

République Algérienne Démocratique et Populaire

Université A. MIRA de Béjaïa

Faculté des Sciences Exactes

Département de Recherche Opérationnelle

Mémoire présenté pour l'obtention du diplôme de master



Spécialité : Sciences de Données et Aide à la Décision

Reconnaissance Automatique des Caractères Tamahaqt en Langue Amazighe

Présenté par :

Djoughira Nada

Ben Haddadi Chahinaz

Encadrant : Dr A.Zaidi

Co-Encadrant : Dr L.Asli

Défendu le 01/07/2024, devant le jury composé de :

Mr K.ABASS

Professeur

Président de jury

UAMB - Bejaia.

Mr L.IDIR

M.A.A

Examineur

ENS Assia Djebar - Constantine.

Mme S.AMROUN

M.C.D

Examineur

UAMB - Bejaia.

Année Universitaire 2023 – 2024

Remerciements

Avant de commencer la présentation de ce travail, Nous tenons à remercier en premier notre DIEU tout puissant de nous avoir donné le courage, la patience, la foi et la volonté pour pouvoir continuer et réaliser ce travail.

Nous tenons tous d'abord à remercier nos parents pour leurs soutiens et leurs encouragements durant nos années d'études.

Nous remercions sincèrement les membres du jury : Monsieur K. Abass, notre président du jury ; notre examinateur, Monsieur L. Idir ; et notre examinatrice, Madame S. Amroun, de nous faire l'honneur d'évaluer ce travail avec leur expertise et leur attention. Votre contribution est précieuse pour la qualité et la pertinence de notre projet.

Nous remercions Monsieur Zaidi Ali d'avoir accepté de nous encadrer pour notre projet de fin d'études, et à toute l'équipe du Centre de Recherche en Langue et Culture Amazighes pour leur accueil durant toute la période du stage.

Un grand remerciement spécial à notre éminent et respecté docteur Asli Larbi, qui a mis à notre disposition tous les outils nécessaires pour nous aider à réaliser notre projet, ainsi que pour son soutien, ses remarques pertinentes et son encouragement.

Nous profitons de l'occasion pour remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'études.

Djoughira Nada

Ben Haddadi Chahinaz

Dedicace

À mes parents, ma mère et mon père, pour leur patience, leur amour, leur soutien et leur encouragement. Que Allah vous garde toujours près de moi.

À mon cher mari, Yehia, qui m'a accompagné tout au long de ce parcours et qui m'encourage le plus dans ma vie.

À mes chères sœurs, Melissa et Ilham, qui sont ma fierté.

À mon adorable petit frère, Oussama.

À ma famille et à ma belle-famille.

À mes amis, Aniece et Samiha, avec qui j'ai vécu les plus belles expériences de ma vie.

À ma chère amie et binôme, Chahinaz, ainsi qu'à toute sa famille.

À tous ceux qui ont toujours cru en moi et m'ont poussé à réussir.

À toute ma famille Djouhira.

Je vous dédie ce travail.

Nada

Dedicace

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que nos parents méritent pour tous les sacrifices qu'ils n'ont cessé de nous offrir.

Pour ma chère maman, qui a tout fait pour que nous réussissions dans nos études et dans notre vie.

À l'âme de mon père, qui même s'il n'est plus parmi nous, restera toujours dans mon cœur.

Je tiens à rappeler qu'aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler. Ce travail est le fruit de leurs sacrifices.

Je dédie aussi ce modeste travail à : mon frère Azedine et ma sœur Lahna, ainsi que mes amis qui m'ont aidée, que ce soit moralement ou autrement.

À ma meilleure amie Djidji, sans oublier Asma, Tina, Hadda, Sarah, Nihad, et tous mes oncles. Merci à tous pour votre soutien indéfectible.

À ma chère amie et binôme Nada, ainsi qu'à toute sa famille.

Chahinaz

Table des matières

Remerciements	I
Dedicace	II
Dedicace	III
Liste des figures	VII
Liste des tables	VIII
Liste d'abréviations et notations	IX
Introduction générale	1
1 Préliminaire sur l'intelligence artificielle	3
Introduction	3
1.1 L'intelligence artificielle	4
1.2 Une brève Histoire de l'IA	4
1.3 Sous ensembles de l'IA	6
1.3.1 Machine learnig (ML)	6
1.3.2 Réseaux de neurones	9
1.3.3 Deep Learnig (DL)	12
1.3.4 Traitement du langage naturel(NLP)	15
1.4 Reconnaissance optique de caractères(OCR)	17
1.4.1 Fonctionnement de l'OCR	17
1.4.2 L'utilisation d'OCR	19
Conclusion	20
2 Le dialecte Tamahaqt	21
Introduction	21
2.1 La langue Amazighe	22
2.2 L'origine et la géographie de la langue Amazighe	22
2.3 Les variantes de la langue Amazighe	24
2.4 Le parlé Tamahaqt	24
2.5 L'écriture du dialecte Tamahaqt	25
2.5.1 Les verbes	27
2.5.2 Les noms	27
2.5.3 Les pronoms	27
2.5.4 Les nombres	28
2.6 Présentation d'organisme d'accueil (CRLCA)	28
2.6.1 Les divisions du centre CRLCA	29

TABLE DES MATIÈRES	V
2.7 Problématique	30
Conclusion	31
3 Création de corpus et méthodologie	32
Introduction	32
3.1 Corpus linguistique	33
3.1.1 Caractéristique	33
3.2 Méthodologie et outils de création de corpus	33
3.2.1 Procédures de collecte et d'analyse des données	33
3.2.2 Microsoft Access	35
3.2.3 Environnement de développement et Bibliothèques	36
3.3 Prétraitement de l'image	39
3.3.1 Caractéristiques de représentation informatique d'image	40
3.3.2 Types de représentation informatique d'image	42
3.3.3 Méthodes du prétraitement	43
3.4 Représentation de modèle CNN	45
3.4.1 Utilisation des CNN pour la Reconnaissance de Caractères	46
3.4.2 Architecture de CNN	46
Conclusion	53
4 Implémentation et résultats	54
Introduction	54
4.1 Environnement matériel	54
4.2 Environnement du travail	55
4.3 Méthodologie de l'implémentation	55
4.3.1 Importation des bibliothèques et des données nécessaires	56
4.3.2 Pré-traitement des images de la base donnée	56
4.3.3 Initialisation des hyperparamètres et division du corpus	57
4.3.4 Construction de l'architecture CNN	57
4.3.5 Compilation et ajustement du modèle	58
4.3.6 Résumer des résultats	59
4.3.7 Évaluation du modèle	60
4.3.8 Teste du modèle	60
4.3.9 Analyse des résultats	61
Conclusion	62
Conclusion générale	63
Bibliographie	67
4.4 Annexes 1	68
4.5 Annexes 2	69
4.6 Annexes 3	70
Annexes	70
Résumé	71

Table des figures

1.1	histoire de IA	4
1.2	Shémas de décomposition sous-domaine de IA	6
1.3	Les types de ML	7
1.4	Fonctionnement de ML	9
1.5	Flux de ML	9
1.6	Un exemple de perceptron	10
1.7	Traitement à l'intérieur du neurone	10
1.8	Graphe de fonction logistique	11
1.9	Graphe de fonction tanh	11
1.10	Graphe de fonction tanh	11
1.11	Un exemple de réseau de neurones à quatre couches dont deux couches cachées	12
1.12	Différence entre ML et DL	13
1.13	exemple de fonctionnement de DL	14
1.14	exemple de traduction	16
1.15	Analyse des sentiments	16
1.16	Filtrage des Spams	17
1.17	Exemple de redimensionnement	18
1.18	Exemple de la binarisation	18
1.19	Exemple de la suppression du bruit	18
1.20	Étapes d'une reconnaissance optique des caractères	19
2.1	Géographie	23
2.2	Organigramme de CRLCA	30
3.1	Présentation des données sous Excel	35
3.2	Présentation de corpus	36
3.3	Logo de python	36
3.4	Logo de Anaconda	37
3.5	Logo de Jupyter	37
3.6	logo de TensorFlow	37
3.7	Logo de Keras	38
3.8	Représentation des notions image et pixel	40
3.9	Exemple d'une image binaire	42
3.10	Les différentes couleurs en niveau de gris	43
3.11	Exemple d'Architecture standard de CNN	47
3.12	Exemple d'image RGB en couche de convolution	48

3.13	Exemple d'entrant 4 x 4 x 1 avec couche de marge à zéro de taille 1 auquel on applique un noyau 2 x 2 x 1.	49
3.14	représentation de maxpooling	50
3.15	représentation de meanpooling	51
3.16	Représentation graphique de la fonctionnalité Relu.	52
3.17	Représentation de la couche entièrement connectée(FC)	53
4.1	représentation de meanpooling	55
4.2	Les 5 premiers lignes de dataframe	56
4.3	Les résultats d'entraînement du modèle	58
4.4	Table qui résume les résultat obtenue	59
4.5	Évaluation du modèle sur l'ensemble de validation	60
4.6	La lettre prédite par le modèle	61
4.7	Visualisation des résultats	61

Liste des tableaux

2.1	Alphabet Tamahaqt	26
2.2	genre des substantifs	27
2.3	Pronoms Personnels en Tamahaqt	28
2.4	Nombre en Tamahaqt	28
3.1	Voisinage à 4, Voisinage à 8	42

Liste d'abréviations et notations

Abréviation	Signification
IA	Intelligence Artificielle
ML	Machine Learnig
AA	Apprentissage Automatique
MLP	Perceptron Multicouche
DL	Deep Learning
NLP	Traitement du Langage Naturel
OCR	Reconnaissance Optique de Caractères
TIC	Les Technologies de l'Information et de la Communication
HCA	Haut-Commissariat à l'Amazighité
SGBD	Système de Gestion de Base de Données
PPP	Points Par Pouce
DPI	Dots Per Inch
CNN	Convolutional Neural Network
MLP	Perceptrons Multicouches
CONV	La Couche de Convolution
POOL	La Couche de Pooling
ReLU	Rectified Linear Units (La couche de correction)
FC	Fully-Connected (La Couche Entièrement Connectée)
LOSS	La couche de perte
LMDT	Langue Métalangue et Didactique de Tamazight
TAL	Terminologie Traduction et Traitement Automatique des Langues
LAPA	Littérature Art et Patrimoine Amazighe
CA	Civilisation Amazighe
API	Application Programming Interface
RGB	Red, Green, Blue

Introduction générale

Ces dernières années, avec l'évolution des systèmes informatiques et des logiciels sophistiqués, une technologie avancée s'est intégrée dans de nombreux domaines et a touché presque tous les secteurs, allant de la construction, des transports et de la fabrication à l'intelligence économique, à l'éducation et aux soins de santé, ainsi que la reconnaissance automatique.

Toujours plus en quête d'avancées technologiques, la reconnaissance des caractères devient le nouveau terrain d'essai pour diverses solutions avancées. Le but ultime et partagé par tous les développeurs de cette technologie est de mettre l'innovation, sous toutes ses formes, au service des utilisateurs pour améliorer les outils de demain. Cette technologie émergente s'annonce comme l'un des éléments clés du futur développement des systèmes de reconnaissance [7].

Le deep learning, ou apprentissage profond, est une branche de l'intelligence artificielle issue du machine learning (apprentissage automatique). Contrairement à la programmation traditionnelle, où la machine suit des instructions précises, le deep learning permet à la machine d'apprendre de manière autonome.

Ce processus repose sur le traitement de vastes quantités de données par des ordinateurs utilisant des réseaux de neurones artificiels, dont la structure s'inspire de celle du cerveau humain[42].

La reconnaissance automatique des caractères est un domaine de recherche très vaste. Lorsqu'il s'agit de l'appliquer à une langue spécifique, comme le tamahaqt dans la langue amazighe, cela exige une connaissance approfondie et une expansion cognitive de cette langue, utilisée par une minorité de la population en Algérie.

Le tamahaqt, langue des Touaregs, fait face à des défis importants pour sa préservation et son enseignement. L'apprentissage de cette langue à des fins de connaissance est compliqué par le manque d'outils et d'informations disponibles[1].

Ce mémoire vise à répondre à ces besoins en fournissant de la documentation et des données sur la langue tamahaqt, en utilisant la reconnaissance automatique des caractères de cette langue à l'aide de réseaux de neurones convolutifs (CNN).

L'objectif est d'approfondir la compréhension de cette problématique en examinant ses divers aspects, en analysant ses implications et en proposant des perspectives pour son développement futur.

Dans notre travail, nous allons nous focaliser sur les CNN, un des algorithmes les plus performants du deep learning, particulièrement efficaces pour la reconnaissance d'images, y compris les caractères de la langue tamahaqt.

Notre démarche repose sur une méthodologie bien élaborée, visant à créer un corpus puissant malgré le manque d'outils et de ressources documentaires. Pour la collecte de données, nous allons utiliser un livre sur la langue tamahaqt et quelques mémoires de fin d'étude de l'université de Tamanrasset.

Ce mémoire est structuré en quatre chapitres, chacun abordant un aspect spécifique de la reconnaissance automatique des caractères tamahaqt :

Le premier chapitre présente des généralités sur l'intelligence artificielle, le machine learning et le deep learning, en mettant l'accent sur les réseaux de neurones et la reconnaissance optique de caractères (OCR).

Le deuxième chapitre se concentre sur la langue amazighe, en présentant ses origines, sa position géographique et ses variantes. Nous détaillons ensuite la langue tamahaqt, fournissant un tableau de ses alphabets et expliquant brièvement ses verbes, noms, pronoms et nombres. Il introduit également l'organisme d'accueil, le Centre de Recherche en Langue et Culture Amazighes (CRLCA), et la problématique de notre étude.

Le troisième chapitre décrit la création du corpus linguistique, les outils et la méthodologie utilisés. Nous expliquons les environnements de développement et les bibliothèques utilisées pour le traitement, ainsi que le processus de prétraitement d'image et le modèle CNN choisi, avec les paramètres adoptés.

Enfin, le dernier chapitre détaille notre programme, son implémentation et les résultats obtenus.

En conclusion, cette recherche souligne à quel point il est crucial et complexe d'appliquer le deep learning, en particulier les réseaux de neurones convolutifs (CNN), à la reconnaissance automatique des caractères de la langue tamahaqt. En tant que sous-domaine avancé du machine learning, le deep learning permet aux systèmes d'apprendre et de traiter de vastes ensembles de données de manière autonome, en s'inspirant de la structure cérébrale humaine.

L'utilisation de ces technologies pour la langue tamahaqt, est essentielle pour préserver et faciliter l'apprentissage de cette langue minoritaire. Notre approche méthodique a permis de constituer un corpus linguistique solide et d'explorer les possibilités offertes par les CNN pour améliorer la reconnaissance des caractères tamahaqt.

1

Préliminaire sur l'intelligence artificielle

Introduction

Dans notre monde actuel, la reconnaissance automatique est devenue un élément crucial de l'Intelligence Artificielle (IA). Avec les avancées technologiques rapides que nous rencontrons chaque jour, l'IA vise à rendre les machines capables d'apprendre, de raisonner et de prendre des décisions par elles-mêmes [11].

Ce chapitre a pour objectif de nous aider à comprendre les bases de l'IA, du Machine Learning (ML) et du Deep Learning (DL), en mettant particulièrement en avant le Traitement Automatique du Langage Naturel (NLP). Cela nous permettra d'explorer plus en profondeur des sujets spécifiques, comme la reconnaissance automatique des caractères dans la langue amazighe.

Dans un premier temps, nous aborderons la définition et l'évolution historique de l'IA, ainsi que ses fonctionnalités. Ensuite, nous explorerons en détail les sous-ensembles tels que le ML, le DL et le NLP. Cette approche nous préparera à une compréhension approfondie de la reconnaissance automatique des caractères tamahaqt dans la langue amazighe.

Sommaire

Introduction	3
1.1 L'intelligence artificielle	4
1.2 Une brève Histoire de l'IA	4
1.3 Sous ensembles de l'IA	6
1.4 Reconnaissance optique de caractères(OCR)	17

1.1 L'intelligence artificielle

L'Intelligence Artificielle (IA) est composée de deux termes : **Artificiel**, qui signifie "créé par l'homme", et **intelligence**, désignant le "pouvoir de réflexion". Ainsi, l'IA peut être définie comme un "pouvoir de réflexion créé par l'homme".

Grâce à l'IA, nous pouvons créer des machines intelligentes capables de se comporter comme des humains, de penser comme des humains et de prendre des décisions autonomes en utilisant des algorithmes et des modèles mathématiques pour traiter les données.

L'IA couvre divers domaines, tels que la reconnaissance vocale, la reconnaissance d'images, l'apprentissage automatique, les jeux, la robotique et la gestion des données.

1.2 Une brève Histoire de l'IA

Voici quelques points essentiels de l'évolution de l'IA, qui tracent son chemin depuis sa naissance jusqu'à son développement contemporain [40, 21] :

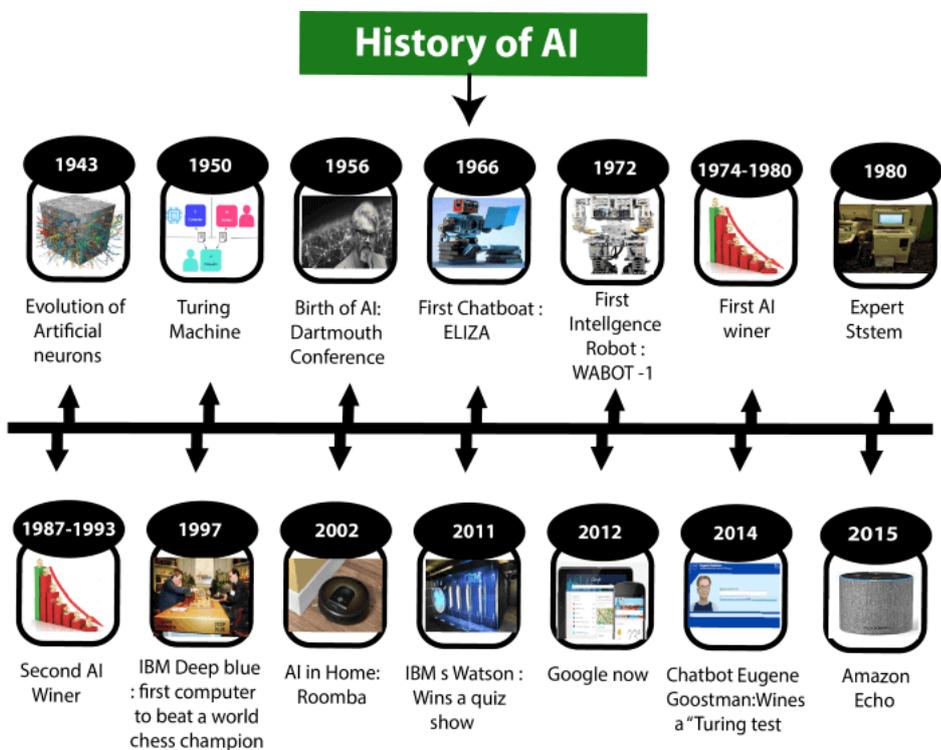


FIGURE 1.1 – histoire de IA

La naissance de l'intelligence artificielle (1952-1956)

- L'histoire de l'intelligence artificielle remonte aux conférences de Macy qui se sont déroulées de 1946 à 1953, précédées par une réunion fondatrice en 1942 sur l'inhibition cérébrale. Ces conférences ont réuni des experts de divers domaines pour discuter des processus cognitifs et du fonctionnement du cerveau.
- Par la suite, le terme « Intelligence Artificielle » a été adopté en 1956 par John McCarthy, marquant le début d'une ère de recherche et de développement dans le domaine de l'IA.

Les années d'or-Les premiers enthousiasmes (1956-1974)

- **En 1966** C'est la création d'algorithmes qui résolvent les problèmes mathématiques et aussi en cette année le premier chatbot a été inventé (eliza).
- **1972** Le WABOT-1 est le premier robot humanoïde intelligent développé au Japon.

Le premier hiver de l'IA (1974-1980) Pendant cette période, l'IA a connu un moment de stagnation.

Un boom de l'IA (1980-1987) Malgré cette période les chercheurs, ce sont reviennent avec un « expert system » qu'est un système expert pour imiter la capacité de prise de décision en 1980.

Le deuxième hiver de l'IA (1987-1993) C'est la seconde période d'AI Winter, en raison des coûts élevés mais des résultats peu efficaces, les investisseurs et le gouvernement ont une fois de plus arrêté de financer la recherche sur l'IA.

L'émergence des agents intelligents (1993-2011)

- Par la suite en 1997 ; (IBM Deep Blue surprend Gary Kasparov), le champion du monde d'échecs, et devient le premier ordinateur à surprendre un champion du monde d'échecs.
- Année 2002 ; l'IA est apparue pour la première fois dans les maisons sous la forme d'un aspirateur Roomba.
- L'IA est arrivée dans le monde des affaires jusqu'en 2006. Des entreprises comme Facebook, Twitter et Netflix ont également commencé à utiliser l'IA. Le Deep learning, big data et intelligence artificielle générale (2011-présent).
- L'IA s'est développée à un niveau remarquable aujourd'hui. La tendance à l'apprentissage profond, aux big data et à la science des données est en plein essor. Des entreprises comme Google, Facebook, IBM et Amazon travaillent aujourd'hui avec l'IA et fabriquent des appareils étonnants. L'avenir de l'IA est inspirant et s'accompagnera d'une intelligence élevée.

1.3 Sous ensembles de l'IA

Jusqu'à présent, nous avons acquis une compréhension de ce qu'est l'Intelligence Artificielle (IA).

Dans cette section, nous allons examiner les différents sous-ensembles de l'IA.

Voici quelques-uns des sous-ensembles les plus courants de l'intelligence artificielle :

- Apprentissage automatique
- L'apprentissage en profondeur
- Traitement du langage naturel

Parmi les domaines de l'IA, nous pouvons citer :

- Système experts
- Robotique
- Vision industrielle
- Reconnaissance de la parole.

Dans cette étude, nous nous basons sur les trois principaux sous-ensembles utilisés, à savoir le Machine Learning, le Deep Learning et le Traitement Automatique du Langage Naturel.

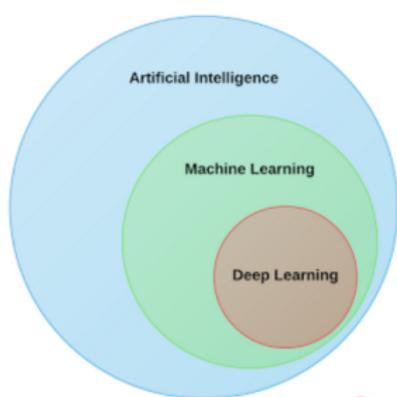


FIGURE 1.2 – Schémas de décomposition sous-domaine de IA

1.3.1 Machine learnig (ML)

L'apprentissage automatique (AA), également connu sous le nom machine Learning en anglais, est un sous-domaine de l'IA qui permet aux systèmes d'apprendre et de s'améliorer à partir des données et des expériences passées sans être précisés programmés.

Il se concentre sur le développement de programmes informatiques capables d'accéder aux données et d'apprendre par eux-mêmes. Il est largement utilisé dans divers domaines, notamment les moteurs de recherche, les voitures autonomes, les systèmes de recommandation (tels que Netflix ou Amazon) et bien d'autres[33, 49].

1.3.1.1 Types d'apprentissage automatique

Il existe plusieurs types d'apprentissage automatique, qui peuvent être classés en fonction de la façon dont les données sont utilisées pour entraîner les algorithmes.

Les principaux types d'apprentissage automatique sont les suivants :

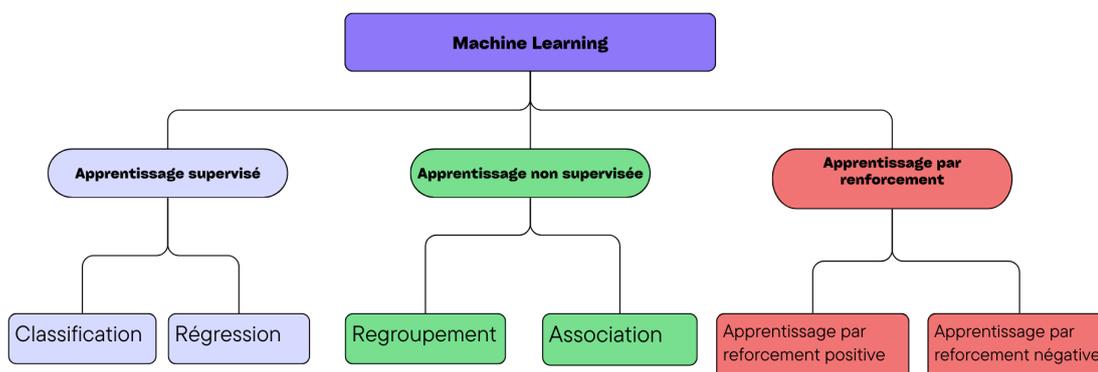


FIGURE 1.3 – Les types de ML

1. **Apprentissage supervisé** : on parle d'apprentissage supervisé lorsque l'on dispose de données d'entraînement étiquetées, c'est à dire dont on connaît la sortie voulue[33].

Il permet aux algorithmes de trouver des corrélations entre les données d'entrée (variables explicatives) et les données de sorties (variables à prédire), pour ensuite inférer la connaissance extraite sur des entrées avec des sorties inconnues[43].

De plus, il existe deux catégories d'algorithmes pour l'apprentissage supervisé :

- **La classification** : est un processus consistant à trouver une fonction qui aide à diviser l'ensemble de données en classes en fonction de différents paramètres[10] ; et sa sortie peut prendre des valeurs discrètes, il s'agit d'une variable réelle. Par un algorithme prédisant la consommation électrique d'une installation ou un algorithme prédisant le cours d'actions en bourse[33].
- **Régression** : est un processus de recherche des corrélations entre les variables dépendantes et indépendantes [33], c'est-à-dire il consiste à choisir une classe (valeur) parmi toutes celles possibles [10]. Il aide à prédire les variables continues. Par exemple Un algorithme prédisant le chiffre manuscrit sur l'image d'entrée ou un algorithme classifiant une tumeur comme « bénigne » ou « maligne »[33].

2. Apprentissage non supervisé : On parle de l'apprentissage non supervisé lorsque les données ne sont pas étiquetées. Il vise à découvrir des structures et des modèles dans les données sans connaître les résultats attendus.

On utilise cette technique pour partitionner les données en groupes d'éléments homogènes, la distance est souvent la plus utilisée comme mesure de similarité entre les groupes. L'objectif du système est d'identifier des caractéristiques communes aux données d'entraînement. Parmi ces catégories on peut citer [10] :

- **Regroupement :** également connu sous le nom de clustering, est un processus de regroupement d'éléments (tels que des données ou des objets) en fonction de caractéristiques ou de critères spécifiques.
- **Association :** est un terme plus général qui se réfère à la relation ou à la connexion entre deux éléments ou plus.

3. Apprentissage par renforcement : Dans ce type d'apprentissage, l'algorithme apprend à choisir des actions en fonction des récompenses ou des pénalités qu'il reçoit de l'environnement (principe d'essai / erreur). Par exemple dans La robotique et la planification d'itinéraires pour les véhicules autonomes utilisent fréquemment l'apprentissage par renforcement[43].

Ce dernier est divisé en deux catégories[10] :

- **Apprentissage par renforcement positif :** lorsque chaque agent fait une bonne action, il reçoit une récompense positive.
- **Apprentissage par renforcement négatif :** lorsque un agent fait une mauvaise action, il reçoit une récompense négative.

1.3.1.2 Fonctionnement de l'apprentissage automatique

L'apprentissage automatique apprend aux ordinateurs à effectuer des tâches qui sont naturelles pour les humains. Les modèles d'apprentissage automatique sont automatiquement créés à l'aide d'algorithmes d'apprentissage qui utilisent des méthodes de calcul pour (apprendre) directement à partir de données brutes.

En d'autres termes, les algorithmes d'apprentissage analysent les données pour créer des modèles. Une fois créé, il sera utilisé sur de nouvelles données pour faire des prédictions[27].

Le schéma fonctionnel suivant décrit comment fonctionne l'algorithme de machine learning :

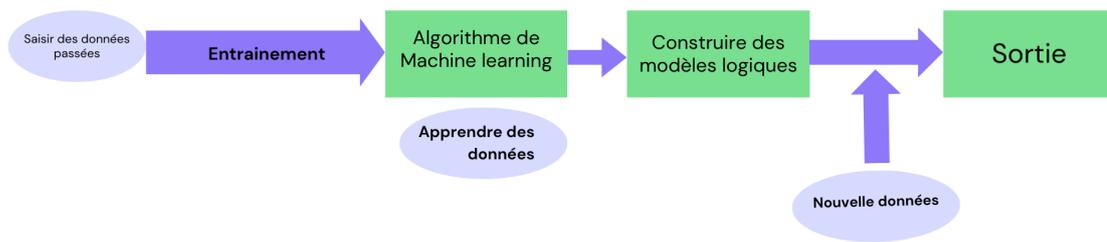


FIGURE 1.4 – Fonctionnement de ML

1.3.1.3 Cycle de vie d'une tâche d'apprentissage automatique

Le cycle de vie d'une tâche d'apprentissage automatique suit généralement le processus décrit dans la figure ci-dessous :

1. Choisissez un algorithme d'apprentissage.
2. Entraînez l'algorithme à l'aide d'instance (appelé ensemble d'apprentissage).
3. Évaluez les performances en exécutant l'algorithme sur un autre ensemble d'instances, appelé ensemble de test.

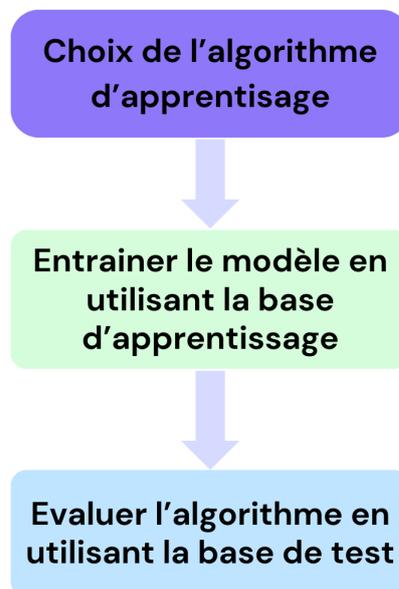


FIGURE 1.5 – Flux de ML

1.3.2 Réseaux de neurones

Les réseaux de neurones artificiels standards (RNA) (Sarle, 1994) sont des modèles d'apprentissage automatique conçus pour simuler le fonctionnement des réseaux de neurones na-

turels du cerveau humain. Ces modèles peuvent être très complexes, avec plusieurs couches de neurones se succédant les unes aux autres. Le perceptron (Rosenblatt, 1958) est considéré comme le réseau de neurones artificiel le plus simple.

Le perceptron se compose d'un seul neurone, également appelé bloc de calcul, qui reçoit des entrées accompagnées de poids associés. Selon une fonction d'activation définie, le perceptron génère une valeur de sortie.

La Figure 1.6 illustre un perceptron, symbolisé par un cercle, qui reçoit en entrée les données x_1 , x_2 et x_3 pour générer une sortie [26].

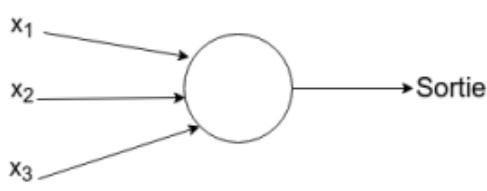


FIGURE 1.6 – Un exemple de perceptron

Le neurone exécute deux opérations essentielles :

Initialement, il calcule une somme pondérée des entrées qu'il reçoit, c'est-à-dire qu'il additionne chaque entrée multipliée par son poids correspondant (θ_i) ($\sum x_i * \theta_i$) ; Durant la phase d'apprentissage, ces poids θ_i sont modifiés en fonction d'un taux d'apprentissage, noté $\lambda \in [0, 1]$, jusqu'à ce que les réponses du neurone soient en accord avec la sortie attendue.

Finalement, cette somme pondérée est traitée par une fonction spécifique, dénommée fonction d'activation. En fonction du résultat de cette fonction, le neurone s'active et fournit une valeur en sortie, comme démontré dans la Figure 1.7 [44].

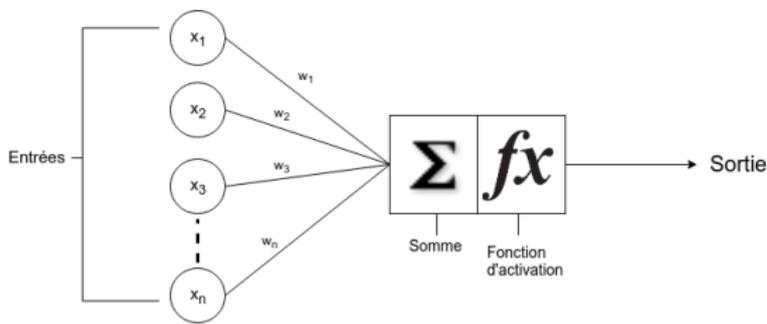


FIGURE 1.7 – Traitement à l'intérieur du neurone

Il existe une variété de fonctions d'activation qui peuvent être utilisées dans les neurones artificiels. La plupart de ces fonctions sont continues et leurs valeurs sont définies sur les intervalles $[0, +1]$ ou $[-1, +1]$. Voici quelques exemples de fonctions d'activation couramment utilisées dans le domaine des réseaux de neurones artificiels [44] :

- fonction sigmoïde (logistique) :

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

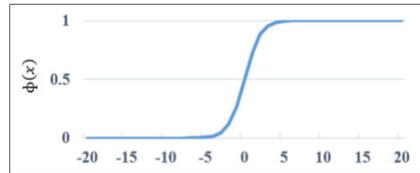


FIGURE 1.8 – Graphe de fonction logistique

- fonction tangente hyperbolique (tanh) :

$$\phi(x) = 2 \left(\frac{1}{1 + e^{-2x}} \right) - 1$$

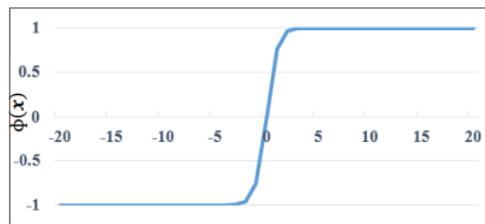


FIGURE 1.9 – Graphe de fonction tanh

- fonction d'unité linéaire rectifiée (relu) :

$$\phi(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$$

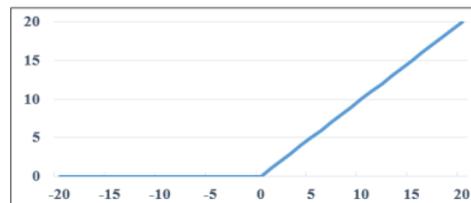


FIGURE 1.10 – Graphe de fonction tanh

Perceptron multicouche (MLP) qui est un type spécifique de réseau de neurones artificiels, est composé de plusieurs perceptrons et donc de nombreux neurones répartis sur différentes couches.

Ce MLP comprend une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie.

Pour obtenir une sortie d'un MLP, l'information est introduite par la couche d'entrée. Ensuite, chaque neurone de chaque couche génère une valeur qui est transmise aux neurones de la couche suivante. Ce processus se poursuit jusqu'à la dernière couche, qui est composée d'un ou plusieurs neurones et qui fournit la sortie finale, comme illustré dans la Figure 1.11 [26].

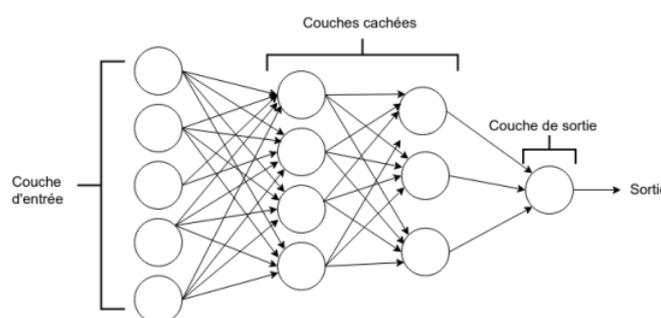


FIGURE 1.11 – Un exemple de réseau de neurones à quatre couches dont deux couches cachées

Chaque neurone dans ces couches utilise une fonction d'activation pour transformer ses entrées en sortie. Les couches de neurones situées entre la couche d'entrée et la couche de sortie sont appelées couches cachées. L'objectif de l'entraînement d'un réseau de neurones est de déterminer de manière itérative les poids optimaux à appliquer entre les neurones pour améliorer la précision du réseau [44].

1.3.3 Deep Learning (DL)

L'apprentissage profond, également connu sous le nom *deep learning* en anglais, peut être considéré comme le dernier terme dans le domaine de l'apprentissage automatique ; C'est une méthode de mise en œuvre de l'apprentissage automatique.

On utilise des algorithmes machines pour analyser les données, en tirer des leçons et prendre des décisions éclairées en fonction de ce qu'elles ont appris. Fondamentalement, le DL est utilisé par couches pour créer un (réseau de neurones artificiels) capable d'apprendre par lui-même et de prendre des décisions intelligentes [3].

Il est particulièrement adapté à la reconnaissance d'images, importante pour résoudre des problèmes tels que la reconnaissance faciale, la détection de mouvements et de nombreuses technologies avancées d'aide à la conduite telles que la conduite autonome, la détection de voie, la détection de piétons et le stationnement automatique [22].

1.3.3.1 ML vs DL

La différence entre ces deux types d'apprentissage est que le DL va être entraîné pour extraire lui-même les éléments influents dans la prédiction souhaitée. Par contre ML prévoit des

algorithmes pour faire sortir ces éléments (features), la figure ci-dessus illustre cette différence. En outre, la nature des données traitées par les différents types d'apprentissage diffère également :

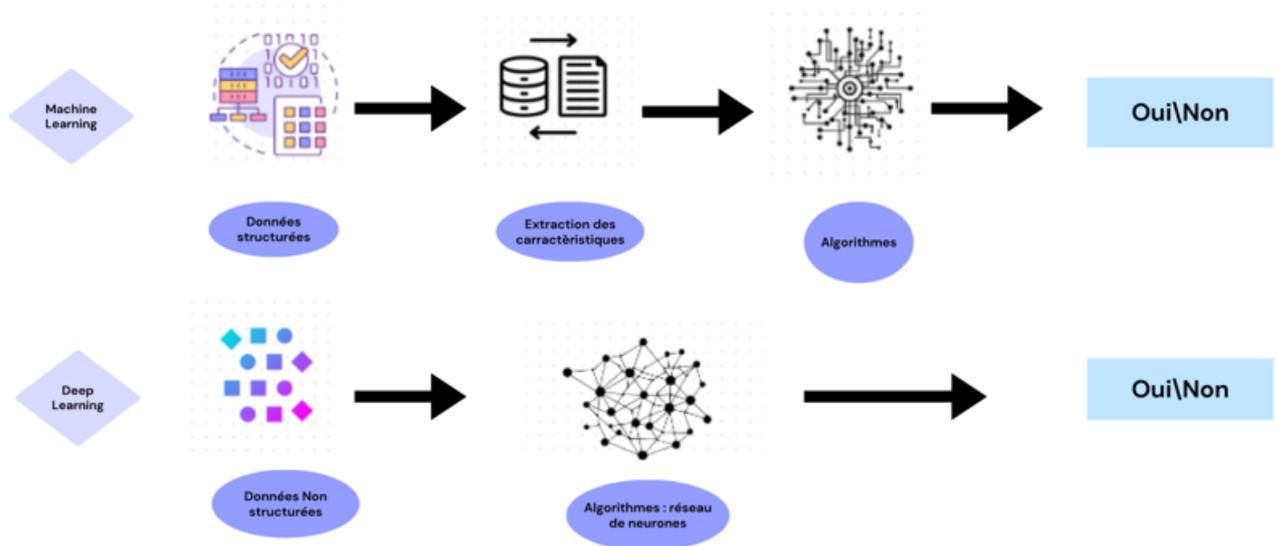


FIGURE 1.12 – Différence entre ML et DL

alors que le ML utilise des données quantitatives et structurées (valeurs numériques), l'apprentissage profond utilise des données non structurées telles que la parole, le texte et les images.

1.3.3.2 Les différentes Architectures du Deep Learning

Dans un réseau neuronal d'apprentissage profond, les informations sont transmises à travers chaque couche, la sortie de la couche précédente devenant l'entrée de la couche suivante. La première couche du réseau est appelée **couche d'entrée** et la dernière couche est appelée **couche de sortie**. Toutes les couches intermédiaires sont appelées **couches cachées**. Plus le nombre de neurones est élevé, plus le réseau est "profond". Chaque couche est généralement un algorithme simple et uniforme contenant un seul type de fonction d'activation [37].

1.3.3.3 Fonctionnement de DL

L'approche classique de l'analyse consiste à concevoir une entité en utilisant les données disponibles pour déduire une nouvelle variable, puis à sélectionner un modèle analytique et enfin à estimer les paramètres (ou inconnues) de ce modèle.

Pour mieux comprendre le fonctionnement de DL, on prend l'exemple suivant :

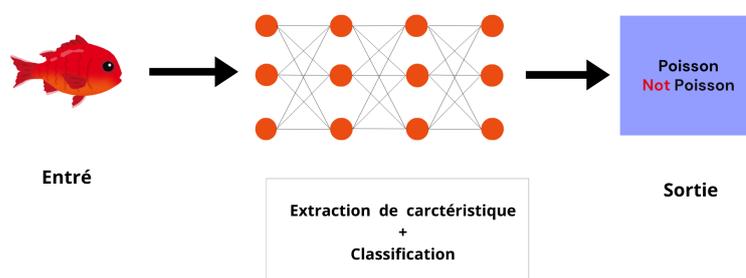


FIGURE 1.13 – exemple de fonctionnement de DL

On veut identifier un poisson parmi un grand groupe de poissons qui sont filmés, quelle que soit la manière dont ils sont photographiés.

Pour ce faire, on doit collecter un grand nombre d'images de poissons et d'autres objets afin d'entraîner un réseau neuronal artificiel pour l'apprentissage profond.

Ces images sont converties en données et envoyées au réseau après que les neurones artificiels ont attribué des poids aux différents éléments. Les informations sont ensuite collectées par la dernière couche de neurones pour déterminer s'il s'agit d'un poisson ou non.

Le réseau neuronal artificiel compare ensuite ce résultat avec la réponse donnée. Si une correspondance est trouvée, le réseau enregistre ce succès, car il l'utilisera dans d'autres communes ; Si le résultat est négatif, le réseau reconnaît un déséquilibre et ajuste les poids des différents neurones pour corriger l'erreur ; Ce processus est répété des milliers de fois jusqu'à ce que l'image soit reconnue.

Ce type d'apprentissage est connu sous le nom d'apprentissage supervisé ; la seconde méthode est l'apprentissage non supervisé et est basée sur des données non classifiées. Cela signifie que le réseau doit identifier des modèles dans l'ensemble de données. Pour apprendre par eux-mêmes que tous les éléments de l'image peuvent être pertinents [3].

1.3.3.4 Domaine d'application de DL

Le Deep Learning est utilisé dans de nombreux domaines comme :

- **Reconnaissance vocale** : par exemple, Xbox, Skype, Google Now et la Syrie d'Apple ont introduit une technologie d'apprentissage profond dans leurs systèmes pour reconnaître les modèles de parole et les sons de l'homme.
- **Reconnaissance d'images** : une application pratique de la reconnaissance d'images est la traduction automatique d'images et la description de scènes. Cette application est très efficace dans le domaine de l'investigation, où l'activité criminelle est déterminée à partir des milliers d'images fournies. Les véhicules peuvent également bénéficier de l'identification d'images grâce à la technologie des caméras à 360°.

- **Système de recommandation** : Amazon et Netflix ont intégré avec succès le concept de système de recommandation, avec une possibilité le degré d'intérêt de chaque utilisateur et il va le recommander dans la prochaine utilisation. Il est utilisé dans d'autre domaine tel que la musique.
- **Traitement du langage naturel** : les réseaux neuronaux font depuis longtemps partie intégrante de l'apprentissage profond pour le traitement et l'analyse de textes écrits. Ils peuvent également être utilisés pour découvrir et analyser les notes des médecins et les rapports d'actualité.
ce qu'on va traiter dans la prochaine section

1.3.4 Traitement du langage naturel(NLP)

PNL signifie le processus de traitement naturel du langage (TLN), ou NLP Natural Language Processing en anglais, qui fait partie de l'informatique, du langage humain et de l'intelligence artificielle ; Il est l'élément qui fait le lien entre la conversation humaine et la compréhension programmée par ordinateur (quels que soient les humains qui parlent, quelle langue est parlée et la façon dont ils parlent, même si c'est peut-être grammaticalement incorrect).

Le composant NLP permet à l'ordinateur d'interpréter le langage humain vaste et compliqué, de comprendre ce qui est dit, de tout traiter, de refléter ce qui est demandé et de réagir efficacement, comme le font les humains [9].

Son objectif est de rendre le langage humain compréhensible par les machines en transformant le langage naturel en une représentation formelle facilement manipulable par les ordinateurs afin d'étudier des problèmes fondamentaux permettant de modéliser des données textuelles en vue d'en extraire des informations [25].

1.3.4.1 Les Principales méthodes utilisées en NLP

D'une manière générale, il existe deux aspects de la PNL[49] :

- La partie "linguistique", qui implique le pré-traitement et la transformation des informations d'entrée en ensembles de données utilisables.
- La partie "apprentissage automatique" ou "science des données" applique des modèles d'apprentissage automatique ou d'apprentissage profond à cet ensemble de données.

1.3.4.2 Intérêt du NLP

NLP présente de nombreux avantages, mais en voici quelques-uns de haut niveau qui peuvent aider les entreprises à devenir plus compétitives :

- Le NLP permet aux machines de comprendre et d'analyser automatiquement de grandes quantités de données textuelles non structurées, par exemple les commentaires sur les réseaux sociaux.
- Aident les machines à apprendre à classer et à acheminer les informations rapidement, efficacement, avec précision et 24 heures sur 24, avec peu ou pas d'intervention humaine.

- Créer automatiquement des résumés lisibles de codes sources complexes et volumineux.
- Analyse des sentiments facile à réaliser

1.3.4.3 Applications de la NLP

Le traitement du langage naturel peut être appliqué dans divers domaines, tels que :

1. Traduction automatique

La traduction automatique est un moyen de traduire un texte ou un discours d'une langue naturelle à une autre, par exemple traducteur Google translate [39].

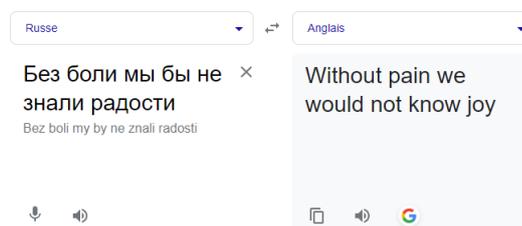


FIGURE 1.14 – exemple de traduction

2. Analyse des sentiments

L'exploration des opinions est un autre nom pour l'analyse des sentiments. Il est utilisé sur Internet pour évaluer l'attitude, le comportement et l'état émotionnel de l'expéditeur. En attribuant des valeurs au texte (positives, négatives ou naturelles), en identifiant l'ambiance du contexte (heureuse, triste, en colère, etc.), cette application est mise en œuvre grâce à une combinaison de NLP et de statistiques [39].



FIGURE 1.15 – Analyse des sentiments

3. Extraction d'informations

L'une des applications les plus importantes de la PNL est l'extraction d'informations. Il concerne l'identification des phrases d'intérêt des données textuelles.

Pour de nombreuses applications, l'extraction d'entités telles que les noms, les lieux, les événements, les dates, les heures et les prix sont un moyen puissant de résumer les informations pertinentes aux besoins d'un utilisateur donc il peut extraire des informations structurées de documents non structurés ou semi-structurés qui sont lisibles par machine[39].

4. Filtrage des spams

Le filtrage du spam est utilisée pour détecter les e-mails indésirables arrivant dans la boîte de réception d'un utilisateur[39].

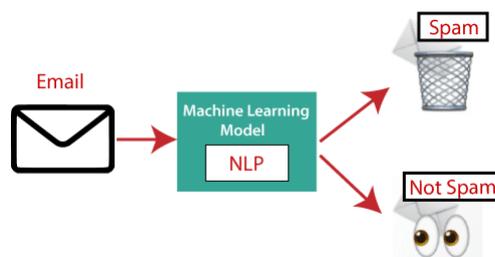


FIGURE 1.16 – Filtrage des Spams

5. Reconnaissance vocal

Les mots prononcés peuvent être convertis en texte à l'aide de la reconnaissance vocale. Il est utilisé dans des applications comme le mobile, la récupération vidéo, la dictée sur Microsoft Word, la biométrie vocale, l'interface utilisateur vocale, etc [30].

6. Reconnaissance des caractères

La reconnaissance des caractères s'effectue à l'aide d'un système spécialisé appelé OCR. Son objectif est de convertir des images de texte en texte lisible par l'ordinateur, avec le moins d'erreurs de conversion de caractères possible. Ce qu'on va traiter dans la section suivante[6].

1.4 Reconnaissance optique de caractères(OCR)

La reconnaissance de caractères est réalisée à l'aide de systèmes dédiés appelés OCR (Optical Character Recognition en anglais) permet de numériser un texte ou des documents, afin de pouvoir les éditer sur un smartphone ou un ordinateur.

Les systèmes OCR reconnaissent automatiquement le texte imprimé et le convertissent en un fichier électronique. En scannant un document, l'appareil peut en "lire" le contenu ; les systèmes OCR peuvent reconnaître différentes polices et différents caractères sur les machines à écrire et les ordinateurs Certains systèmes OCR peuvent également reconnaître du texte manuscrit.

Le texte lu à partir de documents scannés sur un smartphone ou un ordinateur peut être utilisé pour remplir automatiquement des formulaires[50].

1.4.1 Fonctionnement de l'OCR

Les systèmes OCR utilisent les technologies les plus récentes pour collecter des informations à partir de documents scannés (textes, photos) et les convertir en fichiers texte. Pour ce faire, le système OCR compare les couleurs noir et blanc du document pour identifier chaque code alphanumérique. Le système reconnaît ensuite chaque caractère et le convertit en texte ASCII (American Standard Code for Information Interchange).

Cela permet, par exemple, d'éditer, de rechercher et de copier un texte Word le plus rapidement possible [50].

Les étapes du traitement sont les suivantes :

1. **Prétraitement de l'image** : L'amélioration de la qualité des images et l'élimination du bruit et des informations non pertinentes sont importantes pour la reconnaissance

correcte des caractères dans les images.

Parmi les techniques couramment utilisées pour améliorer les images et les documents numérisés sont les suivantes[31] :

- **Le redimensionnement** : Permet de corriger l'angle et a ajusté la photo



FIGURE 1.17 – Exemple de redimensionnement

- **La binarisation** : Convertit une image en une image binaire en distinguant le premier plan (texte) de l'arrière-plan.

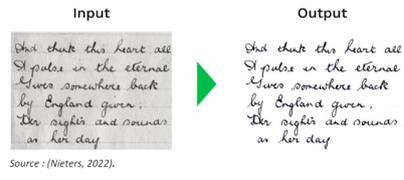


FIGURE 1.18 – Exemple de la binarisation

- **La suppression du bruit** De nombreuses images peuvent contenir des éléments indésirables tels que des rayures, des taches et des points qui interfèrent avec la reconnaissance des caractères et doivent être supprimés[31].



FIGURE 1.19 – Exemple de la suppression du bruit

- **La normalisation** La normalisation est un processus visant à éliminer les différences d'écriture et à obtenir des données normalisées. Il s'agit, par exemple, de convertir les majuscules en minuscules et de supprimer les espaces et les symboles inutiles.

2. **Segmentation** : cette méthode se concentre sur l'identification et la séparation des caractères en extrayant les caractéristiques locales telles que les segments de ligne courts et les régions fermées[31].

3. **Reconnaissance des caractères** : Après l'étape de normalisation, les instances reconnues sont comparées à une bibliothèque de formes connues et l'étape suivante consiste à trouver la forme la plus proche en fonction des propriétés de distance et de vraisemblance. les types des techniques de la reconnaissance sont les suivants [31] :
 - **Classification par caractéristique** est un processus de reconnaissance qui utilise un vecteur de valeurs numériques. Les caractéristiques sont organisées en classes, et une classe de caractères peut être comparée à la classification d'une lettre représentée par un nuage. L'objectif est de déterminer la classe de caractères à laquelle la forme est la plus probablement associée.
 - **Méthode métrique** Compare la forme à reconnaître avec l'algorithme de distance. C'est une approche qui est moins efficace que certaines autres méthodes.
 - **Méthode statique** Dans le domaine de la reconnaissance de l'écriture manuscrite, on utilise fréquemment des méthodes basées sur les probabilités et les statistiques, telles que les chaînes de Markov.
4. **Post-traitement** : Dans cette phase, Nous faisons appel à des approches linguistiques et contextuelles pour minimiser les erreurs.
Nous pouvons utiliser des systèmes basés sur des règles ou des méthodes statistiques ; dans les systèmes industriels, nous employons des techniques spécifiques pour traiter différentes zones de texte[50].



FIGURE 1.20 – Étapes d'une reconnaissance optique des caractères

1.4.2 L'utilisation d'OCR

La technologie OCR est un outil adaptable qui trouve son application dans une multitude de situations, que ce soit pour des documents officiels ou des jeux occasionnels. Il existe également d'autres cas d'utilisation, tels que :

- **Saisie automatique de données** : chèques, passeports, factures, etc.
- **Reconnaissance automatique des caractères d'une langue** : (qui est le sujet de notre mémoire).
- **Reconnaissance des passeports** : par exemple, dans un aéroport.
- **Extraction automatique d'informations clés** : comme les noms, les adresses et les dates dans des documents légaux.
- **Extraction des informations de cartes de visite** : telles que les noms, titres, numéros de téléphone et adresses e-mail.
- **Numérisation de documents volumineux** : livres, mémoires de fin d'études, etc.

- **Conversion de l'écriture manuscrite en temps réel** Par exemple, les centres de tri du courrier utilisent souvent des systèmes OCR pour gérer et trier le courrier, ce qui permet un tri plus rapide et des délais de livraison améliorés. Le système est également populaire dans les applications pour smartphones, où il est utilisé pour numériser des reçus ou reconnaître des documents manuscrits.[50]

Conclusion

Dans ce chapitre, nous avons présenter les bases de l'IA, du Machine Learning, du Deep Learning, du Traitement du langage naturel (NLP) et de l'OCR. Nous avons discuté de leurs applications, de leur importance et de leur fonctionnement.

Ce chapitre sert de fondement aux chapitres suivants, où nous nous concentrerons sur la langue amazighe, en particulier le dialecte tamahaqt.

2

Le dialecte Tamahaqt

Introduction

La langue Tamazight, ou Berbère, dépasse son rôle de simple moyen de communication en Afrique du Nord ; elle incarne une identité culturelle et une diversité linguistique profondes. Parlée principalement au Maroc, en Algérie, en Tunisie, en Libye, et dans certaines parties du Mali et du Niger, elle est l'une des langues autochtones les plus anciennes de la région, avec plusieurs dialectes tels que le kabyle, le Tachelhit, le Tamzabt, le Tarifit et le Tamahaqt.

Ce chapitre se concentre sur le dialecte Tamahaqt, parlé par les Touaregs, son évolution, son histoire, sa situation géographique, et son écriture en Tifinagh. Nous aborderons les défis posés par la complexité de la langue et son faible soutien technologique, ainsi que les approches pour développer des systèmes de reconnaissance automatique des caractères tamahaqt, favorisant ainsi la préservation et l'utilisation de cette langue précieuse.

Sommaire

Introduction	21
2.1 La langue Amazighe	22
2.2 L'origine et la géographie de la langue Amazighe	22
2.3 Les variantes de la langue Amazighe	24
2.4 Le parlé Tamahaqt	24
2.5 L'écriture du dialecte Tamahaqt	25
2.6 Présentation d'organisme d'accueil (CRLCA)	28
2.7 Problématique	30
Conclusion	31

2.1 La langue Amazighe

La langue amazighe est caractérisée par sa diversité dialectale, avec plusieurs variantes régionales et locales. Elle possède également une riche tradition littéraire et culturelle, et elle est reconnue comme l'une des langues officielles en Algérie et au Maroc.

La langue amazighe est importante pour l'identité culturelle des populations berbères et fait l'objet d'efforts de préservation et de promotion, notamment dans le domaine de l'éducation et des médias. Elle joue un rôle essentiel dans la préservation de la culture et de l'histoire des communautés berbères[47].

Parmi les outils dédiés à la préservation de cette langue, les Technologies de l'Information et de la Communication (TIC) contribuent activement à la préservation du patrimoine amazigh en offrant des solutions innovantes telles que la création d'une banque de données numérique, la numérisation et l'archivage, le traitement automatique des langues, ainsi que la transcription automatique et la translittération. L'utilisation des TIC facilite ainsi la conservation, la gestion et la diffusion du patrimoine linguistique et culturel de la langue amazighe à travers des moyens numériques novateurs[15].

2.2 L'origine et la géographie de la langue Amazighe

L'UNESCO a classé le tamazight parmi les langues menacées de disparition, étant la langue maternelle de millions de personnes. En 1996, cet organisme a publié un Atlas des langues en danger dans le monde qui indique que :

La majorité des langues d'Afrique du Nord qui sont en danger sont celles parfois désignées collectivement comme étant le groupe amazigh ou berbère. Elles constituent des poches de communautés linguistiques qui survivent mais disparaissent rapidement, entourées de locuteurs de l'arabe nord-africain qui sont eux-mêmes d'anciens locuteurs du tamazight arabisés.

La trajectoire du tamazight en Algérie est caractérisée par une lutte inlassable visant à obtenir sa reconnaissance et son intégration au sein de la société et du système éducatif[32].

- Avant les années 1980, le tamazight était largement marginalisé au profit de l'arabe et du français, langues dominantes dans l'enseignement et l'administration en Algérie.
- Le Printemps berbère d'avril 1980 est considéré comme un tournant majeur dans la lutte pour la reconnaissance du tamazight comme composante essentielle de l'identité nationale en Algérie. À partir de 1992, des efforts officiels ont été entrepris pour reconnaître la langue amazighe, avec l'ouverture des départements de langue et culture amazighes dans les universités de Tizi-Ouzou et de Bejaïa.
- En 1995, la grève du cartable a conduit à la généralisation de l'enseignement du tamazight et à la création du Haut-commissariat à l'amazighité (HCA).

- En décembre 2003, un Centre national pédagogique et linguistique pour l'enseignement de la langue amazighe a été établi, suivi en 2009 par la création de la télévision nationale en berbère et de deux radios régionales kabyles.
- En 2016, le tamazight a été promu langue officielle par un avant-projet de révision constitutionnelle, et la nouvelle constitution de l'Algérie a ensuite officiellement déclaré le tamazight comme langue nationale et officielle.

L'Algérie a reconnu le Tamazight dans sa constitution de 2016, après des décennies de demandes, symbolisant une avancée significative pour l'identité culturelle des Berbères. Malgré cette reconnaissance, le Tamazight reste marginalisé dans les sphères administratives, judiciaires et éducatives, touchant seulement 6,67% des élèves.

Cette officialisation a soulevé des défis complexes nécessitant une réévaluation des structures existantes pour une intégration pleine et effective, en accord avec les principes de diversité culturelle promus par l'UNESCO.

Le Mouvement Amazigh au Maroc émerge à la fin des années 1960 et au début des années 1970, porté par des intellectuels urbains éduqués et des associations culturelles regroupées sous le nom de Mouvement Culturel Amazigh (MCA).

Il vise à promouvoir la langue et l'identité amazighes, cherchant à redéfinir la diversité culturelle marocaine contre une homogénéisation basée sur l'arabité et l'islamité. Malgré cela, les Amazighs demeurent marginalisés et leur langue et culture ignorées dans d'autres régions berbérophones [45, 14, 8].

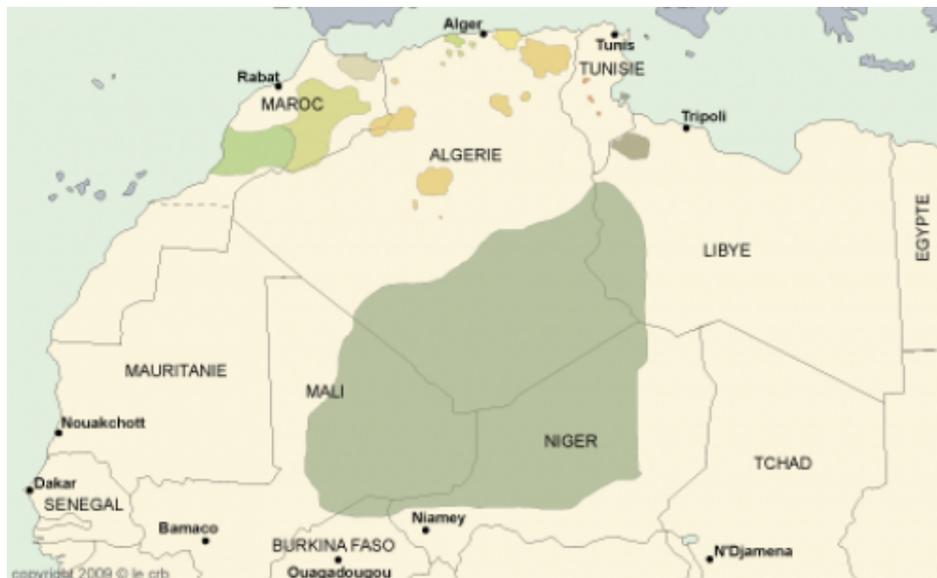


FIGURE 2.1 – Géographie

2.3 Les variantes de la langue Amazighe

En Algérie, on retrouve plusieurs variantes telles que le kabyle, le chaoui, le mozabite, le touareg, entre autres. Bien que ces variantes dérivent originellement de la même langue, le Tamazight, chaque zone dialectale présente des particularités distinctes, principalement au niveau phonétique et lexical. Certaines variantes, comme les dialectes du Sahara, notamment le touareg, ont développé des spécificités distinctes, y compris au niveau syntaxique.

Les variantes de la langue amazighe en Algérie reflètent la diversité linguistique et culturelle de pays.

Les Chaouis, représentant près de 2,8 millions de personnes, parlent le tachawit et sont principalement basés à l'est du pays, notamment dans les régions des Aurès telles que Batna, Khenchla, Tibessa, Souk Ahras, Sétif, Guelma et Biskra.

Les Kabyles, avec près de 5,5 millions de locuteurs, utilisent le thakbaylith, la deuxième variante la plus parlée après le tachelhit. Ils sont concentrés dans des villes telles que Bejaïa, Tizi-Ouzou, Bouira et Boumerdes, avec une diaspora estimée à 7 millions de personnes à travers le monde.

Le Chenoui, avec seulement 750000 locuteurs, est parlé dans les régions de Tipaza et Chlef, tandis que le Mozabite est largement répandu dans le sud, à Ghardaïa, avec près de 800 000 locuteurs.

Les populations touarègues, présentes dans l'extrême sud de l'Algérie, parlent une variante appelée tamasheq ou tamahaq ou tamajaq, comptant environ un million d'individus en Algérie et dans les pays voisins tels que la Libye, le Mali, le Niger et le nord du Burkina Faso[34].

2.4 Le parlé Tamahaqt

Le tamahaqt, également connu sous le nom de tamasheq, touareg ou encore tamajaq, est un dialecte de la langue tamazight parlé par les Touaregs, une communauté nomade présente dans plusieurs pays d'Afrique du Nord, notamment en Algérie, au Mali, au Niger et en Libye. Ce dialecte possède une riche histoire et une importance culturelle significative au sein de la communauté touarègue. Il se caractérise par ses particularités phonétiques, grammaticales et lexicales distinctes, qui reflètent le mode de vie nomade et les traditions des Touaregs.

La langue des Touaregs est une des langues berbères parlées dans les régions montagneuses allant de l'Égypte jusqu'aux côtes occidentales du Maroc. Les différents dialectes de ces régions sont influencés par l'arabe et sont souvent similaires, facilitant la compréhension entre les habitants de différentes régions.

Certains dialectes, tels que ceux de l'oasis de Sokna et de Ghadamisieh Agai, sont moins influencés par l'arabe. Malgré les influences arabes, les langues kabyle en Algérie et chleuh au Maroc partagent une origine commune avec les langues touarègues. La langue touarègue est

considérée comme la plus proche de la forme originale de la langue berbère, bien que certains dialectes du nord du Sahara aient intégré des mots arabes. Les dialectes parlés par les Touaregs du sud du Sahara, comme le Temashught, présentent probablement davantage de similitudes avec les langues soudanaises[18].

2.5 L'écriture du dialecte Tamahaqt

Nous n'avons pas trouvé de tableau d'alphabet officiel qui nous aurait fourni les lettres tamahaqt utilisées officiellement par les Touaregs.

Par conséquent, nous avons collecté toutes les lettres en usage chez les Touaregs et sélectionné celles qui étaient les plus fréquemment rencontrées. Enfin, nous avons constitué ce tableau illustrant les lettres tamahaq accompagnées d'exemples explicatifs [51].

2.5.1 Les verbes

La forme la plus simple du verbe est la troisième personne du singulier du passé simple, considérée comme une racine.

le genres du verbe :

- La troisième personne du singulier a des formes distinctes pour masculin et féminin.
- La deuxième personne du singulier et la première personne du singulier n'ont qu'une seule forme chacune, commune aux deux genres.
- Au pluriel, la troisième et la deuxième personne ont chacune leurs formes masculines et féminines, tandis que la première personne n'a qu'une seule forme commune [18].

2.5.2 Les noms

noms substantifs Tous les noms substantifs en langue tamahaqt sont soit masculins soit féminins. Ceux qui ont pour lettre initiale ou finale, ou l'une d'entre elles, + sont féminins, tous les autres sont masculins.

Français	Tamasheq (approximation)	Tifinagh
Garçon	Abarad	ⴰⴱⴰⴷ
Fille	Tabarat	ⵜⴰⴱⴰⴷⴰⵜ

TABLE 2.2 – genre des substantifs

noms d'adjectifs Les adjectifs sont capables de toutes les inflexions des temps verbaux. Cette particularité, qui semble être commune à tous les dialectes berbères, a conduit certains linguistes à considérer qu'il n'y a pas d'adjectifs dans la langue berbère, leur place étant remplacée par des verbes de qualité.

Cependant, ces mots ne semblent pas apparaître en tamahaqt, du moins pour permettre la formation d'un futur ou de tout autre temps dérivé. De plus, la racine a une signification présente alors qu'elle devrait être passée si c'était un verbe. Elle est considérée comme des adjectifs [18].

2.5.3 Les pronoms

En tamahaqt, les pronoms sont utilisés de manière similaire à d'autres langues pour représenter des personnes, des objets ou des concepts.

Ces pronoms peuvent avoir des formes différentes selon le contexte ou la fonction grammaticale qu'ils remplissent dans la phrase ; Voici une représentation simplifiée des pronoms personnels en tamahaqt.

Français	Tamasheq	Tifinagh (approximatif)
Je	An	ⵏⵉ
Tu (masculin)	Agma	ⵏⵖⵎⵓ
Tu (féminin)	Atma	ⵏⵖⵎⵓ
Il	Aman	ⵏⵎⵏ
Elle	Afata	ⵏⵖⵏⵏⵓ
Nous	Aynan / Negnan	ⵏⵏⵏ / ⵏⵏⵏ
Vous	Aghan / Negghan	ⵏⵖⵏⵏ / ⵏⵖⵏⵏⵏ
Ils	Amagna	ⵏⵎⵏⵏ
Elles	Afatna	ⵏⵖⵏⵏⵏ

TABLE 2.3 – Pronoms Personnels en Tamahaqt

2.5.4 Les nombres

L'écriture des chiffres en Tamahaqt suit une séquence particulière et utilise des caractères spécifiques de l'alphabet pour représenter chaque chiffre voici l'écriture latin des chiffres en tamahaqt [18].

0 - ⵓⵔ (zur)
1 - ⵏ (yan)
2 - ⵉ (sin)
3 - ⵓⵔ (krar)
4 - ⵓⵔ (kuz)
5 - ⵉⵏⵏⵓ (smas)
6 - ⵉⵏⵏⵓ (ses)
7 - ⵉⵏⵏⵓ (sbah)
8 - ⵏⵏⵏ (tam)
9 - ⵏⵏⵏⵓ (tza)
10 - ⵉⵏⵏⵓ (mraw)

TABLE 2.4 – Nombre en Tamahaqt

2.6 Présentation d'organisme d'accueil (CRLCA)

Le Centre de Recherche en Langue et Culture Amazighes (CRLCA)¹ joue un rôle essentiel dans la préservation et la promotion de la langue et de la culture amazighes. Établi en 2017, il s'agit d'un établissement public à caractère scientifique et technologique.

Au sein de la division du Traitement Automatique des Langues (TAL), nous avons eu l'opportunité de collaborer avec la deuxième équipe, dirigée par M. Zaidi. Notre travail s'est concentré sur le traitement automatique des caractères de la langue tamazighe (tamahaqt).

1. Ces information sont extraire dans le site officielle de CRLCA (<https://crlca.dz/fr/>)

Ce stage nous a permis de collecter les données nécessaires pour notre étude. L'équipe nous a fourni tous les documents nécessaires et a grandement contribué à la réussite de notre projet.

Le TAL est un domaine de recherche pluridisciplinaire qui fait collaborer l'informatique et la linguistique. Il vise à créer des outils de traitement du langage naturel pour diverses applications. Au CRLCA, les équipes de recherche se consacrent principalement au développement de systèmes de reconnaissance (OCR) des caractères amazighs Latin et Tifinagh, en se basant sur l'intelligence artificielle.

En somme, notre stage au CRLCA a été une expérience extrêmement enrichissante. Nous avons non seulement acquis de nouvelles compétences techniques, mais nous avons également eu l'occasion de travailler sur un projet d'une grande importance culturelle et linguistique. Nous sommes reconnaissants pour cette opportunité et espérons pouvoir continuer à contribuer à ce domaine crucial à l'avenir.

2.6.1 Les divisions du centre CRLCA

- Langue Métalangue et Didactique de Tamazight LMMDT
- Terminologie Traduction et Traitement Automatique des Langues TAL
- Littérature Art et Patrimoine Amazighe LAPA
- Civilisation Amazighe CA

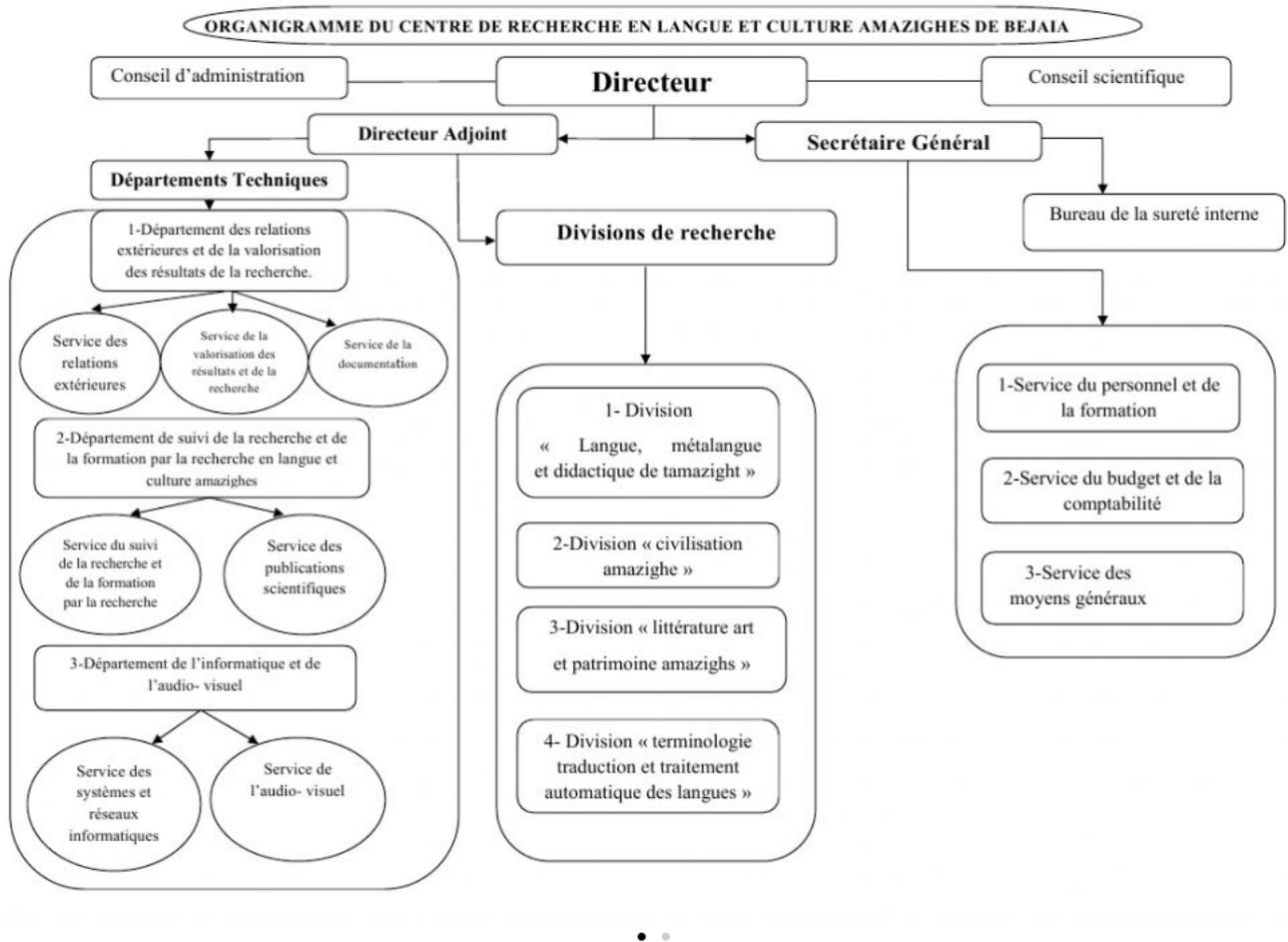


FIGURE 2.2 – Organigramme de CRLCA

2.7 Problématique

Dans le cadre de notre projet, nous visons à créer des outils avancés pour le traitement automatique du dialecte tamahaqt de la langue amazighe. Ce travail s'inscrit dans un contexte où de nombreux défis importants se posent dans le domaine de la linguistique computationnelle. D'une part, le tamahaqt, comme de nombreux autres dialectes, souffre d'un manque, voire d'une absence, d'outils de traitement automatique de la langue.

D'autre part, le tamahaqt se distingue par une morphologie riche en termes de mots et de combinaisons de mots, ce qui complique le traitement automatique. Ces défis nécessitent des efforts de recherche considérables pour développer des outils efficaces de traitement automatique du tamahaqt. La mise au point de ces outils pourrait jouer un rôle significatif dans la préservation et la promotion du dialecte tamahaqt et de la culture amazighe.

Conclusion

Notre étude a démontré la difficulté et la complexité de le dialecte Tamahaqt. nous avons abordé la langue amazighe en mettant la lumière sur les origines et ça géographie et les différent variante de cette langue et analysant les défis uniques liés à la complexité de cette langue et de son système d'écriture, nous mettons l'importance de développer des systèmes de reconnaissance automatique des caractères latins en Tamahaqt.

Nous avons également présenté l'organisme d'accueil, le Centre de Recherche en Langue et Culture Amazighes (CRLCA), qui joue un rôle crucial dans notre recherche. Nous avons également formulé la problématique de notre recherche, qui concerne le développement d'outils pour le traitement automatique des caractères tamahaqt dans la langue amazighe.

Ces efforts visent à préserver et à valoriser cette langue ancienne et précieuse, ouvrant ainsi la voie à une meilleure utilisation et préservation de l'identité culturelle berbère. Ce travail facilitera la communication avec le peuple touareg et sera utile pour ceux qui souhaitent apprendre la langue Tamahaqt.

3

Création de corpus et méthodologie

Introduction

Pour créer un bon corpus en tamahaqt, le Centre de Tamazight de l'Université de Bejaia a mis à notre disposition des mémoires et des livres sur ce dialecte. Ils nous ont également fourni l'unicode Tifinagh, connu sous le nom de Tifinagh Hamza FOFO 2021 ¹.

La création du corpus linguistique demande des exigences compatibles et des recherches approfondies. Dans notre cas, nous avons analysé des mémoires de fin d'études et des livres en tamahaqt afin de créer notre base de données. Nous avons choisi les types de caractères à inclure, ainsi que les formats et les annotations nécessaires.

Sommaire

Introduction	32
3.1 Corpus linguistique	33
3.2 Méthodologie et outils de création de corpus	33
3.3 Prétraitement de l'image	39
3.4 Représentation de modèle CNN	45
Conclusion	53

1. Tifinagh Hamza FOFO 2021 est une police Tifinagh que nous avons utilisée pour la transcription des caractères tamahaqt dans notre data-set linguistique.

Ce chapitre va aider à comprendre la procédure de création de notre corpus tamahaqt. Dans un premier temps, nous avons défini ce qu'est un corpus linguistique et les outils de création que nous avons utilisés, ainsi que les environnements et les bibliothèques que nous utilisons. Ensuite, nous avons exploré le prétraitement d'images et le modèle CNN en détail.

3.1 Corpus linguistique

Un corpus linguistique est défini comme un ensemble homogène et significatif de données linguistiques observées, à partir desquelles la description et la formalisation des faits linguistiques peuvent être élaborées.

Selon Sinclair (1996), la communauté linguistique considère que : *un corpus est une collection de données langagières qui sont sélectionnées et organisées selon des critères linguistiques explicites pour servir d'échantillon de langage* [17].

Dans un contexte plus général, un corpus désigne un ensemble organisé de documents ou de données, souvent numérisés, dans un domaine spécifique. Sa définition et sa manipulation nécessitent une réflexion approfondie pour en tirer tout le potentiel analytique[15, 13].

3.1.1 Caractéristique

Un corpus est caractérisé par différentes caractéristiques, parmi lesquelles on peut citer ses points forts.

- Un ensemble organisé de documents ou de données.
- Souvent numérisé pour faciliter l'accès et la manipulation.
- Employé dans le cadre de recherches historiques.
- Nécessitant une réflexion approfondie pour en tirer tout le potentiel analytique[13].

3.2 Méthodologie et outils de création de corpus

3.2.1 Procédures de collecte et d'analyse des données

Pour créer un corpus linguistique, il est crucial de bien choisir et appliquer les méthodes de collecte et d'analyse des données[17].

La première étape consiste en une planification globale de l'évaluation. Dans cette étape, nous avons évalué le livre intitulé (A Grammatical Sketch of the Tamaheq or Towarek Language), ce livre contient les règles et toutes les bases de la langue, ainsi que des mémoires de fin d'études écrits en langue tamahaqt.

Avant de commencer la collecte des données pour notre corpus linguistique, nous avons dû déterminer les objectifs et les questions clés de notre recherche. En outre, la théorie du changement illustre les liens entre les intrants, qui sont les mémoires et les livres en tamahaqt, les

activités de collecte et les résultats obtenus, y compris des résultats imprévus. Pour assurer la qualité et l'adéquation des données, nous avons pris soin de sélectionner les données à inclure, ainsi que les formats et les annotations nécessaires[23, 24].

Chaque lettre de l'alphabet tamahaqt est représentée par cinq images différentes : une en écriture normale, une en manuscrite, une en italique, une en gras et une en gras italique, toutes annotées de manière appropriée, par exemple de A à A4 pour la lettre A, couvrant diverses variations d'écriture.

Le processus de collecte pour cette base de données a comporté plusieurs étapes clés :

- **Sélection des caractères** : La sélection des caractères qui identifient 28 lettres de l'alphabet Tamahaqt,
- **Création des Images** : L'image des lettres a été créée suivant cinq styles différents : normale, manuscrite, gras, italique et gras souligné et, validation.
- **Vérification et Validation** : Les images ont été vérifiées pour corroboration entre l'original et sa copie s'assurant de la qualité et de l'exactitude de la représentation.
- **Organisation des Données** : Les données ont été organisées et cataloguées systématiquement pour faciliter leur accès et leurs utilisation lors des prochaines analyses linguistiques.

Pour l'analyse des données, plusieurs méthodes peuvent être employées :

- **Analyse Quantitative** : Évaluation de la fréquence d'utilisation de chaque style d'écriture et de chaque lettre, mot, et phrase dans différents contextes linguistiques.
- **Analyse Qualitative** : Examen des variations dans les styles d'écriture pour comprendre les implications culturelles et linguistiques de la langue Tamahaqt.
- **Triangulation des Données** : Utilisation de multiples sources de données pour corroborer les résultats et garantir leur validité.

En intégrant ces procédures de collecte et d'analyse des données, notre base de données linguistique vise à fournir une ressource exhaustive et fiable pour l'étude et la préservation de la langue Tamahaqt.

Après avoir élaboré notre corpus sur Access, nous l'avons converti en un fichier Excel afin de rendre l'utilisation de nos données plus pratique.

A	B	C	D	E
ID	Transcriptior	Image		
1	•	A.jpg;A1.png;A2.png;A3.png		
2	⊖	B.jpg;B1.png;B2.png;B3.png		
3	⊗	C.jpg;C1.png;C2.png;C3.png		
4	∧	D.jpg;D1.png;D2.png;D3.png		
5	E	Dd.jpg;Dd1.png;Dd2.png;Dd3.png		
6	I	F.jpg;F1.png;F2.png;F3.png		
7	×	G.jpg;G.png;G1.png;G3.png		
8	T	Gg.jpg;Gg1.png;Gg2.png;Gg3.png		
9	⊖	Gh.jpg;GH1.png;Gh2.png;GH3.png		
10	⊖	H.jpg;H1.png;H2.png;H3.png		
11	⊖	I.jpg;i1.png;i2.png;i3.png		
12	×	J.jpg;J1.png;J2.png;J3.png		
13	∴	K.jpg;K1.png;K2.png;K3.png		
14		L.jpg;L1.png;L5.png;L7.png		
15		L2.png;L3.jpg;L4.png;L6.png		
16	C	M.jpg;M1.png;M2.png;M3.png		
17	l	N.jpg;N1.png;N2.png;N3.png;N4.png		
18	...	Q.jpg;Q1.png;Q2.png;Q3.png		
19	○	R.jpg;R.png;R1.png;R2.png		
20	⊖	S.jpg;S1.png;S2.png;S3.png		
21	#	Ss.jpg;Ss1.png;Ss2.png;Ss3.png		
22	+	T.jpg;T1.png;T2.png;T3.png		
23	E	Tt.jpg;Tt1.png;Tt2.png;Tt3.png		
24	-	U.jpg;U1.png;U2.png;U3.png		

FIGURE 3.1 – Présentation des données sous Excel

3.2.2 Microsoft Access

Microsoft Access est un logiciel de gestion de base de données relationnelle (SGBD). Il offre un ensemble d'outils permettant de saisir, de mettre à jour, de manipuler, d'interroger et d'imprimer des données[20].

Nous avons utilisé Microsoft Access pour créer notre base de données d'images. Pour cela, nous avons suivi plusieurs étapes de création. Nous avons commencé par créer une base de données vide, sans données ni objets existants, à laquelle nous avons donné un nom.

Ensuite, nous avons créé une table nommée (Tamahageth). Nous avons complété les champs et les types de données en attribuant une clé primaire (ID) avec le type de données (NuméroAuto); Transcription, un champ de type (Texte); et Image, un champ de type (Pièce jointe).

Une fois la table créée, nous avons utilisé la police Unicode Tifinagh Hamza FOFO 2021 pour remplir les lettres de la langue tamahaqt et ajouté les images correspondantes pour chaque lettre. Pour ajouter des images nous avons ouvert la table, ajouté une nouvelle entrée, renseigné les champs appropriés et ajouté des images pour chaque champ de type (Pièce jointe).

Enfin, nous avons sauvegardé régulièrement notre base de données. En suivant ces étapes, nous avons obtenu une base de données prête à être utilisée [23, 24].

Character	Count
•	(4)
2 Θ	(4)
3 ⊗	(4)
4 ∧	(4)
5 E	(4)
6 I	(4)
7 X	(4)
8 T	(4)
9 ≡	(4)
10 ≡	(4)
11 ≡	(4)
12 X	(4)
13 :	(4)
14	(4)
15 N	(4)
16 C	(4)
17 I	(5)
18 ...	(4)
19 O	(4)
20 O	(4)
21 #	(4)
22 +	(4)
23 E	(4)
24 :	(4)

FIGURE 3.2 – Présentation de corpus

3.2.3 Environnement de développement et Bibliothèques

3.2.3.1 Python

Python est un langage de programmation de haut niveau interprété et orienté objet, apprécié pour sa simplicité et sa sémantique dynamique. Il est facile à apprendre, réduit les coûts de maintenance des codes, et encourage la modularité et la réutilisabilité via ses nombreuses bibliothèques, disponibles gratuitement sur la plupart des plateformes[38].



FIGURE 3.3 – Logo de python

3.2.3.2 Anaconda

Anaconda est une distribution Python, une collection de paquets préconstruits et préconfigurés couramment utilisée pour la science des données. La distribution Anaconda inclut le gestionnaire de paquets Conda en plus des paquets Python préconfigurés et d'autres outils[4].



FIGURE 3.4 – Logo de Anaconda

3.2.3.3 Jupyter

Le projet Jupyter se compose de divers standards, d'une communauté dynamique et de multiples outils logiciels. Un Jupyter Notebook est un document interactif intégrant du code exécutable, des équations, des visualisations et du texte explicatif.

Il permet de combiner des données, du code et du texte pour créer des narrations computationnelles interactives. Employé dans de nombreux domaines, il facilite l'analyse de données, l'illustration de concepts techniques et scientifiques, et favorise une exploration interactive et collaborative. Jupyter est une plateforme open-source, gratuite et accessible à partir d'un navigateur web moderne.[28].



FIGURE 3.5 – Logo de Jupyter

3.2.3.4 TensorFlow

TensorFlow est un framework de programmation pour le calcul numérique qui a été rendu Open Source par Google en Novembre 2015.

Il est particulièrement connu pour son utilisation dans le développement et le déploiement de modèles de machine learning et de deep learning.

Son nom est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multi-dimensionnelles, appelées Tenseurs (Tensor)[38].



FIGURE 3.6 – logo de TensorFlow

3.2.3.5 Keras

Keras est une API (Application Programming Interface) de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow ou Theano.

Keras a été développé pour favoriser l'expérimentation rapide en machine learning, sous la direction de François Chollet dans le cadre du projet ONEIROS.

En 2017, l'équipe TensorFlow de Google a intégré Keras comme interface principale, offrant des abstractions intuitives pour configurer les réseaux neuronaux, indépendamment de la bibliothèque de backend[38].

Nous allons employer les modèles suivants de Keras pour concevoir des réseaux de neurones :

- **Sequential** : Pour créer des modèles linéaires couche par couche.
- **Flatten** : est une couche qui transforme les données de sortie de la couche précédente en un vecteur unidimensionnel.
- **Dropout** : est une technique de régularisation qui consiste à ignorer de manière aléatoire un certain pourcentage de neurones pendant l'entraînement.
- **Dense** : est une couche de neurones entièrement connectée où chaque neurone de la couche précédente est connecté à chaque neurone de cette couche.
- **MaxPooling2D** : est une couche de pooling qui réduit la dimensionnalité des données.
- **Conv2D** : est une couche de convolution bidimensionnelle qui applique des filtres convolutionnels sur les entrées 2D.



FIGURE 3.7 – Logo de Keras

3.2.3.6 pyodbc

pyodbc est une bibliothèque Python utilisée pour l'accès aux bases de données via l'interface ODBC (Open Database Connectivity). Cette bibliothèque est largement employée pour l'exécution de requêtes SQL sur divers systèmes de gestion de bases de données comme SQL Server, Oracle, MySQL, entre autres[36].

3.2.3.7 Pandas

Pandas est une librairie Python qui propose des structures de données rapides, flexibles et expressives, facilitant ainsi la manipulation des données "relationnelles" ou "étiquetées".

Elle a pour objectif de devenir le principal outil de haut niveau pour les analyses de données pratiques et concrètes en Python. De plus, elle ambitionne de devenir l'outil open source le plus puissant et polyvalent pour l'analyse et la manipulation des données, toutes langues confondues. Pandas est adapté à un large éventail de types de données[5].

3.2.3.8 Base64

Base64 est un module Python utilisé pour effectuer des opérations d'encodage et de décodage en utilisant le standard Base64. Ce procédé est fréquemment employé pour transformer des données binaires en une chaîne de caractères ASCII, simplifiant ainsi leur manipulation, leur transfert ou leur stockage sous forme de texte[36].

3.2.3.9 Numpy

NumPy est une bibliothèque Python conçue pour fournir des tableaux (arrays) nettement plus rapides que les listes Python traditionnelles, avec des performances allant jusqu'à 50 fois plus rapides.

Le tableau en NumPy, appelé ndarray, est un objet qui simplifie considérablement le travail grâce à de nombreuses fonctions de support[5].

3.2.3.10 Binascii

Binascii est une bibliothèque Python qui facilite les conversions entre les données binaires et leurs représentations ASCII. Elle est couramment employée pour des opérations telles que l'encodage et le décodage en base64, ainsi que pour d'autres manipulations de données binaires[36].

3.2.3.11 OpenCV

OpenCV (cv2 en Python) est une bibliothèque bien connue et open-source largement adoptée pour le traitement d'images et la vision par ordinateur. Elle offre une gamme complète d'outils pour des activités comme la détection de visages, le suivi d'objets, la reconnaissance de formes, et diverses autres applications dans ce domaine[36].

3.3 Prétraitement de l'image

Le prétraitement désigne une série de procédures appliquées aux données brutes pour les préparer à une analyse thématique approfondie. Cette étape essentielle inclut des opérations générales de traitement d'image, telles que la réduction du bruit et la binarisation, ainsi que des méthodes spécifiquement adaptées à l'analyse de l'écriture manuscrite. Les techniques de prétraitement jouent un rôle crucial en réduisant la variabilité au sein des mêmes catégories grâce à des processus de normalisation standardisés[42].

Représentation informatique de l'image

Une image peut être décrite comme une matrice bidimensionnelle où chaque élément, ou pixel, représente une petite surface de l'image. Ces pixels sont généralement organisés en lignes et en colonnes. Mathématiquement, une image se définit comme une fonction à deux dimensions $f(x, y)$, où x et y sont les coordonnées spatiales, et l'amplitude en chaque point (x, y) correspond à l'intensité ou au niveau de gris.

Lorsque les coordonnées (x, y) et l'amplitude sont discrétisées, on parle alors d'image numérique ou digitale. Dans ce cas, la fonction f est souvent remplacée par la lettre I , et les coordonnées (x, y) deviennent le couple (i, j) [42].

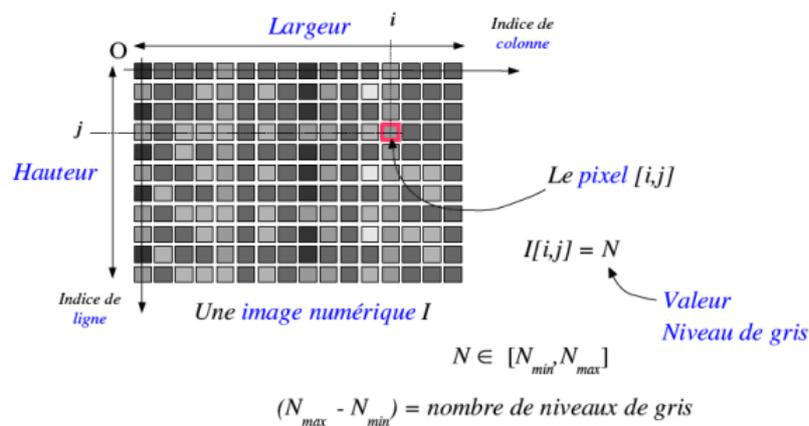


FIGURE 3.8 – Représentation des notions image et pixel

3.3.1 Caractéristiques de représentation informatique d'image

L'image est un ensemble structuré d'information caractérisé par les paramètres suivants :

- Pixel :** Le terme pixel est dérivé de l'expression britannique picture element. Il représente l'unité de base d'une image. L'ensemble de ces pixels, disposés dans un tableau bidimensionnel, forme l'image finale. Chaque pixel correspond à un niveau de gris ou à des niveaux de composantes couleur, codés sur N bits, représentant respectivement la luminosité ou la couleur de la zone correspondante de la scène observée. Les pixels sont localisés par leurs coordonnées x et y .
 Un pixel est généralement rectangulaire ou presque carré, avec une taille variant entre 0,18 mm et 0,66 mm de côté[42].
- Résolution :** La résolution d'une image est le nombre de pixels contenus dans l'image par unité de longueur.
 Elle s'exprime le plus souvent en PPP (Points Par Pouce) ou en DPI (Dots Per Inch), et parfois en points par centimètre. La résolution définit la qualité d'une image ; plus la résolution est élevée, plus l'image est précise dans les détails[42].

- **Bruit** : Le bruit est un phénomène parasite aléatoire, qui peut suivre une distribution de probabilité connue ou inconnue. Il correspond à des perturbations pouvant provenir soit du dispositif d'acquisition, soit de la scène observée elle-même[42].
Les sources de bruit dans une image sont nombreuses et variées :
 - Bruits liés aux conditions de prise de vue (bougé, éclairage de la scène)
 - Bruits liés aux capteurs (appareil numérique de bas de gamme)
 - Bruits liés à l'échantillonnage
 - Bruits liés à la nature de la scène (poussières, rayures).
- **Contour** : Un contour dans une image est défini comme la frontière entre deux pixels (ou groupes de pixels) où il y a une différence notable dans le niveau de gris ou de couleur.
Cela représente un changement marqué d'intensité ou une discontinuité dans les caractéristiques entre deux régions adjacentes. Les contours sont cruciaux pour l'interprétation des formes et la détection des objets dans le traitement des images[42].
- **Dimension** : La dimension d'une image correspond à sa taille, représentée sous forme de matrice dont les éléments sont des valeurs numériques.
Le nombre total de pixels dans l'image est obtenu en multipliant le nombre de lignes par le nombre de colonnes de cette matrice [7].
- **Luminance** : La luminance mesure l'intensité lumineuse émise par une source lumineuse étendue dans une direction spécifique, rapportée à l'aire apparente de cette source vue depuis cette direction.
Elle est exprimée en candelas par mètre carré, dont le symbole est (cd/m²). Cette mesure est essentielle pour caractériser la quantité de lumière qui est perçue par l'œil humain provenant d'une certaine direction [7].
- **Voisinage** : Le plan de l'image est subdivisé en formes rectangulaires ou hexagonales, ce qui facilite l'utilisation du concept de voisinage. Le voisinage d'un pixel est composé de tous les pixels qui l'entourent. On définit également l'assiette comme l'ensemble de pixels qui détermine le voisinage pris en compte autour d'un pixel [42].

Il existe deux types de voisinage :
 - Voisinage à 4 : Seuls les pixels ayant un côté commun avec le pixel considéré sont pris en compte.
 - Voisinage à 8 : Tous les pixels ayant au moins un point de contact avec le pixel considéré sont pris en compte.
- **Contraste** : Le contraste est l'écart perceptible entre deux éléments distincts, où l'un met en évidence l'autre. Il est mesuré par un coefficient qui reflète la capacité de l'œil à distinguer les différences d'intensité lumineuse entre deux zones adjacentes sur une image.

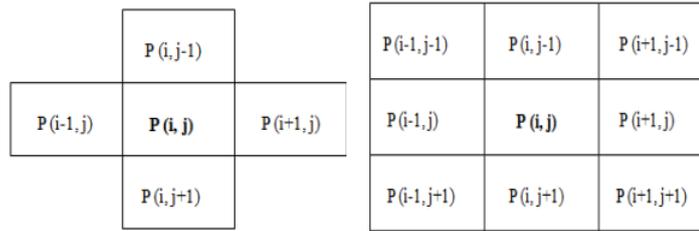


TABLE 3.1 – Voisinage à 4, Voisinage à 8

En termes simples, le contraste d'une image révèle l'écart entre les tons les plus lumineux et les plus sombres. Une image avec un contraste élevé présentera des zones de blanc pur et de noir profond, tandis qu'une image avec un faible contraste aura une grande variété de tons gris, sans atteindre les extrêmes du noir et du blanc.

Mathématiquement, si $L1$ et $L2$ représentent les niveaux de luminosité de deux zones adjacentes $A1$ et $A2$ dans une image, le contraste C peut être exprimé par la formule suivante :

$$C = \frac{L1 - L2}{L1 + L2}$$

où C est le contraste entre les deux zones. Cette formule permet de quantifier la différence de luminosité pour évaluer le contraste visuel de l'image[42].

3.3.2 Types de représentation informatique d'image

- **binaire**

Dans une image binaire, chaque pixel est limité à deux états possibles : il peut soit prendre la valeur de 0, soit celle de 1. Cette catégorie d'image peut être conceptualisée comme une matrice booléenne, où chaque élément correspond à un pixel qui est soit en état Vrai (représentant le blanc avec la valeur 1), soit en état Faux (représentant le noir avec la valeur 0) [42].

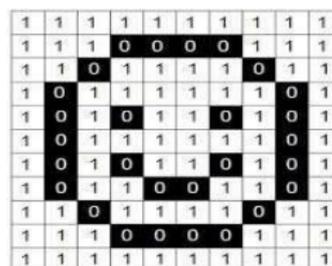


FIGURE 3.9 – Exemple d'une image binaire

- **Image couleur RGB**

Dans le modèle de couleur RGB chaque pixel a une couleur définie par la quantité de rouge (R), de vert (G) et de bleu (B). Chacune de ces trois composantes est codée sur une échelle allant de 0 à 255, ce qui permet d'obtenir $256^3 = 16777216$ couleurs possibles [7].

- **Image en niveau de gris**

Dans une image en niveaux de gris, chaque pixel est représenté par une valeur entière comprise entre 0 et 255. La valeur 0 est utilisée pour le noir, la valeur 255 pour le blanc, et les valeurs intermédiaires représentent différents niveaux de gris, allant du noir au blanc [7].

256 niveaux de gris

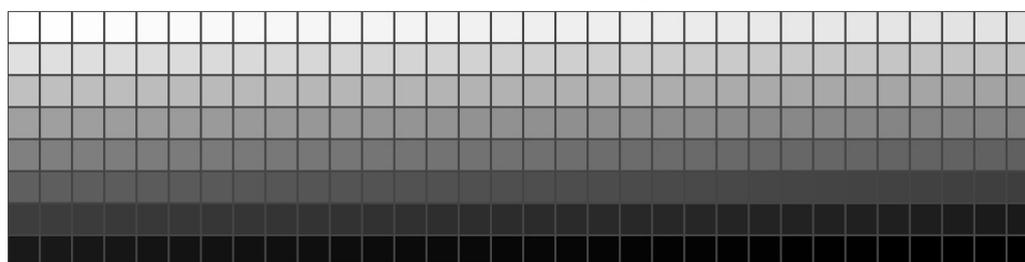


FIGURE 3.10 – Les différentes couleurs en niveau de gris

- **L'image matricielle (ou bitmap)**

L'image matricielle, également appelée bitmap, est constituée de minuscules points nommés « pixels » que l'on ne perçoit pas à l'œil nu. Lorsqu'on agrandit une image de ce type, elle devient floue car les pixels deviennent visibles, apparaissant sous forme de petits carrés à l'écran [42].

- **L'image vectorielle**

Elle est constituée de segments de lignes reliés entre eux par des formules mathématiques. Ce système repose sur la proportionnalité et les coordonnées. Grâce à la vectorisation, chaque élément occupe une position précise, ce qui évite toute déformation de l'image lors de son redimensionnement [42].

3.3.3 Méthodes du prétraitement

Avant que l'image ne soit traitée, l'image source doit d'abord être prétraitée. Il s'agit d'un processus visant à améliorer la qualité de l'image pour faciliter l'extraction des informations nécessaires.

Au fur et à mesure que les progrès en informatique se sont accélérés, le traitement d'images est devenu de plus en plus courant grâce à son utilisation dans divers domaines, notamment la reconnaissance d'objets, l'imagerie satellitaire et médicale. La combinaison de plusieurs techniques telles que le redimensionnement, la réduction du bruit, la binarisation et la normalisation de l'image est susceptible de donner des résultats optimaux. [29].

3.3.3.1 Chargement d'image

Le chargement d'images est une étape fondamentale dans le traitement d'images. Pour gérer les chemins de fichiers et de dossiers on a utilisé deux outils pratiques, notamment (`os.listdir`) et le module (`pathlib`).

- **os.listdir** Fournit une méthode simple pour lister tous les fichiers et dossiers dans un répertoire spécifié.
- **pathlib** Offre une interface orientée objet pour manipuler les chemins de fichiers et de dossiers de manière plus intuitive et lisible.

3.3.3.2 Redimensionnement de l'image

Le redimensionnement de l'image permet d'ajuster la taille de l'image à des dimensions spécifiques.

Dans notre cas on a redimensionné nos images à 200x200 pixels ce qui est utile pour normaliser les images avant de les utiliser dans des algorithmes de machine learning, des réseaux de neurones, ou pour les afficher de manière uniforme.

3.3.3.3 Binarisation de l'image

La binarisation c'est de convertir une image en niveaux de gris en une image binaire. Une image binaire contient seulement deux valeurs de pixel 0 (noir) et 255 (blanc). Cela simplifie l'image en isolant les objets d'intérêt (comme le texte) du fond.

3.3.3.4 Réduction du bruit

La réduction du bruit dans une image est une étape essentielle pour améliorer la qualité et la précision des traitements ultérieurs dans la reconnaissance de caractères.

Dans cette étape, des filtres de lissage sont utilisés, dans notre cas, on a utilisé le filtre gaussien.

3.3.3.4.1 Fonctionnement du Filtre Gaussien Le filtre gaussien est un filtre passe-bas qui lisse l'image en réduisant les variations brusques d'intensité entre les pixels.

Il utilise une matrice de filtrage basée sur une fonction gaussienne pour effectuer la convolution avec l'image d'origine[36].

3.3.3.4.2 Étape du Filtre Gaussien

- **Matrice de Filtrage :** Généralement une matrice de taille 3x3, 5x5, ou 7x7 est utilisée pour l'application de filtre gaussien cette taille soit déterminée par le paramètre de la taille de la fenêtre (`t_fen`) cette matrice est calculée à partir d'un vecteur de filtrage, obtenu en appliquant la fonction gaussienne.
- **Calcul des Coefficients de la Matrice de Filtrage :** On dérive les coefficients de matrice de filtrage en multipliant la transposée du vecteur de filtrage par lui-même.

On calcule le vecteur de filtrage par l'équation suivante :

$$v_{\text{filtr}} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

où σ est l'écart type et μ est la moyenne.

- **Convolution** : Chaque pixel de l'image est transformé en multipliant ses valeurs ainsi que celles de ses pixels voisins par les coefficients de la matrice de filtrage. Les produits obtenus sont ensuite additionnés et divisés par la somme de tous les coefficients de la matrice.
- **Gestion des Bords** : On remplit les bords de l'image avec des zéro ou on dupliqués en raison des erreurs qui apparaître aux marge de l'image causé par l'étape de la convolution[36].

3.3.3.5 Détection des lignes avec la transformée de Hough (Hough Line Transform)

La détection des lignes dans une image est une étape clé dans de nombreux processus de traitement d'images, notamment pour l'analyse de documents. La transformée de Hough est une technique utilisée pour détecter des lignes dans une image en transformant les pixels de l'image en un espace de paramètres[35].

3.3.3.6 Squelettisation

L'image squelettique est une technique qui réduit l'épaisseur et la largeur des objets et des lignes représentés dans un certain cadre tout en maintenant la structure topologique critique [46].

Le squelettisation est définie mathématiquement en termes de la notion de squelette, ce qui peut différer en fonction du contexte d'utilisation et des images données, y compris binaire, où seules les valeurs de pixel noir et blanc sont autorisées, et en niveaux de gris.[46]

3.3.3.7 Normalisation

La normalisation d'image consiste à ajuster les valeurs d'intensité des pixels d'une image dans une plage spécifique, souvent entre 0 et 255 pour les images en niveaux de gris de 8 bits. En appliquant cette étape, on améliore le contraste de l'image en étalant la gamme des valeurs d'intensité des pixels, ce qui rend également l'image plus adaptée aux algorithmes de traitement[31].

3.4 Représentation de modèle CNN

Un CNN (Convolutional Neural Network), ou réseau de neurones convolutifs en français, désignés par l'acronyme CNN, est un type de réseau de neurones artificiels particulièrement efficace pour les tâches de reconnaissance d'images et de traitement d'images.

Ils se composent de deux parties distinctes.

- La première partie, convolutive, extrait des caractéristiques d'une image en la faisant passer à travers des filtres pour créer des cartes de convolutions. Ces cartes sont ensuite mises à plat et concaténées en un vecteur de caractéristiques (code CNN).
- Le code précédent est ensuite utilisé en entrée de la deuxième partie, composée de couches entièrement connectées, qui combine les caractéristiques pour classer l'image. La sortie finale donne une distribution de probabilité sur les catégories[38].

3.4.1 Utilisation des CNN pour la Reconnaissance de Caractères

Les CNN sont particulièrement efficaces pour la reconnaissance de caractères grâce à leur capacité à extraire automatiquement des caractéristiques complexes des images.

- **Prétraitement** : On a prétraitées les images des caractères de notre corpus pour standardiser la taille et améliorer la qualité de l'image (binarisation, redimensionnement, etc.).
- **Entraînement** : Un large jeu de données de caractères étiquetés est utilisé pour entraîner le CNN. Le réseau apprend à reconnaître les motifs distinctifs de chaque caractère à travers des milliers d'exemples.
- **Prédiction** : Une fois entraîné, le modèle CNN peut prendre une image de caractère comme entrée et prédire avec une certaine probabilité à quel caractère elle correspond.

3.4.2 Architecture de CNN

Les réseaux de neurones convolutionnels (CNN) sont des modèles inspirés du fonctionnement du cortex visuel des vertébrés. Ils sont très efficaces pour le traitement d'images. Cependant, les perceptrons multicouches (MLP), qui sont la base des CNN, rencontrent des difficultés avec les images de grande taille.

Pourquoi ? Imaginons une image de dimensions 32x32x3 (32 pixels de large, 32 pixels de haut, 3 canaux de couleur). Dans un réseau MLP, un seul neurone dans la première couche cachée aurait 3072 entrées ($32 \times 32 \times 3$). Si l'image est plus grande, comme une image de 200x200, chaque neurone devrait traiter 120 000 entrées ! Cela devient rapidement problématique lorsque l'on multiplie ce nombre par le nombre total de neurones dans le réseau.

Les CNN résolvent ce problème en utilisant des couches de convolution. Au lieu de connecter chaque pixel à chaque neurone, les CNN utilisent des filtres pour extraire des caractéristiques importantes de l'image. Ces filtres sont appliqués localement, ce qui réduit considérablement le nombre de connexions nécessaires. Ainsi, les CNN sont bien adaptés pour gérer des images de toutes tailles.

La figure 3.11 présente l'empilement de couches dans le réseau de neurone convolutif, afin de reconnaître le caractère. [2] .

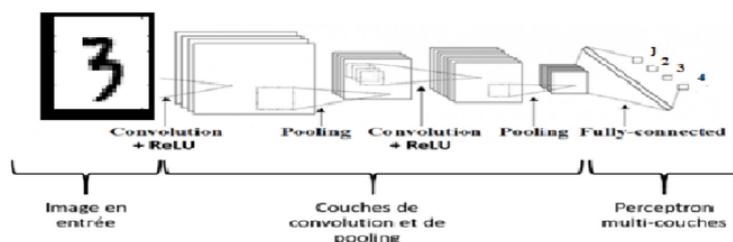


FIGURE 3.11 – Exemple d'Architecture standard de CNN

Une architecture CNN se compose d'une série de couches de traitement distinctes[37] :

- La couche de convolution (CONV) qui analyse les données d'une région réceptrice spécifique.
- La couche de pooling (POOL) qui compresse les informations en diminuant la taille de l'image intermédiaire, généralement par sous-échantillonnage.
- La couche d'activation ReLU, souvent désignée simplement sous le nom de ReLU, en référence à la fonction d'activation Rectified Linear Unit.
- La couche entièrement connectée (FC), qui fonctionne comme une couche de perceptron.
- La couche de perte (LOSS).

3.4.2.1 La couche de convolution (CONV)

Lorsqu'une nouvelle image est présentée à un réseau de neurones convolutif (CNN), celui-ci ne sait pas exactement si les caractéristiques spécifiques seront présentes dans l'image ni où elles pourraient se trouver. Par conséquent, le CNN cherche à les détecter dans toute l'image, quelle que soit leur position.

Pour ce faire, il effectue un filtrage en calculant la présence de ces caractéristiques sur l'ensemble de l'image. Cette opération mathématique est appelée convolution, d'où le nom des réseaux de neurones à convolution.

La couche de convolution est un élément clé des réseaux de neurones convolutifs et constitue généralement leur première couche. Son objectif est de repérer la présence d'un ensemble de caractéristiques dans les images en entrée.

La fonction principale d'un réseau de neurones convolutif (CNN) est d'extraire des caractéristiques à partir d'images. Lorsqu'une image est soumise au CNN, elle passe par une série de filtres, également appelés noyaux de convolution.

Pour chaque paire (image, filtre), le CNN génère une carte d'activation, également connue sous le nom de feature map.

Cette carte indique où se trouvent les caractéristiques dans l'image : plus la valeur est élevée dans la feature map, plus l'endroit correspondant dans l'image ressemble à la caractéristique recherchée.

Enfin, les valeurs des dernières feature maps sont concaténées pour former un vecteur. Ce vecteur définit la sortie du premier bloc et l'entrée du bloc suivant [7].

Pour dimensionner une couche de convolution, trois hyperparamètres sont essentiels [42] :

1. **Profondeur de la couche (K) :** Le nombre de filtres (ou noyaux) choisis par couche de convolution détermine la profondeur de la sortie associée à la convolution.
2. **Pas (S) :** Lors de la convolution, le pas (stride) détermine le déplacement du noyau à travers l'entrée. Le pas horizontal représente le déplacement du noyau horizontalement dans l'image, tandis que le pas vertical représente le déplacement du noyau verticalement. Pour simplifier, nous utiliserons un pas horizontal égal au pas vertical.
3. **La marge (à 0) ou zéro padding :** Parfois, il est pratique d'ajouter des zéros autour du volume d'entrée, ce que l'on appelle le 'zéro-padding'. Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est souhaitable de conserver la même surface que celle du volume d'entrée.

Lorsque nous appliquons un pas et une marge à l'image d'entrée pour gérer le nombre de champs récepteurs (surface de traitement), la profondeur devient un paramètre important.

De manière similaire à la façon dont une image a une profondeur (par exemple, 3 canaux RGB pour une image couleur), la couche de convolution en sortie présente également une profondeur. C'est pourquoi nous parlons de "volume de sortie" et de "volume d'entrée", car l'entrée d'une couche de convolution peut être soit une image, soit la sortie d'une autre couche de convolution (voir Figure 3.12).

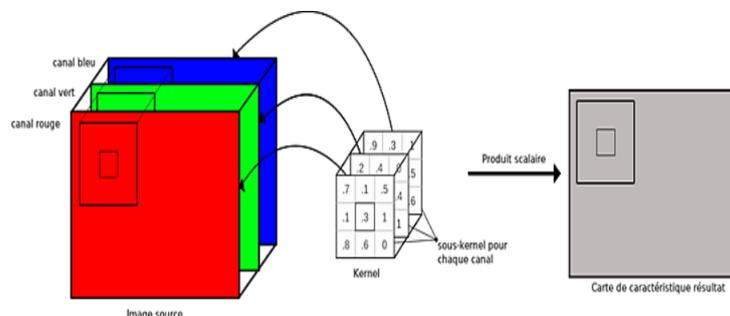


FIGURE 3.12 – Exemple d'image RGB en couche de convolution

La taille spatiale du volume de sortie (W_0) peut être calculée en fonction de la taille du volume d'entrée (W_i), de la surface de traitement k (nombre de champs récepteurs),

du pas S avec lequel ils sont appliqués et de la taille de la marge P .

La formule pour calculer le nombre de neurones dans le volume de sortie est la suivante :

$$W_0 = \frac{w_i - k + 2P}{S} + 1 \tag{3.1}$$

Si le résultat n'est pas un nombre entier, les neurones périphériques n'auront pas autant d'entrées que les autres.

Dans ce cas, il est nécessaire d'augmenter la taille de la marge (pour recréer des entrées virtuelles). Souvent, on considère un pas $S = 1$, ce qui nous permet de calculer la marge comme suit :

$$P = \frac{K - 1}{S} \tag{3.2}$$

Si nous souhaitons un volume de sortie de même taille que le volume d'entrée, la couche est dite "localement" (voir Figure 3.13).

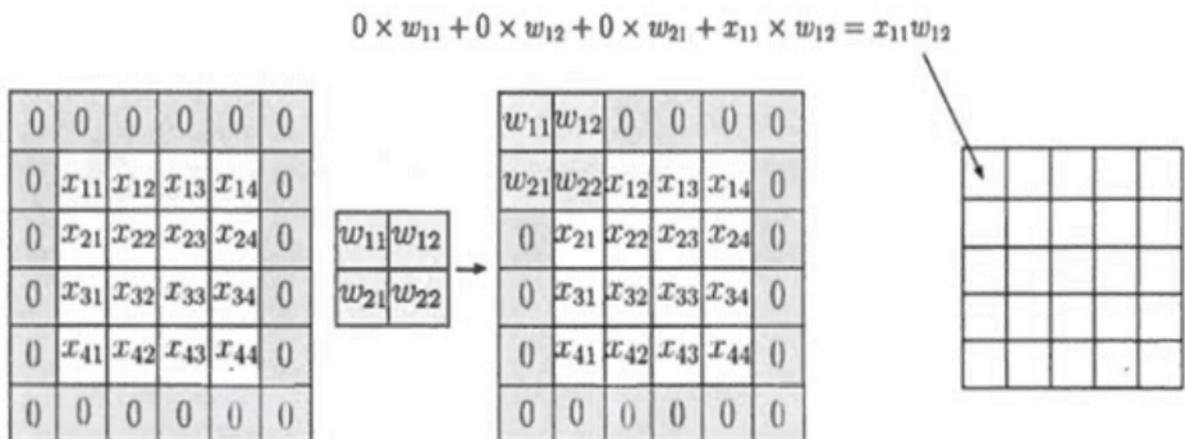


FIGURE 3.13 – Exemple d'entrant 4 x 4 x 1 avec couche de marge à zéro de taille 1 auquel on applique un noyau 2 x 2 x 1.

Par exemple, si nous avons un volume d'entrée de $4 \times 4 \times 1$ avec une marge de zéro de taille 1, et que nous appliquons un noyau de taille $2 \times 2 \times 1$, la sortie de notre convolution aura des dimensions de $5 \times 5 \times 1$.

Nous avons utilisé un pas de 1 pour les déplacements du noyau, ce qui donne les valeurs suivantes : $S = 1$, $K = 2$, $P = 1$ et $s = 1$."

3.4.2.2 Couche de pooling (POOL)

Les couches de pooling (POOL) sont des opérations de sous-échantillonnage couramment appliquées après une couche de convolution. Leur objectif est de réduire la taille des images, ce qui permet de diminuer le nombre de paramètres et de calculs dans le réseau. En conséquence, l'efficacité du réseau est améliorée et le risque de sur-apprentissage est réduit, tout en préservant les caractéristiques importantes des images.

Pour ce faire, on découpe l'image en cellules régulières, puis on conserve la valeur maximale au sein de chaque cellule. En pratique, on utilise souvent des cellules carrées de petite taille pour éviter de perdre trop d'informations. Les choix les plus courants consistent à utiliser des cellules adjacentes de taille (2×2) pixels qui ne se chevauchent pas [37].

Différents types de pooling sont utilisés en traitement d'image et dans les réseaux de neurones convolutifs (CNN) [2] :

- **Le max pooling** : consiste à sélectionner la valeur maximale dans une région donnée. C'est le type de pooling le plus couramment utilisé car il est rapide à calculer et permet d'effectuer une simplification efficace de l'image (voir la figure 3.14).

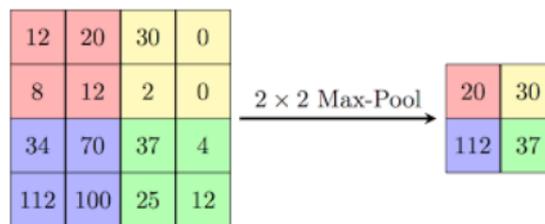


FIGURE 3.14 – représentation de maxpooling

- **Le mean pooling (ou pooling moyen) :** calcule la moyenne des valeurs des pixels dans la région sélectionnée. On additionne toutes les valeurs et on divise par le nombre de pixels pour obtenir une valeur intermédiaire représentant ce groupe de pixels (voir la figure 3.15).

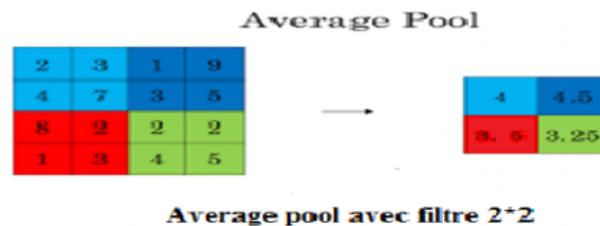


FIGURE 3.15 – représentation de meanpooling

- **Le sum pooling :** calcule simplement la somme des valeurs sans diviser par le nombre de pixels.

En général, le max pooling est recommandé, car il se distingue du mean pooling dans les cas extrêmes, mais il est presque équivalent au mean pooling dans les autres situations

3.4.2.3 Couches de correction (RELU)

Pour améliorer l’efficacité du traitement, on insère entre les couches de traitement une couche qui applique une fonction mathématique (fonction d’activation) aux signaux de sortie[42].

L’une de ces fonctions couramment utilisées est la fonction ReLU (Rectified Linear Units), définie comme suit :

$$ReLU(x) = \max(0, x) \tag{3.3}$$

La couche de correction ReLU remplace toutes les valeurs négatives en entrée par des zéros.

Cependant, elle n’affecte pas les données positives, préservant ainsi les caractéristiques mises en évidence par la convolution.

Au contraire, elle accentue ces caractéristiques en creusant l’écart entre les valeurs négatives et positives.

La fonctionnalité ReLU est illustrée à la, (Figure 3.16).

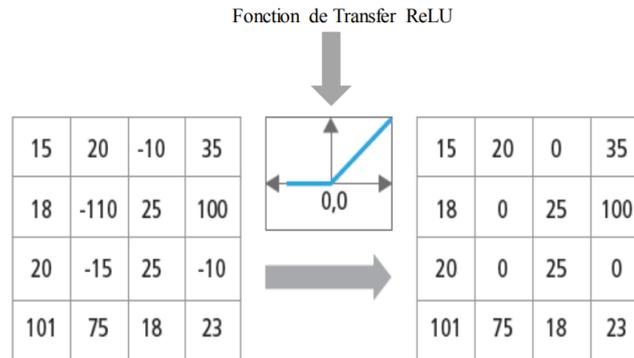


FIGURE 3.16 – Représentation graphique de la fonctionnalité Relu.

3.4.2.4 Couche entièrement connectée (FC)

Les cartes de caractéristiques résultantes des couches de convolution, de pooling et de ReLU sont finalement combinées pour former un vecteur de caractéristiques, appelé "code CNN". Ce code représente les informations essentielles extraites de l'image sous forme d'une carte 2D [42].

Ce code CNN, issu de la partie convolutive, est ensuite utilisé en entrée d'une deuxième partie du réseau. Cette partie est constituée d'une ou de plusieurs couches entièrement connectées (FC ou fully connected), qui peuvent être assimilées à un perceptron multicouches. Dans une couche entièrement connectée, chaque neurone est connecté à toutes les sorties de la couche précédente.

Leurs fonctions d'activation sont calculées en effectuant une multiplication matricielle entre le vecteur d'entrée H et la matrice des poids F , à laquelle on ajoute le vecteur des biais des neurones C :

$$A = FH + C \quad (3.4)$$

Le nombre de neurones dans la dernière couche entièrement connectée est égal au nombre de classes. Le rôle de cette partie est donc de combiner les caractéristiques extraites du code CNN pour classer une image.

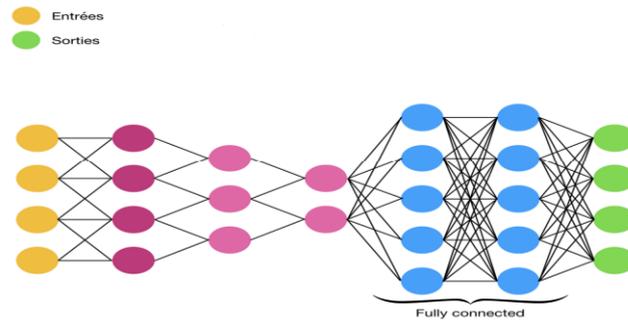


FIGURE 3.17 – Représentation de la couche entièrement connectée(FC)

3.4.2.5 Couche de perte (LOSS)

La couche de perte détermine comment l’entraînement du réseau pénalise la différence entre la prédiction et la valeur réelle. Elle se trouve généralement en dernière position dans l’architecture du réseau. Différentes fonctions de perte, adaptées à diverses tâches, peuvent y être appliquées. Par exemple, la fonction « Softmax » permet de calculer la distribution de probabilités pour les différentes classes de sortie [42].

Conclusion

Dans ce chapitre, nous avons examiné le corpus de la langue tamahaqt qui a été créé dans le chapitre précédent, ainsi que sa méthodologie de création et les outils que nous avons utilisés.

Nous avons détaillé les environnements et bibliothèques de notre travail, discuté du processus de prétraitement des images et des méthodes spécifiques employées.

Ensuite, nous avons abordé le modèle CNN en décrivant ses différentes couches et les choix de hyperparamètres que nous avons faits. Ce chapitre sert d’introduction au suivant, où nous détaillerons l’implémentation et présenterons les résultats obtenus.

4

Implémentation et résultats

Introduction

Après avoir traité du sujet de la tamahaqt en langue amazighe dans le deuxième chapitre, nous avons détaillé dans le chapitre précédent les Réseaux de neurones convolutifs (CNN), présenté notre corpus, ainsi que les différents outils d'implémentation, le langage et l'environnement de développement que nous allons utiliser.

Ce dernier chapitre est consacré à l'aspect expérimental du travail sur la reconnaissance automatique des caractères de la langue tamahaqt. Nous commençons par l'importation des packages nécessaires, ainsi que des données définies dans le chapitre précédent. Ensuite, nous effectuons le prétraitement des données, appliquons le modèle CNN sur les données prétraitées, puis passons à l'apprentissage du modèle. Enfin, nous réalisons la reconnaissance et les tests, et nous expliquons les résultats obtenus.

Sommaire

Introduction	54
4.1 Environnement matériel	54
4.2 Environnement du travail	55
4.3 Méthodologie de l'implémentation	55
Conclusion	62

4.1 Environnement matériel

- Type de PC : Lenovo

- Processeur : Intel(R) Core(TM) i7-4510UCPU @2.00GHz 2.60 GHz
- Mémoire : 8GO
- Système d'exploitation : Windows 8, processeur 64bits.

4.2 Environnement du travail

Dans un premier temps, nous avons ouvert l'outil Anaconda, comme présenté dans la figure 4.1. Nous avons cliqué sur l'onglet (Environments), puis sur le bouton (Create) afin de créer un environnement nommé (tensorflow). Nous l'avons nommé (tf-new) et avons sélectionné Python (car nous avons choisi de travailler avec ce langage). Ensuite, dans la barre de recherche, nous avons téléchargé les bibliothèques (TensorFlow, Keras, OpenCV) une par une. Cependant, il est important de s'assurer de choisir des versions des bibliothèques compatibles avec la version de Python utilisée. Après avoir créé l'environnement du travail et importé les packages

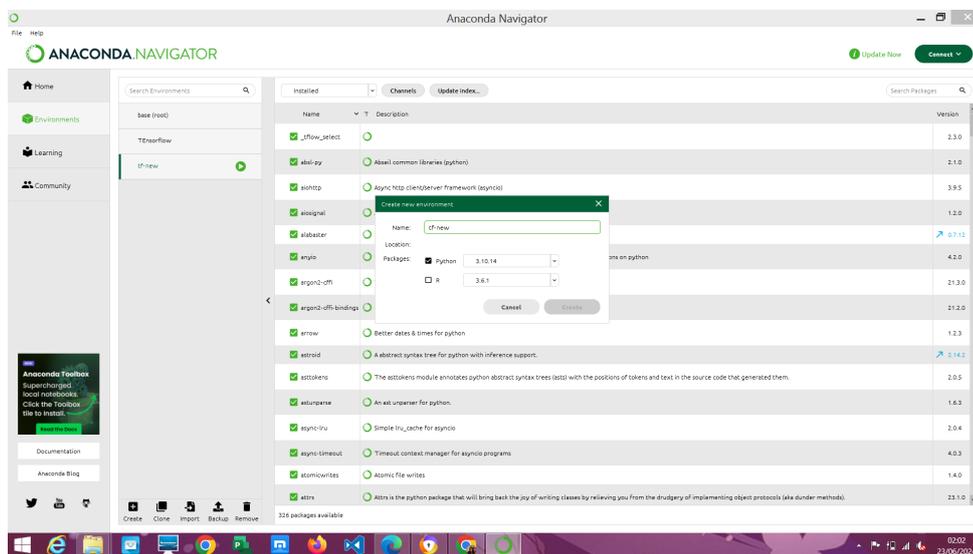


FIGURE 4.1 – représentation de meanpooling

nécessaires pour la reconnaissance, nous ouvrons l'outil de programmation (Jupyter Notebook). Nous créons un nouveau projet nommé (Reconnaissance) et passons à l'implémentation.

4.3 Méthodologie de l'implémentation

Pour la méthodologie de la partie implémentation nous présenterons les différentes étapes de notre code source qui sont :

- Importation des bibliothèques nécessaires.
- Importation de la base donnée.
- Pré-traitement des images de la base donnée.
- Initialisation des hyperparamètres.

- Construction de l'architecture CNN.
- Compilation et ajustement du modèle.
- Évaluation du modèle.

4.3.1 Importation des bibliothèques et des données nécessaires

- Après avoir téléchargé les bibliothèques nécessaires que nous avons déjà définies dans le chapitre précédent.
- À l'aide de la bibliothèque Pandas, nous avons chargé les données d'un fichier Excel dans un DataFrame nommé (data).
- La sortie affiche les premières lignes de notre ensemble de données.

La colonne (Transcription) contient le caractère correspondant à l'écriture Tamahaqt, ce qui est crucial pour entraîner un modèle à reconnaître ces caractères.

La colonne (Image) contient une liste de noms de fichiers séparés par des points-virgules, chacun représentant une image différente du caractère, capturée sous différentes polices. Par exemple, la première ligne est associée à quatre fichiers image : (A.jpg), (A1.png), (A2.png) et (A3.png), qui représentent tous le caractère Tamahaq (A)

	ID	Transcription	Image
0	1	A	A.jpg;A1.png;A2.png;A3.png
1	2	b	B.jpg;B1.png;B2.png;B3.png
2	3	H	C.jpg;C1.png;C2.png;C3.png
3	4	d	D.jpg;D1.png;D2.png;D3.png
4	5	D	Dd.jpg;Dd1.png;Dd2.png;Dd3.png

FIGURE 4.2 – Les 5 premiers lignes de dataframe

4.3.2 Pré-traitement des images de la base donnée

Nous savons que pour appliquer le deep learning, nous avons besoin d'un corpus volumineux. Cependant, la langue tamahaqt présente un manque de données, notamment en ce qui concerne les différentes polices de caractères (l'unicode ne contient pas beaucoup de polices).

Pour pallier cela, nous avons décidé de procéder au prétraitement des images. L'objectif est d'augmenter la taille du corpus et d'améliorer la qualité des images, afin de les préparer pour l'analyse.

Les étapes clés du pré-traitement des images que nous avons abordées dans le chapitre précédent qui sont les suivantes : Chargement des images, Redimensionnement, Binarisation, Réduction de bruit, Détection et correction des lignes de texte, Squellettisation, Normalisation, Ségmentation.

4.3.3 Initialisation des hyperparamètres et division du corpus

Nous avons mis en place un script Python utilisant TensorFlow pour préparer un jeu de données d'images destiné à l'entraînement d'un modèle CNN.

Nous chargeons les images à partir d'un répertoire spécifié (dossier_images) et les redimensionnons uniformément à une taille de 224x224 pixels.

Pour assurer une préparation robuste des données, nous divisons le jeu de données en deux ensembles distincts : l'ensemble d'entraînement (train_ds) et l'ensemble de validation (val_ds). Nous réservons ainsi 20% des images pour la validation afin d'évaluer rigoureusement la performance du modèle pendant l'entraînement, tandis que les 80% restants sont utilisés pour entraîner efficacement le modèle.

La cohérence et la reproductibilité de cette division sont garanties grâce à l'utilisation d'une graine (seed=123). Chaque ensemble de données est traité par lots de 32 images à la fois, optimisant ainsi le traitement parallèle des données pendant l'entraînement.

Cette approche méthodique assure que notre modèle est formé sur une variété de données représentatives et que sa performance est évaluée de manière précise sur des données de validation distinctes.

4.3.4 Construction de l'architecture CNN

Nous avons créé un modèle d'extraction de caractéristiques en utilisant TensorFlow. Ce modèle est défini comme une séquence de couches dans un réseau de neurones convolutif (CNN). Chaque couche apporte une transformation spécifique aux données d'entrée, qui sont des images redimensionnées à une taille de 224x224 pixels avec 3 canaux de couleur (RGB).

- **Couches Conv2D et MaxPooling2D** : Nous avons ajouté trois couches Conv2D avec des configurations différentes pour extraire des caractéristiques des images. Chaque couche utilise un filtre de taille 3x3 pour détecter des motifs locaux dans les images.

Entre chaque couche Conv2D, nous avons inclus une couche MaxPooling2D avec une fenêtre de pooling de taille 2x2. Cela réduit la dimension spatiale des représentations des caractéristiques, tout en conservant les informations les plus importantes.

- **Couche Flatten** : Après les opérations de convolution et de pooling, nous avons ajouté une couche Flatten qui transforme les représentations 2D en un vecteur 1D. Cela prépare les données pour l'étape de classification finale.
- **Couche Dense** : Enfin, nous avons ajouté une couche Dense avec 128 neurones et une fonction d'activation ReLU. Cette couche est responsable de combiner les caractéristiques extraites des images pour produire des prédictions ou des classifications finales.

Par la suite nous avons compilé notre modèle de réseau neuronal en utilisant l'optimiseur Adam pour ajuster efficacement les poids du modèle avec des taux d'apprentissage adaptatifs.

Nous avons choisi la fonction de perte (`sparse_categorical_crossentropy`) pour mesurer la différence entre nos prédictions de modèle et les étiquettes réelles, adaptée aux problèmes de classification où les étiquettes sont des entiers.

En utilisant la métrique (`accuracy`), nous avons évalué la précision du modèle en calculant le pourcentage d'images correctement classifiées. Cette configuration nous permet d'optimiser le modèle pour minimiser la perte pendant l'entraînement et d'améliorer sa capacité à prédire avec précision les classes d'images.

4.3.5 Compilation et ajustement du modèle

Nous avons entraîné notre modèle de réseau neuronal en utilisant TensorFlow/Keras sur une période de 15 epochs, en intégrant un jeu de données d'entraînement (`train_ds`) et un jeu de données de validation (`val_ds`). La fonction (`model.fit()`) a été utilisée pour ce processus, configurée pour afficher les résultats de chaque epoch en détail. Bien que le code ne montre pas explicitement l'utilisation d'une technique de (`early stopping`), l'entraînement s'arrête automatiquement après 15 epochs.

Pour chaque epoch, les résultats montrent la perte et la précision sur le jeu d'entraînement, ainsi que la perte et la précision sur le jeu de validation. Les premiers résultats indiquent une perte initiale élevée qui diminue légèrement avant de se stabiliser autour de 15.9152, tandis que la précision sur le jeu d'entraînement atteint environ 0.0103. La perte de validation reste stable à 15.8160 avec une précision de 0.0124, suggérant que le modèle n'améliore pas significativement sa performance au fil des epochs.

```
Epoch 1/15
152/152 - 293s - loss: 17.9457 - accuracy: 4.1229e-04 - val_loss: 15.9019 - val_accuracy: 0.0116 - 293s/epoch - 2s/ste
Epoch 2/15
152/152 - 267s - loss: 15.9152 - accuracy: 0.0101 - val_loss: 15.8160 - val_accuracy: 0.0124 - 267s/epoch - 2s/step
Epoch 3/15
152/152 - 268s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 268s/epoch - 2s/step
Epoch 4/15
152/152 - 267s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 267s/epoch - 2s/step
Epoch 5/15
152/152 - 269s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 269s/epoch - 2s/step
Epoch 6/15
152/152 - 269s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 269s/epoch - 2s/step
Epoch 7/15
152/152 - 2732s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 2732s/epoch - 18s/step
Epoch 8/15
152/152 - 289s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 289s/epoch - 2s/step
Epoch 9/15
152/152 - 282s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 282s/epoch - 2s/step
Epoch 10/15
152/152 - 302s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 302s/epoch - 2s/step
Epoch 11/15
152/152 - 283s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 283s/epoch - 2s/step
Epoch 12/15
152/152 - 286s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 286s/epoch - 2s/step
Epoch 13/15
152/152 - 276s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 276s/epoch - 2s/step
Epoch 14/15
152/152 - 270s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 270s/epoch - 2s/step
Epoch 15/15
152/152 - 293s - loss: 15.9152 - accuracy: 0.0103 - val_loss: 15.8160 - val_accuracy: 0.0124 - 293s/epoch - 2s/step
```

FIGURE 4.3 – Les résultats d'entraînement du modèle

4.3.6 Résumer des résultats

Nous avons utilisé la fonction (model.summary()) pour obtenir un aperçu détaillé de la structure de notre modèle de réseau neuronal. Le modèle est séquentiel et comporte plusieurs couches.

La première couche est une couche de convolution (Conv2D) avec 32 filtres, produisant une sortie de forme (222, 222, 32) et comportant 896 paramètres. Elle est suivie d'une couche de max-pooling (MaxPooling2D) qui réduit la taille de la sortie à (111, 111, 32).

La deuxième couche de convolution a 64 filtres et produit une sortie de (109, 109, 64) avec 18 496 paramètres, suivie par une couche de max-pooling qui réduit la taille à (54, 54, 64).

La troisième couche de convolution utilise 128 filtres, produisant une sortie de (52, 52, 128) avec 73 856 paramètres, suivie par une autre couche de max-pooling réduisant la taille à (26, 26, 128). Ensuite, nous avons une couche de flatten qui transforme les sorties en un vecteur de 86 528 éléments.

La dernière couche est une couche dense (Dense) avec 128 neurones, totalisant 11 075 712 paramètres. Au total, notre modèle possède 11 168 960 paramètres, tous entraînaibles, ce qui permet de capturer des caractéristiques complexes des images d'entrée.

```

Model: "sequential_1"
-----
Layer (type)                Output Shape                Param #
-----
conv2d_3 (Conv2D)           (None, 222, 222, 32)       896
max_pooling2d_3 (MaxPooling  (None, 111, 111, 32)       0
2D)
conv2d_4 (Conv2D)           (None, 109, 109, 64)       18496
max_pooling2d_4 (MaxPooling  (None, 54, 54, 64)         0
2D)
conv2d_5 (Conv2D)           (None, 52, 52, 128)        73856
max_pooling2d_5 (MaxPooling  (None, 26, 26, 128)        0
2D)
flatten_1 (Flatten)         (None, 86528)              0
dense_1 (Dense)             (None, 128)                11075712
-----
Total params: 11,168,960
Trainable params: 11,168,960
Non-trainable params: 0

```

FIGURE 4.4 – Table qui résume les résultat obtenue

4.3.7 Évaluation du modèle

Nous avons évalué notre modèle de réseau neuronal en utilisant l'ensemble de validation pour mesurer sa performance après l'entraînement. Le code commence par la fonction (`model.evaluate(val_ds)`), qui calcule la perte (`loss`) et la précision (`accuracy`) sur l'ensemble de validation. Cette évaluation nous permet de comprendre comment notre modèle se généralise aux nouvelles données qui n'ont pas été vues lors de l'entraînement.

Le résultat de cette évaluation montre que le modèle a une perte de validation de 15.8160 et une précision de validation de 0.0124, ce qui signifie que notre modèle n'est pas encore performant

Les valeurs obtenues sont ensuite imprimées pour un examen plus approfondi. Avec une perte de validation affichée à 15.81 et une précision de validation à 0.0123.

```
38/38 [=====] - 17s 385ms/step - loss: 15.8160 - accuracy: 0.0124  
Validation Loss: 15.815993309020996  
Validation Accuracy: 0.012376237660646439
```

FIGURE 4.5 – Évaluation du modèle sur l'ensemble de validation

4.3.8 Teste du modèle

Nous avons testé notre modèle en utilisant une image de l'ensemble de validation pour prédire son caractère correspondant. Après avoir défini l'index de l'image à tester, nous avons extrait et affiché cette image. Nous avons ensuite utilisé notre modèle pour faire une prédiction, en ajoutant une dimension à l'image pour correspondre aux attentes du modèle.

La prédiction a donné l'index de la classe avec la probabilité la plus élevée, que nous avons converti en caractère alphabétique. Dans ce test, le modèle a prédit que la lettre correspondante était (D).

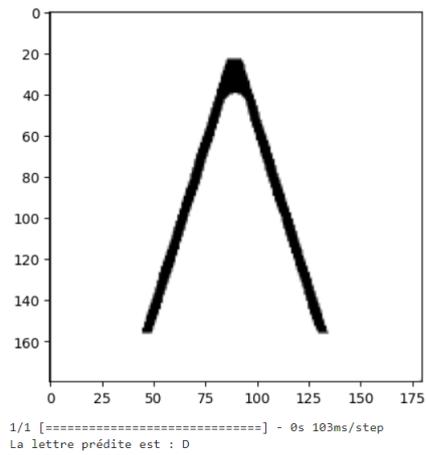


FIGURE 4.6 – La lettre prédite par le modèle

4.3.9 Analyse des résultats

Nous avons visualisé les courbes de perte et de précision de notre modèle pendant l'entraînement à l'aide de Matplotlib. Les valeurs de perte et de précision pour l'entraînement et la validation ont été extraites de l'historique de l'entraînement.

Nous avons tracé ces valeurs, en rouge pour l'entraînement et en vert pour la validation. La courbe de perte montre une diminution de la perte d'entraînement de 17.9457 à 15.9152, tandis que la perte de validation passe de 15.9019 à 15.8160. Pour la précision, l'entraînement reste stable à environ 0.0103 et la validation augmente légèrement de 0.0116 à 0.0124. Ces résultats indiquent que le modèle améliore progressivement sa performance sur l'ensemble de validation, bien que de manière marginale.

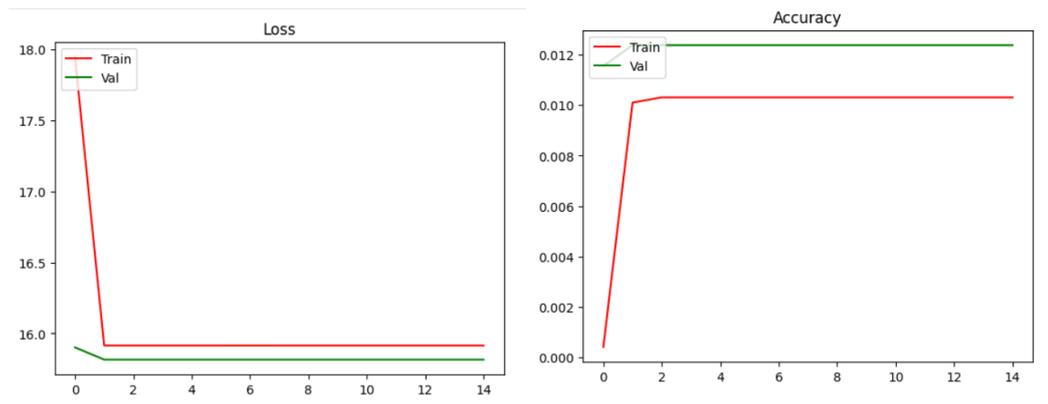


FIGURE 4.7 – Visualisation des résultats

Problème rencontré Les résultats obtenus ne sont pas satisfaisants en raison de plusieurs difficultés rencontrées pendant notre travail. L'une des principales difficultés a été le manque de données ; notre jeu de données était trop restreint. Comme mentionné précédemment, nous

avons contacté l'Université de Tamanghasset pour obtenir des documents afin d'enrichir notre corpus. Cependant, cette limitation a empêché le modèle d'apprendre suffisamment. Malgré l'étape de pré-traitement des données, nous avons également rencontré des problèmes de stockage, nécessitant un ordinateur performant.

Solution proposé Pour effectuer le traitement automatique, il est essentiel de disposer d'un matériel performant afin que le prétraitement et l'entraînement ne prennent pas trop de temps et que l'ordinateur ne plante pas fréquemment. La meilleure solution pour minimiser le temps d'exécution et maximiser la précision des prédictions est d'utiliser un serveur tel que Google Cloud Platform (même si c'est une solution payante), à condition que la base de données soit suffisamment grande pour appliquer les méthodes de deep learning.

Conclusion

En conclusion, notre étude a exploré la reconnaissance automatique des caractères de la langue tamahaqt avec python en utilisant des Réseaux de neurones convolutifs (CNN). Après avoir préparé notre corpus et détaillé les outils et l'environnement de développement, nous avons entrepris l'expérimentation en important les packages et les données, en effectuant le prétraitement, et en appliquant le modèle CNN.

Les résultats montrent des performances limitées principalement en raison du manque de données et des contraintes matérielles. Pour surmonter ces défis, nous recommandons l'utilisation de services de cloud computing tels que Google Cloud Platform, qui peuvent offrir la puissance de calcul nécessaire et améliorer la précision des prédictions, à condition de disposer d'une base de données adéquate pour les méthodes de deep learning.

Conclusion générale

Aujourd'hui, notre société devient plus intelligente, et tout comme d'autres domaines du deep learning, celui de la reconnaissance d'images a connu des avancées significatives depuis l'émergence du deep learning.

Les réseaux de neurones à convolution jouent un rôle dans la reconnaissance des caractères de la langue tamahaqt par leur capacité à extraire automatiquement des caractéristiques pertinentes à partir d'images, ce qui est particulièrement adapté pour le traitement de la langue tamahaqt en raison de son écriture complexe et de sa faible représentation dans les technologies existantes.

Ce projet, centré sur l'analyse des données, démontre comment les avancées en deep learning, avec leur capacité d'apprentissage autonome et leur aptitude à traiter de vastes ensembles de données, peuvent être appliquées pour répondre aux besoins urgents de préservation et de promotion de l'apprentissage d'une langue minoritaire parlée par les Touaregs en Algérie.

Notre étude a détaillé les choix méthodologiques, y compris les techniques de prétraitement des images et l'optimisation des hyperparamètres, pour construire une architecture de réseau performante. Les résultats obtenus indiquent une amélioration significative dans la reconnaissance des caractères tamahaqts, soulignant ainsi le potentiel de ces technologies pour des applications linguistiques.

Nous avons abordé les principes fondamentaux de l'apprentissage automatique des caractères, des réseaux de neurones en général et des réseaux de neurones convolutionnels en particulier. Nous avons introduit ces derniers en décrivant les différentes couches utilisées dans le processus de reconnaissance : couche de convolution, couche de pooling, couches de correction, couche entièrement connectée et couche de perte.

En conclusion, notre étude sur la reconnaissance automatique des caractères de la langue tamahaqt avec Python et les réseaux de neurones convolutifs (CNN) a souligné les défis liés au manque de données et aux limitations matérielles. Pour surmonter ces obstacles, l'utilisation de services de cloud computing tels que Google Cloud Platform pourrait améliorer la précision des prédictions, sous réserve d'une base de données adéquate pour les méthodes de deep learning.

Bibliographie

- [1] A. BOUCHIKHI, A. BOUDRAA, AND J. CHRISTOPHE. Tht et transformation de hough pour la détection de modulations linéaires de fréquence. *Ensieta/E3I2-EA3876*.
- [2] A. HALED, AND M. SLIMANI. Reconnaissance automatique des chiffres avec le deep learning. Mémoire de master, Université A. Mira Bejaïa, Faculté de Technologie, Département de Génie Électrique, 2019/2020.
- [3] A. MANSEUR. L'apprentissage profond(deep learning) pour la classification des lames anapath numérisées. Mémoire de master, Université Larbi Tébessi - Tébessa, Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie Département : Mathématiques et Informatique, Juin 2019.
- [4] A. MISHRA. *Machine Learning for iOS Developers*, première édition ed. Hoboken, New Jersey, États-Unis, 2020.
- [5] A. SAPRE, AND S. VARTAK. Scientific computing and data analysis using numpy and pandas. *International Research Journal of Engineering and Technology (IRJET) 07*, 12 (2020).
- [6] B. ABDEL, AND C. HUBERT. Reconnaissance de caractères : évaluation des performances. Master's thesis, Université Nancy 2 - LORIA, 16 Janvier 2017.
- [7] B. GHENAM, AND S. SMARA. Les réseaux de neurones convolutionnels (cnn) pour la classification des images associées aux places de stationnement d'un parc de véhicules. Mémoire de master, Université Djilali Bounaama, Khemis Miliana, Faculté des Sciences et de la Technologie, Département de Mathématique et d'Informatique, 2018/2019.
- [8] B. HAMEK. Graphie amazighe à base latine : problème des signes diacritiques. *timsal n Tamazighet* (Décembre 2015), 139.
- [9] B. LIU. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, University of Toronoto, 2012.
- [10] B. NADJOUA. Détection des communautés par une méthode d'apprentissage automatique. Mémoire de fin d'études master, Université de 8 Mai 1945 – Guelma, juin 2022.
- [11] C. LARSONNEUR. Intelligence artificielle et/ou diversité linguistique : les paradoxes du traitement automatique des langues. *Hybrid*, N°7 du journal (8 avril 2021).
- [12] D. PRASAD, M. SANHITA, AND J. SMRUTISIKHA. Image classification using machine learning. *Indonesian Journal of Electrical Engineering and Computer Science 31*, 3 (2023).
- [13] E. MAGNANI. Qu'est-ce qu'un corpus ? compte-rendu de la journée d'études. *Les carnets de l'IRHT* (2017).
- [14] F. AITMOUS. Les eunjeux de l'amazighité au maroc. 121 à 131.

- [15] F.ATAALLAH, AND S.BOULAKNADEL. La promotion de l'amazighe à travers les technologies de l'information et de la communication. *Asinag*, 9 (2014), 33–48.
- [16] F.SANANES. L'intelligence artificielle : une solution à la prise de décision quotidienne? Mémoire de fin d'étude, École Supérieure de Génie Informatique 5ESGI, 2018/2019.
- [17] G.DOULAN. De la représentativité à la spécialisation : exemple d'un petit corpus sur la synonymie. *Les petits corpus* (2018), 24.
- [18] G.KARL. *A Grammatical Sketch of the Tamaheq or Towarek Language*, 1861 ed. 4, Royale Crescent, december.
- [19] G.LACHARME, AND H.CARDOT. Optimisation d'hyperparamètres appliquée aux réseaux de neurones convolutifs (cnn).
- [20] G.PEERSMAN. Présentation des méthodes de collecte et d'analyse de données dans l'évaluation d'impact. *Note méthodologique Évaluation d'impact n° 10*, 24.
- [21] G.SAPORTA. Une brève histoire de l'intelligence artificielle. *HAL (Hyper Articles en Ligne)* (11 Décembre 2020).
- [22] H.TOLGUI. Deep learning pour reconnaissance du visage. Mémoire de master, Université Mohamed Khider Biskra, Faculté des Sciences et de la Technologie, Département de Génie Électrique, Filière de Télécommunication, 23 juin 2018.
- [23] I.LAXSAN, AND X.MXMAD. ehoal a n taktot tintarabin. Mémoire master, Université de Tamanghasset, 2021 2022.
- [24] I.LKAN, AND X.MLENAD. imodrn oin t asi li t an ah gar d ta saq tat lt girisn doadmn. Mémoire master, Université de Tamanghasset, 2021 2022.
- [25] K.ABASSIA. Détection de fausses informations dans les réseaux sociaux. Mémoire de master, Université Larbi Tébessi - Tébessa, Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie, Département Mathématiques et Informatique, 14 septembre 2020.
- [26] K.ASKIA. Analyse de données massives dans le contexte académique : indexation et prédictions basées sur l'apprentissage automatique. Mémoire présenté comme exigence partielle de la maîtrise, Université du Québec à Montréal, mai 2020.
- [27] L.ADADA, AND N.HAMIDACHE. Déanonimisation de clients dans le réseau bitcoin à l'aide de l'apprentissage automatique. Mémoire de master, Université Mouloud Mammeri de Tizi-Ouzou, Faculté de : Génie électrique et d'informatique Département : Informatique, 29 septembre 2020.
- [28] L.BARBA, L.BARKER, AND D.BLANK. Teaching and learning with jupyter.
- [29] L.BEAUMIER. Proposition d'une méthode d'inférence de la séquence dans l'image d'une signature manuscrite. Dissertation or thesis, École Polytechnique de Montréal, Faculté des Sciences et de Technologie Département Mathématiques et Informatique, Octobre 1996.
- [30] L.BOUALLEG. Une méthode intelligente pour la détection et la classification des opinions. Mémoire de master, Université Larbi Tébessi - Tébessa, Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie, Département Mathématiques et Informatique, 11 juillet 2021.
- [31] L.GAUTHY. État des lieux et analyse de l'apport potentiel du cloud computing et des techniques de reconnaissance optique de caractères dans le métier d'auditeur. Mémoire de master, HEC - École de gestion de l'Université de Liège, 2022-2023.

- [32] L.KHELIL. Le tamazight, une langue, une culture. de l'oralité à l'écriture. 49–58.
- [33] L.MATTEIS, S.JANNY, S.NATHAN, AND W.SHU-QUARTIER. Introduction à l'apprentissage automatique. *La Revue 3EI*, 108 (avril 2022).
- [34] L.NACHEF. *Normativisation et enseignement de la langue amazighe au Maroc : état des lieux, méthodes et réalisations*. PhD thesis, L'Université Lyon 3, septembre 2016.
- [35] L.SAADAOUY. Technique de traitement numérique d'image en vue de la reconnaissance des formes. Master's thesis, L'école Nationale Polytechnique d'Alger, 1991.
- [36] M.BABAKIDI. Développement d'un filtre gaussien pour la réduction du bruit dans les images médicales. *International Journal of Innovation and Applied Studies* (mai 2022).
- [37] M.BETAOUAF, AND AHMED, D. Détection et reconnaissance de visage dans une image par le deep learning. Mémoire de master, Université – Ain Temouchent - Belhadj Bouchaib, Faculté des Sciences et de Technologie, Département Mathématiques et Informatique, 2020/2021.
- [38] M.BETAOUAF, AND D.BENAHMED. Détection et reconnaissance de visage dans une image par le deep learning. Mémoire de master, Université – Ain Temouchent - Belhadj Bouchaib, Faculté des Sciences et de Technologie, Département Mathématiques et Informatique, 2020/2021.
- [39] M.DJERRAD, AND S.ZIDOUNE. Analyse des sentiments des tweets liés au hirak. Mémoire de master, Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arreridj, Faculté des Mathématiques et d'Informatique, Département d'Informatique, 2020 - 2021.
- [40] M.ZOUINAR. Evolutions de l'intelligence artificielle : quels est enjeux par activité humaine et la relation humain-machine au travail? *Activités 17*, N°1 du journal (20 janvier 2020). Article soumis le 8 janvier 2019, accepté le 20 janvier 2020.
- [41] P.ÉDAGOGUES. microsoft office access 2007 utilisation de base.
- [42] R.IDOUI, AND A.HIMEUR. Reconnaissance d'images par les réseaux de neurones convolutionnels (cnn). Mémoire de master, Université Mohammed Seddik BenYahia Jijel, Faculté des Sciences et de la Technologie, Département d'Électronique, 2019/2020.
- [43] R.MIFDAL. Application des techniques d'apprentissage automatique pour la prédiction de la tendance des titres financiers. Mémoire (exigence partielle), École de Technologie Supérieure, Université du Québec, 12 novembre 2019.
- [44] R.MIFDAL. Application des techniques d'apprentissage automatique pour la prédiction de la tendance des titres financiers. Mémoire présenté comme exigence partielle à l'obtention de la maîtrise, École de Technologie Supérieure, Université du Québec, 12 novembre 2019.
- [45] R.TABTI. L'évolution de status de tamazight. *Comparative Legilinguistic* 50, 17 (2022), 15.
- [46] R.YOUSSEF. Squelettisation d'images en niveaux de gris et applications. Master's thesis, Université Sorbonne Paris Cité; Université de Carthage (Tunisie), 2015. Français, Submitted on 28 Aug 2017.
- [47] S.CHAKER. Encyclopédie berbère iv. Master's thesis, 1987.
- [48] S.LABIDI, AND W.LEJOUAD. *De l'intelligence artificielle distribuee aux systemes multi-agents*. Rapport de recherche, INRIA Sophia-Antipokis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex, France, Aout 1993.

-
- [49] S.SAIDJ. Techniques de nlp pour la détection des fausses nouvelles. Mémoire de master, Université Ibn Khaldoun - Tiaret, Faculté des Mathématiques et de l'Informatique, Département d'Informatique, 27 juin 2022.
- [50] T.BENSAHLA, AND M.BEREKSI. Réalisation d'un système autonome de contrôle d'accès de véhicules par reconnaissance optique des plaques d'immatriculation. Mémoire de master, Université Aboubakr Belkaïd Tlemcen, Faculté de Technologie, juin 2022.
- [51] T.ZINAB, AND T.SAGHIDA. Initn dghtmnghtu. Mémoire master, Université de Tamanghasset, 2021 2022.

Annexes

4.4 Annexes 1

[51]

+EIEI.:	+O.O.θ+	+H.O.EIE+	∩/∧	∩/∧	∩/∧
•	ا	A	•∩.ا	امان	الماء Aman
θ	ب	B	∫θ.∫E	ابقي	الذئب Ebagy
+	ت	T	+∫E	تظ	العين Tet
θ	س	S	Eθ.ا	اسان	اللحم Isan
X	ج	J	∫X∫C.O	اجمر	الخروف Ejemar
::	ح	KH	•::	اخ	الحليب Akh
∧	د	D	•∩.∧C	اوادم	الانسان Awadm
*	ذ	Z	•∩.*.∩	امراغ	العائلة Amazzagh
O	ر	R	•∩.OC	اغرم	المدينة Aghrm
#	ظ	Z	•#OI	اظرف	الفضة Azraf
E	ط	T	+∩.CE	تمط	المرأة Tamaṭ
∩	ك	K	•∩.∩	اكال	البرا Akal
∩	ل	L	∫∩C	الم	الجلد Elam
C	م	M	E∩.ا	امان	الروح Iman
∩	ن	N	•∩.	انو	البر Anou
E	ض	D	•EO	اضر	الرجل Aḍar
∩	ع	GH	•∩.∩	اورغ	الذهب Oragh
I	ف	F	+∩.∩+	توفات	الصباح Toufat
-	ق	Q	+∩.-	تلاق	الطين Tlaq
E	ش	CH	•E.∩	اشك	الشجرة Achk
∩	ح	H	∫∩.ا	اهن	الخيمة Ehan
:	و	«W «o,Ou	•∩.+∩E	اواتاي	السنة Awatay
E	ي	« I « y	Eθ.ا	اسان	اللحم Isan
∫	()	« E « é	∫E∧E	ايدي	الكلب Eydi edi
X	ق	G	∫X.E∫E	اقضيض	العصفور Egaded
#	ص	S	•#∩+	اصاهت	القوة Asahat
T	ج	G	•T.∩C.∩	اجلمام	البركة agalmam
†	(غنتر)	aña	•∩.	انا	الاغ Aña

Résumés

Résumé

Dans cette étude, nous avons exploré une méthode d'apprentissage automatique pour identifier les caractères alphabétiques. Cette méthode s'appuie sur l'utilisation de réseaux de neurones profonds, qui sont actuellement parmi les plus performants en matière de classification d'images. Le modèle que nous avons mis au point a été formé avec un ensemble de données composé d'images de lettres, enrichi grâce à la coopération de l'Université de Tamanghasset. Les résultats obtenus ne sont pas entièrement satisfaisants, principalement à cause de plusieurs obstacles comme le manque de données et les limitations de stockage. Malgré ces difficultés, le modèle a montré une amélioration progressive de ses performances sur l'ensemble de validation. En résumé, cette étude met en évidence les défis et les opportunités dans le domaine de la reconnaissance des caractères alphabétiques. Bien que les résultats soient modestes, il existe des perspectives prometteuses d'amélioration avec des ressources et des données appropriées.

Mots-clés

Reconnaissance de caractères, Apprentissage profond, Apprentissage automatique, Collecte de données, Réseaux de neurones convolutifs (CNN), Traitement du langage naturel (NLP).

Abstract

In this study, we explored a machine learning method for identifying alphabetical characters. This method relies on the use of deep neural networks, which are currently among the most effective in image classification. The model we developed was trained on a dataset of letter images, enriched through cooperation with the University of Tamanghasset. The results obtained were not entirely satisfactory, primarily due to several obstacles such as data scarcity and storage limitations. Despite these difficulties, the model showed a progressive improvement in its performance on the validation set. In summary, this study highlights the challenges and opportunities in the field of character recognition. Although the results are modest, there are promising prospects for improvement with adequate resources and data.

Keywords

Character Recognition, Deep Learning, Machine Learning, Data Collection, Convolutional Neural Networks (CNN), Natural Language Processing (NLP).

