ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : 3520

THESE

En vue de l'obtention du : ***DOCTORAT***

Structure de Recherche *: Laboratoire de Recherche en Informatique et Télécommunications*

Discipline *: Sciences de l'Ingénieur*

Spécialité : *Informatique*

Présentée et soutenue le 18/09/2021 par :

Khadija EL GAJOUI

La reconnaissance optique des caractères : Cas de la langue amazighe

JURY

Moulay Driss RAHMANI	PES, Faculté des Sciences, Rabat	Président
Mounir AIT KERROUM	PES, Faculté des Sciences, Kénitra	Rapporteur/ Examinateur
Mohamed EL HAZITI	PES, Ecole Supérieure de Technologie, Salé	Rapporteur/ Examinateur
Khalid MINAOUI	PES, Faculté des Sciences, Rabat	Rapporteur/ Examinateur
Fadoua ATAA ALLAH	DR, Institut Royal de la Culture Amazighe, Rabat	Examinateur
Siham BOULAKNADEL	DR, Institut Royal de la Culture Amazighe, Rabat	Invité
Mohammed OUMSIS	PES, Ecole Supérieure de Technologie, Salé	Directeur de thèse

Année Universitaire : 2021/2022

Remerciement

Les travaux présentés dans le cadre de ce mémoire sont effectués au sein du Laboratoire de Recherche en Informatique et Télécommunications (LRIT), à la Faculté des Sciences de Rabat, sous la direction du Professeur Mohammed OUMSIS et l'encadrement du Professeur Fadoua ATAA ALLAH.
Je tiens, tout d'abord, à exprimer ma plus vive gratitude à mon directeur de thèse Pr. Mohammed OUMSIS, professeur d'enseignement supérieur à l'école supérieure de technologie de Salé (EST), pour la confiance qu'il m'a accordée en acceptant de diriger ce travail, pour sa présence, son sens d'écoute et de compréhension. J'en profite de cette occasion pour présenter ma profonde gratitude à son égard et l'estime respectueuse que je lui porte.
Je tiens particulièrement à exprimer mes chaleureux remerciements à mon encadrante, Pr. Fadoua ATAA ALLAH, directrice de recherche à l'Institut Royal de la Culture Amazighe à Rabat (IRCAM), pour ses qualités humaines, sa disponibilité, ainsi que pour son professionnalisme. Je souhaite également la remercier pour ses précieux conseils, son aide inconditionnel et le soutien moral et scientifique qu'elle a manifesté à mon égard.
Je voudrais adresser toute ma gratitude aux membres du jury, qui ont accepté d'évaluer mon travail de thèse. Merci à M. Moulay Driss RAHMANI, professeur d'enseignement supérieur à la faculté des sciences de Rabat, pour m'avoir fait l'honneur d'accepter de présider le jury.
Je remercie aussi M. Mounir AIT KERROUM, professeur d'enseignement supérieur à la faculté des sciences de Kénitra, d'avoir accepter de rapporter ce travail. Je le remercie pour le temps qu'il a consacré pour lire et évaluer le rapport.
Je voudrais également remercier M. Mohamed EL HAZITI, professeur d'enseignement supérieur à l'école supérieure de technologie de Salé (EST), d'être un rapporteur et examinateur de cette thèse.
Je tiens également à exprimer mes remerciements à M. Khalid MINAOUI, professeur d'enseignement supérieur à la faculté des sciences de Rabat, qui a pris de son temps précieux pour rapporter ce modeste travail.
Je souhaite remercier profondément Mme Siham BOULKNADEL, directrice de recherche à l'Institut Royal de la Culture Amazighe à Rabat (IRCAM), qui a accepté d'examiner ce travail.
Finalement, mais pas pour autant moins important à mes yeux, je voudrais adresser

tout mon amour et ma reconnaissance à ma chère famille, à qui je dédie cette thèse. Je tiens à remercier chaleureusement ma mère, mon père, mes chères sœurs Fatima, Aicha, Naima, Malika et mon cher frère Brahim et mon mari Adil. J'embrasse très fort mes amours Imane, Douae, Fatimazohra, Salma, Lina, Mohammed et le petit Ziad. Ma famille a toujours été mon support et ma motivation pour tout avancement dans ma vie. Sans leurs encouragements, leurs soutiens moral et matériel, je n'aurai jamais arrivée à ce que j'en suis aujourd'hui. Je n'oublierais pas de saluer Hafsa, Bachir, Marouane et Lahsen pour leur présence et leur encouragement. Merci à vous, même si je ne pourrai jamais vous remercier assez.

Résumé

La reconnaissance optique des caractères est un processus qui permet de convertir un texte présenté par une image numérique en un texte modifiable. Le problème de l'OCR a été exploré en profondeur pour plusieurs langues. Néanmoins, il n'y a pas beaucoup de systèmes OCR fiables disponibles pour la langue amazighe. Les études concernant les systèmes existants d'OCR pour cette langue se sont intéressées à l'écriture amazighe en alphabet tifinaghe. Cependant, cet alphabet n'a été généralisé que récemment avec la création de l'Institut Royal de la Culture Amazighe en 2001. D'où l'intérêt de traiter les documents amazighs écrits en alphabet latin et arabe, qui représentaient les alphabets les plus utilisés au Maroc.
Dans cette thèse, nous avons étudié le système OCR ainsi que ses différents modules à savoir le prétraitement, la segmentation, l'extraction des caractéristiques, la classification et le post-traitement. L'objectif de cette thèse est d'élaborer un système capable de reconnaitre des documents scannés anciens et récents, rédigés en amazighe transcrite en caractères latin. Dans ce cadre, nous nous sommes concentrés, en premier lieu, sur la construction d'un corpus représentatif avec différents niveaux : ligne, mot et caractère. Puis, nous avons proposé des systèmes OCR dédiés principalement à notre langue étudiée. Ils sont composés des principaux modules du système OCR et se basent sur les approches les plus pertinentes dans la littérature, et ce dans le but d'étudier leurs comportements par rapport aux caractéristiques de cette langue. Les expérimentations ont été menées sur les systèmes et ont données des résultats satisfaisants exprimés par un taux de reconnaissance qui atteint **98**%.

Mots-clés : *la reconnaissance optique des caractères, écriture amazighe, réseau de neurones, classification, prétraitement*

Abstract

Optical character recognition is a process that converts text presented by a digital image into editable text. The problem of OCR has been explored in depth for several languages. Nevertheless, there are not many reliable OCR systems available for the Amazigh language. Studies concerning existing OCR systems for this language have focused on Amazigh writing in Tifinagh alphabet. However, this alphabet was recently generalized with the creation of the Royal Institute of Amazigh Culture, in 2001. Hence, the interest in dealing with Amazigh documents written in Latin and Arabic alphabets, which represented the most common alphabets used in Morocco.
In this thesis, we have studied the OCR system as well as its different modules, namely the preprocessing, the segmentation, the features extraction, the classi- fication and the post-processing. The objective of this thesis is to develop a system enable to recognize scanned (old and recent) documents, written in Amazigh transcribed in La- tin. In this context, we focused, first, on building a representative corpus with different levels : line, word and character. Then, we proposed OCR systems, dedicated mainly to our studied language. They are composed of the main modules of the OCR system, and are based on various approaches that are most relevant in the literature, with the aim of studying their behavior in relation to the Amazigh characteristics. The experiments were carried out on the systems and gave good results expressed by a recognition rate that reaches up to 98 %.

Key-words : *Optical character recognition , Amazighe writing, neural network, classification, pre-processing*

TABLE DES MATIRES

Liste des abréviations

ANN : Artificial Neural networks (Réseau de neurones artificiels)
CA : Classifieur adaptatif
HMM : Hidden Markov Models (Modèle de Markov Caché)
IRCAM : Institut Royal de la Culture Amazighe
K-NN : k-Nearest Neighbors (k-Voisin le Plus Proche)
LDF : Fonction Discriminante Linéaire
LSTM : Long Short-Term Memory (La Mémoire Court terme et Long Terme)
MLP : Multilayer Perceptron (Perceptron Multicouche)
NN : Neural networks (Réseau de Neurones)
OCR : Optical Character Recognition (La reconnaissance optique des caractères)
PDF : Portable Document Format (Format de document portable)
PPP : Point Par Pouce
RNA : Réseau de Neurones Artificiels
ROC : Reconnaissance Optique des Caractères
SOCR : Système de Reconnaissance Optique des Caractères
SVM : Support Vector Machine (Machines à Vecteurs de Support)
TALN : Traitement Automatique du Langage Naturel
TIC : Technologies de l'information et de la communication
TTS : Transaction Tracking System (Système de Suivi des Transactions)

LISTE DES FIGURES

Liste des tableaux

INTRODUCTION GÉNÉRALE

Contexte

La réplication de la machine des fonctions humaines, comme la lecture, a toujours été un rêve ancien pour l'homme. Cependant, au cours des cinq dernières décennies, la lecture automatique est passée du rêve à la réalité concrète. Récemment, la numérisation des documents est devenue un élément important des systèmes d'information et des flux de travail associés. Grâce à la numérisation, les documents issus des flux des travaux gouvernementaux, administratifs, éducatifs et d'édition sont traités sous une forme plus accessible, consultable et gérable. Pour reconvertir les documents imprimés en texte numérique, les outils de numérisation utilisent la reconnaissance optique des caractères (ROC), connue aussi sous le signe anglais OCR de Optical Character Recognition.
Les moteurs OCR sont utilisés pour reconnaître les éléments de texte individuels d'un document numérisé et les sortir dans un format plus approprié, par exemple sous forme de texte riche ou de fichier PDF consultable. Dans le but de gérer les diverses mises en page des documents et d'écritures spécifiques, les chercheurs dans le domaine ont mis en place des algorithmes spécialisés afin de concourir, avec les systèmes OCR, à produire des résultats optimaux et utilisables. Le critère du succès d'un système OCR est la capacité d'obtenir une précision proche du celle de l'être humain et de le faire avec une rapidité raisonnable. La vitesse est mesurée par le nombre de caractères, ou groupes de caractères prédéfinis (tels que mots, lignes, etc.) reconnus par seconde. Tandis que la précision est mesurée par le pourcentage de caractères ou de groupes de caractères mal classés.

Motivations

La langue amazighe(tamazight) est l'une des plus anciennes langues de l'humanité. Actuellement, elle est parlée par les peuples d'Afrique du Nord, l'oasis égyptienne de Siwa et les Touaregs du Sahara. Malgré le grand nombre des locuteurs et la masse importante de la population marocaine qui utilisent la langue amazighe dans leurs communications quotidiennes, l'utilisation de cette langue n'a pas dépassé officiellement le niveau oral

qu'après la création de l'Institut Royal de la Culture Amazighe (IRCAM). L'objectif de cet Institut est, parmi d'autres, la promotion de la langue et la culture amazighes. Depuis sa création, l'IRCAM veille à la standardisation de l'amazighe au niveau national pour aménager les variantes disponibles. Ce processus de standardisation a abouti à l'élaboration des lexiques, à l'homogénéisation de l'orthographe, à l'élaboration des règles de grammaire. La langue amazighe pose de nombreux défis en matière de traitement automatique du langage naturel. La variation des systèmes d'écriture, le besoin d'uniformiser les structures morphologiques et grammaticales en exploitant toutes les variantes et le manque de corpus linguistiques rendent le traitement informatisé de la langue amazighe très difficile. Les amazighs possèdent depuis l'antiquité un système d'écriture qui leur est propre. Cependant, depuis l'aube de l'histoire, lorsqu'il s'agit de rédiger des documents consistants, les amazighs ont eu recours aux langues et/ou aux alphabets des peuples dominants avec lesquels ils étaient en contact : punique, latin puis arabe. Pour transcrire l'amazighe, de nos jours, trois systèmes d'écriture sont utilisés : le tifinaghe, l'alphabet arabe et l'alphabet latin.

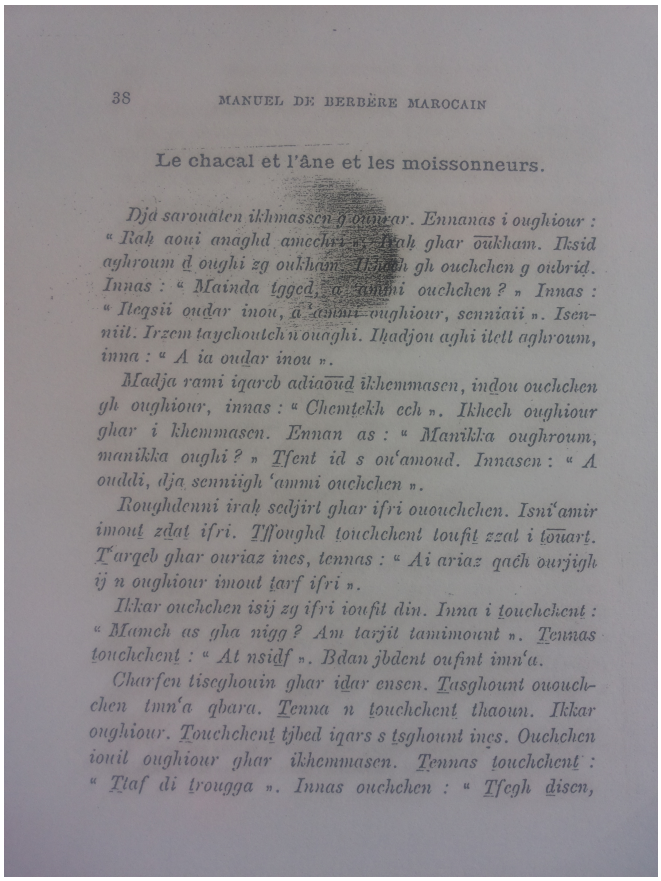
Au Maroc, la langue amazighe a été, récemment, reconnue comme étant une langue officielle du pays. Depuis cette officialisation, nombreux chercheurs ont commencé à s'intéresser à cette langue de plus en plus, et par conséquent plusieurs travaux et études ont été réalisés dans différents domaines, notamment le domaine de la reconnaissance optiques des caractères. Néanmoins, les études existantes sur les systèmes OCR pour l'amazighe se sont concentrées sur l'écriture en alphabet tifinaghe. Par contre les travaux sur la langue amazighe transcrite en alphabet latin sont introuvables.

Etant donné le nombre important des imprimés, des documents et des livres écrits en amazighe transcrit en latin que ça soient anciens ou récents, le traitement de ce type d'écriture devient une nécessité surtout en utilisant la technologie OCR.

Plusieurs difficultés subsistentconcernant le traitement de ce type d'écriture, à savoir : l'insuffisance des travaux et études antécédents axés sur cette écriture, l'absence d'un corpus dédié,ainsi que la proximité du jeu de caractères utilisé dans cette transcription avec l'alphabet français.

Ajoutant à ces défis les difficultéspratiques de ce domaine. Généralement, les documents scannésprésentent souvent des problèmes tels que le faiblecontraste, la variation de la luminosité et le bruit dû à l'opération de la numérisation. En outre dans le cas des documents anciens, la qualité du papier et sa vieillesse peuvent également être un obstacle réel. Ces problèmes peuvent impacter la qualité de l'extraction textuelle et par conséquent la performance de reconnaissance.

La figure ci-aprèsprésente un extrait d'un document ancien écrit en langue amazighe transcrite en latin, affichant une partie du jeu de caractères utilisé et illustrant les différentsproblèmes de numérisation et d'ancienneté cités :



38 MANUEL DE BERBÈRE MAROCAIN

Le chacal et l'âne et les moissonneurs.

Dja sarouaten ikhmassen g ounzar. Ennanas i oughiour : « Rah aoui anaghd ameckri n rah ghar oukham. Iksid aghroum d oughi zg oukham. Ihadh gh ouchchen g oubrid. Innas : « Mainda tgged, a 'ammi ouchchen ? » Innas : « Ilegsii oudar inou, a 'ammi oughiour, senniaii ». Isenniit. Irzem taychoutch n ouaghi. Ihadjou aghi ilelt aghroum, inna : « A ia oudar inou ».

Madja rami iqareb adiaoud ikhemmasen, indou ouchchen gh oughiour, innas : « Chemtekh ech ». Ikhech oughiour ghar i khemmasen. Ennan as : « Manikka oughroum, manikka oughi ? » Tfent id s ou'amoud. Innasen : « A ouddi, dja senniigh 'ammi ouchchen ».

Roughdenni irah sedjirt ghar ifri ououchchen. Isni'amir imout zdat ifri. Tffoughd touchchent toufit zzat i touart. T'arqeb ghar ouriaz ines, tennas : « Ai ariaz qach ourjigh ij n oughiour imout tarf ifri ».

Ikkar ouchchen isij zg ifri ioufit din. Inna i touchchent : « Manch as gha nigg ? Am tarjit tamimount ». Tennas touchchent : « At nsidf ». Bdan jbdent oufint imn'a.

Charfen tisczhouin ghar idar ensen. Tasghount ououchchen tmn'a qbara. Tenna n touchchent thaoun. Ikkar oughiour. Touchchent tjbed iqars s tsghount ines. Ouchchen iouit oughiour ghar ikhemmasen. Tennas touchchent : « Ttaf di trougga ». Innas ouchchen : « Tfegh disen,

FIGURE 1 – Un extrait de texte écrit en amazighe transcrite en caractère latin illustrant les problèmes de numérisation et ancienneté.

Contributions et organisation de la thèse

Dans cette thèse, nous nous sommes concentrés sur l'exploitation de la technologie de la reconnaissance optique des caractères au profil de la langue amazighe, dans l'esprit de la préservation de la culture amazighe stockée dans les documents anciens et lagarantied'une meilleure exploitation des ressources disponibles en langue amazighes. Le nombre important des écrits, ouvrages et livres rédigés particulièrement en langue amazighe transcrite en latin, nous a encouragé à se focaliser sur ce type d'écriture afin de répondre au besoin de la conversion en texte exploitable de l'ensemble de ces documents. Ce manuscrit est organisé en une introduction et conclusion générales en plus de quatre chapitres, dont l'aperçu est présenté ci-dessous :
Le premier chapitre présente une introduction au principe de la reconnaissance optique des caractères. Ce champ de l'Intelligence Artificielle, qui s'intègre dans le sous-domaine de la Vision par Ordinateur et lié à la Reconnaissance de Formes, consiste à lire un

texte dans une image et de le convertir en un fichier consultable. Différents aspects peuvent être distingués notamment l'écriture en ligne / hors ligne et l'écriture imprimée/manuscrit. Le chapitre liste également un ensemble de problèmes et d'intérêts de l'OCR ainsi que ses domaines d'application. D'autre part, ce premier chapitre organise un état de l'art des différentes approches et méthodes développées pour chaque module du système OCR à savoir : l'acquisition, le prétraitement, la segmentation, l'extraction des caractéristiques, la classification et le post-traitement.
L'amazighe est une langue considérée comme peu dotée, dont la population est distribuée sur quelque pays du Nord d'Afrique mais plus présente au Maroc. La création de l'Institut Royal de la Culture Amazighe a été un point de départ pour le traitement de la langue amazighe au Maroc. Le deuxième chapitredécrit les spécificités de cette langue ainsi que les différentes transcriptions utilisées pour présenter cette langue au cours des années. Dans ce chapitre une première contribution dans le domaine est présentée. Elle consiste à élaborer un corpus d'OCR dédié à la langue amazighe transcrite en latin. Ce corpus respecte les particularités de cette langue et il est subdivisé en trois niveaux notamment le corpus ligne, le corpus mot et le corpus caractères.
Dans le troisième chapitre, un système de reconnaissance optique des caractères est élaboré. Dans un premier temps l'étude est axée sur la phase de l'extraction de la caractéristique où le choix est porté sur l'approximation polygonale comme caractéristiques du type statique.Les résultats de ce type de caractéristiques sont assez satisfaisants pour être adopté dans le système global. Dans la phase de classification, le Classifieur Adaptatif (CA) est sélectionné. Des tests ont été effectués pour étudier l'apport du prétraitement ainsi que la robustesse du système conçu. Les résultats enregistrés ont atteint un taux de reconnaissance de 89%.
Dans le quatrième chapitre, deux systèmes ont été développés, en se basant sur l'approche des réseaux de neurones dans la phase de classification. Dans ce cadre, deux types de classifieurs ont était choisi, à savoir : le perceptron multicouche(MLP) et la mémoire court terme et long terme (LSTM). Le test effectué sur ces deux systèmes a donnédes résultats satisfaisants exprimés par des taux de reconnaissance de 92% et 95% respectivement pour le système basé sur MLP et le système basé sur LSTM. La deuxième partie de ce chapitre constitue une comparaison entre les systèmes élaborés. La comparaison a porté dans un premier temps entre le système basé sur MLP et le système basé sur LSTM et dans un deuxième temps entre le système basé sur LSTM et le système basé sur CA. Les résultats ont montré que le système LSTM est le plus approprié à la langue étudiée.

CHAPITRE

1 LA RECONNAISSANCE OPTIQUE DES CARACTÈRES

Sommaire

1.1 Introduction

Au cours des cinq dernières décennies, le rêve de la lecture automatique est devenu une réalité. Malgré que la saisie des données dans un ordinateur à travers le clavier est la manière traditionnelle, cette solution n'est pas toujours la meilleure ni la plus efficace. Dans de nombreux cas, la reconnaissance automatique peut être une alternative (Singhal *et al.*, 2019). Diverses technologies de reconnaissance automatique existent et couvrent les besoins de différents domaines d'application. Parmi les plus pertinentes de ces technologies, il existe la reconnaissance optique des caractères (Optical Character Recognition, OCR).

La reconnaissance optique de caractères est une technologie qui appartient à la famille des technologies d'identification automatique. Elle permet de convertir différents types de documents tels que les documents papiers scannés, les fichiers PDF ou les photos numériques vers des formats modifiables et exploitables (Mori *et al.*, 1999). En d'autres termes, l'OCR est une technique qui transforme les images bitmap au niveau de gris en son code ASCII ou Unicode correspondant.

Un système OCR (SOCR) est dit réussi, si sa capacité d'obtenir une précision est assez élevée et peut être comparable à la capacité humaine. Un autre critère de réussite d'un système est la rapidité raisonnable du traitement qui peur différer d'une application à une autre. La vitesse est mesurée par le nombre de caractères reconnu par seconde, tandis que la précision est mesurée par le pourcentage de caractères ou de groupes de caractères mal classés.

Dans ce chapitre, nous présentons d'abord, les différents aspects de la reconnaissance automatique de l'écriture, ses avantages et limites ainsi que quelques domaines d'applications. Puis, nous introduisons les différentes étapes d'un système de reconnaissance optique des caractères. Ensuite, nous décrivons les approches et les méthodes développées pour les différentes phases du SOCR, à savoir : l'acquisition, le prétraitement, la segmentation, l'extraction de caractéristiques, la classification et enfin le post-traitement. A la fin, nous présentons une conclusion.

1.2 Différents aspects

Sur le plan méthodologique, l'OCR peut être subdivisé, suivant le mode d'écriture, en deux modes : la reconnaissance de l'écriture manuscrite et la reconnaissance de l'écriture imprimée. Dans le texte imprimé, la variabilité est due à la vaste collection de polices ainsi qu'à la qualité de l'impression dans les mécanismes d'impression (matrice de points, jet d'encre, laser, etc.). Dans le texte manuscrit, la variabilité est due à la perte de synchronisme entre les muscles de la main ainsi qu'à la variation des styles qui est due à plusieurs facteurs, incluant mais non limité à l'éducation, l'humeur, la culture, etc. (Es Saady, 2012).

Dans le domaine de la reconnaissance de l'écriture, suivant la nature des informations

disponibles, deux domaines distincts sont considérés. Il s'agit de la reconnaissance statique, dite encore « hors ligne » et la reconnaissance dynamique ou « en ligne ». La reconnaissance optique hors ligne est effectuée après l'écriture ou l'impression, contrairement à la reconnaissance en ligne où l'ordinateur reconnaît les caractères au fur et à mesure qu'ils sont dessinés. Un exemple d'étude se focalisant sur la reconnaissance en ligne est présenté sur (Es Saady, 2012).

Reconnaissance en ligne

Ce mode de reconnaissance s'opère en temps réel (pendant l'écriture). Les symboles sont reconnus au fur et à mesure qu'ils sont écrits à la main. Ce mode est réservé généralement à l'écriture manuscrite, c'est une approche « signal » où la reconnaissance est effectuée sur des données à une dimension. Dans ce type de reconnaissance, l'utilisateur écrit sur une table spéciale. Le processus dynamique d'écriture est capturé, via un numériseur ou une tablette, comme dans le cas dans les systèmes informatiques actuels basé sur le stylo. L'écriture est représentée alors comme un ensemble de points dont les coordonnées sont en fonction du temps. Le système reconnaît l'écriture et envoie le résultat à l'ordinateur.

Reconnaissance hors ligne

Dans le cas de reconnaissance hors ligne, l'écriture de l'utilisateur est acquise par un numériseur. L'entrée du système de ce type de reconnaissance est une image numérisée d'un document préalablement rédigé. On dit que le système travaille sur une instance d'encre numérique (sur une image).

1.3 Intérêts et problèmes

La technologie d'OCR a été appliquée ces dernières années à travers tout le spectre d'industries où elle a participé à révolutionner le processus de gestion des documents. Cependant, cette technologie se confronte à plusieurs défis.

1.3.1 Intérêts

L'objectif principal d'un système OCR est d'importer un texte imprimé vers la machine et de permettre à des documents numérisés de se transformer en documents entièrement consultable et reconnu par les ordinateurs. Il permet aussi à l'utilisateur de récupérer le texte et le modifier avec un minimum d'effort.

Il n'y a plus besoin de ressaisir manuellement les documents importants car l'OCR extraits les informations pertinentes et il se met automatiquement à les transcrire en moins de temps.

Il existe d'autres objectifs et beaucoup plus d'utilisation moderne de cette technologie. Certains d'entre eux sont énumérés ci-dessous.

Conversion de document : L'OCR peut créer un document texte éditable avec peu d'effort humain qui peut également gagner de l'espace et du temps. Il peut être utilisé pour publier le contenu en ligne sous forme de texte au lieu d'images. Cela peut aider non

seulement à économiser de l'espace de stockage, mais aussi à créer une énorme collection de corpus en n'importe quelle langue.
Recherche en ligne : Les moteurs de recherche modernes peuvent être équipés d'un système OCR. Par conséquent, l'utilisateur peut rechercher des images en ligne à partir d'un texte donné. L'importance augmente avec le fait que la majorité des sites d'information utilisent encore des images pour afficher du texte surtout pour les langues peu dotées.
Synthèse vocale : Le but dans ce cas est d'extraire le contenu textuel d'un document dans une image pour qu'il soit lu et exploité. Ce qui peut aider à améliorer l'applicabilité et à développer les systèmes de la synthèse de la parole pour aider les personnes non et malvoyantes.

1.3.2 Problèmes

Malgré les deux décennies de recherche sur la numérisation des documents, toujours il n'existe pas de solution fiable permettant de passer d'un document numérisé à une version en mode texte sans erreurs. Les SOCR permettent de détecter et transposer un mot à partir d'une image, mais ils laissent encore quelques imperfections pour parvenir à une réédition du document ce qu'il peut être dû à différents problèmes. Parmi ces problèmes nous citons :
La qualité du document : un document télécopié ou photocopié plusieurs fois est plus difficile à traiter que la copie originale. L'écriture peut devenir plus mince ou au contraire plus épaisse, dégradée avec des parties du texte qui manquent ou des tâches qui apparaissent, des ouvertures ou des bouchages de boucles ...
L'impression : un document composé est de meilleure qualité qu'un document dactylographié qui, à son tour, est plus clair qu'un texte issu d'une imprimante matricielle. Une imprimante à jet d'encre peut introduire des tâches d'encre et un étalement des caractères, une imprimante laser peut générer des lignes ou des fonds ...
La résolution : la qualité du contenu de l'image dépend principalement de sa résolution. La difficulté souvent rencontrée est de pouvoir adapter la résolution aux différentes tailles de caractères et épaisseurs de graphiques présents dans le document, ne nécessitant pas le même niveau de précision. Le mauvais choix de la résolution peut conduire à un manque d'information dans le cas de la réduction de la résolution et à l'apparition d'un bruit plus abondant dans le cas contraire.
La discrimination de la forme : selon le style de la fonte utilisée (son corps et sa graisse...), le caractère change du graphisme. Le nombre de formes est d'autant plus important que le nombre de styles d'écriture est élevé. De plus, plusieurs caractères présentent une forte ressemblance.
Le support de l'information : tel que le papier, joue également sur les performances de la reconnaissance par sa qualité : son grammage, sa granulation et sa couleur.
L'acquisition : la numérisation en temps réel introduit souvent des distorsions dans l'image. Dans le cas hors ligne la qualité du texte numérisé est un compromis entre les

variations de la position (inclinaison, translation, rétrécissement...), la propreté de la vitre du dispositif de numérisation et sa résolution.
L'inclinaison : est une source d'erreur classique, relativement gênante pour les systèmes de reconnaissance qui utilisent l'horizontal comme référentiel de base pour l'extraction des lignes de texte et la modélisation de la forme des lettres. Elle est de plus en plus maîtrisée grâce à l'existence de logiciels de redressement appliqués systématiquement sur les documents à leur entrée.
Les variations des dimensions : le « pitch » qui représente la distance en millimètres entre les centres de deux pixels adjacents, peut varier entre 10, 12 ou 16. Cette valeur indique la finesse de l'image et par conséquent plus cette distance est réduite, plus le point est net.
Nombre de scripteurs : la difficulté de reconnaissance croît avec ce nombre, divisant l'échelle en trois : mono, multi et omni-scripteurs. En multi-scripteur, le système doit s'adapter à l'écriture de plusieurs scripteurs, tandis qu'en omni-scripteur, le système doit être capable de généraliser son apprentissage à n'importe quel type d'écriture.
Taille du vocabulaire : les applications à vocabulaire limité ($<$ 100 mots) se distinguent de celles à vocabulaire très étendu ($>$ 10 000 mots). Il est évident que dans le premier cas, la complexité est moindre, car la réduction du nombre limite l'encombrement mémoire et favorise l'utilisation des méthodes directes de reconnaissance et donc rapides, par balayage systématique de l'ensemble des mots du lexique.

1.4 Applications des systèmes OCR

L'importance de la reconnaissance de texte provient du rôle qu'elle peut jouer dans différents domaines (Srivastava *et al.*, 2019), (Laique *et al.*, 2020),(Gattawar *et al.*, 2021). Les succès des travaux de recherches ont donné lieu à de nombreux systèmes qui peuvent être intégrés dans plusieurs domaines d'activité parmi lesquelles nous citons :
Culturel : L'utilisation de l'OCR comme moyen de numérisation des documents patrimoniaux peut être considérée aujourd'hui comme la solution ultime aux problèmes de conservation posés par les documents anciens, rares ou précieux. Néanmoins, elle permet d'améliorer indirectement la conservation de ces documents en réduisant la fréquence de leurs utilisations par les usagers.
Bancaire : Une application bien connue est dans le secteur bancaire, où l'OCR est utilisé pour le traitement des chèques sans intervention humaine. Ce traitement peut réduire les temps d'attente dans de nombreuses banques.
Juridique : Le secteur juridique a également connu un important mouvement de numérisation des documents papier. L'OCR a simplifié le processus de la diffusion des documents de recherche textuelle, de sorte qu'ils sont plus faciles à localiser et à travailler avec, une fois classifiés dans une base de données.
La saisie des données : Cette zone couvre les technologies permettant d'entrer de grandes quantités de données restreintes. L'automatisation de ce processus peut réduire le nombre des intervenants, accélérer les transactions et minimiser les coûts de traite-

ment.
Autres applications. Les domaines ci-dessus sont ceux dans lesquels l'OCR a été le plus réussi et le plus largement utilisé. Cependant, de nombreux autres domaines d'application existent à savoir : Aide aux aveugles, lecteurs de plaques d'immatriculation automatiques, cartographie automatique, lecteurs de formulaires, vérification de signature et identification...etc.

1.5 Architecture d'un SOCR

Dans un système OCR, un document est d'abord numérisé par un scanner optique, qui produit image. Ensuite, le système procède à la traduction du texte dans l'image en codes de caractères modifiables tel que l'ASCII ou l'Unicode. Un système de reconnaissance fait appel généralement à six étapes, qui sont : l'acquisition, le prétraite- ment, la segmentation, l'extraction des caractéristiques, la classification et la phase de post-traitement (Prajapati, 2019).
L'architecture d'un système OCR varie d'un système à un autre en fonction des besoins. La figure suivante illustre la structure générale d'un SOCR, sachant que certains de ces étapes ne sont pas obligatoires.

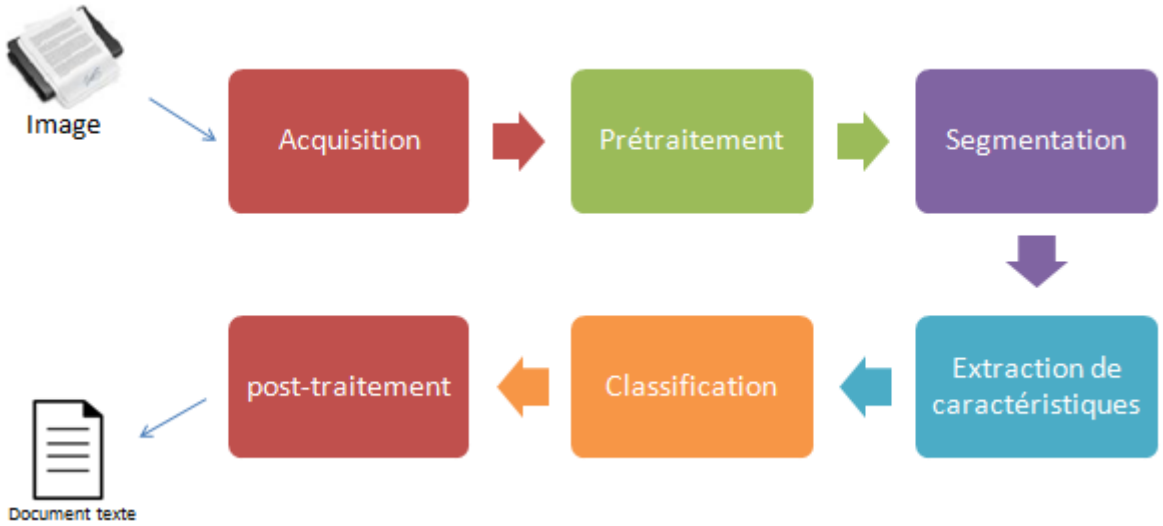


FIGURE 1.1 – L'architecture d'un système OCR.

La première étape est l'acquisition suivie par le prétraitement, où différentes opérations peuvent être effectuées. La troisième étape est la segmentation, qui peut être explicite ou implicite suivie par l'étape d'extraction de caractéristiques. Ensuite vient la reconnaissance qui est l'étape la plus importante, où plusieurs classifieurs peuvent être utilisés.

Cette dernière peut être succédée du post-traitement qui est une étape optionnelle, servant à confirmer les résultats obtenus (Belaïd et Cecotti, 2006), (Martınek *et al.*, 2020).

1.6 Acquisition

La première étape de tout système OCR est l'acquisition d'image (Yasser, 2010). Cette étape consiste à obtenir une image numérique et à la convertir en une forme appropriée qui peut être facilement traitée par ordinateur. Ceci peut être fait hors ligne à l'aide d'un scanner ou en ligne par un stylo numérique (Islam *et al.*, 2017).
L'acquisition hors ligne d'un document est opérée par un balayage optique (Islam *et al.*, 2017). Le résultat est rangé dans un fichier de points, appelés pixels, dont la taille dépend de la résolution qui est exprimée en nombre de points par pouce (ppp) (Kholladi, 2013). Les images résultantes peuvent être de type binaire, où les pixels prennent la valeur 1 ou 0 ; des images en niveau de gris dont la valeur de pixels varie entre 0 et 255 ; ou des images en couleur qui comprennent trois canaux de valeurs de couleurs entre 0 et 255.

1.7 Prétraitement

Afin d'atténuer les variations dues aux problèmes liés à la phase d'acquisition et d'augmenter les chances d'une bonne reconnaissance, certains traitements sont nécessaires. La phase de prétraitement, qui suit généralement l'acquisition (Kholladi, 2013), regroupe l'ensemble de ces traitements, visant le conditionne- ment indispensable à l'identification de l'image. Elle inclut plusieurs opérations qui sont appliquées afin de réduire, autant que possible, les bruits et les variabilités sur l'image numérique (Morita, 2003). Parmi les prétraitements courants existants, nous citons : la binarisation, la correction d'inclination, l'encadrement, l'élimination de bruits, la normalisation et la squelettisation. Dans ce qui suit nous expliquons le principe de chaque traitement ainsi que les travaux réalisés autour de chacun.

1.7.1 Binarisation

Le principe de la binarisation consiste à transformer une image en niveau de gris en une image en noir et blanc, donc de séparer l'information utile (traits des gravures et des caractères en noir) du fond de l'image (image du support papier en blanc). Cette opération permet de réduire la qualité d'informations à traiter, tout en conservant le signal à traiter dans sa quasi-intégralité. Le traitement devient beaucoup plus facile une fois la technique de binarisation est mise en œuvre.

FIGURE 1.2 – Binarisation d'une image en niveau de gris

La binarisation d'une image se fait, en général, à l'aide d'un seuil. Le seuil de binarisation correspond à la limite entre les contrastes forts et faibles de l'image. La binarisation par seuillage consiste à comparer les niveaux de gris d'une image avec un seuil pré-calculé pour décider à quelle des deux classes appartient un point. Dans ce cadre, deux approches ont été développées :

1.7.1.1 Approche globale

Dans cette approche de binarisation (Tao *et al.*, 2003),(Tabbone et Wendling, 2003), un seuil unique est calculé à partir d'une mesure globale sur toute l'image. Le seuil est donc fixe et il nous permet de décider l'appartenance d'un pixel à l'objet ou au fond. Il existe plusieurs méthodes globales telles que la méthode d'Otsu (Jin Soo Noh et Kang Hyeon Rhee, 2005), (Otsu, 1979), la méthode ISODATA (Kefali *et al.*, 2010), la méthode de Kapur (Kefali *et al.*, 2010), la méthode de Tsai (Haji, 2005) et la méthode de Li-Lee (Yin, 2008).
Ces méthodes ne sont pas efficaces pour des sources trop bruitées (Gabarra, 2008). Il devient alors nécessaire d'employer des techniques avec un seuillage adaptatif.

1.7.1.2 Approche locale

Pour la binarisation locale, la classification d'un pixel dépend non seulement du pixel soi-même mais aussi de ses informations locales. Le seuil n'est donc plus unique, mais il est déterminé pour chaque pixel ou pour des régions de pixels aux caractéristiques voisines. Par exemple, c'est la moyenne des pixels du voisinage qui est prise en compte lorsque l'histogramme de deux dimensions est construit. Pour les méthodes locales (ou adaptatives), nous pouvons citer : la méthode de Bernsen (Kim *et al.*, 2017), la méthode non linéaire de Niblack (Niblack, 1990), la méthode de Sauvola (Sauvola et Pietikäinen, 2000), la méthode de Wolf (Kim *et al.*, 2017) et la segmentation hiérarchique floue (Kefali *et al.*, 2010). Fixer ce seuil est très difficile quand le contraste varie dans l'image

(Viard-Gaudin, 2007).
Ces approches sont plus précises, mais elles utilisent différents seuils ou une actualisation selon la région considérée. Ils sont plus sensibles au bruit, mais donnent de meilleurs résultats dans la séparation des composantes. D'autres travaux utilisent la combinaison des approches globales et locales pour faire la binarisation. Le principe est d'appliquer des techniques qui consistent à déterminer localement une région modèle dont les caractéristiques sont ensuite utilisées pour traiter l'image entière (Chang *et al.*, 1999), (Gabarra et Tabbone, 2005). Ces approches sont généralement utilisées pour traiter les documents graphiques (Tabbone *et al.*, 2006).

1.7.2 La correction d'inclination

L'inclinaison peut être produite lors de la saisie, si le document a été placé en biais, ou lors de la numérisation, si la page a été incliné au scanne. La correction de l'angle d'inclinaison est une opération qui consiste à corriger la pente ou à redresser l'inclinaison des lettres dans un mot. Les méthodes de correction d'inclinaison des lignes de texte (également appelées cor- rection de "skew") sont utilisées pour redresser horizontalement les lignes d'écriture obliques. A cet effet, deux étapes sont appliquées : Dans un premier temps, la ligne de base est détectée et l'angle d'inclinaison s est estimé (Chen *et al.*, 2010). Dans un deuxième temps, l'inclinaison est corrigée. La ligne de base est une ligne imaginaire sur laquelle s'aligne l'œil de toutes les lettres. La détection de la ligne de base est l'une des priorités dans le prétraitement du système OCR (Farooq *et al.*, 2005), (Latfi *et al.*, 2006). La ligne de base peut être utilisée dans la normalisation, la détection d'inclinaison (Pechwitz et Margner, 2002), ou pour la segmentation de texte en mots ou en caractères (Amin, 1998), (Arica et Yarman-Vural, 2002). L'estimation de la ligne de base peut aussi être poursuivie pour extraire des traits dépendants (El-Hajj *et al.*, 2005). En utilisant la ligne de base, les caractères et la forme sont classés en trois groupes : ascendants, descendants et marques spéciales appelés diacritiques. Il existe plusieurs méthodes de détection de la ligne de base dans la littérature, mais les méthodes les plus connues sont les suivantes : La méthode de projection horizontale (Al-Badr et Mahmoud, 1995), (Sabbour et Shafait, 2013), la méthode des k-plus proches voisins (Antonacopoulos, 1997) et La méthode de transformation de Hough (Burrow, 2004).

En raison de sa robustesse et sa certitude, la méthode de transformation de Hough reste la méthode la plus utilisée dans la littérature et c'est la méthode que nous avons choisi d'appliquer dans nos expérimentations.

1.7.3 Encadrement

L'encadrement est le processus de la localisation d'un caractère dans une image afin d'éliminer l'espace vide dans cette image. C'est la définition des cordonnées d'un

caractère dans l'image (Bouslimi, 2006). La figure 1.3 montre sur la gauche un exemple de la localisation du caractère 'A' dans une image et sur la droite un encadrement de ce dernier.

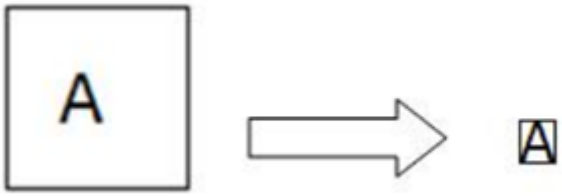


FIGURE 1.3 – Exemple d'encadrement de la lettre 'A'.

Cette fonction permet de localiser une lettre ou un mot dans une image tout en parcourant toute l'image et en localisant les pixels noirs. D'où l'importance d'une telle fonction dans un système de prétraitement puisqu'elle offre un gain de temps que ça soit dans l'apprentissage ou dans la reconnaissance du mot.

1.7.4 Elimination de bruit

Les documents scannés contiennent souvent du bruit provenant de l'imprimante, le scanner, la qualité d'impression ou l'ancienneté du document, d'où la nécessité d'éliminer ce bruit avant procéder au traitement de l'image. Il est plus particulièrement visible dans les zones peu éclairées, où le rapport signal/bruit est faible, mais aussi dans les parties uniformes. Il a pour conséquence la perte de netteté dans les détails. Le bruit peut être réparti en trois catégories : un bruit dépendant du signal, un bruit non dépendant du signal ou un bruit noir et blanc. Dans le domaine de reconnaissance de l'écriture, plusieurs méthodes ont été utilisées pour éliminer le bruit. Les techniques de réduction du bruit sont : les filtres, les opérations morphologiques et la modélisation du bruit.

1.7.4.1 Les filtres

Les filtres peuvent être conçus pour le lissage, l'affinage, le seuillage, la suppression de l'arrière-plan légèrement texturé ou pour le processus de réglage du contraste. Les types de filtres disponibles sont : Les filtres linéaires (filtre de masque de moyennage)(Yasser, 2010) et les filtres non linéaires (Noor et Habib, 2005), (Charles *et al.*, 2012). Le bruit présent dans l'image de la figure en dessous, par exemple, est supprimé en appliquant un filtre médian.

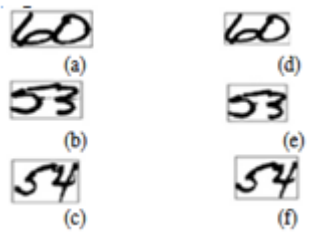


FIGURE 1.4 – Lissage des images, où (a),(b) et (c) sont les images originales (d), (e) et (f) sont les images après lissage.

1.7.4.2 Les opérations morphologiques

Diverses opérations morphologiques peuvent être conçues pour connecter des traits cassés, décomposer les traits connectés, lisser les contours, couper les points non désirés, affiner les caractères et extraire les limites...etc.
Les opérations morphologiques se décomposent en trois types : la dilatation (Heijmans et Ronse, 1990), l'érosion (Heijmans et Ronse, 1990) et l'ouverture et fermeture binaires (Haralick *et al.*, 1987).

1.7.4.3 La modélisation du bruit

La modélisation du bruit est définie pour les images détectées à distance. Les statistiques de bruit sont estimées en utilisant les moyennes et les variances de petits blocs d'image (4x4 ou 8x8). Étant donné que la plupart des images contiennent de nombreuses zones petites mais homogènes, un diagramme de dispersion de la variance vs ($moyenne^2$) révèle les caractéristiques du bruit (Lee et Hoppel, 1989).

1.7.5 Normalisation

La normalisation est un processus linéaire qui permet de ramener les images du caractère à des tailles standards. La différentielle pousse le principe de normalisation à un degré plus fin, en essayant de normaliser localement les différentes parties du mot, de manière à augmenter la ressemblance de la taille d'une image à une autre.
Après la normalisation de la taille, les images de tous les caractères se retrouvent définies dans une matrice de même taille, pour faciliter les traitements ultérieurs (Azizi *et al.*, 2009), (Hassin *et al.*, 2004), (Kessentini *et al.*, 2010).
Le processus de normalisation produira les régions qui ont les mêmes dimensions constantes (Kumar *et al.*, 2013). Pendant la normalisation, chaque caractère est redimensionné dans une taille qui a été fournie lors de l'initialisation du modèle. Toutes les autres opérations de classificateur seront effectuées sur les caractères redimensionnés (normalisés).

La normalisation des caractères permet de réduire la quantité de données utilisées dans la classification des caractères. L'autre objectif de la normalisation est de permettre au processus de classification de reconnaître des caractères de différentes tailles. Les systèmes de reconnaissance de caractères peuvent appliquer des normalisations de taille horizontale et verticale.
En appliquant la méthode basée sur la recherche de voisinage pour omettre des blobs de bruit isolés dans l'image de caractère pendant le calcul de la boîte de délimitation d'image. Wl et H1 indiquent la largeur et la hauteur du caractère. La largeur et la hauteur du caractère normalisé sont notées W2 et H2, et la taille du plan standard est notée L. Cette dernière est généralement considérée comme un carré et sa taille est typiquement de 32×32 ou 64×64.
Les proportions R1 et R2 du caractère original et normalisé sont :

$$R_1 = \frac{min(W_1,H_1)}{max(W_1,H_1)}$$
$$R_2 = \frac{min(W_2,H_2)}{max(W_2,H_2)}$$

1.7.6 Squelettisation

Le but de cette technique est de réduire l'épaisseur du tracé d'un mot à un pixel afin de simplifier l'image du caractère en une image à « ligne » plus facile à traiter, en la réduisant au tracé du caractère. Les algorithmes de squelettisation se basent sur des méthodes itératives. Le processus s'effectue sur une image binaire, par passage successive sur un pixel pour déterminer si ce dernier est essentiel dans le tracé et par conséquent le garder ou non (Muaz, 2010).
Le squelette doit préserver la forme, la connexité, la topologie et les extrémités du tracé, et ne doit pas introduire d'éléments parasites (Cheng et Hsu, 1985), (Cheriet *et al.*, 2007).

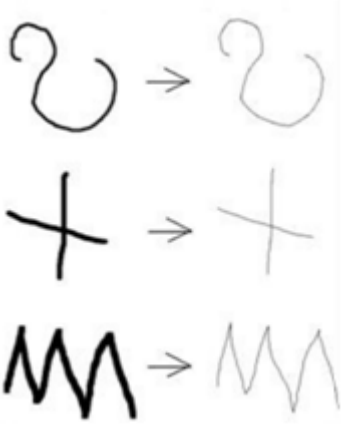


FIGURE 1.5 – Exemple de Squelettisation.

Dans le domaine de la reconnaissance optique des caractères, certains traitements sont nécessaires tandis que d'autres sont liés à la nature des modules choisis dans le système. Dans nos contributions, nous nous basons sur deux traitements primordiaux dans les systèmes proposés, à savoir : la binarisation, nécessaire pour avoir une image en niveau de gris et la détection et correction d'inclinaison, responsable de la correction des déformations produites dans la phase d'acquisition. Les deux traitements seront détaillés dans les chapitres suivants.

1.8 La segmentation

La segmentation est une étape importante dans beaucoup de systèmes d'OCR de script complexes (Shah *et al.*, 2021). La mise en œuvre de cette étape permet de diviser l'image en différentes imagettes de tailles moins importantes qui peuvent être des figures, des graphèmes, des mots…etc. Généralement, les unités générées à la suite du prétraitement ne sont pas les blocs de construction de base de ce script particulier. Il doit donc y avoir un algorithme pour séparer les plus petites formes possibles qui sont utilisées pour construire les unités complexes. Ces formes plus petites sont appelées segments.
Dans un premier temps, il est nécessaire de localiser les régions du document où les données ont été imprimées et de les distinguer des figures et des graphiques. Dans un deuxième temps, la segmentation est appliquée au texte extrait dont l'objectif est d'isoler les différentes composantes d'un bloc de texte en lignes, en mots, et en caractères, avant la phase de reconnaissance des caractères.

1.8.1 Segmentation de page

Le principe de cette phase consiste à extraire les différentes parties logiques à partir d'une image acquise, et ce par la séparation des blocs de texte et des blocs graphiques. Le résultat devrait être une image avec seulement du texte. Les techniques de segmentation des documents peuvent être classées en trois grandes catégories : les techniques descendantes, ascendantes et hybrides.
Les méthodes descendantes segmentent récursivement les grandes régions d'un document en sous-régions plus petites. La segmentation s'arrête lorsque certains critères sont satisfaits et les images obtenues à ce stade constituent les résultats finaux de la segmentation. L'algorithme RLSA (Run-Length Smearing Algorithm) est l'un des algorithmes descendants les plus utilisés. Il est appliqué sur les images binaires, en reliant entre les pixels noirs voisins qui se trouvent dans un certain seuil. Cette méthode est appliquée ligne par ligne et colonne par colonne, puis les deux résultats sont combinés dans une opération OU logique et finalement un seuil de lissage est utilisé pour produire le résultat final de la segmentation. A partir des résultats RLSA, des blocs noirs de lignes de texte et d'images sont produits (Wahl *et al.*, 1982).
D'autre part, les méthodes ascendantes commencent par regrouper les pixels d'intérêt et

les fusionner en blocs plus grands ou en composants connexes, tels que des caractères qui sont ensuite regroupés en mots, en lignes ou en blocs de texte. Un exemple d'algorithme ascendant est la méthode X-Y récursive, également connue sous le nom de « projection profile cuts ». Elle suppose que les documents sont présentés sous la forme d'un arbre de blocs rectangulaires imbriqués (Nagy *et al.*, 1985). Bien que les découpes X-Y récursives puissent décomposer une image de document en un ensemble de blocs rectangulaires, aucun détail n'a été donné sur la façon de définir les découpes.
Les méthodes hybrides combinent des stratégies descendantes et ascendantes. Un exemple de méthode hybride est l'approche de segmentation de Kruatrachue et Suthaphan qui consiste en deux étapes : une méthode d'extraction par blocs descendante suivie d'une méthode de détection et de segmentation multi-colonnes ascendante (Kruatrachue et Suthaphan, 2001). La segmentation est basée sur des blocs de colonnes extraits par un algorithme de suivi de bord modifié, qui utilise une fenêtre de 32 x 32 pixels pour qu'un paragraphe puisse être extrait à la place d'un caractère.
Dans la littérature, les enquêtes de (Mao *et al.*, 2003) et (Tang *et al.*, 1996) fournissent des explications détaillées sur l'analyse de documents et les algorithmes de représentation de mise en page.

1.8.2 Segmentation de texte

Le principe de la segmentation de texte consiste à l'extraction, à partir d'un bloc de texte, les lignes, ensuite les mots, les caractères et les pseudo-caractères à partir des lignes extraites. Dans ce cadre deux types d'approches ont été développées.

1.8.2.1 Types d'approches de segmentation

Les approches permettant la mise en œuvre de la segmentation varient entre approches globales et approches analytiques.
Approche globale
L'approche globale a une vision générale sur le mot. Elle se base sur le mot en entier comme étant une entité indivisible caractérisée par une quantité d'informations lui permettant d'absorber facilement les variations au niveau de l'écriture. Cependant, l'aspect généraliste dans cette approche la limite à des vocabulaires distincts et réduits. En effet, la discrimination entre deux mots proches est difficile et nécessite un nombre important d'échantillons dans la phase d'apprentissage. Dans le cas des vocabulaires de grandes tailles, cette approche est utilisée pour réduire la liste des mots candidats après classification. Pour les vocabulaires réduits et distincts, cette approche reste parfaitement envisageable.
Approche analytique
L'approche analytique permet de dépasser les limites de l'approche globale car elle est basée sur un découpage du mot. Elle consiste à segmenter le mot en lettres ou en parties inférieures aux lettres appelées graphèmes et à retrouver les lettres puis le mot par

combinaison de ces éléments. La segmentation, implicite ou discrète, est basée sur la sélection d'un ensemble de points appelés des points de segmentation. Ces points peuvent être :

- Des minima locaux du contour supérieur (Olivier *et al.*, 1996),(Sari *et al.*, 2002), (Ding et Liu, 2006).
- Des espaces entre les caractères ou bien les sous-mots (Motawa *et al.*, 1997), (Lorigo et Govindaraju, 2005), (Xiu *et al.*, 2006).
- Des points d'intersection les plus probables par une analyse des composantes dans le mot.

Le segment résultant de cette étape est appelé graphème, il est considéré comme l'entité de base pour un mot (Elbaati *et al.*, 2006), (Almuallim et Yamaguchi, 1987), (Romeo-Pakker et Ameur, 1993), (Wshah *et al.*, 2009).

Cette approche est utilisée dans le cas de vocabulaires de grande taille. Elle s'adapte au changement de vocabulaire et peut faire la discrimination des mots en se basant sur la reconnaissance des parties qui composent le mot. L'inconvénient principal de cette approche réside dans la difficulté de segmentation et les problèmes de sous-segmentation ou de sur-segmentation que cela peut impliquer.

1.8.2.2 Techniques de segmentation

Il existe de nombreuses techniques développées pour la segmentation de texte en lignes, caractères ou en pseudo-caractères. Parmi les techniques les plus simples, nous pouvons citer : l'analyse des composants connectés (Miled, 1998) et les histogrammes de projection horizontaux (El Ayachi *et al.*, 2011). Toutefois, dans des situations complexes où les caractères se chevauchent, se brisent ou un certain bruit est présent dans l'image, des techniques avancées de segmentation de caractères sont utilisées, à savoir la fenêtre glissante et la segmentation par détection du point de jonction (Kaur et Bathla, 2015). Parmi les techniques de segmentation, quelques-unes d'entre elles peuvent être spécifiques à un script et ne pas fonctionner pour d'autres scripts. Dans notre cas, le script est composé d'un texte non cursif, donc nous utilisons la technique de la segmentation par un histogramme vertical et horizontal. Cette technique sera détaillée plus tard dans ce rapport.

1.8.2.3 Problèmes de segmentation de texte

Les principaux problèmes de segmentation de texte peuvent être groupés en quatre catégories. Ces problèmes peuvent apparaître plus fréquemment dans la segmentation des caractères car les caractères peuvent être de taille et formes différentes surtout dans un document manuscrit.
Parmi ces problèmes, nous pouvons citer :

- Problèmes de chevauchement ;

— Extraction des caractères touchants et fragmentés ;
— Problème du caractère brisé ;
— Problème du caractère superposé ;
— Problème du caractère asymétrique ;
— Problème de bruit ;
— Problème de confusion du graphique avec le texte ;
— Problème de confusion du texte avec le graphique.

1.9 Extraction des caractéristiques

L'objectif de l'extraction des caractéristiques (encore appelées primitives) consiste à sélectionner les caractéristiques discriminantes des éléments issues de la phase de segmentation pour être utilisés dans la phase de reconnaissance. Son but est également de réduire le volume d'informations qui sera fourni au système. Cette étape est généralement admise comme une phase critique lors de la construction d'un système de reconnaissance dans le domaine de reconnaissance des formes (Deepa *et al.*, 2014). La façon la plus directe est de décrire la matrice de l'image réelle du caractère. Cependant, une autre approche consiste à extraire certains éléments qui caractérisent toujours les segments, mais cette approche peut omettre les attributs importants.
Dans la littérature, nous distinguons plusieurs classes de caractéristiques qui peuvent etre groupées en trois grands groupes : Caractéristiques statistiques, caractéristiques structurelles et transformations globales.

1.9.1 Caractéristiques statistiques

Dans ce type de caractéristique, l'image est représentée par des mesures statistiques. Il est possible d'utiliser, par exemple, la distribution des pixels dans différentes régions de l'image, ou bien des histogrammes (nombre de points noirs par colonne, par ligne, ou dans d'autres directions). Des méthodes plus complexes comme l'analyse en composantes principales peuvent également produire ce type de caractéristiques. De même, il est possible de calculer le nombre de pixels présentant une caractéristique particulière dans différentes régions de l'image.
Les caractéristiques statistiques décrivent une forme en un ensemble de mesures statistiques extraites à partir de cette forme (El-Hajj *et al.*, 2005). La raison principale de l'utilisation des caractéristique statistiques est de donner des informations locales sur le contenu de l'écriture. Il s'agit entre autres de :

— Le profil de projection des densités de pixels noir/blanc qui représente le nombre de pixels sur chaque ligne ou chaque colonne de l'image.
— L'histogramme directionnel permettant de compter le nombre de pixels sur une ligne dans une direction quelconque de l'image.

— L'histogramme des transitions qui permet de retenir le nombre des transitions 0-1 et 1-0 entre pixels.
— Les moyennes locales de pixels de l'image situées à l'intérieur d'un masque rectangulaire (Zoning).
— Les directions des contours dans une fenêtre locale.
— Les moments invariants sont des mesures statistiques de la distribution des pixels autour du centre de gravité du caractère. Ils ont été initialement appliqués à la reconnaissance de l'écriture latine et arabe, puis récemment ont été appliqués aux caractères tifinaghes (El Yachi *et al.*, 2010).

Avec ce type de caractéristiques, aucune interprétation directe n'est faite sur le contenu de l'écriture. Ainsi, ces caractéristiques se trouvent moins fortes que les primitives structurelles.

1.9.2 Caractéristiques structurelles

Les caractéristiques structurelles décrivent les propriétés topologiques et géométriques de l'écriture dans le but de capturer la structure ou la forme du mot (Chaker *et al.*, 2011). Nous cherchons par ces caractéristiques à décrire la composition physique du caractère et donc à détecter des éléments de base dans l'image.
Les caractéristiques structurelles sont extraites à partir du squelette ou du contour de la forme présentée dans l'image (Kapoor *et al.*, 2002), (Rath *et al.*, 2003). Les caractéristiques les plus couramment utilisées sont :

— les points d'extrémité ;
— les intersections entre les lignes et les boucles ;
— les directions principales du tracé ;
— les segments de droite et leurs attributs (position, orientation, ...) ;
— les positions relatives entre segments ;
— les arcs, les boucles et les concavités, les mesures de pentes et autres paramètres de courbures pour évaluer des orientations principales ;
— le nombre de points diacritiques et leur position par rapport à la ligne de base de l'écriture ;
— le nombre de lignes horizontales et verticales et diagonales ;
— la longueur normalisée de toutes les lignes ;
— autres paramètres tels que : la longueur et l'épaisseur des traits, le nombre de trous, les surfaces et les périmètres.

Comparé à d'autres techniques, l'analyse structurale génère des fonctions avec une tolérance élevée aux variations du style et au bruit. Toutefois, les caractéristiques sont, le plus souvent, modérément sensibles aux différentes variations que doivent subir les mots. Les caractéristiques structurelles ont la propriété de localité. C'est-à-dire, qu'elles sont attachées très spécifiquement aux différentes zones de l'image.
Les primitives structurelles constituent des caractéristiques très informatives et discriminantes. Elles permettent de prendre des décisions rapides dans la reconnaissance de

l'écriture avec une complexité de calcul modérée pendant l'extraction, en comparaison aux caractéristiques statistiques.

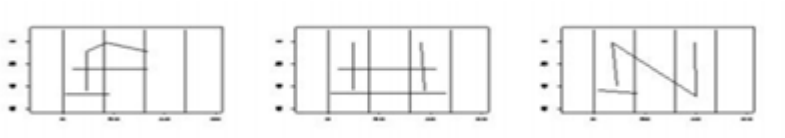


FIGURE 1.6 – L'analyse structurelle d'un caractère.

1.9.3 Transformations globales

Nous parlons de caractéristiques globales quand nous nous intéressons à représenter la forme générale d'un caractère. Ils sont donc calculés sur la globalité de l'image et sur des images relativement grandes sans chercher à distinguer les différentes zones. Parmi les méthodes les plus utilisées, nous pouvons citer : Les filtres de Gabor (Vautrot, 1996), la transformée de Hough (Wilson et Ritter, 2000), la transformée de Fourier (Zhang *et al.*, 2008), les moments (El Ayachi *et al.*, 2011) et les ondelettes (Bultheel, 2003).
Le choix d'un bon ensemble de caractéristiques est crucial dans tout processus de classification pour diverses raisons. L'ensemble des caractéristiques considéré doit inclure toutes les informations nécessaires pour discriminer les échantillons appartenant à différentes classes, les performances réalisables peuvent être insatisfaisantes, quelle que soit l'efficacité de l'algorithme d'apprentissage. D'autre part, la taille du jeu de caractéristiques utilisé pour décrire les échantillons détermine l'espace de recherche à explorer pendant la phase d'apprentissage. Par conséquent, des fonctionnalités non pertinentes et bruyantes agrandissent l'espace de recherche, ce qui augmente la complexité du processus. Enfin, le coût de calcul de la classification dépend du nombre des caractéristiques utilisées pour décrire les modèles.
Dans nos études, nous utilisons différentes catégories de caractéristiques que nous détaillons plus trad dans ce rapport.

1.10 Classification

La classification est le processus d'identification de chaque caractère et de lui attribuer la classe correcte, en utilisant les éléments extraits lors de la phase d'extraction des caractéristiques. Les méthodes développées pour la classification peuvent être groupées selon la nature de l'apprentissage : supervisé, non supervisé et semi supervisé.

1.10.1 Apprentissage supervisé

La majorité des apprentissages automatiques pratiques (machines Learning) utilisent l'apprentissage supervisé. C'est le cas où nous disposons des variables d'entrée (x) et des variables de sortie (Y) et nous utilisons un algorithme pour apprendre la fonction de mappage entre l'entrée et la sortie.

$$Y = f(X)$$

Le but est d'approximer la fonction de mappage afin qu'elle soit capable de prédire les variables de sortie (Y) pour chaque nouvelle donnée d'entrée (x). L'algorithme fait des prédictions itératives sur les données d'apprentissage qui seront corrigées par le superviseur. L'apprentissage s'arrête lorsque l'algorithme atteint un niveau de performance acceptable.
Les problèmes d'apprentissage supervisé peuvent être regroupés en types.

- un problème dit de classification, lorsque la variable de sortie est une catégorie, comme « rouge » ou « bleu », ou « maladie » et « pas de maladie » ;
- un problème dit de régression, lorsque la variable de sortie est une valeur réelle, telle que « dollars » ou « poids ».

Les types de problèmes les plus courants, qui s'ajoutent à la classification et à la régression, incluent, respectivement, la prédiction de séries de recommandations et de séries temporelles.
Quelques exemples populaires d'algorithmes d'apprentissage automatique supervisés sont : la régression linéaire pour les problèmes de régression, le random forest pour les problèmes de classification et de régression et le support vector machines pour les problèmes de classification.

1.10.2 Apprentissage non supervisé

Dans le cas de l'apprentissage non supervisé, seules les données d'entrée (X) sont mises en jeu, et aucune variable de sortie correspondante n'est prise en compte. Le but de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'en y apprendre plus.
Ce type d'apprentissage est appelé l'apprentissage non supervisé car, contrairement à l'apprentissage supervisé, il n'y a pas de bonnes réponses et il n'y a pas de superviseur. Les algorithmes sont laissés à leurs propres moyens pour découvrir et présenter la structure la plus intéressante dans les données.
Les problèmes d'apprentissage non supervisés peuvent être regroupés en problèmes de Clustering(regroupement) et d'association :

- un problème est dit de clustering, si nous souhaitons découvrir les regroupements inhérents aux données, tel dans l'exemple du regroupement des clients par comportement d'achat.

— un problème est dit d'apprentissage de règle d'association si nous souhaitons découvrir des règles qui décrivent les liaisons entre les données, par exemple, les personnes qui achètent X ont également tendance à acheter Y.

Quelques exemples populaires d'algorithmes d'apprentissage non supervisés sont :

— K-means pour les problèmes de clustering ;
— Algorithme à priori pour les problèmes d'apprentissage des règles d'association.

1.10.3 Apprentissage semi-supervisé

On appelle problèmes d'apprentissage semi-supervisés les problèmes où les données d'entrée (X) sont composées d'une grande quantité de données et seulement certaines des données sont étiquetées (Y). Ces problèmes se situent entre l'apprentissage supervisé et non supervisé.
Il existe aujourd'hui, dans le domaine de la reconnaissance de formes, un grand nombre de classifieurs et différentes approches de classification qui sont plus ou moins bien adaptés à la reconnaissance de l'écriture (Islam *et al.*, 2017). Cependant, malgré les années de recherche, aucune méthode de classification n'a pu mettre en évidence sa supériorité incontestable par rapport à d'autres.
Dans la littérature (Jain *et al.*, 2000) (Liu et Fujisawa, 2008), les techniques de reconnaissance et de classification de textes sont regroupées en quatre catégories principales : le pattern matching (ou appariement de formes), méthodes statistiques, méthodes neuronales et méthodes structurelles ou syntaxiques. Les méthodes stochastiques peuvent aussi être ajoutées à cette catégorisation et peuvent être considérées comme une sous-famille des méthodes statistiques.

1.10.3.1 L'appariement de formes

L'appariement de formes, connu sous le nom anglais le Pattern Matching, est l'une des méthodes de classification les plus courantes (Eikvil, 1993). Elle couvre un groupe de techniques représentant une particularité par rapport aux autres techniques, dans le sens qu'aucune caractéristique n'est réellement extraite. Ces techniques sont basées sur des mesures de similarité où la distance entre le vecteur caractéristique, décrivant le caractère extrait, et la description de chaque classe est calculée.

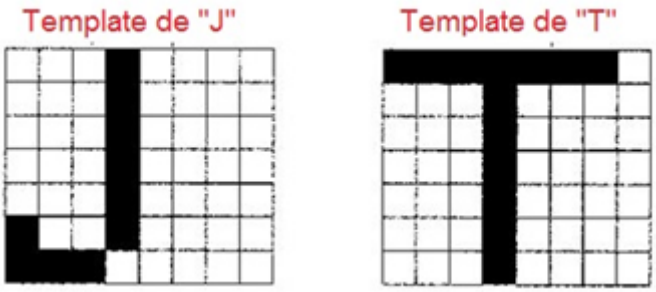


FIGURE 1.7 – Template de la lettre « J » et « T ».

On peut donc distinguer trois types de mesures couramment utilisées (Islam *et al.*, 2017) :
- les mesures de ressemblance du type inter-corrélation ou inter-corrélation normées ;
- les mesures de dissemblance telles que les distance de Hamming, Chebychev ou euclidienne ;
- les mesures de similarité du type Jaccard ou Yule.

Différentes mesures peuvent être utilisées, mais la distance euclidienne reste la mesure la plus connue dans la littérature.
Cette approche a été l'une des premières approches proposées pour la reconnaissance. Elle est simple et facile à mettre en œuvre dans le matériel. Elle a été utilisée dans de nombreuses machines OCR commerciales. Cependant, elle a eu un succès limité à l'adaptation à l'écriture du fait de sa grande variabilité qui implique un grand nombre de représentants pour chaque classe surtout pour le cas de l'écriture manuscrite (Es Saady, 2012).

1.10.3.2 Les méthodes statistiques

L'approche statistique est une approche qui repose essentiellement sur des fondements mathématiques (probabilité et statistique). Dans la classification statistique, une approche probabiliste de la reconnaissance est appliquée. L'idée est d'utiliser un système de classification optimal dans le sens où, en moyenne, son utilisation donne la plus faible probabilité de faire des erreurs de classification. (Bouslimi, 2006).
Dans les classificateurs statistiques, les formes sont vues comme des points dans un espace à n dimensions, n étant le nombre des caractéristiques de l'espace. La représentation du vecteur des caractéristiques x est comme suit : $x = x_1x_2x_3....x_n$, où les x_i représentent les n mesures caractéristiques adéquates et significatives. Toute forme x appartenant à une classe u_i est liée aléatoirement à une distribution de probabilité de la classe : $p(u/x_i)$.
Les méthodes statistiques sont connues par leur simplicité et leur faible coût en termes de temps de calcul surtout pour l'approche paramétrique et moyennement sensible au-bruit (Bishop, 2006).
Les méthodes statiques peuvent être classées en deux grandes familles : les méthodes

non-paramétriques et les méthodes paramétriques.
Pour les **méthodes non-paramétriques** (appelées aussi approches modélisantes), le but est de définir les frontières des classes dans l'espace de représentation, de façon à pouvoir classer le point inconnu par une série de tests simples. Parmi ces méthodes, nous pouvons trouver : la fenêtres de Parzen (Duda *et al.*, 2012), mixture de gaussiennes (Duda *et al.*, 2012), K-plus proches voisins (Kuncheva, 2004) (Amin, 1980).
Dans les **méthodes paramétriques** (appelés aussi bayésiennes ou discriminantes), un modèle de la distribution de chaque classe (en général gaussien) est mis en place dans le but de chercher la classe dont le point a la plus grande probabilité d'appartenir. Comme exemple de ces méthodes, nous pouvons citer l'approche Bayésien (Kopparapu et Desai, 2001), machines à vecteurs de support (Support Vector Machine, SVM) (Abe, 2005) et la fonction discriminante linéaire (Linear Discriminant Fonction, LDF) (Kawatani, 1993). L'une des approches les plus connues dans la littérature et qui peut être considérée comme sous famille de l'approche statistique est l'approche des modèles de Markov cachés (Hidden Markov Modeling, HMM) (Fink, 2014).

1.10.3.3 Les méthodes neuronales

Les méthodes neuronales font référence, comme leur nom l'indique, aux réseaux de neurones. Ce type de méthodes peut aussi être considéré comme une famille des méthodes statistiques. Les réseaux de neurones artificiels (Artificial Neural Network ,ANN), appelés aussi réseaux connexionnistes, sont composés d'éléments simples (ou neurones) connectés entre eux. Ces éléments ont été fortement inspirés par le système nerveux biologique.
Le fonctionnement d'un réseau de neurones est fortement influencé par les connexions des éléments entre eux. Les valeurs de ces connexions (ou poids) sont ajustées durant une phase d'entraînement. Cette phase dite d'apprentissage permet aux réseaux de neurones de réaliser des tâches complexes dans différents types d'application (classification, identification, reconnaissance de caractères, de la voix, vision, système de contrôle). Les réseaux de neurones peuvent souvent apporter une solution simple et rapide à des problèmes très complexes et difficiles à résoudre.

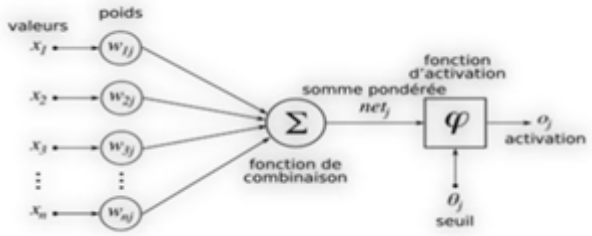


FIGURE 1.8 – Schéma général d'un réseau de neurones.

En OCR, les primitives extraites sur une image d'un caractère (ou de l'entité choisie) constituent les entrées du réseau. La sortie activée du réseau correspond au caractère reconnu. Le choix de l'architecture du réseau est un compromis entre la complexité des calculs et le taux de reconnaissance.
Par ailleurs, le point fort des réseaux de neurones réside dans leur capacité de générer une région de décision de forme quelconque, requise par un algorithme de classification, au prix de l'intégration de couches de cellules supplémentaires dans le réseau.
Parmi les divers classifieurs à base d'ANN, le réseau de neurones avec rétro-propagation, en anglais FeedForward Network, qui est un classificateur principalement utilisé pour les problèmes de reconnaissance manuscrits(Lakshana et Amudha, 2021).
Les réseaux de neurones sont utilisés non seulement pour résoudre les problèmes de reconnaissance de formes, mais aussi pour effectuer les tâches de la catégorisation, l'approximation de la fonction, la prévision, l'optimisation, la mémoire associative et le contrôle des E / S de divers systèmes. Le réseau d'anticipation le plus couramment utilisé pour le problème de reconnaissance de modèle est formé avec un algorithme de rétro-propagation d'erreur, qui est basé sur des règles d'apprentissage de correction d'erreur.

Algorithme de rétro-propagation d'erreur :

L'algorithme de rétro-propagation a été développé par Werbos en 1974 et il a été redécouvert par Parker et LeCun en 1975.Ces développements ont été rapportés en 1986 (Jain *et al.*, 1996). Quand le réseau de feed-back est formé avec l'algorithme de rétro-propagation d'erreur, le réseau se compose de deux genres de signaux :

1. Le signal direct (signal d'entrée), qui provient du neurone d'entrée de la couche d'entrée et qui est transmis dans le sens direct à travers le réseau, apparaît en sortie sur le neurone de sortie de la couche de sortie. Il est utilisé pour mapper les données d'entrée vers la sortie désirée.
2. Le signal en arrière (signal d'erreur), qui provient du neurone de sortie de la couche de sortie, et qui est propagé vers l'arrière à travers le réseau et utilisé pour mettre à jour les poids du réseau.

En fait, ce réseau n'a pas de rétroaction en retour, mais les erreurs sont propagées en retour lors de l'apprentissage en réseau et les poids sont ajustés dynamiquement. Comme déjà mentionné, un algorithme de rétro-propagation d'erreur est basé sur une règle d'apprentissage de correction d'erreur. L'objectif est de rapprocher la sortie réelle v_k de la sortie souhaitée d_K équivalent à la minimisation de la fonction du coût d'erreur quadratique. La règle delta est utilisée pour mettre à jour les poids du neurone de la couche de sortie. Le réseau d'alimentation multicouche avec algorithme de rétro-propagation consiste en nombre de couches cachées en plus de la couche de sortie.
La règle delta (1) est étendue pour changer les poids de la couche cachée et donc cette règle est également appelée règle delta généralisée. L'ajustement du poids entre les neurones j et k est proportionnel au gradient négatif de l'erreur, généré au niveau du k_i^{me} neurone, par rapport au poids, c'est-à-dire,

$$\Delta W_{jk}(t) = -\eta \frac{dE(t)}{dW_{jK}} \qquad (1)$$

Règles d'apprentissage de correction d'erreur :
Dans le cas d'apprentissage supervisé, le réseau est entraîné avec une sortie exacte pour chaque modèle d'entrée. Cet apprentissage est réalisé dans diverses itérations. Dans une itération donnée, la sortie générée par un réseau n'est pas égale à la sortie exacte / souhaitée. Si v_k est la sortie réelle générée et d_K est la sortie désirée au k_i^{me} neurone dans la couche de sortie dans l'itération, alors l'erreur de sortie est $e_K(t) = d_K(t) - v_K(t)$
Le signal d'erreur $e_K(t)$ est utilisé pour ajuster tous les poids du k_ime neurone. L'ajustement des poids est effectué pour rapprocher la sortie $v_K(t)$ du k_i^{me} neurone de la sortie désirée équivalente à la minimisation de la fonction du coût d'erreur au carré exprimée en termes de signal d'erreur comme suit :

$$E(t) = \frac{1}{2}e_K^2(t)$$

Cet ajustement des poids est effectué par étapes en nombre d'itérations et lorsque cette erreur est minimale, le processus d'apprentissage est terminé. La règle d'apprentissage utilisée pour minimiser l'erreur de cette manière est appelée règle d'apprentissage de correction d'erreur.
Considérons un vecteur d'entrée $(u_1, u_2, u_3, ...u_i, ...u_n)$, un vecteur de sortie actuelle $(v_1, v_2, v_3, ...v_i, ...v_n)$ et un vecteur de sortie $(s_1, s_2, s_3, ...s_i, ...s_n)$. Les poids dus à ce vecteur d'entrée au $kime$ neurone sont $w1k, w2k, w3k, ..., wik, ...wnk$, alors un petit changement dans le poids de la connexion entre le i^{me} noeud d'entrée au K^{me} neurone dans la t^{me} itération est donnée par la règle du delta de Widrow et Hoff (Widrow et Hoff, 1960), appelé la règle de Widrow-Hoff.

$$\Delta W_{ik}(t) = \eta(s_K(t) - v_K(t))u_i(t) \qquad (2)$$

Où η est le paramètre de taux d'apprentissage qui est une valeur constante positive. Elle est indiquée comme étant un ajustement fait sur les poids d'un neurone qui est proportionnel au produit du signal d'erreur et du signal d'entrée à ce neurone (Das et Mohanty, 2020).
A coté des classifieurs à rétro-propagation, il existe plusieurs d'autres types des réseaux de neurones artificielles à savoir le MLP et le LSTM qui seront décrites plus tard dans le chapitre 4.

1.10.3.4 Les méthodes structurelles ou syntaxiques et de tests

Les méthodes structurelles reposent sur la représentation hiérarchique et la structure physique des caractères composée d'un ensemble de sous-formes (patterns), où chaque sous-forme est constituée de patterns plus petits (primitives). Elles cherchent à trouver ces éléments, simples ou primitifs, et à décrire leurs relations. Les primitives sont de type topologique telles que : une boucle, un segment de droite, une occlusion ou un arc.... Une relation peut être la position relative d'une primitive par rapport à une autre.

La mesure de similarité basée sur la relation entre les composants structurels peut être formulée en utilisant le concept de la grammaire. L'idée est que chaque classe a sa propre grammaire qui définit la composition de ses caractères. La grammaire peut être représentée comme une chaîne de caractères ou une arborescence et les composantes structurelles extraites du caractère inconnu sont comparées à la grammaire de chaque classe.
Parmi les méthodes structurelles nous pouvons citer :

Les méthodes de tests :

Elles consistent à appliquer des tests sur chaque caractère concernant la présence ou l'absence des éléments simples ou des primitives afin de déterminer sa classe. Le processus le plus habituel divise à chaque test l'ensemble de choix en deux suivant la présence ou l'absence d'une primitive jusqu'à n'obtenir qu'une seule forme correspondante au caractère entré.
Ce choix dichotomique est très rapide et très simple à mettre en œuvre, mais il est très sensible aux variations du tracé.

La comparaison de chaînes :

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

Comparaison des graphes :

Le but dans cette méthode est d'abord de construire un graphe ou l'ensemble des nœuds contient les primitives et liens entre ces derniers. La reconnaissance consiste alors de comparer le graphe correspondant au caractère avec les graphes des caractères de références qui ont été construits pendant la phase d'apprentissage du modèle. Cette méthode a été utilisée par (Jain *et al.*, 1996) ;(Lebourgeois, 1992) ;(Zahour *et al.*, 2004).

Méthodes syntaxiques :

Ce type de méthodes est basé sur la grammaire formelle, dont l'idée est de décomposer une forme pour avoir une séquence de primitives sous forme d'une succession de mots ou de phrases. La représentation du caractère est alors une phrase dont le vocabulaire utilisé est l'ensemble des primitives constituant le caractère. La détermination de l'appartenance d'un caractère à une classe consiste à vérifier si sa phrase correspondante peut être générée par la grammaire de cette classe. Le problème de l'absence d'algorithmes efficaces pour l'inférence grammaticale directe peut être considéré comme limitation principale de ce type de méthodes.
L'avantage des méthodes structurelles par rapport à celles statistiques est que les premières utilisent un nombre réduit de prototypes pour représenter une classe alors que les deuxièmes demandent un nombre important d'échantillons. Ainsi, le coût de mise en correspondance entre le caractère et les prototypes est réduit. D'autre part, les méthodes structurelles permettent la représentation des prototypes ayant une forme particulière et par conséquent la variabilité au sein d'une même classe est assurée. Malgré les avantages précédemment cités, l'approche structurelle soufre de la sensibilité aux problèmes de la segmentation et au bruit.

1.11 Post-traitement

Le post-traitement est effectué quand le processus de reconnaissance aboutit à la génération d'une liste de lettres ou de mots possibles, éventuellement classés par ordre décroissant de vraisemblance. Le but est de faire une correction des résultats de la classification en vue de valider l'opération de numérisation. Cette opération peut se faire soit automatiquement par l'utilisation de dictionnaires et de méthodes de correction linguistiques en faisant intervenir des contraintes de niveaux successifs (lexical, syntaxique ou sémantique), soit manuellement à travers des interfaces dédiées.
Une fois le caractère classé, différentes approches peuvent être utilisées pour améliorer la précision des résultats des systèmes OCR. L'une des approches consiste à utiliser plus d'un classifieur pour faire la reconnaissance de l'image. Les classifieurs peuvent être utilisés en cascade, en mode parallèle ou hiérarchique. Les résultats des classifieurs peuvent ensuite être combinés en utilisant différentes approches. Une autre approche est basée sur une analyse contextuelle qui peut être effectuée au niveau du résultat. Le contexte géométrique et documentaire de l'image peut aider à réduire les risques d'erreurs. Le traitement lexical basé sur les modèles de Markov et le dictionnaire peut également aider à améliorer les résultats de l'OCR (Ciresan *et al.*, 2011). Nous pouvons citer deux types de post-traitement : le regroupement ainsi que la détection et la correction d'erreurs

1.11.1 Regroupement

Le résultat de la reconnaissance des caractères sur un document est un ensemble de caractères isolés. Cependant, ces caractères en eux-mêmes ne contiennent pas assez d'informations. Toutefois, nous souhaitons associer les différents caractères qui appartiennent à la même chaîne pour construire des mots et phrases. Le processus d'association de caractères en chaînes est appelé groupement. Le regroupement est basé sur l'emplacement des caractères dans le document. Les caractères qui se trouvent assez proches sont regroupés. Pour les fontes à pas fixe, le processus de regroupement est assez facile car la position de chaque caractère est connue. Pour les caractères manuscrits, la distance est variable. Toutefois, la distance entre les mots est généralement beaucoup plus grande que la distance entre les caractères, et le regroupement est donc reste possible. Les vrais problèmes se produisent pour des caractères manuscrits ou lorsque le texte est incliné.

1.11.2 Détection et correction d'erreurs

Jusqu'à la phase de regroupement, chaque caractère a été traité séparément et le contexte dans lequel chaque caractère apparaît n'a en général pas été exploité. Cependant, pour les problèmes de reconnaissance optique du texte, un système qui reconnait uniquement les caractères ne sera pas suffisant. Même les meilleurs systèmes de recon-

naissance ne donnent pas une identification correcte à 100%, de tous les caractères, mais certaines de ces erreurs peuvent être détectées ou même corrigées par l'utilisation du contexte. Il existe deux principales approches : La première utilise la possibilité qu'une séquence de caractères apparaisse en même temps. Ceci peut être réalisé par l'utilisation des règles définissant la syntaxe du mot. En outre, pour différentes langues, la probabilité que deux ou de plusieurs caractères apparaissent ensemble dans une séquence peut être calculée et utilisée pour détecter des erreurs.
La deuxième méthode, la moins complexe pour consolider les données de contexte, est l'utilisation d'un dictionnaire pour modifier les erreurs mineures des frameworks OCR (Hamad et Kaya, 2016). Il s'est avéré être la méthode la plus efficace pour la détection et correction d'erreurs. Étant donné un mot, dans lequel une erreur peut être présente, le mot est recherché dans le dictionnaire. Si le mot n'est pas dans le dictionnaire, une erreur a été détectée, et peut être corrigée en changeant le mot par le mot le plus proche. Les probabilités obtenues à partir de la classification, peuvent aider à identifier le caractère qui a été mal classé. Si le mot est présent dans le dictionnaire, cela ne prouve pas, malheureusement, qu'aucune erreur n'est produite. Une erreur peut transformer un mot correct à un autre, et ces erreurs ne sont pas détectables par cette procédure. L'inconvénient des méthodes de dictionnaire, c'est que les recherches et les comparaisons implicites prennent un temps considérable.

1.12 Conclusion

Ce chapitre présente une vue globale sur la reconnaissance optique des caractères en tant que technique connue pour la conversion des documents en forme d'image en une forme exploitable. Nous avons décrit les différents aspects de l'OCR, l'intérêt derrière cette technique et les problèmes liés à cette dernière. Dans ce chapitre, nous avons aussi dressé les différentes étapes constituant un système de reconnaissance optique des caractères à savoir : l'acquisition, le prétraitement, la segmentation, l'extraction de caractéristiques, la classification et enfin le post-traitement. Le chapitre introduit aussi les domaines ainsi que les applications qui peuvent intégrer les systèmes OCR.
D'autre part, ce chapitre présente un état de l'art des approches et des méthodes développées au profil des phases d'un système OCR.
Dans le chapitre suivant, nous nous focaliserons sur l'introduction de la langue amazighe qui représente notre langue d'étude.

Chapitre

2 La langue amazighe

Sommaire

2.1 Introduction

La langue amazighe, connue aussi sous le nom du berbère ou Tamazight (ⵜⴰⵎⴰⵣⵉⵖⵜ en tifinaghe) est une langue afro-asiatique ou chamito-sémitique (Rowan, 2006), dérivée du berbère ancien. Elle est l'une des plus anciennes langues de l'humanité. Actuellement,

elle couvre une très grande superficie en Afrique du Nord. Elle est présente dans une dizaine de pays et constitue la langue des populations autochtones de l'Afrique du Nord : Maroc, Algérie, Tunisie, Libye, et l'Oasis Siwa de l'Egypte. Également, elle est parlée par les populations de certaines régions du Niger, du Mali et du Burkina Faso, ainsi que par les communautés amazighes immigrées partout dans le monde (cf. Figure 2.1). Le nombre des locuteurs est estimé à 45 millions (Imane, 2016). Cependant, le Maroc et l'Algérie sont, de loin, les deux pays qui comptent, respectivement, les populations amazigho-phones les plus importantes 40% [1] et 27,4% [2]. L'amazighe est une langue orale qui possède une histoire écrite plus que millénaire. Elle est riche d'une tradition orale qui a su intégrer les médias modernes. De plus, la renaissance volontariste de l'alphabet traditionnel, le Tifinaghe, a permis de suppléer à la mémoire collective, de traduire les œuvres majeures du patrimoine mondial et de développer une littérature amazighe qui répond à une forte demande et qui a été transmise de génération en génération pendant des milliers d'années (environs 5000 ans). La figure ci-dessous illustre les différentes zones géographiques amazighes [3].

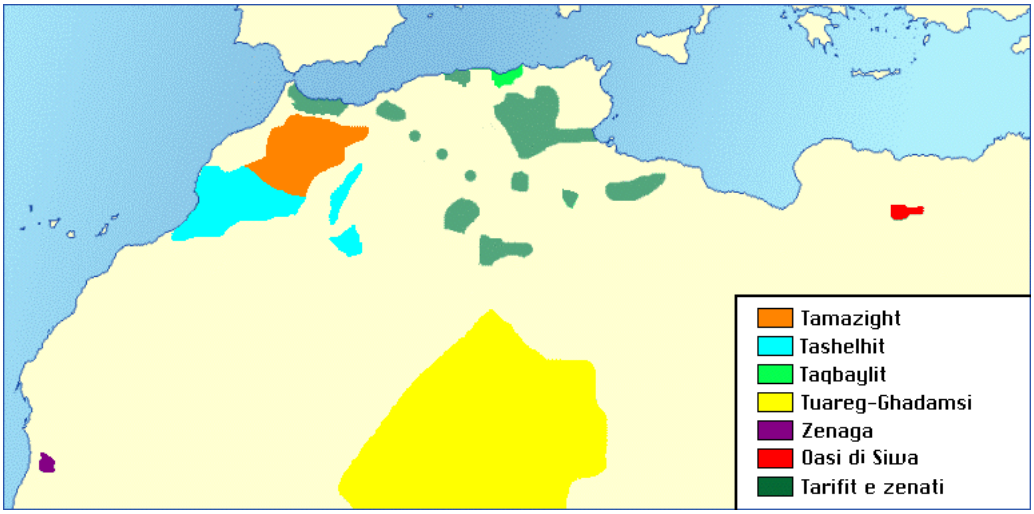
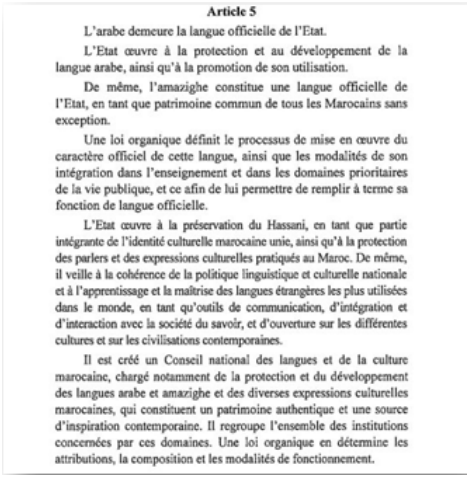


FIGURE 2.1 – Zones géographiques amazighes.

Au Maroc, 50% de la population parle l'amazighe (Boukous, 1995). Cependant, malgré cette importante masse qui utilisent la langue amazighe dans toutes leurs communications quotidiennes, l'usage de cette langue n'a pas dépassé, officiellement, le niveau oral qu'après la création de l'Institut Royal de la Culture Amazighe (IRCAM). L'objectif de cet institut est, entre autres, la promotion de la langue et la culture amazighes. L'IRCAM veille à la standardisation de l'amazighe au niveau national pour aménager les variantes de façon à uniformiser les structures morphologiques et grammaticales et

1. http ://www.axl.cefan.ulaval.ca/afrique/maroc-1demo.htm (visité le 15 Séptembre 2021).
2. http ://www.axl.cefan.ulaval.ca/afrique/algerie-1demo.htm (visité le 15 Séptembre 2021).
3. https ://fr.wikipedia.org/wiki/Fichier :Berber-map-ITA.png (visité le 15 Séptembre 2021).

à exploiter la variation pour l'enrichissement de la langue. Ce processus de standardisation a abouti à l'élaboration des lexiques, à l'homogénéisation de l'orthographe et à l'élaboration des règles de grammaire (Boukhris *et al.*, 2008). Par ailleurs, les travaux effectués sur cette langue ont débouché sur le commencement de l'enseignement de l'amazighe dans plusieurs écoles primaires marocaines et son intégration au niveau des établissements universitaires, à travers l'ouverture de filières d'études amazighes et des Masters spécialisés. Au niveau des médias, une chaîne en langue amazighe a été lancée en 2010, par la société Nationale de Radiodiffusion et de Télévision (SNRT). Les journaux en amazighe, tel que le « Monde amazighe », « Agraw Amazigh », « Twiza »..., apparaissent de plus en plus. La langue amazighe pose de nombreux défis en matière du traitement automatique du langage naturel. Le système d'écriture, la morphologie basée sur le processus unique de formation des mots par des racines et des motifs, et le manque de corpus linguistique rendent les approches informatiques de la langue amazighe difficiles. L'intérêt envers cette langue a été exprimé par des grandes organisations telles que Facebook qui a adopté la langue amazighe, transcrite par son système d'écriture Tifinaghe, comme langue d'utilisation. Cette initiative vient après l'introduction de l'amazighe dans la compagnie Microsoft qu'avait adoptée la langue amazighe pour Windows 8, et par Apple dans leurs systèmes d'exploitation IOS 9.
En juillet 2011, la langue amazighe est devenue une langue offcielle du pays à côté de l'arabe, grâce à la nouvelle constitution dans laquelle il est stipulé dans son article 5 (figure ci-dessous) la création d'un conseil national des langues et de la culture marocaine. La mission principale de ce conseil est le développement des langues arabe et amazighe, ainsi que les diverses expressions culturelles marocaines.



Article 5

L'arabe demeure la langue officielle de l'Etat.

L'Etat œuvre à la protection et au développement de la langue arabe, ainsi qu'à la promotion de son utilisation.

De même, l'amazighe constitue une langue officielle de l'Etat, en tant que patrimoine commun de tous les Marocains sans exception.

Une loi organique définit le processus de mise en œuvre du caractère officiel de cette langue, ainsi que les modalités de son intégration dans l'enseignement et dans les domaines prioritaires de la vie publique, et ce afin de lui permettre de remplir à terme sa fonction de langue officielle.

L'Etat œuvre à la préservation du Hassani, en tant que partie intégrante de l'identité culturelle marocaine unie, ainsi qu'à la protection des parlers et des expressions culturelles pratiqués au Maroc. De même, il veille à la cohérence de la politique linguistique et culturelle nationale et à l'apprentissage et la maîtrise des langues étrangères les plus utilisées dans le monde, en tant qu'outils de communication, d'intégration et d'interaction avec la société du savoir, et d'ouverture sur les différentes cultures et sur les civilisations contemporaines.

Il est créé un Conseil national des langues et de la culture marocaine, chargé notamment de la protection et du développement des langues arabe et amazighe et des diverses expressions culturelles marocaines, qui constituent un patrimoine authentique et une source d'inspiration contemporaine. Il regroupe l'ensemble des institutions concernées par ces domaines. Une loi organique en détermine les attributions, la composition et les modalités de fonctionnement.

FIGURE 2.2 – Article 5 de la nouvelle Constitution du Maroc 2011.

Dans ce contexte, et pour les raisons citées en dessus et bien d'autres, vient l'intérêt de contribuer à la promotion de cette langue et participer au développement du patrimoine marocain. Dans ce chapitre, nous explorons la langue amazighe commençant par

une présentation du processus de son informatisation, ses spécificités et ses différentes transcriptions. Nous donnerons un aperçu des critères de création d'un corpus pour l'OCR. Puis, nous présenterons notre première contribution, qui consiste en la construction d'un corpus pour la langue amazighe transcrite en latin. Ensuite, nous terminerons par une conclusion.

2.2 Informatisation de la langue amazighe

Du mot informatisation, le Grand Robert de la Langue Française donne la définition : *« Introduction dans une activité des méthodes informatiques »*. Idéalement, informatiser une langue c'est donc mettre à la disposition de l'utilisateur humain tous les moyens dont il a besoin dans sa langue, qu'elle soit écrite ou non.
L'informatisation des langues représente une étape importante et primordiale dans le processus de leur développement. Cependant, cette opération est délicate et coûteuse en termes de temps, notamment dans le cas des langues peu dotées. Elle consiste à affecter à la langue, toute sorte d'outils et de ressources linguistiques qui garantissent son intégration dans le domaine des technologies de l'information et de la communication, à savoir : dialogue avec la machine, outils pour écrire ou lire un texte (« en local »), envoyer un courrier électronique (« en réseau »), traduction informatisée dans une autre langue (Boukous, 2013), des dictionnaires, des corpus, ... ainsi qu'un système de reconnaissance optique des caractères dédié.
Depuis la création de l'IRCAM, et grâce à plusieurs initiatives de recherches scientifiques par certains académiciens marocains, la langue amazighe s'intègre de plus en plus dans le domaine des technologies de l'information et de la communication. En effet, le processus suivi par l'IRCAM dans le cadre de cette intégration a passé par plusieurs étapes, à savoir le codage spécifié par l'ASCII étendu, suivi de la création des polices de caractères tifinaghe. Ensuite, le codage propre dans le standard Unicode et l'élaboration des normes appropriées concernant la disposition du clavier amazighe, ainsi que le développement des applications du TALN (Zenkouar, 2004), (Ataa Allah et Boulaknadel, 2014). En ce qui concerne l'alphabet et pour des raisons historiques et culturelles, Tifinaghe est devenu le système graphique officiel pour l'écriture en amazighe.

2.3 Spécificités de la langue amazighe

Malgré sa position de langue officielle au Maroc, la langue amazighe manque encore d'études du point de vue informatique. Cette situation est due à plusieurs facteurs. D'une part, les travaux s'intéressant à cette langue n'ont commencé que récemment, et d'autre part, la complexité et la difficulté du traitement de cette langue. Ainsi, nous pouvons décrire trois difficultés principales qui doivent être prises en compte. Ces difficultés re-

lèvent de la multiplicité des systèmes d'écriture, de la variation de l'orthographe et de la complexité de la morphologie.

2.3.1 Système d'écriture

L'amazighe fait partie des langues ayant utilisé différentes graphies pour sa transcription, y compris la graphie latine, la graphie arabe et le tifinaghe. Par conséquent, un chercheur intéressé par l'étude de cette langue est confronté à la non-homogénéité du système de transcription des documents. Une description sur les transcriptions utilisées pour la langue amazighe est détaillée dans la section 2.4.

2.3.2 Orthographe

L'amazighe est restée, pendant des siècles, une langue essentiellement à prédominance orale. Par conséquent, les textes amazighes ne respectaient pas une convention d'écriture standard. Elle avait plusieurs orthographes possibles, qui varient non seulement en fonction de la variation dialectale ([tfucht] [tafukt] (soleil)), mais aussi à cause du système de transcription adopté ([tafuct] [tafukt]). La segmentation des mots n'était pas toujours stable ([tadartino] [tadart ino] (ma maison)).

2.3.3 Morphologie

La morphologie de l'amazighe est très riche mais au même temps très complexe. Ci-dessous, nous décrivons trois critères de difficultés principales de la morphologie amazighe.

2.3.3.1 Ambiguté du discours

L'ambiguïté est l'un des défis dans plusieurs études s'intéressant à la langue amazighe. La même forme de surface peut avoir des annotations différentes selon son utilisation dans la phrase exemple (Ataa Allah *et al.*, 2014) :

ⵜⵍⵍⵉ [illi] (ma fille),
ⵓⵔ ⵜⵍⵍⵉ [ur illi] (il n'existe pas).

2.3.3.2 Procession fléchissant et dérivationnelle

La langue amazighe est fortement fléchie et présente également une dérivation très poussée. Le premier processus présente plus de difficultés, en particulier pour l'inflexion nominale. À titre d'exemple pour le pluriel, il existe quatre formes différentes reposant

principalement sur des concaténations de préfixes et de suffixes, mais il n'existe pas de règles spécifiques qui déterminent l'utilisation de l'une ou l'autre forme pour fléchir un nom. De plus, la forme de base elle-même peut être modifiée selon différents paradigmes tels que celui de la dérivation, où en cas de présence d'une lettre géminée dans la forme de base, celle-ci sera modifiée sous la forme de dérivation (ⴽⴽⵙⵎ [qqim] "fait s'assoir" => ⵙⵉⵎ [s im] "assis").

2.3.3.3 Contraction

Une autre spécificité du traitement de la langue amazighe est la contraction. Par exemple, le terme de parenté ⴱⴰⴱⴰ [baba] "père" suivi d'un pronom ⵏⵙ [nns] "son" devient le mot unique ⴱⴰⴱⴰⵙ [babas] "son père".

2.4 Transcriptions de l'amazighe

Le Maroc a assisté à un débat dont le point central est la langue et l'identité amazighe/ berbère. Ce débat sur l'identité amazighe n'existait pas dans le Maroc précolonial. Les autorités du protectorat français (1912-1956) ont souligné une dichotomie arabe / amazighe dans le cadre de leur politique de «division et règle» au Maroc. Cependant, cet effort n'a pas réussi à créer de profondes divisions entre les arabes et les amazighes. Plus récemment, certaines composantes de l'identité amazighe ont changé, donnant naissance à une communauté amazighe «réinventée», basée sur la langue et la culture (Crawford et Hoffman, 2000). Un des derniers débats sur l'amazighe concerne la langue : doit-elle être codifiée en alphabet latin, arabe ou l'ancien tifinaghe ?
Les amazighes possèdent donc depuis l'antiquité un système d'écriture qui leur est propre. Cependant, depuis l'aube de l'histoire, lorsqu'il s'agit de rédiger des documents consistants, les amazighes ont eu recours aux langues et/ou aux alphabets des peuples dominants avec lesquels ils étaient en contact : punique, latin puis arabe. Pour transcrire l'amazighe, de nos jours, trois systèmes d'écriture sont utilisés : le tifinaghe, l'alphabet arabe et l'alphabet latin (Pouessel, 2008), (Bachir et Ait Ben Ali, 2016).

2.4.1 Tifinaghe

On pense que l'alphabet tifinaghe provient de l'ancien script amazighe. Le nom tifinaghe signifie peut-être 'les lettres phéniciennes', ou peut-être de l'expression tifinaghe, qui signifie 'notre invention' (El Maadani, 2014). Différentes versions de tifinaghe sont utilisées pour écrire en amazighe au Maroc, en Algérie, au Mali et au Niger. Le manuscrit moderne du tifinaghe est aussi connu comme Touareg, Berbère ou Néo-Tifinaghe, pour le distinguer de l'ancien manuscrit berbère. Il est également utilisé par les Touaregs, en particulier les femmes, pour les notes privées, les lettres d'amour et dans la décoration.

En 2003, le tifinaghe est devenu la graphie officielle de la langue amazighe au Maroc. Parmi les différentes variantes du tifinaghe présentes au Maroc, l'IRCAM n'a conservé, après une étude approfondie du système phonético-phonologique, que des phonèmes pertinents pour la transcription de l'amazighe. Il s'agit d'une sélection de 33 graphèmes. Les critères pris en compte pour le choix de ces phonèmes sont : la fréquence des graphèmes dans les variantes, leur simplicité au niveau de l'écriture manuelle, l'esthétique des symboles et la cohérence d'ensemble du système d'écriture proposé.
Le processus du codage s'est basé sur le standard Unicode, qui a réservé à l'alphabet tifinaghe quatre sous-ensembles. Le premier sous-ensemble représente les 33 lettres alphabétiques de base. Les labiovélaires ⵡ (Gw) et ⴽⵯ (kw) ont été aménagées de sorte que la diacrité soit indépendante et considérée comme un caractère autonome. Le deuxième sous-ensemble contient les 8 caractères de la liste étendue, qui a été définie par l'IRCAM pour l'intérêt historique, scientifique et stylistique. Le troisième sous-ensemble est formé de 4 lettres néo-tifinaghes utilisées fréquemment dans le reste du Maghreb. Et le quatrième sous-ensemble contient 11 lettres touarègues modernes dont l'usage est attesté. Le nombre total de ces caractères est 55 (Agnaou *et al.*, 2017). La figure 2.3 représente le répertoire de Tifinaghe qui est reconnu et utilisé au Maroc.

FIGURE 2.3 – Alphabet tifinaghe utilisé au Maroc.

Le script amazighe est écrit de gauche à droite. Il utilise des signes de ponctuation conventionnels acceptés en alphabet latin. Les majuscules, cependant, ne se trouvent ni au début des phrases ni à l'initiale des noms propres. Il n'y a donc pas de concept de caractères majuscules et minuscules en langue amazighe. En ce qui concerne les chiffres, il utilise les chiffres arabes occidentaux. La majorité des modèles graphiques des caractères sont composés de segments. De plus, tous les segments sont verticaux, horizontaux ou diagonaux.

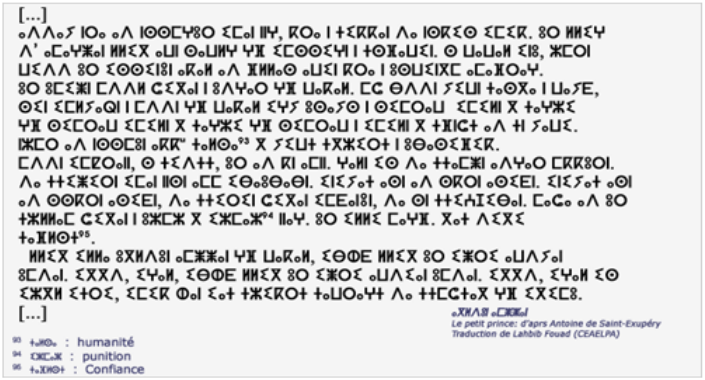


FIGURE 2.4 – Exemple de texte en amazighe.

2.4.2 L'alphabet arabe

Depuis l'indépendance (1956) avec le rattachement à l'identité étatique arabo-musulmane et sa politique linguistique d'arabisation, l'alphabet arabe s'est imposé comme la norme de l'écriture officielle de la langue du pouvoir et de la religion. L'influence de la culture arabe, est présente en grande partie à l'Afrique du nord, mais apparait également moins fréquemment dans le nord du Mali et du Niger, fusionné avec la culture islamique qui a apporté, quant à elle, son propre scénario. Depuis toujours, la culture arabe et l'écriture arabe ont été indissociables mais avec l'arrivée de l'islam, la langue arabe s'est imposée surtout avec la révélation coranique qui lui a conféré son statut de langue sacrée. Son caractère unique et sa beauté ont forgé l'admiration des musulmans, au-delà des disparités ethniques et géographiques (Abu-Absi, 2016).
Les amazighes ont écrit en arabe depuis le seizième siècle. De nos jours, l'écriture arabe est souvent utilisée par les auteurs amazighes et les gens ordinaires pour créer des écrits et pour écrire des lettres personnelles. La figure ci-après représente un exemple de transcription de la langue amazighe avec le script arabe (Ennaji, 2005).

FIGURE 2.5 – Sourate Al Fatiha en tashelhit écrit en caractères arabes Traduction Hassan Jouhadi.

2.4.3 L'alphabet Latin

Dès le début, sous prétexte de préserver la particularité berbère, les Français cherchaient à protéger les amazighes de la contamination arabe. Ils ont porté leur attention sur la culture amazighe, en recueillant leur patrimoine oral, enregistrant leurs traditions et écrivant la langue amazighe en caractères latins. Ce faisant, les colons français visaient à créer une conscience amazighe à travers un processus de réinvention identitaire. La création de l'autre n'était pas un processus aléatoire, mais une entreprise systématique dans laquelle le pouvoir et la connaissance travaillaient main dans la main. Ce processus de construction de nouvelles identités commodes des colons n'est pas exclusif aux Français. Tout au long de l'histoire, les cultures dominées ont été réinterprétées dans les récits des cultures qui les gouvernaient. Cette collaboration est un moyen efficace de contrôler les nations, comme l'a souligné Foucault (1980, p. 52), « il n'est pas possible d'exercer le pouvoir sans le savoir, il est impossible que le savoir n'engendre pas le pouvoir ». La connaissance n'est ni objective ni impartiale, et les faits ne sont pas découverts, mais créés. Le pouvoir crée des faits, tandis que la connaissance produite par les intellectuels confirme et met à jour ces faits. En développant une nouvelle identité amazighe à la fin du 19^e siècle, , les colons français ont été aidés par une armée de scientifiques, les anthropologues, les missionnaires et les écrivains de voyage qui ont facilité le processus de colonisation en soulignant et en diffusant les faits créés par les autorités françaises. Ils ont affirmé que ce script faciliterait la communication avec le monde extérieur et aiderait les amazighes à adopter la modernité et à accéder à l'information globale, y compris aux dernières technologies.

FIGURE 2.6 – Tableau de transcription de l'amazighe par la graphie arabe, latine et tifinaghe..

Les signes de ponctuation sont présents dans les différentes transcriptions de la langue amazighe. D'ailleurs, l'IRCAM a adopté les signes de ponctuation conventionnels, utili-

sées dans les écritures latines :« » (espace), «. », « , », « ; », « : », « ? », « ! », « ... », etc, et les chiffres arabes utilisés dans la numérotation occidental (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Dans ce travail, nous nous intéressons à la transcription de la langue amazighe en latin.

2.5 Corpus pour l'OCR

En traitement automatique des langues (TAL), un corpus est une collection de documents de taille importante, constituée de données authentiques rassemblées selon des critères spécifiques et collectées sous format électronique. Ces corpus sont utilisés pour effectuer des analyses statistiques et des tests d'hypothèses, vérifier des occurrences ou valider des règles linguistiques dans un territoire linguistique donné.
Un corpus peut contenir des données relevant d'une seule langue (corpus monolingue) ou des données dans plusieurs langues (corpus multilingue).
Il existe plusieurs types de corpus, dont le corpus linguistique et le corpus de reconnaissance. Un corpus linguistique est généralement utilisé à des fins linguistiques, il se caractérise par sa grande taille pouvant atteindre des dizaines de millions dans certains cas. Cependant, l'objectif d'un corpus de reconnaissance est de former pour qu'un système puisse reconnaître ultérieurement un texte. Ce type de corpus se caractérise par la diversité des exemples de styles d'écriture et de polices, aussi par une assez grande taille pour un bon apprentissage. Dans notre travail, nous nous intéressons au dernier type de corpus. Un bon corpus implique un bon apprentissage et donc une bonne reconnaissance.

2.5.1 Critères d'un corpus

Dans certains axes de recherche, l'utilisation d'un corpus est indispensable. Dans ce cas l'absence d'un corpus adapté aux besoins implique l'obligation de concevoir un corpus ajusté aux critères demandés. Pour constituer un corpus, il faut prendre en compte un certain nombre de facteurs à titre d'exemple la taille, l'équilibre et la représentativité.

2.5.1.1 Taille

La taille du corpus dépend directement du type de traitements qui lui seront proposés. En général, un corpus plus grand est considéré comme meilleur, vu la présence d'une variété d'échantillons représentatifs. Cependant, il est possible d'obtenir un nombre important de données utiles d'un petit corpus, en particulier lors de la recherche d'éléments à haute fréquence. En fait, il est peut-être souhaitable de se contenter d'un petit corpus plutôt que d'être submergé par trop de données provenant d'un grand corpus.
Ainsi, le choix de la taille du corpus peut changer selon le cas d'utilisation et son niveau de complexité et aussi selon la disposition des échantillons.

2.5.1.2 Reprsentativité

Nous pouvons dire qu'un corpus est représentatif si les conclusions de ce corpus sont généralisables à la langue ou à un aspect particulier de la langue dans son ensemble. De toute évidence, il n'est pas possible de collecter toutes les données d'une langue pour tester la représentativité d'un corpus. Cependant, nous pouvons utiliser la notion de «saturation» (également appelée «fermeture»). La saturation (au niveau lexical) peut être testée en prenant un corpus et en le divisant en sections égales en termes de nombre de mots. Si une autre section de la même taille est ajoutée, le nombre de nouveaux éléments dans la nouvelle section devrait être approximativement le même que dans les autres sections.

2.5.1.3 Equilibre

La notion de l'équilibre est une notion étroitement liée à celle de la représentativité dans un corpus. Généralement, un corpus est dit équilibré quand la taille de ses sous-catégories (genres, registres etc.) est proportionnelle à leurs fréquences d'occurrence dans ce corpus. Bien que l'équilibre soit souvent considéré comme une condition sine qua non de la conception du corpus, car il n'existe aucune mesure scientifique fiable pour calculer l'équilibre du corpus. Au contraire, la notion repose fortement sur l'intuition et les meilleures estimations.
En général, il est important d'être pragmatique tout au long du processus de construction, car il peut y avoir des problèmes inattendus. Le processus de construction d'un corpus est cyclique. Au fur et à mesure que l'apprentissage avance, il est possible d'appliquer ces connaissances à l'ensemble du corpus et d'apporter des modifications, notamment en omettant les données collectées, si cela améliore le corpus final. Il faut également gardez une trace détaillée des données collectées. Si ces données sont textuelles par exemple, il faut noter les références de la source du texte, les noms des auteurs, l'année de la publication, l'origine, etc. Ces informations peuvent être intégrées ultérieurement dans le corpus et utilisées pour un plus grand nombre de recherches.

2.5.2 Méthodologie adoptée pour la création du corpus

Le corpus est réalisé en utilisant une structure systématique à partir des documents collectés. Avant de construire un corpus, il est nécessaire de passer par deux étapes : L'analyse de la langue et la réalisation du corpus.

2.5.2.1 L'analyse de la langue

Cette étape fait référence à une analyse détaillée sur la langue étudiée ainsi que les propriétés de cette écriture telles que l'alphabet utilisé, la structure des phrases, la

ponctuation, ...

2.5.2.2 *La réalisation du corpus*

Cette étape consiste à faire la collection des données et les organiser de manière à former un corpus qui répond aux besoins. Pour se faire, il est primordial de passer par trois étapes. La première étape est la composition, elle consiste à déterminer les éléments de base pour construire le corpus. La deuxième étape est la segmentation du document dans le but d'avoir l'entité du traitement souhaité. La troisième et la dernière étape est la conversion des images dans laquelle, pour chaque image, un fichier contenant le texte de l'image est créé.

2.6 Analyse de la langue étudiée

Comme indiqué précédemment dans ce chapitre, la langue amazighe est l'une des langues les plus anciennes de l'humanité. Pour la transcrire, trois systèmes d'écriture sont utilisés : Le tifinaghe, l'alphabet arabe et l'alphabet latin. Nous nous intéressons dans notre étude à la transcription de la langue amazighe en latin. Un exemple de texte transcrit en latin est présenté dans la figure suivante.

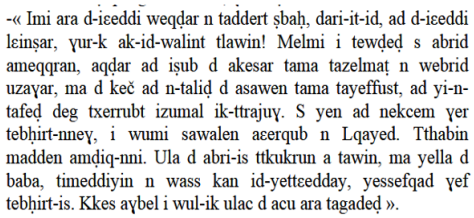

-« Imi ara d-iɛeddi weqḍar n taddert ṣbaḥ, dari-it-id, ad d-iɛeddi leinṣar, ɣur-k ak-id-walint tlawin! Melmi i tewḍeḍ s abrid ameqqran, aqḍar ad iṣub d akesar tama tazelmaṭ n webrid uzaɣar, ma d keč ad n-taliḍ d asawen tama tayeffust, ad yi-n-tafeḍ deg txerrubt izumal ik-ttrajuɣ. S yen ad nekcem ɣer tebḥirt-nneɣ, i wumi sawalen aɛerqub n Lqayed. Tthabin madden amḍiq-nni. Ula d abri-is ttkukrun a tawin, ma yella d baba, timeddiyin n wass kan id-yettɛedday, yessefqad ɣef tebḥirt-is. Kkes aɣbel i wul-ik ulac d acu ara tagadeḍ ».

FIGURE 2.7 – Extrait de "Azal n tayri" (AMARA, 1999).

2.6.1 Observations sur cette langue

Notre langue d'étude est l'amazighe transcrite en latin, qui se base sur des caractères avec diacritiques. Les diacritiques sont des signes qui apparaissent, comme illustré dans la figure 2.7, en dessous, au-dessus, avant ou après les caractères. Outre ces caractères, la transcription utilise également des caractères spéciaux notamment epsilon « E / ϵ » et gamma «Γ / γ» (El Gajoui *et al.*, 2016).
Dans cette étude qui s'articule autour ce type d'écriture, nous avons utilisé des livres de différentes catégories et dates. Parcourir ces livres, nous a permis de faire quelques observations sur cette langue. Parmi ces observations, nous pouvons citer l'apparition

fréquente de certains caractères et mots. Les caractères et les séquences de caractères les plus fréquents sont représentés dans le tableau suivant.

Caractères fréquents	Mots fréquents	
	Mots	Traduction en Français
A	Iγ	Si
E	ma	Qu'est ce que
I	ur	Ne pas
ṭ	ar	En train de
ṛ	skarn	Faire
Ż	nns	Son / Sa
ϵ	win	de(préposition)
ṣ	ula	Et / aussi

TABLEAU 2.1 – Exemple de caractères et de mots fréquemment utilisés

2.6.2 Jeux de caractères

Au fil du temps, plusieurs versions de jeux de caractères ont été utilisées pour la transcription de la langue amazighe en latin. Il y a des jeux de caractères qui sont anciens et d'autres plus récents. Pour cette étude, nous avons rassemblé différents jeux de caractères. Divers types et âges de livres sont couverts.
Les jeux de caractères pouvant être une généralisation de tous les caractères utilisés pour la transcription de la langue amazighe en latin sont répertoriés dans les ouvrages listés dans le tableau suivant :

Livre	**Auteur**	**Année**
L'Arganier et son lexique Tashelhiyt Berber (Stroomer, 2008)	Harry Stroomer	2008
Conte berbère grivois du haut atlas (Leguil, 2000)	Alphonse Leguil	2000
Choix de la version berbère du sud-ouest marocain (Roux, 1951)	Arsène Roux	1951
Manuel de Berbère Marocain (Dialecte Rifain) (Justinard, 1926)	Léopold JUSTINARD	1926
Mots et choses berbères (Laoust, 1920)	Emile Laoust	1920

TABLEAU 2.2 – Liste des livres contenants les différents jeux de caractères utilisés

2.6.3 Unicode

Les caractères avec des signes diacritiques sont considérés comme des caractères spéciaux. Étant donné que de nombreux éditeurs de texte ne peuvent pas lire ce type de caractère, nous devons spécifier leur Unicode. Unicode est un type d'encodage standard et universel pour représenter des données. Ainsi, elles peuvent être traitées par différents éditeurs et facilement accessibles à d'autres chercheurs.
Ci-après un tableau exposant l'Unicode de certains caractères spéciaux utilisés dans la transcription.

Caractère	**Unicode**	**Caractère**	**Unicode**
Ă	0103	ů	016F
˙B	1E05	ŭ	016D
ḍ	1E0D	ṛ	1E5B
ì	011B	ḷ	1E37
ġ	0121	š	0161
¯g	1E21	š	1E61
ḥ	1E25	ṭ	1E6D
ẖ	1E2B	ẓ	1E93
ŏ	014F		

TABLEAU 2.3 – Unicode de certains caractères spéciaux utilisés dans la transcription amazighe.

2.7 Corpus pour l'OCR de l'amazighe

Le but de ce travail est de construire un corpus pour la langue amazighe transcrite en latin.
A notre connaissance, à l'exception de nos contributions (EL Gajoui *et al.*, 2015b), (EL Gajoui *et al.*, 2015a), aucune étude n'a été faite ni sur un système OCR dédié à la langue amazighe transcrite en latin ni sur la construction d'un corpus pour une telle transcription. Cette langue est considérée comme une langue diacritique basée sur des caractères latins ornés de signes diacritiques.
Dans ce contexte, nous nous sommes basés sur des documents anciens et d'autres plus récents pour constituer un corpus prenant en compte les deux types de documents.
Étant donné que les types de caractères utilisés dans la transcription ne sont pas compatibles avec toutes les polices par défaut. Nous avons pris en compte ce critère lors de la construction du corpus.

2.7.1 Composition

Notre corpus est composé d'un ensemble de fichiers sous format image contenant un texte écrit en amazighe transcrite en latin. Chaque image est associée à un fichier texte correspondant à la transcription textuelle du fichier image. L'image et le fichier texte correspondant doivent porter le même nom pour que le système puisse les identifier.
Notre objectif derrière la création de ce corpus, est la formation d'un système permettant de reconnaître des documents récents ainsi que d'anciens documents. À cette fin, nous avons divisé notre corpus en deux parties.

1. La création de la première partie du corpus est basée sur un texte saisi en amazighe transcrite en latin. Ce texte respecte un ensemble de critères, à savoir :
   - le texte dans les images doit avoir un sens (extrait d'un livre ou document) ;
   - l'existence des signes de ponctuation ;
   - la variation de la langueur des lignes de texte ;
   - l'existence des différents jeux de caractères trouvés pour la transcription de la langue amazighe ;
   - la présence des mots les plus fréquemment confrontés mentionnés en haut ;
   - la présence au moins 5 fois de chaque caractère.

   Nous avons appliqué à ce texte différentes tailles et différentes polices adaptées à ce type de caractères. A titre d'exemple des polices utilisées, nous citons : Arial, Cambric, Charissil, Tahoma, Calibri, Doulos.
   À partir des textes résultants, de taille et de polices différentes, nous avons généré un ensemble d'images de texte. Chaque image de texte correspond à un fichier de texte avec la transcription du texte sur une image.

2. La deuxième partie du corpus est basée sur des documents anciens et aussi récents. Les anciens documents sont caractérisés par des polices rares ou parfois inutilisées actuellement ainsi que d'autres facteurs à savoir : les ligatures, caractères déplacés, caractères aux limites floues, fond perdu dans la page suivante, taches, etc. Le type de papier et l'effet du scanner sur des documents anciens peuvent également influencer la qualité du document et donner, dans certain cas, un effet de transparence ce qui constitue une particularité pour ce type de documents.

Les images que nous utiliserons pour la construction de la 1ère partie du corpus sont des images numérisées des textes saisis, tandis que dans la 2^e partie, ce sont des images scannées des documents et livres imprimés.
Vous trouverez un exemple de liste de livres utilisés et la répartition entre les différentes catégories dans le tableau ci-dessous.

Catégorie	**Livre**
Linguistique	Tirra - Aux origines de l'écriture au Maroc (Skounti *et al.*, 2003)
Romance	Ijawwan n tayri (Lasri, 2008)
Littérature	Conte berbère grivois du haut atlas (Leguil, 2000)
Anthologie	Une anthologie de contes folkloriques Tashelhiyt Berber (Stroomer, 2001)

TABLEAU 2.4 – Différentes catégories de livres.

2.7.2 Etapes de construction

Afin de garantir une variété dans notre corpus construit, nous choisissons de distinguer trois niveaux à étudier : la ligne, le mot et le caractère.

2.7.2.1 Segmentation des images

La première étape vers la construction de notre corpus est la segmentation des images. Un histogramme horizontal et vertical est utilisé pour effectuer cette segmentation.
L'histogramme est un mode de représentation graphique de la distribution tonale d'une image (El Ayachi *et al.*, 2011), (Patel *et al.*, 2013). Autrement dit, c'est la représentation visuelle des intensités de chacun des niveaux qui composent une image numérique. L'histogramme associe à chaque niveau, de 0 pour noir à 255 pour blanc, le nombre de pixels correspondant dans l'image considérée (Kharate *et al.*, 2013), (Nguyen et Nakagawa, 2016). Le niveau 128 représente en toute théorie le gris moyen, qui se situe donc à mi-chemin entre le noir et le blanc.
Dans le cas de la segmentation de texte, nous utilisions l'histogramme donné par :

$$HistV = \sum \sum pixel(x, y)$$

où la fonction pixel (x, y) représente l'intensité du pixel de cordonnées (x, y)).
Dans cette opération, tout d'abord, nous calculons les histogrammes horizontaux et verticaux. Ensuite, l'histogramme horizontal est parcouru dans deux directions : de haut en bas et de bas en haut, respectivement jusqu'à la première réunion de pixels noirs. Enfin, l'histogramme vertical est aussi parcouru dans deux directions : de gauche à droite et de droite à gauche, respectivement, jusqu'à la première réunion de pixels noirs.

Après l'obtention de la position des premiers pixels noirs, les zones non désirées sont éliminées.
Cette méthode est utilisée pour l'extraction des lignes. Elle est considérée comme l'une des plus faciles méthodes de segmentation et celle la plus ancienne. Elle a été largement appliquée dans les travaux pour différentes langues et scripts. Pour l'extraction des caractères, nous utilisons souvent les approches basées sur l'analyse de l'histogramme de projection vu sa simplicité à l'application et à la manipulation (Syiam *et al.*, 2006). Cependant, cette approche ne peut pas être utilisée dans le cas de l'écriture manuscrits à cause de la présence de l'aspect cursif, de ligatures verticales et aussi à cause de la variabilité de l'épaisseur du trait d'écriture. Une autre limitation de cette approche se manifeste pour les documents fortement inclinés où les caractères se chevauchent d'où la difficulté d'extraction.
Dans notre cas, nous pouvons utiliser l'histogramme dans la phase de segmentation. Le pseudo-code de la segmentation du texte de l'image est présenté comme suit :

Procedure : Segmentaion d'une Image texte
Entrée: Image texte **M**
Sortie: Images contiennent un mot ou un caractère **m**

1) Prétraitement de l'image **M**
2) Appliquer l'histogram vertical **H**
3) Segmentation de l'image **M** en **n** image de lignes (**H**=0 correspond au blanc entre les lignes)
4) Pour i=1,...,**n** faire
5) Appliquer un histogramme horizontal **h** sur l'image **Mi**
6) Segmenter en image mot/ caractére (corespondant à la succession du blancs tel que **h**=0) **mi**
7) Rassembler **mi** dans **m**

FIGURE 2.8 – Le pseudo-code de la segmentation du texte de l'image.

2.7.2.2 Conversion des images

Après avoir obtenu des images segmentées, nous procédons à la conversion de ces images en texte afin de créer les fichiers texte correspondant à chaque image.
Pour effectuer cette conversion, nous pouvons utiliser l'une de ces deux méthodes :

- La première méthode est la plus simple et la plus évidente. Elle consiste à faire appel à un système de reconnaissance optique de caractères. Comme le système n'atteint pas un taux de reconnaissance de 100%, il est impératif de procéder à la vérification des résultats après la reconnaissance en tant que phase de post-traitement manuel.
- La deuxième méthode est la méthode manuelle, où nous devons parcourir les images de texte une par une, créer le fichier texte avec le même nom que l'image de texte et y placer le texte correspondant. Ce processus est assez difficile et prend du temps compte tenu du grand nombre d'images existant dans un corpus. Cependant, cette méthode est efficace.

Etant donné le nombre important des échantillons dans notre corpus nous avons choisi d'appliquer la première méthode pour la conversion et nous avons utilisé notre système OCR développé.

Pour construire notre corpus, nous avons mélangé les deux parties précédemment construites.

2.7.3 Caractéristiques des corpus

Choisir le bon corpus est un aspect important dans l'apprentissage et le test d'un système de reconnaissance optique de caractères. La taille et le type sont des critères particuliers pour un corpus, ils jouent un rôle important dans le fonctionnement du corpus. Ce dernier peut être présenté sous l'une des trois formes possibles : ligne, mot ou caractère. Les images composant le corpus peuvent contenir une ligne, un mot ou un caractère, en fonction du type de corpus. La différence entre les 3 types réside dans la quantité d'informations dans une seule image. Une image avec une ligne ou un mot représente une succession de caractères fréquemment trouvés. L'image au niveau du caractère ne contient pas d'informations sur la succession fréquente de caractères, mais elle symbolise l'unité de base d'une écriture significative.
Dans ce travail, nous avons essayé de construire un corpus pour la langue amazighe transcrite en latin. Ce corpus correspond à trois niveaux spécifiques.
Corpus de lignes : Ce corpus est composé d'images contenant des lignes de texte. C'est un mélange des deux parties mentionnées ci-dessus. La première partie correspond aux images de texte générées à partir du texte créé. Pour cette partie, nous avons conçu 2 000 images de lignes de texte avec un fichier texte de transcription pour chaque image. La deuxième partie est basée sur l'extraction de lignes de livres cités précédemment. À cette fin, nous avons utilisé différents échantillons de pages de chaque livre. Nous avons extrait 5 à 10 pages par livre. Les exemples de pages sont variés : pages de couverture, résumés et types de paragraphes différents.
Dans cette partie, nous avons utilisé l'histogramme horizontal pour extraire les lignes. 672 images de texte ont été obtenues après la segmentation des pages de différents livres et nous avons créé leur transcription sous forme de fichier texte.
À l'heure actuelle, ce type de corpus comprend un total de 2672 images. Comme nous avons utilisé des documents de différentes pages et tailles de polices. Chaque ligne contient entre 1 et 21 mots. Un exemple du corpus de lignes est présenté sur la figure suivante :

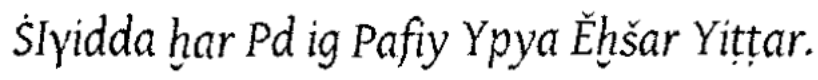


FIGURE 2.9 – Un exemple du corpus de lignes.

Corpus de mots : Dans ce corpus, nous trouvons des images contenant un mot chacune.

Comme dans le corpus de ligne, nous avons également mélangé les deux parties expliquées précédemment. Dans la 1ère partie, nous avons généré 4500 mots avec leurs transcriptions dans des fichiers texte. Pour la 2e partie, nous avons utilisé l'histogramme horizontal pour extraire les lignes. Ensuite, nous avons utilisé l'histogramme vertical pour obtenir des images de mots. Nous avons rassemblé 1100 images de mots, dont nous avons créé le fichier texte de transcription correspondant.
Au total, ce corpus est composé de 5600 images de mots. Chaque image est associée à un fichier texte de transcription. Un échantillon de ce corpus est illustré dans la figure ci-après.

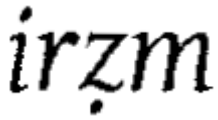


FIGURE 2.10 – Un exemple du corpus de mots.

Corpus de caractères : Les images contenues dans ce type de corpus sont des images de caractères isolés. Pour les raisons évoquées précédemment, les deux parties ont également été réunies dans ce corpus. Le résultat pour la 1ère partie est un ensemble de 9800 images de caractères avec leurs transcriptions. Dans la deuxième partie, comme dans le corpus de type mot, nous avons utilisé un histogramme horizontal et vertical pour extraire les caractères des pages du livre. Le résultat étant 2500 images chacune associée à sa transcription sous forme de fichier texte. Nous avons un total de 12 300 images de caractères avec leurs transcriptions. Un exemple de ce corpus est présenté sur la figure suivante.

FIGURE 2.11 – Un exemple du corpus de caractères.

Ci-après un tableau récapitulatif des types de corpus avec leurs tailles.

Niveau du corpus	Taille du corpus		Taille totale
	1ère partie	2e partie	
Ligne	2000	672	2672
Mot	4500	1100	5600
Caractère	9800	2500	12300

TABLEAU 2.5 – Tableau récapitulatif des niveaux et tailles du corpus.

2.8 Conclusion

La langue amazighe est parlée par une population importante en Afrique. Elle est utilisée par des dizaines de millions de personnes en Afrique du Nord principalement pour la communication orale et a été introduite dans les médias et dans le système éducatif en collaboration avec plusieurs ministères marocains. En termes linguistiques, la langue est caractérisée par la prolifération de dialectes en raison de facteurs historiques, géographiques et sociolinguistiques. Au Maroc, le terme amazighe englobe les trois principales variantes marocaines : tarifite, tamazight et tachelhit.
La création de l'Institut Royal de la Culture Amazighe a mené une action majeure visant à normaliser la langue amazighe. Dans la même démarche, et depuis 2003, la langue amazighe est enseignée dans les classes primaires des différentes écoles marocaines, dans la perspective d'une généralisation progressive au niveau de l'école et de son extension à de nouvelles écoles. Différents alphabets ont été utilisés pour la transcription de la langue amazighe suite à la colonisation et le contact avec d'autres peuples par exemple les arabes et les romains. Dans ce chapitre, les documents imprimés écrits en amazighe et transcrits en latin sont pris en compte. Notre première contribution vise à développer un corpus dédié à cette langue en l'absence d'un corpus représentant l'amazighe transcrite en latin, qui servira de base à l'apprentissage d'un système de reconnaissance optique de caractères, en mesure de transformer ce type de document en version électronique avec un taux de reconnaissance considérable. Dans ce contexte, nous avons d'abord mené une étude analytique sur la langue amazighe, ensuite, nous avons créé un corpus qui correspond à trois niveaux, qui sont : la ligne, le mot et le caractère.

Chapitre

3 Système basé sur l'approximation polygonale et le classifieur adaptatif

Sommaire

3.1 Introduction

Dans ce chapitre, nous présentons notre deuxième contribution, qui consiste en un système OCR assurant le traitement de la langue amazighe transcrite en lettres latines, caractérisées par la présence des signes diacritiques. Dans ce contexte, nous avons utilisé la méthode de binarisation non linéaire Niblack dans la phase de prétraitement et nous avons adopté une approche structurelle basée sur l'approximation polygonale dans la phase d'extraction des caractéristiques. Concernant la phase de classification, nous avons choisi d'utiliser un classifieur adaptatif basé sur la règle du maximum de vraisemblance.

Dans la suite de ce chapitre, nous introduisons l'architecture du notre système OCR proposé, ainsi que les modules qui le compose dans la section 2. Dans la section 3, nous exposons les différents prétraitements utilisés et nous présentons une contribution

axée sur la phase d'extraction des caractéristiques basée sur l'approximation polygonale dans la section 4. Dans la section 5, nous développons la phase de classification par la présentation du classifieur adaptive choisi. Puis, nous introduisons, dans la section 6, le type de corpus utilisé pour l'apprentissage du classifieur ainsi que l'évaluation du système proposé. Enfin, dans la section 7, nous déclinons des conclusions.

3.2 Architecture du système proposé

Dans le cadre du traitement automatique de l'amazighe, nous avons élaboré un système OCR dont le but est de reconnaître la langue amazighe transcrite en lettres latines avec des signes diacritiques.
La première étape dans la création du SOCR est la conception d'une architecture rassemblant les principaux modules du système, qui fonctionnent selon un processus bien déterminé.

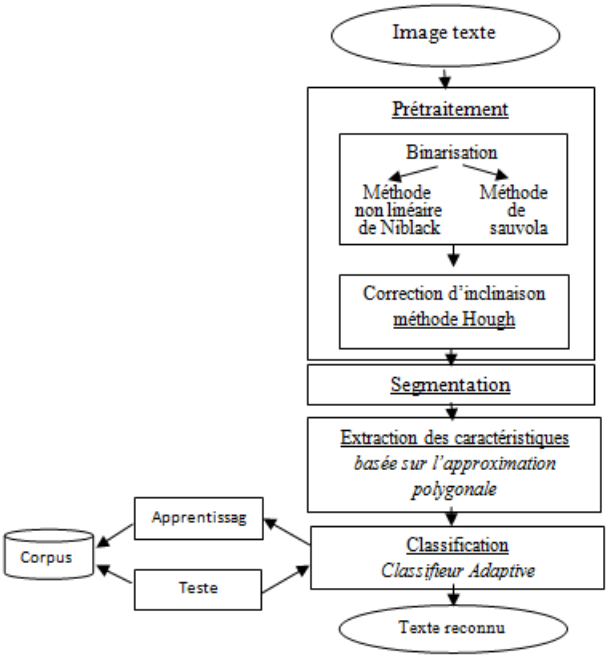


FIGURE 3.1 – L'architecture de notre système

Ce processus s'initie par une image de texte en entrée du système après l'avoir numérisée par un scanner. Ensuite, l'image subit un ensemble de prétraitements afin d'augmenter la qualité de l'image. Le premier traitement dans cette phase est la détection et la correction de l'inclinaison. Pour cela, nous utilisons la transformation de Hough. Le deuxième traitement est la binarisation, pour cela nous essayons deux méthodes : la méthode de

binarisation non linéaire, qui est une méthode de calcul intensif, très adaptée pour les documents historiques et de qualité dégradée ; et la méthode de Sauvola, basée sur le calcul du seuil à l'aide de la plage dynamique de l'écart-type du niveau de gris de l'image. Dans la phase d'extraction des caractéristiques, nous avons choisi d'utiliser les caractéristiques de type statistique et structurel. Dans ce cadre, nous utilisons les fragments d'approximation polygonale comme entité. Ensuite, nous utilisons, dans la phase de classification, un classifieur adaptatif basé sur la règle du maximum de vraisemblance. La classification est procédée par une phase d'apprentissage et une autre phase de test basée sur le corpus construit.

3.3 Prétraitement

Le prétraitement est une phase importante du système OCR. Il vise à améliorer la qualité de l'image pour augmenter le taux de reconnaissance.
Dans cette phase, nous nous intéressons à transformer l'image scannée en noir et blan dans le but de faciliter son traitement, et minimiser l'effet de l'inclinaison, produite généralement dans la phase d'acquisition.

3.3.1 Binarisation

La binarisation d'une image se fait, en général, à l'aide d'un seuil. Le seuil de binarisation correspond à la limite entre les contrastes forts et faibles de l'image. D'après l'étude bibliographique détaillée dans le chapitre 1 section 7, nous avons choisi d'adopter des approches locales pour le seuillage, à savoir : la méthodes non linéaire Niblack et la méthode de Sauvola.

3.3.1.1 Méthode non linaire Niblack

Dans le seuillage local, les valeurs de seuil sont spatialement variées et déterminées sur la base du contenu local de l'image cible. Par comparaison avec les techniques globales, les techniques de seuillage locales ont de meilleures performances contre le bruit et l'erreur, en particulier lorsqu'il s'agit d'informations à proximité de textes ou d'objets. Selon la littérature, la méthode non linéaire Niblack (Niblack, 1990) est une des méthodes de seuillage locales les plus performantes grâce à sa simplicité et efficacité. En conséquence, nous avons décidé de nous concentrer dans nos travaux sur cette dernière. La méthode non linéaire Niblack est basée sur le calcul de la moyenne locale et de l'écart type local. Le seuil est déterminé par la formule :

$$T(x,y) = m(x,y) + k * s(x,y) \tag{3.1}$$

où m(x, y) et s(x, y) sont respectivement la moyenne d'une zone locale et l'écart type. La taille du voisinage doit être suffisamment petite pour préserver les détails locaux, mais en même temps suffisamment grande pour supprimer le bruit. La valeur de k est utilisée pour ajuster la quantité de la limite totale de l'objet. Zhang et Tan ont proposé une version améliorée de l'algorithme de la méthode non linéaire Niblack (Zhang et Tan, 2001) :

$$T_w(x,y) = m(x,y) + k\sqrt{\frac{\sum(I_w(x,y) - m(x,y))}{NP}} \tag{3.2}$$

Le paramètre k est utilisé pour déterminer le nombre des pixels des contours considérés comme des pixels d'objet et il prend des valeurs négatives. NP représente le nombre des pixels de la fenêtre w. La méthode non linéaire Niblack, dans sa version améliorée, donne de bons résultats du faite que le seuil dépend du pixel et de l'information extraite à partir de son voisinage où le paramètre k est utilisé pour réduire la sensibilité au bruit. Cependant, cette méthode n'est pas efficace dans le cas où le fond n'est pas uniforme.

3.3.1.2 *Méthode de Sauvola*

Les images de document en niveaux de gris contiennent des valeurs d'intensité comprises entre 0 et 255. Contrairement à la binarisation globale, les méthodes de binarisation locale calculent un seuil (x, y) pour chaque pixel de sorte que

$$b(x,y) = \begin{cases} 0 & if g(x,y) \leq t(x,y) \\ 255 & ailleurs \end{cases} \tag{3.3}$$

Dans la méthode de binarisation de Sauvola (Sauvola et Pietikäinen, 2000), le seuil t(x, y) est calculé en utilisant la moyenne μ(x, y) et l'écart type σ(x, y) des intensités de pixels dans une fenêtre w × w centrée autour du pixel (x, y) :

$$t(x,y) = \mu(x,y) * [1 + k * (\frac{\sigma(x,y)}{R} - 1)] \tag{3.4}$$

où R est la valeur maximale de l'écart type (R = 128 pour un document en niveaux de gris), et k est un paramètre qui prend des valeurs positives. La formule précédente a été conçue de manière à ce que la valeur du seuil soit adaptée en fonction du contraste dans le voisinage local du pixel en utilisant la moyenne locale μ(x, y) et le standard local écart (x, y). Pour cette raison, il tente d'estimer le seuil t(x, y) approprié pour chaque pixel dans les deux conditions possibles : contraste élevé et faible. Dans le cas d'une région locale à contraste élevé (σ(x, y) R), le seuil t(x, y) est presque égal à μ(x, y). Dans une région de contraste assez faible (σ « R), le seuil y passe en dessous de la valeur moyenne en supprimant avec succès les régions relativement sombres de l'arrière-plan. Le paramètre k contrôle la valeur du seuil dans la fenêtre locale de telle sorte que plus la valeur de k est élevée, plus le seuil est bas par rapport à la moyenne locale m(x, y).

3.3.2 Détection et correction d'inclinaison

L'inclinaison peut être produite lors de la saisie, si le document a été placé en biais, ou lors de la numérisation, si la page a été incliné au scanne. La correction de l'angle d'inclinaison est une opération qui consiste à corriger la pente ou à redresser l'inclinaison des lettres dans un mot.


Dot Net introduced

Guntur: Gadde Srinivas of Tata Consultancy Services has said that Dot Net has a vast job potential. Speaking on 'Introduction to Dot Net' at the Training and Placement Cell of Vignan's Engineering College, he said it is user-friendly and any critical applications can be solved easily.

FIGURE 3.2 – Image inclinée.

Il existe plusieurs méthodes de détection de la ligne de base dans la littérature, mais la méthodes la plus connue et la plus utilisée dans la littérature est la méthode de transformation de Hough.
Cette méthode est appliquée sur les centres de gravité des composantes connexes (Burrow, 2004) , (Khorsheed, 2003) et les profils des histogrammes de projection. C'est une technique de détection de lignes et de courbes dans une image. Elle est utilisée aussi pour détecter l'angle d'inclinaison avec un intervalle de détection compris entre 0 ° et 180 °. Cette méthode est exacte, robuste et appropriée pour des documents multi-colonnes, mais elle nécessite un espace mémoire important et un temps de traitement prohibitif (Zahour *et al.*, 2004).
Chaque droite $D(\theta, \rho)$ est définie par :

$$M(x, y)/\rho = x * cos\theta + y * sin\theta$$

, où ρ est la distance entre l'origine et la ligne, et θ est l'angle entre la normale et l'axe des X.

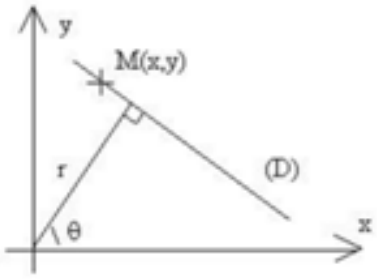


FIGURE 3.3 – La représentation d'une droite (D) dans l'espace

Deux types de transformations sont à distinguer suivant le principe de calcul :

- La transformation « Many to One », ou communément « m à 1 », fait correspondre à une droite passant par m points un et un seul point dans l'espace de Hough.
- La transformation « Many to One », ou « m à 1 », associe à chaque point de l'image la totalité des droites passant par ce point.

En raison de sa robustesse et sa certitude, nous avons opté d'appliquer la méthode de transformation de Hough dans nos expérimentations.

3.4 Extraction des caractéristiques basées sur l'approximation polygonale

La phase d'extraction des caractéristiques consiste à sélectionner les caractéristiques discriminantes des éléments segmentés. Elle est généralement admise comme une phase critique lors de la construction d'un système de reconnaissance dans le domaine de reconnaissance des formes.

Dans le cadre du développement de notre système, nous avons proposé une contribution dont le but est d'évaluer l'approximation polygonale comme type de caractéristiques à extraire, et étudier leur adaptation à la nature des caractères de la langue étudiée. Ce type de caractéristiques peut être considéré comme une combinaison entre des caractéristiques structurelles et statistiques décrites dans le chapitre 1 section 9.

Dans la littérature, l'approximation polygonale a montré sa capacité de présenter les caractères des scripts très complexes ainsi que sa résistance vis-à-vis du changement et variation du script.

3.4.1 Approximation polygonale

L'approximation des courbes bidimensionnelles arbitraires par des figures polygonales est une technique impérative dans le traitement d'image numérique. Une courbe numérique peut être efficacement simplifiée par un polygone sans perte de sa propriété visuelle. Les techniques d'approximation des polygones de la courbe numérique suscitent l'intérêt des chercheurs depuis des décennies (Srivastava *et al.*, 2019). Le nombre de segments de lignes utilisés dans le processus de création d'un polygone détermine la précision de l'algorithme d'approximation. Pour qu'un algorithme soit efficace et précis, il ne doit pas dépasser le nombre minimum de côtés requis pour conserver la forme réelle de la courbe. Un polygone ainsi créé avec uniquement le nombre minimum requis de segments de lignes est souvent nommé polygone à périmètre minimum. Un nombre plus élevé d'arêtes dans une figure polygonale approximative ajoute à la source le bruit du modèle. L'approximation du polygone supprime les points de courbe non essentiels et nous fournit une figure discrète qui est délimitée par des segments de lignes (Gupta et Bag, 2019). L'image résultante est constituée d'une simple région polygonale dans le plan délimité par un chemin polygonal fermé non auto-intersecté, comme illustré sur la figure 3.5.

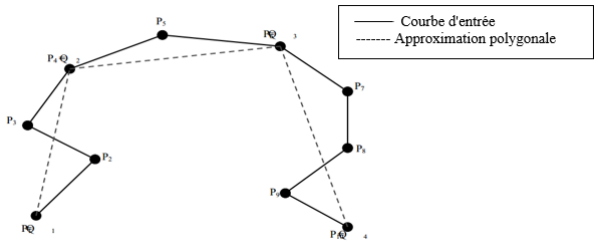


FIGURE 3.4 – Un exemple de l'approximation polygonale

Un algorithme correspondant à la méthode de l'approximation polygonale (Ngo, 2019) est décrit ci-dessous :

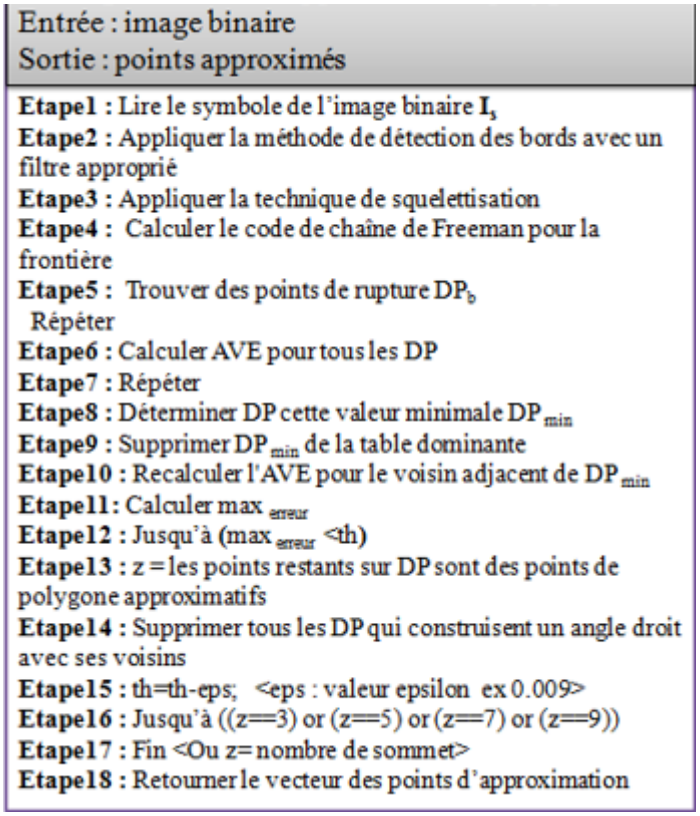


FIGURE 3.5 – Algorithme de l'approximation polygonale

3.4.2 Identification des critères optimales

Dans cette étude, nous avons utilisé l'approximation polygonale dans la phase d'extraction des caractéristiques pour la langue amazighe transcrite en latin. Pour implémenter et tester cette solution, nous avons utilisé l'outil de reconnaissance Tesseract.

3.4.2.1 Tesseract

Tesseract est un moteur de reconnaissance optique de caractères pour divers systèmes d'exploitation. Il a été initialement développé chez HP entre 1984 et 1994 (Nick, 2012). Il a été modifié et amélioré en 1995 avec une plus grande précision. Fin 2005, HP a publié Tesseract pour l'open source. Maintenant, il est développé et maintenu par Google.
La spécificité de la langue amazighe transcrite en caractères latins est la présence de diacritiques en dessous et au-dessus d'un grand nombre de caractères. Les expériences sur Tesseract pour les langues diacritiques, comme le grec ancien (Nick, 2012) et l'ourdou (Hussain *et al.*, 2014), ont montré qu'il est assez fort pour ce type de langues, d'où

l'intérêt d'utiliser cet outil. Pour procéder, il est impératif de faire l'apprentissage de Tesseract par la langue amazighe transcrite en caractères latins. Le processus d'apprentissage passe par des étapes : la constitution du corpus, la création du fichier traineddata et l'apprentissage (Smith, 2007).

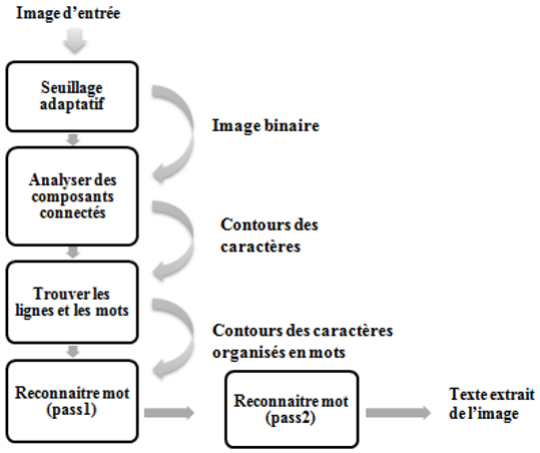


FIGURE 3.6 – Architecture de Tesseract OCR

La reconnaissance se déroule en deux étapes :

— Lors de la première passe, une tentative est faite pour reconnaître tour à tour chaque mot. Chaque mot satisfaisant est transmis à un classifieur adaptatif en tant que données d'apprentissage. Ensuite, le classifieur adaptatif a la chance de reconnaître plus précisément le texte plus bas dans la page.

— Dans la deuxième passe, le classifieur adaptatif parcourt la page pour reconnaître les mots qui n'étaient pas suffisamment reconnus lors de la première passe. Une phase finale résout les espaces flous et vérifie des hypothèses alternatives correspondantes à la hauteur x pour localiser le texte en petite majuscule.

3.4.2.2 L'apprentissage de Tesseract

L'apprentissage de tesseract passe par trios étapes. La première étape consiste en la génération des box qui correspondent au corpus. La deuxième étape est la création du fichier traineddata, utilisé pour l'apprentissage. La dernière étape concerne l'apprentissage du modèle.

Génération des BOX

La première étape consiste à générer un corpus composé de différents caractères utilisés dans la transcription de l'amazighe en latin. Pour cela, nous utilisons jTessBoxEditor qui est un éditeur de boîtes(BOX) et un formateur pour Tesseract OCR . Il fournit l'édition des données de boîtes pour les formats Tesseract 2.0x et 3.0x, et une automatisation complète de la formation Tesseract. Il peut lire des images de formats courants d'image,

y compris TIFF multipage. Le programme nécessite Java Runtime Environment 7 ou version ultérieure.
L'interface illustrée par la Figure 3.8 permet d'ajouter un fichier texte contenant des caractères à former, définir la police souhaitée et spécifier le degré de bruit afin de générer des boîtes.

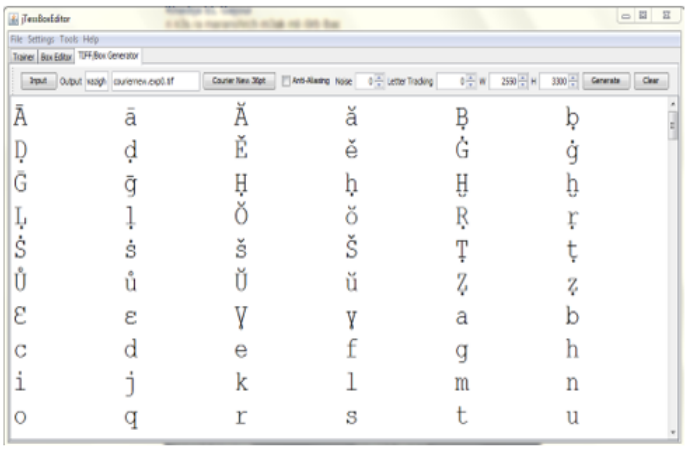


FIGURE 3.7 – Onglet générateur de boîtes de l'outil jTessBoxEditor

Après avoir créé notre fichier contenant les caractères, nous le téléchargeons et nous choisissons les polices appropriées au type de caractères. Les polices utilisées sont Arial, Calibri, Cambria, Charis SLI, Tahoma et Times new roman en gras ou / et en italique. Au total, nous avons 24 formats d'écriture qui diffèrent selon la police, la taille et la typographie.
Nous définissons le niveau maximum de bruit afin d'augmenter la qualité de reconnaissance. Ensuite, nous générons les boîtes. Chaque case correspond à un format d'écriture spécifique.
La figure 3.9 présente une boîte. Elle contient les coordonnées de chaque caractère dans l'image créée.

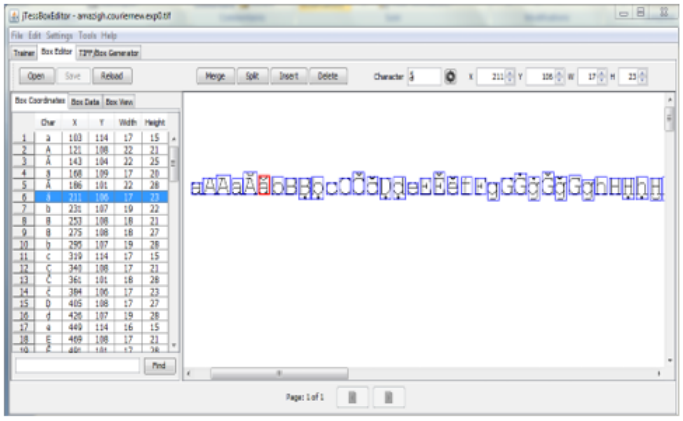


FIGURE 3.8 – Un exemple de BOX dans l'onglet éditeur de BOX

Création du fichier traineddata

Le fichier traineddata permet l'apprentissage du modèle. Pour créer ce fichier, nous avons besoin d'un ensemble de fichiers qui sont :

- Fichier BOX : l'ensemble des fichiers box générés précédemment.
- Fichier FONT_PROPERTISES : contient les polices utilisées avec leurs propriétés. Respectivement <italic>, <bold>, <fixed>, <serif> et < fraktur > sont tous de simples indicateurs 0 ou 1 indiquant si la police a la propriété nommée.
- Fichier FREQUENT_WORD_LIST (. Frequent_words_list) : fichier contenant des mots fréquemment utilisés dans la langue d'apprentissage tels que "nna" (qui), "n" (de), "nns" (son) et "I_γ" (si) en langue amazighe.
- Fichier WORDS_LIST (. Words_list) : liste de mots, contient au moins un mot de la langue.

Tous les fichiers doivent commencer par le nom de la langue définie lors de la phase de génération de la boîte.

Le fichier traineddata est créé dans l'onglet Formateur de l'outil jTessBoxEditor. Dans notre cas, les fichiers sont :

- Amazigh.arial.exp0.box : pour la police « arial ».
- Amazigh.font_properties
- Amazigh.frequent_words_list
- Amazigh.words_list

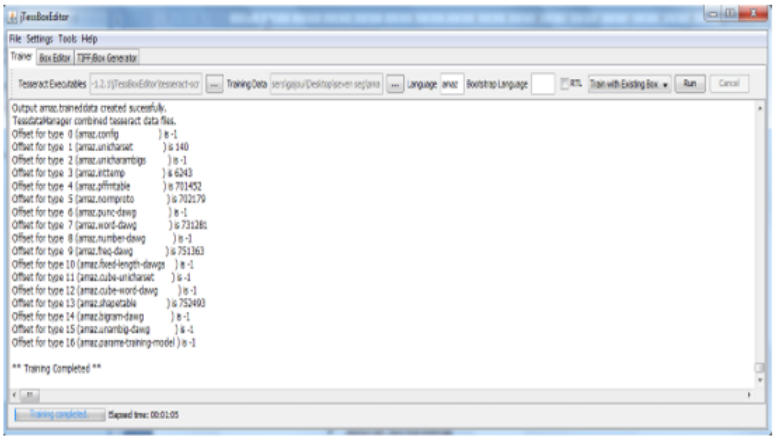


FIGURE 3.9 – L'onglet apprentissage de l'outil jTessBoxEditor

Apprentissage Tesseract

Dans cette étape, le fichier *traineddata*, considéré comme un corpus d'apprentissage pour le modèle, déjà généré, sera intégrer dans le modèle, plus exactement dans le dossier *tessdata* de Tesseract, afin de pouvoir effectuer les expérimentations avec un corpus de test.

3.4.2.3 Résultats et analyse

Le but étant d'identifier les critères optimaux de l'extraction des caractéristiques basés sur l'approximation polygonale avec ce modèle, nous utilisons l'interface graphique VietOcr. Après avoir utilisé le fichier *traineddata* généré pour faire apprendre le modèle, nous utilisons un ensemble de documents extraits de différents livres pour évaluer le système. Une partie de cette collection a subi un prétraitement pour augmenter la qualité de l'image tandis que l'autre est conservée avec une faible qualité afin de visualiser le comportement du système dans les deux cas.

Les documents sont divisés en deux parties :

- Doc 1 : documents de bonne qualité ;
- Doc 2 : documents de faible qualité.

Dans le cadre de cette évaluation, nous avons varié deux propriétés. La première est la taille de la police d'apprentissage et la seconde est la qualité du document à reconnaître. Les taux de reconnaissance sont indiqués dans le tableau suivant :

		Variation de la taille de la police		
		14pt	36pt	48pt
Variation de la qualité du document	Doc1	25%	92%	85%
	Doc2	18%	75%	70%

Tableau 3.1 – Taux de reconnaissance relatifs aux variations de la qualité du document et la taille de la police

Les résultats montrent que la reconnaissance améliorée est obtenue à 92% sur des documents de bonne qualité avec la taille de police 36pt. Nous notons que le pourcentage de reconnaissance varie considérablement entre le corpus en fonction de la taille 14pt, 36pt et 48pt.
Pour la taille de police 14pt, nous avons remarqué nombreuses erreurs de classification, par exemple :

- les majuscules sont confondues avec des minuscules ;
- un seul caractère est reconnu par plusieurs caractères : "n" est reconnu comme "rr", "m" comme "rrr" et "a" comme "zt" ;
- le point souscrit est généralement ignoré, "ṭ" confondu avec "t", "ḍ" avec "d" et "ḥ" avec "h" ;
- autres erreurs telles que "a" reconnu comme "z" et "γ" comme "Y".

Pour la taille de police 48 pt, il existe aussi des erreurs de classification telles que :

- les majuscules sont confondues avec des minuscules ;
- le point souscrit est ignoré dans "ẓ" et "ṛ" ;
- le diacritique non reconnu, "ḫ" est confondu avec "ḥ".

Bien que les expériences pour la taille de police 36pt montrent que cette taille est la meilleure pour l'apprentissage, les erreurs de classification avec cette taille de police ont été réduites à :

- les majuscules sont confondues avec des minuscules ;
- le caractère "ḍ" est confondu avec "d", "ṭ" avec "t" et "γ" avec "Y".

Malgré l'influence de la qualité du document sur les résultats de la reconnaissance, nous remarquons que même avec une qualité médiocre, le système basé sur l'approximation polygonale atteint un taux de reconnaissance important de 75% qui peut être amélioré dans la phase de prétraitement. D'autre part la taille de police peut être fixée à 36 d'après les résultats.
En conclusion, l'approximation polygonale choisie comme type de caractéristiques extraites a montré sa capacité à traiter les caractères de la langue amazighe transcrite en latin spécifié par la présence importante des diacritiques. Cette capacité est interprétée dans les taux de reconnaissance marqués dans cette expérience.

3.5 Classification

Comme mentionné précédemment dans ce chapitre, nous avons utilisé un classifieur adaptatif (CA) basé sur la règle du maximum de vraisemblance dans la phase de classification. Ce classifieur fait partie des méthodes de classification statiques, présentées dans le chapitre 1, section 10.
Un classifeur adaptatif a été proposé (Nagy, 1992) pour atténuer le problème de l'échantillon de l'apprentissage. Ce classifieur adaptatif améliore l'estimation statistique et améliore ainsi la précision de la classification de manière itérative en utilisant les échantillons semi-étiquetés, en plus des échantillons d'apprentissage originaux, dans l'estimation statistique subséquente (Marosi, 2007). Dans ce processus itératif, les échantillons sont initialement classés en fonction des statistiques estimées en utilisant uniquement les échantillons d'apprentissage originaux. Ensuite, les résultats classifiés sont utilisés avec les échantillons d'apprentissage originaux pour mettre à jour les statistiques de classe, et les échantillons sont reclassés par les statistiques mises à jour. Ce processus se répète jusqu'à ce que la convergence soit atteinte. Le classifieur adaptatif proposé présente potentiellement les avantages suivants :

1. Le grand nombre d'échantillons semi-étiquetés peut améliorer les estimations statistiques et par conséquent diminuer l'erreur d'estimation.
2. Les statistiques estimées sont plus représentatives de la véritable répartition des classes, car les échantillons utilisés pour estimer les statistiques proviennent d'une plus grande partie de l'ensemble de données.
3. Ce classifieur est adaptatif dans le sens où il peut améliorer la précision en utilisant les informations extraites de sa sortie.
4. Cette approche augmente l'automatisation du classifieur. Il est possible que pour commencer avec un petit nombre d'échantillons d'apprentissage (entrée minimale de l'analyste), ce classifieur puisse être en mesure d'extraire continuellement des informations utiles des données et de s'ajuster en conséquence.
5. Étant donné que les échantillons semi-étiquetés peuvent être réinjectés avant ou après l'extraction de toute caractéristique, il offre une flexibilité de mise en œuvre, c'est-à-dire que, selon l'exigence de précision et la charge de calcul, les échantillons semi-étiquetés peuvent être utilisés de plus d'une façon.

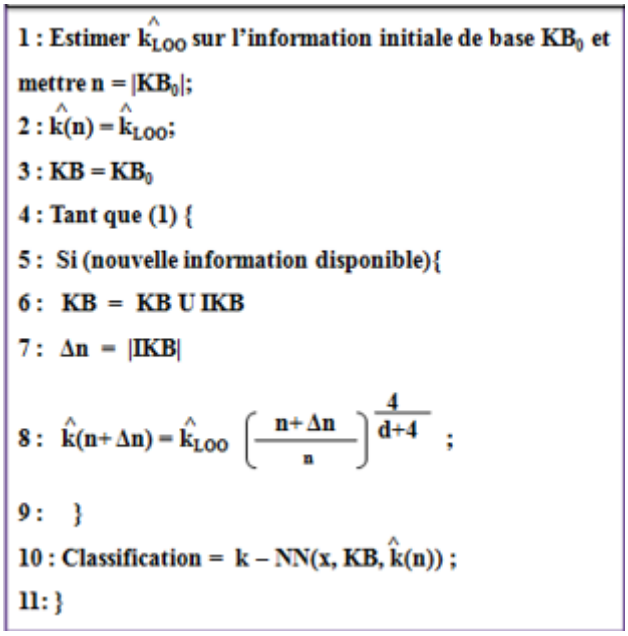


FIGURE 3.10 – Algorithme du classifieur adaptive

Dans notre cas, le classifieur adaptatif basé sur la règle du maximum de vraisemblance (en anglais maximum likelihood, ML) est proposé pour améliorer l'estimation statistique, en utilisant des échantillons semi-étiquetés en plus des échantillons d'apprentissage. Dans ce nouveau classifieur, les informations partielles de l'étiquette de classe obtenues dans le processus de classification sont utilisées de telle sorte que chaque échantillon semi-étiqueté n'affecte que les statistiques de la classe dans laquelle il a été partitionné. De plus, ce classifieur attribue un poids complet aux échantillons d'apprentissage, mais donne automatiquement un poids réduit aux échantillons semi-étiquetés. Par conséquent, il utilise les informations d'étiquettes de classes supplémentaires fournies par les échantillons semi-étiquetés correctement classés et en même temps limite l'influence indésirable des échantillons mal-classés.
L'algorithme EM est une méthode itérative pour estimer numériquement les estimations du maximum de vraisemblance (ML) des paramètres dans un modèle de mélange. Sous ce modèle, la distribution d'une observation xÎR P est donnée comme suit :

$$f(x|\Phi) = sum_{i-1}^{L} \alpha_i f_i(x|\Phi =) \tag{3.5}$$

où a 1 ,..., a L sont les probabilités à priori de classe et donc les proportions de mélange, f I est la densité de composants paramétrée par f I et L est le nombre total de composants. La densité de mélange f est alors paramétrée par F = (1 , ..., L , 1 , ... L).
Supposons que $y = (y_1, ..., y_{mi})$ sont les m_i échantillons d'apprentissage de la classe i. De

plus, il existe L classes et un total de n échantillons non étiquetés notés $x = (x_1, ..., x_n)$. L'ensemble de paramètres F contient alors toutes les probabilités, vecteurs moyens et matrices de covariance antérieurs. Supposons que 1 , ..., L soient mutuellement indépendants. L'algorithme EM peut alors être exprimé comme l'équation itérative suivante :

E-étape

$$\tau_{ij}^c = \tau_i(x_i|\phi_i^c) = \alpha_i^c f_i(x_j|\phi_i^c) / \sum_{i=1}^{L} \alpha_i^c f_i(x_j|\phi_i^c)$$

où π_{ij}^c est la probabilité postérieure que x_j appartient à la classe i

M-étape

$$\alpha_i^+ = \sum_{j=1}^{n} \tau_{ij}^c / n$$
$$\phi_i^+ \in argmax(\sum_{k=1}^{m_i} \ln(f_i(y_k|\phi_i)) + \sum_{k=1}^{n} \tau_{ik} \ln(f_i(x_k|\phi_i)))$$

3.6 Expérimentation et résultats

Nous avons mené des expérimentations sur l'ensemble du système OCR dédié à notre langue étudiée. D'abord nous avons évalué l'influence de deux propriétés qui sont la composition et la taille de police sur le corpus. Ensuite, nous avons testé le comportement du système avec et sans prétraitements.
Pour évaluer notre système proposé, nous utilisons deux métriques :

— le taux de reconnaissance qui calcule le pourcentage des caractères reconnus par rapport à la totalité des caractères ;

$$T = \frac{\sum caractres - reconnus}{\sum caractres}$$

— la matrice de confusion qui évalue les performances du système en calculant le pourcentage d'erreurs de confusion entre les classes de caractères.

$$M(x, y) = \frac{nombre - de - caractre - x - reconnu - en - tant - que - y}{200} * 100$$

3.6.1 Corpus utilisé

Dans ce travail, nous avons construit un corpus de deux manières différentes et testé l'effet de ce changement sur les résultats de reconnaissance.
Corpus1 : ce corpus est organisé comme suit : les caractères sont classés par ordre alphabétique, en majuscules suivis de minuscules, séparés par un espace. Les caractères

non-lettres (ponctuation, parenthèses, accolades, ...) sont placés à la fin de l'ensemble. Les caractères globaux sont répétés 10 fois, la fréquence d'apparition est donc la même pour chaque caractère.
Corpus 2 : il est construit à partir d'un texte extrait d'un livre écrit dans la langue étudiée. Il contient tous les caractères, y compris les majuscules, les minuscules et les non-lettres. La fréquence d'apparition est différente, élevée pour les caractères fréquemment utilisés, tels que «ṡ» et «ẓ», et égale à 5 pour les caractères rarement utilisés, tels que «Ŭ» et «Ā».
Dans le but de tester l'impact de la composition, nous avons utilisé un corpus d'apprentissage et de test, basé sur des documents de bonne qualité. Cependant, dans la suite de nos expérimentations, nous nous sommes basés sur un corpus, d'apprentissage et de test, dont les documents reflètent la réalité et par conséquent ces derniers nécessitent, généralement, un prétraitement.

3.6.2 Impact de la composition

Afin de déterminer la composition la plus adéquate, nous avons mené une première expérience. La taille de la police est fixée à 36 d'après l'expérience dans la section précédente. Les tests sont effectués sur des documents de bonne qualité. Les résultats obtenus sont présentés dans le tableau suivant :

Size (pt)	**Corpus 1**	**Corpus 2**
36	82%	92%

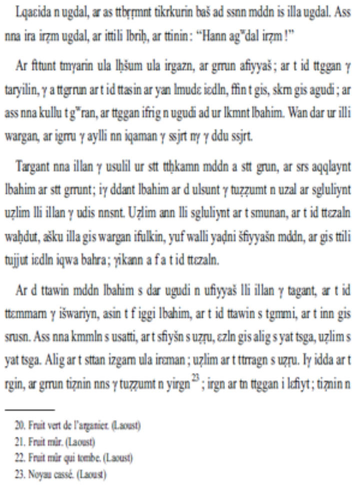
Tableau 3.2 – Taux de reconnaissance la composition du corpus

Le Corpus 2 donne de meilleurs résultats par rapport au corpus 1 dans la plupart des cas. Ce constat est dû à la présence d'une quantité d'information dans le corpus 2, qui représente un texte réel extrait d'un document, par contre le corpus 1 est une succession de caractères qui ne porte aucune information et ne respecte pas les fréquences d'apparition des caractères dans cette langue.
Par conséquent, le corpus d'apprentissage approprié à utiliser est le corpus 2 avec une taille de police de 36 pt.

3.6.3 Apport du prétraitement

Pour effectuer ces tests, nous utilisons une collection de documents. Dans un premier temps, cette collection est conservée avec une faible qualité et dans un deuxième temps,

elle a subi un prétraitement pour augmenter la qualité de l'image, afin de visualiser le comportement du système dans les deux cas. Les figures 3.11 et 3.12 montrent un exemple de ces documents.



Lqacida n ugdal, ar as ttbṛṛmnt tikrkurin baš ad ssnn mddn is illa ugdal. Ass nna ira irẓm ugdal, ar ittili lbriḥ, ar ttinin : "Hann agʷdal irẓm !"

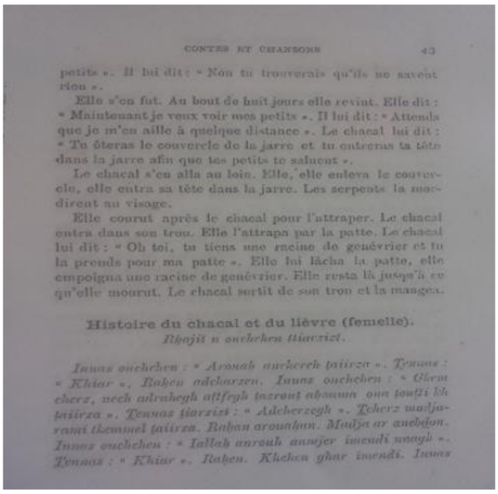
Ar fttunt tmyarin ula lḥšum ula irgazn, ar grrun afiyyaš ; ar t id ttggan γ taryilin, γ a ttgrrun ar t id ttasin ar yan lmudɛ iɛdln, ffin t gis, skrn gis agudi ; ar ass nna kullu t gʷran, ar ttggan ifrig n ugudi ad ur lkmnt lbahim. Wan dar ur illi wargan, ar igrru γ aylli nn iqaman γ ssjrt nγ γ ddu ssjrt.

Targant nna illan γ usulil ur stt tthkamn mddn a stt grun, ar srs aqqlaynt lbahim ar stt grrunt ; iγ ddant lbahim ar d ulsunt γ tuẓẓumt n uzal ar sgluliynt uẓlim lli illan γ udis nnsnt. Uẓlim ann lli sgluliynt ar t smunan, ar t id ttɛzaln waḥdut, ašku illa gis wargan ifulkin, yuf walli yaḍni šfiyyašn mddn, ar gis ttili tujjut iɛdln iqwa bahra ; γikann a f a t id ttɛzaln.

Ar d ttawin mddn lbahim s dar ugudi n ufiyyaš lli illan γ tagant, ar t id ttɛmmarn γ išwariyn, asin t f iggi lbahim, ar t id ttawin s tgmmi, ar t inn gis srusn. Ass nna kmmln s usatti, ar t sfiyšn s uẓṛu, ɛzln gis alig s yat tsga, uẓlim s yat tsga. Alig ar t sttan izgarn ula irɛman ; uẓlim ar t ttrragn s uẓṛu. Iγ idda ar t rgin, ar grrun tiẓnin nns γ tuẓẓumt n yirgn[23] ; irgn ar tn ttggan i lɛfiyt ; tiẓnin n

20. Fruit vert de l'arganier. (Laoust)
21. Fruit mûr. (Laoust)
22. Fruit mûr qui tombe. (Laoust)
23. Noyau cassé. (Laoust)

Figure 3.11 – Exemple de document de bonne qualité



CONTES ET CHANSONS 43

petits ». Il lui dit : « Non tu trouverais qu'ils ne savent rien ».

Elle s'en fut. Au bout de huit jours elle revint. Elle dit : « Maintenant je veux voir mes petits ». Il lui dit : « Attends que je m'en aille à quelque distance ». Le chacal lui dit : « Tu ôteras le couvercle de la jarre et tu entreras ta tête dans la jarre afin que tes petits te saluent ».

Le chacal s'en alla au loin. Elle, elle enleva le couvercle, elle entra sa tête dans la jarre. Les serpents la mordirent au visage.

Elle courut après le chacal pour l'attraper. Le chacal entra dans son trou. Elle l'attrapa par la patte. Le chacal lui dit : « Oh toi, tu tiens une racine de genévrier et tu la prends pour ma patte ». Elle lui lâcha la patte, elle empoigna une racine de genévrier. Elle resta là jusqu'à ce qu'elle mourut. Le chacal sortit de son trou et la mangea.

Histoire du chacal et du lièvre (femelle).

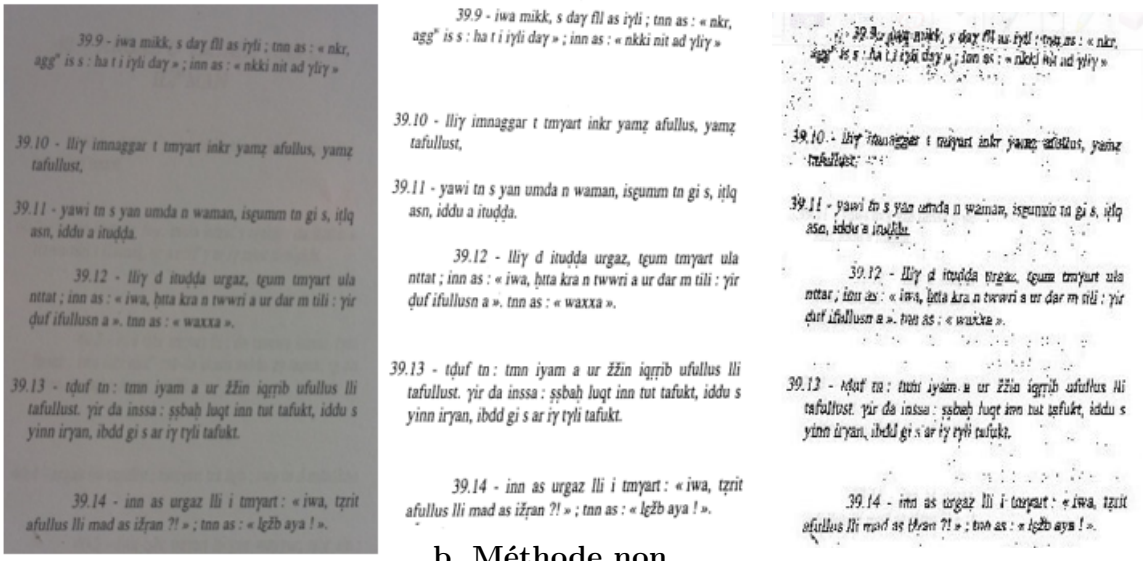
Rbajit n ouchchen ttiarzizt.

Innas ouchchen : « Aronah auchereh taiirza ». Tennas : « Khiar ». Rahen adcharzen. Innas ouchchen : « Ghem cherz, nech adrahegh atffegh tasrout abaoun oua toufti kh taiirza ». Tennas tiarzizt : « Adcherzegh ». Tcherz madjarani themmel taiirza. Rahan arouahan. Madja ar anebdou. Innas ouchchen : « Iallah anrouh annejer imendi nnagh ». Tennas : « Khiar ». Rahen. Khchen ghar imendi. Innas

Figure 3.12 – Un exemple de document de mauvaise qualité

Dans le but d'étudier l'impact de l'application des prétraitements sur les images

entrées, nous avons choisi d'appliquer deux traitements souvent utilisés surtout dans le domaine de la reconnaissance optique des caractères, il s'agit de la binarisation et la détection et correction d'inclinaison.



a. Image originale **b. Méthode non linéaire de Niblack** **c. Méthode de Sauvola**

FIGURE 3.13 – Binarisation avec différentes méthodes : a-Image originale b-Méthode non linéaire Niblack c-Méthode de Sauvola.

Dans la la phase de binarisation, nous avons essayé deux méthodes : les méthodes de binarisation non linéaire de Niblack et Sauvola. L'apport de ces méthodes est illustré sur la figure 3.13. La comparaison entre ces deux méthodes révèle que la méthode de binarisation non linéaire Niblack donne des résultats remarquables visualisés sur la figure.

Les résultats de l'utilisation de la méthode de Hough pour la détection et la correction de l'inclinaison sont présentés sur la figure suivante :

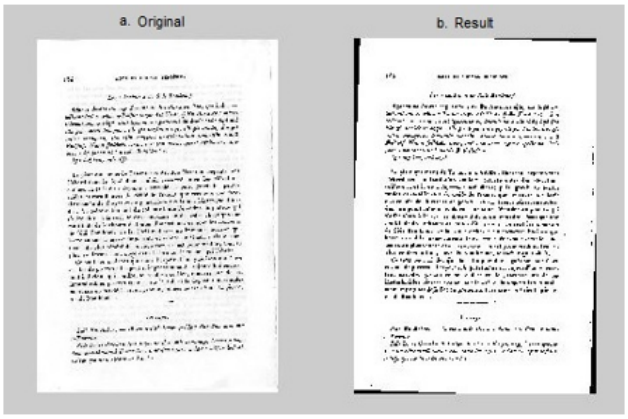


FIGURE 3.14 – Détection et correction de l'inclinaison dans un document

D'après cette figure, nous pouvons observer l'alignement des lignes de texte dans l'image corrigée. Cette correction est primordiale pour la suite du processus du système OCR, notamment la phase de segmentation.
Les résultats de la combinaison des prétraitements sont donnés sur le tableau 1.3

Image	Origine	Avec correction d'inclinaison		
		Sans binarisation	Binarisation non linéaire de Nilback	Binarisation de Sauvola
Taux de reconnaissance	24%	75%	89%	82%

TABLEAU 3.3 – Taux de Reconnaissance

Les figures et le tableau ci-dessus montrent que la reconnaissance avec prétraitement donne de meilleurs résultats par rapport au document d'origine. Ces résultats confirment l'importance du rôle de cette phase dans le processus d'un système OCR.

3.6.4 Evaluation du système

Le jeu de caractères utilisé pour écrire la langue amazighe transcrite en latin est composé de caractères latins avec des signes diacritiques.
Pour poursuivre l'évaluation de notre système, basée sur la matrice de confusion, nous avons restreint la matrice aux caractères diacritiques, étant donné le nombre élevé de caractères étudiés (115 caractères), en particulier, que la recherche sur les alphabets latins sans diacritique a connu un succès, et les taux de reconnaissance basés sur l'approximation polygonale et le classifieur adaptif sont élevés pour ces caractères.

Nous notons que ce test est effectué sur un document composé de 200 caractères pour chaque classe. La matrice représente le pourcentage de reconnaissance entre les différents caractères. À partir de la matrice de confusion, nous remarquons que des erreurs de classification sont détectées, en particulier pour les caractères diacritiques et son corps, tels que "ṭ" qui est confondu avec "t", "ḍ" avec "d" et "ḥ" avec "h".

3.7 Conclusion

Dans ce chapitre, nous avons présenté deux contributions menées dans la cadre du développement d'un système OCR pour la langue amazighe transcrite en latin.
Dans la première contribution, nous nous sommes concentrés sur la phase d'extraction des caractéristiques, qui est une phase importante dans un système OCR. Dans cette contribution, nous avons utilisé l'approximation polygonale qui sont des caractéristiques de type structurel-statistique. L'expérience a prouvé que ce type de caractéristiques donne de bons résultats même pour le cas de documents de mauvaise qualité. Dans la deuxième contribution, nous avons utilisé un classifieur adaptatif basé sur la règle du maximum de vraisemblance dans la phase de classification, la phase primordiale dans n'importe quel système OCR. Dans l'expérimentation, nous avons effectué un ensemble de méthodes de prétraitement, afin de définir la méthode la plus appropriée pour la langue étudiée. Le taux de reconnaissance pour ce système a atteint 89%.

	ā	ă	ḅ	ḍ	ě	ġ	ḡ	ḥ	ḫ	ḷ	ǒ	ṛ	ṡ	š	ṭ	ů	ŭ	ẓ	ϵ	γ
ā	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ă	2	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ḅ	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ḍ	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ě	0.5	1	0	0	98.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ġ	0	0	0	0	0	97	3	0	0	0	0	0	0	0	0	0	0	0	0	0
ḡ	0	0	0	0	0	2	98	0	0	0	0	0	0	0	0	0	0	0	0	0
ḥ	0	0	1	0	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0
ḫ	0	0	0	0	0	0	0	3	97	0	0	0	0	0	0	0	0	0	0	0
ḷ	0	0	0	0	0	0	0	0	0	97	0	2	0	0	1	0	0	0	0	0
ǒ	0	1	0	0	0	0	0	0	0	0	99	0	0	0	0	0	0	0	0	0
ṛ	0	0	0	0	0	0	0	0	0	2	0	96	0	0	0	0	0	0	0	0
ṡ	0	0	0	0	0	0	0	0	0	0	0	0	97	3	0	0	0	0	0	0
š	0	0	0	0	0	0	0	0	0	0	0	0	4	96	0	0	0	0	0	0
ṭ	0	0	0	0	0	0	0	0	0	2	0	2	0	0	96	0	0	0	0	0
ů	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0	0
ŭ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	95	0	0	0
ẓ	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	96	0	0
ϵ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
γ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

TABLEAU 3.4 – Matrice de confusion entre les caractères

Chapitre

4

OCR AMAZIGHE BASE DES RÉSEAUX DE NEURONES

Sommaire

4.1 Introduction

La langue amazighe est parlée par une partie importante de la population au Maroc. Après son officialisation en 2011, plusieurs études ont été réalisées sur cette langue. Cependant, les études existantes sur les systèmes OCR pour l'amazighe se sont concentrées sur l'écriture amazighe en alphabet Tifinaghe, d'où l'importance de développer un système d'OCR traitant l'écriture amazighe transcrite en latin. Dans ce contexte, nous avons mené une étude qui vise à développer un système de reconnaissance adapté à notre langue étudiée, basé sur la synthèse réalisée au niveau de l'état de l'art et la conclusion déduite après la comparaison des méthodes et approches utilisées dans la littérature.
Dans ce chapitre, nous décrivons, dans un premier temps, notre contribution qui consiste à développer un système OCR fondé sur les réseaux de neurones connus par leurs capacités d'apprentissage même dans les cas les plus complexes. Les deux approches de réseaux de neurones utilisées sont : le perceptron multicouche et la mémoire court terme et long terme. Dans un deuxième temps, nous présentons notre deuxième contribution. Il

s'agit de l'élaboration d'une étude comparative des approches précédemment présentées afin de déterminer, dans des conditions similaires, l'approche la plus appropriée à notre langue étudiée.
Dans le reste de ce chapitre, nous présentons dans la section 2, les approches de réseaux de neurones utilisées ainsi que les types de caractéristiques choisis pour chaque approche. Dans la section 3, nous exposons les expérimentations et les résultats obtenus et nous détaillons notre étude comparative des approches élaborées sur les systèmes OCR développés pour la langue amazighe transcrite en Latin. Dans la section 4, nous dressons une conclusion et proposons les perspectives de nos travaux de recherche.

4.2 Les méthodes neuronales

Les méthodes neuronales font partie de la famille des méthodes statistiques. Les réseaux de neurones artificiels sont des processeurs massivement parallèle-distribués, qui ont la propension naturelle à stocker des connaissances expérientielles et à les rendre disponibles pour utilisation. L'élément clé de ce paradigme est la structure originale du système de traitement de l'information. Il est composé d'un grand nombre d'éléments de traitement (neurones) hautement inter-connectés travaillant à l'unisson pour résoudre des problèmes spécifiques. Les réseaux de neurones artificiels (Artificial Neural Network, ANN) apprennent par l'exemple. Un ANN est configuré pour une application spécifique, telle que la reconnaissance de formes ou la classification de données, via un processus d'apprentissage similaire à celui des systèmes biologiques qui implique des ajustements aux connexions synaptiques existant entre les neurones. Les réseaux de neurones peuvent souvent apporter une solution simple et rapide à des problèmes très complexes et difficiles à résoudre. Il existe plusieurs techniques développées dans le cadre des méthodes neurales dont on peut citer : le perceptron multicouche et la mémoire court terme et long terme. Ces deux méthodes seront l'objet de nos contributions dans ce chapitre.

4.2.1 Le perceptron multicouche

Le perceptron multicouche (multilayer perceptron, MLP) est une architecture typique des réseaux de neurones artificiels. Il contient une série de couches, composées de neurones et de leurs connexions. Un neurone artificiel a la capacité de calculer la somme pondérée de ses entrées puis applique une fonction d'activation pour obtenir un signal qui sera transmis au neurone suivant. Le choix de ce type de réseau neuronal est basé sur sa vaste utilisation dans la littérature ainsi que son modèle polyvalent, avec un grand nombre d'applications. Les MLP sont conçus pour se rapprocher de n'importe quelle fonction continue et peuvent résoudre des problèmes qui ne sont pas linéairement séparables. Les principaux cas d'utilisation de MLP sont la classification, la reconnaissance, la prédiction et l'approximation des formes. Ils se caractérisent par leur capacité

à modéliser des fonctions complexes et leur robustesse. En outre, ils ont démontré une puissance à ignorer les entrées et le bruit non pertinents.

4.2.1.1 Architecture d'un MLP

Le perceptron multicouche (MLP) est un complément du réseau de neurones à réaction directe. Il se compose de trois types de couches : la couche d'entrée, la couche de sortie et la couche cachée, comme le montre la figure 4.1. La couche d'entrée reçoit le signal d'entrée à traiter. La tâche requise telle que la prédiction et la classification est effectuée par la couche de sortie. Un nombre arbitraire de couches cachées placées entre les couches d'entrée et de sortie est le véritable moteur de calcul du MLP. Semblable à un réseau à réaction directe, dans un MLP, les données circulent dans le sens aller de la couche d'entrée à la couche de sortie. Les neurones du MLP sont entraînés avec l'algorithme d'apprentissage de la rétro-propagation.
L'architecture du perceptron multicouche est présentée sur la figure suivante :

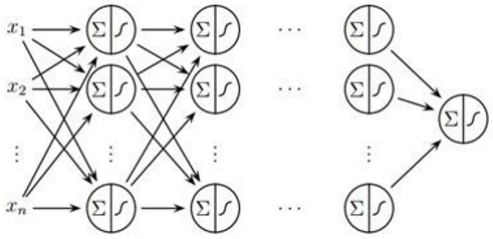


FIGURE 4.1 – L'architecture du MLP

Équation du perceptron multicouche :
Le cas le plus classique d'un réseau neuronal est le réseau avec une couche cachée unique où l'équation de mappage entre un vecteur d'entrée d et un vecteur de sortie m, est donnée par :

$$g(x) = b + Wtanh(c + V_x)$$

Où :
x est un d-vecteur (entrée)
b est une matrice k * d (appelée poids entrée-au-caché)
c est un K-vecteur (appelé décalage d'unités cachées ou biais d'unités cachés)
b est un m -vecteur (appelé unité de sortie compensées ou biais d'unités de sortie)
W est une matrice m * h (appelée poids cachés -à- sortie).
La fonction à valeur vectorielle $h(x) = tanh(c + V_x)$ est appelée la sortie de la couche cachée qui est une transformation. Une non-linéarité peut y être ajoutée dans certaines

architectures de réseau. Les éléments de la couche cachée sont appelés des unités cachées. Le type d'opération calculé par h (x) peut être appliqué sur h (x) elle-même, mais avec des paramètres différents (biais et poids différents). Cela donnerait naissance à un réseau multicouche avec deux couches cachées. Plus généralement, on peut construire un réseau neuronal profond en empilant davantage plus de couches. Chacune de ces couches peut avoir une dimension différente (k ci-dessus).

Avec leur remarquable capacité à tirer du sens de données compliquées ou imprécises, les réseaux de neurones MLP sont une excellente solution pour la classification OCR. Ils peuvent être utilisés pour extraire des modèles très complexes, difficiles à traiter par les être humains ou par d'autres techniques informatiques. Le processus OCR est caractérisé par des entrées bruyantes, une distorsion de l'image et des différences entre les polices et les tailles.

4.2.1.2 MLP en OCR

Les perceptron multi-couche a suscité un regain d'intérêt de la communauté de recherche au milieu des années 1980, car à ce moment-là, le «réseau Hopfield» offrait le moyen de comprendre la mémoire humaine et de calculer l'état d'un neurone. Initialement, la complexité informatique de la recherche de poids associés aux neurones a entravé l'application des réseaux de neurones. Avec l'avènement des architectures neuronales profondes (c'est-à-dire de nombreuses couches), comme les réseaux neuronaux récurrents (Recurrent Neural Networks, RNN) et les réseaux neuronaux convolutifs (Convolutional Neural Network, CNN), les réseaux neuronaux se sont imposés comme l'une des meilleures techniques de classification pour les tâches de reconnaissance, particulièrement pour l'OCR (Nawaz *et al.*, 2003),(Sharif et Khan, 2019).
L'implémentation précoce de MLP dans l'OCR a été réalisée par (Shamsher *et al.*, 2007) pour la langue Urdu. Les chercheurs dans (Al-Jawfi, 2009) ont proposé un algorithme de réseau de neurones à feed forward de MLP. Dans (Liu et Suen, 2009), les auteurs ont utilisé MLP sur les chiffres farsi et bangla. Une couche cachée a été utilisée avec les poids de connexion estimés par l'algorithme de rétro-propagation d'erreur (BP) qui minimisait le critère d'erreur au carré. D'autre part, les auteurs de (Cireşan *et al.*, 2010) ont formé cinq MLP avec deux à neuf couches cachées et des nombres variables des unités cachées pour la reconnaissance des chiffres anglais.

4.2.1.3 Caractéristiques utilisées avec MLP

Les réseaux de neurones Perceptron multicouches ont été appliqués à une grande variété de problèmes où la phase d'acquisition utilise un scanner ou une caméra numérique pour transformer le document texte sous forme d'image. L'image numérisée doit être une image en niveaux de gris ou une image binaire, où l'image binaire est une image en niveaux de gris étirés par contraste.

Une image, peut toujours être représentée par un ensemble de points distribués dans le plan. De cette distribution, différentes caractéristiques peuvent être déduites. Dans notre cas, le type de caractéristiques utilisé est le moment.
Ce type est très efficace pour décrire la forme des caractères. Il a été observé que ce type de caractéristiques peuvent devenir très efficaces si certaines opérations telles que la normalisation de la taille du caractère et les opérations géométriques sont effectuées correctement par l'arithmétique en virgule flottante (Singh et Sharma, 2007), (El Ayachi *et al.*, 2011).
Parmi les nombreuses familles de moments introduites dans le passé, le moment Invariant et le Moment Invariant modifié, qui sont considérés comme les moments les plus populaires et largement utilisés dans de nombreuses applications (Pithadia et Nimavat, 2015).

Le Moment Invariant :

Les moments invariants sont connus pour être invariants par translation (Ramteke, 2010), par mise à l'échelle, par rotation et par réflexion. Ils sont basés sur des mesures de la distribution des pixels autour du centre de gravité du caractère.
Soit $f(x, y)$ égale à 1 sur une région fermée et bornée R et égale à 0 sinon. Le $(p, q)^{me}$ moment est calculé comme suit :

$$m_{pq} = \iint x^p y^q f(x, y) dx dy, \text{ où } p, q = 0, 1, 2, ...$$

avec $x = m_{10} m_{00}, y = m_{01} m_{00}$
Cependant, pour l'image numérique, l'intensité continue de l'image est remplacée par une matrice où x et y sont les positions discrètes des pixels de l'image. L'intégrale précédente est donc approchée par la sommation suivante :

$$\mu_{pq} = \sum_x \sum_y P(x - \bar{x})^p (y - \bar{y})^q f(x, y)$$
$$\text{où } \bar{x} = \frac{m_{10}}{m_{00}} et \bar{y} = \frac{m_{01}}{m_{00}}$$

Le Moment Invariant modifié :

Dans le cas du moment invariant modifié, les caractéristiques sont extraites à partir du contour externe du caractère. Pour faire la différence entre les caractères qui ont le même contour externe, nous extrayons d'autres caractéristiques, par exemple C-ext, H-ext, C-int, et V-int avec :

- C-ext est le nombre des contours externes ;
- H-ext est l'histogramme horizontal, utilisé pour calculer le nombre des contours externes ;
- V-ext est l'histogramme vertical, utilisé pour calculer le nombre des contours externes ;

— C-int est le nombre des contours internes ;
— H-int est l'histogramme horizontal, utilisé pour calculer le nombre des contours internes ;
— V-int est l'histogramme vertical, utilisé pour calculer le nombre des contours internes.

La définition du moment est modifiée en utilisant les bornes du caractère seulement :

$$m_{pq} = RCx^p y^q f(x,y)ds \text{ , où } p,q = 0,1,2,...$$
$$\text{Avec } ds = p(dx^2 + dy^2)$$

Le moment central modifié peut être définie comme suit :

$$\mu_{pq} = \sum_x \sum_y P(x\bar{x})^p (y\bar{y})^q ds$$

4.2.2 La mémoire court terme et long terme

Dernièrement, la recherche dans le domaine de la reconnaissance optique des caractères s'est orienté vers une approche d'apprentissage en profondeur (Naz *et al.*, 2017), (Al-Ayyoub *et al.*, 2018) avec peu ou pas d'accent sur les caractéristiques conçues manuellement (Memon *et al.*, 2020).
Dans le cadre de notre deuxième contribution, nous avons choisi d'utiliser l'approche de la mémoire court terme et long terme (Long Short-Term Memory, LSTM), du fait que l'architecture LSTM est devenue une architecture largement utilisée en raison de ses performances dans la modélisation des dépendances de données à court et à long terme avec plus de précision par rapport aux RNN conventionnels.
Le problème de la disparition du gradient, souvent soulevé dans les RNNs est résolu par LSTM, qui essaie de ne pas imposer de biais aux observations récentes et fait en sorte que les erreurs se répètent constamment. Elle suit essentiellement le principe de l'architecture RNN, à la différence qu'elle implémente une unité de traitement interne plus élaborée appelée cellule.
La robustesse et la particularité de LSTM lui ont permis de résoudre plusieurs tâches avec un niveau de difficulté élevé. Parmi ces tâches, nous pouvons citer : la reconnaissance de l'ordre temporel d'événements très séparés dans des flux d'entrée bruyants, stockage robuste des nombres réels de haute précision sur des intervalles de temps prolongés, opérations arithmétiques sur des flux d'entrée continus, extraction d'informations véhiculées par la distance temporelle entre les événements, reconnaissance des modèles temporellement étendus dans les séquences d'entrée bruyantes, génération stable de rythmes précisément synchronisés ainsi que de trajectoires périodiques lisses et non lisses.
Le LSTM est alors capable de dépasser les autres types de RNN dans différentes tâches en termes de fiabilité et de rapidité.

4.2.2.1 Architecture de LSTM

L'architecture profonde de LSTM est aujourd'hui largement utilisée dans l'apprentissage par séquence et a démontré un grand pouvoir expérimental. Cependant, cette architecture est sujette à de nombreuses modifications. Par conséquent, de nombreuses variantes et topologies de LSTM ont été développées motivées par une analyse du flux d'erreurs dans les RNN existants.
La couche dans l'architecture LSTM est composée de cellules de mémoire, qui sont un ensemble de blocs connectés de manière récurrente. Ces blocs sont comparables aux puces de mémoire d'un ordinateur numérique. Chaque cellule de l'architecture LSTM contient trois unités multiplicatives - les portes d'entrée, les portes de sortie et les portes d'oubli - qui fournissent des analogues continus des opérations d'écriture, de lecture et de réinitialisation de la cellule. Les portes sont soit entièrement ouvertes ('O') soit entièrement fermées ('—').
Un exemple de cellule LSTM avec les unités de déclenchement est présenté dans la figure ci-dessous.

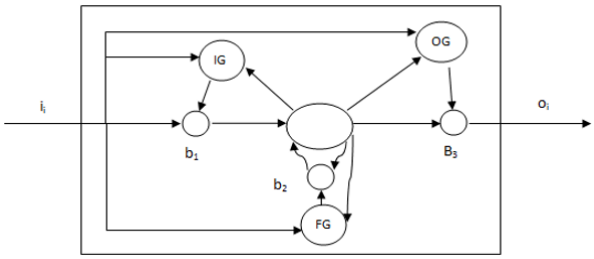


FIGURE 4.2 – Cellule de mémoire LSTM avec des unités de déclenchement.

L'unité LSTM ajoute plusieurs étapes intermédiaires : Tout d'abord, nous appliquons la fonction d'activation à i_i et nous multiplions le résultat par un facteur b_1. Ensuite, nous multiplions la valeur d'activation interne du pas de temps précédent par la quantité b_2 et nous ajoutons le résultat de la multiplication due à la connexion automatique récurrente. Enfin, nous mettons à l'échelle le résultat par b_3 et appliquons une autre fonction d'activation afin de trouver o_i. Les petits cercles sur la figure représentent les facteurs $b_1; b_2; b_3 \in (0; 1)$. Ils sont contrôlés par les cercles IG, OG et FG correspondant respectivement à l'entrée, à la sortie et à la porte d'oubli.

$$
\begin{aligned}
I &= \sigma(x_i U^i + S_{t-1} W^i) \\
F &= \sigma(x_t U^f + S_{t-1} W^f) \\
O &= \sigma(x_t u^o + S_{t-1} w^o \\
G &= tanh(x_t u^g + S_{t-1} W^g \\
c_t &= c^o_{t-1} f + g^o i
\end{aligned}
$$

où :
I : la porte d'entrée indiquant la quantité de nouvelles informations laissées à travers la cellule de mémoire ;
F : la porte d'oubli responsable de l'information et qui devrait être jetée de la cellule de mémoire ;
O :la porte de sortie indiquant la quantité d'informations à transmettre et à exposer au prochain pas de temps ;
G : auto-récurrent égal au RNN standard ;
c_t : mémoire interne de la cellule mémoire ;
s_t : état caché ;
y : sortie finale.

4.2.2.2 LSTM en OCR

Comme nous l'avons expliqué précédemment, le LSTM a été utilisé pour résoudre différents problèmes. Parmi ces problèmes, nous pouvons trouver la reconnaissance optique des caractères.
Les recherches effectuées sur le LSTM dans le domaine de la reconnaissance optique des caractères ont montré que cette architecture était capable de surmonter le problème des réseaux de neurones antérieurs pour oublier les informations précédemment acquises (Sabir *et al.*, 2017).En outre, il s'est avéré très efficace dans les tâches de reconnaissance de formes telles que la reconnaissance de l'écriture manuscrite, même dans le contexte des manuscrits médiévaux (Sundermeyer *et al.*, 2012).
Les réseaux LSTM ont été utilisés pour la reconnaissance optique de caractères (OCR), pour des documents récents ainsi que pour les documents historiques. Les excellents résultats d'OCR obtenus prouvent que LSTM est également valable et puissant sur ce type de documents (Ul-Hasan, 2016) , (Dwivedi *et al.*, 2020).

4.2.2.3 Caractéristiques utilisées avec LSTM

La phase qui précède généralement l'apprentissage du système est la phase d'extraction des caractères. Les différents groupes de caractéristiques peuvent être évalués en fonction de leur sensibilité au bruit et à la déformation et la facilité de mise en œuvre et d'utilisation.
Dans la littérature plusieurs travaux ont proposé de faire la combinaison de différentes familles de caractéristiques dans le but d'obtenir une représentation variée et complète d'une forme donnée par conséquent améliorer la discrimination entre les formes (Britto *et al.*, 2004), (Xue et Govindaraju, 2006)). L'augmentation du nombre de caractéristiques à des niveaux trop élevés implique de faire appel à des méthodes de sélection de caractéristiques (Guyon et Elisseeff, 2003), (Oliveira *et al.*, 2006).
Afin de faire apprendre le LSTM, nous devons extraire différents types de caractères. Les caractères que nous avons utilisés sont les suivants : les gradients, les points singuliers

du squelette, la présence des trous ainsi que des informations géométriques.

Les gradients

Le gradient est une quantité vectorielle comprenant des magnitudes ainsi qu'une composante directionnelle calculée en appliquant ses dérivés dans les deux directions horizontale et verticale (Aggarwal *et al.*, 2015). Le gradient d'une image peut être calculé soit en utilisant par exemple l'opérateur Sobel, Roberts ou Prewitt. La force et la direction du gradient peuvent être calculées à partir du vecteur gradient.

Pour calculer l'histogramme du gradient, nous calculons d'abord la magnitude du gradient et la direction du gradient de chaque pixel de l'image d'entrée. L'image de gradient est ensuite divisée en quelques petites cellules, et dans chaque cellule, nous générons l'histogramme du gradient dirigé en attribuant la direction du gradient de chaque pixel dans une certaine plage d'orientation qui est uniformément répartie sur 0 à 180 degrés ou 0 à 360 degrés (voir Figure ci-dessous). Les cellules d'histogramme sont ensuite normalisées avec un plus grand chevauchement des blocs connectés.

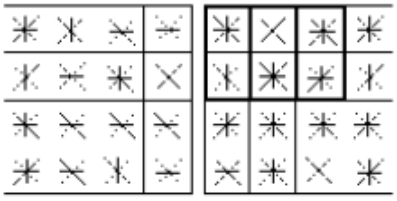


FIGURE 4.3 – Une image avec des cellules d'histogramme orientées 4x4 et des blocs descripteurs 2x2 superposés sur des cellules 2x1.

Les points singuliers du squelette

Les points singuliers des caractères tels que les extrémités, les points de croisement et les boucles sont identifiés par certaines méthodes de squelettisation. Dans une squelettisation sans pixel, les approches globales gèrent les points. Des répétitions de cette dernière peuvent avoir lieu (Rani et Kothuru, 2017).

La présence des trous

Ces caractéristiques font partie de caractéristiques topologiques extraites de la squelette de la forme. Elles reflètent la présence et le nombre de trous dans la forme (Kumar et Bhatia, 2014).

Les informations géométriques

Les informations géométriques utilisées sont des informations codées unaires, telles que l'emplacement par rapport à la ligne de base et le rapport de format d'origine, ainsi que l'inclinaison avant la correction d'inclinaison (Tawde et Kundargi, 2013).

4.3 Systèmes proposés

Dans le cadre de cette contribution nous avons proposée deux systèmes OCR basés sur l'approche de réseaux de neurones dans la phase de classification.

4.3.1 Architecture du système OCR

Le but étant d'évaluer et comparer les deux approches statistiques, nous avons proposé un système commun pour les deux approches. Le système se base sur les mêmes approches au niveau de la phase de prétraitement, d'extraction des caractéristiques et de segmentation. L'architecture du système proposé est illustrée par la figure suivante (El Gajoui et Ataa Allah, 2014).

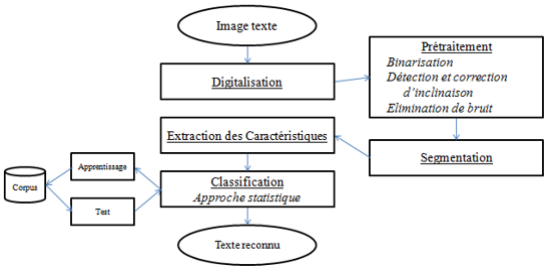


FIGURE 4.4 – L'architecture du système proposé

4.3.2 Phases du système OCR

Notre système OCR pour la langue amazighe transcrite en latin est composé de :
textitNumérisation : . Lors de cette première phase, les documents sont d'abord numérisés à l'aide d'un scanner optique. Les images résultantes sont présentées au système où elles subissent le processus de reconnaissance.
Phase de prétraitement : Dans cette phase, nous choisissons d'utiliser trois traitements (chapitre 3, section 6). La première est la binarisation, où nous avons comparé trois méthodes de binarisation afin de sélectionner la plus appropriée. La méthode choisie est la méthode de binarisation non linéaire. Le deuxième traitement est la détection et la correction de l'inclinaison. Dans ce traitement, nous avons utilisé la transformation de Hough. Cette transformation donne de bons résultats pour la détection des lignes d'écriture et l'angle de rotation du document. Le dernier traitement est l'élimination du bruit. Ce traitement est très important pour notre système, car nous traitons des documents anciens qui contiennent un bruit particulier dû à la qualité du papier. Après plusieurs tests sur différents filtres, nous avons choisi d'appliquer le filtre médian.
Phase de segmentation : Puisque notre langue a un script cursif non continu, nous

avons utilisé un histogramme horizontal pour extraire les lignes et un histogramme vertical pour extraire les caractères.
Phase d'extraction des caractéristiques : à ce niveau, nous avons extrait différentes caractéristiques telles que les gradients, les points singuliers du squelette, etc.
Phase de classification : Dans cette phase, nous avons choisi d'utiliser des approches statistiques pour la reconnaissance. Nous avons comparé deux méthodes afin de visualiser le comportement de chacune par rapport au script utilisé dans la transcription de la langue amazighe caractérisé par la présence de signes diacritiques. Les deux méthodes choisies sont le MLP et le LSTM.

4.4 Expérimentations et résultats

Dans cette section, nous présentons les différentes expérimentations et résultats des systèmes proposés, basés sur les réseaux de neurones. Puis, nous illustrons les résultats d'une comparaison entre les différentes approches utilisées pour la reconnaissance optique des caractères pour la langue amazighe transcrite en latin, à savoir : Le MLP, le LSTM et le classifieur adaptatif (CA).

4.4.1 Expérimentation MLP

4.4.1.1 Données

Pour faire l'apprentissage de notre système, nous avons créé des images contenant une ligne de texte. Le corpus d'apprentissage est composé de 10 000 images et le corpus de test de 1 000 images. L'apprentissage se déroule en plus de 20 000 itérations et donne naissance à 20 modèles de réseaux neuronaux différents. Les tests ont montré que le meilleur modèle donne un pourcentage de 97%. Ce modèle est ensuite choisi et le système est testé sur un ensemble de documents extraits de différents livres. Les documents utilisés sont 220 pages collectées dans 4 livres différents (Leguil, 2000) et (Roux, 1951) et(Justinard, 1926) et (Laoust, 1920) écrits en langue amazighe transcrite en latin.
Les tests sur cette collection se fait sur deux étapes. Dans la première étape, la collection est conservée avec une faible qualité, tandis que dans la deuxième, elle a subi un prétraitement pour augmenter la qualité de l'image afin de visualiser le comportement du système dans les deux cas. Les documents sont donc divisés en deux parties :
Doc 1 : documents de bonne qualité.
Doc 2 : documents asymétriques de faible qualité.
Les figures 4.5 et 4.6 présentent un exemple de ces documents.

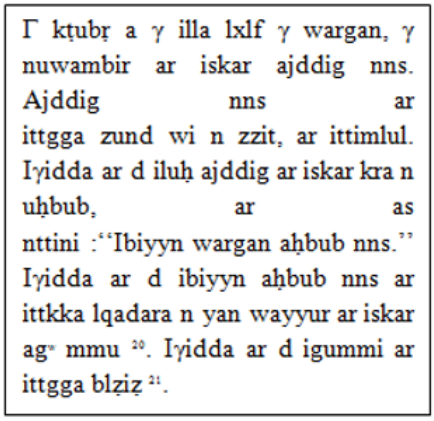

Γ kṭubṛ a ɣ illa lxlf ɣ wargan, ɣ nuwambir ar iskar ajddig nns. Ajddig nns ar ittgga zund wi n zzit, ar ittimlul. Iɣidda ar d iluḥ ajddig ar iskar kra n uḥbub, ar as nttini :"Ibiyyn wargan aḥbub nns." Iɣidda ar d ibiyyn aḥbub nns ar ittkka lqadara n yan wayyur ar iskar ag" mmu [20]. Iɣidda ar d igummi ar ittgga blẓiẓ [21].

FIGURE 4.5 – Un exemple de Doc 1

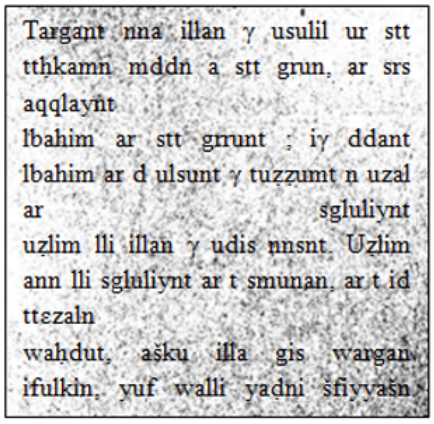

Targant nna illan ɣ usulil ur stt tthkamn mddn a stt grun, ar srs aqqlaynt
lbahim ar stt grrunt ; iɣ ddant lbahim ar d ulsunt ɣ tuẓẓumt n uzal ar sgluliynt uẓlim lli illan ɣ udis nnsnt. Uẓlim ann lli sgluliynt ar t smunan, ar t id ttɛzaln
wahdut, aśku illa gis wargan ifulkin, yuf walli yadni śfiyyaśn

FIGURE 4.6 – Un exemple de Doc2

4.4.1.2 *Résultats et analyse*

Nous testons notre système avec des documents de différentes qualités contenant du texte en amazighe transcrit en alphabet latin. Le but de ce test est d'étudier le comportement de ce système basé sur les réseaux de neurones par rapport à une langue diacritique (amazighe) contre une langue sans diacritique (anglais). D'autre part, le but est de visualiser l'impact de la variation de la qualité de l'image sur le système proposé. Les résultats obtenus sont présentés dans le tableau suivant :

Taux de Reconnaissance		Nature de la langue	
		Langue Diacritique	Langue non-diacritique
Variation de la qualité du document	Doc 1	96%	99%
	Doc 2	60%	65%

TABLEAU 4.1 – Taux de Reconnaissance

Nous notons que la reconnaissance de documents de bonne qualité (Doc 1) fournit des pourcentages importants pour une langue diacritique et sans diacritique. Le taux de reconnaissance de la langue amazighe atteint 96%. Cependant, il existe des erreurs de classification pour les exemples :

— Les majuscules sont confondues avec des minuscules dans certains cas.
— Le caractère "G" est confondu avec "G","Ů" avec "U" et "ϵ" avec "s".
— "W" n'est généralement pas reconnu.
— Les espaces sont parfois manqués.

Dans le cas des documents de faible qualité (Doc 2), le taux varie considérablement par rapport aux documents de bonne qualité. Plusieurs erreurs de reconnaissance apparaissent dans les deux cas. Ces erreurs sont généralement dues à des ruptures de caractères qui sont soit cassés, soit ils ont perdu une partie de leur corps.

Les erreurs relevées sur le langage diacritique sont :

— Caractères reconnus comme deux caractères, tels que : “ů” est reconnu en tant que “ii” ou bien “u” en tant que “r̰r”.

— Confusion entre les caractères : “d̲” et “t̲” avec “l̰”,“g” avec “ġ”,....

Les mêmes types d'erreurs se retrouvent pour les langues sans diacritique. On remarque une confusion entre “e” et “c”, “a” et “u”, “nn” et “m”, “h” et “lr”,....

4.4.2 Expérimentation LSTM

Après la création du corpus correspondant à la langue amazighe transcrite en latin (chapitre 2, section 7), nous avons évalué les corpus en utilisant l'approche LSTM. Cette dernière est connue par sa capacité de reconnaissance même dans des cas complexes, comme dans notre cas où nous traitons des documents anciens et récents. Un autre point fort de cette approche réside dans sa capacité de traitée les différents niveaux de ce corpus à savoir : la ligne, le mot et le caractère. Cette évaluation permet de tester la robustesse et la certitude du corpus construit avec ses trois types, en se reposant sur deux critères : le taux de reconnaissance et la convergence en nombre d'itérations. Cette évaluation permet aussi d'expérimenter l'approche de LSTM avec notre langue étudiée.

4.4.2.1 Expérimentations et résultats

Nous avons effectué des expériences sur chaque corpus séparé.

Corpus lignes : ce corpus contient 2450 images de texte. Pour distinguer le corpus d'apprentissage et de test, nous avons prélevé des échantillons aléatoires dans le corpus principal pour former ces deux corpus. Le corpus d'apprentissage et de test comprenant respectivement 1960 et 490 images de texte soit un pourcentage de 75% et 25%.

Afin d'étudier la convergence en termes de nombre d'itérations, nous avons configuré le système pour construire un modèle toutes les 500 itérations. Les taux d'erreur pour quelques modèles liés à ce corpus sont présentés dans le tableau ci-dessous :

Itération	500	1000	2000	2500	3500	4000	4500	5000	6000	7000	8000	10000	20000
%	65.8	28.1	17.7	15.3	12.1	11.2	10.8	10.5	9.3	6.6	4.8	4.9	4.8

Tableau 4.2 – Taux d'erreur pour le corpus de lignes.

Corpus mots : Ce corpus contient 5600 images correspondant à des images de mots. Nous avons divisé le corpus en corpus d'apprentissage et un corpus de test. Les deux corpus sont composés respectivement de 4480 et 1120 images soit 75% et 25% du corpus total.

Après plusieurs expériences, nous avons choisi de configurer le système pour créer un

modèle toutes les 1000 itérations.
Les taux d'erreur notés pour quelques modèles sont indiqués dans le tableau suivant :

Itération	500	2000	4000	6000	8000	9000	10000	11000	11500	12000	14000	20000
%	100	100	90	76	59.6	52	45	41	39.4	34	34.1	34

Tableau 4.3 – Taux d'erreur pour le corpus de mots.

Corpus caractères : ce corpus est composé d'images de caractères isolés. Il contient 12300 images. Pour former le corpus d'apprentissage et de test, nous avons choisi des exemples aléatoires du corpus principal afin d'obtenir 9225 images dans le corpus d'entraînement et 4 3075 images dans le corpus de test soit 75% et 25% du corpus total.
Les expériences réalisées pour ce corpus, nous ont conduit à choisir une étape de 10000 itérations avant la construction du modèle.
Nous notons que dans le cas des trois types de corpus, les modèles sont construits sur la base des modèles précédents et non de manière indépendante.
Les résultats des taux d'erreur pour ce corpus sont présentés dans le tableau suivant :

Itération	10000	20000	30000	40000	60000	100000	200000	250000	300000
%	100	97	96	86	69	50	39	41	39

Tableau 4.4 – Taux d'erreur pour le corpus de caractères.

4.4.2.2 Observations et analyse

Selon les expériences, le taux d'erreur du corpus lignes diminue et atteint une valeur de 9,3% après 6000 itérations et devient stable. Pour le corpus de niveau mots, le taux d'erreur suit une courbe décroissante et indique un pourcentage de 34% à partir de l'itération 12000 où il se stabilise.
Pour le corpus de niveau caractère, le taux d'erreur converge très lentement. Après 200000 itérations, le taux d'erreur atteint 39%. Nous notons que le temps pris dans chaque itération est retardé d'un corpus à un autre. Il est considérablement plus long dans le cas du corpus de lignes que dans le corpus de mots et le corpus de caractères, mais cette diversité peut être négligée compte tenu de la différence entre les nombres d'itérations. Nous pouvons remarquer, d'après les tableaux et la figure ci-après, que les corpus de lignes et de mots donnent de bons taux de reconnaissance avec une différence dans le temps de convergence, où le temps de convergence du corpus en ligne est beaucoup plus rapide que pour le corpus de mots. Cependant, le corpus de caractères reste loin de converger vers un bon taux d'erreur même après un grand nombre d'itérations.

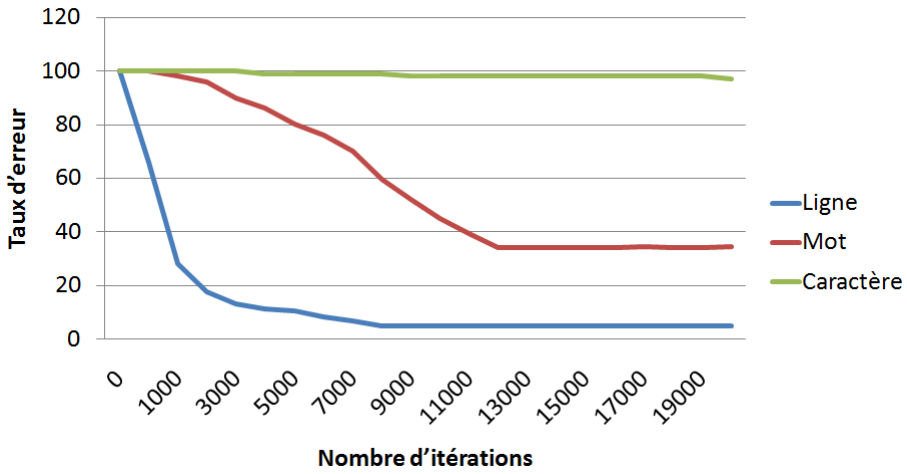


FIGURE 4.7 – Courbes des taux d'erreur des trois niveaux.

Les bons résultats donnés par le corpus de lignes et de mots peuvent être une preuve de la stabilité et de la correction des corpus construits.
A partir de ces résultats, nous pouvons facilement voir que le corpus de lignes est le plus fort comparé aux autres corpus. Par conséquent, la segmentation en ligne est la meilleure. L'avantage du corpus de lignes est dû à la quantité importante d'informations dans la ligne qui comporte une succession de caractères, ce qui explique le taux d'erreur considérable dans le cas du niveau caractère. Un autre point fort de ce corpus réside dans l'utilisation d'un système basé sur le LSTM caractérisé par une mémoire forte.
Les taux de reconnaissance observés pour ces expériences sont significativement élevés, en particulier dans le cas du corpus de ligne. Ce résultat est la preuve de la bonne construction et des performances du corpus. Comme ce corpus est le premier du genre pour la langue amazighe transcrite en latin, il peut constituer une base solide pour différents travaux effectués sur cette langue.

4.4.3 Comparaison des approches

Comme mentionné précédemment, nous avons mené, dans cette section, une comparaison des deux approches statistiques, à savoir : les réseaux de neurones et la classification adaptative (CA). L'objectif de cette comparaison est de déterminer l'approche la plus appropriée aux caractéristiques de notre langue étudiée.

4.4.3.1 Corpus exploité

D'après nos études faites au niveau de l'état de l'art, aucun des travaux antérieurs n'ayant été réalisé sur la langue amazighe transcrite en latin. Par conséquent, il n'existe

pas de corpus standard pour cette langue, d'où la nécessité d'utiliser notre corpus, dont les détails de construction sont déclinés au niveau du chapitre 2. Le corpus choisi est le niveau 'ligne', composé de corpus d'apprentissage et de test. Les deux corpus contiennent respectivement 7 000 et 3 000 images. Le corpus de validation est composé d'images de documents brutes contenant un texte écrit en langue amazighe transcrite en latin. Il est préparé sur la base de pages numérisées collectées des différents livres (Laoust, 1920), (Justinard, 1926), (Roux, 1951), (Leguil, 2000), (Stroomer, 2008). Certains livres de cette collection sont anciens et d'autres sont récents. Par conséquent, la qualité des documents diffère d'un document à l'autre selon l'état et l'ancienneté du livre. Il existe 2 types de documents :

— Doc 1 : document récent de bonne qualité
— Doc 2 : document ancien.

4.4.3.2 Comparaison MLP et LSTM

Dans nos contributions précédentes, nous avons choisi deux types d'approches basés sur les réseaux de neurones à savoir : MLP et LSTM. MLP est utilisé depuis des décennies et a pu enregistrer de bonne résultats et des taux de reconnaissance satisfaisante surtout dans le domaine de la reconnaissance optique des caractères. D'autre part, malgré sa récence, LSTM a pu démontrer sa capacité de surmonter les limites de plusieurs approches de classification.
Dans la suite, nous effectuons une comparaison entre les deux approches (MLP et LSTM) afin de déterminer la méthode la plus appropriée à notre langue étudiée.

Résultats et analyse

Les tests sont effectués sur les deux systèmes basés sur les approches MLP et LSTM en utilisant les documents Doc 1 et Doc 2 avec et sans prétraitement. Le résultat des tests est présenté sur le tableau suivant :

Taux de Reconnaissance		Approches	
		LSTM	MLP
Variation de la qualité du document	Doc original 1	95%	93%
	Doc pré-traité 1	97%	96%
	Doc original 2	60%	55%
	Doc pré-traité 2	95%	92%

Tableau 4.5 – Taux de reconnaissance de la comparaison MLP et LSTM

A partir du tableau ci dessous, nous pouvons remarquer que la classification basée sur l'approche MLP donne des taux de reconnaissance élevés surtout pour les documents pré-traités qui atteins un taux de 96% pour Doc 1. D'autre part, les taux de reconnaissance enregistrés pour le système basé sur LSTM sont aussi élevés que celui basé sur MLP, avec un avancement léger de 1% pour l'LSTM. Les expérimentations dévoilent

aussi le rôle important du prétraitement. Nous pouvons noter que les taux de reconnaissance sont considérablement importants pour les documents pré-traités par rapport aux document non pré-traités pour les deux systèmes.
Cette étude comparative prouve que les réseaux de neurones, aussi bien pour ses versions anciennes (MLP) ou récentes (LSTM), présente une solution efficace pour la classification dans notre système de reconnaissance optiques des caractères développé au profit de la langue amazighe transcrite en latin.

4.4.3.3 Comparaison LSTM et classifieur adaptatif

Pour étudier le comportement des approches de classification par rapport à une langue diacritique, avec l'exemple de la langue amazighe transcrite en latin dans notre cas, nous avons choisi deux approches : les réseaux de neurones et la classifieur adaptatif. Dans la littérature, il a été démontré que le classifieur adaptatif donne de bons résultats pour les langues diacritiques telles que le grec, l'ourdou et l'arabe. Cependant, la méthode du réseau de neurones, principalement LSTM, est connue par sa robustesse et sa capacité à s'adapter à des cas compliqués. Ainsi, nous étudierons le comportement de chacune de ces deux méthodes par rapport à notre langue étudiée afin de choisir la plus appropriée.

Apprentissage et test du modèle

La phase d'apprentissage est une étape principale des deux approches. Nous avons utilisé notre corpus créé pour apprendre les deux systèmes. Pour le système de réseau de neurones LSTM, nous avons entrepris l'apprentissage pendant plus de 30 000 itérations. Après chaque 1000 itérations, un modèle est créé sur la base des modèles précédents. Nous avons généré 20 modèles de réseaux neuronaux différents au total. Le test montre que le meilleur modèle est obtenu après 20 000 itérations et donne un taux de reconnaissance de 98%.
L'apprentissage du système basé sur le classifieur adaptatif passe par trois étapes : la génération de boîtes, la création du fichier de données formé et l'apprentissage. Le test sur le système a donné un taux de reconnaissance réussi à 96%.

Modèle d'évaluation

Pour évaluer les deux approches, nous avons utilisé le corpus de validation défini dans la section de la construction de corpus(chapitre 2, section 7). Afin d'analyser le comportement du système envers le prétraitement, nous avons exécuté le système sur des documents en deux étapes. Dans un premier temps, nous avons utilisé des documents bruts, sans aucun prétraitement. Dans la deuxième étape, les documents ont subi des prétraitements, qui sont la binarisation, la détection d'inclinaison et l'élimination du bruit, précédemment discutés dans la phase de prétraitement. Les taux de reconnaissance sont indiqués dans le tableau 6 :

Afin d'observer l'impact de chaque approche sur la reconnaissance des caractères avec et sans diacritiques, nous avons calculé le taux de classification de ces caractères pour les deux approches. Pour cela, nous avons utilisé des documents de type Doc 2 avec

Taux de Reconnaissance		Approches	
		LSTM	CA
Variation de la qualité du document	Doc original 1	95%	94%
	Doc pré-traité 1	97%	95%
	Doc original 2	60%	70%
	Doc pré-traité 2	95%	89%

TABLEAU 4.6 – Taux de reconnaissance de la comparaison LSTM et CA

prétraitement. Les résultats sont affichés sur le tableau 7.

	Nature de classifieur	
	LSTM	CA
Caractère avec diacritique	80%	71%
Caractère sans diacritique	98%	94%

TABLEAU 4.7 – Taux de reconnaissance des caractères avec et sans diacritiques pour chaque approche

Analyse des résultats

Selon le tableau 6, nous remarquons que le classifieur adaptatif donne de meilleurs résultats tout en traitant des documents anciens bruts. De plus, nous remarquons que le taux de reconnaissance diminue à mesure que la qualité du document se détériore de Doc 1 à Doc 2. Alors que l'importance du prétraitement augmente pour les deux approches, même si elle est plus efficace pour l'approche LSTM. Ainsi, l'approche du réseau neuronal donne de meilleurs résultats par rapport au classifieur adaptatif qui atteignent respectivement 95% et 89% pour les documents anciens prétraités. Dans les documents traités, le taux de reconnaissance est remarquablement faible par rapport aux documents avec traitement dans les deux approches. Plusieurs erreurs de reconnaissance apparaissent dans les deux cas. Ces erreurs sont généralement dues au bruit ou aux ruptures de caractères causées par certains traitements. Cependant, il existe des erreurs de classification erronées, telles que :

— Les majuscules et les minuscules sont parfois confuses.
— Problème de détection de l'absence ou de la présence de signes diacritiques pour certains caractères comme “G” est confusé avec “G̱”, et “Ů” avec “U”.
— ”w” n'est généralement pas reconnu.
— Les espaces sont parfois manqués.

La comparaison des taux de reconnaissance des caractères avec et sans diacritiques, pour les deux approches, montre que les erreurs de classification se font principalement dans les caractères avec diacritiques. La différence est remarquable pour les deux approches mais les réseaux de neurones LSTM sont plus adaptés à la reconnaissance de ces caractères. Les erreurs relevées sur les caractères diacritiques sont telles que :

— Les caractères reconnus comme deux caractères, tels que : “ů” est reconnu comme “ii” ou “u” comme “r̥r”.

— Confusion entre les caractères :“ḏ” et “ṯ” avec “ḷ”, “g” avec “ġ”,....
Il y a aussi quelques erreurs dans les caractères sans signes diacritiques comme la confusion entre “e” et “c”, “a” et “u”, “nn” et “m”, “h” et “lr”,....
La discrimination entre les diacritiques diffère d'une approche à l'autre. Certains de ces signes diacritiques se distinguent, mais d'autres sont confus. Le tableau 4 montre que le taux de reconnaissance des marques diacritiques est élevé pour les deux approches. Cependant, la fusion des diacritiques avec le corps du caractère empêche la reconnaissance d'atteindre 100%. Des problèmes de fusion peuvent survenir dans la phase d'acquisition, lorsque nous numérisons le document, ou dans la phase de prétraitement. Certains diacritiques sont similaires, de sorte que les risques de confusion augmentent. D'autres signes diacritiques se présentent sous la forme de caractères en exposant (écrits sous, au-dessus, à droite ou à gauche du caractère). Ces signes diacritiques sont confondus avec le caractère d'origine. La confusion du caractère avec son exposant est due à des problèmes de positionnement du caractère par rapport à la ligne de base. Ces expériences montrent que la phase de prétraitement est une phase importante pour le système OCR. Cependant, les traitements choisis doivent être adéquats à la catégorie des documents. Un traitement insuffisant peut produire des dommages dans la structure des caractères qui influencent les performances de reconnaissance. Les deux approches ont donné un pourcentage intéressant de reconnaissance, ce qui montre que l'apprentissage basé sur notre corpus construit est assez réussi. Les résultats de ces expériences prouvent que l'approche du réseau neuronal à base de LSTM, est la meilleure approche pour la classification du langage diacritique, qui est, dans notre cas, la langue amazighe transcrite en latin. Le classifieur adaptatif donne également un bon taux de reconnaissance mais l'erreur sur les caractères diacritiques est beaucoup plus importante que l'approche du réseau neuronal.
Nous notons que l'absence des travaux de recherches développées pour cette langue, a entrainé l'inexistence d'un système référentiel avec lequel on peut comparer nos systèmes.

4.5 Conclusion

Dans ce chapitre, nous avons présenté deux contributions dans le cadre du développement d'un système OCR dédié pour la langue amazighe transcrite en latin.
Dans un premier temps, nous avons exposé notre système basé sur les réseaux de neurones notamment le perceptron multicouche et la mémoire court terme et long terme dans la phase de classification. Ce type de classifieurs étant connu par leur capacité remarquable à tirer du sens des données quel que soit leur niveau de complexité ou imprécision, il a été choisi afin de l'appliquer sur notre type d'écriture présentant un défi avec la présence des diacritique et l'ancienneté des documents. Les résultats obtenus âpres les expérimentations ont été exprimés par des taux de reconnaissance qui varie entre 60% et 98% selon la nature de la langue, le corpus utilisé et la qualité du document utilisé. La deuxième contribution abordé dans ce chapitre porte sur la comparaison des systèmes précédemment proposés dans le but de déterminer le plus approprié pour

notre langue étudiée. Pour ce faire, nous avons conçu trois systèmes dont le premier est fondé sur les réseaux de neurones à base de MLP, le deuxième sur les réseaux de neurones à base de LSTM et le troisième sur le classifieur adaptif. Pour garantir la crédibilité de la comparaison, nous avons utilisé le même corpus pour l'apprentissage et le test. Les expérimentations ont montré que l'approche du réseau de neurones à base de LSTM donne de meilleurs résultats par rapport au classifieur adaptatif avec un taux de reconnaissance respectivement de 95% et 89% pour les documents anciens prétraités. Les tests ont aussi confirmé une légère supériorité de la méthode de la mémoire court terme et long terme par rapport au perceptron multicouche pour les approches basées sur les réseaux de neurones.

CONCLUSION GENERALE

Ce tapuscrit présente des contributions dans le domaine de l'analyse d'images de documents et de la reconnaissance optique de caractères (OCR), effectuées dans la cadre d'un projet de thèse. L'objectif principal de cette thèse est de développer un système OCR pour la reconnaissance de texte des documents rédigé en langue amazighe transcrite en caractère latin.
De nos jours, plusieurs systèmes ont été proposés pour différentes langues. Ces systèmes se basent sur un ensemble varié d'approches et techniques. Cependant, le traitement de ce type d'écritures n'a pas été pris en charge dans aucune des études établies auparavant.
Au cours des travaux de cette thèse, nous avons pu identifier les différents modules composants un système OCR ainsi que les approches développées pour chaque module. D'autre part, nous avons exploré notre langue étudiée à travers un ensemble de livres et les différents jeux de caractères utilisés pour cette transcription. Nous avons aussi identifié les difficultés que représente ce type d'écriture pour les documents récents mais aussi pour les documents anciens.
Ainsi, nous avons proposé trois systèmes OCR composés des modules suivants : l'acquisition, le prétraitement, la segmentation, l'extraction des caractéristiques et la classification.
Dans la phase de prétraitement, nous avons étudié la particularité des méthodes déjà utilisées dans la littérature afin de déterminer les plus appropriés par rapport à la natures des caractères exploités dans la transcription en latin de la langue amazighe. Le choix des techniques de prétraitement devait aussi prendre en considération la spécificité des documents anciens qui représentaient un obstacle concret à la reconnaissance. Au niveau de la phase d'extraction des caractéristiques, nous avons opté pour la technique de l'approximation polygonale qui a montré sa robustesse pour ce type d'écriture. Pour la phase classification, qui est la phase critique dans un système de reconnaissance optique des caractères, nous avons choisi d'utiliser deux approches connues par leur capacité de reconnaissance dans des cas très complexes. La première approche est les réseaux de neurones, représentée par les deux types MLP et LSTM et la deuxième approche basée sur le classifieur adaptatif. Pour mener les expérimentations des systèmes proposés, nous avons utilisé un corpus, dédié à la langue étudiée, que nous avons élaboré sur 3 niveaux :ligne, mot et caractère, et contient des documents de deux catégories : récents et anciens. La

confrontation des résultats des deux types de réseaux de neurones MLP et LSTM a dévoilé un avancement de la nouvelle méthode LSTM. D'autre part, les tests effectués sur le système basé sur le classifieur adaptatif ont donné des suites satisfaisantes exprimées par des taux de reconnaissance élevés. Cependant, les résultats de la comparaison entre ce système et celui basé sur l'approche des réseaux de neurones LSTM ont montré une supériorité de ce dernier avec un taux de reconnaissance qui atteint 98%.
Le travail effectué durant cette thèse ouvre plusieurs perspectives à envisager dans le futur :
A court terme :

- Enrichir le corpus par des nouvelles polices et agrandir sa taille de façon à améliorer l'apprentissage et par conséquent, accroître le taux de reconnaissance.
- Améliorer la phase de prétraitement surtout pour les documents anciens.
- Effectuer une combinaison des classifieurs utilisés afin d'augmenter la précision de la classification.

A moyen terme :

- Développer un système automatique en JAVA/PYTHON pour exploiter notre système proposé.
- Publier le corpus amélioré et le mettre à la disposition des chercheurs dans ses différents niveaux.

A long terme :

- Entamer un nouvel axe de la reconnaissance optique des caractéristiques pour la langue amazighe transcrite en arabe.
- Etudier le volet de la sécurité pour les systèmes OCR.

Liste des publications

Revues internationales

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « Classification approaches' behavior in optical character recognition systems for diacritical languages : Case of Amazigh language », ***ICIC Express Letters, Part B : Applications***, DOI : 10.24507, 763-772, VOL. 10, Issue 9, September 2019 .

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « A corpus for Amazigh transcribed to Latin OCR systems' evaluation », ***ARPN Journal of Engineering and Applied Sciences***, ISSN 1819-6608,VOL. 13, NO. 22, NOVMBER 2018.

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « Recognition of Amazigh language transcribed into Latin based on polygonal approximation », ***INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING***, VOL. 10, 2016.

Conférences internationales

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « Diacritical Language OCR Based on Neural Network : Case of Amazigh Language », ***Proceedings of International Conference on Advanced Wireless Information and Communication Technologies (AWICT'2015)***, 5-7 Octobre 2015, Sousse, Tunisie.

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « Training TESSERACT Tool for Amazigh OCR », ***Proceedings of the International Conference on Advanced Computing and Services (ACS'15)***, 20-25 Mai 2015 Konya, Turkey.

Khadija EL GAJOUI, Fadouaa ATAA ALLAH « Optical character recognition for multilingual documents : Amazigh-French », ***Proceedings of the Second World Conference on Complex Systems (WCCS'14)***, 10–12 November 2014, Agadir Morocco.

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « Vers un système de reconnaissance des caractères dans des documents multilingues », ***Proceedings of Conference : International conference Human-Machine Interaction and Image (IHMIM'14)***, 03-06 Mai 2014 Hammamet, Tunisie.

Conférences nationales

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « Système de reconnaissance optique des caractères pour la langue Amazighe », ***Journée URAC'15***, à l'Institut Scientifique Rabat le 28 novembre 2015.

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « Système de reconnaissance optique des caractères pour la langue amazighe transcrite en alphabet Latin », ***la 2eme édition des journées doctorales de la FSR*** , à la faculté des sciences de Rabat le 19, 20 et 21 Février 2015.

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « La reconnaissance optique des caractères : Etat de l'art », ***à la 3ème édition des Journées doctorales***, du 6 au 8 Février 2014. à FSR.

Khadija EL GAJOUI, Fadouaa ATAA ALLAH, Mohammed OUMSIS « La reconnaissance optique des caractères des documents multilingues », ***aux Journées Doctorales JDTIC'14***, à ENSIAS.

Bibliographie

Abe, S. (2005). *Support vector machines for pattern classification*, volume 2. Springer.

Abu-Absi, S. (2016). The arabic language. *History of Islam : An Encyclopedia of Islamic History*.

Aggarwal, A., Singh, K. et Singh, K. (2015). Use of gradient technique for extracting features from handwritten gurmukhi characters and numerals. *Procedia Computer Science*, 46 :1716–1723.

Agnaou, F., Ansar, K., Allah, F. A., Bouhjar, A. et Boulaknadel, S. (2017). L'amazighe dans les sciences du numérique : expérience de l'ircam. *In 24e Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, page 4.

Al-Ayyoub, M., Nuseir, A., Alsmearat, K., Jararweh, Y. et Gupta, B. (2018). Deep learning for arabic nlp : A survey. *Journal of computational science*, 26 :522–531.

Al-Badr, B. et Mahmoud, S. A. (1995). Survey and bibliography of arabic optical text recognition. *Signal processing*, 41(1) :49–77.

Al-Jawfi, R. (2009). Handwriting arabic character recognition lenet using neural network. *Int. Arab J. Inf. Technol.*, 6(3) :304–309.

Almuallim, H. et Yamaguchi, S. (1987). A method of recognition of arabic cursive handwriting. *IEEE transactions on pattern analysis and machine intelligence*, 5 :715–722.

AMARA, N. B. (1999). Utilisation des modèles de markov cachés planaires en reconnaissance de l'écriture arabe imprimée. *In Ecrit et multimedia (Tours, 9 septembre 1999)*, pages 5–6.

Amin, A. (1980). Hand written arabic character recognition by the irac system. *In 5th international conference on pattern recognition, Miami, Florida*, pages 729–731.

Amin, A. (1998). Off-line arabic character recognition : the state of the art. *Pattern recognition*, 31(5) :517–530.

Antonacopoulos, A. (1997). Local skew angle estimation from background space in text regions. *In Proceedings of the Fourth International Conference on Document Analysis and Recognition*, volume 2, pages 684–688. IEEE.

Arica, N. et Yarman-Vural, F. T. (2002). Optical character recognition for cursive handwriting. *IEEE transactions on pattern analysis and machine intelligence*, 24(6) :801–813.

Ataa Allah, F. et Boulaknadel, S. (2014). Amazigh verb conjugator. *In Proceedings of the 9th edition of the Language Resources and Evaluation Conference*.

Ataa Allah, F., Boulaknadel, S. et Souifi, H. (2014). Jeu d'étiquettes morphosyntaxiques de la langue amazighe. *Asinag*, 09 :171–184.

Azizi, N., Farah, N., Khadir, M. T. et Sellami, M. (2009). Arabic handwritten word recognition using classifiers selection and features extraction/selection. *Recent Advances in Intelligent Information Systems*, pages 735–742.

Bachir, F. et Ait Ben Ali, A. (2016). *Quel système d'écriture pour l'enseignement de la langue amazighe (tifinagh, arabe, latin) ?* Thèse de doctorat, Universite Mouloud Mammeri de Tizi-Ouzou.

Belaïd, A. et Cecotti, H. (2006). Reconnaissance de caractères : évaluation des performances.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Boukhris, F., Boumalk, A., Moujahid, E. H. E. et Souifi, H. (2008). *La nouvelle grammaire de l'amazighe*. Institut royal de la culture amazighe.

Boukous, A. (1995). La langue berbère : maintien et changement. *International journal of the sociology of language*, 112(1) :9–28.

Boukous, A. (2013). L'officialisation de l'amazighe enjeux et stratégies. *Asinag*, 8 :15–34.

Bouslimi, R. (2006). Système de reconnaissance hors-ligne des mots manuscrits arabe pour multi-scripteurs". *Memoire de mastère*.

Britto, A. d. S., Sabourin, R., Bortolozzi, F. et Suen, C. Y. (2004). Foreground and background information in an hmm-based method for recognition of isolated characters and numeral strings. *In Ninth International Workshop on Frontiers in Handwriting Recognition*, pages 371–376. IEEE.

Bultheel, A. (2003). Wavelets with applications in signal and image processing. *Course material University of Leuven, Belgium*.

Burrow, P. (2004). Arabic handwriting recognition. *Report of Master of Science School of Informatics, University of Edinburgh*.

Chaker, I., Benslimane, R. et Harti, M. (2011). Nouvelle approche pour la reconnaissance des caractères arabes imprimés. *Revue Méditerranéenne des Télécommunications*, 1(2).

Chang, F., Liang, K.-H., Tan, T.-M. et Hwan, W.-L. (1999). Binarization of document images using hadamard multiresolution analysis. *In Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, pages 157–160. IEEE.

Charles, P. K., Harish, V., Swathi, M. et Deepthi, C. (2012). A review on the various techniques used for optical character recognition. *International Journal of Engineering Research and Applications*, 2(1) :659–662.

Chen, J., Cao, H., Prasad, R., Bhardwaj, A. et Natarajan, P. (2010). Gabor features for offline arabic handwriting recognition. *In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 53–58.

Cheng, F. et Hsu, W. (1985). A new parallel thinning algorithm for binary image. *In 1985 National Computer Symposium, Kaohsiung, Taiwan, Dec. 1985*. Institute of Electrical and Electronics Engineers.

Cheriet, M., Kharma, N., Liu, C.-L. et Suen, C. (2007). *Character recognition systems : a guide for students and practitioners*. John Wiley & Sons.

Cireşan, D. C., Meier, U., Gambardella, L. M. et Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12) :3207–3220.

Ciresan, D. C., Meier, U., Gambardella, L. M. et Schmidhuber, J. (2011). Convolutional neural network committees for handwritten character classification. *In 2011 International Conference on Document Analysis and Recognition*, pages 1135–1139. IEEE.

Crawford, D. et Hoffman, K. E. (2000). Essentially amazigh : urban berbers and the global village. *The Arab-African and Islamic Worlds : Interdisciplinary Studies*, 119.

Das, A. et Mohanty, M. N. (2020). Use of deep neural network for optical character recognition. *In Advancements in Computer Vision Applications in Intelligent Systems and Multimedia Technologies*, pages 219–254. IGI Global.

Deepa, A., Rao, R. R. *et al.* (2014). Feature extraction techniques for recognition of malayalam handwritten characters. *Int. J. Adv. Trends Comput. Sci. Eng. IJATCSE*, 3 :481–485.

Ding, X. et Liu, H. (2006). Segmentation-driven offline handwritten chinese and arabic script recognition. *In Summit on Arabic and Chinese Handwriting Recognition*, pages 196–217. Springer.

Duda, R. O., Hart, P. E. et Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

Dwivedi, A., Saluja, R. et Kiran Sarvadevabhatla, R. (2020). An ocr for classical indic documents containing arbitrarily long words. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 560–561.

Eikvil, L. (1993). Optical character recognition. *citeseer. ist. psu. edu/142042. html*.

El Ayachi, R., Fakir, M., Bouikhalene, B. et Mori, M. (2011). Recognition of tifinaghe characters using dynamic programming & neural network. *Recent Advances in Document Recognition and Understanding*, page 35.

El Gajoui, K. et Ataa Allah, F. (2014). Optical character recognition for multilingual documents : Amazigh-french. *In 2014 Second World Conference on Complex Systems (WCCS)*, pages 84–89. IEEE.

EL Gajoui, K., Ataa Allah, F. et Oumsis, M. (2015a). Diacritical language ocr based on neural network : Case of amazigh language. *Procedia computer science*, 73 :298–305.

EL Gajoui, K., ATAA ALLAH, F. et OUMSIS, M. (2015b). Training tesseract tool for amazigh ocr. *In Recent Researches in Applied Computer Science : Proceedings of the 15 th International Conference on Applied Computer Science*, pages 20–22.

El Gajoui, K., Ataa Allah, F. et Oumsis, M. (2016). Recognition of amazigh language transcribed into latin based on polygonal approximation. *International journal of circuits, systems and signal processing*, 10.

El-Hajj, R., Likforman-Sulem, L. et Mokbel, C. (2005). Arabic handwriting recognition using baseline dependant features and hidden markov modeling. *In*

Eighth International Conference on Document Analysis and Recognition (ICDAR'05), pages 893–897. IEEE.

El Maadani, S. (2014). Le nouveau tifinagh un alphabet disparu sauvera-t-il les langues et cultures berbères ? *Coordonné par Nelly Carpentier*, page 25.

El Yachi, R., Moro, K., Fakir, M., Bouikhalene, B., PETER, S. J., REDDI, K. K., RAO, T. R. K., RAO, G. N., SHARAFI, S. M., ESMAEILY, H. R. *et al.* (2010). On the recognition of tifinaghe scripts. *Journal of Theoretical and Applied Information Technology*, 20(2) :61–66.

Elbaati, A., Kherallah, M., Alimi, A. M., Ennaji, A. et Sfax, T. (2006). De l'hors-ligne vers un systeme de reconnaissance en-ligne : Applicationa la modélisation de l'écriture arabe manuscrite ancienne. *Semaine du document numrique, SDN (ANAGRAM).*

Ennaji, M. (2005). *Multilingualism, cultural identity, and education in Morocco.* Springer Science & Business Media.

Es Saady, Y. (2012). *Contribution au développement d'approches de reconnaissance automatique de caractères imprimés et manuscrits, de textes et de documents Amazighes.* Thèse de doctorat, Université Ibnou Zohr, Faculté des Sciences, Agadir.

Farooq, F., Govindaraju, V. et Perrone, M. (2005). Pre-processing methods for handwritten arabic documents. *In Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 267–271. IEEE.

Fink, G. A. (2014). *Markov models for pattern recognition : from theory to applications.* Springer Science & Business Media.

Gabarra, E. (2008). *De la binarisation de documents vers la reconnaissance de symboles dans l'analyse de schémas électriques.* Thèse de doctorat, Pau.

Gabarra, E. et Tabbone, A. (2005). Combining global and local threshold to binarize document of images. *In Iberian Conference on Pattern Recognition and Image Analysis*, pages 371–378. Springer.

Gattawar, A., Vanwadi, S., Pawar, J., Dhore, P. et Mhaske, H. (2021). Automatic number plate recognition using yolo for indian conditions. *International Research Journal of Engineering and Technology.*

Gupta, D. et Bag, S. (2019). Handwritten multilingual word segmentation using polygonal approximation of digital curves for indian languages. *Multimedia Tools and Applications*, 78(14) :19361–19386.

Guyon, I. et Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar) :1157–1182.

Haji, M. M. (2005). Farsi handwritten word recognition using continuous hidden markov models and structural features. *Iran : MSC, Compter Engineering Shiraz Universitty Shiraz*.

Hamad, K. A. et Kaya, M. (2016). A detailed analysis of optical character recognition technology. *International Journal of Applied Mathematics, Electronics and Computers*, 4(1) :244–249.

Haralick, R. M., Sternberg, S. R. et Zhuang, X. (1987). Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, 9(4) :532–550.

Hassin, A. H., Tang, X.-L., Liu, J.-F. et Zhao, W. (2004). Printed arabic character recognition using hmm. *Journal of Computer Science and Technology*, 19(4) :538–543.

Heijmans, H. J. et Ronse, C. (1990). The algebraic basis of mathematical morphology i. dilations and erosions. *Computer Vision, Graphics, and Image Processing*, 50(3) :245–295.

Hussain, S., Niazi, A., Anjum, U., Irfan, F. *et al.* (2014). Adapting tesseract for complex scripts : an example for urdu nastalique. *In 2014 11th IAPR International Workshop on Document Analysis Systems*, pages 191–195. IEEE.

Imane, T. (2016). *Traduction automatique de la langue amazigh*. Thèse de doctorat, Mohamed V University, faculty of sciences.

Islam, N., Islam, Z. et Noor, N. (2017). A survey on optical character recognition system. *arXiv preprint arXiv :1710.05703*.

Jain, A. K., Duin, R. P. W. et Mao, J. (2000). Statistical pattern recognition : A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1) :4–37.

Jain, A. K., Mao, J. et Mohiuddin, K. M. (1996). Artificial neural networks : A tutorial. *Computer*, 29(3) :31–44.

Jin Soo Noh et Kang Hyeon Rhee (2005). Palmprint identification algorithm using hu invariant moments and otsu binarization. *In Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05)*, pages 94–99.

Justinard, L. V. (1926). *Manuel de berbère marocain :(dialecte rifain)*. Geuthner.

Kapoor, R., Bagai, D. et Kamal, T. S. (2002). Representation, extraction of nodal features of devnagri letters. *In ICVGIP*.

Kaur, K. et Bathla, A. K. (2015). A review on segmentation of touching and broken characters for handwritten gurmukhi script. *International Journal of Computer Applications*, 120(18).

Kawatani, T. (1993). Handprinted numeral recognition with the learning quadratic discriminant function. *In Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 14–17. IEEE.

Kefali, A., Sari, T. et Sellami, M. (2010). Evaluation of several binarization techniques for old arabic documents images. *In The First International Symposium on Modeling and Implementing Complex Systems MISC*, volume 1, pages 88–99.

Kessentini, Y., Paquet, T. et Hamadou, A. B. (2010). Off-line handwritten word recognition using multi-stream hidden markov models. *Pattern Recognition Letters*, 31(1) :60–70.

Kharate, R., Jagade, S. et Holambe, S. N. (2013). A brief review and survey of segmentation for character recognition. *International Journal of Engineering Sciences*, 2(1) :14–17.

Kholladi, M. M.-K. (2013). *Combinaison de classifieurs pour la reconnaissance de mots arabes manuscrits*. Thèse de doctorat, Université 20 aout 1955 de Skikda.

Khorsheed, M. S. (2003). Recognising handwritten arabic manuscripts using a single hidden markov model. *Pattern Recognition Letters*, 24(14) :2235–2242.

Kim, H., Ahn, E., Cho, S., Shin, M. et Sim, S.-H. (2017). Comparative analysis of image binarization methods for crack identification in concrete structures. *Cement and Concrete Research*, 99 :53–61.

Kopparapu, S. K. et Desai, U. B. (2001). *Bayesian approach to image interpretation*, volume 616. Springer Science & Business Media.

Kruatrachue, B. et Suthaphan, P. (2001). A fast and efficient method for document segmentation for ocr. *In Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology. TENCON 2001 (Cat. No. 01CH37239)*, volume 1, pages 381–383. IEEE.

Kumar, G. et Bhatia, P. K. (2014). A detailed review of feature extraction in image processing systems. *In 2014 Fourth international conference on advanced computing & communication technologies*, pages 5–12. IEEE.

Kumar, G., Bhatia, P. K. et Banger, I. (2013). Analytical review of preprocessing techniques for offline handwritten character recognition. *International Journal of Advances in Engineering Sciences*, 3(3) :14–22.

Kuncheva, L. (2004). Combining pattern classifiers methods and algorithms. john wiley&sons. *Inc. Publication, Hoboken*.

Laique, S. N., Hayat, U., Sarvepalli, S., Vaughn, B., Ibrahim, M., McMichael, J., Qaiser, K. N., Burke, C., Bhatt, A., Rhodes, C. *et al.* (2020). Application of optical character recognition with natural language processing for large-scale quality metric data extraction in colonoscopy reports. *Gastrointestinal Endoscopy*.

Lakshana, M. et Amudha, L. (2021). Handwritten recognition using deep convolution neural network. *International Journal for Modern Trends in Science and Technology*.

Laoust, E. (1920). *Mots et choses berbères : notes de linguistique et d'ethnographie : dialectes du Maroc*. Société marocaine d'édition.

Lasri, B. (2008). Ijawwan n tayri,(roman). *Marrakech, Imp Imal*.

Latfi, F., Nader, F. et Mouldi, B. (2006). Arabic word recognition by using fuzzy classifier. *Journal of Applied Scinces*, 3 :617–650.

Lebourgeois, F. (1992). *Approche mixte pour la reconnaissance des documents imprimes*. Thèse de doctorat, INSA Lyon, France.

Lee, J. et Hoppel, K. (1989). Noise modeling and estimation of remotely-sensed images. *In 12th Canadian Symposium on Remote Sensing Geoscience and Remote Sensing Symposium,*, volume 2, pages 1005–1008. IEEE.

Leguil, A. (2000). *Contes berbères grivois du Haut-Atlas*. Editions L'Harmattan.

Liu, C.-L. et Fujisawa, H. (2008). Classification and learning methods for character recognition : Advances and remaining problems. *In Machine learning in document analysis and recognition*, pages 139–161. Springer.

Liu, C.-L. et Suen, C. Y. (2009). A new benchmark on the recognition of handwritten bangla and farsi numeral characters. *Pattern Recognition*, 42(12) :3287–3295.

Lorigo, L. et Govindaraju, V. (2005). Segmentation and pre-recognition of arabic handwriting. *In Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 605–609. IEEE.

Mao, S., Rosenfeld, A. et Kanungo, T. (2003). Document structure analysis algorithms : a literature survey. *In Document Recognition and Retrieval X*, volume 5010, pages 197–207. International Society for Optics and Photonics.

Marosi, I. (2007). Industrial ocr approaches : architecture, algorithms, and adaptation techniques. *In Document Recognition and Retrieval XIV*, volume 6500, page 650002. International Society for Optics and Photonics.

Martinek, J., Lenc, L. et Král, P. (2020). Building an efficient ocr system for historical documents with little training data. *Neural Computer and Application*.

Memon, J., Sami, M., Khan, R. A. et Uddin, M. (2020). Handwritten optical character recognition (ocr) : A comprehensive systematic literature review (slr). *IEEE Access*, 8 :142642–142668.

Miled, H. (1998). *Stratégies de résolution en reconnaissance de l'écriture semi-cursive : Application aux mots manuscrits arabes*. Thèse de doctorat, Rouen.

Mori, S., Nishida, H. et Yamada, H. (1999). *Optical character recognition*. John Wiley & Sons, Inc.

Morita, M. E. (2003). *Automatic recognition of handwritten dates on brazilian bank cheques*. Thèse de doctorat, École de technologie supérieure.

Motawa, D., Amin, A. et Sabourin, R. (1997). Segmentation of arabic cursive script. *In Proceedings of the fourth international conference on document analysis and recognition*, volume 2, pages 625–628. IEEE.

Muaz, A. (2010). Urdu optical character recognition system. *Unpublished, MS Thesis Report, NUCES, Lahore, Pakistan*.

Nagy, G. (1992). At the frontiers of ocr. *Proceedings of the IEEE*, 80(7) :1093–1100.

Nagy, G., Seth, S. C. et Stoddard, S. D. (1985). Document analysis with an expert system. *In Pattern recognition in practice II*, pages 149–155.

Nawaz, S. N., Sarfraz, M., Zidouri, A. et Al-Khatib, W. G. (2003). An approach to offline arabic character recognition using neural networks. *In 10th IEEE International Conference on Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003*, volume 3, pages 1328–1331. IEEE.

Naz, S., Umar, A. I., Ahmad, R., Siddiqi, I., Ahmed, S. B., Razzak, M. I. et Shafait, F. (2017). Urdu nastaliq recognition using convolutional–recursive deep learning. *Neurocomputing*, 243 :80–87.

Ngo, P. (2019). A discrete approach for polygonal approximation of irregular noise contours. *In International Conference on Computer Analysis of Images and Patterns*, pages 433–446. Springer.

Nguyen, K. C. et Nakagawa, M. (2016). Text-line and character segmentation for offline recognition of handwritten japanese text. *IEICE technical report*, 115(517) :53–58.

Niblack, W. (1990). *An introduction to digital image processing*. Prentice-Hall, Inc.

Nick, W. (2012). Training tesseract for ancient greek ocr. *Google Inc "eutypon28-29*.

Noor, N. A. et Habib, S. (2005). *Bangla optical character recognition*. Thèse de doctorat, School of Engineering and Computer Science (SECS), BRAC University.

Oliveira, L. S., Morita, M. et Sabourin, R. (2006). Feature selection for ensembles applied to handwriting recognition. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(4) :262–279.

Olivier, G., Miled, H., Romeo, K. et Lecourtier, Y. (1996). Segmentation and coding of arabic handwritten words. *In Proceedings of 13th International Conference on Pattern Recognition*, volume 3, pages 264–268. IEEE.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1) :62–66.

Patel, C., Patel, A. et Shah, D. (2013). A review of character segmentation methods. *International Journal of Current Engineering and Technology*, 3(5) : 2075–2078.

Pechwitz, M. et Margner, V. (2002). Baseline estimation for arabic handwritten words. *In Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 479–484. IEEE.

Pithadia, N. J. et Nimavat, D. V. D. (2015). A review on feature extraction techniques for optical character recognition. *Int. J. Innov. Res. Comput. Commun. Eng*, 3.

Pouessel, S. (2008). Écrire la langue berbère au royaume de mohamed vi. les enjeux politiques et identitaires du tifinagh au maroc. *Revue des mondes musulmans et de la Méditerranée*, 124 :219–239.

Prajapati, P. K. (2019). *Optical character recognition (OCR) feature extraction and classification*. Thèse de doctorat, Dhirubhai Ambani Institute of Information and Communication Technology.

Ramteke, R. (2010). Invariant moments based feature extraction for handwritten devanagari vowels recognition. *Int. J. Comput. Appl*, 1(18) :1–5.

Rani, T. J. et Kothuru, M. (2017). Personal identification using quality image resulting from binarization and thinning techniques. *International Journal of Advanced Scientific and Technical Research*, 7(5) :70–82.

Rath, T. M., Lavrenko, V. et Manmatha, R. (2003). A statistical approach to retrieving historical manuscript images without recognition. Rapport technique, Space and Naval Warfare Systems Center San Diego CA.

Romeo-Pakker, K. et Ameur, A. (1993). Une mèthode rapide de segmentation et de reconnaissance de caractères manuscrits arabes. *In 14° Colloque sur le traitement du signal et des images, FRA, 1993*. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images.

Roux, A. (1951). Choix de version berbères parler du sud-ouest marocaine.

Rowan, K. (2006). Meroitic-an afroasiatic language ? *SOAS Working Papers in Linguistics*, 14 :169–206.

Sabbour, N. et Shafait, F. (2013). A segmentation-free approach to arabic and urdu ocr. *In Document Recognition and Retrieval XX*, volume 8658, page 86580N. International Society for Optics and Photonics.

Sabir, E., Rawls, S. et Natarajan, P. (2017). Implicit language model in lstm for ocr. *In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 7, pages 27–31. IEEE.

Sari, T., Souici, L. et Sellami, M. (2002). Off-line handwritten arabic character segmentation algorithm : Acsa. *In Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 452–457. IEEE.

Sauvola, J. et Pietikäinen, M. (2000). Adaptive document image binarization. *Pattern recognition*, 33(2) :225–236.

Shah, H., Lomte, V., Nale, P., Panchpor, S. et Agrawal, T. (2021). Review on segmentation and recognition methodologies for ocr system of devanagari script. *International Journal of Advances in Engineering Research*.

Shamsher, I., Ahmad, Z., Orakzai, J. K. et Adnan, A. (2007). Ocr for printed urdu script using feed forward neural network. *In Proceedings of World Academy of Science, Engineering and Technology*, volume 23, pages 172–175.

Sharif, H. et Khan, R. A. (2019). A novel framework for automatic detection of autism : A study on corpus callosum and intracranial brain volume. *arXiv preprint arXiv :1903.11323*.

Singh, H. et Sharma, R. (2007). Moment in online handwritten character recognition. *In National Conference on Challenges & Opportunities in Information Technology (COIT-2007)*.

Singhal, A. *et al.* (2019). A review on optical character recognition. *IITM Journal of Management and IT*, 10(1) :15–19.

Skounti, A., Lemjidi, A. *et al.* (2003). *Tirra : aux origines de l'écriture au Maroc*, volume 1. Institut royal de la culture amazighe.

Smith, R. (2007). An overview of the tesseract ocr engine. *In Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE.

Srivastava, S., Priyadarshini, J., Gopal, S., Gupta, S. et Dayal, H. S. (2019). Optical character recognition on bank cheques using 2d convolution neural network. *In Applications of Artificial Intelligence Techniques in Engineering*, pages 589–596. Springer.

Stroomer, H. (2001). *An Anthology of Tashelhiyt Berber Folktales (South Morocco)*. Rüdiger Köppe.

Stroomer, H. (2008). The argan tree and its tashelhiyt berber lexicon. *Études et documents berbères*, 1 :107–121.

Sundermeyer, M., Schlüter, R. et Ney, H. (2012). Lstm neural networks for language modeling. *In Thirteenth annual conference of the international speech communication association*.

Syiam, M., Nazmy, T., Fahmy, A. E., Fathi, H. et Ali, K. (2006). Histogram clustering and hybrid classifier for handwritten arabic characters recognition. *In SPPRA*, pages 44–49.

Tabbone, S., Nguyen, T. O. et Masini, G. (2006). Une méthode de binarisation hiérarchique floue. *In Colloque International Francophone sur l'Écrit et le Document - CIFED2006*.

Tabbone, S. et Wendling, L. (2003). Multi-scale binarization of images. *Pattern Recognition Letters*, 24(1-3) :403–411.

Tang, Y. Y., Lee, S.-W. et Suen, C. Y. (1996). Automatic document processing : a survey. *Pattern recognition*, 29(12) :1931–1952.

Tao, W.-B., Tian, J.-W. et Liu, J. (2003). Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. *Pattern Recognition Letters*, 24(16) :3069–3078.

Tawde, G. Y. et Kundargi, J. (2013). An overview of feature extraction techniques in ocr for indian scripts focused on offline handwriting. *International Journal of Engineering Research and Applications*, 3(1) :919–926.

Ul-Hasan, A. (2016). *Generic text recognition using long short-term memory networks*. Thèse de doctorat, Department of Computer ScienceUniversity of Kaiserslautern.

Vautrot, P. (1996). *Segmentation et classification d'images texturees par filtrage spatio-frequentiel : ondelettes splines et filtres de gabor*. Thèse de doctorat, Reims.

Viard-Gaudin, C. (2007). *Reconnaissance d'écriture manuscrite hors-ligne par reconstruction de l'ordre du tracé en vue de l'indexation de documents d'archives*. Thèse de doctorat, Institut de Recherche en Informatique et Systèmes Aléatoires.

Wahl, F. M., Wong, K. Y. et Casey, R. G. (1982). Block segmentation and text extraction in mixed text/image documents. *Computer graphics and image processing*, 20(4) :375–390.

Widrow, B. et Hoff, M. E. (1960). Adaptive switching circuits. Rapport technique, Stanford Univ Ca Stanford Electronics Labs.

Wilson, J. N. et Ritter, G. X. (2000). *Handbook of computer vision algorithms in image algebra*. CRC press.

Wshah, S., Shi, Z. et Govindaraju, V. (2009). Segmentation of arabic handwriting based on both contour and skeleton segmentation. *In 2009 10th International Conference on Document Analysis and Recognition*, pages 793–797. IEEE.

Xiu, P., Peng, L. et Ding, X. (2006). Multi-queue merging scheme and its application in arabic script segmentation. *In Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, pages 6–pp. IEEE.

Xue, H. et Govindaraju, V. (2006). Hidden markov models combining discrete symbols and continuous attributes in handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(3) :458–462.

Yasser, A. (2010). Preprocessing techniques in character recognition. *Intech. doi*, 10 :9776.

Yin, P.-Y. (2008). *Pattern Recognition : Techniques, Technology and Applications*. BoD–Books on Demand.

Zahour, A., Taconet, B. et Ramdane, S. (2004). Contribution à la segmentation de textes manuscrits anciens. *In Conférence Internationale Francophone sur l'Ecrit et le Document (CIFED 04)*.

Zenkouar, L. (2004). L'écriture amazighe tifinaghe et unicode. *Etudes et documents berbères. Paris (France)*, 22 :175–192.

Zhang, G., Ma, Z., Tong, Q., He, Y. et Zhao, T. (2008). Shape feature extraction using fourier descriptors with brightness in content-based medical image retrieval. *In 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 71–74. IEEE.

Zhang, Z. et Tan, C. L. (2001). Restoration of images scanned from thick bound documents. *In Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 1, pages 1074–1077. IEEE.