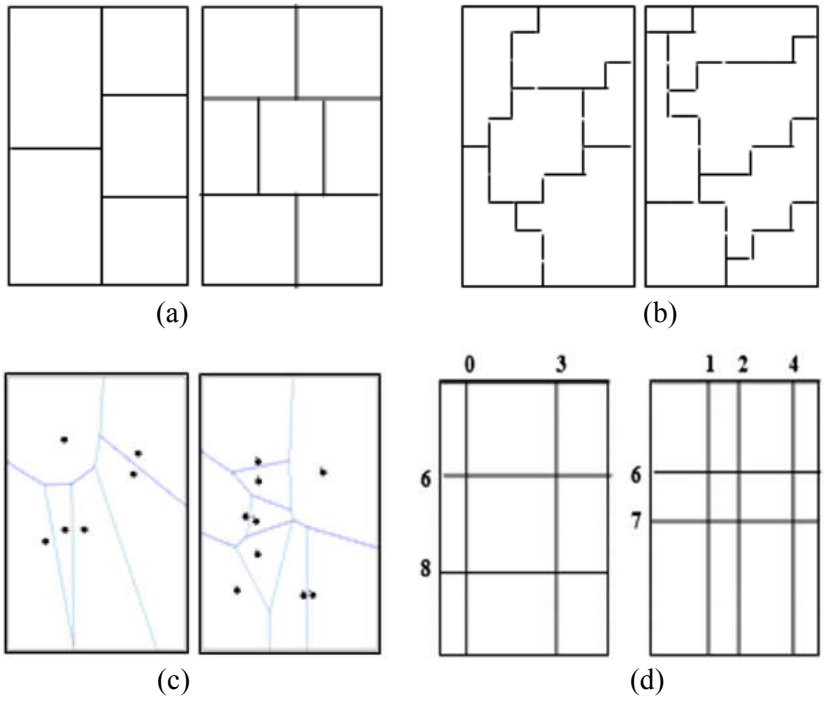


Figure 20. Exemples du zonage par les techniques dynamique [67]

(a) Technique basée sur la perception
(b) Technique basée sur la capacité discriminatives
(c) Technique basée sur les diagrammes de Voronoi
(d) Technique basée sur les gabarits

La Figure 20 montre quelques exemples des topologies du zonage obtenues en utilisant différentes techniques dynamiques.

Une fois les zones sont choisies, des caractéristiques telles que les densités, les directions des contours, et d'autres caractéristiques statistiques ou géométriques [67] peuvent être calculées dans chaque zone.

1. 2. Projections et profiles

Les histogrammes des projections et des profils externes suivant les différentes orientations (horizontale, verticale et les deux diagonales) sont souvent utilisées dans le domaine de la reconnaissance des formes. Elles fournissent une description statistique de la forme des caractères en créant une représentation unidimensionnelle à partir de l'image bidimensionnelle des caractères [155].

1. 2. 1. Les histogrammes des projections

Les histogrammes des projections sont obtenus en projetant les pixels constituant le caractère suivant les différentes directions (horizontale, verticale et les deux diagonales) et calculer le nombre de ces pixels dans chaque ligne suivant ces directions (Figure 21).

(a) (b) (c)

Figure 21. Histogrammes des projections horizontale et verticale de la lettre yey (ⵢ)

(a) Image du caractère yey(ⵢ)

(b) Histogramme de projection horizontale

(c) Histogramme de projection verticale

Les caractéristiques extraites des histogrammes des projections peuvent être tout simplement les valeurs obtenues par ces histogrammes ou bien des caractéristiques extraites à partir de ces derniers telles que les moments [167], les ondelettes [72], les dérivées des histogrammes [18], les caractéristiques de l'ombre [17], etc.

1. 2. 2. **Les histogrammes des profils externes**

D'autres caractéristiques liées aux histogrammes, appelées les profils externes, sont souvent utilisées, elles comptent le nombre de pixels (distance) entre la zone de délimitation de l'image du caractère et le bord du caractère (Figure 22). Les profils décrivent bien la forme externe des caractères [155].

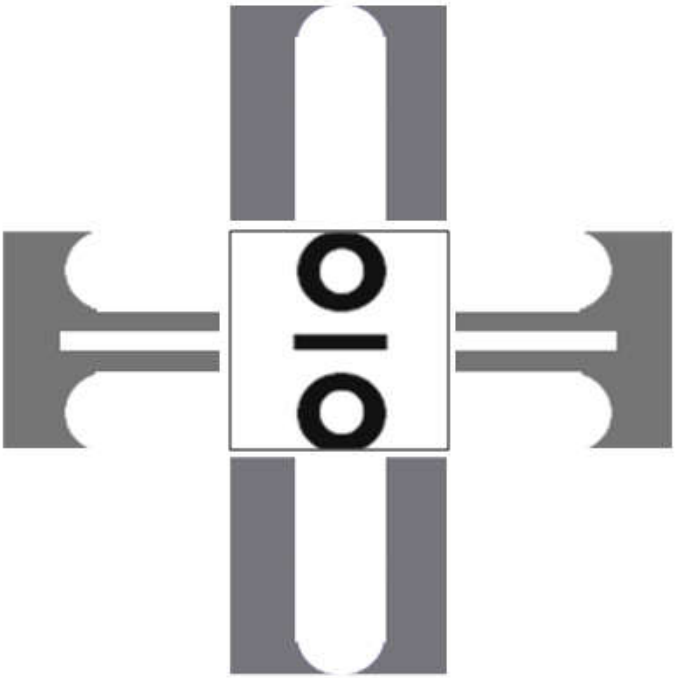


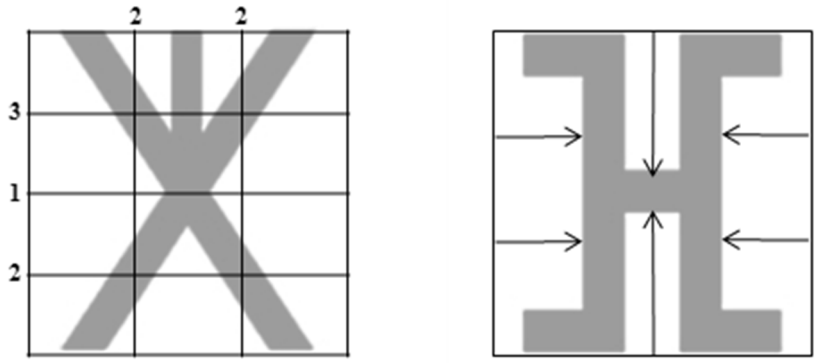
Figure 22. Histogrammes des profils externes du caractère yey (ⵢ)

Ces caractéristiques dépendent de la taille de l'image et sont sensibles à la rotation et au changement d'échelle.

1. 3. Croisements et distances

Les croisements et les distances sont des caractéristiques statistiques extraites à partir du caractère et peuvent être utilisées de différentes manières, on cite :

- Calcul du nombre des croisements du contour du caractère avec un ensemble de lignes dans des directions spécifiées (Figure 23.a);
- Calcul de certaines distances du contour du caractère par rapport aux frontières de la fenêtre englobant le caractère (Figure 23.b).



(a) (b)

Figure 23. Exemples des croisements et des distances pour les caractères Yax(ⵅ) et Yaf(ⵀ)

Cependant, ces caractéristiques seules ne produisent pas des systèmes de reconnaissance robustes. Lorsque le nombre de lignes utilisées augmente, les caractéristiques résultantes sont moins robustes aux variations de la polices pour les caractères imprimés et à la variabilité des formes et des styles d'écriture des caractères manuscrits [154].

2. Caractéristiques structurelles

Les caractéristiques structurelles décrivent les propriétés géométriques ou topologiques, globales ou locales, d'un caractère. Ces caractéristiques considèrent que les caractères sont constitués d'un ensemble de primitives (points, droites, concavités, convexités, boucles, extremas, intersections, etc.) qu'on peut les représenter de différentes manières : présence ou absence ; nombre d'apparitions, positions, directions, codage, etc [143].

Les méthodes structurelles fournissent une représentation de la structure et la forme des caractères et ont la capacité de gérer des informations vigoureusement déformées [16]. Certaines des techniques structurelles souvent utilisées pour la reconnaissance des caractères sont présentés dans ce qui suit.

2. 1. Propriétés géométriques et topologiques

Ces propriétés dépendent de l'alphabet du script à étudier. Pour les caractères Tifinagh, les caractères sont constituées de traits horizontaux, verticaux ou diagonaux, arcs, cercles, points

d'extrémité ou d'intersection, jonctions, etc (Figure 24). Leurs nombres, leurs positions et leurs longueurs en cas d'existence peuvent être utilisés comme des caractéristiques topologiques. Certaines quantités géométriques peuvent être aussi extraites telles que le rapport largeur/longueur de la fenêtre englobant le caractère, la circularité, la position du centre de gravité, l'axe dominante, la convexité, etc [89].

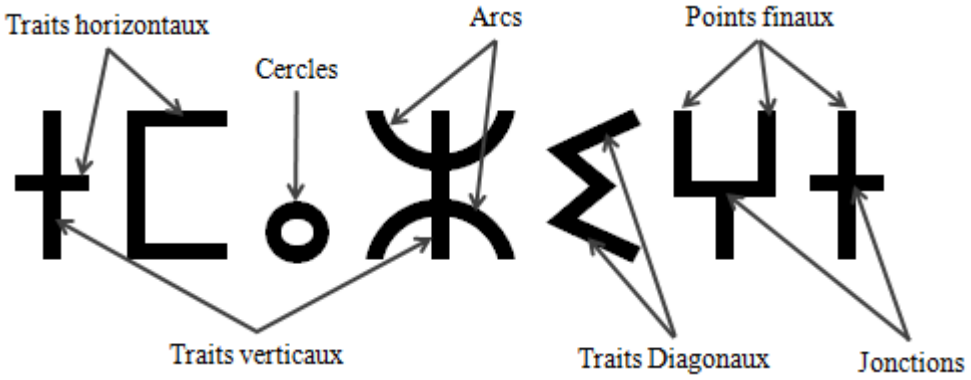
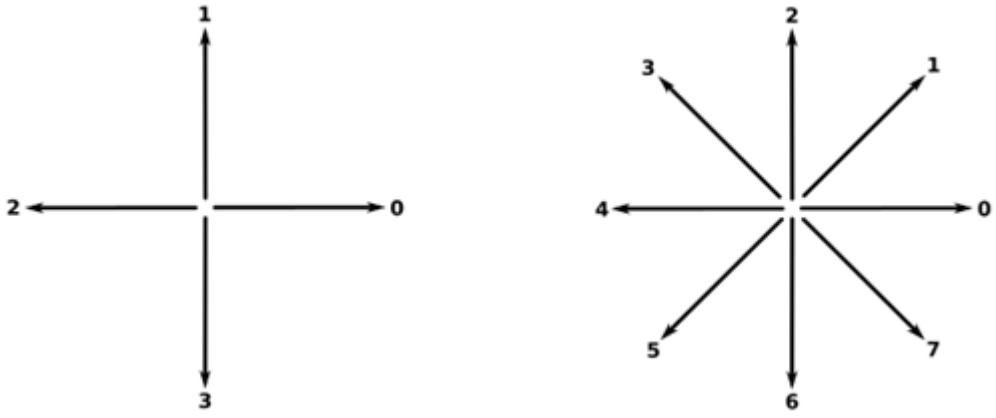


Figure 24. Primitives structurelles constituant les caractères Tifinagh

Un excellent survol des caractéristiques topologiques et structurelles couramment utilisées est présenté dans [160].

2. 2. **Codage en chaine des contours des caractères**

Une manière de représenter les contours ou les traits squelitisés des caractères est le codage de ces derniers en une chaine de codants (chaine-code). Ce codage est essentiellement obtenu en mappant les traits d'un caractère dans un espace de paramètres, à deux dimensions, constitué de codes [89]. En effet, La direction successive d'un pixel au pixel suivant devient un élément dans le code et tous ces éléments sont concaténés pour former une chaîne représentant le contour du caractère. Les directions utilisées sont basées sur le voisinage du pixel, la Figure 25 montre les deux voisinages couramment employées.



(a) (b)

Figure 25. Les voisinages couramment utilisés pour le codage en chaine

(a) Voisinage 4-connexions

(b) Voisinage 8- connexions

Freeman [49] affirme qu'en général, le codage fournit par ces techniques doit répondre à trois objectifs :

- préserver fidèlement les informations d'intérêt;
- permettre un stockage compact;
- faciliter tout traitement requis.

Pour comparer les codes en chaines obtenus, des distances métriques telles que la distance de Hamming ou la distance de Levenshtein sont utilisées.

Les chaine-codes fournissent une représentation efficace du contour d'un caractère et sont largement utilisées dans le domaine de la reconnaissance des formes. Il existe de nombreuses versions du codage en chaîne. Une description des codes les plus populaires est donnée dans ce qui suit.

2. 2. 1. **Code de Freeman**

La première approche pour représenter les contours en utilisant les codes en chaîne a été introduite par Freeman en 1961 [49]. Pour générer le code de Freeman, l'image du caractère est scannée de gauche à droite pour trouver le point de départ du contour, ensuite le contour est parcouru dans le sens contraire des aiguilles d'une montre jusqu'au pixel final. En suivant le contour, la direction de chaque segment est spécifiée et codée, en utilisant le schéma de numérotation de la Figure 25, pour former la chaine finale décrivant le caractère (Figure 26).

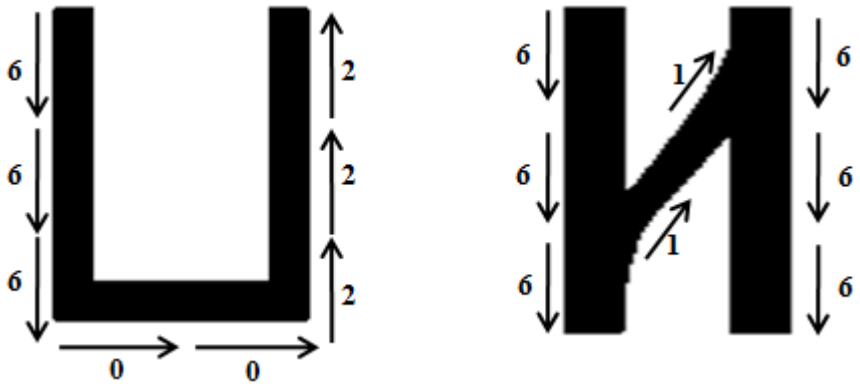


Figure 26. Génération du code de Freeman pour les caractères Yaw (Ш) et Yal (И)

Le code de Freeman est influencé par le choix du point de départ, invariant à la translation et sensible à la rotation et au changement de l'échelle [48].

2. 2. 2. **Code en chaine différentiel**

Le code en chaine différentiel est une extension du code de Freeman obtenu en calculant le nombre de transitions nécessaires, dans le sens inverse des aiguilles d'une montre, entre chaque deux entiers successifs dans la chaine initiale de Freeman, comme montré dans la Figure 27.

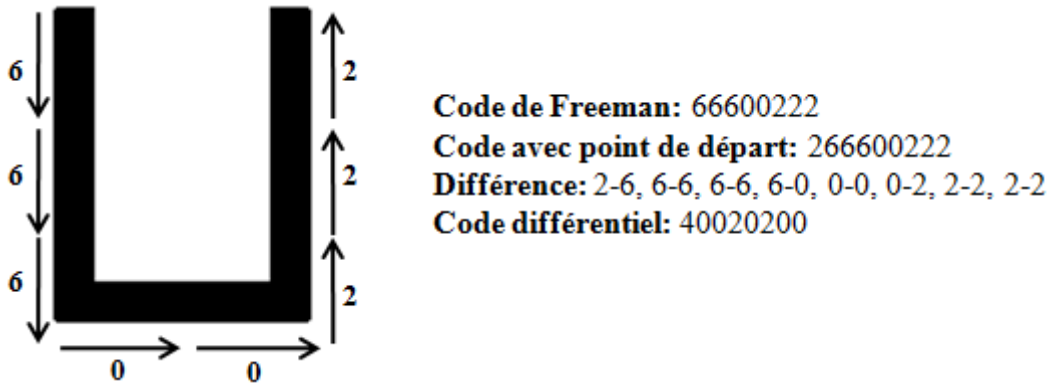


Figure 27.Génération du chaine-code différentiel pour le caractère Yaw (Ш)

L'avantage de ce code par rapport à celui de Freeman est qu'il est invariant à la rotation [48].

2. 2. 3. **Histogramme de chaine-code**

Une autre variété du code de Freeman est l'histogramme de ce code, cet histogramme est construit en comptant le nombre d'occurrence de chaque codant dans la chaine de Freeman (Figure 28). Ceci signifie que l'information sur l'ordre d'apparition des 8 directions est perdue. Cependant, au lieu de stocker toute la chaîne, seulement 8 valeurs de données sont stockées.

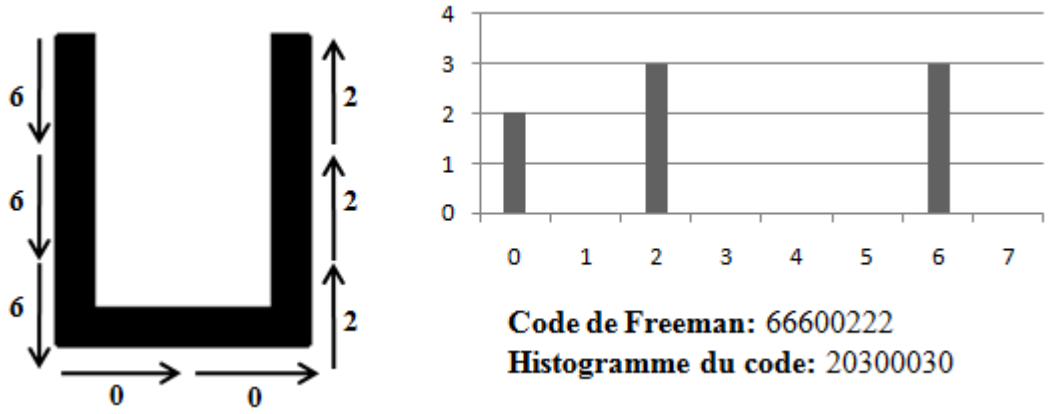


Figure 28. Histogramme de chaine-code pour le caractère yaw (Ш)

Ce code est simple, rapide et invariant à la translation, au changement d'échelle et à la rotation multiple de 90◦ [48].

2. 3. **Caractéristiques du gradient**

Le gradient est un vecteur comprenant la magnitude ainsi qu'une composante directionnelle calculée pour chaque pixel de l'image en appliquant les dérivées dans les deux orientations horizontale et verticale. L'opérateur du gradient génère un vecteur à chaque pixel de l'image de sorte qu'il pointe sur la direction de la plus grande augmentation d'intensité possible, et que sa magnitude correspond au taux de changement dans cette direction [6]. Le gradient peut être calculé par une convolution de l'image avec les deux opérateurs horizontal et vertical de Sobel,

de Robertz ou de Prewitt, Obtenant ainsi deux images Gx et Gy du gradient suivant les deux directions (Figure 29).

(a) (b) (c)

Figure 29. Résultats de la convolution par l'opérateur de Sobel [6]

(a) Image du caractère

(b) Image convoluée par le masque horizontal de Sobel

(c) Image convoluée par le masque vertical de Sobel

La magnitude du gradient et son orientation sont calculées à partir des images convoluées Gx et Gy par [6] :

$$Magnitude = |G(i,j)| = \sqrt{G_x(\mathrm{i,j})^2 + G_y(\mathrm{i,j})^2}$$

$$\theta(i,j) = \tan^{-1}\{G_x(\mathrm{i,j})/G_y(\mathrm{i,j})\}$$

Les histogrammes des gradients orientés (HOG) [36] sont les caractéristiques du gradient les plus populaires.

3. Les transformations globales

Elles sont basées sur une transformation globale de l'image. Généralement, elles convertissent la répartition des pixels du caractère en une forme plus compacte en utilisant les coefficients d'une combinaison linéaire d'une série de fonctions bien définies, fournissant ainsi un codage connu sous le nom d'expansion en séries. Les transformations communément utilisées incluent les moments, la transformée de Fourier, de Gabor et la transformée en ondelettes[89].

3. 1. Moments

Les techniques basées sur les moments ont été largement utilisées dans diverses applications de traitement d'images, en particulier en reconnaissance des formes, vu leur invariance par rapport à la translation, la rotation et le changement d'échelle. Les moments sont des mesures de la distribution des pixels autour du centre de gravité de la forme, ils ont été introduits par Hu [64] et étendus plus tard par Li [93].

Nombreuses familles de moments, introduites dans le passé, sont utilisées pour la reconnaissance automatique des caractères. On cite, entre autres, les moments de Hu [127], les moments de Zernike [5] et moments de Legendre [2, 74]. Certains travaux combinent différents types de moments pour la description des caractères [115, 148].

Les moments sont considérés comme des représentations d'expansion en séries car la forme originale peut être entièrement reconstruite à partir des coefficients des moments.

3. 2. Transformées de Fourier

La transformation de Fourier est largement utilisée pour l'analyse et la reconnaissance des formes. Les descripteurs de Fourier décrivent la forme en se basant sur ses contours [149], Ceci implique de trouver les coefficients de Fourier discrets a_k et b_k pour 0 ≤ k ≤ L-1, où L est le nombre total de points constituant le contour de la forme :

$$a_k = \frac{1}{L}\sum_{m=1}^{L} x_m exp^{-jk(2\pi/L)m}$$

$$b_k = \frac{1}{L}\sum_{m=1}^{L} y_m exp^{-jk(2\pi/L)m}$$

Où x_m and y_m sont les cordonnées x et y du $m^{ième}$ point du contour.

Les coefficients transformés de Fourier représentent la forme dans un domaine fréquentiel. Les fréquences inférieures contiennent des informations sur les caractéristiques générales de la forme, et les fréquences élevées contiennent des informations sur les détails plus fins de la forme. Bien que le nombre de coefficients générés par la transformation de Fourier soit généralement important, un sous-ensemble des coefficients est suffisant pour capturer les caractéristiques globales de la forme [89].

Des travaux utilisant la transformée de Fourier dans la reconnaissance automatique des caractères sont présentés dans [128, 136, 149].

3. 3. Transformée ou filtre de Gabor

Un filtre Gabor bidimensionnel est un filtre linéaire qui agit comme un filtre spatial passe-bande avec la possibilité de régler les orientations et les fréquences spatiales. Sa fonction de réponse impulsionnelle, également connue sous le nom de porteuse, est une sinusoïde complexe qui est modulée par une enveloppe gaussienne de forme elliptique. Son calcul dans le domaine spatial est donné par [53] :

$$g(x, y, f_0, \theta) = \frac{1}{2\pi\sigma_x\sigma_y} exp^{\left(\frac{x'^2}{2\sigma_x{}^2}+\frac{y'^2}{2\sigma_y{}^2}\right)}.exp^{i2\pi(u_0x+v_0y)}$$

Où
$x' = x.\cos\theta + y.\sin\theta$
$y' = -x.\sin\theta + y.\cos\theta$
$u_0 = f_0.\cos\theta$
$v_0 = f_0.\sin\theta$

où (x, y) sont les coordonnées spatiales, f_0 est la fréquence centrale (où la réponse du filtre est maximale), θ est l'orientation de l'onde plane sinusoïdale, x 'et y' sont les coordonnées de rotation, σ_x et σ_y sont la largeur ou la propagation de l'enveloppe gaussienne elliptique le long des axes x et y, (u_0, v_0) sont les fréquences spatiales centrales de l'onde sinusoïdale en coordonnées cartésiennes, (f_0, θ) sont leurs équivalents en coordonnées polaires : amplitude

fréquentielle ($f_0 = \sqrt{(u_0^2 + v_0^2)}$) ; et direction ($\theta = \tan^{-1}(u_0 / v_0)$). La sinusoïde complexe a une composante symétrique paire (partie réelle) et une composante antisymétrique impaire (partie imaginaire), Ce sont des fonctions distinctes qui existent indépendamment dans la partie réelle et imaginaire de la fonction sinusoïdale complexe.

$$g_{even}(x, y, f_0, \theta) = \frac{1}{2\pi\sigma_x\sigma_y} exp^{\left(\frac{x'^2}{2\sigma_x^2}+\frac{y'^2}{2\sigma_y^2}\right)}.\cos 2\pi f_0 x'$$

$$g_{odd}(x, y, f_0, \theta) = \frac{1}{2\pi\sigma_x\sigma_y} exp^{\left(\frac{x'^2}{2\sigma_x^2}+\frac{y'^2}{2\sigma_y^2}\right)}.\sin 2\pi f_0 x'$$

L'image d'entrée est convoluée avec chacun des filtres Pair et Impair de Gabor dans la banque de filtres constitués suivant les différentes fréquences et orientations (Figure 30). Ensuite, de nombreuses caractéristiques peuvent être obtenues à partir des images filtrées [158].

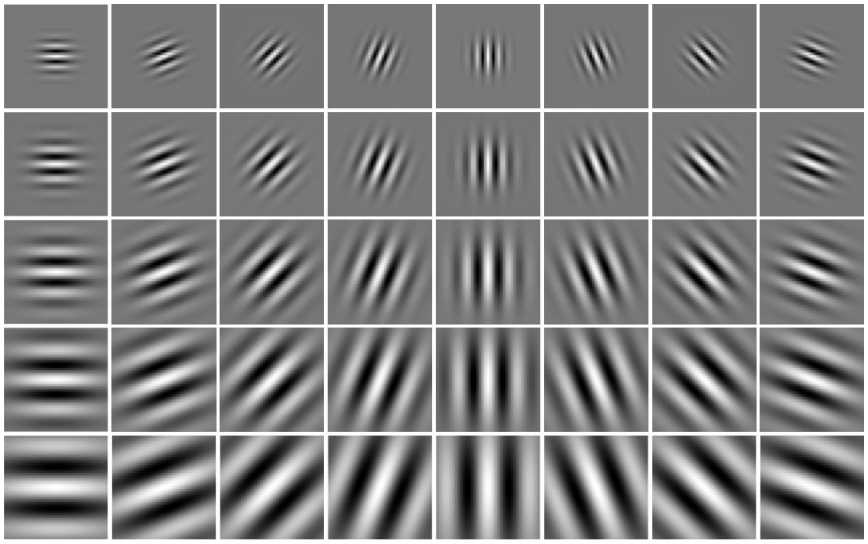


Figure 30. Banque des filtres de Gabor f=5 et θ =8

En tant que technique d'analyse de texture bien connue, le filtrage de Gabor a été appliqué dans les différents domaines tels que l'identification des scripteurs [141] et la classification des documents [41] et a atteint des performances exceptionnelles. Certains travaux utilisant le filtrage de Gabor pour la reconnaissance automatique des caractères sont présentés dans [38, 65, 84].

3. 4. **Transformée en ondelettes**

La transformée en ondelettes s'est avérée être un outil très utile pour de nombreuses applications de traitement d'images qui permet d'effectuer une analyse multi-résolution de l'image d'entrée. En effet, elle décompose l'image en un ensemble de sous-images de résolutions différentes, correspondant aux différentes bandes de fréquences. Cela entraîne une localisation de la fréquence spatiale utile pour extraire les caractéristiques pertinentes [16].

La transformée en ondelettes décompose l'image originale en quatre sous-bandes : l'image d'approximation et les trois sous-images comportant les détails suivant les directions horizontale,

verticale et diagonale [71]. Ce processus peut être itéré jusqu'à un nombre prédéfini de niveaux pour obtenir la représentation multi-résolution de l'image.

Afin d'extraire les caractéristiques, les coefficients d'ondelettes de la sous-bande d'approximation et les sous-bandes de détails (orientations) de tous les niveaux de décomposition sont utilisés pour formuler les caractéristiques d'énergie d'ondelettes [30].

D'autres nombreuses caractéristiques peuvent être extraites des coefficients d'ondelettes, elles incluent des valeurs statistiques telles que les moyennes et les écarts type, les propriétés des histogrammes et les caractéristiques de la matrice de cooccurrence [30].

Il existe une variété de familles d'ondelettes [79], on cite entre autres : Haar ; Daubechies ; Meyer ; Symlet ; etc.

3. 5. Autres transformées

D'autres transformées globales ont été aussi utilisées pour la reconnaissance automatique des caractères telles que Hough [153], Walsh [19], Radon [102], Karhunen Loeve Expansion [81].

III. Classification et reconnaissance des caractères

Après l'extraction des caractéristiques les plus représentatives, formant ainsi un vecteur descripteur, pour chaque image du caractère, une phase de classification est requise pour décider de la classe du caractère parmi l'ensemble des classes prédéfinies constituant l'alphabet de la langue traitée.

La classification supervisée est l'une des tâches les plus importantes dans les systèmes intelligents dont le but est de construire un modèle capable de séparer entre les différentes classes du problème en se basant sur un ensemble de données, appelé ensemble d'apprentissage, dont ses exemples ou instances sont définies sur un espace de caractéristiques bien déterminé. Ensuite, ce modèle doit être capable d'attribuer une nouvelle instance définie sur le même espace de caractéristiques à une classe parmi celles prédéfinies [85].

Dans les systèmes de reconnaissance des caractères, après l'extraction des caractéristiques les plus pertinentes à partir de l'image du caractère, ces caractéristiques sont regroupées dans un vecteur qui servira comme entrée pour la phase de classification pour décider de sa classe. Ce chapitre présente certains algorithmes largement utilisés, dans la littérature, ces algorithmes se basent sur différentes techniques qui peuvent être statistiques, structurelles, neuronales, etc. Ce chapitre présente aussi les démarches communément adoptées pour évaluer ces algorithmes et leur intégrer l'option du rejet.

1. Algorithmes de classification supervisée

1. 1. Méthodes statistiques

La théorie de la décision statistique considère un ensemble de critères d'optimalité qui maximise la probabilité qu'un caractère d'entrée appartient à une classe donnée parmi les classes

prédéfinies du problème. Les techniques statistiques reposent principalement sur trois hypothèses principales [16] :

- La distribution de l'ensemble de caractéristiques est gaussienne ou, dans les pires des cas, uniforme.
- Il y a suffisamment d'exemples disponibles pour chaque classe.
- À partir d'un ensemble d'images, on peut extraire un ensemble de caractéristiques qui représente distinctement chaque classe de caractères.

Dans les approches statistiques, chaque caractère est représenté en termes de caractéristiques ou de mesures f_i dans un espace de dimension n. Le but est de choisir les caractéristiques qui permettent aux vecteurs des caractères appartenant à différentes catégories d'occuper des régions compactes et disjointes dans l'espace de caractéristiques n-dimensionnel. Étant donné un ensemble de modèles d'entraînement de chaque classe, l'objectif est d'établir des limites de décision dans l'espace des caractéristiques qui séparent les caractères appartenant aux différentes classes, les limites de décision sont déterminées par des distributions de probabilité qui doivent être spécifiées ou apprises [69].

1. 1. 1. **Réseaux Bayésiens**

Les réseaux bayésiens RB reposent sur un formalisme basé sur les théories des probabilités et les graphes. Il représente un ensemble de variables aléatoires et leurs dépendances conditionnelles via un graphe acyclique orienté. Introduit par Pearl en 1985 [120], RB = (G, θ) est un réseau bayésien si

- $G = (X, E)$ est un graphe acyclique dirigé (DAG) où les sommets représentent un ensemble de variables aléatoires $X = \{X_1, X_2, ..., X_n\}$ et E représente l'ensemble des arcs modélisant les liens de causalité ou parenté entre les couples (X_i,X_j)
- $\theta_i = [P(X_i / Parents\ (X_i))]$ est la table des probabilités conditionnelles du nœud X_i connaissant l'état de ses parents.

Un réseau bayésien RB représente une distribution de probabilité sur X qui admet la distribution conjointe suivante :

$$P(X_1, X_2, ..., X_n) = \pi(P(X_i \mid Pa(X_i)))$$

Cette décomposition de la distribution conjointe permet d'avoir des puissants algorithmes d'inférence qui rendent les réseaux bayésiens très utiles pour modéliser et raisonner lorsque les situations sont incertaines ou si les données sont incomplètes. Ils sont aussi utiles pour les problèmes de classification où les interactions entre les différentes caractéristiques ou variables peuvent être modélisées par des relations de probabilités conditionnelles [109].

Il existe deux type d'approches d'inférence : exactes (inférence par énumération, algorithme d'élimination des variables et algorithme de regroupement) ; et approchées (Méthodes d'échantillonnage direct, pondération par la vraisemblance et inférence par simulation des chaines de Markov Monte-Carlo) [132]. En outre, deux problèmes s'imposent lors de l'utilisation des réseaux bayésiens : le choix de la structure du graphe ; et l'estimation de ses

distributions de probabilités. Concernant le premier problème, dans certains cas, la structure du réseau bayésien est fournie a priori par un expert, cependant, la détermination automatique de cette structure à partir de l'ensemble d'apprentissage est un problème NP-difficile, les chercheurs font recours aux méthodes évolutionnistes (Bayésien naïf, Hill Climbing, algorithme K2, recherche gloutonne, algorithme génétique, etc.). Pour le deuxième problème, on peut utiliser l'apprentissage par maximum de vraisemblance, l'estimation bayésienne, l'algorithme EM, etc [60]. L'avantage majeur des réseaux bayésiens et la possibilité de modéliser les connaissances a priori mais il est moins robuste par rapport aux autres classifieurs surtout si le nombre de variables augmente [85]

1. 1. 2. **Méthodes basées sur les instances**

Une approche naïve consiste à considérer que les points voisins ont une grande chance d'appartenir à une même classe. Alors, étant donné un caractère d'entrée inconnu, on peut décider que ce caractère appartient à la même classe de l'exemple le plus proche dans l'ensemble d'apprentissage. De manière plus générale, on peut considérer les k plus proches voisins du caractère d'entrée inconnu, et d'affecter celui-ci à la classe à laquelle appartient la majorité des k exemples les plus proches (Figure 31).

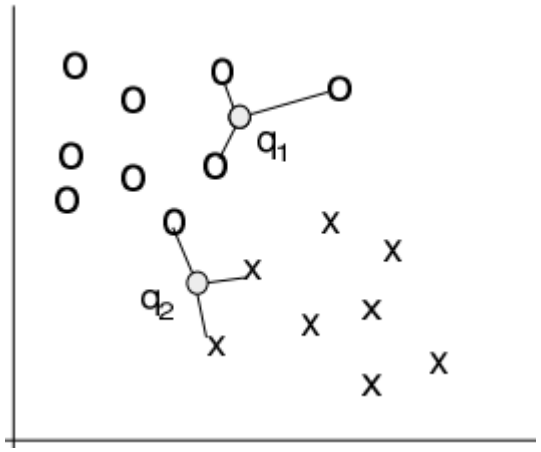


Figure 31. Exemple de classification binaire par les 3 plus proches voisins [35]

Cette approche est appelée méthode des k plus proches voisins (KNN). Le degré de voisinage est calculé en se basant sur une distance métrique qui détermine à quel point une nouvelle instance d'entrée est proche de chaque instance de l'ensemble d'apprentissage. De nombreuses fonctions calculant la distance entre deux instances sont proposées dans la littérature, les plus populaires sont : distance euclidienne ; distance de Minkowski ; distance de Manhattan et distance de Chebyshev [21, 35].

Cet algorithme, contrairement aux autres, ne nécessite pas une phase d'apprentissage et ne cherche pas à établir un modèle ou une fonction de classification pour classer une nouvelle instance, mais il la compare à une base d'exemples des données pré-classifiées, puis l'algorithme prédit la classe de sortie comme étant la classe des k-instances les plus proches. Le choix de nombre k des voisins à considérer influence les résultats de classification, pratiquement, on utilise la validation croisée pour estimer le taux des erreurs de classification pour différentes valeurs de k et on choisit celui qui conduit au plus bas taux des erreurs [55].

La méthode des k-plus proches voisins a montré sa puissance dans un grand nombre de domaines et sa performance dépend largement de la distance utilisée et du nombre k des voisins choisis.

1. 1. 3. **Modèles de Markov Cachés HMM**

Les modèles de Markov Cachés HMM sont parmi les techniques les plus utilisés dans le domaine de la reconnaissance des caractères. Ils sont définis comme un processus stochastique généré par deux mécanismes interdépendants [33] :

- Une chaîne de Markov ayant un nombre fini d'états et un ensemble de fonctions aléatoires, chacune étant associée à un état. La chaîne de Markov sert de représentation abstraite des contraintes structurelles, ces contraintes proviennent généralement de notre connaissance du problème et des données, ainsi que de la manière dont les données sont transformées en chaînes séquentielles.
- Les fonctions probabilistes de sortie incarnent le deuxième mécanisme stochastique et modélisent la variabilité inhérente des caractères ou de toute unité de base du langage que nous traitons.

Selon le type de données d'entrée, les éléments de sortie générés par état peuvent être discrets ou continus. Cette dernière représentation est mieux adaptée à la reconnaissance d'écriture manuscrite, étant donné que des vecteurs d'entrée à valeurs réelles provenant d'un espace de caractéristiques à grande dimension R^n sont traités. Par conséquent, les distributions de probabilité des sorties statistiques du modèle doivent pouvoir définir des distributions continues sur R^n. Ces distributions de probabilité sont habituellement approchées par des mélanges de gaussiennes spécifiques à l'état [124].

Les systèmes de reconnaissance basés sur les modèles de Markov utilisent deux composants de modélisation distincts. Dans le premier composant, l'apparence de l'écriture, définie par ses caractéristiques locales capturées, est décrite par des modèles de Markov cachés. Ce composant définit généralement des modèles de base pour des unités élémentaires telles que des caractères. Le deuxième composant de modélisation décrit des restrictions de séquençage à long terme, c'est-à-dire au niveau des séquences de caractères (mots) ou de mots (phrases). Ce modèle est réalisé par des modèles de chaines de Markov [124].

Les HMM nécessitent que les données à analyser soient séquentiellement ou temporellement ordonnées. Par conséquent, une technique appropriée pour convertir des images bidimensionnelles en une représentation séquentielle est d'adopter l'approche de la fenêtre glissante qui consiste à faire glisser une petite bande d'analyse, qui n'est généralement que de quelques pixels de largeur (c'est-à-dire, beaucoup plus étroite qu'une image de caractère). Ainsi, en se basant sur ces petites bandes verticales, des caractéristiques statistiques ou géométriques sont extraites pour représenter localement l'image [75].

Bien que certaines publications concernant la reconnaissance fondée sur un modèle de Markov pour des caractères isolés existent, l'approche HMM montre sa force en particulier pour les séquences de caractères ou de mots [124].

1. 2. Réseaux de neurones artificiels

Les réseaux de neurones artificiels (RNA) ont été initialement étudiés dans l'espoir de faire des machines intelligentes de perception en simulant la structure physique des cerveaux humains [33]. Ils peuvent être considérés comme des systèmes de calcul massivement parallèles constitués d'un très grand nombre de processeurs simples avec de nombreuses interconnexions. Les modèles de réseaux neuronaux tentent d'utiliser certains principes organisationnels (apprentissage, généralisation, adaptabilité, représentation distribuée, etc.) dans un réseau de graphes orientés pondérés dans lesquels les nœuds sont des neurones artificiels organisés en couches et les arrêtes dirigées (avec des poids) sont les connexions entre les sorties de neurones de chaque couche et les entrées de neurones de la couche suivante. Les réseaux de neurones ont la capacité d'apprendre des relations non linéaires complexes entre les entrées et les sorties du problème, d'utiliser des procédures séquentielles d'apprentissage et de s'adapter aux données [69].

Les familles de réseaux neuronaux les plus couramment utilisées pour les tâches de reconnaissance des formes sont le perceptron multicouche (PMC) et les réseaux de neurones convolutifs (CNN).

1. 2. 1. Perceptron multicouches PMC

L'inspiration originale derrière cette technique vient des réseaux bioélectriques du cerveau humain formé par les neurones et leurs synapses. De même, un modèle neuronal de base appelé perceptron possède un ensemble de connexions pondérées (semblables aux synapses dans les neurones biologiques), une unité de sommation et une fonction d'activation comme le montre la Figure 32.

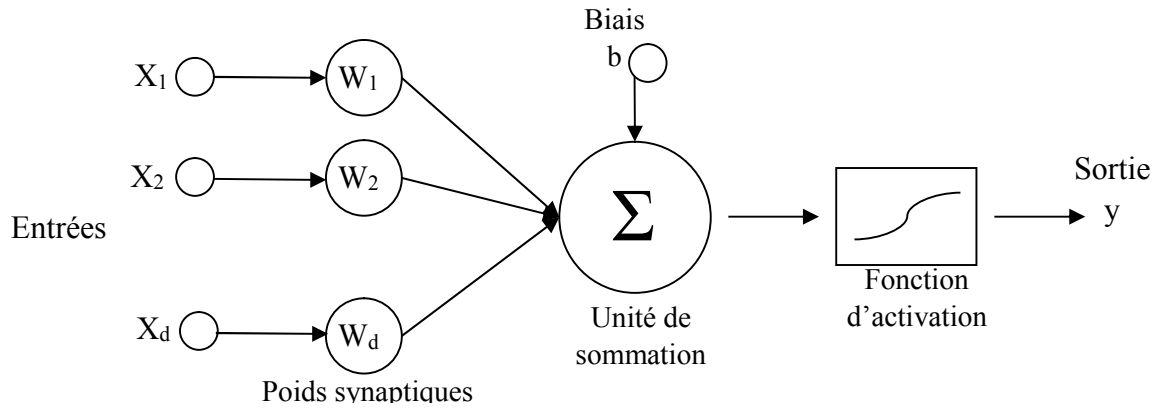


Figure 32. Modèle et fonctionnement d'un neurone [33]

La sortie de l'unité de sommation et une combinaison linéaire des entrées dont les coefficients sont les poids. La fonction d'activation peut être linéaire ou non-linéaire, la fonction d'activation sigmoïde est la plus utilisée [33].

Le PMC est un réseau de neurones statique qui comporte trois types de couches : la couche d'entrée qui présente les données au réseau, la couche de sortie qui sert à la décision, et en fin, les couches cachées qui effectuent le traitement (Figure 33).

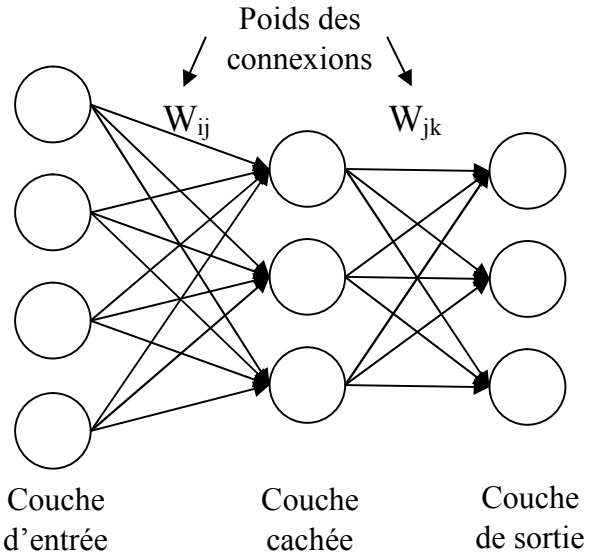


Figure 33. Exemple d'un perceptron à une seule couche cachée [85]

L'apprentissage supervisé de ces réseaux se fait via l'algorithme de rétro-propagation du gradient [156], qui opère en quatre étapes :

- **Etape 1** : initialisation arbitraire des poids ;
- **Etape 2** : propagation avant d'un exemple de l'apprentissage et calcul de sa sortie;
- **Etape 3** : propagation arrière de l'exemple en fonction de sa sortie désirée;
- **Etape 4** : mise à jour des poids de connexion.

Les trois dernières étapes s'effectuent jusqu'à ce que l'erreur sur les neurones de sorties devient suffisamment petite ou après un certain nombre d'itérations déterminé à l'avance.

Le PMC peut avoir n'importe quel nombre de couches cachées et n'importe quel nombre de neurones par couche cachée, néanmoins, l'utilisation d'une seule couche cachée est suffisante pour résoudre un problème complexe non linéaire. Le choix du nombre de couches cachées et du nombre de neurones de chaque couche est toujours un défi [76]. Dans la majorité des cas, l'architecture du PMC est déterminée expérimentalement.

1. 2. 2. **Réseaux de neurones convolutifs**

Les réseaux de neurones convolutifs CNN ont été proposés dans les années 80 comme des algorithmes de traitement d'images non linéaires inspirés du cortex visuel des mammifères qui reposent principalement sur les champs réceptifs locaux. Récemment, ils ont été largement adoptés par les communautés de vision par ordinateur et de reconnaissance des formes [91]. Les CNN possèdent plusieurs avantages par rapport aux RNA conventionnels :

- La typologie spatiale de l'entrée est bien représentée par la structure du réseau via les champs réceptifs locaux;

- L'étape d'extraction des caractéristiques est intégrée à l'étape de classification, et les deux sont générées par le processus d'apprentissage ;
- Le concept de partage des poids réduit le nombre de paramètres entraînables du réseau, ce qui réduit la complexité du réseau et améliore la généralisation ;

Ces réseaux sont organisés en plusieurs couches (Figure 34), à savoir : les couches convolutives et ReLU; les couches de pooling et enfin les couches entièrement connectées [92].

- Couches convolutives : L'objectif principal de ces couches est l'extraction, à partir de l'image d'entrée, des caractéristiques telles que des traits, contours, points d'extrémité, couleur simple, etc. La convolution préserve la relation spatiale entre les pixels en apprenant les caractéristiques de l'image à l'aide de petits carrés de données d'entrée. Une couche de convolution contient généralement plusieurs filtres différents de convolution dont la taille doit être spécifié à l'avance, cette couche produira ainsi plusieurs images convoluées appelées cartes des caractéristiques. La taille de ces dernières dépend du pas de convolution utilisé et de l'ajout ou non des pixels de bourrage avant convolution.
- Couches ReLU : Il est conventionnel d'appliquer immédiatement, après chaque couche de convolution, une couche non linéaire (ou couche d'activation) appelée ReLU. Le but de cette couche est d'augmenter les propriétés non linéaires du modèle et de l'ensemble du réseau tout en accélérant l'apprentissage [61]. La couche ReLU change toutes les valeurs négative du volume d'entrée (carte des caractéristiques) par 0 en appliquant, tout simplement, la fonction f(x) = Max (0, x).

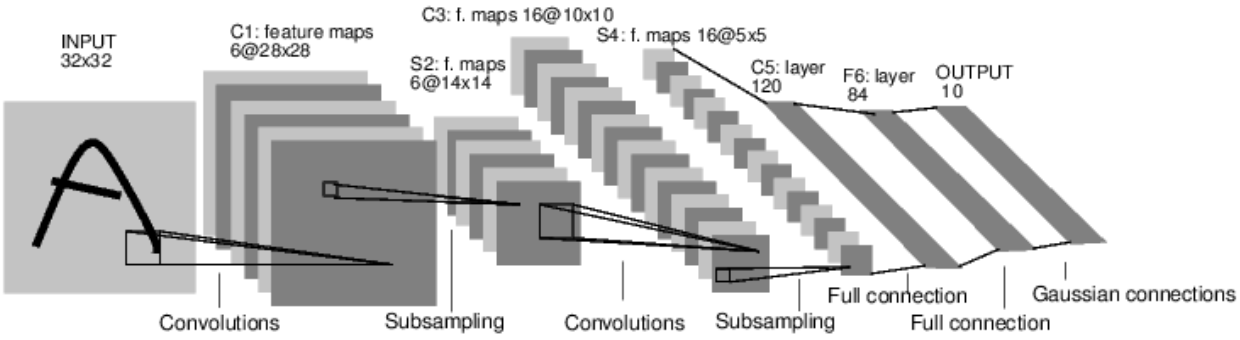


Figure 34. Architecture du réseau de neurones convolutifs LeNet5 [92]

- Couches de pooling : Les couches de pooling ou de regroupement prennent des petits blocs rectangulaires de neurones des cartes de caractéristiques issues des couches convolutives et les sous-échantillonnent pour produire un neurone de sortie unique à partir de chaque bloc. L'objectif ici et de réduire la taille des cartes de caractéristiques produites par la convolution en ne retenant que l'information importante évitant ainsi les petites variations et distorsions. Il y a plusieurs façons de faire le pooling, comme prendre la moyenne ou le maximum, ou une combinaison linéaire apprise des neurones dans le bloc.
- Couches entièrement connectées : Finalement, après plusieurs couches de convolution et de pooling qui permettaient d'extraire les caractéristiques à partir de l'image d'entrée, le raisonnement du haut niveau dans le réseau neuronal, à savoir la classification, se fait via

des couches entièrement connectées. Ces couches agissent comme un perceptron multicouche PMC qui utilise les caractéristiques extraites à partir des couches précédentes pour classifier l'image d'entrée parmi différentes classes en se basant sur un ensemble d'apprentissage.

Le processus d'apprentissage du CNN est effectué par la rétro-propagation du gradient. Après l'initialisation aléatoire des poids et des filtres de convolution, pour chaque image de l'ensemble d'apprentissage, le CNN calcule la sortie par une propagation avant et déduit l'erreur totale pour effectuer une propagation arrière qui permettra de mettre à jour les poids des connexions mais aussi les valeurs des filtres de convolution jusqu'à ce que l'erreur totale soit minimale et les poids et les filtres de convolution soient optimisés.

Depuis les années 90s, plusieurs architectures des CNN ont été développées dans la littérature pour la classification des images, les plus populaires et les plus influençantes sont LeNet [92], AlexNet [87], ZF Net [165], GoogLeNet [150] et VGGNet [144].

1. 3. **Machine à vecteurs supports SVM**

Actuellement, les machines à vecteurs de supports (SVM) sont parmi les approches les plus utilisées pour la reconnaissance des formes. Trois propriétés les rendent séduisantes [132] :

- Les SVM construisent un séparateur à marge maximale qui est une frontière de décision avec la plus grande distance possible entre les points d'apprentissage ; cela leurs permet de bien généraliser (Figure 35).

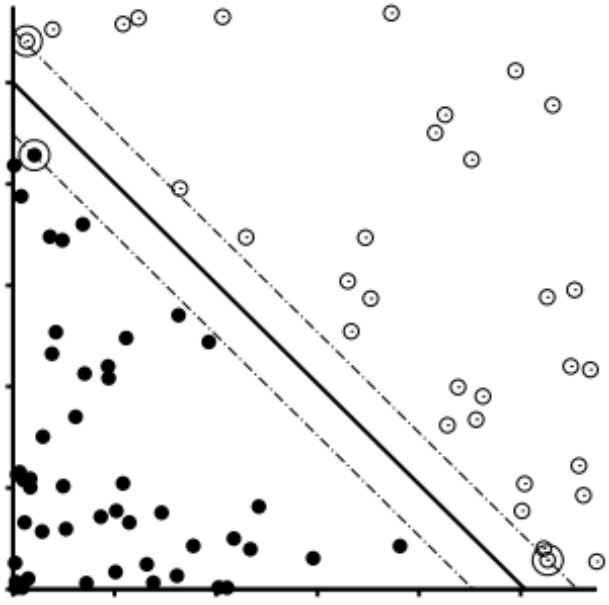
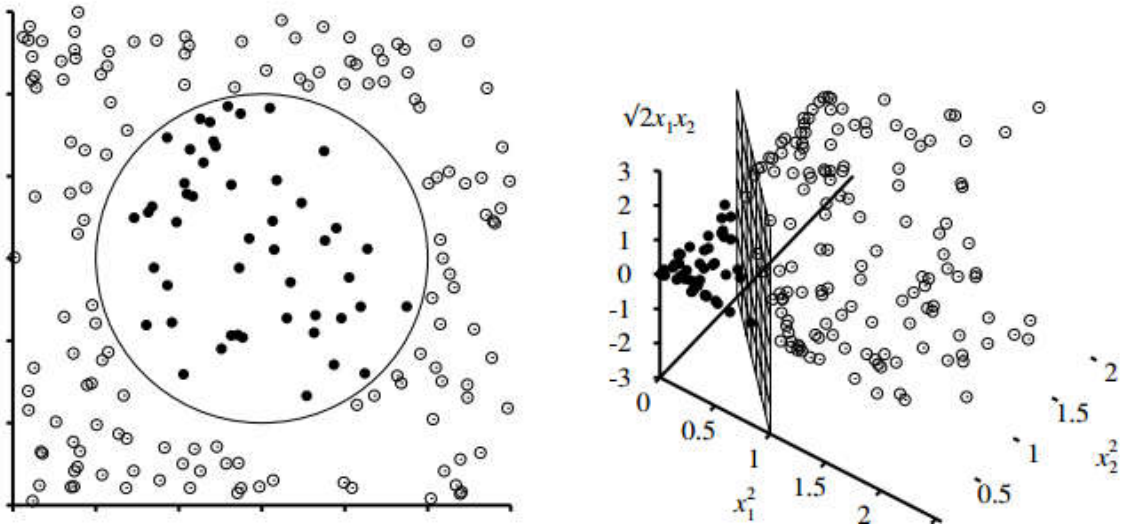


Figure 35. Séparateur à marges dans une classification binaire [132]

- Les SVM sont des méthodes non paramétriques capables de représenter les fonctions complexes et sont résistantes aux problèmes de sur-apprentissage.

- Même si les données ne sont pas linéairement séparables, les SVM explosent les données dans un espace de dimensionnalité supérieur en utilisant des fonctions noyaux pour trouver un séparateur linéaire (Figure 36).



(a) (b)

Figure 36. Problème de classification des données non-linéairement séparables [132]
(a) Exemple des données deux dimensions non-linéairement séparables
(b) Séparation linéaire après explosion des données en trois dimensions

Soit $A=(x^1, y^1),\ldots,(x^N, y^N)$ un ensemble d'apprentissage issu d'un problème de classification binaire tel que les exemples des deux classes sont soit étiquetés par +1 soit par -1. Les poids ω qui définissent l'hyperplan séparateur des SVM dans l'espace φ sont donnés par [132] :

$$\omega = \sum_{j=1}^{N} \alpha_j y^j \varphi(x^j)$$

Tel que les α_j sont les coordonnées de la solution du problème dual suivant :

$$(D) \quad \begin{cases} Min \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y^i y^j \varphi(x^i) \varphi(x^j) \\ SC: \\ \sum_{i=1}^{n} \alpha_i y^i = 0 \\ \alpha_i > 0 \end{cases}$$

La classe d'un nouvel objet x est estimée par l'équation :

$$y = signe(\omega.\varphi(x))$$

Comme il est difficile de connaitre l'espace des caractéristique, certains chercheurs ont proposé de remplacer le terme $\varphi(x^i)\varphi(x^j)$ par $K(x^i, x^j)$ où K est une fonction suffisamment

complexe pour pouvoir séparer linéairement les données, ce genre de fonctions sont appelées fonctions noyaux [132], une variétés de fonctions noyaux sont disponibles dans la littérature, on cite entre autres : fonction linéaire, fonction gaussienne RBF, fonction polynomiale, fonction sigmoïde [62].

À l'origine, les SVM ont été conçues pour les problèmes de classification binaire. Cependant, il existe deux approches pour adapter les SVM à des problèmes multi-classes : la première consiste à considérer toutes les classes dans la formulation du problème d'optimisation ; la deuxième approche vise à combiner un ensemble de classifieurs binaires [63].

1. 4. **Méthodes structurelles**

Les méthodes de reconnaissance dites structurelles sont plus souvent utilisées dans la reconnaissance de caractères en ligne que hors ligne. Contrairement aux méthodes statistiques et aux réseaux de neurones qui représentent le modèle de caractère en tant que vecteur de caractéristiques de dimensionnalité fixe, les méthodes structurelles représentent un modèle en tant que structure (chaîne, arbre ou graphique) de taille flexible. Chaque classe est représentée comme un ou plusieurs modèles basés sur les primitives structurelles du caractère, le modèle du caractère d'entrée à reconnaitre est apparié avec les modèles prédéfinis en utilisant des mesures de similarité pour décider de sa classe de sortie [33].

Les méthodes structurelles font face à deux difficultés majeures : la détection et l'extraction de primitives structurelles à partir des caractères d'entrée ; et l'apprentissage de modèles à partir des échantillons [16]. Les méthodes syntaxiques et les méthodes graphiques sont les plus utilisées dans ce contexte. Dans les méthodes syntaxiques, l'entraînement se fait en décrivant chaque caractère par une grammaire G_i faisant recours à des outils tels que la théorie de langage, et la reconnaissance d'une nouvelle entrée consiste à l'analyser syntaxiquement afin de décider à quelle grammaire il appartient. Tandis que pour les méthodes graphiques, les caractères sont représentés par des arbres ou des graphes où les nœuds peuvent être des points d'inflexion, des points de branchement ou des points finaux, et les arcs ou arrêtes sont les squelettes des traits des caractères. Pour la phase de reconnaissance, des techniques d'appariement des graphes et d'étiquetage par relaxation probabiliste sont utilisées pour décider la classe d'une nouvelle entrée [33].

1. 5. **Les arbres décisionnels**

Les arbres décisionnels sont l'une des méthodes les plus populaires dans l'apprentissage supervisé. Un arbre décisionnel est un arbre dans le sens du mot en informatique. Les nœuds internes sont appelés nœuds de décision où chaque nœud est étiqueté par un test. En général, chaque test examine la valeur d'un seul attribut dans l'espace des caractéristiques. Les réponses possibles au test correspondent aux branches de sortie de ce nœud. Tandis que, chaque feuille est étiquetée par une classe parmi les classes prédéfinies du problème considéré. Chaque nœud interne ou feuille est identifié par sa position depuis la racine, i.e. la liste des branches qui permettent d'y accéder depuis la racine. L'objet a classifié, défini dans l'espace des caractéristiques, suit un chemin à base des tests subits, depuis la racine, sur l'ensemble de ses caractéristiques jusqu'à l'atteinte d'une feuille qui décide de sa classe [27].

La construction de arbres décisionnel à partir de données d'apprentissage passe par deux étapes : l'étape de croissance ou de construction de l'arbre ; et l'étape d'élagage. L'arbre est construit dans la première étape de manière descendante en divisant récursivement l'ensemble d'apprentissage sur la base de critères optimaux locaux jusqu'à ce que la totalité ou la plupart des instances appartiennent à chacune des partitions portant la même étiquette de classe. L'élagage d'arbre est fait de manière ascendante dans une deuxième étape, Il est utilisé pour améliorer la précision de la prédiction et de la classification de l'algorithme en évitant le sur-apprentissage. Cette étape généralise l'arbre en supprimant le bruit et les valeurs aberrantes [80].

En principe, le nombre d'arbres décisionnels qu'on peut construire, à partir d'un ensemble donné de caractéristiques, pour un problème donné est exponentiel, la recherche de l'arbre optimal est infaisable en raison de la complexité de l'espace de recherche. Néanmoins, des algorithmes efficaces ont été développés pour induire un arbre décisionnel raisonnablement précis dans un temps assez rapide. Ces algorithmes utilisent généralement une stratégie gloutonne qui développe un arbre de décision en prenant une série de décisions localement optimales concernant l'attribut à utiliser pour le partitionnement des données. L'algorithme de Hunt est l'un des algorithmes les plus anciens dans ce cadre qui constitue la base de nombreux algorithmes d'induction des arbres décisionnels existants, y compris ID3, C4.5 et CART [80].

Les forêts d'arbres décisionnels ou forêts aléatoires, introduites en 1995 par Ho [152], visent à augmenter la précision de la généralisation en combinant la décision de plusieurs arbres décisionnels construits sur des sous-espaces choisis au hasard dans l'espace des caractéristiques afin de limiter la complexité et éviter le sur-apprentissage. Les forêts aléatoires ont montré leur puissance dans des différentes applications de l'apprentissage automatique [54].

1. 6. **Combinaison des classifieurs**

La combinaison de plusieurs classifieurs a été longtemps utilisée pour améliorer la précision de la classification, ceci est généralement accompli en combinant les décisions (sorties) des différents classifieurs utilisés. C'est un domaine de recherche très actif depuis les années 90 et différentes méthodes ont été proposées pour réaliser cette combinaison, Liu et al [94] ont classé ces méthodes en deux classes : méthodes parallèles et séquentielles. Les méthodes parallèles (horizontales) sont souvent adoptées pour augmenter la précision, tandis que les méthodes séquentielles (verticales ou en cascade) sont sollicitées pour les problèmes dont le nombre de classes de sorite est assez important.

Dans cette thèse, nous étions intéressés par les méthodes parallèles, et spécialement par les méthodes de vote par majorité et ses variantes en raison de leur simplicité et leur très haut niveau de précision et de robustesse [126]. En effet, selon le type d'information de sortie des classifieurs, différentes méthodes de fusion des décisions peuvent être utilisées. Si le classifieur produit seulement des étiquettes en sortie, un vote majoritaire est effectué. Dans le cas où on a des sorties continues telles que des probabilités, une moyenne ou d'autres combinaisons linéaires de ces probabilités sont considérées [82].

En effet, supposons que l'on a R classifieurs à combiner pour décider dans un problème de classification à m classes, les différentes règles de combinaison, à savoir, le vote majoritaire, la

moyenne des probabilités de sortie, leur produit et leur maximum sont formulées mathématiquement comme suit [82] :

- Vote majoritaire

Assigner X_i à la classe ω_j si :

$$\sum_{i=1}^{R} \Delta_{ji} = \max_{k=1,..,m} \sum_{i=1}^{R} \Delta_{ki}$$

Où

$$\Delta_{ki} = \begin{cases} 1 & \text{si } P(w_k \mid x_i) = \max\limits_{j=1,...,m} P(w_j \mid x_i) \\ 0 & \text{sinon} \end{cases}$$

- Moyenne des probabilités

Assigner X_i à la classe ω_j si :

$$\frac{1}{R} \sum_{i=1}^{R} P(w_j \mid x_i) = \max_{k=1,...,m} \frac{1}{R} \sum_{i=1}^{R} P(w_k \mid x_i)$$

- Produit des probabilités

Assigner X_i à la classe ω_j si :

$$P(w_j) \prod_{i=1}^{R} P(x_i \mid w_j) = \max_{k=1,..,m} P(w_k) \prod_{i=1}^{R} P(x_i \mid w_k)$$

- Maximum des probabilités

Assigner X_i à la classe ω_j si :

$$\max_{i=1,..,R} P(w_j \mid x_i) = \max_{k=1,..,m} \max_{i=1,..,R} P(w_k \mid x_i)$$

2. Évaluation des classifieurs

Il existe généralement deux manières pour évaluer la performance des classifieurs selon la taille des données en main. La première, en cas de données en masse, consiste à diviser l'ensemble de données en deux parties, les 2/3 servent pour l'apprentissage et le reste (1/3) pour le test. La deuxième manière, appelée k-validation croisée, est souvent utilisée en cas de données de petite taille, elle consiste à diviser l'ensemble de données original en k parties, Une partie est réservée au test et les k-1 parties restantes effectuent l'apprentissage. L'opération se répète k fois de telle sorte que chaque partie a servi exactement une fois comme ensemble de test. Ainsi, La performance du classifieur est la moyenne des performances obtenues dans les k exécutions [83].

Les mesures de la qualité des classifieurs sont construites à partir de la matrice de confusion qui reporte les prédictions correctes et incorrectes des exemples de chaque classe. Table 1 présente la matrice de confusion pour une classification binaire et quelques mesures de performances communément utilisées (précision, recall, performance, etc.) [146].

Table 1. Matrice de confusion et mesures de performances utilisées dans un problème de classification binaire

			Données prédites		
			Positif	**Négatif**	**Mesures**
Données observées		**Positif**	True Positive (TP)	False Negative (FN)	**Sensibilité ou Recall**
		Négatif	False Positive (FP)	True Negative (TN)	**Spécificité**
		Mesures	**Précision**	**Prédiction négative (NPV)**	**Performance (Accuracy)**

Où :

- TP (True Positive) : le nombre d'instances de la classe 1 correctement classifiées ;
- FN (False Negative) : le nombre d'instances de la classe 1 incorrectement classifiées ;
- FP (False Positive) : le nombre d'instances de la classe 2 incorrectement classifiées ;
- TN (True Negative) : le nombre d'instances de la classe 2 correctement classifiées ;
- La performance est la mesure la plus utilisée pour évaluer la performance globale des classifieurs :

$$Performane = \frac{TP + TN}{TP + FP + TN + FN}$$

- La précision évalue la puissance prédictive des classifieurs :

$$Précision = PPV = \frac{TP}{TP + FP}$$

- La sensibilité/spécificité évalue l'efficacité des classifieurs sur une seule classe en estimant la probabilité que les prédictions positives/négatives soient vraies :

$$Recall = Sensibilité = \frac{TP}{TP + FN}$$

$$Spécificité = \frac{TN}{TN + FP}$$

- La F-Mesure est une mesure composite combine la précision et le recall privilégiant une par rapport à l'autre en paramétrant le réel β> 0. Une valeur inférieure à 1 favorise le recall et vice versa.

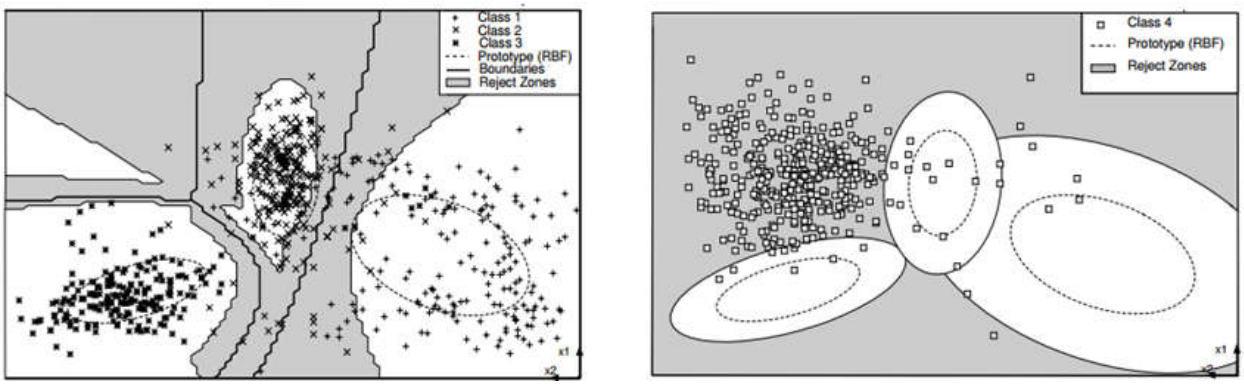
$$F - Mesure = \frac{(\beta^2 + 1) * Precision * Recall}{\beta^2 * Precision + Recall}$$

3. Option de rejet

L'idée générale derrière l'intégration de l'option du rejet pour les classifieurs est de les rendre capables de ne pas répondre et rejeter le caractère d'entrée afin d'éviter une erreur de classification qui est causée principalement par [105] :

- Une confusion de classifieur pour décider la classe de sortie entre deux ou trois classes ;
- L'entrée correspond à un caractère ou une forme non rencontrée lors de l'apprentissage.

Ces deux types d'erreurs ont conduit à deux types de rejet utilisés dans des contextes différents : le rejet d'ambiguïté et le rejet de distance [106] (Figure 37).



(a) (b)

Figure 37. Illustration des deux types du rejet pour un problème de classification

(a) Rejet d'ambiguïté

(b) Rejet de distance

Le rejet d'ambiguïté consiste à rejeter l'entrée lorsque le classifieur a des fortes chances de commettre une erreur vue que l'entrée est sur les frontières de décision et crée une confusion pour le classifieur en activant de façon équivalente au moins deux classes (Figure 37.a). Pour le rejet de distance, il consiste à délimiter les connaissances du classifieur en rejetant toute nouvelle forme d'entrée non vue lors de l'apprentissage, ce type de rejet nécessite l'existence d'une base de données des contre-exemples (Figure 37.b).

Dans cette thèse, dans l'absence d'une base des contre-exemples et afin d'améliorer la précision des classifieurs, nous nous sommes focalisées sur le rejet d'ambiguïté et spécialement les architectures basées sur l'utilisation des seuils. En effet, Chow [34] utilise les probabilités a posteriori de sortie des classifieurs pour traiter le rejet, selon lui, un caractère est rejeté si sa meilleure probabilité d'appartenance à une classe i est inférieur à un certain seuil T e [0,1] :

$$\max_{k=1,\dots,N} P(w_k \mid x) = P(w_i \mid x) < T$$

L'objectif ici et de rejeter les caractères les plus susceptibles d'être mal classés. Cependant, les classifications correctes peuvent également être rejetées. D'après la règle de Chow, un seuil bas donne plus de rejets et moins d'erreurs et vise versa. Le choix du seuil optimal est un compromis entre le taux de rejet et le taux d'erreurs et dépend de l'application en question, certaines applications telles que la reconnaissance des montants sur les chèques est plus sensible

aux erreurs nécessitant ainsi un taux de rejet élevé alors que d'autres sont plus tolérants aux erreurs.

Seuil	Précision	Performance	Taux de Rejet
0,00	98,87	98,87	0,00
0,04	98,83	95,64	3,23
0,08	98,83	95,64	3,23
0,12	98,83	95,64	3,23
0,16	98,83	95,64	3,23
0,20	98,83	95,64	3,23
0,24	98,83	95,64	3,23
0,28	98,83	95,64	3,23
0,32	98,83	95,64	3,23
0,36	98,83	95,63	3,24
0,40	98,85	95,64	3,25
0,44	98,85	95,64	3,25
0,48	98,85	95,64	3,25
0,52	98,93	95,45	3,61
0,56	98,98	95,27	3,90
0,60	99,01	95,15	4,08
0,64	99,09	95,05	4,34
0,68	99,12	94,91	4,54
0,72	99,15	94,76	4,74
0,76	99,14	94,51	4,99
0,80	99,17	94,24	5,32
0,84	99,21	94,00	5,64
0,88	99,30	93,73	6,08
0,92	99,37	93,34	6,60
0,96	99,43	92,29	7,79
1,00	100,00	1,13	100,00

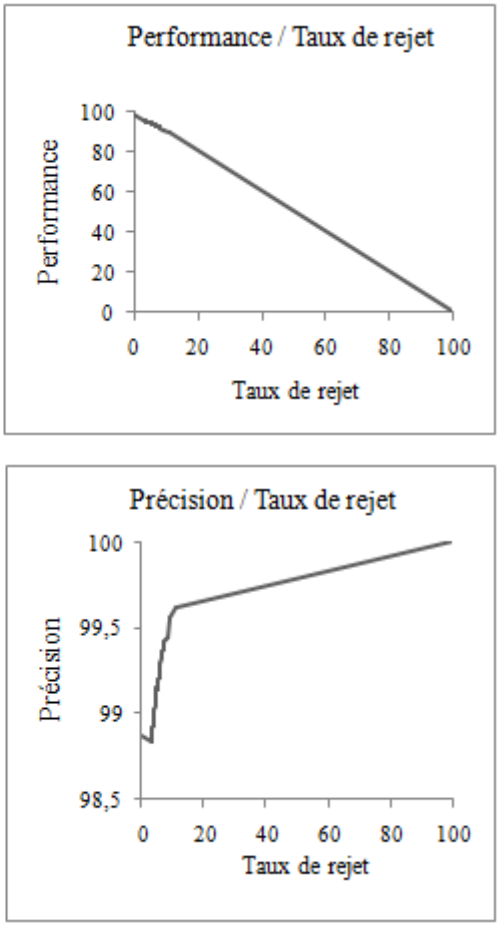


Figure 38. Courbes (Performance/Taux de rejet) et (Précision/Taux de rejet) pour l'exemple présenté au tableau

Pour évaluer l'utilisation du seuil pour le rejet et choisir le seuil optimum, la courbe performance/taux de rejet (ou la précision / taux de rejet) est construite en faisant varier le seuil *T* dans l'intervalle [0,1] et mesurer pour chaque valeur *T* la performance (précision) et le taux de rejet (Figure 38). Cette courbe permet de choisir le seuil du rejet qui s'adapte mieux au besoin de l'application cible.

Une autre courbe, appelée la courbe ROC, est souvent utilisée pour déterminer le seuil optimal dans les problèmes de classification [105]. Cette courbe représente l'évolution de la sensibilité en fonction de (1-spécificité) lorsque l'on fait varier le seuil T (Figure 39). L'aire sous la courbe ROC (AUC) donne une bonne estimation de la capacité de rejet du système, c'est-à-dire sa capacité à rejeter des motifs mal classés sans rejeter les caractères bien classés.

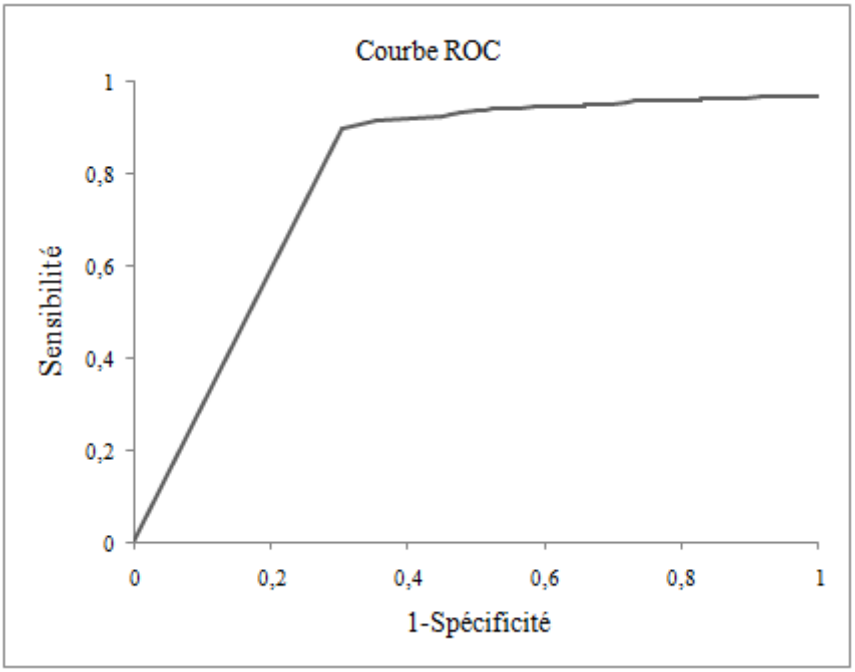


Figure 39. La courbe ROC de l'exemple précédent

L'option de rejet proposée par Chow utilise un seuil global pour toutes les classes, Fumera et al [52] ont prouvé que l'utilisation des multi-seuils (un seuil par classe) dans les problèmes de classification multi-classes peut être bénéfique. La règle du rejet devient alors :

$$\max_{k=1,\ldots,N} P(w_k \mid x) = P(w_i \mid x) < T_i$$

Un caractère est rejeté si sa meilleure probabilité d'appartenance à une classe k est inférieure au seuil correspondant à cette classe. Le problème réside dans le choix de la combinaison des seuils T_i qui optimise la performance du classifieur. Dans leur travail, les auteurs présentent un algorithme pour l'estimation de ces seuils. Une autre manière de trouver les seuils consiste à choisir, pour chaque classe i le seuil Ti permettant de maximiser les performances du rejet du classifieur pour cette classe [29]. i.e. trouver le meilleur seuil maximisant certains scores du rejet tels que la performance, la précision, la F-mesure, etc. tout en gardant un raisonnable taux de rejet. Les scores du rejet sont construits en utilisant la matrice de confusion [123] comme montré dans la Table 2.

Table 2. Matrice de confusion pour l'option du rejet

		Option de rejet	
		Accepté	**Rejeté**
Prédictions du classifieur	**Correcte**	Correctement Accepté (TA)	Incorrectement Rejeté (FR)
	Incorrecte	Incorrectement Accepté (FA)	Correctement Rejeté (TR)

Où :

- TA : est le nombre de caractères acceptés et correctement classifiés ;
- FR : est le nombre de caractères rejetés mais correctement classifiés ;
- FA : est le nombre de caractères acceptés mais mal classifiés ;
- TR : est le nombre de caractères rejetés et sont déjà mal classifiés.

Ainsi, les scores du rejet deviennent alors :

- $Performance = \frac{TA+TR}{TA+FA+TR+FR}$
- $Précision = \frac{TA}{TA+FA}$
- $Recall = \frac{TA}{TA+FR}$
- $F - Mesure = \frac{(\beta^2+1)*Precision*Recall}{\beta^2*Precision+Recall}$

Il faut noter que l'implémentation du rejet à seuil nécessite que les classifieurs soient probabilistes, dans le cas où l'on a des classifieurs dont les sorties sont purement discriminantes, la solution et de convertir ces sorties en probabilités.

IV. Post-traitement

L'objectif du post-traitement est de détecter et corriger les fautes d'orthographe commises par le système OCR après que l'image d'entrée soit complètement traitée. La manière évidente de corriger les erreurs OCR consiste à éditer le texte de sortie manuellement par des linguistes, cette méthode nécessite une intervention humaine manuelle continue qui est dans une certaine mesure considérée comme une longue et coûteuse pratique. Une autre solution consiste à automatiser la correction des sorties OCR, Il existe deux approches principales dans ce contexte [4].

La première approche est basée sur une correction d'erreur lexicale, dans cette méthode, un lexique est utilisé pour épeler les mots reconnus par OCR et les corriger s'ils ne sont pas présents dans le dictionnaire. Bien que cette technique soit facile à mettre en œuvre et à utiliser, elle présente encore diverses limitations qui l'empêchent d'être la solution parfaite pour la correction des erreurs OCR : la première est qu'elle nécessite un dictionnaire étendu couvrant tous les mots de la langue ; la deuxième limitation est que les dictionnaires conventionnels ne prennent pas en charge les noms de régions, d'emplacements géographiques, de mots clés techniques et de termes spécifiques au domaine et ciblent normalement une seule langue spécifique dans une période donnée, et ne peuvent donc pas prendre en charge des documents historiques avec des styles d'écriture différents [4].

La deuxième approche du post-traitement OCR est la correction des erreurs basée sur le contexte. Ces techniques sont fondées sur des corpus linguistiques et sur la modélisation du langage statistique et les n-grammes de mots [4].

L'absence des corpus et des grands dictionnaires des mots amazigh en Tifinagh rend difficile l'intégration de l'étape du post-traitement dans l'architecture des systèmes OCR Amazighs.

Conclusion

Dans ce chapitre, nous avons présenté un petit historique, les principaux concepts et les étapes nécessaires pour construire les systèmes OCR. Ces systèmes ont une grande importance et utilité vue leurs diverses applications, cependant, ces systèmes présentent certaines complexités telles que la variation des polices et styles, écriture manuscrite cursive, documents dégradés, documents bruités, arrière-plans complexes, multilinguisme, etc. En outre, les systèmes OCR sont constitués de plusieurs étapes, où chaque étape du système est largement influencée par les résultats de celle qui la précède et l'étape de l'extraction des caractéristiques est celle considérée comme la plus importante dans ces systèmes. Actuellement, les systèmes OCR sont loin d'imiter la lecture humaine, et les recherches sont encore intenses dans ce domaine pour surmonter les difficultés et les défis rencontrés.

Chapitre. 2 : La langue Amazighe et son alphabet Tifinagh

Introduction

Dans ce chapitre, nous nous penchons sur la langue Amazighe et l'évolution de son script Tifinagh depuis ses premiers jours jusqu'à nos jours. Ce chapitre reporte aussi les principales réalisations effectuées pour promouvoir et informatiser cette langue et permettre son traitement automatique.

Comme l'attestent tous les historiens de l'Afrique du nord, les Amazighs ont peuplé, depuis les anciens temps, la région appelée Tamzgha (Du Maroc au Oasis Siwa à l'Égypte, passant par la Mauritanie, l'Algérie, la Tunisie, la Lybie, Mali, Niger et Burkina Faso) [23]. Ils ont leur propre langue, appelée la langue Amazighe comportant actuellement jusqu'à 23 variantes parlées par environ de 30 millions de personnes dans différentes régions de cette vaste zone (Figure 40).

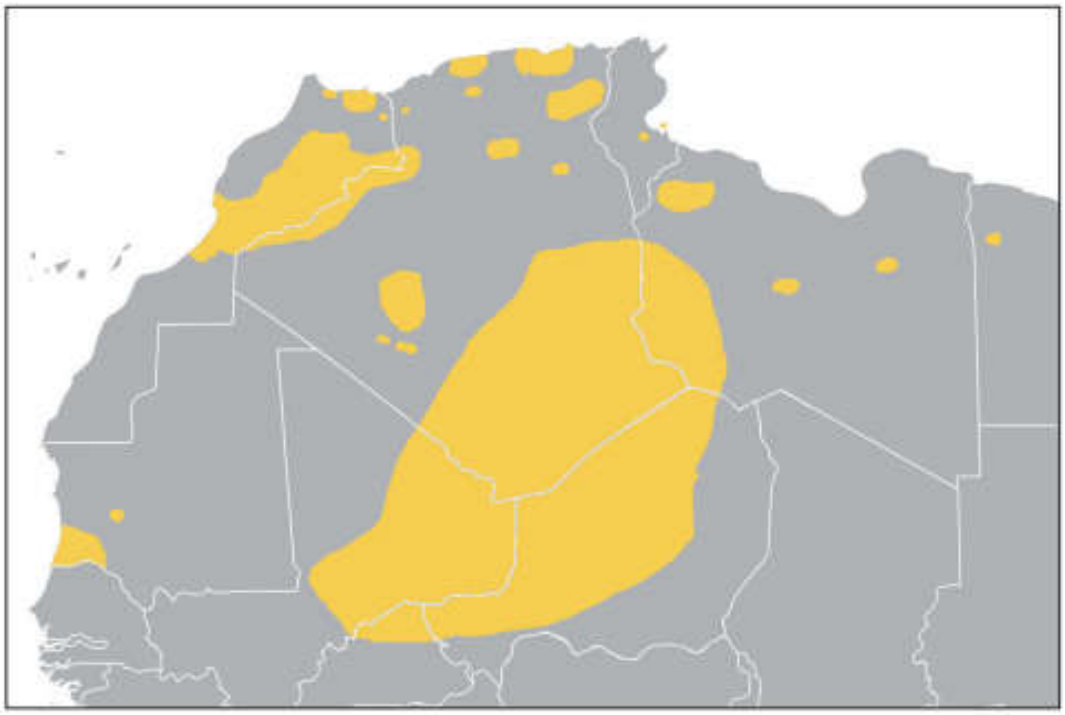


Figure 40. Les zones parlantes la langue Amazighe actuellement en Afrique du Nord [123]

Dès ses premiers jours, la langue Amazighe a connu de nombreuses langues à l'est et à l'ouest, telles que l'ancienne langue égyptienne et latine. Bien que ces deux langues ont construit

une civilisation magnifique et urbaine et ont laissé des textes écrits et gravés sur les pierres des temples, des palais et des tombes, mais ils ont disparu pour toujours et ont apporté avec eux des trésors de connaissance. En revanche, la langue Amazigh qui n'a pas été écrit ni enregistré, sauf dans quelques rares cas, et qui n'est utilisé que dans un cadre informel et familial a pu survivre et rester en vie dans la bouche des dizaines de millions des Amazighs.

Au Maroc, où presque 50% des gens sont des Amazighs, la langue Amazighe est divisée en trois variantes selon la région : Tarifit au nord ; Tamazight au centre et au sud-est et Tachelhit au sud-ouest et au grand Atlas [9].

Cependant, malgré le grand nombre des locuteurs amazighs, l'utilisation du script Tifinagh est encore rare au Maroc. Cette utilisation s'est considérablement amélioré au cours des dernières décennies à la suite des revendications des mouvements culturels amazighs dans les années 1990 [23] qui ont abouti à la standardisation de la langue Amazighe, son introduction au système scolaire marocain et la création du centre IRCAM. Ce dernier a adopté l'alphabet original Tifinagh pour la retranscription officielle de la langue Amazighe. Cet alphabet sera décrit dans la section suivante.

I. Le script Tifinagh

Les amazighs possèdent, depuis l'antiquité, leur propre écriture alphabétique de nature consonantique appelée Tifinagh. L'origine de cette écriture est toujours une question et plusieurs hypothèses ont été avancées aboutissant à deux théories différentes : la première qui supporte l'idée de l'origine orientale du Tifinagh; tandis que la deuxième défend l'idée de l'autochtonie de cette écriture [20].

L'alphabet Tifinagh a enduré plusieurs variations tout au long de l'histoire subissant ainsi plusieurs modifications [14] :

- **Le Libyque** : existe sous deux formes, l'oriental et l'occidental. C'est les plus anciennes des variétés du Tifinagh ;
- **Les Tifinagh sahariens** : connue aussi sous l'appellation libyco-berbère ou touareg ancien, cette variété contient plus de signes par rapport au libyque ;
- **Les Tifinagh touarègues** : Ces variétés sont utilisées par les touarègues avec une légère différence au niveau de la forme et le nombre des signes d'une région à autre ;
- **Les néo-Tifinagh** : c'est les plus récentes des variétés, elles ont été développées pour la transcription des parlers Amazighs du Maghreb. Le Tifinagh-IRCAM adopté au Maroc et développé durant la période de standardisation de la langue amazighe constitue une de ces variétés.

Quelle que soit son origine ou son histoire, des preuves historiques révèlent que le Tifinagh est utilisé depuis plusieurs siècles par les nord-africains en tant que système d'écriture mais aussi comme une marque symbolique et identitaire. Cependant, cette utilisation était limitée aux messages courts, dédicaces et des inscriptions sur des objets (pierres, bijoux, armes, tapis, etc.) (Figure 41).

Figure 41. Inscription sur une pierre en Tifinagh [23]

1. L'alphabet standardisé Tifinagh-IRCAM

Afin de préserver le patrimoine langagier Amazigh, en majorité oral, et permettre le passage de l'oral vers l'écrit, et dans le but d'unifier la graphie des dialectes amazighs marocains contemporains, le Centre d'Aménagement Linguistique (CAL) de l'IRCAM a développé un système de formes alphabétiques [14] permettant une notation simple et exhaustive de la langue amazighe tout en tenant en considération :

- **L'historicité :** l'alphabet amazigh doit comporter des graphèmes historiques ou authentiques.
- **La simplicité :** les graphèmes choisis doivent être simples, c'est-à-dire sans signe additionnel. Les lettres composées et les signes diacritiques ont été évités.
- **L'univocité du signe :** ce principe assure la correspondance univoque entre un son et un graphème. Il permet d'éviter les diagraphes et les bi-consonnes ou les ligatures.
- **L'économie :** seuls les graphèmes correspondants aux phonèmes de base de l'amazigh figurent dans l'alphabet proposé.

Ainsi, l'alphabet proposé par l'IRCAM comporte Trente-trois caractères correspondants aux phonèmes de la langue amazighe dont :

- Vingt-sept consonnes
    - Vingt consonnes simples (ⴱ, ⴳ, ⴳⵯ, ⴷ, ⴼ, ⵀ, ⵃ, ⵅ, ⵅⵯ, ⵇ, ⵊ, ⵍ, ⵎ, ⵏ, ⵔ, ⵙ, ⵜ, ⵡ, ⵣ, ⵢ, ⵥ, ⵯ)
    - Cinq consonnes emphatiques (ⴹ, ⵟ, ⵠ, ⵚ, ⵛ)
    - Deux semi-consonnes (ⵡ, ⵢ)
- Quatre voyelles
    - Trois voyelles de base (ⴰ, ⵉ, ⵓ)
    - Voyelle neutre (ⵄ)

- Deux labiovélaires (ⴽ⵿, ⵅ⵿)

Le tableau ci-dessous présente les caractères de l'alphabet Tifinagh de base, leurs appellations en français et leurs valeurs phonétiques.

Table 3. L'alphabet Tifinagh-IRCAM de base

Caractère Amazigh	Appellation en français	valeur phonétique
ⴰ	Ya	a
ⴱ	Yab	b
ⴳ	Yag	g
ⴳ□	Yagw	g^w
ⴷ	Yad	d
ⴹ	Yadd	ḍ
ⴻ	Yey	∂
ⴼ	Yaf	f
ⴽ	Yak	k
ⴽ□	Yakw	k^w
ⵀ	Yah	h
ⵃ	Yahh	ħ
ⵄ	Yaa	ɛ
ⵅ	Yakh	x
ⵇ	Yaq	q
ⵉ	Yi	i
ⵊ	Yaj	ʒ
ⵍ	Yal	l
ⵎ	Yam	m
ⵏ	Yan	n
ⵓ	You	u
ⵔ	Yar	r
ⵕ	Yarr	ṛ
ⵖ	Yagh	γ
ⵙ	Yas	s
ⵚ	Yass	ṣ
ⵛ	Yach	ʃ
ⵜ	Yat	t
ⵟ	Yatt	ṭ
ⵡ	Yaw	w
ⵢ	Yay	y
ⵣ	Yaz	z
ⵥ	Yazz	ẓ

Après une proposition de l'IRCAM en 2004 [14], par son Centre des Études Informatiques, des Systèmes d'Information et de Communication (CEISIC), pour l'addition de l'alphabet Tifinagh au répertoire de L'ISO/CEI 10646, l'organisation de standardisation internationale (ISO) a introduit l'alphabet Tifinagh-IRCAM (ainsi que d'autres caractères pour des besoins historiques et dialectaux) à l'Unicode et à l'ISO 10646 annonçant ainsi la normalisation

internationale du Tifinagh. La figure ci-dessous illustre le bloc Unicode réservé à l'alphabet Tifinagh.

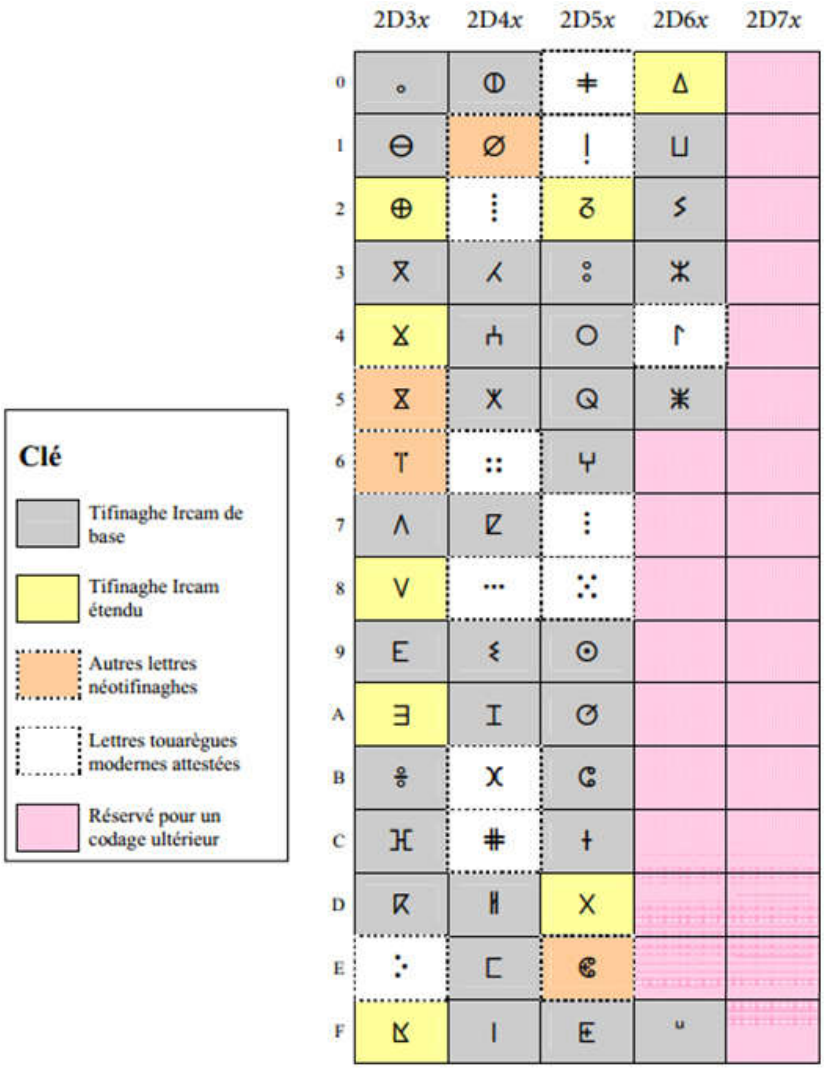


Figure 42. Codes hexadécimaux de l'alphabet Tifinagh

Comme la Figure 42 l'indique, le bloc Unicode réservé à l'alphabet Tifinagh comporte quatre sous-ensembles :

- Tifinagh IRCAM de base : sert à coder les 33 caractères de base de l'alphabet Tifinagh-IRCAM. Ce sous ensemble présente un codage direct pour 31 caractères et un codage du signe de labellisation « □ » qui participe à la formation des 2 caractères labiovélaires restants « □□ » et « □□ » ;
- Tifinagh IRCAM étendu : contient les 8 lettres ajoutées pour des raisons historiques et scientifiques.
- Lettres néo-tifinaghes : Ce sous ensemble sert à designer quatre autres caractères néo-tifinaghs utilisés couramment dans le reste du Maghreb.
- Lettres touarègues : code les 11 lettres touarègues modernes.

La proposition de l'IRCAM pour la normalisation comportait quelques spécifications à prendre en comptes lors de l'utilisation de cet alphabet :

- **Ponctuation** : pour la ponctuation, le CAL a recommandé l'utilisation des signes conventionnels répandus dans l'écriture latine. (« . », « , », « ; », « : », « ? », « ! », etc.) ;
- **Chiffres** : le CAL a retenu le système de numération arabe (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) pour les chiffres ;
- **Orientation** : l'orientation de l'écriture adoptée est horizontale de gauche à droite ;
- **Casse** : Pas de symboles majuscules ou minuscules ;
- **Ligatures** : aucune ligature n'est retenue ;
- **Cursivité** : l'écriture Tifinagh ne présente aucune forme de cursivité ;
- **Diacritiques** : Aucun signe diacritique.

L'analyse de la morphologie de l'alphabet Tifinagh révèle quelques caractéristiques intéressantes, en particulier, la redondance des traits horizontaux, verticaux et diagonaux dans la majorité des lettres ainsi que la redondance des arcs circulaires (Figure 43). Ainsi on pourra regrouper les caractères selon leur structure géométrique :

- Groupe des ronds (ⴻ, ⵔ, ⵙ, ⵀ, ⵇ, ⵕ, ⵟ) ;
- Groupe des doubles ronds (ⵓ, ⵖ, ⵄ) ;
- Groupe des bidents carrés (ⵛ, ⵡ) ;
- Groupe des bidents et tridents rectangulaires (ⵎ, ⴼ, ⵒ, ⵍ) ;
- Groupe stellaire (ⵜ, ⵣ, ⵥ, ⵅ, ⵆ) ;
- Groupe des éclairs (ⵉ, ⵋ, ⵢ) ;
- Groupe pyramidal (ⴷ, ⴹ) ;
- Groupe des épaules (ⴽ, ⵃ) ;
- Groupe fluet (ⵏ, ⵊ).

Figure 43. Segments de base observés pour les caractères Tifinagh

2. Implantation du Tifinagh dans le domaine de l'informatique

Dans cette partie, nous allons survoler l'état actuel des travaux réalisés pour l'intégration du script Tifinagh dans les outils informatiques et technologiques.

« Il y a huit ans, il était impossible d'envoyer des documents en tifinaghes sans se référer à un codage de police privé. Aujourd'hui, on peut créer des pages HTML, des documents XML en tifinaghes, envoyer des courriels. Il existe un clavier normalisé pour saisir des textes tifinaghes, une norme de tri, Microsoft fournit une police qui prend en charge les tifinaghes. L'utilisateur peut désormais voir des pages HTML sans qu'il n'ait à explicitement installer des polices

tifinaghes sur son système. Des bibliothèques logicielles comme ICU prennent également en charge les tifinaghes et il est possible, en théorie, d'avoir des noms de domaine Internet en tifinaghes » [13].

En effet, depuis sa normalisation et son intégration dans l'ISO, plusieurs travaux et projets ont été lancé par l'IRCAM en collaboration avec des organisations nationales et internationales afin de rendre le caractère Tifinagh accessible et utilisable dans les nouvelles technologies d'information et de communication, notamment internet et les applications informatiques.

2. 1. **Claviers des caractères Tifinagh**

Après la normalisation internationale de l'alphabet Tifinagh, le CAL a proposé deux types de claviers conformes à la norme de claviers ISO/CEI 9995 en se basent sur les caractères Tifinagh définis dans le standard Unicode 4.1. Le premier clavier, clavier Tifinagh de base (Figure 44.a), a pour rôle la saisie des 33 caractères de l'alphabet de base Tifinagh-IRCAM ; tandis que le deuxième, clavier Tifinagh étendu (Figure 44.b), permet la saisie des 55 caractères du Tifinagh présents dans le standard Unicode.

Ces claviers ont été confirmés par le Service de Normalisation Industrielle Marocaine (SNIMA) [140] et intégrés par Microsoft à partir de sa version 8.0 mais aussi par plusieurs fabricants Linux.

(a)

(b)

Figure 44. Claviers proposés par le centre IRCAM

(a) Clavier Tifinagh de base

(b) Clavier Tifinagh étendu

En outre, depuis janvier 2014, une nouvelle application (*Tifinagh Font)* pour installer le clavier Tifinagh pour des interfaces Android est également disponible (Figure 45).

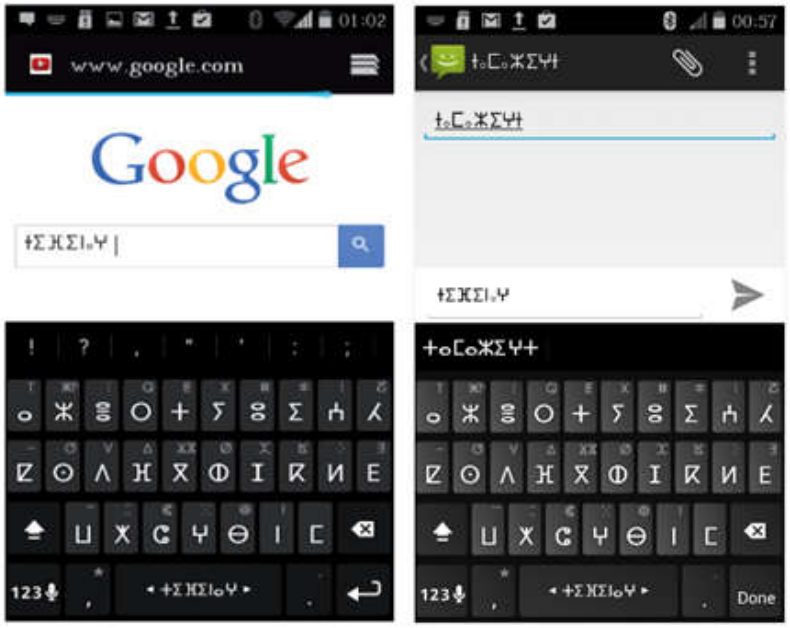
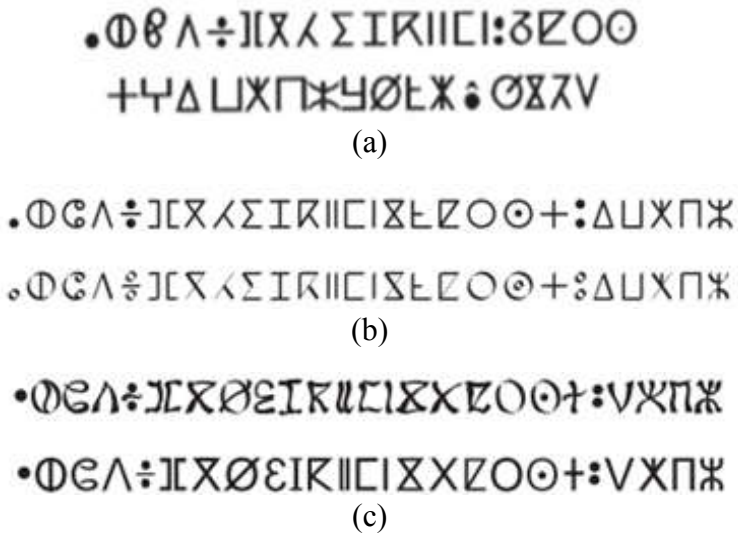


Figure 45. Clavier Tifinagh pour Android

2. 2. Les polices créées pour le script Tifinagh

Les premières tentatives de conception d'alphabets néotifinaghs sont apparues à une époque où la standardisation du script était toujours en cours. *Agraw Imazighen* par l'Académie berbère (1967), Afus deg Wfus basé sur l'alphabet proposé par Salem Chaker (1993), et Massensen et Jugurthen proposés par Arezqi Buzefran (1995) sont des exemples de ces tentatives (Figure 46) [23].



(a) (b) (c)

Figure 46. Exemples des premières tentatives de conception d'alphabets néotifinaghs [23]

(a) Alphabet proposé par l'académie berbère (1967)

(b) Alphabet Afus deg Wfus basé sur l'alphabet proposé par Salem Chaker (1993)

(c) Polices Massensen et Jugurthen proposés par Arezqi Buzefran (1995)

La première décennie du 21ème siècle est probablement la plus influente en ce qui concerne le développement des polices numériques pour le Tifinagh et sa consolidation en tant que script pour la langue amazighe. En effet, après la fondation du centre IRCAM, ce dernier a publié sa première police en 2003, *Tifinagh-IRCAM-UNICODE*, qui est très courante aujourd'hui dans les livres scolaires publiés en Tifinagh. Des nouvelles gammes de polices ont été publiées en 2006 et 2011 permettant ainsi d'enrichir le répertoire des polices Tifinagh, ces nouvelles polices explorent des problèmes typographiques communs comme la modulation, la largeur, le poids et l'utilisation des empattements mais étaient aussi une nouvelle tentative de développement d'un jeu de minuscules (Figure 47).

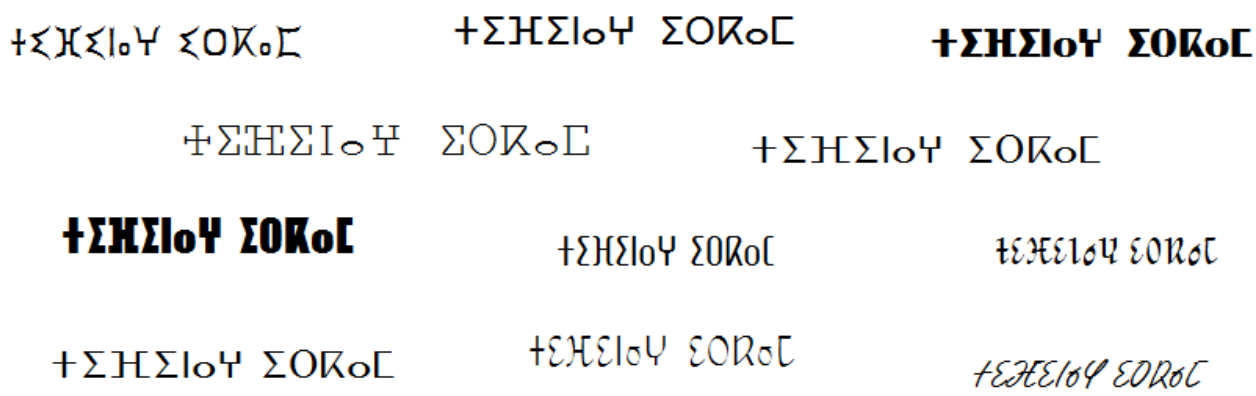


Figure 47. Exemples des nouvelles gammes des polices Tifinagh proposées par l'IRCAM

Ces polices sont utilisables sur les différentes plateformes Windows, Linux et Mac OS et sont disponibles (format True Type Fonts TTF) avec les pilotes adéquats sur le site officiel du centre IRCAM.

En parallèle avec le centre IRCAM, en 2007, Microsoft a conçu la police *Ebrima* pour supporter un grand nombre de langues africaines y compris le Tifinagh. Cette police ressemble à la police *Tifinagh-IRCAM-UNICODE* avec quelques ajustements optiques (Figure 48).

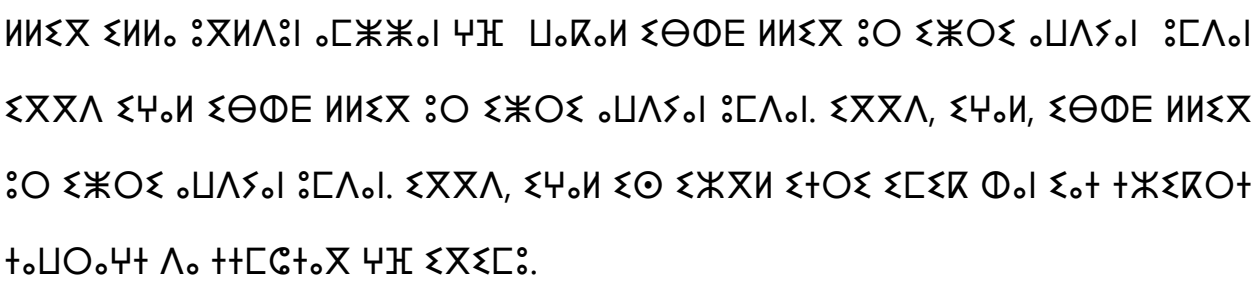


Figure 48. Extrait du livre le petit prince en Tifinagh avec la police Ebrima

Outre ces exemples, de nombreuses conceptions indépendantes produites ces dernières années sont disponibles sur Internet (Figure 49). Ce sont, en majorité, des polices d'affichage qui ne traitent pas les problèmes spécifiques du script Tifinagh dans les textes longs, à savoir la lisibilité et l'uniformité du style.

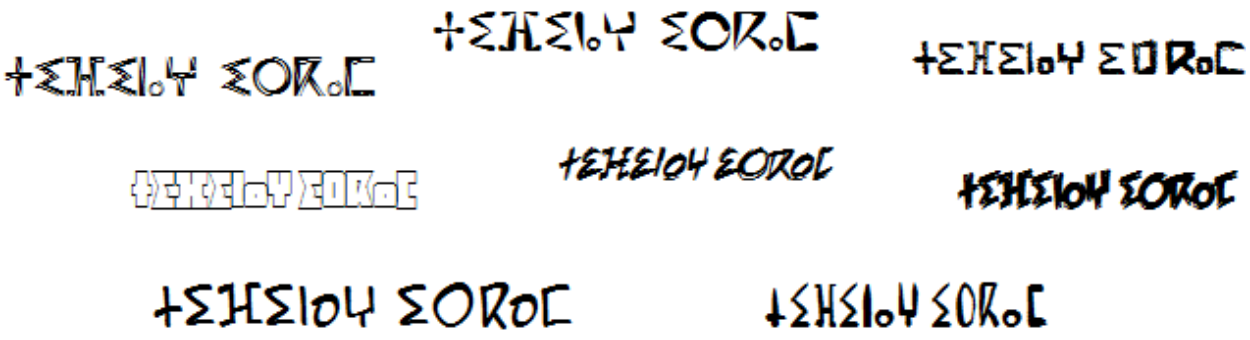


Figure 49. Exemples des conceptions indépendantes des polices Tifinagh

2. 3. Indicatif de langue pour les documents Amazighs

L'ajout de l'indicatif de langue « zgh », désignant l'amazigh standard marocain, dans la liste officielle des noms des langues a été approuvé par le comité de l'ISO 639-2 en 2012. Désormais, on peut indiquer que le contenu d'un document numérique (écrit ou parlé) est en amazighe standard marocain.

La Figure 50 montre un exemple d'un fichier HTML dont le contenu et en Tifinagh.

```
<Html>
    <Body lang= 'zgh'>
        <P>
        ⴰⵏⵏⴰⵋ ⵍⵄⴰ ⴰⵏ ⵍⵙⵙⵎⵢⵄⵙ ⵅⵎⴰⵍ ⵍⵍⵢ ⵣⵔⴰ ⵏ ⵜⵅⵣⵣⴰⵍ ⵏⴰ ⵍⵙⵣⵅⵙ ⵅⵎⵅⵣ ⵄⵙ ⵡⵡⵅⵢ ⵏ
        ⴰⵎⴰⵢⵣⴰⵍ ⵡⵡⵅⵜ ⴰⵍⵍ ⵙⴰⵍⵡⵢ ⵢⵜ ⵅⵎⵙⵙⵅⵢⵍ ⵏ ⵜⵙⵜⴰⵍⵅⵍ ⵙ ⵍⴰⵍⴰⵡ ⵅⵍⵄ ⵣⵎⵙⵍ ⵍⵅⵏⵏ ⵄⵙ
        ⵅⵙⵙⵅⵍⵄⵍ ⴰⵣⴰⵡ ⴰⵏ ⵜⵡⵡⴰⵙ ⵅⵍⵅ ⵏ ⵄⵙⵍⵅⵍⵜⵎ ⴰⵎⵜⴰⵢⵎ
        </P>
    </Body>
</Html>
```

Figure 50. Fichier HTML dont le contenu est en Tifinagh

2. 4. XML et HTML en contenu Tifinagh

En général les fichiers XML, HTML ou encore XHTML peuvent contenir du contenu Tifinagh sans problèmes comme le montre la Figure 51 affichant la page Web d'accueil officielle de *Facebook* en Tifinagh. La présence de ce dernier sur Internet est en augmentation depuis que les nouvelles polices numériques ont été adaptées et encodées pour le web.

Dans ce genre de fichiers le problème réside dans l'utilisation de cet alphabet au niveau des éléments (balises) ou des attributs et leurs valeurs. Cependant, dans sa cinquième édition de XML 1.0, publiée en 2008, le World Wide Web Consortium (W3C) chargé de la normalisation de XML et XHTML a permis l'utilisation de la quasi-totalité des caractères Unicode, y compris le Tifinagh, dans les noms d'éléments ainsi que dans les noms d'attributs et leurs valeurs. Mais cette utilisation risque d'être sans intérêt vu que sa prise en charge par les analyseurs XML nécessite leur modification au niveau des règles de formation des noms (par exemple, une balise en Tifinagh n'aura pas de sens pour le navigateur) [13].

Figure 51. Page officielle d'accueil de Facebook en Tifinagh

Au Maroc, la standardisation de l'alphabet Tifinagh, en vue de la retranscription officielle de la langue Amazighe, et son introduction dans les nouvelles technologies a donné renaissance aux caractères Tifinagh. Désormais, cette écriture est de plus en plus utilisée dans les différents domaines de l'information et de communication tels que : la publicité, Les médias, le Web, l'éducation, les panneaux de signalisation et des établissements publiques, etc. (Figure 52).

(a) (b)

(c) (d)

Figure 52. Exemples de l'utilisation actuelle du Tifinagh

Les chercheurs ont vite remarqué cette expansion du Tifinagh et ont constaté la nécessité de son traitement automatique, mais les travaux réalisés jusqu'à l'instant restent loin d'être satisfaisants.

II. Vers le traitement automatique du Tifinagh

Toute langue visant à : survivre dans un monde en perpétuelle mondialisation ; à préserver son patrimoine langagier ; et être utilisée normalement dans l'éducation, les médias et les nouvelles technologies de l'information et de communication, doit nécessairement disposer des ressources et des outils linguistiques. Ces dernières peuvent être des dictionnaires ou des corpus, de préférence sous forme numérique, et des outils informatiques permettant d'effectuer les différentes tâches du traitement automatique du langage naturel (TALN).

Dans ce contexte, après la standardisation de la langue amazighe, largement négligée, beaucoup d'efforts ont été fournis par le centre IRCAM et les chercheurs afin d'intégrer cette langue dans les nouvelles technologies de l'information et de communication, créer des ressources linguistiques et fournir des outils du traitement automatique de la langue Amazighe. Cette partie décrit les principales réalisations effectuées en vue d'un traitement automatique du script Tifinagh, en particulier, les ressources et les systèmes OCR visant à contribuer à la sauvegarde et la numérisation des fonds documentaires Amazighs.

1. Développement des corpus

Les corpus numériques offrent aux chercheurs l'occasion de traiter les données du langage avec une variété d'outils et de techniques pour bien mener des recherches linguistiques. Ces corpus sont de différents types. En fait, c'est une tâche très cruciale de classer les corpus linguistiques en différents types. Cependant, corpus écrit, corpus parlé, corpus général, corpus monolingue, corpus bilingue, corpus non annoté, corpus annoté méritent d'être mentionnés [10].

Au départ, les corpus linguistiques ont été rarement associés au développement des logiciels pour leurs analyses. Dans les corpus linguistiques modernes, les linguistes et les informaticiens partagent un objectif commun, celui de créer des corpus avec des données linguistiques réelles ou actuelles et pouvoir en effectuer n'importe quel type d'analyse linguistique. Ces dernières années, il y a eu un énorme intérêt pour la construction et le développement de corpus numériques.

La langue Amazighe souffre encore de la rareté des outils et des ressources linguistiques et ne possède pas de grands corpus. Par conséquent, les chercheurs de l'Institut IRCAM ont tenté de construire des corpus amazighs jusqu'à atteindre un corpus à grande échelle. En effet, trois travaux ont été réalisés dans ce contexte, mais malheureusement, ces corpus sont privés et de petite taille [10, 25, 117]. Ces travaux sont basés sur des textes extraits des différents romans, contes, poèmes, magazines, sites WEB et supports pédagogiques.

2. Bases des images des caractères Tifinagh

Afin d'évaluer la performance des systèmes OCR, les chercheurs utilisent des bases de données des images textuelles. Ces bases de données sont souvent communes et disponibles

offrant ainsi la possibilité de comparer entre les différents systèmes OCR. Malheureusement, la langue amazighe souffre d'un manque à ce niveau. La majorité des systèmes OCR amazighs ont été testé sur des bases de données développées localement contenant un nombre restreint de caractères. Dans cette section, nous allons décrire deux bases de données les plus utilisées dans ce contexte, la base AMHCD [44] des caractères amazighs manuscrits et la base des caractères amazighs imprimés [8].

2. 1. La base d'images des caractères manuscrits isolés (AHMCD)

Cette base de donnés se compose en totalité de 25740 images de caractères manuscrits isolés. Pour collecter les données, les auteurs ont demandé à 60 scripteurs de différents âges, sexes et niveaux d'éducation de remplir des formulaires de 13 exemples pour chaque caractère (Figure 53).

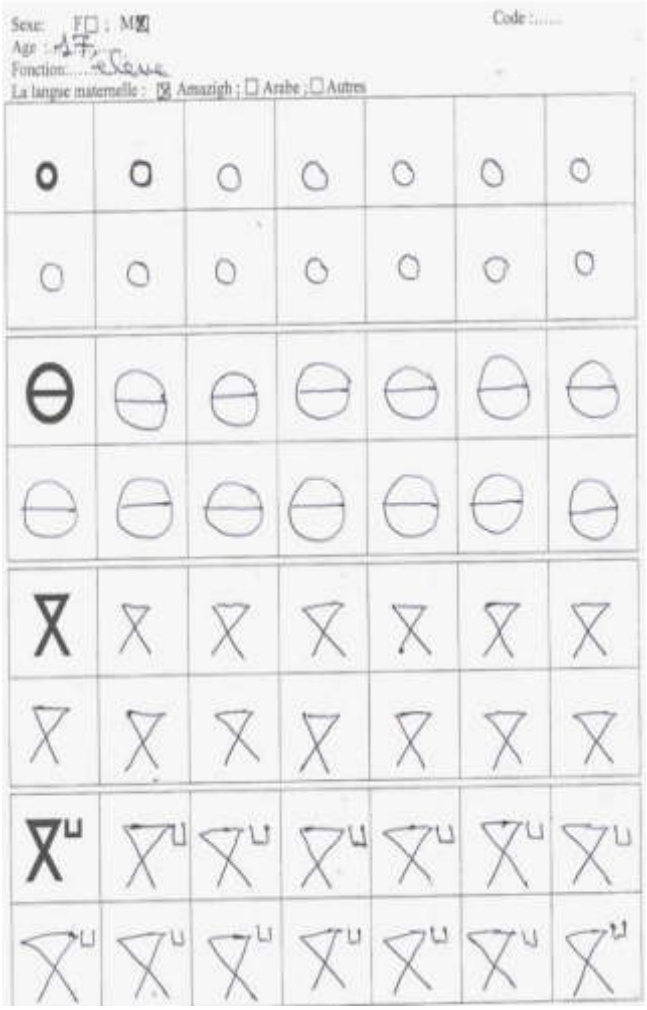
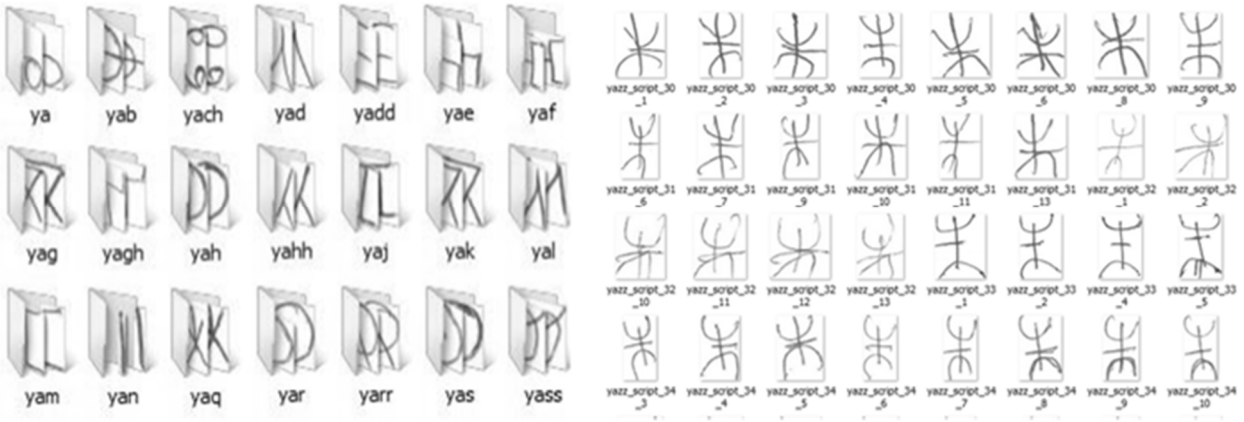


Figure 53. Exemple du formulaire pour la génération de la base d'images

Les formulaires collectés ont été numérisés par la suite avec une résolution de 2400 dpi. Enfin, les auteurs ont développé un système automatique pour traiter ces formulaires et extraire les images des caractères isolés. Ces images ont été labellisées et enregistrées dans des dossiers où chaque dossier contient seulement les images d'un seul caractère (Figure 54).



(a) (b)

Figure 54. Stockage sur disque de la BD AHMCD

Cette base est disponible sur demande par email aux auteurs et sera utilisée comme source d'apprentissage et de test dans le travail présenté dans la section I. 1. du chapitre 3.

2. 2. Bases des images des caractères imprimés

Les systèmes OCR développés pour reconnaitre les caractères amazighs imprimés ont été testés sur des bases de données locales de petites tailles. Certains travaux utilisent la base d'images développée par Ait Ouguengay et al qui est indisponible et mal référencée. Ceci nous a motivé pour créer une base d'images publique des mots imprimés en Tifinagh pour assurer la comparaison entre les systèmes OCR développés. Cette base d'images sera décrite dans le chapitre suivant.

3. Systèmes de reconnaissance automatique des caractères

La reconnaissance automatique des caractères Tifinagh n'as pas reçu la même attention que les autres langues telles que le latin, le chinois et l'arabe. En effet, le premier travail se date de 1988 où Oulamara et al [116] ont introduit le problème pour l'écriture amazighe imprimée et manuscrite et ont développé un système de reconnaissance basé sur la transformée de Hough. Neuf ans après, Djematene et al [39] ont présenté une méthode géométrique visant à extraire, à partir des images des caractères, des caractéristiques structurelles telles que les points d'intersection, les corners, les jonctions, etc. Les auteurs ont utilisé une distance métrique lors de la reconnaissance. Ce n'est qu'après la standardisation marocaine de la langue amazighe en 2003 par le centre IRCAM et son introduction officielle dans le système éducationnel que les chercheurs ont commencé à donner de l'intérêt à l'écriture amazighe et la dernière décennie a connu un léger épanouissement des systèmes OCR amazighs.

Dans cette section, nous allons survoler les principales techniques développées pour reconnaitre les caractères amazighs imprimés et manuscrits.

3. 1. Reconnaissance des caractères imprimés

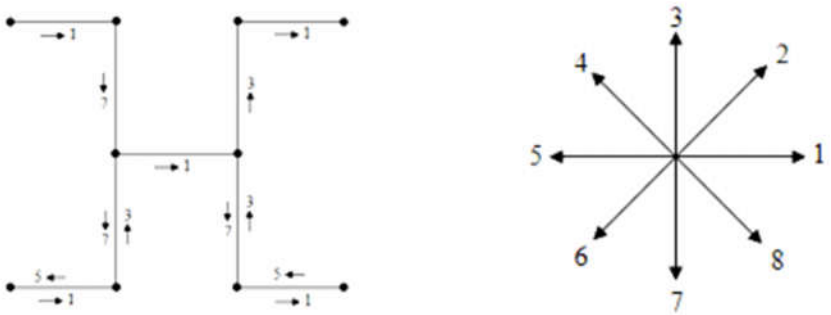
Comme mentionné précédemment, la première tentative de reconnaissance des caractères amazighs était élaborée en 1988 où les auteurs ont présenté une technique à base de la

transformée de Hough à la fois pour l'écriture imprimée et manuscrite, mais ils n'ont présenté des résultats expérimentaux que pour les caractères imprimés. L'analyse de l'image du caractère dans l'espace de Hough permet d'extraire des caractéristiques de longueur, orientation et location à partir des segments droits individuels détectés. Cependant, ce système ne prend pas en charge les caractères circulaires vu que tous les caractères ont été générés à base des segments droits (Figure 55) et les 29 caractères étudiés ne représentent pas l'alphabet Tifinagh-IRCAM.

Figure 55. Génération digitale des caractères [116]

Les auteurs ont testé leur système sur 341 caractères et ont obtenu un taux de 98%.

Les recherches ont été reprises en 2011, Essady et al [45] ont proposé une approche syntaxique en tenant en compte la structure des caractères, les primitives structurelles extraites à partir des squelettes des caractères telles que les segments, les points et les arcs ont permis de construire les chaines de Freeman en se basant sur les différentes orientations de Freeman (Figure 56).



(a) (b)

Figure 56. Extraction de la chaine de Freeman '1775131715331' du caractère Yaf (ⵅ)

Les codes de Freeman obtenus pour chaque caractère ont permis de déduire les grammaires régulières relatives et les caractères sont ensuite décrits par des automates finis. La Figure 57 présente un exemple de l'automate fini défini pour le caractère Yaf (ⵅ).

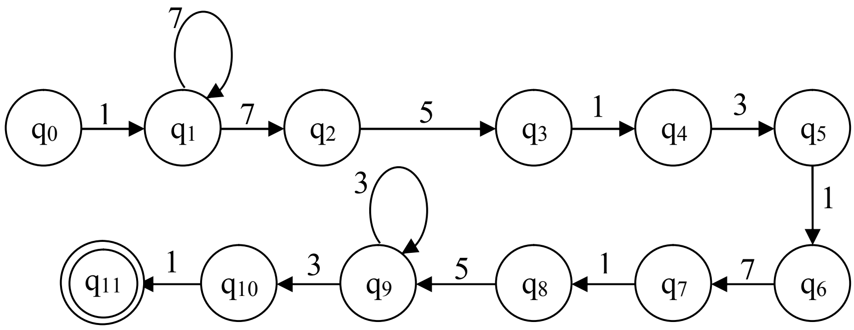


Figure 57. Automate fini pour le caractère Yaf (ⴼ) [45]

Les auteurs ont testé leur système sur une base d'images de 630 caractères obtenue à partir des 21 caractères (ⵊ, ⵏ, ⵜ, ⵖ, ⵀ, ⴼ, ⴽ, ⵍ, ⵇ, ⵛ, ⵡ, ⴻ, ⵟ, ⴷ, ⴹ, ⴳ, ⵅ, ⵉ, ⵄ, ⵣ, ⵥ) en évitant ainsi les caractères présentant une forme circulaire. Le taux de reconnaissance obtenu atteint 93,5%.

El Ayachi et al [19] ont développé un système de reconnaissance des caractères imprimés basé sur la transformée de Walsh pour extraire les caractéristiques à partir des images des caractères isolés après une segmentation de l'image d'entrée qui a subit certains prétraitements (binarisation, filtrage et correction de l'inclinaison). Dans la phase de reconnaissance, ils ont utilisé le perceptron multicouche avec une seule couche cachée de 3 neurones et 49 neurones en entrée correspondants aux 49 caractéristiques de Walsh extraites, cependant, la couche de sortie est composée de 6 neurones ce qui ne correspond pas au nombre de caractères de l'alphabet Tifinagh.

Bencharef et al [22] ont utilisé les descripteurs géodésiques supposés robustes face au bruit et les distorsions géométriques pour extraire les caractéristiques. Une phase de prétraitements et nécessaire pour extraire les quatre extrémités géométriques de chaque caractères, ensuite, les six distances géodésiques sont calculées (Figure 58).

Figure 58. Détection des extrémités et distances géodésiques

Lors de la reconnaissance, les auteurs ont combiné le PMC et les arbres de décisions qui ont abouti à un taux de reconnaissance de 93% sur une base de 528 caractères.

Amrouch et al [11] ont proposé une approche structurelle où les caractères ont été pré classifiés en 2 classes selon leur circularité en utilisant la transformée de Hough, ensuite, les caractères non circulaires ont été décomposés aussi en deux sous-classes selon leur nombre des composants connectés CC. Chaque classe de caractères subit un prétraitement approprié pour les préparer à la phase d'extraction des caractéristiques où plusieurs caractéristiques structurelles ont

été utilisées telles que le nombre de trous, la longueur minimale et maximale des traits, les surfaces, les diamètres, les périmètres, les positions et orientations des segments, les intersections, les jonctions, etc. Dans la phase de reconnaissance, les auteurs ont utilisé les modèles de Markov cachés pour prédire la classe du caractère d'entrée. Ils ont testé leur système sur la base de données citée plus haut [8] composé de 19437 caractères, les 2/3 de la base ont servi pour l'apprentissage et le reste pour le test. Ce système a enregistré un taux de reconnaissance de 98,76%.

Oujaoura et al [114] ont amélioré le travail précédent de Bencharef en étendant le vecteur des caractéristique par l'ajout des moments de Zernike. Les auteurs ont présenté une comparaison de différents classifieurs (CART, KNN, SVM, ADaBoost, ANFIS et le PMC) sur une base de 2000 images. La meilleure performance a été enregistrée par le PMC avec un taux de 94%. Les mêmes auteurs [115]ont présenté un autre système hybride combinant les différents moments (Hu, Legendre et Zernike), la transformée de Walsh, les descripteurs GIST et certains caractéristiques de texture. Ils ont testé leur vecteur de caractéristiques en utilisant quatre différents classifieurs (KNN, SVM, PMC et les réseaux bayésiens), la décision de classification est celle obtenue par le vote de ces classifieurs. Les expérimentations ont donné un taux de 99,39%, mais malheureusement, ce taux a été obtenu sur une base d'image de très petite taille (165 images, i.e. 5 images par caractères).

Boutaounte et al [26] ont tenté d'extraire les primitives structurelles telles que les arcs et les segments en utilisant les points clés de chaque caractère (Extrémités, jonctions, inflexions) comme montré dans la Figure 59.

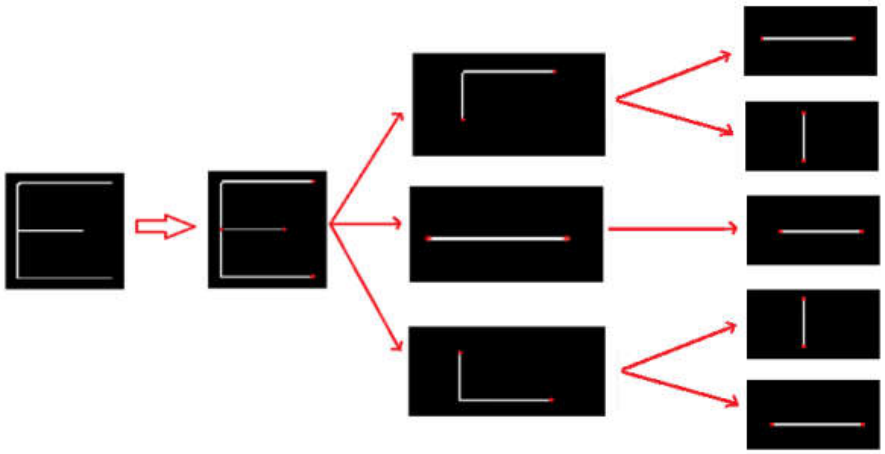


Figure 59. Détection des primitives structurelles

Ensuite, le nombre des points clés par type et le nombre des primitives structurelles par type et orientation sont calculés pour former le vecteur descripteur de chaque caractère. Enfin, les auteurs ont utilisé SVM pour reconnaitre les caractères d'entrée. Ce système a été appris par 2000 images de caractères et a été testé sur 1300 donnant ainsi un taux de reconnaissance de 98,94%.

Ouadid et al [113] ont proposé une approche basée sur les graphes. D'abord, ils ont extrait les points d'intérêt de Harris à partir des squelettes des images des caractères, puis, ils ont

construit des matrices d'adjacence permettant de décrire chaque caractère par un graphe non orienté (Figure 60).

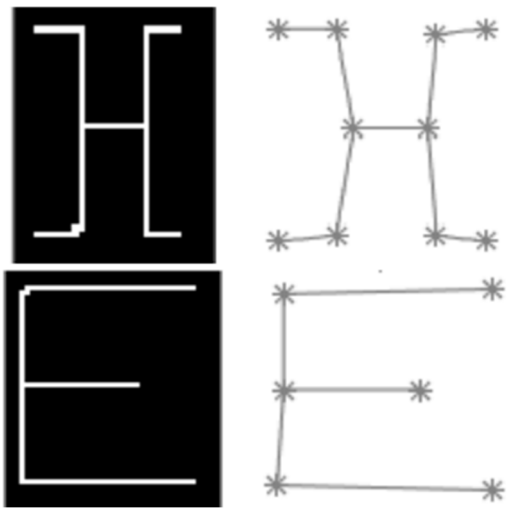


Figure 60. Représentation des caractères par des graphes à base des points d'intérêt

Les auteurs ont utilisé l'analyse spectrale des graphes dans la phase de reconnaissance permettant de trouver le rapport entre la matrice d'adjacence du caractère à reconnaitre et les matrices de références décrivant les 33 caractères de l'alphabet Tifinagh. Les tests ont été menés sur 3267 images des caractères avec un taux de reconnaissance de 99%.

3. 2. **Reconnaissance des caractères manuscrits**

En 1997, Djematene et al [39] ont considéré que la technique proposée par Oulamara et al [116] n'est pas appropriée pour les caractères manuscrits vu qu'ils présentent des traits courbés, les auteurs ont présenté une approche structurelle décrivant chaque caractères à base de ces points d'intérêt (intersections, extrémités, corners) par une chaine code construite sur neuf régions de l'image du caractère. Pour comparer les chaines codes, une distance métrique a été utilisée, les expérimentations ont été menées sur une base d'images locale de 1700 caractères et ont abouti à un taux de reconnaissance de 92,2%.

Ce n'est qu'en 2011 que les chercheurs ont repris les travaux concernant la reconnaissance des caractères amazighs manuscrits où Gounane et al [57] ont proposé une comparaison entre les cartes auto-adaptatives (SOM) et le Fuzzy KNN comme classifieurs dont le vecteur des caractéristiques d'entrée et constitué des 63 densités calculées sur 7x9 zones de l'image du caractère. Les auteurs n'ont pas présenté des résultats expérimentaux numériques mais ils ont affirmé que Le Fuzzy KNN dépasse les cartes auto-adaptatives.

Dans la même année, Essady et al [46] ont développé un vecteur des caractéristiques basé sur les deux lignes centrales, horizontale et verticale (Figure 61).Différentes caractéristiques statistiques ont été calculées sur les différentes zones considérées construisant ainsi un vecteur de 90 composants pour chaque caractère.

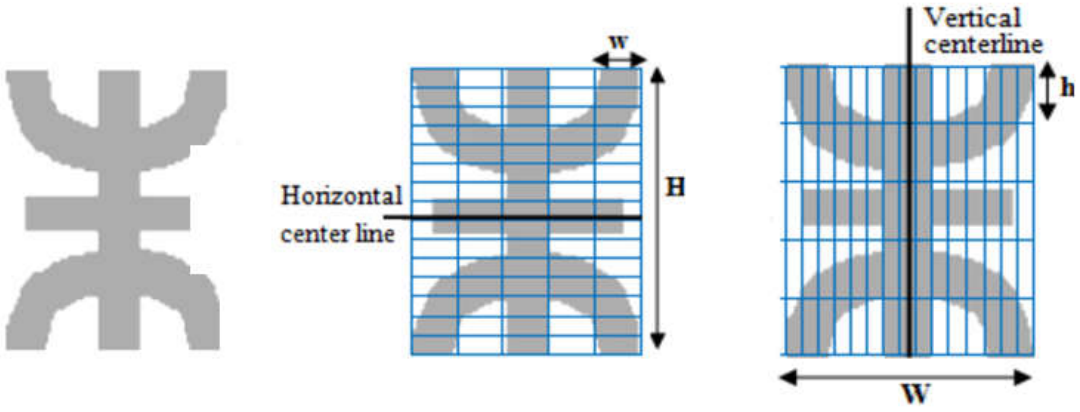


Figure 61. Décomposition suivant les lignes centrales

Les auteurs ont utilisé le PMC dans la phase de reconnaissance. Ce classifieur a été testé sur 20150 caractères de la base des images AMHCD, décrite plus haut, en utilisant la 10-validation croisée. Ce système atteint un taux de reconnaissance de 96,32%. 3 ans plus tard [43], ils ont testé le même système sur toute la base composée de 24180 caractères et ont obtenu un taux de 94,96%.

Amrouch et al [12] ont proposé une approche globale basée sur les modèles de Markov cachées continues et des caractéristiques directionnelles. Le système proposé est une suite d'étapes : les prétraitements ; la squelettisation ; l'extraction des primitives et la classification. Les auteurs ont utilisé le classifieur Viterbi pour reconnaitre les caractères décrits par un vecteur extrait par la transformée de Hough sur les différentes 16x16 zones de l'image de caractère en utilisant une fenêtre glissante, une séquence de numéros, qui correspond aux directions dominantes dans les différentes zones adjacentes, est générée pour décrire chaque caractère (Figure 62).

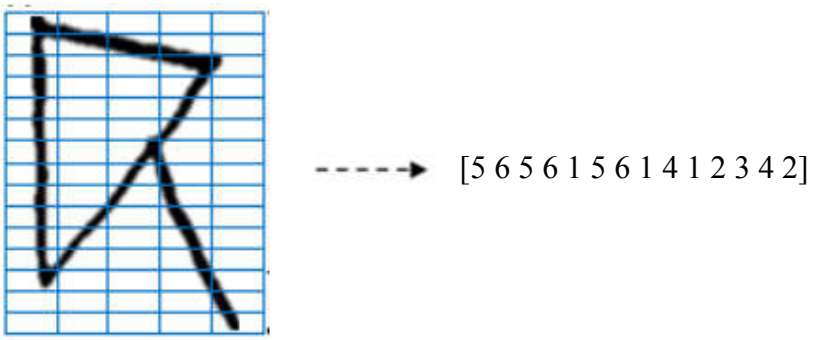


Figure 62. Division de l'image du caractère en cellules et génération de la séquence de numéros

Les numéros émis dans la séquence sont de 1 à 6 correspondants aux 6 orientations (0° ; 30° ; 60° ; 90° ; 120° et 150°).

Le système a été testé sur la base d'images AMHCD, Les 2/3 de la base ont servi pour l'apprentissage et le reste pour le test. Le taux de reconnaissance obtenu est de 97,89%.

Moudni et al [107] ont présenté une étude comparative entre deux descripteurs, celui des caractéristiques robustes accélérées (SURF) et celui de GIST combinés avec le PMC. Pour le

descripteur SURF, les auteurs ont travaillé avec une version réduite (SURF-36) permettant ainsi une rapidité du calcul. Tandis que pour le descripteur GIST, ils ont utilisé l'analyse en composantes principales (ACP) pour réduire la taille du vecteur descripteur. Pour comparer les deux descripteurs, le PMC a été utilisé sur la base des images AMHCD. Les expérimentations ont enregistré un taux de reconnaissance de 75% pour le descripteur SURF et 83% pour celui de GIST.

Le système proposé par Gounane et al [58] présente une phase de post-traitement en utilisant les modèles de langage bi-gramme qui s'appuient sur le lexique amazigh. D'abord, les auteurs ont extrait les caractéristiques à partir de l'image du caractère divisée en 5x5 zones égales, la densité et la distance du centre de gravité par rapport au corner gauche bas de la zone considérée sont calculées. Ensuite, ils ont utilisé le Fuzzy KNN (appris par 1940 caractères manuscrits) pour prédire les candidats de chaque caractère. Enfin, les candidats sont vérifiés par les modèles de langage bi-gramme construits sur un lexique de 1059 mots amazighs. Cette approche aboutit à un taux de reconnaissance de 91,05%. Deux ans plus tard, les auteurs [59]ont enrichi leur travail par l'ajout d'une comparaison des différents classifieurs (Fuzzy KNN, PMC et SVM) et par l'ajout des différents modèles de langage n-grammes (bi-gramme, 3-gramme et 4-gramme). Par cette amélioration le taux de reconnaissance a augmenté à 95,05%.

Abaynarh et al [2] ont utilisé les moments de Legendre, invariants par rapport à la translation, rotation, inclinaison et la mise à échelle, combiné avec le PMC. Afin de déterminer l'ordre d'expansion des moments, les auteurs ont opté pour le principe d'entropie maximale, comme critère de sélection des caractéristiques, pour estimer le nombre optimal des moments. Ils ont testé leur système sur une base d'images développée localement, Cette base a été générée par 57 différents scripteurs qui ont produit 4 exemples par caractère créant ainsi 7524 images des 33 caractères de l'alphabet Tifinagh (Figure 63).

Figure 63. Exemple du formulaire de génération de la base de données

Le PMC à 400 neurones dans la couche cachée a été entrainé par 6600 images et testé sur 924 images. Le taux de reconnaissance obtenu est de 97,46%.

Les mêmes auteurs [1] ont amélioré leur travail par étendre le vecteur descripteur en ajoutant des caractéristiques de densité et de distances euclidiennes. Cette amélioration a enregistrée une augmentation du taux sur la même base de données locale à 97, 94%.

Conclusion

Dans un contexte visant la promotion, la préservation et la revitalisation de la langue amazighe largement négligée, les efforts fournis ces dernières années ont permis à cette langue et à son script original, Tifinagh, de jouir d'un statut meilleur. La standardisation et la normalisation du script et son introduction dans les nouvelles technologies de l'information et de communication sont des exemples de cet épanouissement, néanmoins, les travaux réalisés en ce qui concerne son traitement automatique, en particulier les systèmes OCR sont loin d'être satisfaisants en les comparant avec ceux réalisés pour ses confrères tels que le latin, le chinois, l'arabe, l'indien, etc. La partie suivante décrit les contributions proposées pour surmonter certaines limites liées au domaine de la reconnaissance automatique hors ligne de l'écriture Amazighe manuscrite et imprimée.

Partie. 2 : Contributions à la reconnaissance hors ligne de l'écriture Amazighe imprimée et manuscrite en Tifinagh

Simuler la lecture humaine par une machine était toujours une préoccupation pour les chercheurs pendant ces dernières décennies. Dans ce sens, beaucoup de langues ont bénéficié du développement technologique et des nouveaux algorithmes puissants pour concevoir des systèmes capables d'atteindre un certain niveau de lecture assez intéressent. Cependant, les travaux réalisés pour la langue Amazigh sont loin de la perfection et le domaine de la reconnaissance de l'écriture amazighe manuscrite ou imprimée est devenu actuellement un sujet très actif. Cette langue souffre encore de l'absence des systèmes de reconnaissance automatique de son script Tifinagh manuscrit ou imprimé qu'on peut juger performants, mais aussi des bases d'images de référence permettant l'évaluation et la comparaison fiables des systèmes développés.

Les travaux réalisés dans le cadre de cette thèse visent à surmonter certaines des limites rencontrés en survolant la littérature. Cette partie présente nos différentes contributions apportées dans le domaine de la reconnaissance automatique hors ligne de l'écriture Amazighe, en Tifinagh, imprimée et manuscrite.

Chapitre. 3 : Reconnaissance automatique hors ligne des caractères manuscrits Amazighs en Tifinagh

Introduction

Ce chapitre présente les travaux réalisés dans le but de concevoir un système robuste de reconnaissance automatique hors ligne de l'écriture Amazighe manuscrite en Tifinagh. Premièrement, nous avons développé un système de reconnaissance automatique des caractères amazighs manuscrits qui se caractérise par la proposition d'un nouveau descripteur statistique extrait à partir de l'image du caractère en la décomposant en plusieurs zones chevauchées. Ce descripteur sera amélioré dans un travail ultérieur fournissant ainsi un système de reconnaissance robuste pour les caractères amazighs manuscrits. Les deux descripteurs ont été testés sur une base d'image de référence des caractères Amazighs manuscrits appelée AMHCD obtenue en emailant les auteurs [44].

I. **Reconnaissance des caractères Amazighs manuscrits par une nouvelle méthode de zonage et un réseau de neurones artificiels**

La reconnaissance optique de l'écriture manuscrite est l'un des sujets les plus actifs dans le domaine de la reconnaissance des formes. En effet, les chercheurs ont vite constaté l'importance de ce domaine vu ses divers applications et ont développé plusieurs techniques et systèmes OCR pour différentes scripts (latin, arabe, farsi, chinois, indien, etc.). Cependant, peu de travaux ont été consacrés à la l'écriture Amazighe manuscrite et les performances des systèmes déjà développés dans ce contexte ne répondent pas aux exigences ou ne sont pas assez satisfaisants.

Dans ce travail, nous avons développé un système de reconnaissance automatique de l'écriture Amazighe manuscrite en s'appuyant sur des méthodes statistiques pour extraire des caractéristiques discriminantes à partir de l'image de chaque caractère. Avant l'extraction des caractéristiques, l'image d'entrée subit certains prétraitements nécessaires pour récupérer le texte présent dans l'image et isoler les caractères. Les caractéristiques extraites à partir de l'image du caractère servent comme entrée pour un perceptron multicouches entrainé qui décide de la classe du caractère.

Durant la phase de l'extraction des caractéristiques, l'image du caractère est décomposée en plusieurs zones chevauchées suivant différentes directions, ensuite, dans chaque zone, des caractéristiques statistiques de densité et des profils des projections sont calculées. Le choix des méthodes statistiques lors de l'extraction des caractéristiques est du à leur rapidité et leur faible complexité par rapport aux autres approches [16].

Les résultats obtenus ont été assez satisfaisants avec un taux de reconnaissance de 96,47 % sur la base de données AMHCD composée en totalité de 25740 images des caractères amazighs isolés. Dans ce qui suit, nous allons présenter les différentes étapes utilisées lors du développement du système de reconnaissance automatique de l'écriture Amazighe manuscrite, ainsi que les résultats obtenus.

1. Système proposé

Pour répondre aux exigences concernant la reconnaissance automatique de l'écriture manuscrite de la langue Amazigh à partir des images, nous avons proposé un système composé de plusieurs étapes (Figure 64).

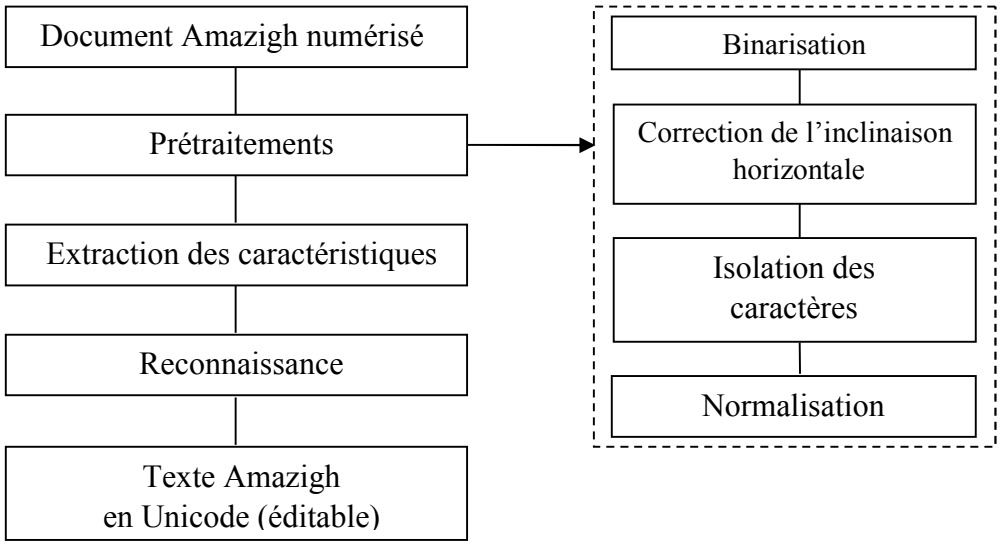


Figure 64. Système OCR Amazigh adopté

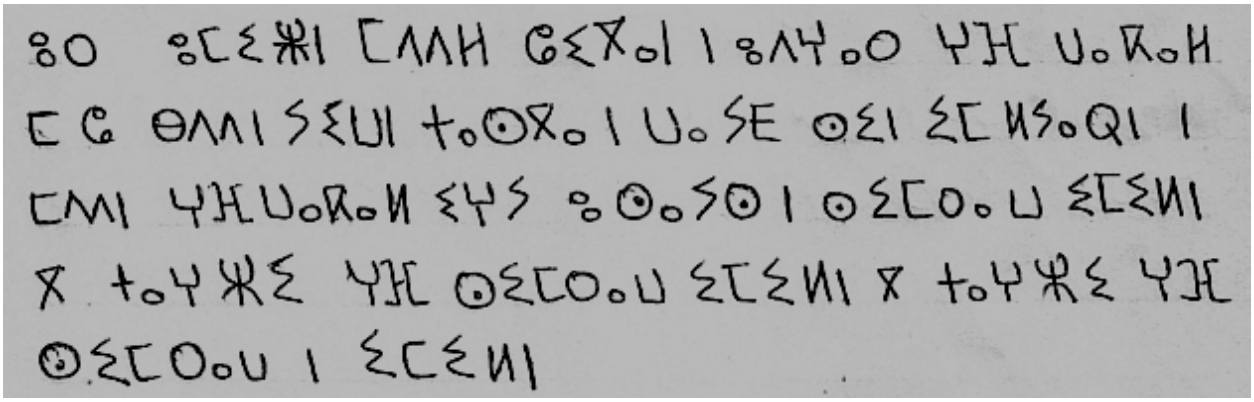
Dans cette partie, les différentes phases du système adopté seront détaillées ainsi que la nouvelle approche statistique développée pour l'extraction des caractéristiques à partir des images des caractères isolés pour les préparer à la phase de reconnaissance.

1. 1. Prétraitements

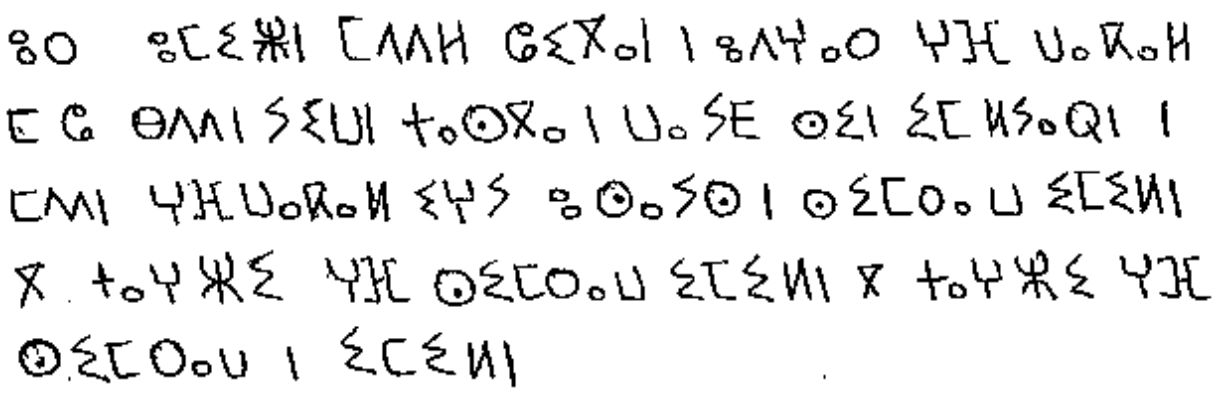
En réalité, le document d'entrée à reconnaitre n'est pas toujours prêt pour le traitement et présente plusieurs lacunes nécessitant des prétraitements (élimination du bruit, binarisation, correction de l'inclinaison, etc.). L'image subit plusieurs transformations pour préparer le texte de l'image à la phase de reconnaissance.

1. 1. 1. Binarisation

Dans ce travail, nous n'avons considérer que les documents texte manuscrits où la séparation du texte de l'arrière-plan est simple et ne nécessite qu'une opération de binarisation. Le résultat de cette opération est une image binaire où les pixels noirs présentent le texte et les pixels blancs définissent l'arrière-plan comme montré dans la Figure 65.



(a)



(b)

Figure 65. Résultat de la binarisation par la méthode d'Otsu

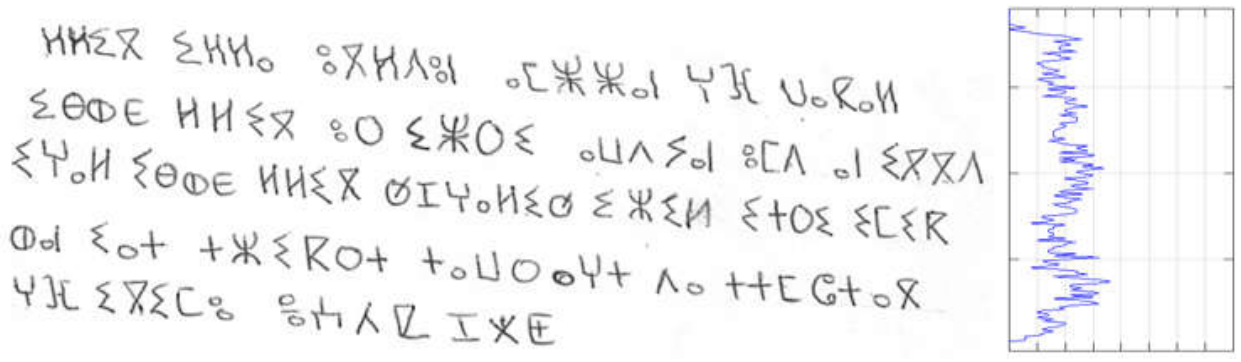
(a) Image originale

(b) Image binarisée

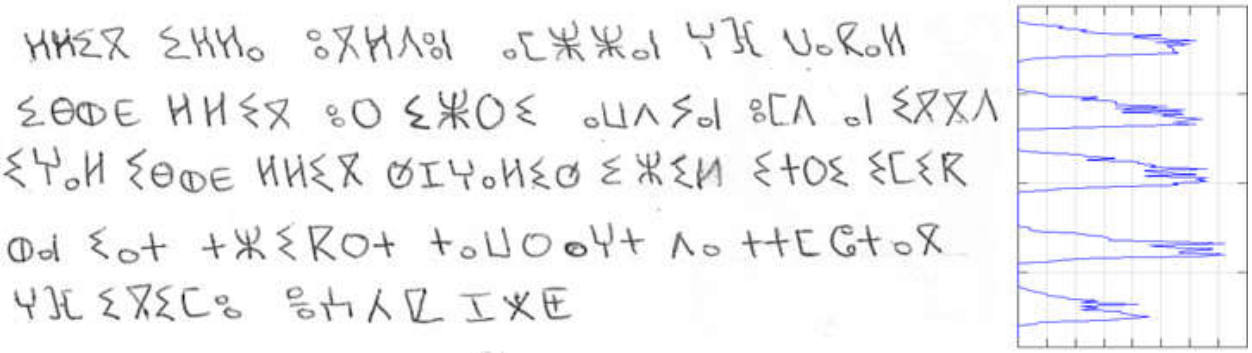
La méthode de binarisation utilisée, dans ce travail, est la méthode d'Otsu [112] qui est très souvent utilisé dans le domaine de l'OCR. Cette méthode non paramétrique et non supervisée effectue un seuillage automatique qui consiste à itérer à travers toutes les valeurs de seuil possibles et à calculer pour chaque seuil la variance intra-classe pondérée entre les niveaux de gris des pixels de chaque côté du seuil. La valeur du seuil qui minimise cette variance est choisie comme seuil. Cet algorithme est facile à implémenter et produit des bons résultats (Figure 65).

1. 1. 2. **Correction de l'inclinaison horizontale**

Pour corriger l'inclinaison horizontale du texte, phénomène souvent apparu après la numérisation des documents, une rotation de l'image est effectuée pour aligner le texte à l'axe horizontal (Figure 66). Néanmoins, l'estimation de l'angle de rotation n'est pas triviale. Dans ce contexte, l'image est scannée selon un intervalle des angles et l'histogramme de projection horizontale est calculé pour chaque angle. L'angle qui minimise la largeur des bandes de l'histogramme est choisi comme angle d'inclinaison [15].



(a)



(b)

Figure 66. Correction de l'inclinaison horizontale du texte

(a) Image originale inclinée

(b) Image après correction

Après avoir obtenu l'angle d'inclinaison, le texte est aligné par la simple transformation mathématique de rotation, c'est-à-dire :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Le centre des images à rectifier est choisi comme centre de rotation et la taille de ces images après la rotation sera modifiée pour éviter toute perte d'informations. En outre, la rotation peut générer des pixels dont les cordonnées sont non entiers ce qui nécessite un algorithme d'interpolation pour calculer la valeur du pixel en question [51]. Il faut noter que la correction de l'inclinaison verticale n'est pas prise en considération dans ce travail.

1. 1. 3. **Segmentation du texte**

Dans l'absence d'un grand lexique ou corpus de la langue Amazighe, la reconnaissance automatique des mots Amazigh en utilisant l'approche holistique est quasi-impossible, rendant ainsi la segmentation des mots en caractères une tâche obligatoire. Le processus de segmentation commence par séparer le bloc de texte en lignes, puis les lignes en mots et enfin les mots en caractères [Kaur2015].

a. Détection des lignes

Pour détecter et extraire les lignes de texte dans les images des documents manuscrits Amazighs, le profil de projection horizontale est généralement utilisé, les pics du profil désignent les lignes de texte et les vallées indiquent les espaces interlignes (Figure 67).

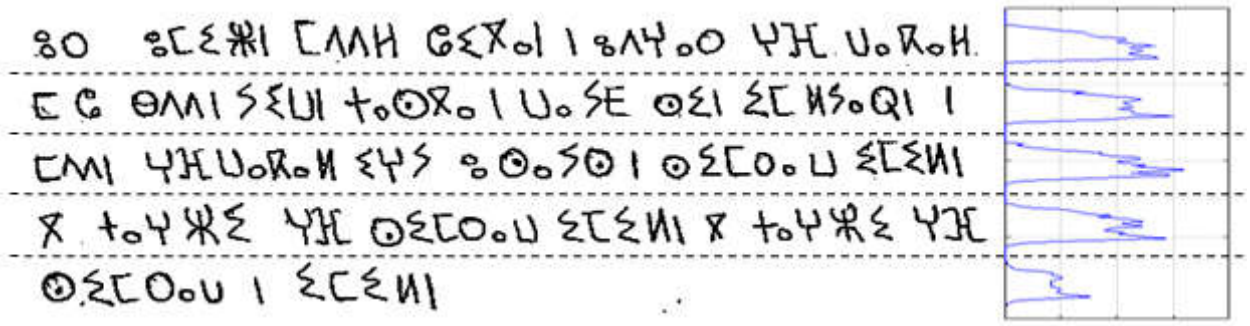
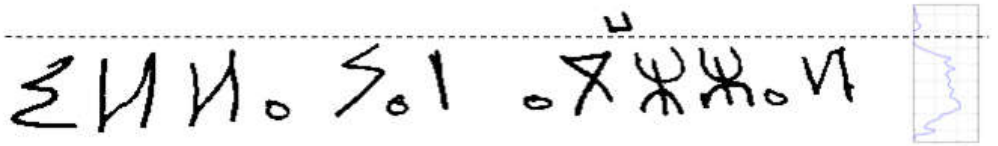
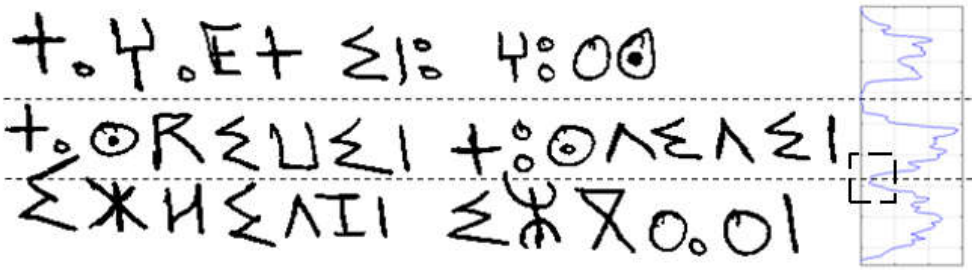


Figure 67. Détection des lignes dans un bloc de texte

Cependant, en pratique dans l'écriture manuscrite sans contrainte, la présence de l'une des deux labiovélaires (□□et □□) dans certains mots ou la connexion entre les lignes adjadentes peut perturber l'extraction des lignes et causer la division d'une ligne en deux ou la fusion des lignes en une seule ligne comme le montre la Figure 68.



(a)



(b)

Figure 68.Difficultés de segmentation par le profil de projection horizontale

(a) Division d'une ligne en deux lignes

(b) Fusion de deux lignes en une seule

Pour remédier à ces phénomènes, une analyse profonde du profile est nécessaire en se basant sur les densités des pixels noirs dans les différentes lignes.

b. Segmentation des lignes en mots

Avant de segmenter les caractères, il faut d'abord segmenter la ligne en mots. Ceci est généralement effectué en utilisant le profil de projection verticale. On rencontre rarement des

problèmes dans ce contexte vu que les mots sont souvent séparés par un espace assez large (Figure 69).

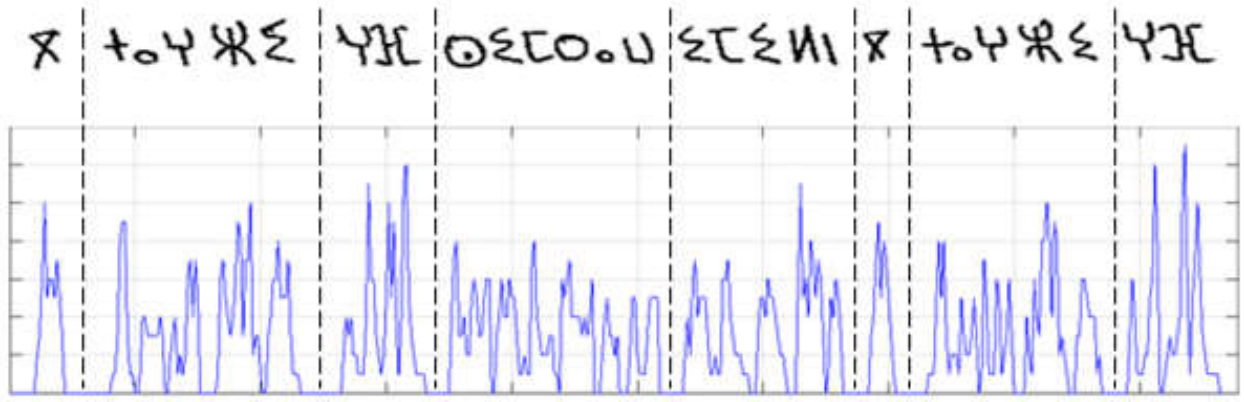


Figure 69. Segmentation d'une ligne en mots par le profil de projection verticale

c. Segmentation des mots en caractères

Parmi les propriétés principales de l'écriture manuscrite ou imprimée amazighe est la non cursivité. Ceci facilite la segmentation des mots en caractères. L'histogramme de projection verticale est souvent utilisé, les lettres correspondent aux zones de forte densité et les zones de basse densité indiquent les régions de segmentation (Figure 70).

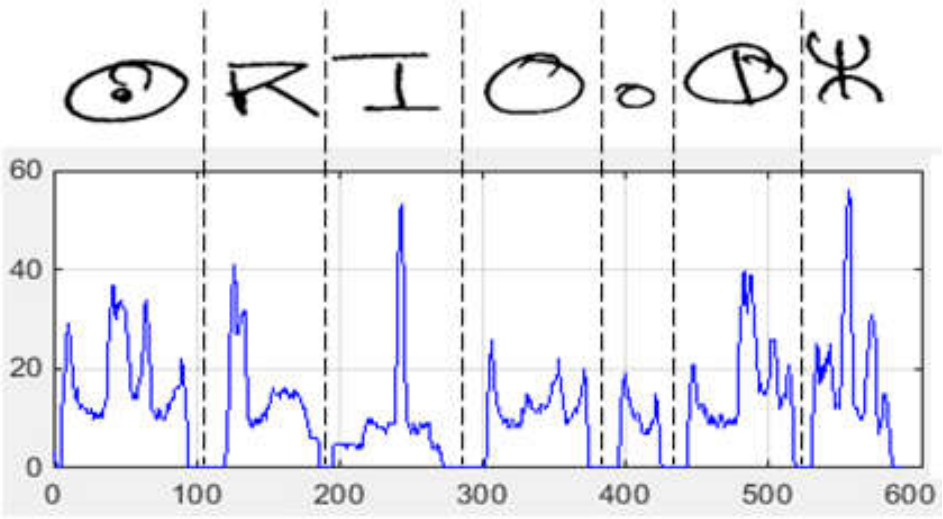
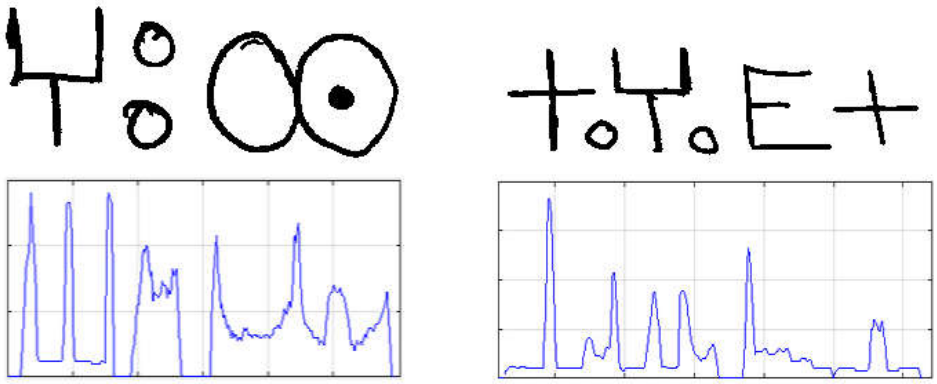


Figure 70. Segmentation des caractères par l'analyse du profile de projection verticale

Cette méthode peut échouer face aux mots présentant des caractères touchés ou chevauchés, phénomènes souvent rencontrées dans l'écriture manuscrite, nécessitant ainsi des techniques développées de segmentation. La Figure 71 présente des exemples de ces phénomènes.



(a) (b)

Figure 71. Phénomènes de segmentation présentant des problèmes

(a) Profile de projection d'un mot présentant des caractères touchés

(b) Profile de projection d'un mot présentant des caractères chevauchés

1. 1. 4. **Normalisation des caractères**

La segmentation des mots en caractères produit des images des caractères isolés et de tailles différentes. Pour s'assurer de l'uniformité des caractères et d'une meilleure performance lors de l'extraction des caractéristiques, les images des caractères isolés sont redimensionnées à une taille commune de 30x30. Cette taille s'adapte bien aux techniques de zonage que nous avons proposé durant l'extraction des caractéristiques.

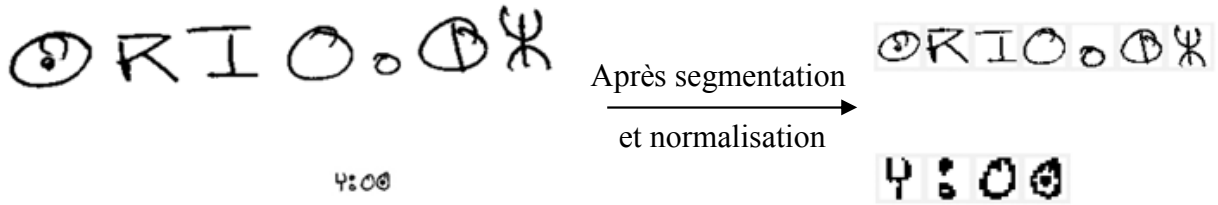


Figure 72. Normalisation des caractères

Dans ce travail, nous avons utilisé la méthode proposée par [108], cette méthode se base sur une transformation géométrique et un ré-échantillonnage en utilisant une interpolation B-spline, elle produit des bons résultats et permet de réduire les artefacts et d'améliorer significativement le rapport signal/bruit (Figure 72).

1. 2. **Extraction des caractéristiques**

L'extraction des caractéristiques pertinentes, permettant de distinguer chaque caractère, influence largement la performance d'un système OCR. Dans ce contexte, le but de ce travail est de proposer un nouveau descripteur capable de distinguer chaque caractère par rapport aux autres. Les composants de ce descripteur sont calculés par des méthodes statistiques en décomposant l'image carrée du caractère en plusieurs zones chevauchées, la densité du caractère et la longueur maximale des bandes du profile de projection sont calculés dans chaque zone.

Les décompositions conçues (Figure 73) ont été proposées après plusieurs expérimentations, elles visent à prendre en compte la structure des caractères Amazighs qui sont

constitués des arcs et des traits horizontaux, verticaux ou diagonaux. Le choix des caractéristiques de densité et du profile de projection est du à leur rapidité et leur facilité d'implémentation. Dans ce qui suit, nous allons détailler le travail effectué pour extraire les composants du descripteur.

1. 2. 1. **Caractéristiques de densité**

Pour créer le premier sous ensemble du vecteur descripteur, nous avons décomposé l'image du caractère en 37 zones chevauchées suivant différentes directions (Figure 73), ensuite, la densité du caractère dans chaque zone est calculée. Cette dernière est obtenue en divisant le nombre des pixels noirs sur le nombre total des pixels de la zone considérée. Les 37 caractéristiques de densité obtenues fournissent une description de la dispersion spatiale du caractère.

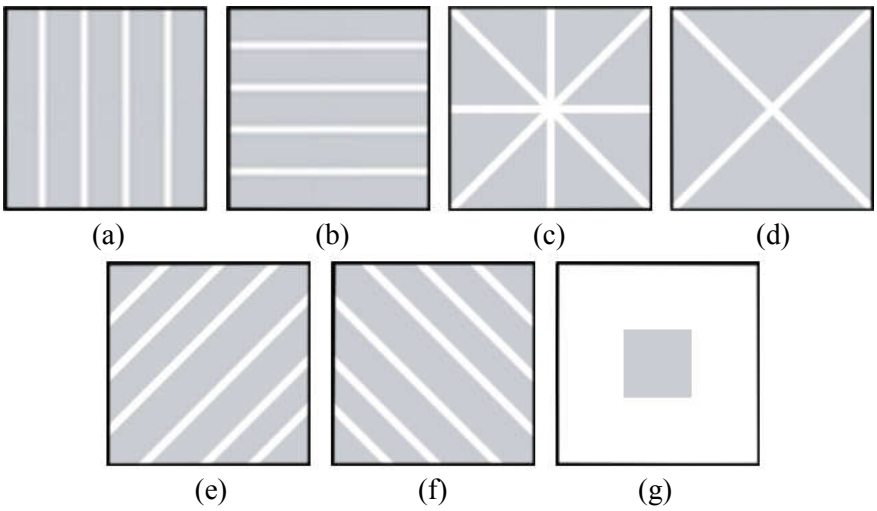


Figure 73. Décompositions proposées de l'image du caractère pour les caractéristiques de densité

(a) Décomposition verticale en 5 zones égales
(b) Décomposition horizontale en 5 zones égales
(c) Décomposition en 8 octants
(d) Décomposition en 4 quadrants
(e) Décomposition diagonale droite en 7 zones
(f) Décomposition diagonale gauche en 7 zones
(g) Zone centrale de taille 10x10

Comme nous l'avons mentionné ci-dessus, les décompositions effectuées ont été proposées pour s'adapter avec la structure des caractères Amazighs et prendre en compte les traits horizontaux, verticaux et diagonaux constituants les caractères. La zone centrale de taille 10x10 a été ajoutée pour distinguer entre quelques caractères presque similaires.

1. 2. 2. **Caractéristiques d'ombre**

Le deuxième sous ensemble du descripteur, constitué de 42 composants, est obtenu en sélectionnant, dans chaque zone, la valeur maximale des longueurs des bandes du profile de projection suivant différentes directions.

La Figure 74 montre les différentes décompositions considérées et les différentes directions de projection adoptées.

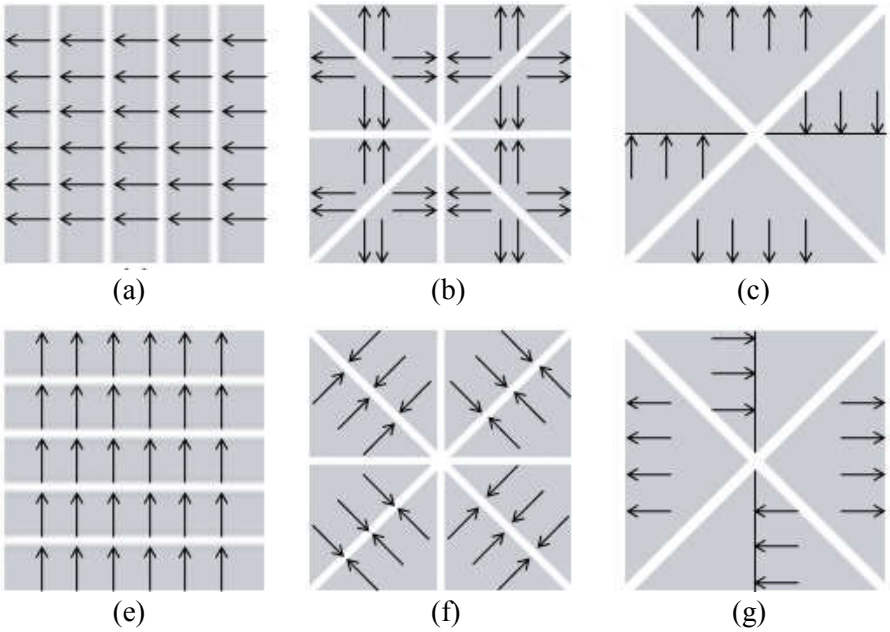


Figure 74. Décompositions proposées de l'image du caractère pour les caractéristiques du profile des projections

(a) Décomposition verticale en 5 zones égales et projection horizontale
(b) Décomposition horizontale en 5 zones égales et projection verticale
(c) Décomposition en 8 octants et projections horizontale et verticale
(d) Décomposition en 8 octants et projection diagonale
(e) Décomposition en 4 quadrants et projection verticale
(f) Décomposition en 4 quadrants et projection horizontale

Chaque valeur calculée doit être normalisée en la divisant sur la longueur maximale possible du côté correspondant à la projection (Figure 75).

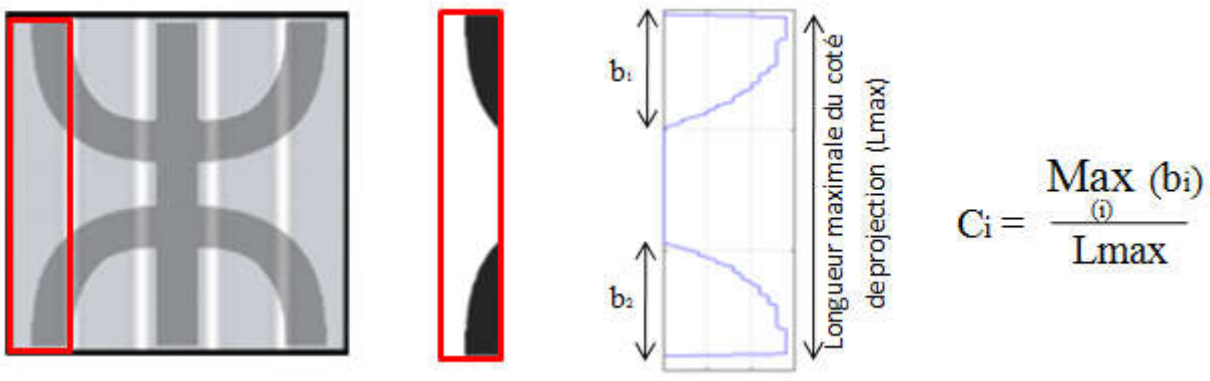


Figure 75. Extraction de la longueur maximale des bandes du profile de projection

Les caractéristiques de l'ombre extraites des profils des projections fournissent des informations sur la structure des caractères.

1. 3. Phase de reconnaissance des caractères

L'extraction des caractéristiques à partir de l'image normalisée de chaque caractère isolé produit un vecteur de 79 composants, ce vecteur est utilisé comme entrée par un classifieur qui décide la classe du caractère parmi les classes de l'alphabet Amazigh. La littérature est très riche par différents types de classifieurs. Dans ce travail, nous avons utilisé le perceptron multicouches (PMC) qui est largement utilisé dans le domaine de l'OCR et l'un des classifieurs les plus performants et les plus connus dans la famille des réseaux de neurones (voir section III. 1. 2. du chapitre 1). L'architecture utilisée du PMC complètement connecté consiste en trois couches (Figure 76) :

- Couche d'entrée : cette couche est composée de 79 neurones qui correspondants aux 79 composants du descripteur caractérisant chaque caractère ;
- Couche cachée : le nombre de neurones de cette couche est choisi expérimentalement ;
- Couche de sortie : contient 31 neurones qui représentent les 31 caractères de la langue Amazigh étudiés dans ce travail.

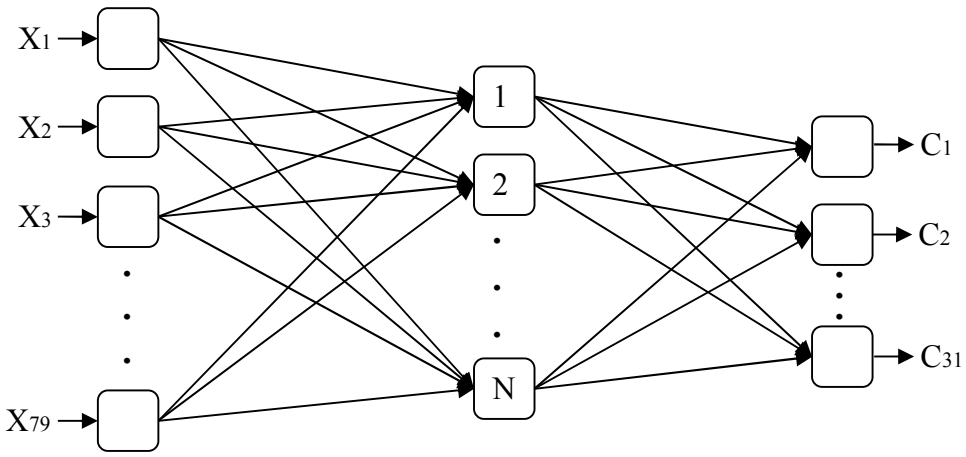


Figure 76. Architecture utilisée du perceptron multicouches

Il faut noter qu'une seule couche cachée peut résoudre les problèmes non linéaires et que le choix optimal du nombre de neurones des couches cachées est toujours un défi [76].

2. Résultats et discussions

Dans cette partie, nous allons décrire les données de test utilisées et la démarche suivie pour tester la performance du descripteur proposé dans ce travail.

2. 1. Base de données de test

Dans le but d'évaluer la performance du système proposé, la base de données AMHCD a été utilisée comme source d'apprentissage et de test. Cette base de données des caractères manuscrits isolés se compose en totalité de 25740 images des 33 caractères de la langue Amazighe. Généralement, les systèmes OCR Amazighs excluent les deux labiovélaires yagw (□□) et yakw (□□) qui n'ont pas de codage Unicode et qu'on peut les obtenir en combinant entre les caractères yag (□) ou yak (□) et le signe de labellisation (□). Par conséquent, les

expérimentations ont été effectuées sur 24180 images dont 70% (16926 images) ont été utilisé pour l'apprentissage et le reste (7254 images) pour le test. Il faut noter que la base de données contient le même nombre d'images pour chaque caractère et que l'ensemble d'apprentissage et de test sont composés respectivement de 546 et 234 images de chaque caractère.

2. 2. **Configuration de perceptron multicouche**

Les expérimentations ont été conduites sur une machine HP, Intel (R) Core (TM), Duo CPU 1.4 GHz, et 2 GB de mémoire RAM. Différentes architectures du PMC ont été entrainé par l'algorithme itératif de rétro-propagation du gradient [156] en variant le nombre de neurones de la couche cachée à chaque fois en utilisant la configuration suivante :

- Fonction d'activation : la fonction sigmoïde ;
- Nombre d'itérations : 16926 itérations ;
- Taux d'apprentissage : 0,1 ;
- Élan : 0,25 ;
- Poids des connexions : initialiser aléatoirement dans l'intervalle [-0,7 ; 0,7].

2. 3. **Résultats et discussions**

Les expérimentations menées sur la base de données AMHCD selon les différentes architectures du PMC citées ci-dessus, et en utilisant le descripteur proposé dans ce travail, ont produit les résultats reportés dans la Table 4.

Selon les résultats de la Table 4, la meilleure performance du PMC est obtenue lorsque le nombre de neurones de la couche cachée est à 95.

Dans ces expériences nous n'avons pas dépassé 100 neurones dans la couche cachée pour éviter le sur-apprentissage et choisir un nombre de neurones qui fournit un compromis entre la performance est le temps de reconnaissance, ainsi, nous avons opté pour l'architecture 79-95-31 qui nous a permis d'atteindre un taux de reconnaissance de 96,47% et un temps raisonnable lors de la phase de reconnaissance de 4ms par caractère.

Table 4. Résultats des différentes architectures du PMC

Architecture du PMC	Précision (%)	Recall (%)
79-55-31	95,60	95,6
79-60-31	95,79	95,8
79-65-31	95,92	95,9
79-70-31	96,01	96
79-75-31	96,25	96,2
79-80-31	96,19	96,2
79-85-31	96,22	96,2
79-90-31	96,26	96,3
79-95-31	**96,47**	**96,5**
79-100-31	96,46	96,5

En se basant sur cette architecture et sur le descripteur proposé, nous avons calculé la performance du système au niveau de chaque classe de caractères. La Table 5 reporte les résultats obtenus.

Table 5. Performance du système pour chaque classe de caractères

Caractère	Taux individuel de reconnaissance (%)
ⵄ	100
ⵅ	84.0
ⵃ	97.7
ⵇ	94.15
ⵜ	97.54
ⵣ	95.23
ⵀ	97.23
ⵂ	96.92
ⵛ	98.46
ⵅ	98.77
ⵣ	96.62
ⵙ	93.85
ⵍ	99.38
ⵎ	97.85
ⵏ	95.38
ⵔ	97.85
ⵉ	98.15
ⵅ	97.38
ⵉ	97.38
ⵛ	98.92
ⵡ	99.85
ⵅ	92.46
ⵛ	98.15
ⵖ	97.69
ⵙ	98.15
ⵏ	99.85
ⵅ	86.0
ⵔ	96.15
ⵜ	90.77
ⵜ	94.62
ⵟ	93.69

En analysant les résultats montrés dans la Table 5, nous avons remarqué que certains caractères présentent un taux de reconnaissance relativement bas par rapport aux autres classes, surtout pour les caractères yaz (ⵅ), yatt (ⵜ), yazz (ⵅ) et yas (□). Les erreurs de classification remarquées sont dues à deux facteurs. Le premier facteur est la similarité structurelle entre certains caractères (yax (ⵅ), yaz (ⵅ) et yazz (ⵅ) ; yatt (ⵜ) et yadd (ⵜ) ; yas (ⵙ) et yar (□) ; yay (□) et yi (□) ; etc.). La Table 6 montre la matrice de confusion entre les différentes classes. Rappelons que chaque classe contient 234 images du caractère en question.

Table 6. Matrice de confusion entre les différentes classes

	ⵔ	ⵣ	ⵢ	ⵇ	ⵜ	ⵥ	ⵒ	ⵄ	ⵀ	ⵅ	ⵍ	ⵇ	ⵏ	ⵎ	ⵣ	ⵔ	ⵉ	ⴽ	ⵖ	ⵔ	ⵛ	ⵣ	ⵚ	ⵜ	ⵟ	ⵡ	ⵓ	ⵔ	ⵕ	ⵙ	ⵗ	ⵘ
ⵔ	234																															
ⵣ		202													3		2					4					23					
ⵢ			229		1		3	1																								
ⵇ				221								7								1	1							3				
ⵜ		1			229																					1	3					
ⵥ			1			222		8		2					1																	
ⵒ			4				228										1						1									
ⵄ						3		229									2															
ⵀ					1				232										1													
ⵅ									1	231			1														1					
ⵍ				1							227							4					1						1			
ⵇ				13								217				1								2			1					
ⵏ										1			232						1													
ⵎ					1									231					1										1			
ⵣ		1				1							1		224			3				2					2					
ⵔ												3				226		1										2				
ⵉ								2									230			1						1						
ⴽ		2									2			1				224	1										4			
ⵖ													3	1					226		3			1								
ⵔ								1									2			230										1		
ⵛ																			1		233											
ⵣ		6								1			1									222					4					
ⵚ				2														1					230					1				
ⵜ					2														1		1			230								
ⵟ												3													229			1				1
ⵡ																										234						
ⵓ		12	2		2																	1					217					
ⵔ				4								2				1										1		226				
ⵕ										1	2	1													1				213	16		
ⵙ								1													7								4	222		
ⵗ				4												1		2					3	4			2				218	

Le deuxième facteur est la dysgraphie ou la mauvaise écriture de certains caractères dans la base de données de test dont leur classification est difficile même pour un humain, la Figure 77 montre quelques images de caractères mal écrits contenues dans la base de données de test.

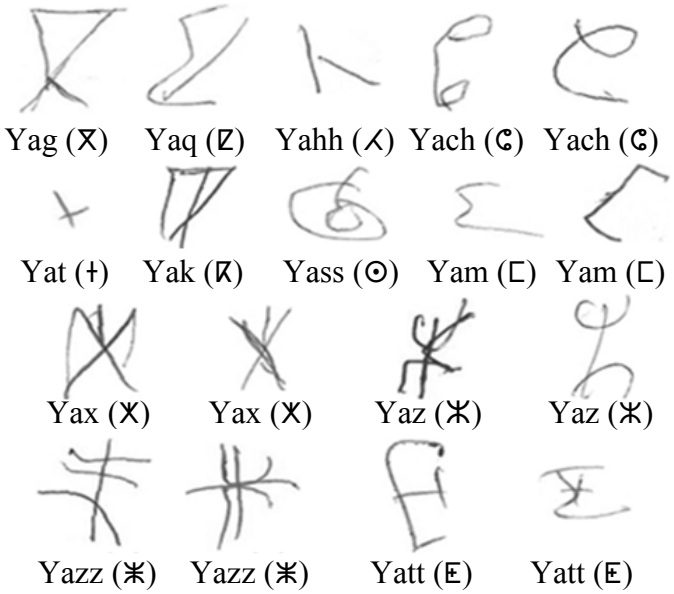


Figure 77. Exemples de caractères mal écrits dans la base de données

Pour mettre en valeur notre approche, nous avons comparé nos résultats avec ceux obtenus par les meilleures approches développées dans la littérature comme le montre Table 7.

Table 7. Comparaison des résultats avec les différentes approches de la littérature

Approches	**Taux de reconnaissance (%)**	**Taille des données d'apprentissage**	**Taille des données de test**
Approche proposée	96,47	16926	7254
Amrouch et al [12]	**97,89**	**16120**	**8060**
Es Saady et al [46]	96,32	18135	2015
Es Saady et al [43]	94,62	24180	Validation croisée
Djematene et al [39]	92.30	1000	700
Gounane et al [58]	91.05	1940	Non spécifié

Comme montré dans la Table 7, le meilleur taux de reconnaissance obtenu est enregistrée par Amrouch et al [12] qui ont utilisé une approche structurelle basée sur la transformée de Hough et les modèles de Markov cachées. Un autre travail mérite d'être mentionné est celui de Es Saady et al [43] où ils ont développé un descripteur statistique basé sur la ligne centrale des caractères, ce système a été testé deux fois sur la base d'images AMHCD et a donné des bons résultats. Les autres auteurs ont testé leur systèmes sur des bases d'images locales ce qui empêche une comparaison fiable des résultats.

II. Un ensemble robuste de caractéristiques pour les caractères Amazighs manuscrits

Dans le même cadre des systèmes de reconnaissance automatique hors ligne de l'écriture Amazighe manuscrite, les résultats obtenus lors du travail proposé précédemment ou ceux des travaux existants dans la littérature restent assez satisfaisants mais ne répondent pas aux exigences en termes de taux et de temps de traitement.

En analysant les résultats du travail précédent, nous avons constaté que les erreurs de classification ont été généralement dues à la ressemblance structurelle entre certains caractères et que le descripteur développé nécessite certaines améliorations pour la surpasser. En outre, au lieu d'utiliser le PMC seulement comme classifieur, nous avons utilisé quatre classifieurs très connus dans le domaine de la classification supervisée afin de tester la robustesse du descripteur amélioré proposé mais aussi pour bénéficier de leur combinaison lors de la phase de reconnaissance. Enfin, l'option du rejet a été utilisée comme post-traitement pour améliorer la précision du système.

Les résultats obtenus ont été très satisfaisants. La précision atteinte sur la même base de données AMHCD est de 99,03% sans rejet et de 99,10 % avec rejet tout en gardant un taux de rejet raisonnable (0,37%).

Dans ce qui suit, nous allons présenter les différentes améliorations apportées dans ce travail ainsi que les résultats obtenus.

1. Améliorations proposés

Dans cette partie, nous allons présenter les apports effectués par rapport au travail précédent pour améliorer les résultats et construire un système OCR Amazigh robuste. En effet, plusieurs améliorations ont été effectuées dans les différentes phases du système, à savoir, l'extraction des caractéristiques, la reconnaissance et le post-traitement.

1. 1. Descripteur amélioré lors de l'extraction des caractéristiques

Le taux de reconnaissance (96,47%) obtenu dans le travail précédent en utilisant le descripteur proposé a été jugé assez satisfaisant par rapport aux taux existants. Rappelons que nous avons utilisé des méthodes statistiques pour extraire les caractéristiques à partir des images des caractères en les décomposant en plusieurs zones chevauchées. Le descripteur proposé contient en totalité 79 composants constitués de deux sous-ensembles, Le premier est constitué de 37 caractéristiques de densité et le deuxième de 42 caractéristiques extraites à partir du profile des projections suivant différentes directions. La Figure 78 rappelle les différentes décompositions proposées et leurs relatives directions de projection, pour plus de détails concernant ce descripteur, veuillez-vous reporter à la section I. 1. 2. de ce chapitre.

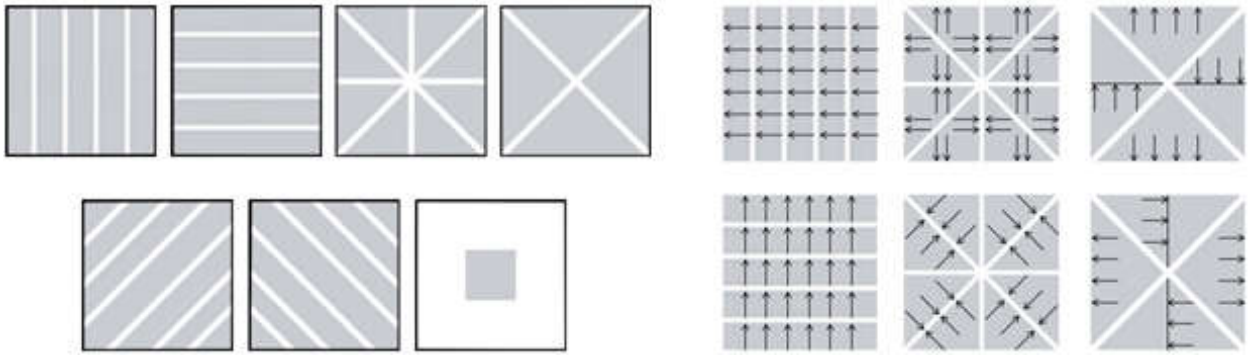


Figure 78. Rappel des différentes zones et projections utilisées pour extraire les caractéristiques

(a) Zones pour les caractéristiques de densité

(b) Zones pour les caractéristique du profile des projections

Dans le nouveau descripteur proposé, nous avons suivi les mêmes décompositions et les mêmes orientations de projection décrites dans la Figure 78, ce qui nous a mené à préserver la même taille de notre vecteur de caractéristiques. Nous avons construit le premier sous ensemble de 37 caractéristiques de densité de la même manière qu'au précédent puisqu'elle nous a permis d'extraire des informations sur la dispersion spatiale du caractère. Tandis que pour le deuxième sous ensemble, nous avons remarqué que le fait de sélectionner seulement la valeur maximale des largeurs des bandes des profils des projections ne reflète pas l'extension du caractère dans la zone considérée et conduit à une grande perte d'informations vu qu'on a ignoré les autres largeurs des bandes. Par conséquence, nous avons proposé l'utilisation de la somme des largeurs des bandes au lieu de la valeur maximale (Figure 79).

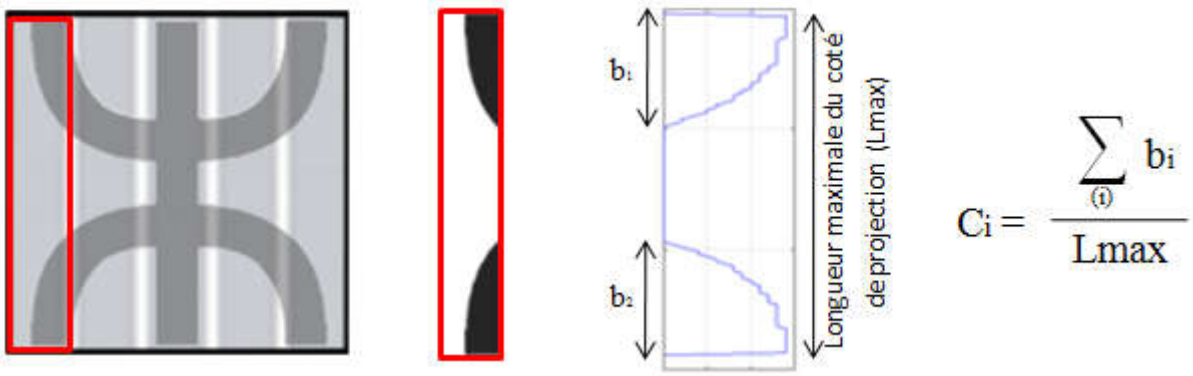


Figure 79. Extraction de la somme des longueurs des bandes du profile de projection

Ainsi, nous avons obtenu les 42 caractéristiques formant le deuxième sous ensemble. Les résultats de cette amélioration seront présentés ci-dessous.

1. 2. Phase de reconnaissance des caractères

Dans la phase de reconnaissance, pour tester la robustesse du nouveau descripteur proposé, nous avons opté pour les méthodes de classification supervisée les plus utilisées dans la littérature, à savoir : le perceptron multicouches PMC déjà utilisé; les réseaux bayésiens RB; l'algorithme du plus proche voisin KNN et la machine à vecteurs supports SVM.

Pour le PMC, nous avons utilisé la même configuration et la même variation du nombre de neurones de la couche cachée.

L'architecture du graphe du classifieur bayésien influence la décision de classification précise pour une entrée donnée, pour ceci, nous avons utilisés différents algorithmes de construction du réseau bayésien développés dans la littérature. Ces algorithmes ce résument en :

- Réseau bayésien augmenté par arbre;
- Algorithme de recherche tabou;
- Algorithme K2;
- Algorithme Hill Climber.

L'algorithme KNN se base sur le calcul des distances ou des mesures de similarité pour classifier une nouvelle entrée, plusieurs distances ont été proposées. Dans ce travail, nous avons utilisé l'algorithme KNN avec : la distance euclidienne ; la distance de Manhattan ; la distance de Chebychev et la distance de Minkowski.

SVM a été configuré en variant la fonction noyau utilisée entre les noyaux linéaire, polynomial et gaussien RBF.

En outre, pour améliorer les résultats, tous ces classifieurs ont été combinés afin de voter pour la décision finale de classification. En effet, l'utilisation des probabilités de sorties de chaque classe nous a permis de combiner les différents classifieurs selon plusieurs variantes, à savoir : vote par majorité ; vote par maximum, produit ou moyenne des probabilités.

1. 3. Option de rejet comme post-traitement

Le post-traitement est souvent utilisé pour les systèmes OCR pour corriger les erreurs de classification et améliorer les résultats. Ceci repose généralement sur des modèles statistiques construit à partir des lexiques ou dictionnaires correspondants à la langue en question.

Malheureusement pour la langue Amazighe, dans l'absence des ressources linguistiques telles que les dictionnaires de validation ou les modèles statistiques construit sur le vocabulaire de référence, l'option du rejet est une solution qui permet d'améliorer la précision du système en rejetant, selon un seuil, les caractères les plus susceptibles d'être mal classés, néanmoins, certaines classifications correctes peuvent être également rejetées. En effet, garder un taux de rejet raisonnable est indispensable et le choix de ce dernier dépend de l'application.

Dans ce travail, nous avons testé l'option du rejet en utilisant un seuil global pour toutes les classes, mais aussi l'option du rejet avec un seuil local pour chaque classe. Les résultats seront présentés ci-dessous.

2. Résultats et discussions

Toutes les expérimentations ont été effectuées sur une machine HP ProBook, Intel (R) Core (TM) i5-2520M CPU 2.50 GHz, et 4 GB de mémoire RAM en utilisant la base de données AHMCD décrite plus haut comme base d'apprentissage et de test. Les classifieurs ont été appris avec 70% des données de la base et les 30% restantes ont servi pour le test.

2. 1. Performance des différents classifieurs

Dans cette partie, nous allons présenter les performances observées pour les différents classifieurs et leurs différentes configurations. En effet, les tables suivantes reportent des résultats obtenus pour chaque classifieur.

Table 8. Résultats du perceptron multicouches selon différentes architectures

Architecture du PMC	Taux de reconnaissance (%)	Rappel (%)	F-mesure (%)
70	98,41	98,4	98,4
75	98,32	98,3	98,3
80	98,69	98,7	98,7
85	**98,86**	**98,9**	**98,9**
90	98,75	98,7	98,7
95	98,68	98,7	98,7

Table 9. Résultats du réseau bayésien selon différents algorithmes d'apprentissage de structure

Structure du réseau Bayésien	Taux de reconnaissance (%)	Rappel (%)	F-mesure(%)
Naïf augmenté par arbre	96,41	97,2	97,2
Recherche Tabou	96,84	96,8	96,9
Algorithme K2	96,56	96,6	96,6
Algorithme de Hill Climber	**96,85**	**96,9**	**96,9**

Table 10. Résultats de SVM selon différentes fonctions noyaux

Fonction noyau	Taux de reconnaissance (%)	Rappel (%)	F-mesure (%)
Linéaire	97,61	97,6	97,6
Gaussien RBF	98,56	98,6	98,6
Polynomiale	**98,71**	**98,7**	**98,7**

Table 11. Résultats de KNN selon différentes distances

Distance utilisée	Taux de reconnaissance (%)	Rappel (%)	F-mesure (%)
Distance euclidienne	96,18	96,2	96,2
Distance de Manhattan	**96,60**	**96,6**	**96,6**
Distance de Chebyshev	94,11	94,1	94,1
Distance de Minkowski	96,18	96,2	96,2

Table 12. Résultats du classifieur combiné selon différentes méthodes de vote

Méthode de vote	Taux de reconnaissance (%)	Rappel (%)	F-mesure (%)
Vote par majorité	98,76	98,8	98,8
Maximum des probabilités	98,56	98,6	98,6
Produit des probabilités	98.87	98,9	98,9
Moyenne des probabilités	**99.03**	**99,0**	**99,0**

2. 2. **Résumé et discussion**

La Table 13 résume et compare entre les résultats des différents classifieurs combinés avec le nouveau descripteur proposé en ne retenant que la meilleure performance des configurations utilisées pour chaque classifieur. Noter que le temps de test mentionné dans les résultats correspond au temps moyen (en ms) pris par le classifieur pour reconnaitre une entrée.

Table 13. Résultats des différents classifieurs avec le descripteur proposé

Classifieurs	Taux de reconnaissance (%)	Rappel (%)	F-mesure (%)	Temps de test (ms)
PMC	98,86	98,9	98,9	0,36
RB	96,84	96,8	96,9	0,72
KNN	96.60	96,6	96,6	29, 7
SVM	98.71	98,7	98,7	6,69
Combinaison	99.03	99,0	99,0	47,5

On peut remarquer dans la table 13 que l'amélioration apportée au descripteur a donné ses fruits vu l'augmentation du taux de reconnaissance du perceptron multicouches de 96,47% (obtenu par le descripteur initial) à 98,86%.

La meilleure performance enregistrée est celle obtenue par la combinaison des classifieurs, en utilisant la moyenne des probabilités des sorties de chaque classifieur, avec un taux de reconnaissance de 99,03%. La Table 14 montre la matrice de confusion du classifieur combiné, la majorité des erreurs de classification sont dues à la ressemblance structurelle entre certains caractères.

Table 14. Matrice de confusion pour le classifieur combiné

	ⵔ	ⵣ	ⵢ	ⵇ	ⵜ	ⵙ	ⵡ	ⵠ	ⵀ	ⵅ	ⵄ	ⵇ	ⴰ	ⵃ	ⵅ	ⵇ	ⵉ	ⴽ	ⵏ	ⵛ	ⵍ	ⵅ	ⵖ	ⵟ	ⵇ	ⵉ	ⵣ	ⵚ	ⵄ	ⵄ	ⵕ
ⵔ	234																														
ⵣ		230																					1					3				
ⵢ			232				2																								
ⵇ				228								5														1					
ⵜ					231																						3				
ⵙ						233									1																
ⵡ			1				233																								
ⵠ			1			1		230			2																				
ⵀ					1				232	1																					
ⵅ										233			1																		
ⵄ											234																				
ⵇ				4								228																			2
ⴰ													234																		
ⵃ														233					1												
ⵅ															234																
ⵇ																233														1		
ⵉ																	233			1											
ⴽ											1							232												1	
ⵏ																			234												
ⵛ																	1			233											
ⵍ																					234										
ⵅ		1			1					1					3							228									
ⵖ																							234								
ⵟ																			1					233							
ⵇ																									234						
ⵉ																										234					
ⵣ		12												1								1					220				
ⵚ				1																								233			
ⵄ																		1											228	5	
ⵄ																			1										2	231	
ⵕ												1													1		1				231

Le nouveau descripteur proposé et l'utilisation de plusieurs classifieurs ont nettement amélioré les résultats par rapport au travail précédent. En effet, nous avons pu surmonter légèrement le problème de la mauvaise écriture de certains caractères dans la base de données de test et aussi diminuer les erreurs de classification pour les caractères présentant une ressemblance structurelle à l'exception des deux caractères yazz (□) et yaz (□) comme montré dans la matrice de confusion.

Pour mettre en valeur notre approche, nous avons comparé nos résultats avec celles les plus performantes dans la littérature. Cette comparaison est présentée dans la Table 15.

Table 15. Comparaison des résultats avec les différentes approches de la littérature

Approche	Taux de reconnaissance (%)	Taille de l'ensemble d'apprentissage	Taille de l'ensemble de test
Approche améliorée	**99,03**	**16926**	**7254**
Amrouch et al [12]	97,89	16120	8060
Approche proposée	96,47	16926	7254
Es Saady et al [46]	96,32	18135	2015
Djematene et al [39]	92.30	1000	700
Gounane et al [58]	91.05	1940	Non spécifié

Nous signalons, selon notre connaissance, que le taux de reconnaissance obtenu est le meilleur taux jusqu'à l'instant dans la littérature.

2. 3. Résultats avec l'option du rejet

Nous allons présenter, dans cette partie, les résultats obtenus lors de l'ajout de l'option de rejet comme post-traitement. Ceci consiste à rejeter les caractères dont sa probabilité d'appartenance à la classe en question est inférieure à un seuil défini. Le seuil utilisé peut être global pour toutes les classes ou local pour chaque classe. Les résultats ci-dessous comparent entre l'utilisation d'un seuil global pour toutes les classes et l'utilisation de multiples seuils locaux (i.e., un seuil par classe).

2. 3. 1. Seuil global

La règle de Chow [34] permet de chercher un seuil fournissant un compromis entre le taux des erreurs et celui de rejet. Selon Chou, un caractère est accepté et attribué à la classe Ci si le maximum des probabilités de sortie est supérieur à un seuil donné t ∈ [0, 1]. Sinon, le caractère est rejeté.

$$\max_{k=1,\dots,N} P(C_k \mid x) = P(C_i \mid x) > T$$

La courbe ROC est souvent utilisée pour déterminer un seuil optimal dans les problèmes de classification, la courbe présente l'évolution de la sensibilité en fonction de la spécificité en variant le seuil t. la surface AUC sous la courbe ROC donne une bonne estimation de la capacité du système à rejeter les caractères mal classifiés sans rejeter ceux correctes.

On peut aussi, toujours en variant le seuil, présenter la précision du système (respectivement, la performance) par rapport au taux de rejet. Dans ce cas, on cherche le seuil permettant de maximiser la précision (respectivement, la performance) tout en gardant un taux de rejet raisonnable.

La Figure 80 illustre les trois courbes ROC, précision / taux de rejet et performance / taux de rejet des classifieurs utilisés en variant le seuil dans l'intervalle [0,1] d'un pas de 0,01.

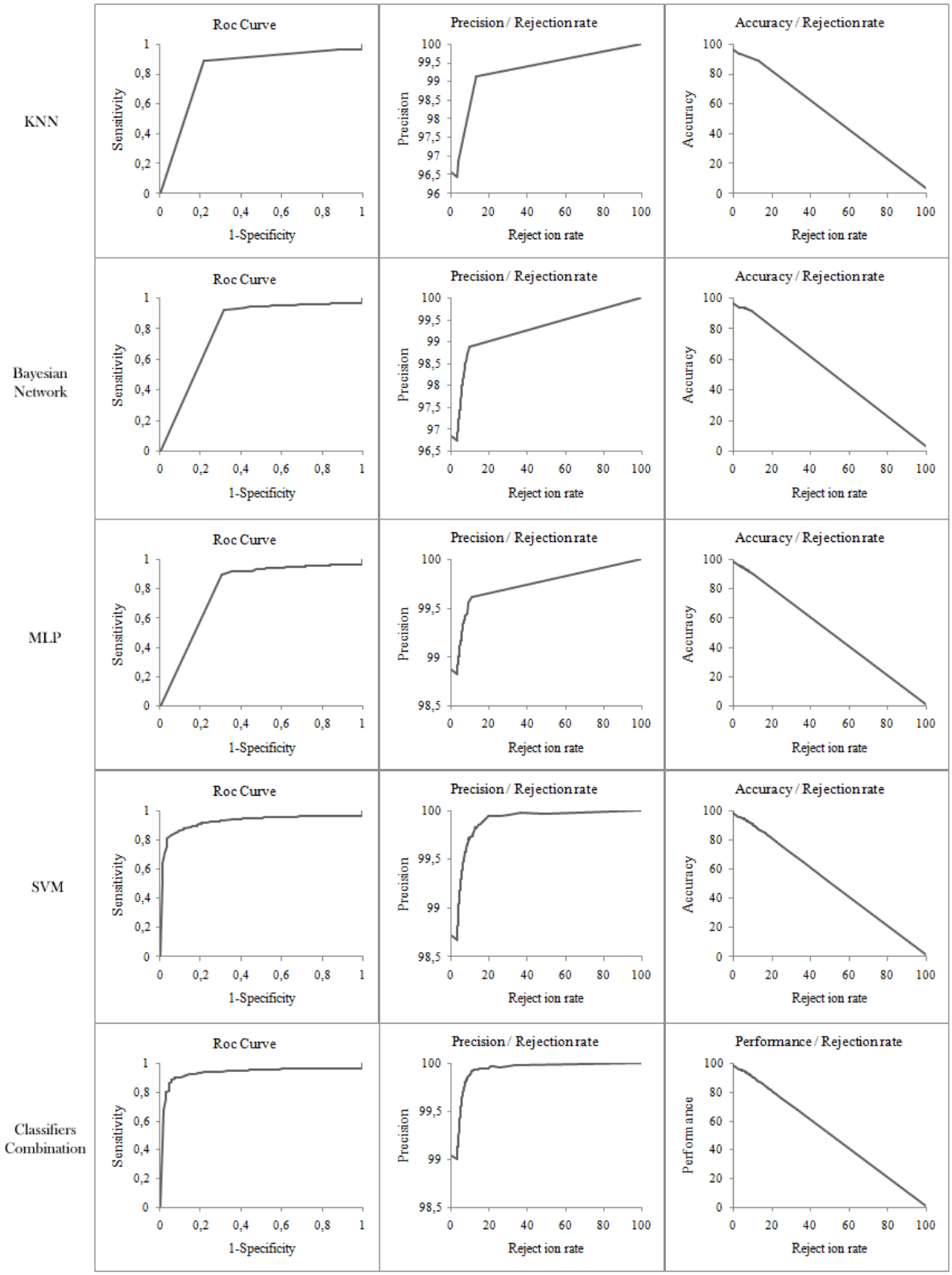


Figure 80. Courbes ROC, précision / taux de rejet et performance / taux de rejet des classifieurs utilisés

Dans tous les cas, le choix du seuil optimal à partir de ces courbes dépend de l'application sous considération. Dans ce travail, nous étions intéressés par la performance du système, les courbes présentant la performance en fonction du taux de rejet affirment que la performance diminue pour tous les classifieurs lors de l'utilisation de l'option de rejet. Ceci est du a l'utilisation d'un seuil global pour toutes les classes.

2. 3. 2. **Multi-seuils locaux**

Pour remédier au problème causé par le choix d'un seuil global pour toutes les classes, l'utilisation d'un seuil local au niveau de chaque classe peut être bénéfique. Comme nous l'avons mentionné, nous étions intéressés par la performance du système, c'est-à-dire, accepter les caractères bien classés en rejetant ceux mal classifiés. Alors, le seuil choisi pour chaque classe Ci est celui qui maximise le taux de reconnaissance au niveau de la classe i tout en gardant un taux de rejet raisonnable.

Table 16. Résultats de l'option de rejet avec multiples seuils pour les différents classifieurs

Classifieurs	**Taux de reconnaissance avec rejet (%)**	**Taux de reconnaissance sans rejet (%)**	**Taux de Rejet (%)**
PMC	98,91	98,86	0,17
SVM	98,92	98.71	0,35
RB	97,04	96,84	1,80
KNN	96,71	96.60	0,37
Classifieur combiné	99,10	99.03	0,37

Les résultats reportés dans la Table 16 montrent que la précision a augmenté pour tous les classifieurs avec des faibles taux de rejet. Ceci revient à dire que la majorité des caractères rejetés sont ceux mal classifiés et confirme que l'utilisation des multiples seuils dans un problème de classification multi-classes a été bénéfique.

Conclusion

Nous avons proposé un système de reconnaissance automatique hors ligne des caractères amazighs manuscrits en utilisant des méthodes statistiques lors de l'extraction de caractéristiques. Le descripteur proposé se base sur le calcul local des densités et des longueurs maximales des traits après la décomposition des images des caractères en plusieurs zones. Le descripteur obtenu contient 79 composants qui ont servi comme entrées pour un PMC entrainé pour décider de la classe du caractère. Afin d'augmenter les performances du système, une amélioration de ce descripteur a été proposée en calculant les sommes des longueurs des traits au lieu des longueurs maximales. Selon les analyses expérimentales, ces améliorations proposées ont produit de bons résultats avec des taux de reconnaissance très satisfaisants.

Chapitre. 4 : Contributions à la reconnaissance automatique hors ligne des caractères imprimés Amazighs en Tifinagh

Introduction

Ce chapitre présente les contributions dans le cadre de la reconnaissance automatique hors ligne de l'écriture Amazigh imprimée. Les résultats satisfaisants obtenus par notre descripteur proposé pour l'écriture manuscrite nous ont encouragé à tester sa performance sur les caractères imprimés. Malheureusement, dans l'absence d'une base d'images de référence des caractères amazighs imprimés, les chercheurs testent et évaluent la performance de leurs systèmes sur des bases d'images développées localement. Ceci nous a poussé à créer une nouvelle base d'images des mots Amazighs imprimés multi-polices, multi-tailles et multi-styles fournissant ainsi une large variabilité dans les données. Cette base, publique et gratuitement disponible sur internet, pourra être utilisée comme une base de référence pour évaluer et comparer les différents systèmes proposés de reconnaissance automatique des caractères Amazighs imprimés. Ensuite, nous avons introduit pour la première fois le problème du traitement du texte Amazigh incorporé, dans un environnement multilingue, dans les différents types d'images (Web, les affiches et flyers, les panneaux de signalisation, les scènes naturelles, etc.). Ce genre d'images nécessite un traitement particulier commençant par la détection et l'extraction des blocs de texte contenus dans l'image, passant par l'identification des scripts de ces blocs et arrivant à leurs reconnaissance automatique en utilisant le système OCR adéquat. Concernant la phase de reconnaissance, nous avons utilisé le même système de reconnaissance proposé pour l'écriture manuscrite qui a été entrainé et testé sur la nouvelle base d'images APWID.

I. Base des images des mots Amazighs imprimés APWID

Avec le développement technologique et la puissance de calcul des nouveaux matériels, la simulation de la lecture humaine est devenue un sujet de recherche très actif et la reconnaissance automatique de l'écriture à partir des images a connu un grand essor. En effet, plusieurs systèmes de reconnaissance automatique ont été développés pour différentes langues en utilisant différents méthodes et techniques. Afin d'évaluer et comparer ces systèmes et leur résultats, les chercheurs font recours à des bases d'images de référence. Pour certaines scripts telles que le latin, l'arabe, le chinois et le hindi, plusieurs bases de référence existent déjà et sont disponibles et gratuites sur le Web.

Dans les dernières décennies, les chercheurs ont commencé à donner de l'attention à la langue amazighe, et notamment, son script Tifinagh. Cependant, il y a un énorme manque en

termes de ressources communes permettant l'évaluation et la comparaison des systèmes OCR Amazighs développés.

Ce travail décrit en détails la génération d'une nouvelle base d'images de référence, nommée APWID (Amazigh Printed Words Images Database). Cette base de données contient 1795 mots rendus avec une procédure automatique utilisant différentes polices, tailles et styles offrant ainsi une large variabilité dans la base pour une comparaison à grande échelle des systèmes de reconnaissance automatique des caractères Amazighs imprimés.

1. Construction de la base de données APWID

1. 1. Collection des données

La base de données contient 1795 de mots Amazigh décomposables et non décomposables, les mots décomposables sont ceux générés à partir de verbes amazigh alors que les noms non décomposables sont formés par des noms propres, des jours, des mois, des animaux, etc. Ces mots Amazigh ont été extraits de certains livres publiés par l'institut IRCAM comme le dictionnaire français-amazigh-arabe et le lexique des médias (Figure 81). Les mots collectés ont été regroupés dans un fichier texte contenant un mot Amazigh dans chaque ligne.

Figure 81. Extraits des pages des livres utilisés pour la collection des mots amazighs

L'extraction des mots amazighs s'est effectuée d'une manière automatique en se basant sur le codage Unicode des caractères Tifinagh.

1. 2. Sources de variabilité des données

Les images de la base de données APWID ont été générées en utilisant 16 polices amazighes différentes proposées par l'institut IRCAM (Table 17). Nous avons utilisé toutes ces

polices pour couvrir la complexité des différentes formes des caractères imprimés amazighs, allant des simples polices (Tifinaghe IRCAM STANDARD, Tamzwart Standard UNICODE) à des polices plus complexes riches en chevauchements (Tifinaghe Tazirit UNICODE). Nous avons également utilisé différentes tailles pour chaque police : 8, 9, 10, 11, 12, 14, 16, 18, 20 et 24 points. Et pour chaque combinaison de police et de taille, nous avons utilisé différents styles : Plain, Bold, Italic et Bold-Italic.

Table 17. Polices utilisées pour la génération de la base d'images APWID

Mot Amazigh	Id de la police	Nom de la police
□□□□□□□□□	F01	Tifinaghe-Ircam Unicode
□□□□□□□□□	F02	Tifinaghe-IRCAM2_Unicode
□□□□□□□□□	F03	Tamzwart Standard UNICODE
□□□□□□□□□	F04	Tamalout Standard UNICODE
□□□□□□□□□	F05	Tifinaghe-IRCAM-taromit_unicode
□□□□□□□□□	F06	Teddus Standard UNICODE
□□□□□□□□□	F07	Tassafout Standard UNICODE
□□□□□□□□□	F08	Tifinaghe-Tazdayt Standard UNICODE
□□□□□□□□□	F09	Tifinaghe-IRCAM-Agoug_unicode
□□□□□□□□□	F10	Tifinaghe-IRCAM- taromit 2_unicode
□□□□□□□□□	F11	Tamzwart Noufouss UNICODE
□□□□□□□□□	F12	Tamalout Noufouss UNICODE
□□□□□□□□□	F13	Teddus Noufouss UNICODE
□□□□□□□□□	F14	Tassafout Noufouss UNICODE
□□□□□□□□□	F15	Tifinaghe-Tazdayt Noufouss UNICODE
□□□□□□□□□	F16	Tifinaghe Tazirit UNICODE

Les polices, les tailles et les styles utilisés garantissent une large variabilité dans les images de la base.

1. 3. **Procédure de génération des images**

Les images de notre base APWID ont été générées automatiquement en effectuant le rendu du mot Amazigh avec un programme Java, de sorte que le bruit et les artefacts présents, généralement, dans les images numérisées ne sont pas présents dans les images de la base (Figure 82).

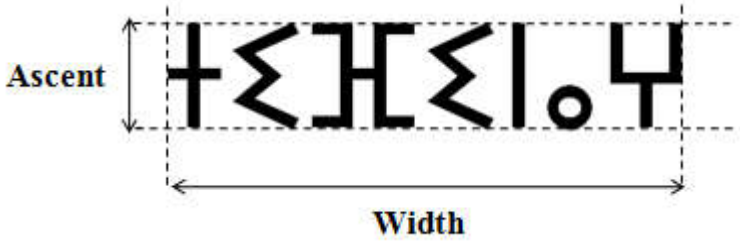


Figure 82. Mesures correspondantes au mot et à la police utilisée

Pour chaque mot, nous avons utilisé différentes combinaisons de polices, tailles et styles et l'image de mot à générer a été rendue avec le filtrage anti-aliasing de texte implémenté dans la bibliothèque standard java par la classe RenderingHints. L'algorithme suivant décrit la procédure du rendu suivie pour générer la base de données APWID :

Algorithme de génération de la base d'image APWID

Pour chaque mot Amazigh Contenu dans le fichier composé de 1795 mots
Pour chaque police des 16 polices de la Table 17
Pour chaque taille dans : 8, 9, 10, 11, 12, 14, 16, 18, 20 et 24 pt
Pour chaque style dans : plain, gras, italique, gras et italique

- Récupérer les mesures correspondantes à la police et au mot rendu (Figure 82)
- Créer une nouvelle image avec des dimensions correspondantes aux mesures précédentes
- Configurer les paramètres du rendu (anti-aliasing, qualité, mode de couleur,…)
- Effectuer le rendu du mot dans l'image et l'enregistrer
- Générer le fichier XML de description

Fin pour
Fin Pour
Fin Pour
Fin Pour

L'algorithme produit à chaque itération une image contenant un mot avec une police, taille et style différent, pour chaque image un fichier XML est généré contenant des informations sur le mot et l'image en cours.

1. 4. Fichier de description

Une description détaillée pour chaque image du mot Amazigh de la base APWID est attachée par un fichier XML rapportant des informations sur la séquence de caractères du mot en question, ainsi que des informations sur l'image et les paramètres du rendu. La Figure 83 illustre un exemple d'une image de la base et son fichier XML correspondant. Ce fichier est composé de 4 annotations pour fournir toutes les informations nécessaires sur l'image du mot telles que le contenu, le style utilisé, les spécifications de l'image et la procédure de génération :

- **Content** : cet élément fournit la transcription du mot amazigh, le nombre de caractères formant le mot amazigh (nPaws) et des sous-éléments donnant, pour chaque caractère, son code UTF correspondant et sa fréquence d'apparition dans le mot.
- **Font** : cet élément présente des informations sur le style utilisé (police, taille et style) pour générer l'image de mot.
- **Specs** : cet élément indique le codage de l'image, sa largeur et sa hauteur.
- **Generation** : cet élément fournit des informations supplémentaires sur la procédure de génération des images.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<wordimage id="IMG_4">
  <content transcription="ⴰⴱⴱⵓⵥⵥⵍ" nPaws="5">
    <paw utf="\u2D30" frequency="1">ⴰ</paw>
    <paw utf="\u2D31" frequency="2">ⴱ</paw>
    <paw utf="\u2D53" frequency="1">ⵓ</paw>
    <paw utf="\u2D65" frequency="2">ⵥ</paw>
    <paw utf="\u2D4D" frequency="1">ⵍ</paw>
  </content>
  <font name="Tifinaghe-IRCAM-Agoug_unicode" fontStyle="Italic" fontSize="20" /:
  <specs encoding="png" width="98" height="22" />
  <generation renderer="java" filtering="antialiasing" />
</wordimage>
```

Figure 83. Fichier XML de description

2. Statistiques et utilisation de la base de données APWID

Cette section est consacrée à la présentation des statistiques sur la base APWID, son stockage sur le disque et quelques modes de son utilisation.

2. 1. Statistiques sur la base de données

La base de données APWID contient 1795 mots Amazighs composés de 10453 caractères et rendus dans les différentes combinaisons de 16 polices, 10 tailles et 4 styles. Nous avons obtenu ainsi, comme montré dans la Table 18, 1148800 images contenant environ de 7 millions de caractères Amazighs fournissant ainsi une large base d'images pour comparer les différents systèmes OCR Amazighs.

Table 18. Statistiques sur la base d'images APWID

	Number of words	Number of characters
	1795	10453
Styles	16 fonts * 10 sizes * 4 styles	
Total	1148800	6689920

Les fréquences d'apparition de chaque caractère Amazigh dans les mots de la base sont reportées dans la Table 19.

Table 19. Fréquences d'apparition des caractères dans la base APWID

Caractère	Fréquence d'apparition dans la base APWD
ⵔ	1312000
ⵣ	140160
ⵗ	3840
ⵕ	336640
ⵜ	636800
ⵥ	128000
ⵖ	417920
ⵚ	529280
ⵛ	13440
ⵅ	33280
ⵆ	82560
ⵙ	477440
ⵏ	190720
ⵀ	160000
ⵡ	211200
ⵟ	7040
ⵉ	44160
ⵇ	135040
ⵍ	275840
ⵎ	400000
ⵓ	133120
ⵅ	31360
ⵄ	58240
ⵢ	120960
ⵟ	107520
ⵉ	341760
ⵣ	84480
ⵄ	28800
ⵒ	32640
ⵒ	119040
ⵇ	77440
ⵔ	19200

2. 2. **Stockage**

La base d'images APWID occupe environ de 900 Mo d'espace disque et elle est publiquement téléchargeable via le lien suivant : https://goo.gl/eCJKjF.

Les fichiers de la base sont organisés en 16 répertoires représentant les 16 polices amazighes, chaque répertoire contient 10 sous-répertoires pour les 10 tailles de police utilisées et chaque sous-répertoire de taille contient 4 sous-répertoires pour les 4 différents styles (Figure 84).

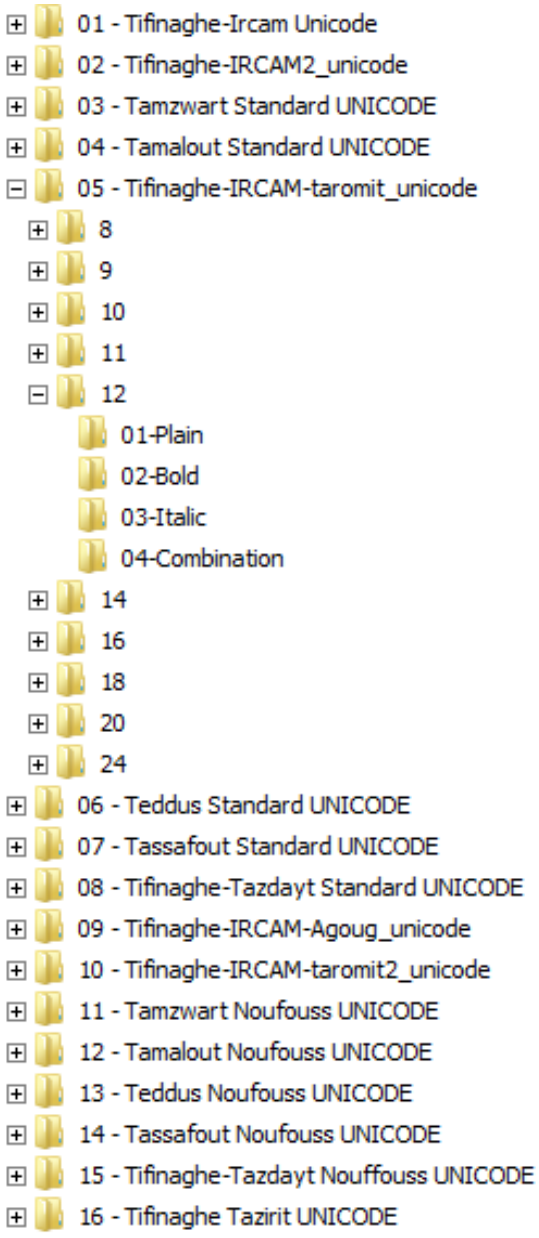


Figure 84. Stockage sur disque de la base APWID

2. 3. **Utilisation**

La base d'images APWID peut être utilisée comme source d'apprentissage et de test dans de nombreuses applications telles que les systèmes OCR, les systèmes d'identification du script et les algorithmes de segmentation de caractères. Nous avons proposé dans la Table 20 quelques protocoles à utiliser pour tester l'impact de la variabilité des données dans la base de données sur les systèmes testés.

Table 20. Quelques protocoles d'utilisation de la base d'images APWID

Protocol	Training Set Train (font, size, style)	Testing Set Test (font, size, style)
APWID1	Train (F01, 14, P)	Test (F01, 14, P)
APWID2	Train ([F01,F03,F05,F06], [12,14], [P, I])	Test ([F02,F04,F08,F09], [12,14], [P,I])
APWID3	Train([F01,F02], [14,16], [P,I])	Test ([F01,F02], [14,16], [B,BI])
APWID4	Train ([F11-F15], [9,13,20], P)	Test ([F11-F15], [12,16,24], P)
APWID5	Train (All, All, All)	Test (All, All, All)
APWID6	Train ([F01, F05, F08, F09], [20,24], All)	Test ([F01, F05, F08, F09], [20,24], All)

Ces protocoles utilisent les notations Train (police, taille, style) et Test (police, taille, style) pour définir les conditions d'apprentissage et de test où :

- Police : est l'identifiant de police indiqué dans la Table 17;
- Taille : indique la taille de police utilisée (en pt);
- Style : style utilisé où P, B, I et BI définit respectivement les styles Plain, Bold, Italic et Bold & Italic.

Les protocoles proposés ont des objectifs bien définis et sont les suivants:

- **APTWID1** : Ceci est le protocole de base étant donné qu'il n'y a pas de différences entre les conditions d'apprentissage et de test ainsi que l'utilisation d'une seule police. La performance des systèmes OCR, dans ce cas, devrait être la plus élevée possible;
- **APTWID2** : Celui-ci consiste à tester la capacité des systèmes à reconnaître les polices non vues lors de l'apprentissage.
- **APTWID3** : Ce protocole vise à évaluer la capacité des systèmes à traiter des nouveaux styles.
- **APTWID4** : dans ce protocole, nous mesurons la capacité du système à reconnaître les nouvelles tailles de polices;
- **APTWID5** : Ce protocole est un protocole global où toutes les données sont utilisées pour l'expérimentation;
- **APTWID6** : Le dernier protocole est destiné aux systèmes d'identification du script.

La base de données peut être également utilisée pour tester différents algorithmes de segmentation de caractères pour les polices complexes telles que 'Tifinaghe-IRCAM-taromit2_unicode' où l'algorithme de segmentation utilisant l'histogramme de projection verticale ne peut pas conclure (Figure 85).

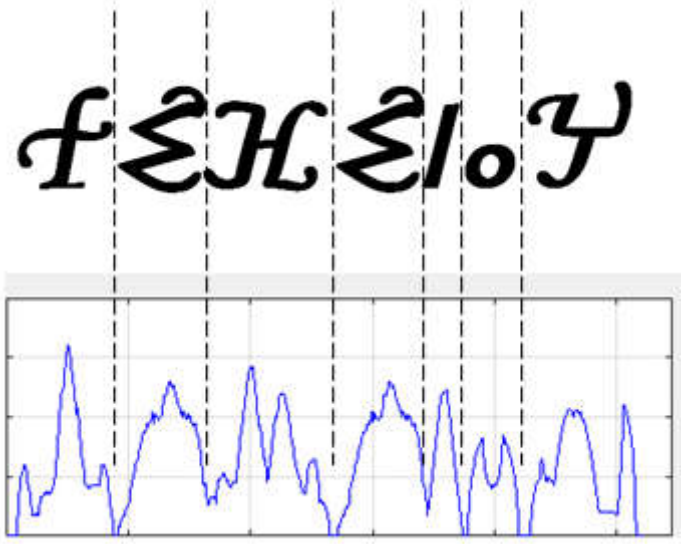


Figure 85. Échec de segmentation de l'algorithme basé sur l'histogramme de projection verticale

Les utilisateurs de la base de données APWID sont libres de créer leurs propres combinaisons des ensembles d'apprentissage et de test en fonction de leurs propres besoins en bénéficiant de la variabilité des données.

II. Système de bout en bout pour la reconnaissance de l'écriture Amazighe dans les différents documents images

Les informations textuelles contenues dans les images Web, les affiches et flyers imprimées et les images de scènes naturelles prises par les caméras intégrées dans différents appareils mobiles sont de plus en plus intéressantes dans le domaine de la vision par ordinateur. Le texte intégré dans ce genre d'images est considérées comme très important pour la compréhension de l'image. En effet, ceci rend son traitement indispensable pour les chercheurs en raison de ses diverses applications telles que la reconnaissance automatique du texte, la compréhension des scènes, la géolocalisation automatique, la navigation robotique, la lecture des plaques d'immatriculation, l'indexation des documents, etc.

D'une autre part, dans les systèmes OCR, un des problèmes souvent rencontrés dans ce contexte est que les images peuvent contenir du texte multilingue, rendent ainsi l'identification des scripts utilisés une tâche primordiale pour déterminer le système de reconnaissance adéquat à utiliser. Au Maroc, après la standardisation de la langue amazighe et la normalisation du script Tifinagh, il est plus courant de voir la cohabitation des trois scripts (Tifinagh, arabe et latin). Actuellement, le script Tifinagh est de plus en plus utilisé dans divers domaines tels que la littérature, le Web, la publicité, l'éducation et les médias (Figure 86).

Figure 86. Cohabitation du script Tifinagh avec les autres scripts

Malheureusement, tous les travaux consacrés à la reconnaissance automatique du script Tifinagh ne traitaient pas le problème de multilinguisme et supposaient que le document en cours de traitement est un document amazigh où le texte est extrait au préalable. Ceci nous a poussé à concevoir dans ce travail un système de bout en bout, traitant le texte contenu dans les images depuis l'extraction jusqu'à la reconnaissance.

1. Système proposé

Pour traiter le texte Amazigh diffusé dans les images Web ou de scènes naturelles, nous avons proposé un système à plusieurs étapes qui extrait d'abord les blocs de texte contenus dans les images d'entrée, identifie le script du texte extrait, et procède à la reconnaissance automatique de celui correspondant au script Tifinagh.

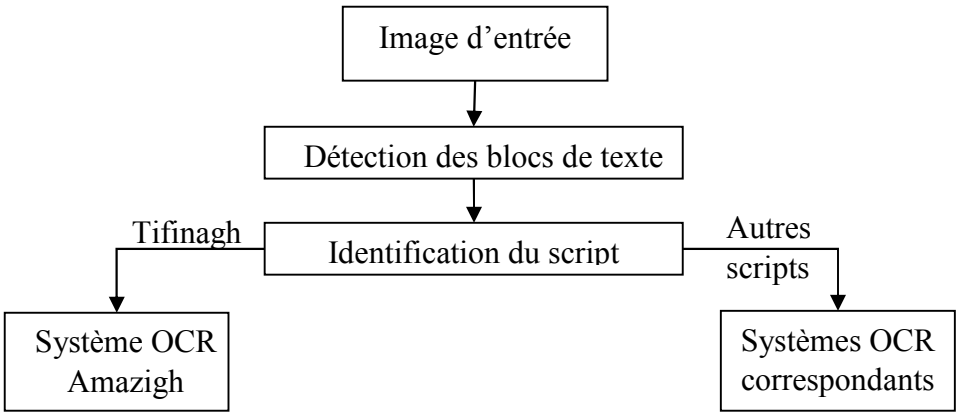


Figure 87. Système proposé pour la reconnaissance du Tifinagh dans les images Web et de scènes naturelles

La Figure 87 montre les étapes nécessaires pour construire le système qui seront décrites en détails dans ce qui suit.

1. 1. **Détection de texte**

Le texte est l'un des moyens de communication les plus expressifs, il est présent tout autour de nous sous différentes formes (imprimé et manuscrit) et contient des informations assez importantes. D'une autre part, en plus des scanners, la disponibilité croissante de dispositifs mobiles de haute performance avec une capacité d'imagerie et de calcul a créé une opportunité pour l'acquisition et le traitement d'images à tout moment et n'importe où. Ceux-ci ont rendu le texte présent dans les images capturées comme un aspect très important pour la compréhension de l'image, de sorte que sa détection et son extraction sont devenues des problèmes clés qui ont généré trop d'intérêt pour les chercheurs scientifiques de nos jours.

La détection et l'extraction du texte à partir des images est une sous opération primordiale vu ses différentes applications. Nombreux sont les travaux qui ont été réalisés dans ce domaine, une excellente étude des algorithmes largement utilisés est présentée dans [161]. La plupart de ces algorithmes utilisent des méthodes basées sur la détection des contours, l'analyse de texture et les composants connectés.

Figure 88. Résultats de l'algorithme de détection de texte

Dans ce travail, pour détecter le texte incorporé dans les images, nous avons utilisé l'algorithme proposé par Liu et al [95]. Cet algorithme est basé sur les contours, étant donné que leur force, leur densité et leur variation d'orientation sont considérées comme des caractéristiques distinctives pour détecter le texte. Ainsi, les régions contenant du texte ont normalement tendance à avoir des valeurs de densité, force et variation d'orientation des contours plus élevées que celles des régions non textuelles. Cet algorithme comprend trois étapes : la détection de régions de texte candidats ; la localisation de régions de texte ; et l'extraction de caractères.

Cet algorithme peut détecter les différents scripts y compris le Tifinagh et a montré sa robustesse face aux variations de police, taille, style, orientation, couleur / intensité, etc. (Figure 88)

1. 2. **Identification du script du texte**

Le multilinguisme est courant dans les pays nord africains et le Maroc ne fait pas l'exception. Les trois principales langues utilisées sont l'arabe, le français et l'amazigh. Par conséquence, les images capturées des scènes naturelles, les affiches ou flyers imprimées, les panneaux de signalisation ou les panneaux des établissements publics (écoles, universités, tribunaux, ministères, etc.) peuvent contenir du texte multilingue. Dans le domaine de l'OCR, la reconnaissance des différents textes multilingues dans un seul module OCR reste une tâche difficile vue les différences dans le style, l'orientation et la structure des scripts en question. Une autre solution consiste à utiliser une banque des systèmes OCR correspondants à tous les scripts différents qui devraient être reconnus. Les textes d'un document d'entrée peuvent ensuite être reconnus de manière fiable en sélectionnant le système OCR approprié pour chaque script dans la banque des systèmes OCR. Néanmoins, cela nécessitera l'identification a priori du script de texte pour bien orienter le bloc de texte vers le système OCR adéquat.

La reconnaissance du texte Amazigh contenu dans les images Web ou de scènes naturelles n'a jamais été évoquée par les chercheurs et aucun travail n'a traité l'identification du script Amazigh dans un environnement multilingue. Dans ce travail, nous visons à créer un système capable de classer les blocs de texte détectés par l'algorithme précédent selon les différents scripts utilisés au Maroc (Arabe, latin, Tifinagh), le système doit également être capable d'identifier les nombres. À cette fin, en raison de ses performances, nous avons entrainé un réseau de neurones convolutifs (CNN), c'est l'une des architectures les plus connues des réseaux de neurones. C'est un algorithme d'apprentissage en profondeur qui fait l'objet de recherches intenses, c'est une séquence de couches où chaque couche transforme un volume d'activations en un autre via une fonction différentiable. Les trois couches les plus utilisées pour construire un CNN sont les couches convolutives, les couches de pooling et les couches entièrement connectées (voir Section III. 1. 2. du chapitre 1). Les détails d'apprentissage et de test du système proposé sont présentés dans ce qui suit.

1. 2. 1. **Base d'image d'apprentissage et de test**

Pour entrainer et tester le CNN utilisé, nous avons développé une base d'images locale composée, en total, de 28800 images (7200 par classe) où chaque image contient une séquence de caractères (mots ou nombres) des différentes classes de sortie (Tifinagh, arabe, latin, nombre).

Les 66,66% de la base (19200 images) ont fait l'objet de l'ensemble d'apprentissage et le reste (9600 images) ont servi pour le test. Certains exemples extraits de la base sont donnés dans la Figure 89.

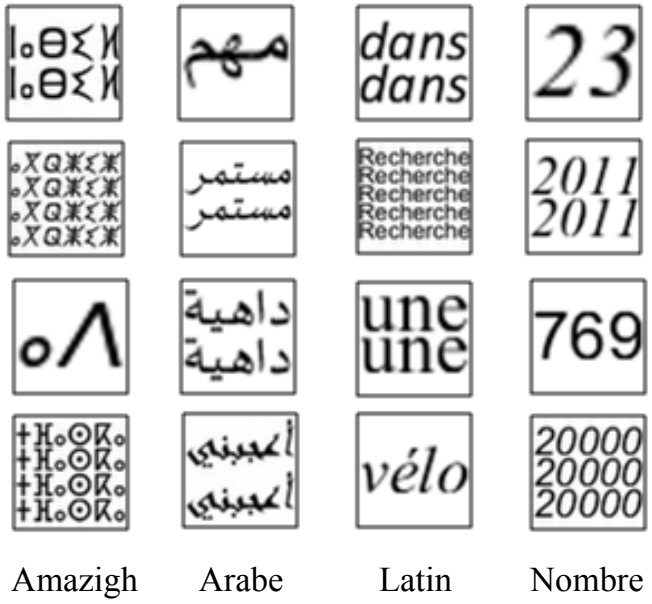


Figure 89. Extraits de la base de données utilisée

Pour créer la base d'images, nous avons suivi plusieurs étapes. D'abord, nous avons collecté 1800 séquences de caractères différentes pour chacune des quatre classes. Les mots utilisés pour le Tifinagh sont ceux de la base d'images APWID. Les mots arabes et français ont été choisis à partir de des mots les plus couramment utilisés recueillis à partir des sous-titres des films, ces mots sont regroupés dans des fichiers sous différents formats et téléchargeables gratuitement à partir du site Web 101languages.net. Alors que les nombres sont des nombres entiers générés aléatoirement entre 0 et 20000. Ensuite, Les mots collectés ont été rendus par la même procédure que la base d'images APWID, chaque mot a été rendu quatre fois en utilisant une police, une taille et un style différents pour garantir une grande variété des styles des textes dans la base. Enfin une étape de post-traitement est nécessaire pour redimensionner toutes les images de la base à une taille de 50x50 pixels identique à celle de l'image d'entrée du CNN. Cette opération est automatique qui prend en compte le rapport entre la largeur et la hauteur de l'image du mot et tente de dupliquer le mot, autant de fois nécessaires, pour créer une image finale carrée (Figure 89) qui est redimensionnée à une taille de 50x50 pixels.

1. 2. 2. **Apprentissage**

La base de données créée a servi pour entrainer le CNN composé de deux couches convolutives de 20 et 10 cartes de caractéristiques respectivement, la taille des patchs de convolution est de 5x5 pour les deux couches et le pooling est effectué via la fonction max sur des blocs de taille 2x2. La dernière couche est une couche entièrement connectée avec quatre neurones qui contient les scores pour chaque classe. L'architecture CNN utilisée est illustrée dans la Figure 90.

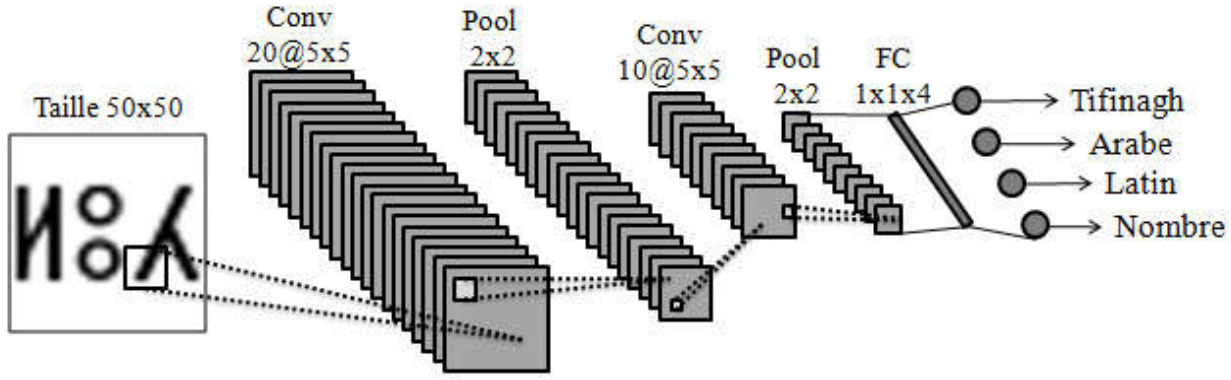


Figure 90. Architecture du CNN pour l'identification du script

Le CNN a été entrainé par la descente du gradient stochastique avec 19200 itérations et un taux d'apprentissage de 10^{-4}.

1. 2. 3. **Résultats**

Le CNN entrainé atteint une précision de 99,12% sur un ensemble de test constitué de 9600 image, 2400 images pour chaque classe.

Table 21. Matrice de confusion du CNN entrainé

	Amazigh	**Arabic**	**French**	**Number**
Amazigh	2375	6	16	3
Arabic	3	2392	3	2
French	30	5	2349	16
Number	0	0	0	2400

Les erreurs de classification illustrées dans la matrice de confusion (Table 21) sont dues à l'existence de quelques symboles en communs entre les différents scripts. La majorité de ces erreurs ont été enregistrées, en particulier, pour des mots courts ou des mots contenant un seul caractère. La Table 22 montre certains mots mal classés dans l'ensemble de test.

Table 22. Exemples de mis-classifications commises par le CNN

Mot	**Script**	**Prédiction du CNN**
□□□	Amazigh	Latin
□□	Amazigh	Arab
□□	Amazigh	Number
أما	Arab	Amazigh
لا	Arab	Latin
زال	Arab	Number
FIN	Latin	Amazigh
Y	Latin	Arab
b	Latin	Number

Les résultats obtenus par le CNN sont très satisfaisants, cependant, le système présente deux inconvénients : le premier est le post-traitement nécessaire pour redimensionner les images d'entrée ; le deuxième est qu'il est limité à des mots.

1. 3. Système de reconnaissance automatique des caractères Amazigh imprimés

Après l'extraction des blocs de texte contenus dans les images et identification des scripts de ces derniers, nous procédons à la reconnaissance automatique du texte. Pour la reconnaissance des nombres ou des deux scripts arabe et latin, la littérature est riche en systèmes OCR performants et on peut rediriger les blocs de texte correspondants vers les systèmes OCR adéquats. Cependant, pour le script Tifinagh, assez négligé par les chercheurs, en parcourant la littérature, nous avons remarqué que tous les systèmes OCR Amazighs proposés (voir section II. 3. 1. du chapitre 2) sont mono-polices et leur performance face aux images contenant des textes en différentes tailles et styles de police, ce qui est le cas en réalité, n'a pas été discutée. D'une autre part, ces systèmes OCR ont été testés sur des bases d'images développées localement et de petites tailles. Ceux-ci nous ont incité à proposer un système de reconnaissance automatique des caractères Amazighs imprimés. Ce système, testé sur la large base de données APWID, est capable de reconnaitre les caractères Amazighs imprimés dans différentes polices, tailles et styles.

1. 3. 1. Système proposé

Le système proposé reprend les étapes du système déjà présenté pour l'écriture manuscrite en utilisant les mêmes opérations mais en ajoutant la squelettisation comme opération de prétraitements. En ce qui concerne l'extraction des caractéristiques, nous avons utilisé le même vecteur descripteur en raison de sa performance. Tandis que pour la phase de reconnaissance, nous avons entrainé et testé les différents classifieurs utilisés sur notre base d'images APWID. La Figure 91 montre les étapes constituantes du système.

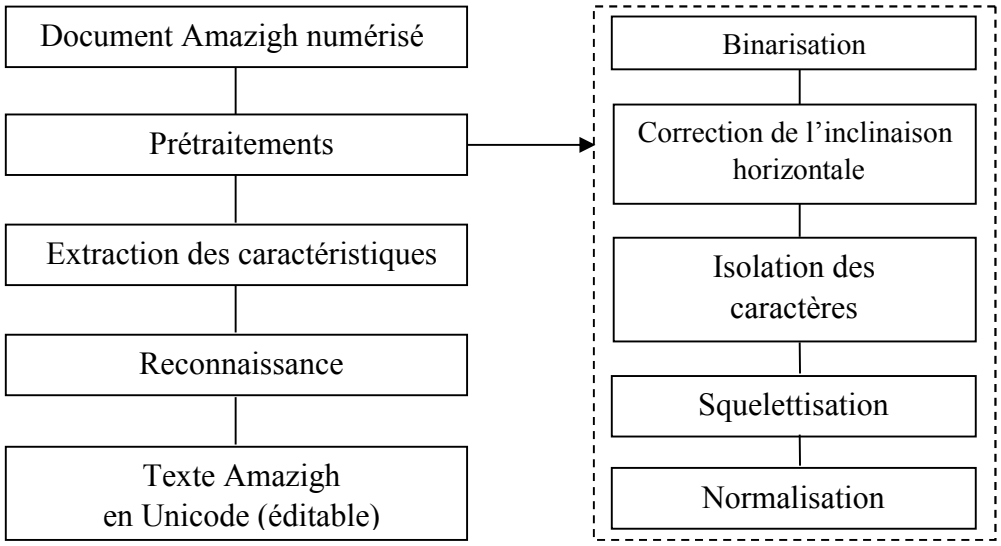


Figure 91. Les étapes du système OCR des caractères Amazighs imprimés

Dans l'étape des prétraitements, nous préparons les caractères contenus dans l'image d'entrée pour la phase d'extraction des caractéristiques. Ainsi, nous avons utilisé la méthode d'Otsu pour binariser chacune des images d'entrée. Puis nous avons corrigé l'inclinaison horizontale du texte. Ensuite, nous avons segmenté les lignes du texte en mots puis en caractères. Et enfin, nous avons normalisé les caractères isolés obtenus en une taille commune de 30x30 après les avoir squelettisés.

Pour l'extraction des caractéristiques de chaque caractère, qui est considérée comme une étape cruciale et décisive influençant largement les performances des systèmes OCR, nous avons extrait des caractéristiques pertinentes tout en tenant en compte de la structure et de la forme des caractères Amazighs et leurs variations selon la police, la taille et le style. Le vecteur descripteur utilisé est décrit précédemment dans la section II. 1. 1. du chapitre 3. Ce descripteur se base sur la décomposition de l'image normalisée et squelettisée du caractère en plusieurs zones chevauchées et le calcul des caractéristiques de densité et des profils de projection selon différentes orientations dans les différentes zones. Ce descripteur a été combiné avec différents classifieurs pour tester sa performance face aux caractères Amazighs imprimés multi-polices, multi-tailles et multi-styles.

Pour la phase de reconnaissance, Nous avons utilisé trois classifieurs parmi les classifieurs les plus utilisées dans le contexte de la classification supervisée, à savoir, le Perceptron Multicouches PMC, la Machine à Vecteurs Support SVM et les forets aléatoires RF. Ces algorithmes ont montré leurs performances dans différentes applications de reconnaissance de formes et spécialement avec les systèmes OCR.

En outre, la combinaison de plusieurs classifieurs en phase de reconnaissance s'est avérée d'une très grande utilité pour améliorer les performances. En effet, l'utilisation d'une fonction de combinaison sur les probabilités de sortie de chaque classifieur permet de prédire la classe du caractère en bénéficiant des décisions des différents classifieurs. Divers fonctions de combinaison peuvent être utilisées telles que la moyenne, le max, le produit, le vote, etc. la Table 23 résume les différentes configurations utilisées pour chaque classifieur.

Table 23. Différentes configurations utilisées pour les classifieurs

Classifieur	Configurations	Détails
PMC	Variation du nombre des neurones cachés	80, 85, 90, 95
SVM	Variation de la fonction noyau	Polynomiale, Gaussienne, linéaire
RF	Variation du nombre d'arbres	250, 500, 750, 1000
Combinaison des classifieurs	Variation de la fonction de combinaison	Vote, Maximum, Moyenne

Pour évaluer les classifieurs, nous avons retenu les trois mesures communément utilisées dans le domaine de la classification supervisée, à savoir, le taux de reconnaissance, le recall et la F-mesure (Voir section III. 2. du chapitre 1).

1. 3. 2. **Tests et expérimentations**

Plusieurs expérimentations ont été effectuées pour entrainer les classifieurs avec les différentes configurations sous un HP ProBook compatible, un processeur Intel (R) Core (TM)

i5-2520M 2,50 GHz, et 4 Go de RAM via le langage Java. Durant la phase d'apprentissage, les classifieurs ont été entrainés et validés, en utilisant la 10-validation croisée, avec des images des caractères amazighs isolés rendus avec la même procédure que la base d'images APWID en utilisant les quatre polices les plus régulières (*Tifinagh-IRCAM, Taromit, Tamalout et Tamzwart*) combinées avec différentes tailles différentes (10, 11, 12, 14, 16, 18, 20 et 24) et quatre styles différents (simple, gras, italique et gras-italique). La taille de l'ensemble d'apprentissage est donc de 4096 images (4 polices * 8 tailles * 4 styles * 32 caractères).

Afin d'évaluer les performances des classifieurs entrainés et vérifier leur capacité de généraliser, nous avons mené deux tests. Le premier test a été effectué sur les mots de la base de données APWID tirés des quatre polices mentionnées précédemment combinées avec les différentes tailles et styles utilisées lors de l'apprentissage, ainsi l'ensemble de test est constitué de 345600 images des caractères. Certains mots dans la base du test ont été enlevés vu qu'ils présentent des difficultés de segmentation en caractères. De plus, comme deuxième test, nous avons testé la capacité du système à reconnaître des mots imprimés avec des polices et des tailles non utilisées dans la phase d'apprentissage. La Table 24 montre les différentes polices utilisées lors de l'apprentissage et de test.

Table 24. Polices Tifinagh utilisées pour l'apprentissage et le test

Amazigh Word image	Font Name	Size	Style
ⵜⵉⴼⵉⵏⴰⵖ	Tifinaghe-Ircam	24 pt	Plain
ⵜⵉⴼⵉⵏⴰⵖ	Tifinaghe-taromit	16 pt	Bold
ⵜⵉⴼⵉⵏⴰⵖ	Tamalout Standard	14 pt	Bold & italic
ⵜⵉⴼⵉⵏⴰⵖ	Tamzwart Standard	18 pt	Italic
ⵜⵉⴼⵉⵏⴰⵖ	Tifinaghe-Agoug	14 pt	Bold
ⵜⵉⴼⵉⵏⴰⵖ	Tifinaghe-IRCAM2	16 pt	Italic
ⵜⵉⴼⵉⵏⴰⵖ	Teddus Standard	20 pt	plain
ⵜⵉⴼⵉⵏⴰⵖ	Tassafout Standard	12 pt	Bold & italic

1. 3. 3. **Résultats et discussions**

Cette partie présente les différents résultats obtenus pour les classifieurs durant la phase d'apprentissage et de test en ne retenant que la meilleure performance pour chaque classifieur suivant les différentes configurations.

La Table 25 reporte les résultats obtenus lors de la validation croisée-10 durant la phase d'apprentissage.

Table 25. Résultats de l'apprentissage par la validation croisée-10

Classifieur	Configuration	Taux de reconnaissance	Recall	F- Mesure
PMC	85 neurones cachés	98,86	98,9	98,9
SVM	noyau polynomial	98,87	98,9	98,9
RF	750 arbres	98,27	98,3	98,3
Combinaison des classifieurs	Vote majoritaire	99.93	99,9	99,9

Le premier test a été effectué pour évaluer les performances des classifieurs entrainés, avec les différentes configurations, sur la base de test APWID en utilisant les mêmes polices, tailles et styles utilisées lors de l'apprentissage. La Table 26 reporte les résultats obtenus sur un ensemble de test de 345600 caractères en retenant que la meilleure performance pour chaque classifieur.

Table 26. Taux de reconnaissance obtenus pour les différents classifieurs pour le premier test

Classifieur	Configuration	Taux de reconnaissance (%)	Recall (%)	F- Mesure (%)
PMC	85 neurones cachés	96,08	96,1	96,1
SVM	noyau polynomial	95.92	95,9	95,9
RF	750 arbres	95.43	95,4	95,4
Combinaison des classifieurs	Vote majoritaire	97.48	97,5	97,5

Comme nous pouvons le remarquer, la meilleure performance a été enregistrée par la combinaison des classifieurs avec un taux de reconnaissance de 97,48%, qui est considéré très satisfaisant vu la taille de la base d'images de test (345600). La plus part des erreurs de reconnaissance commises sont dues à deux facteurs : Le premier est la similarité structurelle de certains caractères Amazighs (Figure 92.a) ; Le second est la perte d'informations (Figure 92.b) due aux opérations des prétraitements (la binarisation, la squelettisation et le redimensionnement), en particulier pour les petites polices.

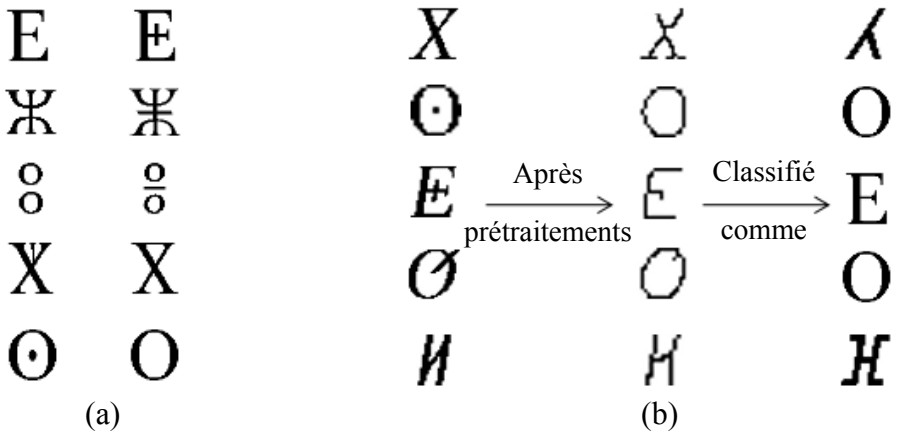


Figure 92. Les facteurs provoquant les erreurs de classification

La matrice de confusion détaillant les erreurs commises, pour chaque caractère, par le classifieur combiné est donnée dans la Table 27.

Table 27. Matrice de confusion du classifieur combiné

	ⵔ	ⵣ	ⵢ	ⵓ	ⵜ	ⵙ	ⵡ	ⵠ	ⵀ	ⵅ	ⵍ	ⵎ	ⵏ	ⵒ	ⵔ	ⵕ	ⵖ	ⵗ	ⵘ	ⵙ	ⵚ	ⵛ	ⵜ	ⵝ	ⵞ	ⵟ	ⵠ	ⵡ	ⵢ	ⵣ	ⵤ	ⵥ	
ⵔ	67840																																
ⵣ		6360												13															526					
ⵢ			182			13	6																	16						1				
ⵓ				17187																						61			344			213		
ⵜ			110		32057	50	123			16																		101						
ⵙ			103			5835	82								124								53										4
ⵡ		169	605			159	20528	85																6				84					
ⵠ			52			322	60	27059							50	100												33					
ⵀ									691	1				16	15																		
ⵅ			12							1308				87		17			35	36			10					67					
ⵍ											4400						140		37					117									
ⵎ				194								24639		223						155						219			210			176	
ⵏ									17				6990							197			196					76					
ⵒ														8852																			
ⵔ							3								7324								454					144					
ⵕ				10												351													3				
ⵖ			51			209		185									1675										131						
ⵗ																		6758	3				9					86					
ⵘ											69	13							13948		36							216	156				
ⵙ											61									20774			63							141	20		
ⵚ				99															210	198	6242				2							72	
ⵛ													31		69							1244					127						
ⵜ																							3270										
ⵝ		59																						5936	128		65						
ⵞ				466																					5342								
ⵟ					65	87		5									301									17173							
ⵠ		520																				128					4135						
ⵡ																			20				45					1433					
ⵢ			46								2		14	2				2											1761	17			
ⵣ											3			121						69									527	5874			
ⵤ																															4374		
ⵥ																																851	

Pour évaluer la capacité de notre système à reconnaître des textes dans des nouvelles polices non utilisées lors de l'apprentissage, nous avons effectué le deuxième test sur quatre autres polices existantes dans la base d'images APWID.

Les résultats reportés dans la Table 28 prouvent que notre système a montré sa robustesse face à des nouvelles polices non utilisées en phase d'apprentissage avec de bons taux de reconnaissance pour les deux polices (*Tifinaghe-Agoug, Tifinaghe-IRCAM2*) et des taux raisonnables pour les deux autres polices (*Teddus Standard, Tassafout Standard*). La différence dans la taille des ensembles de test d'une police à l'autre est due aux éliminations des mots complexes en segmentation selon chaque police, comme mentionné précédemment.

Table 28. Résultats du système sur les nouvelles polices et les nouvelles tailles

Polices	**Nombre de caractères**	**Taux de reconnaissance (%)**
Tifinaghe-Agoug	62182	95,04
Tifinaghe-IRCAM2	68172	95,71
Teddus Standard	26766	93,16
Tassafout Standard	76428	93,87

La police *Tifinaghe-IRCAM2* est une police assez régulière qui présente une légère différence avec la police *Tifinaghe-Ircam*, la police la plus régulière dans l'ensemble d'apprentissage utilisé, ce qui explique le bon taux de reconnaissance atteint pour cette police. Le système montre également sa robustesse face à la police *Tifinaghe-Agoug* avec une légère diminution du taux de reconnaissance. Cette police, lorsqu'elle est utilisée, change par erreur le caractère Yaj (ⵊ) vers le caractère Yi (ⵉ) ce qui justifie la diminution du taux ainsi mentionné. En ce qui concerne les deux autres polices (*Teddus*et *Tassafout*), les taux de reconnaissance obtenus semblent raisonnables vu l'irrégularité de ces deux polices. La Figure 93 illustre des exemples de mots de la base du test avec les deux dernières polices et l'effet des prétraitements sur la forme de leurs caractères.

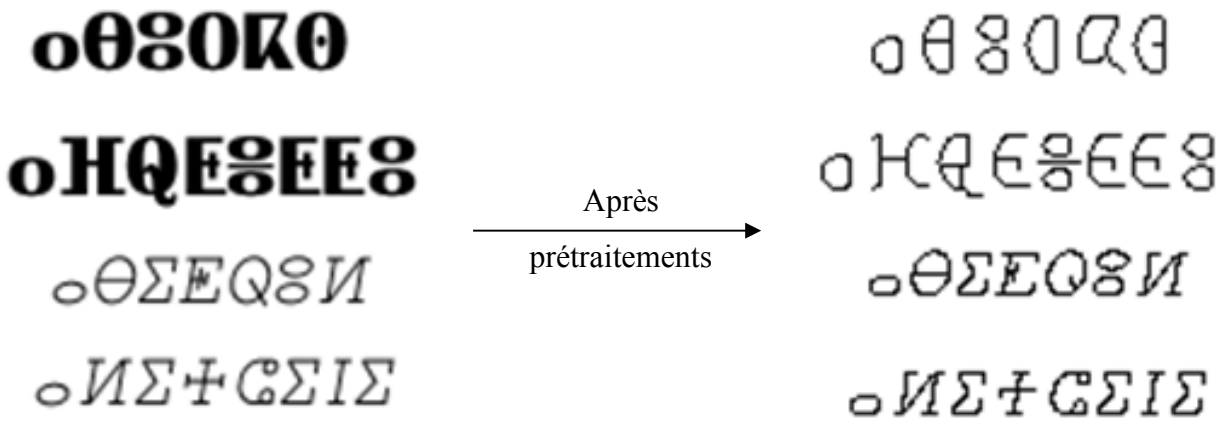


Figure 93. Mots avec les polices Teddus et Tassafout avant et après prétraitements

Conclusion

Ce chapitre a été consacré aux contributions apportées au domaine de la reconnaissance automatique hors ligne de l'écriture Amazighe imprimée. Dans ce contexte, nous avons conçu une base d'images, nommée APWID, des mots Amazighs imprimés. Cette base contient 1148800 images des mots Amazighs rendus avec une variété de polices, tailles et styles permettant ainsi l'évaluation et la comparaison à grande échelle des systèmes OCR Amazighs imprimés. Les images de la base ont été générées par une procédure automatique et sont accompagnées par un fichier XML décrivant la séquence de caractères contenue dans l'image, ses statistiques et certaines informations sur l'image et sa procédure de rendu. Cette base est accessible gratuitement au public sur Internet.

En outre, depuis sa standardisation, le script Tifinagh et de plus en plus utilisé au Maroc dans les différents moyens d'informations et de communication de telle façon qu'il est souvent capturé dans les images de scènes naturelles, de publicité et du Web. D'une autre part, ce script existe souvent dans un environnement multilingue. Ceci nous a mené à concevoir un système de bout en bout pour le traiter. Le système proposé procède en plusieurs étapes commençant par la détection du texte, passant par l'identification de son script jusqu'à sa reconnaissance automatique.

Conclusion générale

Cette thèse a présenté des contributions variées au problème de la reconnaissance automatique hors ligne de l'écriture Amazighe imprimée et manuscrite. En effet, en premier lieu, nous nous sommes attaqués à l'écriture Amazigh manuscrite en Tifinagh. En remarquant le manque existant au niveau des systèmes OCR performants, notre but était de concevoir un descripteur robuste visant à fournir une description discriminante des caractères Amazighs lors de l'extraction des caractéristiques. Ce descripteur se base sur la décomposition de l'image du caractère isolé en plusieurs zones chevauchées qui prennent en compte la structure et la forme des caractères, ensuite, pour chaque zone, des caractéristiques statistiques sont extraites. Un perceptron multicouches a été entrainé afin de décider de la classe d'un caractère d'entrée. Ce descripteur a été amélioré ultérieurement et combiner avec différents algorithmes de classification pour fournir un système robuste de reconnaissance automatique de l'écriture manuscrite Amazighe. Cette amélioration se caractérise aussi par l'utilisation d'un système de rejet pour démunir le taux d'erreurs du système. Les deux systèmes proposés ont été testés sur la base d'images de référence des caractères Amazighs manuscrits AHMCD.

Dans le cas de l'écriture imprimée, le manque d'une base d'image de référence pour tester les différents systèmes développés nous a poussé à créer une nouvelle base d'images APWID contenant chacune un mot amazigh rendu avec une variété de polices, tailles et styles. La variété des données de la base permet d'effectuer une évaluation et une comparaison à grande échelle des systèmes de reconnaissance de l'écriture imprimée. Cette base a servi pour tester un système de reconnaissance automatique hors ligne des caractères Amazighs imprimés, ce système se base sur le même descripteur proposé pour l'écriture manuscrite. Les résultats obtenus par ce système pour les tests effectués sur la base APWID montrent sa robustesse face aux différentes polices, tailles et styles utilisées pour l'écriture. Le but de ce système est de fournir un système capable de reconnaitre les documents Amazighs imprimés avec différents polices mais aussi pour le texte incorporé dans les images de scènes naturelles. Cependant, ce dernier doit être localisé dans l'image et peut exister dans un environnement multilingue. Pour surmonter ces problèmes, nous avons proposé un système composé de plusieurs étapes permettant ainsi de détecter et extraire les blocs de texte pour identifier leurs scripts parmi les trois scripts les plus utilisés au Maroc (Arabe, Latin et Tifinagh) avant de procéder à leur reconnaissance automatique. Dans l'étape d'identification du script, nous avons entrainé un réseau de neurones convolutifs CNN qui a montré sa puissance pour différencier entre les 3 scripts mais aussi pour les nombres. Dans la phase de reconnaissance automatique, pour le script Tifinagh, nous avons utilisé le système déjà proposé et testé sur la base APWID.

Les problèmes rencontrés durant les travaux réalisés nous ont permis d'envisager la poursuite de nos travaux afin d'améliorer les performances. En effet, nous avons remarqué que les opérations des prétraitements telles que le seuillage, la normalisation et la squelettisation étaient parmi les facteurs qui diminuent la performance des systèmes OCR surtout dans le cas imprimé. Ce qui nous encourage à étudier les différents algorithmes des prétraitements pour choisir les mieux adaptés au script Tifinagh. Alors que pour le cas manuscrit, en plus de ces facteurs, la segmentation des mots en caractères était le problème majeur durant les prétraitements. Étant donné que la reconnaissance automatique par l'approche holistique est quasi-impossible, la segmentation des mots en caractères devient obligatoire, malgré la non-cursivité de l'écriture Amazighe, on peut souvent rencontrer des problèmes tels que des

caractères touchés ou chevauchés. Cependant, dans la littérature, aucun des travaux réalisés n'a traité le problème de la segmentation dans le cas du Tifinagh, ce qui nous pousse à effectuer une étude des principaux problèmes rencontrés et évaluer les différents algorithmes de segmentation développés.

D'une autre part, le travail proposé concernant la reconnaissance automatique du texte en Tifinagh à partir des images Web ou de scènes naturelles nous a permis de considérer des nouvelles perspectives. En effet, la puissance du CNN utilisé pour identifier le script Tifinagh dans un environnement multilingue nous a poussé à penser à exploiter la puissance des algorithmes de l'apprentissage en profondeur pour concevoir un système regroupant dans un seul modèle les traitements nécessaires de ce genre d'images depuis la détection du texte jusqu'à sa reconnaissance. Une base de test des de scènes naturelles contenant le script Tifinagh sera développée en parallèle.

Bibliographie

1. Abaynarh M, EL Fadili H, Zenkouar L (2015) Enhanced feature extraction of handwritten characters and recognition using artificial neural networks. Journal of Theoretical and Applied Information Technology 72:355–365
2. Abaynarh M, Zenkouar L (2015) Offline Handwritten Characters Recognition Using Moments Features and Neural Networks. Computer Technology and Application 6:19–29. doi: 10.17265/1934-7332/2015.01.004
3. Abu-Ain W, Abdullah SNHS, Bataineh B, Abu-Ain T, Omar K (2013) Skeletonization Algorithm for Binary Images. Procedia Technology 11:704–709. doi: 10.1016/j.protcy.2013.12.248
4. Afli H, Barrault L, Schwenk H (2015) OCR Error Correction Using Statistical Machine Translation. In: 16th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2015). pp 175–191
5. Aggarwal A, Singh C (2016) Zernike Moments-Based Gurumukhi Character Recognition. Applied Artificial Intelligence 30:429–444. doi: 10.1080/08839514.2016.1185859
6. Aggarwal A, Singh K, Singh K (2015) Use of gradient technique for extracting features from handwritten Gurmukhi characters and numerals. In: Procedia Computer Science. pp 1716–1723
7. Ahmed P (1995) A neural network based dedicated thinning method. Pattern Recognition Letters 16:585–590. doi: 10.1016/0167-8655(95)80004-D
8. Ait Ouguengay Y, Taalabi M (2009) Elaboration d'un réseau de neurones artificiels pour la reconnaissance optique de la graphie amazighe: Phase d'apprentissage. Systèmes intelligents, Théories et applications
9. Ameur M, Bouhjar A, Boukhris F, Boukouss A, Boumalk A, Elmedlaoui M, Iazzi E, Souifi H (2014) Initiation à la langue amazighe, IRCAM
10. Amri S, Zenkouar L, Outahajala M (2017) Build a Morphosyntaxically Annotated Amazigh Corpus. In: Proceedings of the 2nd international Conference on Big Data, Cloud and Applications - BDCA'17. pp 1–7
11. Amrouch M, Es Saady Y, Rachidi A, El Yassa M, Mammass D (2012) A Novel Feature Set for Recognition of Printed Amazigh Text using Maximum Deviation and HMM. International Journal of Computer Applications 44:23–30. doi: 10.5120/6316-8659
12. Amrouch M, Rachidi A, El Yassa M, Mammass D (2012) Handwritten Amazigh Character Recognition System Based on Continuous HMMs and Directional Features. International Journal of Modern Engineering Research 2:436–441
13. Andries P (2005) Normalisation et état des lieux de la prise en charge de l'amazighe et des tifinaghes. Asinag - Numéro 9 1–23
14. Andries P, Yergeau F, LaBonté A (2004) Proposition d'ajout de l'écriture tifinagh au répertoire de l'ISO/CEI 10646
15. Aradhya VNM, Kumar GH, Shivakumara P (2006) Skew Detection Technique for Binary Document Images based on Hough Transform. Int Journal of Information Technology 3:194–200

16. Arica N, Yarman-Vural FT (2001) An overview of character recognition focused on offline handwriting. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS 31:216–233. doi: 10.1109/5326.941845

17. Arora S, Bhattacharjee D, Nasipuri M, Basu DK, Kundu M (2011) Complementary features combined in a MLP-based system to recognize handwritten Devnagari character. Journal of Information Hiding and Multimedia Signal Processing 2:71–77

18. Askari M, Asadi M, Bidgoli AA (2016) Isolated Persian/Arabic handwriting characters: Derivative projection profile features, implemented on GPUs. Journal of AI and Data Mining 4:9–17

19. EL Ayachi R, Fakir M, Bouikhalene B (2011) Recognition of Tifinaghe Characters Using a Multilayer Neural Network. International Journal Of Image Processing 5:109–118. doi: 10.1109/MMCS.2009.5256681

20. El Barkani B (2010) Le choix de la graphie tifinaghe pour enseigner, apprendre l'amazighe au Maroc : conditions, représentation et pratiques. Université Jean Monnet - Saint-Etienne

21. Basseville M (1989) Distance measures for signal processing and pattern recognition. Signal processing 18:349–369

22. Bencharef O, Fakir M, Minaoui B, Bouikhalene B (2011) Tifinagh Character Recognition Using Geodesic Distances, Decision Trees & Neural Networks. International Journal of Advanced Computer Science and Applications 51–55

23. Blanco JL (2014) Tifinagh & the Explorations in Cursiveness and Bicameralism in the Tifinagh Script. Université de Reading

24. Blu T, Thévenaz P, Unser M (2004) Linear interpolation revitalized. IEEE Transactions on Image Processing 13:710–719. doi: 10.1109/TIP.2004.826093

25. Boulaknadel S, Ataa Allah F (2011) Initiative pour le développement d'un corpus de la langue amazighe

26. Boutaounte M, Ouadid Y (2016) Tifinagh characters recognition using simple geometric shapes. Indonesian Journal of Electrical Engineering and Computer Science 3:235–239. doi: 10.11591/ijeecs.v3.i1.pp235-239

27. Breiman L (1984) Classification and Regression Trees, 1st ed. Routledge, New York

28. Casey RG, Lecolinet E (1996) Survey of Methods and Strategies in Character Segmentation. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 18:690–706. doi: 10.1109/34.506792

29. Cecotti H, Vajda S (2013) Rejection Schemes in Multi-class Classification - Application to Handwritten Character Recognition. 12th International Conference on Document Analysis and Recognition 445–449. doi: 10.1109/ICDAR.2013.96

30. Chacko BP, Vimal Krishnan VR, Raju G, Babu Anto P (2012) Handwritten character recognition using wavelet energy and extreme learning machine. International Journal of Machine Learning and Cybernetics 3:149–161. doi: 10.1007/s13042-011-0049-5

31. Chaudhuri A (2010) Some Experiments on Optical Character Recognition Systems for different Languages using Soft Computing Techniques. Patna Campus, India

32. Chaudhuri A, Mandaviya K, Badelia P, Ghosh SK (2017) Optical Character Recognition

Systems. pp 9–41

33. Cheriet M, Kharma N, Liu C, Suen C (2007) Character recognition systems: a guide for students and practitioners

34. Chow CK (1970) On Optimum Recognition Error and Reject Tradeoff. IEEE Transactions on Information Theory 16:41–46. doi: 10.1109/TIT.1970.1054406

35. Cunningham P, Delany SJ (2007) K -Nearest Neighbour Classifiers

36. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Proceedings of CVPR 2005, the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp 886–893

37. Dimauro G, Impedovo S, Pirlo G, Salzo A (1997) Zoning design for handwritten numeral recognition. In: Del Bimbo A (ed) Image Analysis and Processing. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 592–599

38. Ding K, Liu Z, Jin L, Zhu X (2008) A comparative study of Gabor feature and gradient feature for handwritten Chinese character recognition. In: Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, ICWAPR '07. pp 1182–1186

39. Djematene A, Taconet B, Zahour A (1997) A geometrical method for printed and handwritten berber characters recognition. In: International Conference on Document Analysis and Recognition. pp 564–567

40. Dong J, Chen Y, Yang Z, Ling BWK (2017) A parallel thinning algorithm based on stroke continuity detection. Signal, Image and Video Processing 11:873–879. doi: 10.1007/s11760-016-1034-y

41. Eglin V, Bres S, Rivero C (2007) Hermite and Gabor transforms for noise reduction and handwriting classification in ancient manuscripts. International Journal of Document Analysis and Recognition (IJDAR) 9:101–122. doi: 10.1007/s10032-007-0039-z

42. Eikvil L (1993) OCR Optical character recognition. Computer Communications. doi: 10.1016/0140-3664(86)90284-7

43. Es Saady Y, Amrouch M, Rachidi A, El Yassa M, Mammass D (2014) Handwritten Tifinagh Character Recognition Using Baselines Detection Features. International Journal of Scientific & Engineering Research 5:1177–1182

44. Es Saady Y, Rachidi A, El Yassa M, Mammass D (2011) AMHCD : A Database for Amazigh Handwritten Character Recognition Research. International Journal of Computer Applications 27:44–48

45. Es Saady Y, Rachidi A, El Yassa M, Mammass D (2011) Printed Amazigh Character Recognition by a Syntactic Approach using Finite Automata. 10:1–8

46. Es Saady Y, Rachidi A, El Yassa M, Mammass D (2011) Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character. International Journal of Advanced Science and Technology 33:33–50

47. Farahmand A, Sarrafzadeh A, Shanbehzadeh J (2013) Document Image Noises and Removal Methods. In: Proceedings of the Internatioanl MultiConference of Engineer and Computer Scientists

48. Fating K, Ghotkar A (2014) Performance Analysis of Chain Code Descriptor For Hand

Shape Classification. International Journal of Computer Graphics & Animation (IJCGA) 4:9–19

49. Freeman H (1961) On the Encoding of Arbitrary Geometric Configurations. IEEE Transactions on Electronic Computers EC-10:260–268. doi: 10.1109/TEC.1961.5219197

50. FREITAS COA, OLIVEIRA LS, BORTOLOZZI F, AIRES SBK (2007) HANDWRITTEN CHARACTER RECOGNITION USING NONSYMMETRICAL PERCEPTUAL ZONING. International Journal of Pattern Recognition and Artificial Intelligence 21:135–155. doi: 10.1142/S021800140700534X

51. Fu X, Xu Y, Tong L (2011) Document image skew adjusting based on the feedback information recognized by OCR. 2011 IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011 376–378. doi: 10.1109/ICCSN.2011.6013852

52. Fumera G, Roli F, Giacinto G (2000) Reject option with multiple thresholds. Pattern Recognition 33:2099–2101. doi: 10.1016/S0031-3203(00)00059-5

53. Gabor D (1946) Theory of communication. Part 1: The analysis of information. Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering 93:429–441. doi: 10.1049/ji-3-2.1946.0074

54. Genuer R, Poggi J-M, Tuleau-Malot C (2010) Variable selection using random forests. Pattern Recognition Letters 31:2225–2236. doi: 10.1016/j.patrec.2010.03.014

55. Ghosh AK (2006) On optimum choice of k in nearest neighbor classification. Computational Statistics & Data Analysis 50:3113–3123. doi: 10.1016/j.csda.2005.06.007

56. Ghosh D, Dube T, Shivaprasad a P (2010) Script Recognition – A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence XX:1–21. doi: 10.1109/TPAMI.2010.30

57. Gounane S, Fakir M, Bouikhalene B (2011) Recognition of Tifinagh Characters Using Self Organizing Map And Fuzzy K-Nearest Neighbor. Global Journal of Computer Science and Technology 11

58. Gounane S, Fakir M, Bouikhalene B (2013) Handwritten Tifinagh Text Recognition Using Fuzzy K-Nearest Neighbor and Bigram Language Model. International Journal of Advanced Computer Science and Applications 361–367

59. Gounane S, Fakir M, Bouikhalene B (2015) Performance Comparison of Fuzzy K-NN SVM and ANN Combined with N-gram Language Model for Handwritten Tifinagh Character Recognition. In: Second international Conference onBusiness Intelligence (CBI'15). pp 31–38

60. Heckerman D (2008) A Tutorial on Learning with Bayesian Networks. In: Holmes DE, Jain LC (eds) Innovations in Bayesian Networks: Theory and Applications. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 33–82

61. Hinton GE (2010) Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of the 27th International Conference on Machine Learning

62. Hsu C-W, Chang C-C, Lin C-J (2003) A Practical Guide to Support Vector Classification. BJU international 101. doi: 10.1177/02632760022050997

63. Hsu C, Lin C (2002) A comparison of methods for multiclass support vector machines.

Neural Networks, IEEE Transactions on 13:415–425. doi: 10.1109/TNN.2002.1021904

64. Hu M-K (1962) Visual pattern recognition by moment invariants. IRE Transactions on Information Theory 8:179–187. doi: 10.1109/TIT.1962.1057692

65. Hu P, Zhao Y, Yang Z, Wang J (2002) Recognition of gray character using gabor filters. In: Proceedings of the 5th International Conference on Information Fusion, FUSION 2002. pp 419–424

66. Huang W, Lin Z, Yang J, Wang J (2013) Text Localization in Natural Images using Stroke Feature Transform and Text Covariance Descriptors. doi: 10.1109/ICCV.2013.157

67. Impedovo D, Pirlo G (2014) Zoning methods for handwritten character recognition: A survey. Pattern Recognition 47:969–981. doi: 10.1016/j.patcog.2013.05.021

68. Impedovo S, Lucchese MG, Pirlo G (2006) Optimal Zoning Design by Genetic Algorithms. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS 36:833–846

69. Jain AK (2000) Statistical Pattern Recognition : A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22:4–37

70. Jain B, Borah M A Survey paper on skew detection of offline handwritten character recognition system. International Journal of Computer Engineering and Applications 6

71. John J, Pramod K V., Balakrishnan K (2012) Unconstrained handwritten Malayalam character recognition using wavelet transform and support vector machine classifier. Procedia Engineering 30:598–605. doi: 10.1016/j.proeng.2012.01.904

72. John R, Raju G, Guru DS (2008) 1D wavelet transform of projection profiles for isolated handwritten Malayalam character recognition. In: Proceedings - International Conference on Computational Intelligence and Multimedia Applications, ICCIMA 2007. pp 481–485

73. Joshi SR (2016) Restoration of Degraded Images for Text Detection and Recognition. International Journal of Computer Applications 134:25–29

74. Kale K V., Chavan S V., Kazi MM, Rode YS (2013) Handwritten Devanagari compound character recognition using legendre moment: An artificial neural network approach. In: Proceedings - 2013 International Symposium on Computational and Business Intelligence, ISCBI 2013. pp 274–278

75. Kaltenmeier A, Caesar T, Gloger JM, Mandler E Sophisticated topology of hidden Markov models for cursive script recognition. In: Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93). IEEE Comput. Soc. Press, pp 139–142

76. Karsoliya S (2012) Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture. International Journal of Engineering Trends and Technology 3:714–717

77. Kaur A, Baghla S, Kumar S (2015) Study of Various Character Segmentation Techniques for Handwritten Off-Line Cursive Words : a Review. International Journal of Advances in Science Engineering and Technology 3:154–158

78. Kaur A, Dhir R, Lehal GS (2017) A survey on camera-captured scene text detection and extraction: towards Gurmukhi script. International Journal of Multimedia Information Retrieval 1–28. doi: 10.1007/s13735-016-0116-5

79. Kaur A, Malhotra S (2015) Punjabi Handwritten Character Recognition Using Wavelet Based Features. International Journal of New Technologies in Science and Engineering 2:217–226

80. Kaur KA, Bhutani L (2015) A Review on Classification Using Decision Tree. IJCAT - International Journal of Computing and Technology 2:42–46

81. Kawanishi K (1987) Fine classification of printed Thai character recognition using the karhunen-loève expansion. IEE Proceedings E (Computers and Digital Techniques) 134:257–264(7)

82. Kittler J, Hater M, Duin RPW (1996) Combining classifiers. In: Proceedings - International Conference on Pattern Recognition. pp 897–901

83. Kohavi R (1995) A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. International Joint Conference on Artificial Intelligence (IJCAI1995) 1137–1143

84. Kong H (2005) A Study On the Use of 8-Directional Features For Online Handwritten Chinese Character Recognition. In: Eighth International Conference on Document Analysis and Recognition (ICDAR'05). pp 262–266

85. Kotsiantis SB, Zaharakis ID, Pintelas PE (2006) Machine learning: A review of classification and combining techniques. Artificial Intelligence Review 26:159–190. doi: 10.1007/s10462-007-9052-3

86. Krishnamoorthy S, Loganathan R, Soman KP (2010) Recursive Projection Profiling for Text-Image Separation. In: Sobh T, Elleithy K (eds) Innovations in Computing Sciences and Software Engineering. Springer Netherlands, Dordrecht, pp 1–5

87. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. Curran Associates Inc., USA, pp 1097–1105

88. Krylov AS., Lukin AS., Nasonov AV. (2009) Edge-preserving nonlinear iterative image resampling method. In: 2009 16th IEEE International Conference on Image Processing (ICIP). pp 385–388

89. Kumar G (2014) A Detailed Review of Feature Extraction in Image Processing Systems. pp 5–12

90. Kumar SS, Rajendran P, Prabaharan P, Soman KP (2016) Text/Image Region Separation for Document Layout Detection of Old Document Images Using Non-linear Diffusion and Level Set. Procedia Computer Science 93:469–477. doi: 10.1016/j.procs.2016.07.235

91. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. doi: 10.1038/nature14539

92. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE 86:2278–2323. doi: 10.1109/5.726791

93. Li Y (1992) Reforming the theory of invariant moments for pattern recognition. Pattern Recognition 25:723–730. doi: 10.1016/0031-3203(92)90135-6

94. Liu CL, Fujisawa H (2008) Classification and learning methods for character recognition: Advances and remaining problems. Studies in Computational Intelligence 90:139–161.

doi: 10.1007/978-3-540-76280-5_6

95. Liu XLX, Samarabandu J (2007) An edge-based text region extraction algorithm for indoor mobile robot navigation. In: IEEE International Conference Mechatronics and Automation, 2005. pp 2043–2050

96. Louloudis G, Gatos B, Halatsis C (2007) Text line detection in unconstrained handwritten documents using a block-based hough transform approach. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. pp 599–603

97. Lu S, Chen T, Tian S, Lim J-H, Tan C-L (2015) Scene text extraction based on edges and support vector regression. International Journal on Document Analysis and Recognition (IJDAR) 18:125–135. doi: 10.1007/s10032-015-0237-z

98. Lu Y, Tan CL (2003) Improved nearest neighbor based approach to accurate document skew estimation. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. pp 503–507

99. Mady AMM, Omar K (2011) A comparative study of Voronoi algorithm construction in thinning. In: Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, ICEEI 2011

100. Mancas-Thillou C, Gosselin B (2007) Color text extraction with selective metric-based clustering. Computer Vision and Image Understanding 107:97–107. doi: 10.1016/j.cviu.2006.11.010

101. Mantas J (1986) An overview of character recognition methodologies. Pattern Recognition 19:425–430. doi: 10.1016/0031-3203(86)90040-3

102. Miciak M (2008) Character recognition using radon transformation and principal Component analysis in postal applications. In: Proceedings of the International Multiconference on Computer Science and Information Technology, IMCSIT 2008. pp 495–500

103. Mishra A, Alahari K, Jawahar C ~V. (2011) An MRF Model for Binarization of Natural Scene Text. In: Proceedings of International Conference on Document Analysis and Recognition

104. Mori S, Suen CY, Yamamoto K (1992) Historical Review of OCR Research and Development. In: Proceedings of the IEEE. pp 1029–1058

105. Mouchère H (2007) Etude des mécanismes d'adaptation et de rejet pour l'optimisation de classifieurs : Application à la reconnaissance de l'écriture manuscrite en-ligne. l'Institut National des Sciences Appliquées de Rennes

106. Mouchère H, Anquetil E (2006) A Unified Strategy to Deal with Different Natures of Reject. In: The 18th International Conference on Pattern Recognition (ICPR'06)

107. Moudni H, Er-rouidi M, Oujaoura M, Bencharef O (2013) Recognition of Amazigh characters using SURF & GIST descriptors. International Journal of Advanced Computer Science and Applications 41–44. doi: 10.14569/SpecialIssue.2013.030208

108. Muñoz A, Blu T, Unser M (2001) Least-Squares Image Resizing Using Finite Differences. IEEE Transactions on Image Processing 10:1365–1378

109. Naïm P, Wuillemin P-H, Leray P, Pourret O, Becker A (2007) Réseaux Bayésiens, EYROLLES

110. Nguyen DT, Vo DB, Nguyen TM, Nguyen TG (2007) A Robust Document Skew Estimation Algorithm Using Mathematical Morphology. 19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007) 496–503. doi: 10.1109/ICTAI.2007.124

111. Obaidullah SM, Santosh KC, Halder C, Das N, Roy K (2017) Automatic Indic script identification from handwritten documents: page, block, line and word-level approach. International Journal of Machine Learning and Cybernetics 0:0. doi: 10.1007/s13042-017-0702-8

112. Otsu N (1979) A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics 9:62–66. doi: 10.1109/TSMC.1979.4310076

113. Ouadid Y, Boutaounte M, Fakir M, Minaoui B (2016) Tifinagh Character Recognition using Harris Corner Detector and Graph Representation. International Journal of Computer Applications 149:17–23

114. Oujaoura M, EL Ayachi R, Bencharef O, Chihab Y, Jarmouni B (2013) Application of Data Mining Tools for Recognition of Tifinagh Characters. International Journal of Advanced Computer Science and Applications 2–5

115. Oujaoura M, Minaoui B, Fakir M, El Ayachi R, Bencharef O (2014) Recognition of Isolated Printed Tifinagh Characters. International Journal of Computer Applications 85:1–13. doi: 10.5120/14802-3005

116. Oulamara A, Duvernoy J (1988) An application of the hough transform to automatic recognition of berber characters. Signal Processing 14:79–90

117. Outahajala M, Zenkouar L, Rosso P (2011) Building an annotated corpus for Amazighe. In: Proceedings of 4th International Conference on Amazigh and ICT. pp 1–10

118. Papandreou A, Gatos B (2011) A novel skew detection technique based on vertical projections. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. pp 384–388

119. Park J, Govindaraju V, Srihari S (1998) Ocr in a hierarchical feature space. In: IEEE International Conference on Systems, Man, and Cybernetics. pp 4324–4329

120. Pearl J (1985) Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning

121. Phokharatkul P, Sankhuangaw K, Somkuarnpanit S, Phaiboon S, Kimpan C (2007) Off-Line Hand Written Thai Character Recognition using Ant-Miner Algorithm. International Journal of Computer and Information Engineering 1:2586–2591

122. Pirlo G, Impedovo D (2012) Adaptive membership functions for handwritten character recognition by Voronoi-based image zoning. IEEE Transactions on Image Processing 21:3827–3837. doi: 10.1109/TIP.2012.2199328

123. Pitrelli JF, Perrone MP (2003) Confidence-Scoring Post-Processing for Off-Line Handwritten-Character Recognition Verification. In: Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03). pp 1–5

124. Plötz T, Fink GA (2009) Markov models for offline handwriting recognition: A survey. International Journal on Document Analysis and Recognition 12:269–298. doi: 10.1007/s10032-009-0098-4

125. Radtke PVW, Oliveira LS, Sabourin R, Wong T (2003) Intelligent Zoning Design Using Multi-Objective Evolutionary Algorithms. In: Seventh International Conference on Document Analysis and Recognition

126. Rahman A, Alam H, Fairhurst M (2002) Multiple Classifier Combination for Character Recognition: Revisiting the Majority Voting System and Its Variations. Document Analysis Systems 167–178

127. Raj A (2016) An optical character recognition of machine printed Oriya script. In: Proceedings of 2015 3rd International Conference on Image Information Processing, ICIIP 2015. pp 543–547

128. Rajput G., Mali S. (2010) Fourier Descriptor based Isolated Marathi Handwritten Numeral Recognition. International Journal of Computer Applications 3:8–13. doi: 10.5120/724-1017

129. Reddy Ks, Linga Reddy D (2013) Enlargement of Image Based Upon Interpolation Techniques. International Journal of Advanced Research in Computer and Communication Engineering 2

130. Rehman A, Mohamad D, Sulong G (2009) Implicit vs explicit based script segmentation and recognition: a performance comparison on benchmark database. International Journal of Open Problems in Computer Science and Mathematics 2:352–364

131. Rehman A, Mohammad D, Sulong G, Saba T (2009) Simple and effective techniques for core-region detection and slant correction in offline script recognition. In: ICSIPA09 - 2009 IEEE International Conference on Signal and Image Processing Applications, Conference Proceedings. pp 15–20

132. Russell S, Norvig P (2010) Intelligence artificielle

133. Ryu J, Koo H Il, Cho NI (2015) Word Segmentation Method for Handwritten Documents based on Structured Learning. Signal Processing Letters, IEEE 22:1161–1165. doi: http://dx.doi.org/10.1109/LSP.2015.2389852

134. Saba T, Rehman A, Al-Dhelaan A, Al-Rodhaan M (2014) Evaluation of current documents image denoising techniques: A comparative study. Applied Artificial Intelligence 28:879–887. doi: 10.1080/08839514.2014.954344

135. Saba T, Rehman A, Elarbi-Boudihir M (2014) Methods and strategies on off-line cursive touched characters segmentation: a directional review. Artificial Intelligence Review 42:1047–1066. doi: 10.1007/s10462-011-9271-5

136. Sandeep BP, Sinha G., Kavita T (2012) Isolated Handwritten Devnagri Character Recognition using Fourier Descriptor and HMM. International Journal of Pure and Applied Sciences and Technology 8:69–74

137. Sarkar M, Chatterjee S (2016) A Survey of Thinning Techniques on Two Dimensional Binary Images. International Journal of Science and Research (IJSR) 5:1375–1390. doi: 10.21275/v5i7.ART2016439

138. Schantz HF (1982) The History of OCR. Manchester Centre

139. Scurmann J (1982) Reading Machines. In: Proceedings of International Joint Conference on Pattern Recognition. Munich, pp 1031–1044

140. Service de la Normalisation Industrielle Marocaine (2004) Prescriptions des claviers

conçus pour la saisie des caractères tifinaghes

141. Shahabi F, Rahmati M (2006) Comparison of Gabor-Based Features for Writer Identification of Farsi / Arabic Handwriting

142. Shi C, Wang C, Xiao B, Zhang Y, Gao S (2013) Scene text detection using graph model built upon maximally stable extremal regions. Pattern Recognition Letters 34:107–116. doi: 10.1016/j.patrec.2012.09.019

143. Shukla MK, Banka H, Yadav KP (2015) Structural Features Extraction for Devnagari and Bangla Language Documents. Indian Journal of Science and Technology 8:74–78. doi: 10.17485/ijst/2015/v8i13/56453

144. Simonyan K, Zisserman A (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. In: 3rd IAPR Asian Conference on Pattern Recognition (ACPR). pp 1–14

145. Smith R (2009) Hybrid page layout analysis via tab-stop detection. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR. pp 241–245

146. Sokolova M, Japkowicz N, Szpakowicz S (2006) Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In: Sattar A, Kang B (eds) AI 2006: Advances in Artificial Intelligence. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 1015–1021

147. Song Y, Liu A, Pang L, Lin S, Zhang Y, Tang S (2008) A Novel Image Text Extraction Method Based on K-Means Clustering. In: Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008). pp 185–190

148. Sureshkumar C (2010) Handwritten Tamil Character Recognition Using RCS Algorithm. 8:21–25

149. Sutha J, Ramaraj N (2007) Neural Network Based Offline Tamil Handwritten Character Recognition System. In: International Conference on Computational Intelligence and Multimedia Applications. pp 451–455

150. Szegedy C, Reed S, Sermanet P, Vanhoucke V, Rabinovich A (2014) Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp 1–12

151. Thévenaz P, Blu T, Unser M (2009) Image interpolation and resampling. Handbook of Medical Image Processing and Analysis 465–493. doi: 10.1016/B978-012373904-9.50037-4

152. Tin Kam Ho (1995) Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition. IEEE Comput. Soc. Press, pp 278–282

153. Touj S, Amara NE Ben, Amiri H (2005) Generalized Hough Transform for Arabic Printed Optical Character Recognition. Int Arab J Inf Technol 2:326–333

154. Trier OD, Jain AK, Taxt T (1996) Feature extraction methods for character recognition - a survey. Pattern Recognition 29:641–662. doi: 10.1016/0031-3203(95)00118-2

155. Vamvakas G, Gatos B, Petridis S, Stamatopoulos N (2007) An Efficient Feature Extraction and Dimensionality Reduction Scheme for Isolated Greek Handwritten Character Recognition. In: Ninth International Conference on Document Analysis and

Recognition (ICDAR 2007)

156. Villiers J De, Barnard E (1993) Backpropagation Neural Nets with One and Two Hidden Layers. IEEE TRANSACTIONS ON NEURAL NETWORKS 4:136–141

157. Wang R, Sang N, Gao C (2015) Text detection approach based on confidence map and context information. Neurocomputing 157:153–165. doi: 10.1016/j.neucom.2015.01.023

158. Wang X, Ding X, Liu C (2005) Gabor filters-based feature extraction for character recognition. Pattern Recognition 38:369–379. doi: 10.1016/j.patcog.2004.08.004

159. Wu T, Ma S (2003) Feature Extraction by Hierarchical Overlapped Elastic Meshing for Handwritten Chinese Character Recognition. In: Seventh International Conference on Document Analysis and Recognition. pp 1–5

160. Yang M, Kpalma K, Ronsin J, Survey A, Feature S (2008) A Survey of Shape Feature Extraction Techniques. Pattern Recognition

161. Ye Q, Doermann D (2015) Text Detection and Recognition in Imagery: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 37:1480–1500. doi: 10.1109/TPAMI.2014.2366765

162. Yin X, Yin X, Huang K, Hao H (2014) Robust Text Detection in Natural Scene Images. 36:970–983

163. You X, Tang YY (2007) Wavelet-based approach to character skeleton. IEEE Transactions on Image Processing 16:1220–1231. doi: 10.1109/TIP.2007.891800

164. Yu S, Li R, Zhang R, An M, Wu S, Xie Y (2013) Performance Evaluation of Edge-directed Interpolation Methods for Noise-free Images. In: Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service. ACM, New York, NY, USA, pp 268–272

165. Zeiler MD, Fergus R (2013) Visualizing and Understanding Convolutional Networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision -- ECCV 2014. Springer International Publishing, Cham, pp 818–833

166. Zhao Z, Fang C, Lin Z, Wu Y (2015) A robust hybrid method for text detection in natural scenes by learning-based partial differential equations. Neurocomputing 168:23–34. doi: 10.1016/j.neucom.2015.06.019

167. Zheng X, Gao Z, Luo W (2014) Feature of statistical projection algorithm-based image retrieval. Applied Mathematics and Information Sciences 8:721–725. doi: 10.12785/amis/080231

168. Zhu Y, Yao C, Bai X (2015) Scene text detection and recognition: recent advances and future trends. Frontiers of Computer Science 10:1–18. doi: 10.1007/s11704-015-4488-0

Table de matières

Université Sidi Mohammed Ben Abdellah
Faculté des Sciences Dhar El Mahraz- Fès
Centre d'Études Doctorales
"Sciences et Technologies"

Thèse De Doctorat Présentée Par : AHARRANE Nabil

Formation Doctorale : STIC

Spécialité : Informatique

Laboratoire : LIIAN

Titre : Contributions à la reconnaissance automatique hors ligne de l'écriture Amazighe imprimée et manuscrite.

Résumé : Cette thèse se focalise sur le sujet de la reconnaissance automatique hors ligne de l'écriture Amazighe imprimée et manuscrite à partir des images. Dans le cadre de l'écriture manuscrite, un système de reconnaissance automatique a été conçu en proposant un nouveau descripteur basé sur des caractéristiques statistiques. Ce descripteur se base sur la décomposition de l'image du caractère isolé en plusieurs zones chevauchées qui prennent en compte la structure et la forme des caractères, ensuite, pour chaque zone, des caractéristiques relatives aux longueurs des traits et de densité sont extraites. Ce système à été testé sur une base d'images de références pour les caractères Amazighs manuscrits appelée AHMCD et les résultats obtenus ont été très satisfaisants. Pour l'écriture manuscrite, une base d'images de référence APWID pour les mots Amazighs imprimés multi-polices, multi-tailles et multi-styles a été proposer pour l'évaluation et la comparaison des systèmes de reconnaissance (OCR) des caractères Amazighs imprimés. La base APWID a servi pour l'évaluation d'un système OCR multi-polices et multi-styles des caractères amazighs pour répondre à la variété des styles utilisés pour les documents Amazighs imprimés mais aussi pour reconnaitre le texte diffusé dans les images Web ou de scènes naturelles. Pour ce genre d'image le texte en Tifinagh peut exister dans un environnement multilingue et dans des arrière-plans complexes. Pour ceci, nous avons proposé un système de bout en bout capable de localiser le texte dans l'image, identifier son script et procéder à la reconnaissance de celui en Tifinagh. La phase d'identification du script s'appuie sur un réseau de neurones convolutifs qui a montré sa performance face aux trois scripts utilisés au Maroc (arabe, latin, Tifinagh).

Mots clés : Reconnaissance optique des caractères, Langue Amazighe, Alphabet Tifinagh, vision par ordinateur, Extraction des caractéristiques, apprentissage automatique, Système OCR, Écriture imprimée, Écriture manuscrite.