

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Abderrahmane Mira

Faculté de la Technologie



Département d'Automatique, Télécommunication et d'Electronique

Projet de Fin d'Etudes

Pour l'obtention du diplôme de Master

Filière : Automatique

Spécialité : Automatique et Informatique industrielle

Thème

Reconnaissance automatique de l'écriture amazighe.

Préparé par :

- IOUKNANE Idris
- KHETACHE Billal

Dirigé par :

Mme. GAGAOUA

Examiné par :

Mr. MENDIL

Mr. TIGHZERT

Année universitaire : 2021/2022

Remerciement

On remercie dieux le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de Mme GAGAOUA, on le remercie pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.

Nos vifs et chaleureux remerciements vont également au membre de jury Mr. MENDIL et Mr. TIGHZERT pour le grand honneur qu'ils font en acceptant de juger ce travail.

Dédicace

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve.

A mes chers parents

Pour tous leurs sacrifices, leurs amour, leurs tendresse, leurs soutien et leurs prières tout au long de mes études.

A chères sœurs

Sonia et Louna pour leurs encouragements permanents, et leurs soutiens moraux.

A chères frères

Bachir et Faouzi pour leurs appuis et leurs encouragements.

A mes grands parents

Ceci est ma profonde gratitude pour votre éternel amour, et à la mémoire de mon grand-père et qu'il apprécie cet humble geste comme preuve de reconnaissance de la part d'un petit fils qui a toujours priés pour le salut de son âme.

A mes chers oncles

A mes chers cousins et cousines

A mon binôme et cher ami

Billal pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

A mes merveilleux amis

Lounes, Mabrouk, Lounis, Ryad, Haroun, Nassim, Rafik et Célia pour leurs encouragements, leurs aides et supports dans les moments difficiles.

Arif

Dédicace

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve.

A mes chers parents

Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

A mon frère et ma chère sœur

Mazigh et Inès pour leurs encouragements, leurs conseils précieux tout au long de mes études.

A mes grands parents

Ceci est ma profonde gratitude pour votre éternel amour, et à la mémoire de mes deux grands-mères et qu'elles apprécient cet humble geste comme preuve de reconnaissance de la part d'un petit fils qui a toujours priés pour le salut de leurs âmes.

A mes chers oncles et tantes

A mes chers cousins et cousines

A mon binôme et cher ami

Idris pour sa patience et les efforts dont il a fait preuve et sa compréhension tout au long de ce projet.

A mes merveilleux amis

Massi, AzzdineTakhrovth, Khodir, Mehdi, Massil, Nassim, Célia, Ferial, Kamelia et Houa pour leurs encouragements, leurs aides et supports dans les moments difficiles.

Billal

Liste des matières

Liste Des Figures	
Liste Des tableaux	
Liste Des Acronymes	
Introduction générale.....	1

CHAPITRE I : reconnaissance de l'écriture

I.1 Introduction	2
I .2 Différents aspects de l'OCR.....	2
I .2.1 Le mode d'acquisition de l'écriture	2
I .2.2 Approches de reconnaissance	4
I .3 Caractéristique de la langue amazighe	4
I .3.1 Système d'écriture de la langue amazighe	5
I .4 Organisation générale d'un système de reconnaissance	7
I.5 Processus de reconnaissance	7
I.5.1 Phase d'acquisition.....	7
I.5.2 Phase de prétraitement.....	8
I.5.3 Phase de segmentation	9
I.5.4 Phase d'extraction de caractéristique.....	9
I .5.5 Phase de classification	10
I .5.6 Phase de post-traitement	11
Conclusion.....	11

CHAPITRE II : L'apprentissage profond

II.1 Introduction	12
II.2 Définition	12

Liste des matières

II.3 Historique	13
II.4 Apprentissage automatique	13
II.4.1 Types d'apprentissage	14
II.4.2 L'apprentissage automatique selon le type de résultat	14
II.5 Domaine d'application Deep Learning.....	14
II.6 Réseaux de Neurones	15
II.7 Les fonctions d'activation	16
II.8 Réseaux de neurones convolutionnels.....	18
II.8.1 Architecture de réseaux de neurones convolutionnels.....	18
II.8.2 Les couches CNN	19
II.8.2.1 La couche de convolution	19
II.8.2.2 Couche de pooling (POOL)	21
II.8.2.3 La couche de correction ReLU	22
II.8.2.4 La couche fully-connected.....	23
II.8.3 Apprentissage des réseaux de neurones convolutifs	23
II.8.4 Les Avantages de CNN	24
Conclusion	24

CHAPITRE III : Reconnaissance de l'écriture Amazigh avec DL et l'attribut HMB

III.1 Introduction	25
III.2 HMB (Histogram of marked background)	25
III.2.1 Marquage du fond	25
III.2.2 Exemples de marquage	27
III.3 Architecture générale de notre proposition	28.

Liste des matières

III.4 Logiciel et librairies utilisé dans l'implémentation	28
III.4.1 Python	28
III.4.2 Tensorflow	29
III.4.3 keras	29
III.5 conception.....	29
III.5.1 Train	29
III.5.2 Test	30
III .6 configuration utilisés dans l'implémentation	30
III .7 Base de données	30
III.8 Architecture du réseau CNN	31
III.8.1 Réseau neuronal convolutif à 2 couches	31
III .8 .2 réseau neuronal convolutif à 4 couches	32
III.8.3 réseau neuronal à 6 couches	32
III.9 Code source de notre architecture	33
III.10 Les différents cas d'implémentation	34
III.10.1 Sans binarisation et marquage	34
III.10.2 Avec binarisation et marquage	38
III.11 Résultats et discussions pour la combinaison de la base AMHCD (sans binarisation et marquage et avec binarisation et marquage)	43
III.11.1 Résultats obtenus pour le modèle à 2 couches	43
III.11.2 Résultats obtenus pour le modèle à 4 couches.....	45
III.11.3 Résultats obtenus pour le modèle à 6 couches.....	46
III.12 Conclusion.....	48
Conclusion générale	49

Liste des matières

Bibliographie	50
Résumé	53

Liste des figures

CHAPITRE I

Figure I.1 : Texte en tifinagh manuscrit et en imprimé.	4
Figure I.2 : Différents systèmes, représentations et approches de reconnaissance.....	5
Figure I.3 : Caractères Tifinagh.....	5
Figure I.4 : Organisation générale d'un système de reconnaissance d'écriture.....	7
Figure I.5 : normalisation spécifique au caractère « ya »	8
Figure I.6 : Binarisation d'un texte Amazigh en Tifinagh.....	8
Figure I.7 : Squelettisation d'un mot Amazigh en Tifinagh.....	8
Figure I.8 : la segmentation de quelques lettres Amazigh en tifinagh	9
Figure I.9 : l'architecture d'organisation d'un système de reconnaissance.....	10

CHAPITRE II

Figure II.1 : L'intelligence artificielle et ses sous-domaines.	12
Figure II.2 : structure de réseaux de neurone.....	15
Figure II.3 : Structure simplifiée d'un neurone artificiel.....	16
Figure II.4 : Architecture d'un réseau CNN.....	18
Figure II.5 : Exemple simpliste des valeurs des pixels d'une image 5x5 et de valeurs d'une matrice utilisée comme filtre.....	19
Figure II.6 : Appliquer le filtre sur l'image.....	20
Figure II.7 : Parcours de la fenêtre de filtre sur l'image.....	20
Figure II.8 : Max pooling.	21
Figure II.9 : Average pooling.....	22
Figure II.10 : allure de la fonction ReLU.....	22
Figure II.11 : Représentation de la couche fully-connected.....	23

CHAPITRE III

Liste des figures

Figure III.1 : Configuration avec 4 directions.....	26
Figure III.2 :Exemple de marquage d'un pixel de fond ayant un seul voisin d'encre	27
Figure III.3 : exemples d'images de caractères après le processus de binarisation et marquage de la base de caractère amazigh AMHCD.	27
Figure III.4 : Schéma générale de notre système.....	28
Figure III.5 : structure de la base de données AMHCD.....	31
Figure III.6 : Architecture du modèle à 2couches.....	32
Figure III.7 : Architecture du modèle à 4couches.....	32
Figure III.8 : Architecture du modèle à 6 couches.....	33
Figure III.9 : Précision et erreur du modèle à 2 couches (10 époques).....	43
Figure III.10 : Précision et erreur du modèle à 2 couches (20 époques).....	44
Figure III.11 : Précision et erreur du modèle à 2 couches (30 époques).....	44
Figure III.12 : Précision et erreur du modèle à 4 couches (10,20 et 30 époques).....	45
Figure III.13 Précision et erreur du modèle à 6 couches (10 époques).	46
Figure III.14 Précision et erreur du modèle à 6 couches (20 époques).	47
Figure III.15 : Précision et erreur du modèle à 6 couches (30 époques).....	47

Liste Des Tableaux

CHAPITRE I

Tableau I.1 : L'alphabet Tifinagh préconisé par le CAL consacré par L'IRCAM.....	6
---	---

CHAPITRE II

Tableau II.1 Les fonctions d'activation.....	17
---	----

CHAPITRE III

Tableau III.1 : Toutes les possibilités de marques avec quatre directions et leurs codes binaires ainsi que leurs signes décimaux correspondants [31].	26
---	----

Tableau III.2 : configuration utilisés dans l'implémentation.	30
---	----

Tableau III.3: Comparaison entre les différentes architectures et époques cas train 2/3, test 1/3.	35
--	----

Tableau III.4: Comparaison entre les différentes architectures et époques cas train 1/3, test 2/3.	35
--	----

Tableau III.5: Comparaison entre les différentes architectures et époques cas test 1/2, train 1/2.	37
--	----

Tableau III.6: Comparaison entre les différentes architectures et époques cas train 2/3, test 1/3.	38
--	----

Tableau III.7 : Comparaison entre les différentes architectures et époques cas train 1/3, test 2/3.	39
---	----

Tableau III.8 : Comparaison entre les différentes architectures et époques cas test 1/2, train1/2.	40
--	----

Tableau III.9 : Comparaison entre les différentes architectures et époques pour le cas binarisation et marquage sans binarisation et marquage précisions dans le cas train 2/3 – test 1/3.	42
--	----

Tableau III.10: Comparaison de différentes précisions dans le cas train 2/3 – test 1/3.....	43
--	----

Liste Des Acronymes

REA : reconnaissance automatique de l'écriture

OCR : reconnaissance optique de l'écriture

API : alphabet phonétique international

CAL : Centre de l'Aménagement Linguistique

I'IRCAM : Institut royal de la culture amazigh au maroc

ML : machine learning

DL : deep learning

CNN : Convolutional Neural Network

MLP : multiLayer perceptron

CONV : couche de convolution

POOL : pooling Layer

ReLU : Unité de rectification linéaire

AVG : average

FC : fully-connected

AMHCD : database that contains handwritten Amazigh characters

CPU : central processing unit

GPU : graphics processing unit

HMB : Histogram of marked background

Introduction générale

La reconnaissance automatique de l'écriture manuscrite reste encore un sujet de recherche et d'expérimentation. Plusieurs recherches scientifiques ont été effectuées sur le caractère latin, arabe, et autre. Ceci a permis le développement de plusieurs approches de reconnaissance automatique de ces caractères.

Par contre, le caractère amazighe, appelé tfinaghe, est très peu traité. Il tire son origine du vieil alphabet libyque et saharien, déjà utilisé depuis le VIème siècle avant l'ère chrétienne par les populations de l'Afrique du Nord, du Sahel et des Iles Canaries. Cet alphabet a subi des modifications et des variations depuis son origine jusqu'à nos jours.

Dans notre travail on a utilisées la combinaison de deep learning et l'attribut HMB pour la reconnaissance des lettres tfinagh, et ces derniers qui représentent en réalité le dernier maillon dans l'évolution de l'intelligence artificielle.

Ce travail comporte trois chapitre, le premier présente les concepts généraux liés à la reconnaissance de l'écriture manuscrite, ainsi que les étapes et les méthodes utilisées à cet effet. Dans le deuxième chapitre nous présentons d'une part un aperçu général sur le deep learning et le réseau de neurones convolutionnels en particulier. Enfin, le troisième chapitre est consacré sur la partie expérimentale de notre travail, et nous allons présenter et discuter les résultats obtenus.

CHAPITRE I

La Reconnaissance de L'écriture

I.1 Introduction :

L'écriture est un moyen de communication entre les individus. Les humains rêvent aussi de communiquer avec les machines de la même manière, ce qui semble plus simple et plus facile pour les opérations qui accélèrent l'acquisition, la communication et le traitement d'informations, d'où l'avènement de la reconnaissance automatique de l'écriture (RAE).

La reconnaissance automatique de l'écriture manuscrite gagne du terrain en tant que véritable technologie de codage du support écrit et semble rattraper d'autres disciplines connexes, comme la reconnaissance automatique de la parole. En ce qui concerne le script lui-même, cette technologie semble avoir largement contribué à réduire l'écart qui existe entre la performance obtenue à l'impression et celle obtenue sur le manuscrit.

La reconnaissance de l'écriture manuscrite remonte à plus de 30 ans. Aujourd'hui, il existe plusieurs domaines où la reconnaissance de l'écriture est très attendue, comme le tri automatique du courrier, le traitement automatique des documents administratifs, l'enregistrement des questionnaires ou des chèques bancaires etc.

I.2 Différents aspects de l'OCR:

Il n'existe pas de système universel d'OCR (reconnaissance optique de l'écriture) capable de reconnaître n'importe quel caractère dans n'importe quelle police. Tout dépend du type de données traitées et bien sûr de l'application prévue.

Il existe plusieurs modes de classification des systèmes OCR on cite quelques unes :

- Les systèmes qualifiés « en-ligne » ou « hors-ligne » suivant le mode d'acquisition.
- Les approches globales ou analytiques selon l'analyse s'opèrent sur la totalité du mot, ou par segmentation des caractères [1].
- Les approches statistiques, structurelles ou stochastiques relatives aux traits caractéristiques extraits des formes considérées.

I.2.1 Le mode d'acquisition de l'écriture :

Les systèmes de reconnaissance automatique de l'écriture manuscrite sont divisés en deux familles :

➤ Reconnaissance en ligne :

Dans ce type de reconnaissance, l'utilisateur écrit sur un formulaire spécial et le système va reconnaître l'écriture et envoie le résultat à l'ordinateur. Ces systèmes

sont utilisés dans de nombreux appareils électroniques comme PDA, Pocket PC ou Tablet PC.

La reconnaissance de l'écriture en ligne présente des avantages par rapport à la reconnaissance de l'écriture hors ligne. Par exemple, l'utilisateur écrit sur un formulaire spécial, il y a moins de bruit. En outre, il est possible de déterminer comment un caractère est écrit, c'est-à-dire l'ordre des traits qui le composent, tel que la contrainte de temps n'est pas stricte et des algorithmes complexes peuvent être utilisés. Pour cette raison, le taux de reconnaissance de ces systèmes est assez élevé [2].

➤ **Reconnaissance hors ligne :**

Dans le cas de la reconnaissance hors ligne, l'écriture de l'utilisateur est acquise par le scanner. L'entrée de ce type de système de reconnaissance est une image scannée d'un document précédemment écrit, le problème est donc difficile. Par exemple, comment supprimer le bruit qui fait partie de l'image de fond, comment traiter les lignes manquantes, etc. En outre, il n'y a pas d'informations supplémentaires comme en ligne et pour être largement utilisés, ces systèmes doivent avoir un temps de traitement rapide et un taux de reconnaissance élevé [2].

La reconnaissance hors-ligne classée en plusieurs types, parmi ses types on cite les suivants :

✚ **Reconnaissance de texte ou analyse de documents :**

Dans le premier cas, il s'agit d'identifier un texte dont la structure est limitée à quelques lignes ou mots. La recherche consiste simplement à repérer les mots de la ligne, puis à diviser chaque mot en caractères.

Dans le second cas, les données sont bien structurées et la connaissance de la mise en page et de la présentation du document est nécessaire [2].

✚ **Reconnaissance de l'imprimé ou du manuscrit :**

La reconnaissance optique de l'écriture peut être séparée en reconnaissance de l'écriture manuscrite et reconnaissance des caractères.

Les méthodes sont différentes pour la reconnaissance des caractères imprimés et manuscrits.

La reconnaissance des caractères manuscrits est plus difficile à appliquer que la reconnaissance des caractères imprimés. Dans le cas des imprimés, l'image des caractères à traiter va être formatée par des polices des caractères standards, bien alignées et généralement bien séparées verticalement avec simplification de phase de lecture. En outre, la conception des caractères est conforme au style fonte et fournit un modèle de reconnaissance.

Dans les manuscrits, il existe différents styles d'écriture humaine qui sont souvent liés entre eux et dont les graphismes sont de proportions inégales. Cela nécessite l'utilisation de techniques de délimitation très spécifiques et souvent le recours à des connaissances de base pour guider la lecture [2].

La figure **Figure I. 1** illustre un exemple d'un Texte en tifinagh manuscrit et en imprimé.

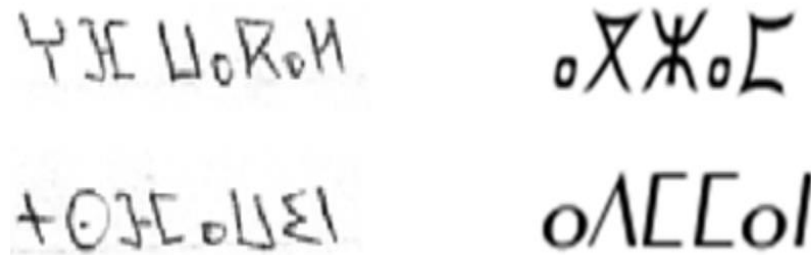


Figure I. 1 Texte en tifinagh manuscrit et en imprimé.

I.2.2 Approches de reconnaissance :

On a deux approches s'opposent en reconnaissance des mots : globale et analytique.

➤ L'approche globale :

Considérer le mot comme une entité unique et le décrire indépendamment des caractères qui le composent. Cette approche a l'avantage de maintenir le personnage dans son environnement, ce qui permet une modélisation plus efficace des changements dans le texte et de la dégradation qu'il peut subir. Cependant, cette approche souffre de la taille de la mémoire, du temps de calcul et de la complexité du traitement, qui augmentent linéairement avec la taille du vocabulaire considéré, ce qui rend ce dernier très limitatif [4].

➤ L'approche analytique :

Contrairement à l'approche globale, les mots sont divisés en caractères ou en segments morphologiquement avec une signification plus petits que les caractères, appelés graphèmes. La reconnaissance lexicale consiste à identifier des entités segmentées, puis à les cibler par une autre reconnaissance d'une tâche délicate qui produit différents types d'erreurs. Selon

cette approche, le processus de reconnaissance est basé sur l'alternance de deux phases : la phase de segmentation et la phase d'identification des segments [7].

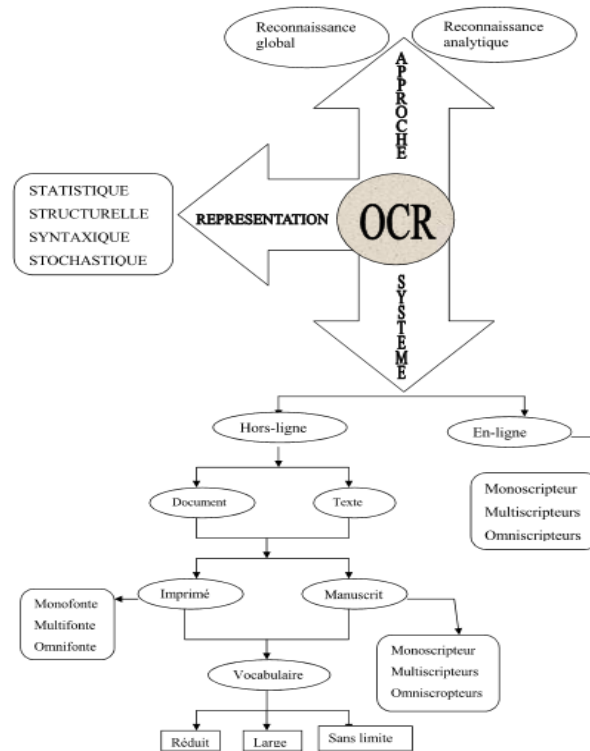


Figure I.2 : Différents systèmes, représentations et approches de reconnaissance.

I.3 Caractéristique de la langue amazighe :

Aujourd'hui, l'amazigh est parlé par environ 14 millions de personnes, principalement au Maghreb. L'écriture amazighe s'écrit généralement verticalement de gauche à droite, de haut en bas, et c'est une écriture non cursive. Il simplifie la division des caractères dans les images de l'écriture amazighe. Figure I.3 Montre les caractères Tifinagh utilisés par l'IRCAM.

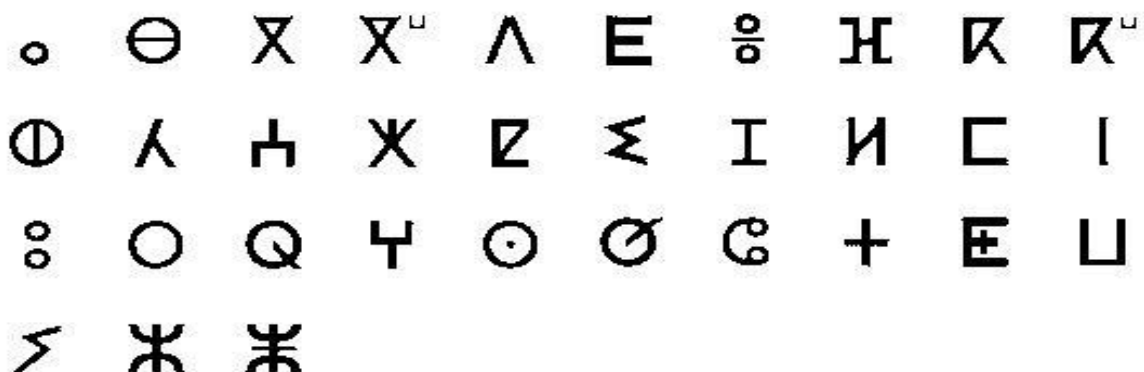


Figure I.3: Caractères Tifinagh

I.3.1 Système d'écriture de la langue amazighe :

Avant la standardisation de l'amazigh, il était écrit et sa modélisation informatique à l'aide des lettres arabes ou latines, enrichies de caractères spéciaux empruntés à l'alphabet phonétique international(API). L'objectif est de fournir un système alphabétique standard pour l'amazigh qui soit plus approprié et applicable à toutes les langues amazighes. Le Centre de l'Aménagement Linguistique (CAL) a préconisé les caractères tifinaghes, présentés dans le Tableau 1 la correspondance des lettres tifinagh avec lettres arabes et latines :

ΣΘΡΚΞΗΙ +ΗΞΙΟΨ			
	Tifinaghe	Correspondance latine	Correspondance arabe
ya	◌	A	ا
yab	⊖	B	ب
yag	⊗	G	گ
yag ^w	⊗ ^w	g ^w	گ ^w
yad	∧	D	د
yaḍ	E	d	ظ
yey	⊖	E	
yaf	⊗	F	ف
yak	⊗	K	ك
yak ^w	⊗ ^w	k ^w	ك ^w
yah	⊖	H	ه
yaḥ	∧	h	ح
yæ	∧	E	ع
yax	⊗	X	خ
yaq	⊗	Q	ق
yi	ξ	I	ي
yaj	⊖	J	ج
yal	⊗	L	ل
yam	⊖	M	م
yan	⊖	N	ن
yu	⊖	U	و
yar	⊖	R	ر
yaṛ	⊗	r	ر ^w
yaγ	⊗	Γ	غ
yas	⊖	S	س
yaş	⊗	s	ص
yac	⊖	C	ص
yat	+	T	ت
yaṭ	E	t	ط
yaw	⊖	W	و
yay	ξ	Y	ي
yaz	⊗	Z	ز
yaž	⊗	z	ژ

Tableau I.1 : L'alphabet Tifinagh préconisé par le CAL consacré par L'IRCAM.[35]

I.4 Organisation générale d'un système de reconnaissance

Un système de reconnaissance fait appel généralement aux étapes suivantes: Acquisition, prétraitements, segmentation, extraction des caractéristiques et classification, suivis éventuellement d'une phase de post-traitement.

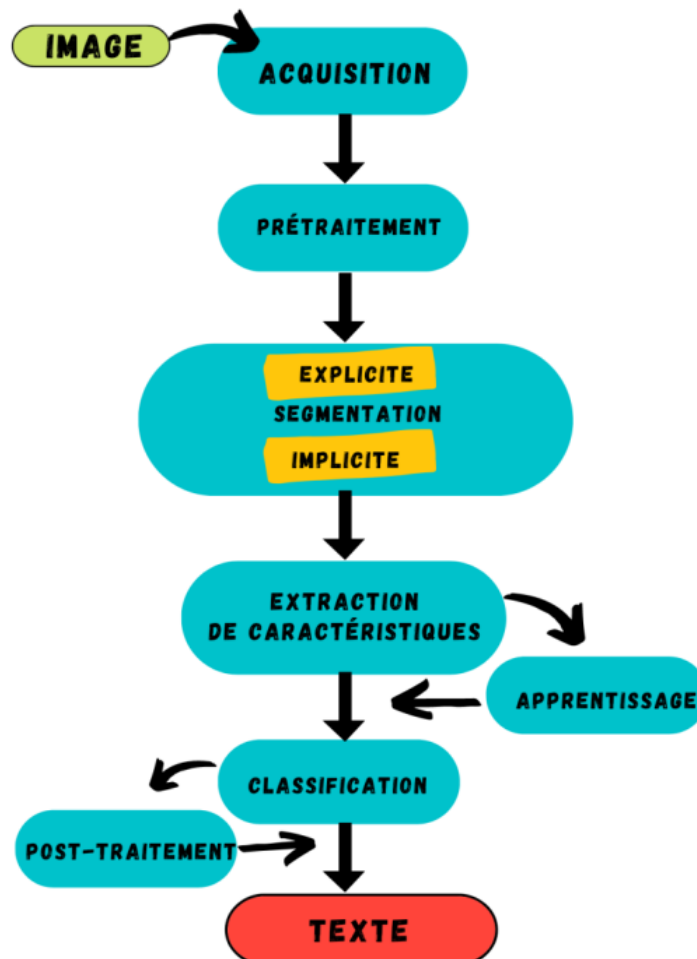


Figure I.4 : Organisation générale d'un système de reconnaissance d'écriture

I.5 Processus de reconnaissance :

I.5.1 Phase d'acquisition:

La phase d'acquisition consiste à capter l'image d'un texte au moyen des capteurs physiques (scanner, caméra,...) et de la convertir en grandeurs numériques adaptés au système de traitement, avec un minimum de dégradation possible.

I.5.2 Phase de prétraitement:

Pour automatiser la reconnaissance de l'écriture, il faut préparer le travail de la machine. De nombreux efforts sont consacrés aux étapes préliminaires nécessaires à la reconnaissance, notamment la normalisation, la binarisation, le redressement et la squelettisation [8-9].

Voici quelques exemples d'application de ces opérations :

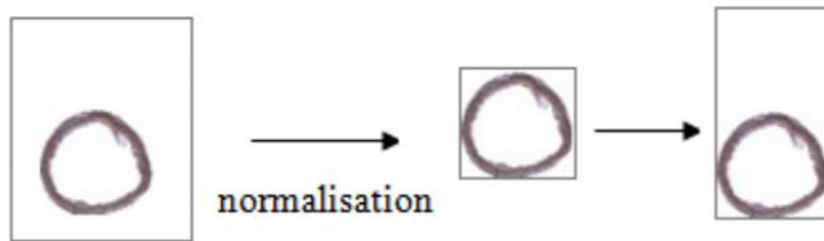


Figure I.5 : normalisation spécifique au caractère « ya »

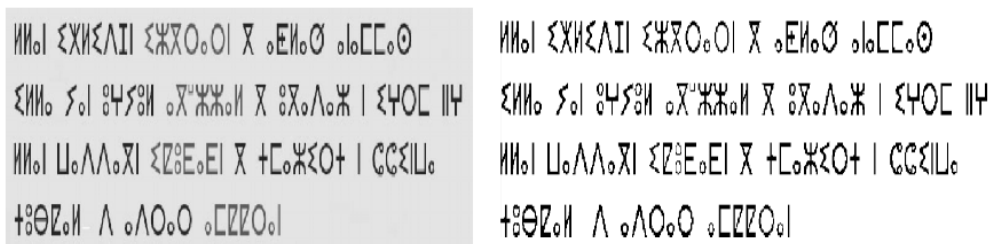


Figure I.6 : Binarisation d'un texte Amazigh en Tifinagh

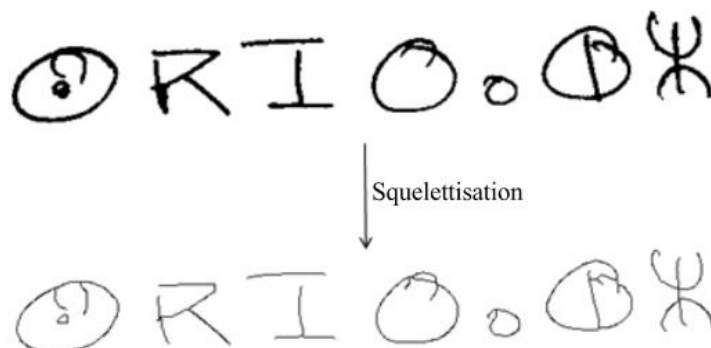


Figure I.7 : Squelettisation d'un mot Amazigh en Tifinagh

I.5.3 Phase de segmentation :

La segmentation est une opération très importante dont le but d'isoler les différents composants d'un bloc de texte en lignes, mots et caractères avant l'étape de reconnaissance des caractères. Le résultat de cette opération est une forme séparée de l'image, qui peut être un caractère ou autre. Les méthodes de segmentation dépendent notamment de la langue, de la fonte, de la justification, des variations de l'écriture manuscrite et de leur interaction avec la phase de reconnaissance. Cela peut conduire, selon les cas, à des opérations de séparation d'une lettre en graphèmes ou à des regroupements de quelques caractères en un seul mot.

Tout système de reconnaissance d'écriture doit impérativement localiser l'information avant de la reconnaître. On distingue quatre types de segmentation :

Texte en lignes, segmentation des lignes en mots, segmentation des mots en lettres et la Segmentation de la page [2] comme illustré dans la Figure I.8 :

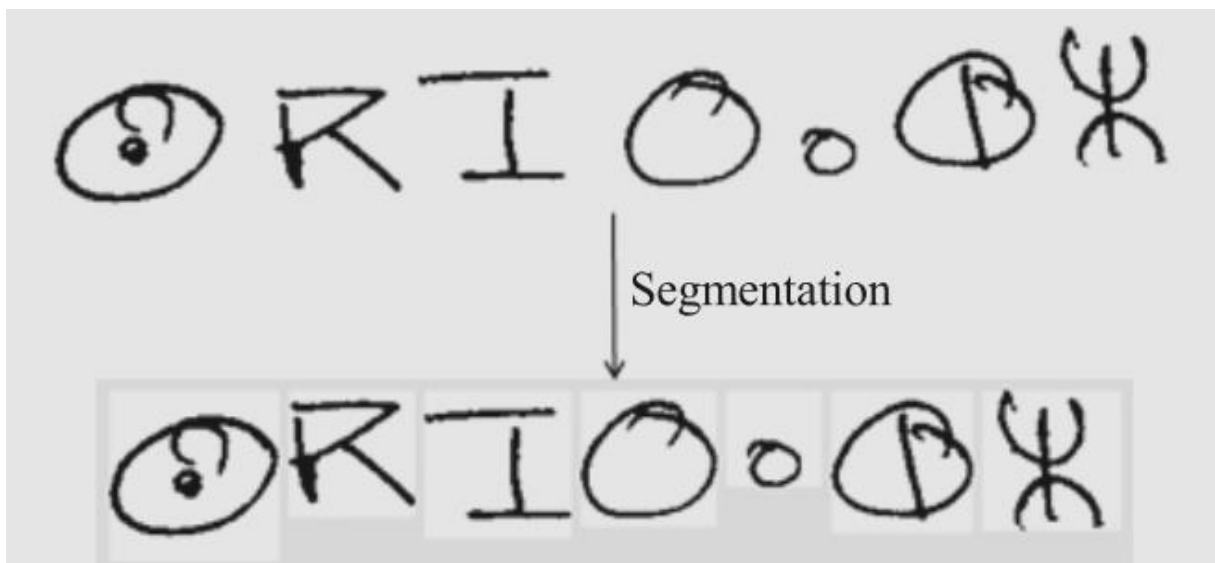


Figure I.8 : la segmentation de quelques lettres Amazigh en tifinagh

I.5.4 Phase d'extraction de caractéristiques :

C'est une étape clé dans la construction d'un système de reconnaissance, L'une des raisons pour lesquelles cette étape est problématique est que de nombreuses techniques d'extraction s'accompagnent d'une perte d'informations. Par conséquent, il y a un compromis entre et la qualité de l'information [28].

Elle permet d'améliorer la visualisation et la compréhension des données, de réduire les temps d'apprentissage et de classification des systèmes, d'améliorer les performances en classification, et permet de réduire la taille des bases d'apprentissage. En Deep Learning la phase d'extraction et de classification sont réunies dans une seule étape. Comme a été représenté dans la Figure I.9 qui contient deux phases : (a) qui représente l'organisation d'un système de reconnaissance avec réseau de neurone, (b) c'est l'organisation classique d'un système de reconnaissance.

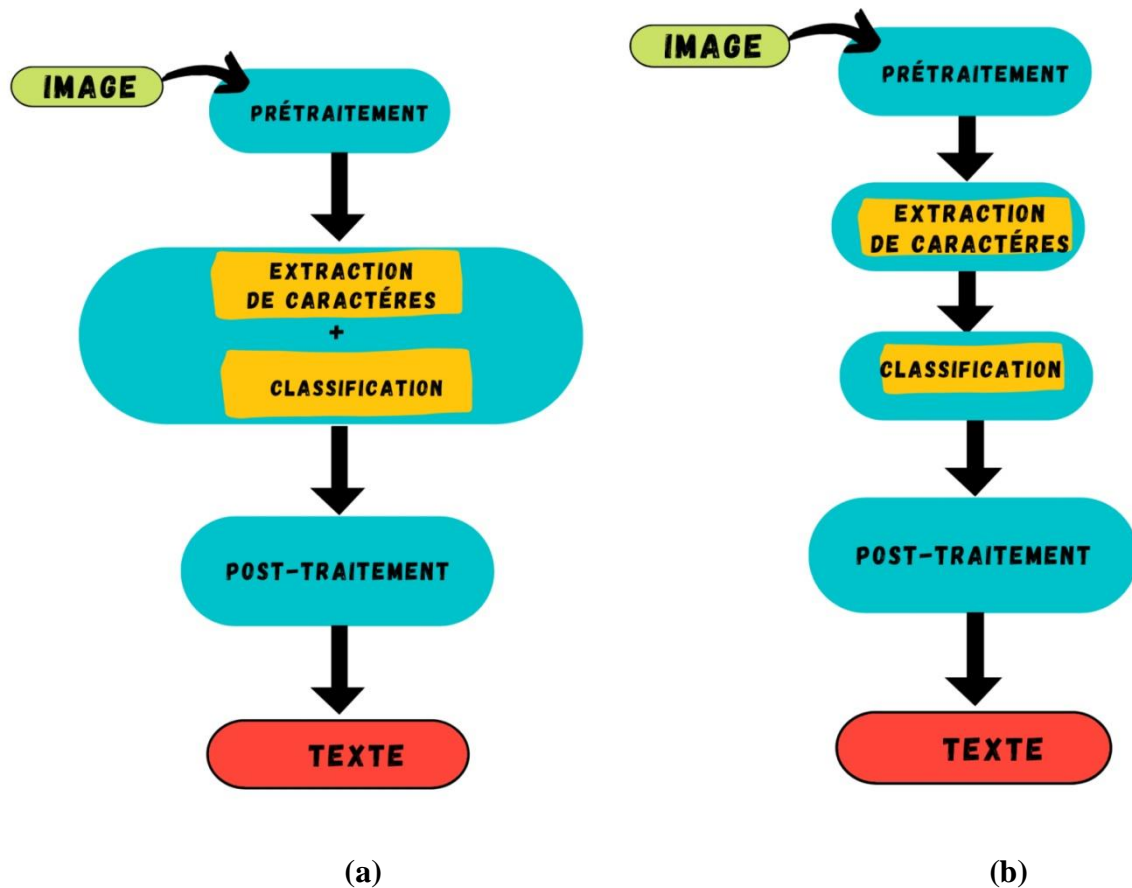


Figure I.9 : l'architecture d'organisation d'un système de reconnaissance

I.5.5 Phase de classification :

La classification est le développement d'une règle de décision qui traduit les attributs précis de forme qui partent à une classe (passage de l'espace de codage vers l'espace de décision) [13].

Les types de méthodes de classification sont divisés en deux familles : le mode supervisé et le mode non supervisé. S'il existe un ensemble de points marqués, on parlera de supervisée, tous ont réalisés une classification non supervisée, également appelée classification automatique [14].

I .5.6 Phase de post-traitement :

Le post-traitement est un traitement ultérieur de la classification, il est effectué lorsque le processus de reconnaissance a produit une liste de lettres ou de mots possibles. L'objectif principal est d'améliorer la reconnaissance par une correction orthographique ou morphologique à l'aide d'un dictionnaire et de corriger les erreurs de l'étape précédente en utilisant des informations lexicales, syntaxiques et sémantiques.

Le principal objectif de la phase de post-traitement au niveau lexicale est de déterminer quel est le meilleur mot du lexique pouvant correspondre aux hypothèses de reconnaissance.

Conclusion :

Nous avons présenté dans ce chapitre le domaine de la reconnaissance automatique de l'écriture, les étapes suivies pour la réalisation d'un système de reconnaissance, les différentes distinctions faites entre les systèmes en-ligne et les systèmes hors-lignes, et entre l'écriture imprimée et manuscrite.

CHAPITRE II

L'apprentissage Profond

II.1 Introduction :

L'intelligence artificielle est une discipline scientifique qui cherche à résoudre des problèmes d'une grande complexité logique ou algorithmique.

Le machine Learning (ML) est un champ d'étude de l'intelligence artificielle. Par conséquent, L'apprentissage profond est un ensemble de méthodes d'apprentissage automatique qui tentent de modéliser les données à un haut niveau d'abstraction par une architecture articulée de différentes transformations non linéaires. Dans ce chapitre, nous allons d'abord présenter les concepts liés à l'apprentissage profond.

II.2 Définition :

Le Deep Learning(DL) ou apprentissage profond est un sous domaine de machine Learning ou apprentissage machine, et aussi d'intelligence artificielle. Par ce dernier on comprend l'acte de la reproduction des tâches par des machines qui sont jugées comme complexes par les humains.

Typiquement : faire raisonner une machine, apprendre à une machine la planification de tâches, ou encore apprendre à une machine la représentation des connaissances d'une manière structurée.

La Figure II.1 représente la relation entre intelligence artificielle, machine Learning et l'apprentissage profond.

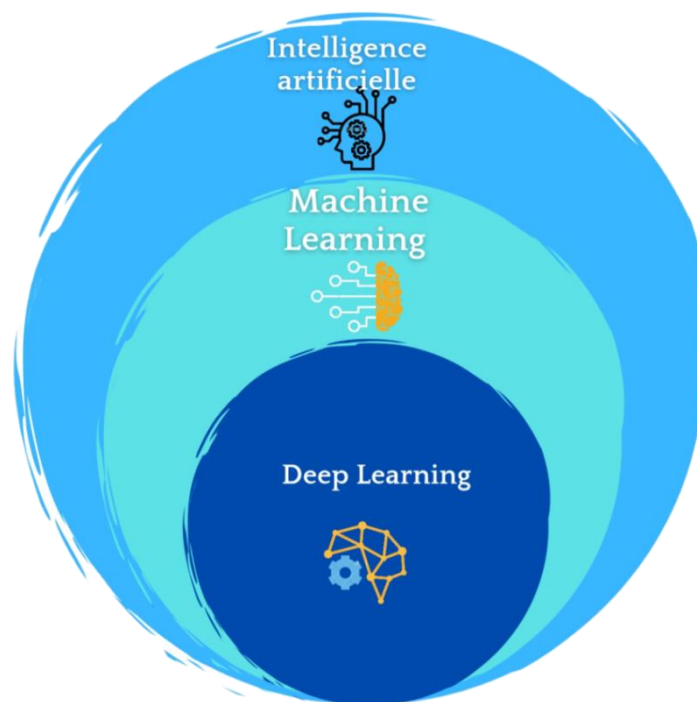


Figure II.1 L'intelligence artificielle et ses sous-domaines.

Le DL regroupe actuellement les méthodes les plus efficaces et les plus performantes appliquées dans la communauté de l'apprentissage automatique.

II.3 Historique :

L'histoire de l'apprentissage profond remonte à un domaine qui a changé de nom pour devenir aujourd'hui la cybernétique. Elle a commencé dans les années 1940 avec McCulloch et Pitts avec l'idée des neurones qui sont des unités de seuil avec des états de marche et d'arrêt.

En 1947, Donald Hebb a eu l'idée que les neurones biologiques apprennent en modifiant la force des connexions entre eux. C'est ce qu'on appelle l'hyper apprentissage, où si deux neurones sont activés ensemble alors la connexion entre eux augmente et s'ils ne le sont pas alors la connexion diminue. Puis il a été développé en 1948 par Norbert Wiener avec l'idée d'avoir des systèmes avec des capteurs et des actionneurs ; par contre en 1957, Frank Rosenblatt a proposé le Perceptron qui est un algorithme d'apprentissage qui modifie les poids de réseaux neuronaux très simples.

Durant les années 1960 le DL s'est éteint pour quelques raisons mais vite il a pris un nouvel essor en 1985 avec l'apparition de la rétro propagation. En 1995, le domaine a périclité à nouveau et la communauté de l'apprentissage machine a abandonné l'idée des réseaux neuronaux. Début 2010, les gens ont commencé à utiliser les réseaux de neurones dans la reconnaissance vocale avec une amélioration considérable des performances, et plus tard, ils ont été largement déployés dans le domaine commercial. En 2013, la vision par ordinateur a commencé à passer aux réseaux de neurones. En 2016, la même transition a eu lieu dans le traitement du langage naturel. Bientôt, des révolutions similaires se produiront dans la robotique, le contrôle et bien d'autres domaines [15].

II.4 Apprentissage automatique :

L'apprentissage automatique est le développement d'algorithmes qui permettent aux machines d'effectuer des tâches complexes sans être explicitement programmées pour le faire. Son objectif est d'extraire et d'exploiter automatiquement les informations présentes dans un ensemble de données [16]. La plupart des méthodes d'apprentissage automatique fonctionnent bien grâce aux représentations et aux fonctions de saisie conçues par l'homme. Il s'agit simplement d'optimiser les poids pour obtenir de meilleures prédictions finales.

II.4.1 Types d'apprentissage :

L'apprentissage automatique est segmenté en fonction de la nature des données d'entrée, puis en fonction de la nature des données de sortie.

- **Apprentissage supervisé :**

C'est une forme d'apprentissage automatique qui crée des modèles d'intelligence artificielle sur la base de données d'apprentissage "étiquetées".

- **Apprentissage non supervisé :**

Il s'agit d'une technique d'apprentissage automatique qui cherche une structure dans des données non étiquetées.

II.4.2 L'apprentissage automatique selon le type de résultat :

- **La classification :**

La classification est un type d'apprentissage supervisé qui détermine la catégorie dans laquelle l'élément appartient sur une base de données qui est un ensemble de données d'apprentissage contenant des observations dont la catégorie est connue [17].

- **La régression :**

La régression est une technique d'apprentissage supervisé qui prédit les valeurs continues d'un ensemble de données à partir d'une base d'apprentissage, qui prédit une variable de résultat continue (y) sur la base des valeurs d'une ou plusieurs variables prédictes (x). Les entrées sont appelées valeurs indépendantes et les sorties sont appelées valeurs dépendantes.

Il existe des poids appelés coefficients, qui déterminent la contribution de chaque valeur d'entrée au résultat, ou son importance [18].

II.5 Domaine d'application Deep Learning:

Les domaines d'application de Deep Learning sont utilisés dans des différents domaines [20] :

- ✓ La reconnaissance d'image.
- ✓ Les classifications de textes ou d'images.

- ✓ Identification d'objets.
- ✓ Prédiction de données.
- ✓ traduction automatique.
- ✓ voiture autonome.
- ✓ diagnostic médical.
- ✓ détection par caméra de pièces défectueuses.
- ✓ robots intelligents.

II.6 Réseaux de Neurons :

Les réseaux neuronaux permettent de simuler le fonctionnement des cellules nerveuses à l'aide d'automates. Un réseau neuronal est constitué d'un ensemble de neurones (nœuds) interconnectés par des liens qui permettent aux signaux de se propager entre eux.

Grâce à leurs capacités d'apprentissage, les réseaux neuronaux peuvent découvrir des relations non linéaires complexes entre un grand nombre de variables sans la nécessiter d'une intervention externe. De ce fait, ils sont largement utilisés dans de nombreux problèmes de classification (reconnaissance de formes, traitement de signal,...) d'estimation (modélisation de phénomènes complexes,...) et prévision (bourse, ventes,...). Il existe un compromis entre la clarté du modèle et son pouvoir prédictif. Plus un modèle est simple, plus il est facile à comprendre, mais moins il peut prendre en compte un trop grand nombre de dépendances différentes.

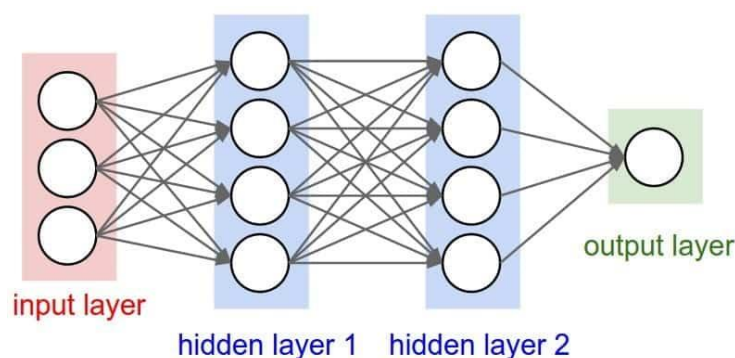


Figure II.2 structure de réseaux de neurones

II.7 Les fonctions d'activation :

Une fonction d'activation est une équation mathématique qui détermine la sortie de chaque élément (perceptron ou neurone) dans le réseau neuronal. Comme illustré en figure I.4, il prend l'entrée de chaque neurone et la transforme en sortie.

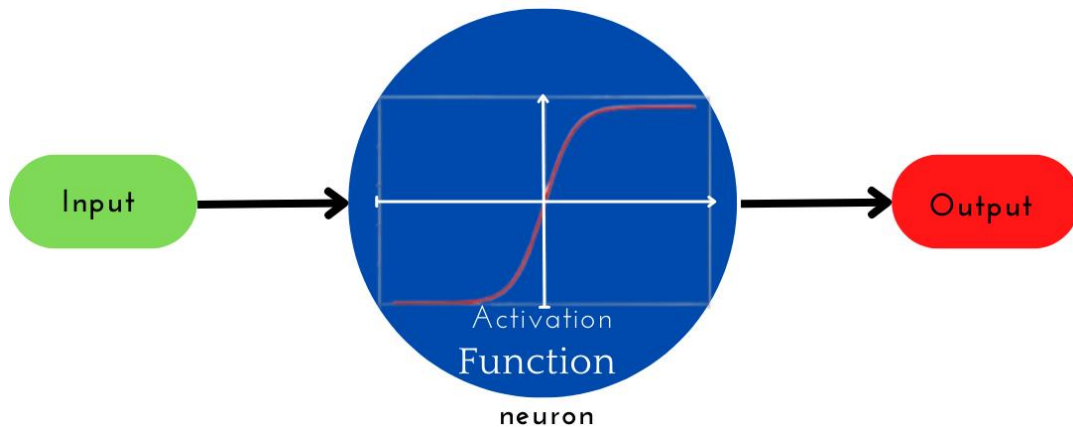


Figure II.3 Structure simplifié d'un neurone artificiel

Dans un réseau neuronal, les entrées, qui sont généralement des valeurs réelles, sont introduites dans les neurones du réseau. Chaque neurone a un poids, et les entrées sont multipliées par les poids et introduites dans la fonction d'activation, tel que cette dernière est une fonction qui permet de traiter l'information qui arrive à un neurone artificiel en machine Learning, comme le fait ceux du cerveau avec les signaux électriques qu'ils reçoivent [21]. La sortie de chaque neurone est l'entrée des neurones dans la couche suivante du réseau, et donc les entrées se succèdent à travers de multiples fonctions d'activation jusqu'à ce que finalement, la couche de sortie génère une prédiction. Les réseaux de neurones reposent sur des fonctions d'activation non linéaires, la dérivée de cette fonction aide le réseau à apprendre à travers le processus de rétro propagation.

Le Tableau II.1 ci-dessous décrit les principales fonctions d'activation existantes :

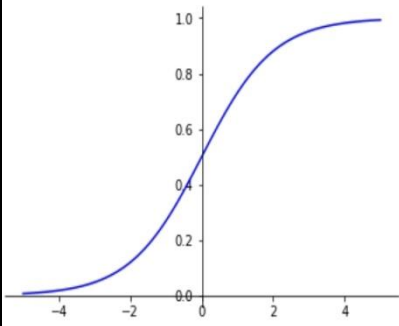
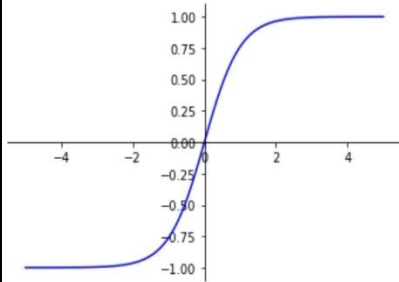
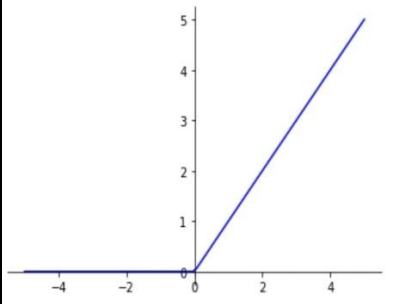
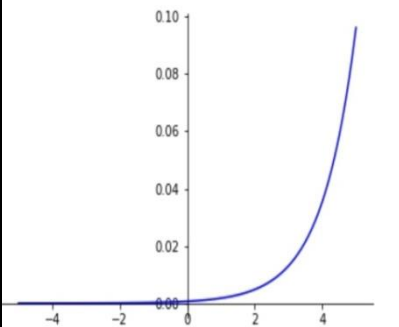
Noms	Description	Equation	Graphe
Sigmoïde	Elle génère des valeurs comprises entre zéro et un. Pour des valeurs très élevées ou basses des paramètres d'entrée, le réseau peut être très lent à atteindre une prédiction.	$f(x) = \frac{1}{1 + e^{-x}}$	
TanH	Centrée sur zéro, ce qui facilite la modélisation d'entrées fortement négatives, fortement positives ou neutres.	$\begin{aligned} \text{Tanh} &= \frac{\sinh(x)}{\cosh(x)} \\ &= \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^x - 1}{e^x + 1} \end{aligned}$	
ReLu	Très efficace en termes de calcul mais n'est pas en mesure de traiter des entrées qui approchent de zéro ou négatives.	$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x > 0 \end{cases}$	
Softmax	Utilisée pour les neurones de sortie. Elle normalise les sorties pour chaque classe entre 0 et 1 et renvoie la probabilité que l'entrée appartient à une classe spécifique.	$\sigma(x)_i = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}}$	

Tableau II.1 Les fonctions d'activation

II.8 Réseaux de neurones convolutionnels:

Les réseaux de neurones convolutionnels (ou CNN pour « Convolutional Neural Network») c'est l'extension des MLP qui remédient efficacement aux principales défauts de ces derniers. Ils sont conçus pour extraire automatiquement les caractéristiques de l'image d'entrée, sont invariants aux légères distorsions de l'image, sont insensibles aux distorsions mineures de l'image et mettent en œuvre le concept de partage des poids pour réduire considérablement le nombre de paramètres du réseau. Ce partage des poids permet également de prendre en compte de manière robuste les corrélations locales contenues dans les images. Afin de détecter des lignes, des points ou des coins à toutes positions possibles dans l'image, les poids sont forcés d'être égaux et mettant ainsi en œuvre l'idée de partage des poids. [22]

II.8.1 Architecture de réseaux de neurones convolutionnels

Le réseau neuronal convolutif est basé sur le perceptron multicouche (MLP), qui s'inspire du comportement du cortex visuel des vertébrés. Il est bien efficace pour le traitement d'images.

Les MLP ont beaucoup de mal à gérer des images de grande taille, ce qui est dû à la croissance exponentielle du nombre de connexions avec la taille de l'image [23].

Les réseaux neuronaux convolutifs sont constitués de plusieurs couches comme le montre la Figure II.4 :

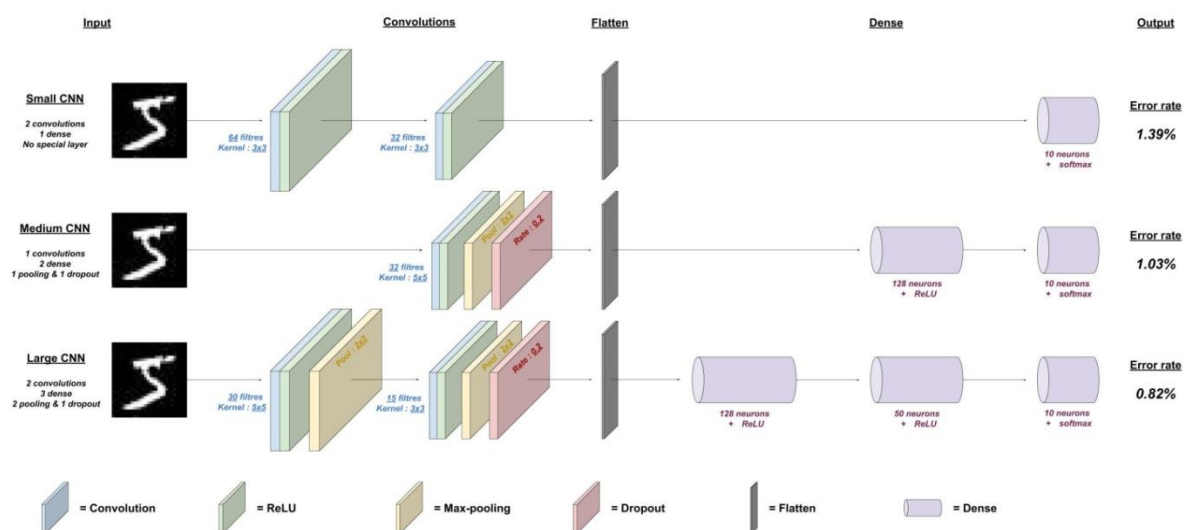


Figure II.4 Architecture d'un réseau CNN

II.8.2 Les couches CNN :

Une architecture CNN est formée par un empilement de couches de traitement indépendantes :

- La couche de convolution (CONV) qui traite les données d'un champ récepteur
- La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
- La couche de correction (ReLU), souvent appelée par abus 'ReLU' en référence à la fonction d'activation (Unité de rectification linéaire) [23].

II.8.2.1 La couche de convolution :

La convolution, d'un point de vue simpliste, est le fait d'appliquer un filtre mathématique à une image et pour un point de vue plus technique, il s'agit de faire glisser une matrice par-dessus d'image, et pour chaque pixel, utiliser la somme de la multiplication de ce pixel par la valeur de la matrice.

Cette technique nous permet de trouver des parties d'image qui pourraient nous être intéressantes. Prenons la Figure ci-dessous à gauche comme exemple d'image et droite comme exemple du filtre.

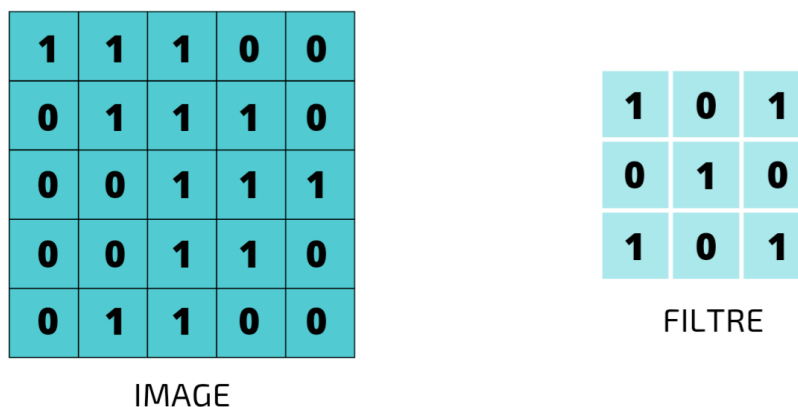


Figure II.5 Exemple simpliste des valeurs des pixels d'une image 5x5 et de valeurs d'une matrice utilisée comme filtre

Dans le cas de la Figure II.6 qui suit, les valeurs sont en binaires et concernant le cas réel, les valeurs devraient varier entre 0 et 255.

Application du filtre sur l'image : dans la matrice d'image, nous pouvons voir que chaque valeur des pixels de l'image tuile est multipliée par chaque valeur correspondante du filtre (1x1, 1x0, 1x1 etc.), Puis additionner tous ces valeurs pour obtenir une seule valeur '4' qui fera partie d'une nouvelle image convolutivee.

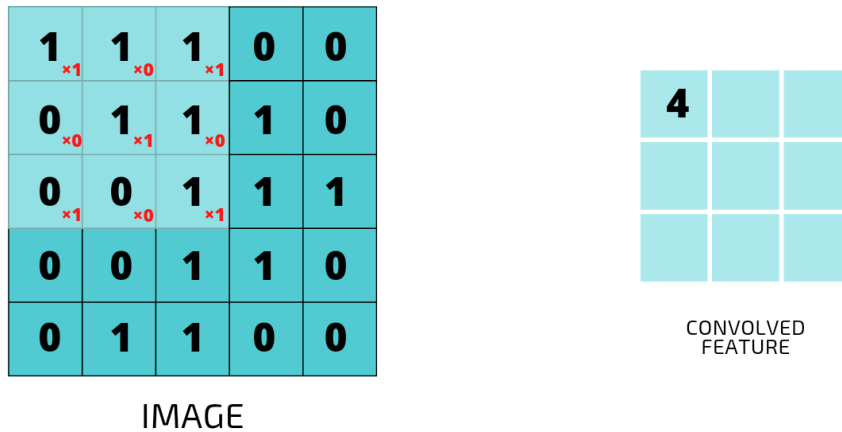


Figure II.6 Appliquer le filtre sur l'image

Le filtre doit se déplacer d'une case à chaque itération jusqu'à ce que la première ligne soit finie. Lorsque nous avons fini la première ligne, le filtre « descend » d'une case et la même procédure se répète pour chaque ligne et colonne. Voir la Figure II.7:

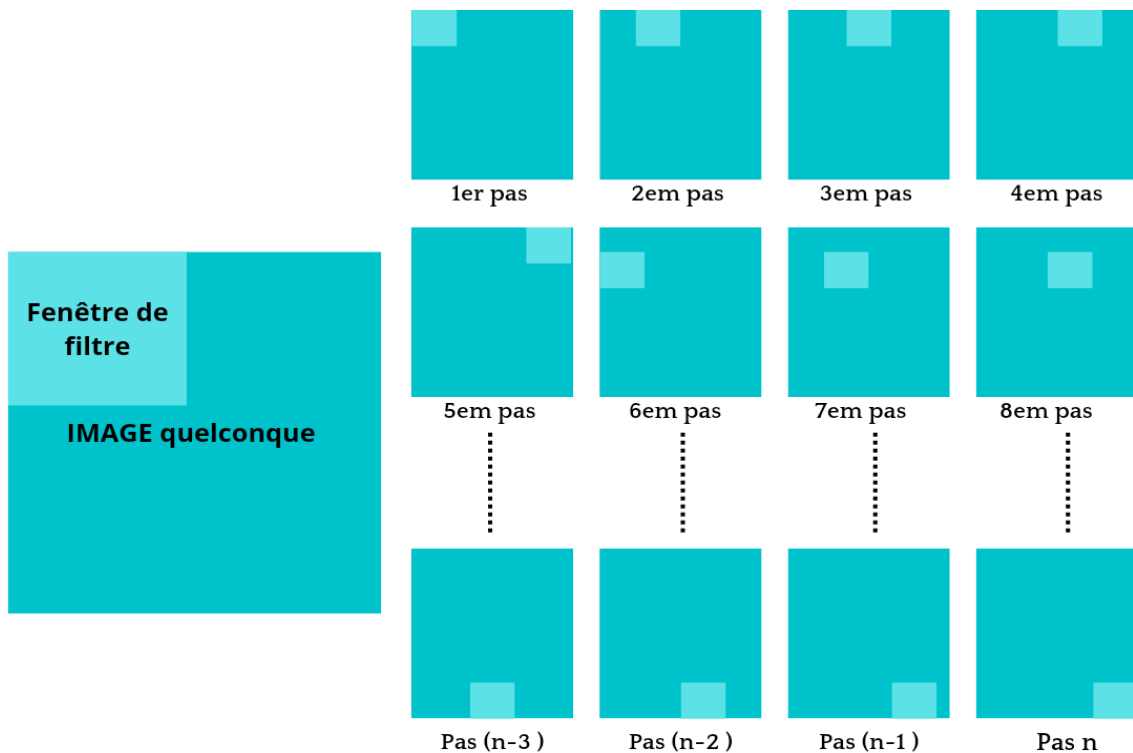


Figure II.7 Parcours de la fenêtre de filtre sur l'image.

II.8.2.2 Couche de pooling (POOL) :

Une couche de pooling (en anglais pooling layer ou POOL) est une opération de sous-échantillonnage qui est généralement appliquée après chaque couche convolutionnelle. [24]

En particulier, les types de pooling les plus populaires sont le max et l'average pooling, où les valeurs maximales et moyennes sont prises, respectivement.

La couche de pooling permet de réduire le nombre de paramètres par conséquent d'accélérer les calculs dans le réseau. Elle permet aussi d'améliorer l'efficacité du réseau et éviter le problème de sur-apprentissage. [24]

On représente deux types de couches de pooling qui sont les suivantes :

✚ Max pooling :

Chaque opération de pooling sélectionne la valeur zone du feature map

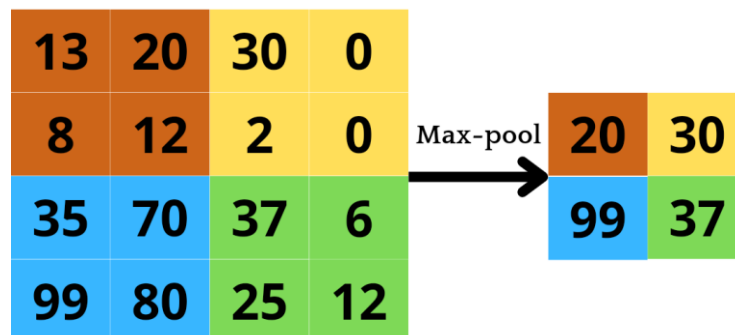


Figure II.8 Max pooling.

✚ Average pooling :

Appelé aussi Mean Poolin calcule la moyenne des éléments présents dans le feature map couverte par le filtre.

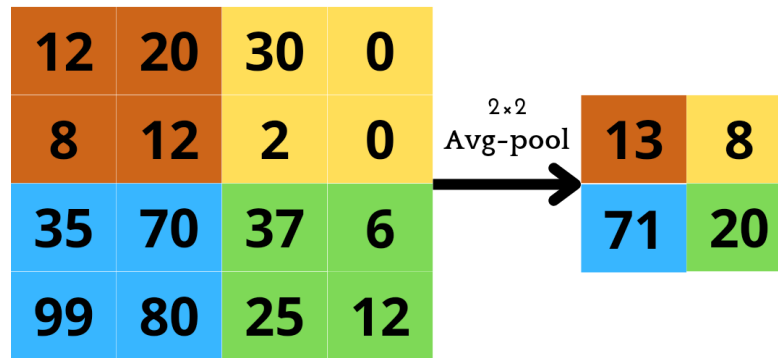


Figure II.9 Average pooling

II.8.2.3 La couche de correction ReLU :

ReLU (Rectified Linear Units) désigne la fonction réelle non-linéaire définie par :

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

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation. [25]

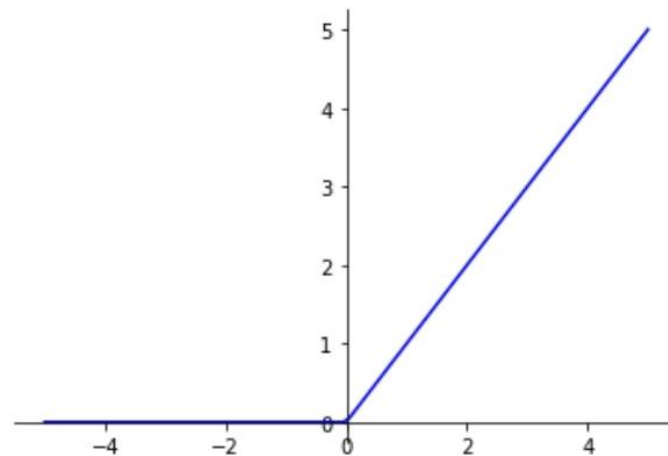


Figure II.10 allure de la fonction ReLU

II.8.2.4 La couche fully-connected:

Constitue toujours la dernière couche d'un réseau de neurones, convolutifs ou non, elle n'est donc pas caractéristique d'un CNN.

Ce type de couche reçoit un vecteur en entrée et produit un nouveau vecteur en sortie. Pour cela, elle applique une combinaison linéaire puis éventuellement une fonction d'activation aux valeurs reçues en entrée.

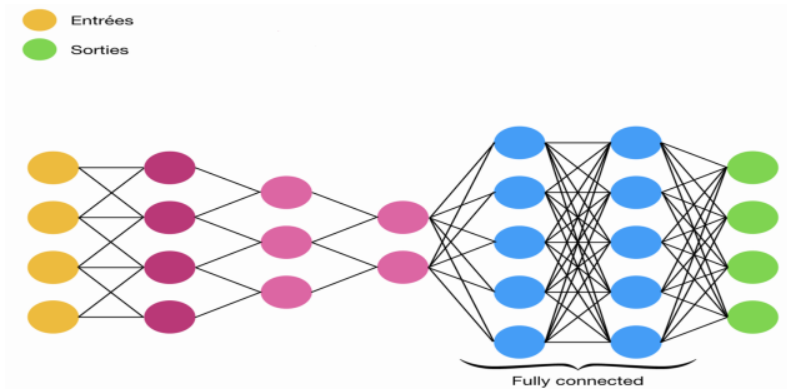


Figure II.11 Représentation de la couche fully-connected

II.8.3 Apprentissage des réseaux de neurones convolutifs :

L'apprentissage profond consiste à former un réseau neuronal composé d'une série de modules, chacun représentant une étape de traitement et Chaque module a des paramètres de Poids similaires à ceux des réseaux neuronaux classiques. Tous ces paramètres sont ajustés de manière à rapprocher la sortie estimée du système de la sortie réelle sur le terrain.

Cela peut être réalisé par des algorithmes d'optimisation basés sur la descente de gradient stochastique par lots (ou mini-batch).

Un élément important dans les algorithmes de descente de gradient est le pas (ou taux) d'apprentissage. Si le pas est trop petit, l'algorithme devra effectuer un grand nombre d'itérations pour converger et prendra beaucoup de temps. Inversement, si le pas est trop élevé, l'algorithme risque de diverger et de s'éloigner ainsi de la bonne solution. En outre, cette étape d'apprentissage est globale, ce qui signifie que tous les neurones utilisent le même taux, plutôt que toutes les données suivent nécessairement la même distribution, de sorte qu'il n'est pas nécessaire que le réseau soit réglé de la même manière.

Pour surmonter ces problèmes, de nombreuses variantes d'algorithme de descente de gradient stochastique ont été proposé. Parmi elles, on cite la méthode adaptative Adam que nous avons

utilisée dans nos tests. Adam est l'un des algorithmes les plus récents et les plus efficaces pour l'optimisation par descente de gradient. Il calcule un taux d'apprentissage adaptatif pour chaque paramètre. En outre, le gradient dépend des estimations adaptatives des moments de premier et second ordre. [26-27]

II.8.4 Les Avantages de CNN :

Un avantage majeur des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution.

Cette méthode réduit l'empreinte mémoire, elle passe aussi à l'amélioration des performances et permet une invariance du traitement par translation. C'est le principal avantage du réseau de neurones convolutifs par rapport au perceptron multicouches, que lui considère chaque neurone indépendant et affecte donc un poids différent à chaque signal entrant. [24]

Conclusion :

Dans ce chapitre, nous avons parlé d'apprentissage automatique et de ses domaines d'application, avec ses notions importantes comme définition, Domaine d'application, etc. Et quelques réseaux de neurones convolutionnels, tous on cite leurs plateformes.

Dans le chapitre qui suit nous allons procéder à l'application de ces réseaux de neurone convolutionnels dans la reconnaissance de l'écriture amazigh.

Chapitre III

Reconnaissance de l'écriture Amazigh avec DL et l'attribut HMB

III.1 Introduction :

L'extraction d'attributs à partir d'images peut être effectuée de deux manières. Premièrement, la manière la plus classique consiste à concevoir des attributs spécialement adaptées à la tâche et parfois même adaptées à un ensemble de données spécifique (écriture, formes, etc.). Ces attributs sont appelées manuels (handcrafted) puisqu'un algorithme a été conçu manuellement pour les extraire, incorporant des informations à priori sur les spécificités des données parmi lesquels on trouve l'attribut HMB (Histogram of marked background) [31]. La deuxième catégorie d'attributs consiste à apprendre automatiquement les attributs à partir des images en utilisant l'apprentissage automatique. C'est une solution qui est de plus en plus utilisée depuis l'avènement du Deep Learning avec les Réseau neuronal convolutif (CNN) [32-34].

Dans le but de palier à l'inconvénient principal de l'utilisation du Deep Learning qui réside dans la nécessité de disposer de beaucoup de données d'apprentissage, nous proposons de combiner entre l'attribut HMB et le CNN pour la reconnaissance des caractères manuscrits Amazighe.

III.2 HMB (Histogram of marked background):

Le principe générale de l'attribut HMB est d'injecter des informations discriminantes dans l'image du texte et cela en attribuant des marques aux pixels du fond de l'image du texte. Les pixels de fond sont marqués selon la répartition spatiale des pixels d'encre dans leur voisinage.

L'attribut HMB est extrait en deux étapes : marquage du fond de l'image et calcul des histogrammes [31]. Dans notre mémoire on utilise uniquement la phase de marquage d'image.

III.2.1 Marquage du fond :

L'idée principale de HMB est d'enrichir l'image du texte pour qu'elle soit plus représentative en intégrant des informations contextuelles dans le fond de l'image du texte. Pour atteindre cet objectif, des marques sont attribuées pour les pixels de fond ayant des pixels d'encre dans leur voisinage. Pour Ceci, les auteurs ont considérés un rayon « r » pour délimiter la région dans laquelle les voisins d'encre sont identifiés et des directions « d » dans lesquelles ils cherchent la présence de pixels d'encre voisins [31].

Dans notre cas nous avons utilisé un rayon de 4 pixels et les directions montrées dans la figure III.1

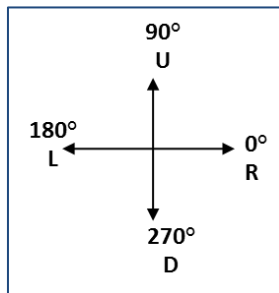


Figure III.1 Configuration avec 4 directions

Si on considère une image d'un mot « I » après nettoyage et binarisation définie par l'équation (1).

$$I(i,j) = \begin{cases} 1 & \text{if } I(i,j) \text{ is ink} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Pour chaque pixel de fond $I(i,j)=0$, les auteurs [31] déterminent une marque $M(i,j)$. Pour ce faire, ils considèrent un cercle « Cr » avec un rayon de « r » pixels centré au pixel de fond $I(i,j)$ à marquer. Ensuite, ils cherchent la présence de pixels d'encre dans chaque direction montrée dans la figure1. Enfin, la marque $M(i,j)$ est définie par les directions dans lesquelles ils ont trouvé des pixels d'encre dans le cercle "Cr". L'image marquée est définie par l'équation (4).

$$I_{marked}(i,j) = \begin{cases} 1 & \text{if } I(i,j) \text{ is ink} \\ M(i,j) & \text{otherwise} \end{cases} \quad (4)$$

Tableau III.1 montre toutes les possibilités de marques en binaire et en décimal en utilisant 4 direction.

R	U	UR	L	LR	LU	LUR	D	DR	DU	DUR	DL	DLR	DLU	DLUR
0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Tableau III.1 : Toutes les possibilités de marques avec quatre directions et leurs codes binaires ainsi que leurs signes décimaux correspondants [31]

III.2.2 Exemples de marquage :

Dans cette section nous présentons un exemple illustrant le marquage des pixels de fond d'une image d'un mot suivant les directions (0°,90°,180°,270°) et délimité par un rayon de 4 pixels. Une couleur différente est utilisée pour chaque marque.

Dans la Figure III.2 le pixel de fond mis en évidence est marqué avec la marque **R** qui correspond au code binaire (0001) et la valeur décimale **2** car il a un seul voisin d'encre dans la région délimitée par le cercle de rayon de 4 pixels et dans la direction droite (R) illustré avec la flèche verte [31].

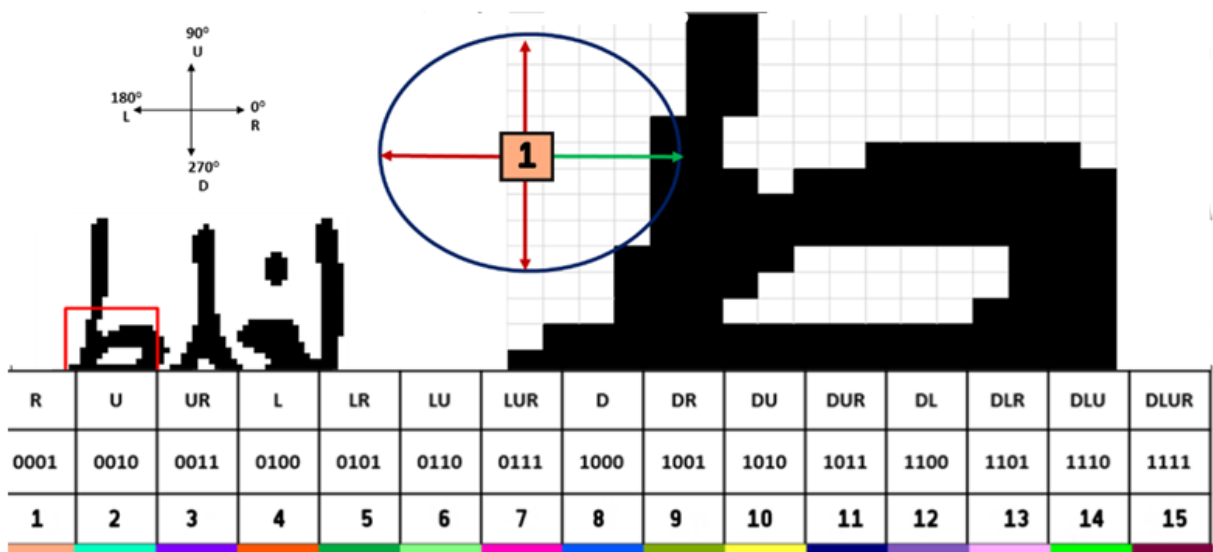


Figure III.2.Exemple de marquage d'un pixel de fond ayant un seul voisin d'encre [31]

La figure III.3 montre un exemple d'images après le processus de binarisation et marquage de quelques caractères de la base de caractère amazigh AMHCD.

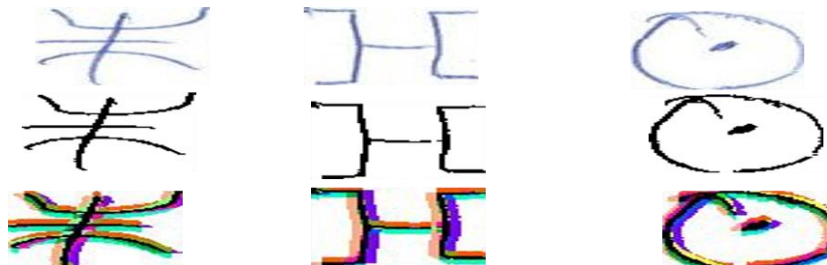


Figure III.3 : exemples d'images de caractères après le processus de binarisation et marquage de la base de caractère amazigh AMHCD.

III.3 Architecture générale de notre proposition :

La figure III.4 présente le schéma général de notre système pour la reconnaissance des caractères amazigh manuscrits. L'image du caractère amazigh passe par l'étape de binarisation puis par le marquage du fond en utilisant l'attribut HMB [33]. Puis l'image marquée sera passé au processus d'apprentissage automatique par Deep Learning en utilisant les CCN.

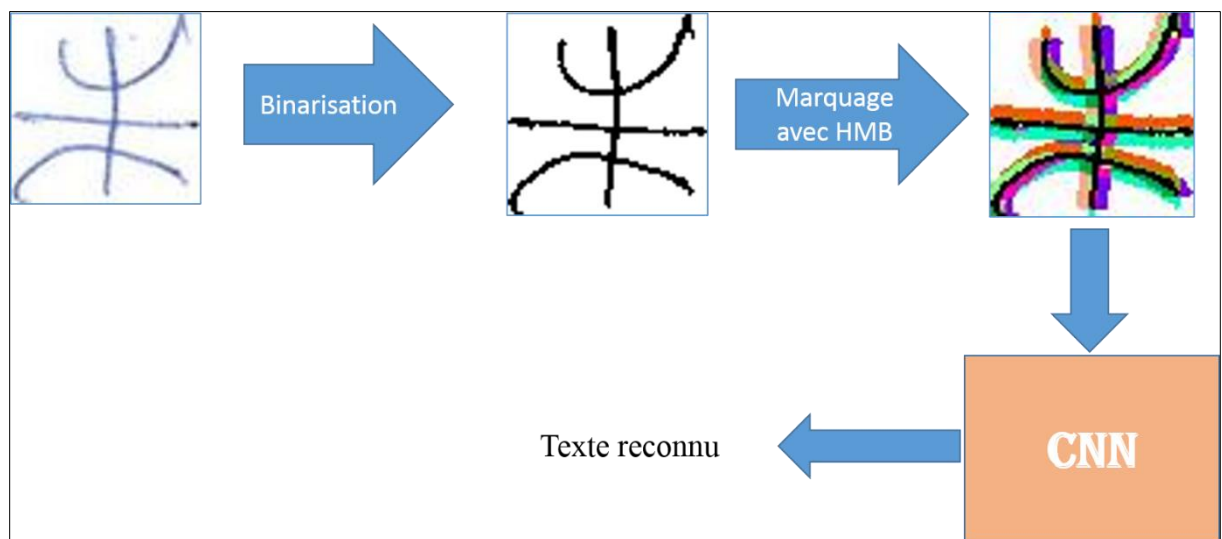


Figure III.4 Schéma générale de notre système

III.4 Logiciel et librairies utilisé dans l'implémentation :

III.4.1 Python :

Python est le langage de programmation open source le plus utilisé par les informaticiens. Ce langage est devenu un leader dans la gestion des infrastructures, l'analyse des données et le développement de logiciels. C'est un langage simple, facile à apprendre et permet une bonne réduction du coût de la maintenance des codes.

Les bibliothèques (packages) encouragent la modularité et la réutilisation du code. Sur la plupart des plates-formes, Python et ses bibliothèques sont librement disponibles (sous forme de code source ou de binaires) et peuvent être redistribués gratuitement.

III.4.2 Tensorflow :

TensorFlow est une bibliothèque logicielle open source utilisé dans l'apprentissage automatique pour les réseaux neuronaux. tensorflow a été développé par Google en 2011 sous le nom de DistBelief et a été officiellement publié gratuitement en 2017. La bibliothèque est capable de fonctionner sur plusieurs CPU et GPU et peut être utilisée sur plusieurs plateformes, y compris les plateformes mobiles. Le nom vient des tableaux multidimensionnels appelés tenseurs, qui sont couramment utilisés dans les réseaux neuronaux [28].

III.4.3 keras :

Keras est une API d'apprentissage profond de haut niveau développée par Google, Elle est écrite en Python et utilisée pour faciliter la mise en œuvre des réseaux neuronaux. Elle prend également en charge le calcul de plusieurs réseaux neuronaux en amont [29].

Parmi Les modules de Keras qu'on a utilisé on a les suivants :

- **Keras.models** : Sequential
- **Keras.layers** : Conv2D MaxPooling2D Flatten Dense et Dropout
- **Keras.preprocessing.image** : ImageDataGenerator

III.5 conception :

Notre conception s'articule autour de deux processus principaux qui sont le train pour l'apprentissage et le test pour les résultats.

III.5.1 Train :

C'est le processus le plus important, car nous allons créer notre modèle en configuration précise.

- **Base de données :**

C'est une base de données de caractères manuscrits de la langue berbère en tifinagh, elle se présente en plusieurs classes :

[ya, Yab, Yach, Yad, Yadd, Yae, Yaf, Yag, Yagh, Yagw, Yah, Yahh, Yaj, Yak, Yakw, Yal, Yam, Yan, Yaq, Yar, Yarr, Yas, Yass, Yat, Yatt, Yaw, Yax, Yay, Yaz, Yazz, Yey, Yi, Yu]

On a aussi un fichier texte sous le nom labels qui portera les noms de 33 classes utilisées dans notre base de données.

- **CNN et paramètres** : c'est notre algorithme de création d'un réseau de neurones convolutionnels qui sera configuré avec des paramètres suivants: nombre d'époques, nombre de filtre, nombre de couche ...etc.

III.5.2 Test :

- **La base de données** : elle s'agit d'une base de données d'images rapportées par classes, contenant un certain nombre d'images.
- **L'affichage de la classification** : nous allons afficher la sortie du modèle (c'est un fichier généré dans notre training.), qui est le nom d'une classe

III .6 configuration utilisés dans l'implémentation :

Appareil utilisé	toshiba-i5-PC
Processeur	Intel(R) Core(TM) i5-4200M CPU @ 2.50GHz 2.50 GHz
Ram	4,00 Go
Type du system	Système d'exploitation 64 bits, processeur x64
Edition	Windows 10 Professionnel

Tableau III.2 configuration utilisés dans l'implémentation

III .7 Base de données :

La base de données de caractères manuscrits amazighs (AMHCD) est la seule grande base de données berbère disponible. Ce jeu de données a été créé et développé au laboratoire IRF-SIC de l'Université d'Ibn Zohr à Agadir, Maroc (Es-Saady, Y., Rachidi, A., El Yassa,

M., & Mammass, D. (2011). amhcd : une base de données pour la recherche sur la reconnaissance de caractères manuscrits amazighs. (International Journal of Computer Applications).

L'AMHCD contient 780 images scannées de chaque caractère parmi les 33 caractères Tifinagh-IRCAM, soit un total de $780 \times 33 = 25\,740$ images écrites par 60 auteurs (homme /femme), d'âge et de profession différents [30]

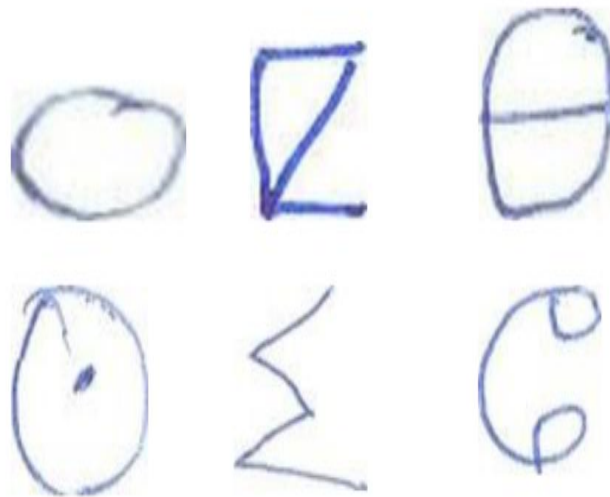


Figure III.5 structure de la base de données AMHCD

III.8 Architecture du réseau CNN :

Dans nos expérimentations, nous avons créée trois modèles avec des architectures différentes (2 couches, 4 couches et 6 couches) qu'on a appliquées à la base de données AMHCD on applique deux modèles différents : avec la binarisation et marquage et sans binarisation et marquage.

Pour chaque modèle, on a fait une évaluation sur le nombre d'époques. Après nous décrirons la structure des trois modèles à deux basses et les résultats obtenus dans chaque cas.

III.8.1 Réseau neuronal convolutif à 2 couches :

L'architecture du réseau neuronal convolutif à 2 couches pour la reconnaissance des caractères tfinagh manuscrit qui est basée sur le réseau de neurones convolutionnel comme illustré dans la **Figure III.6**.

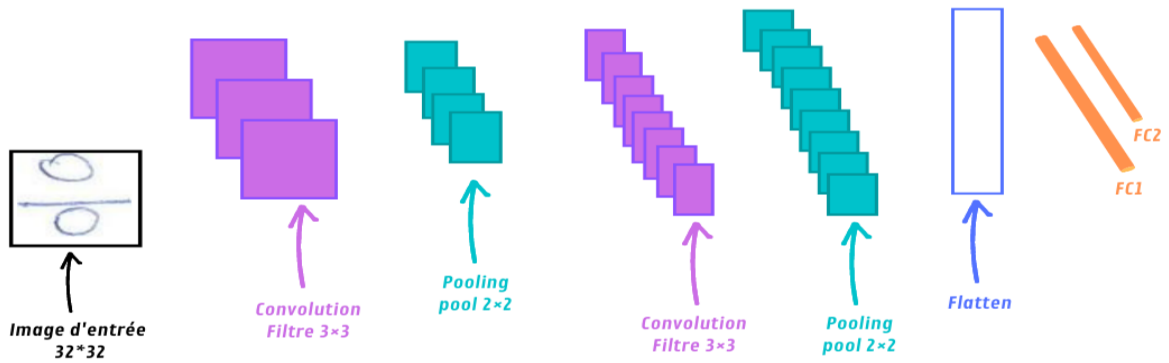


Figure III.6 Architecture du modèle à 2couches

III .8 .2 réseau neuronal convolutif à 4 couches :

Le modèle que nous présentons dans la figure III.7 est composé de quatre couches de Convolution, deux couches de maxpooling et de trois couches de fully-connected.

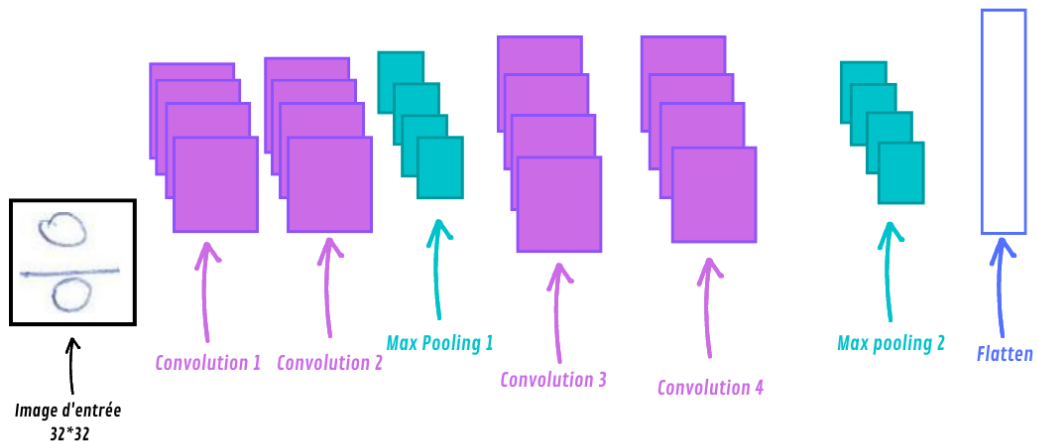


Figure III.7 Architecture du modèle à 4couches.

III.8.3 réseau neuronal à 6 couches :

Dans ce troisième modèle, on a changé le nombre de couches pour la convolution et le Max pooling, notre architecture maintenant est composée de six couches convolution (trois couche de max pooling / trois couche de fully connected).

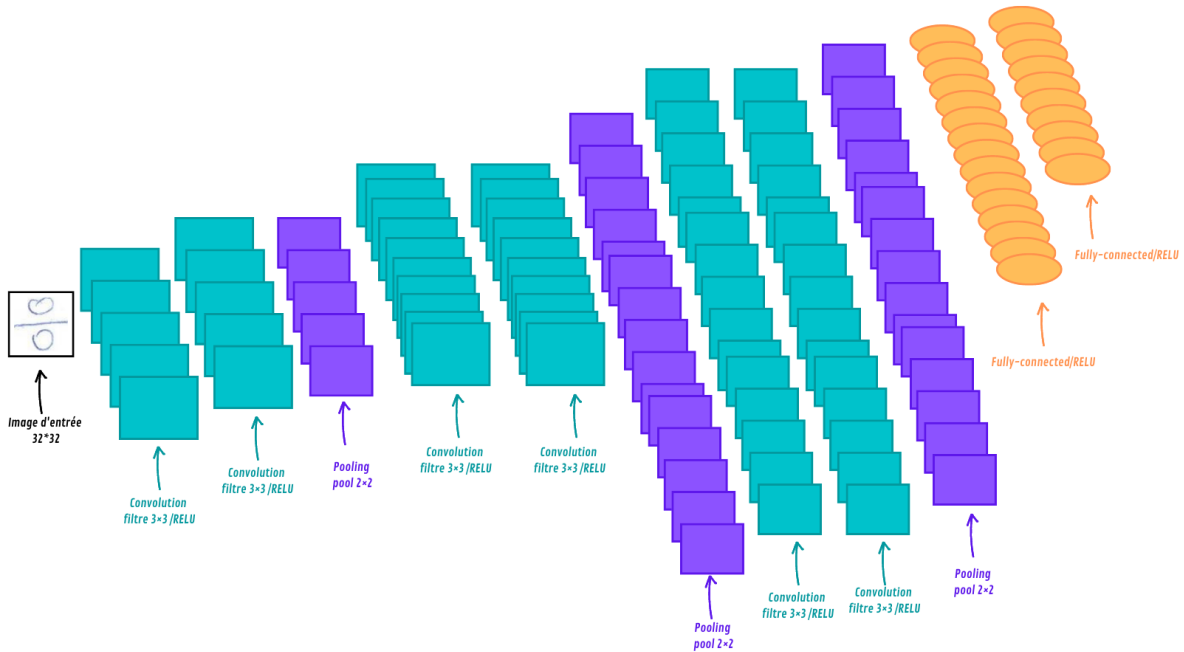


Figure III.8 Architecture du modèle à 6 couches

III.9 Code source de notre architecture

```

train_datagen=ImageDataGenerator(rescale=(1./255),shear_range=0.2,zoom_range=0.2,horizontal_flip=(True))
test_datagen=ImageDataGenerator(rescale=(1./255),shear_range=0.2,zoom_range=0.2,horizontal_flip=(True))
training_set= train_datagen.flow_from_directory(path_train,target_size=(32,32),class_mode='categorical')
test_set= test_datagen.flow_from_directory(path_test,target_size=(32,32),class_mode='categorical',shuffle=False)
import numpy as np
test=np.array(test_set[0][0])
num_classes=33
    
```

- ✓ Charger et ajuster la base de données d'image .
- ✓ 25740 Image pour l'entraînement (training) et tester (test).
- ✓ Compilation du modèle

```

model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',padding='same',input_shape=(32,32,3)))
model.add(Conv2D(32, kernel_size=(3, 3),activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(64, kernel_size=(3, 3),activation='relu'))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2),strides=(2,2),padding='same'))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(33, activation='softmax'))
    
```

- ✓ Construction de notre modèle CNN avec 4 couche

```

model.compile(loss='categorical_crossentropy', optimizer='adam',metrics =['accuracy'])
    
```


- ✓ Compilation et optimisation du modèle

```
history=model.fit(training_set,epochs=30,validation_data=test_set,validation_steps=105)
```

- ✓ Lancement d'apprentissage

```
score= model.evaluate(test_set)
```

- ✓ Phase d'évaluation

III.10 Les différents cas d'implémentation :

III.10.1 Sans binarisation et marquage :

- ✓ 1^{er} cas : « train 2/3, test 1/3 » :

Nous avons utilisé un 1/3 pour le test avec 260 images insérées et 2/3 pour le train avec 520 images pour l'apprentissage parmi 780 images existantes.

	Architecture utiliser			Nombre d'époque	Precision	Erreur
	Couche de convolution	Couche de pooling	Couche de Fully connected			
Modèle 1 2 couches	2	2	2	10	95%	5%
				20	96%	4%
				30	96%	4%
Modèle 2 4 couches	4	2	3	10	98%	2%
				20	98%	2%
				30	98%	2%

Modèle 3 6 couches	6	3	3	10	97%	3%
				20	98%	2%
				30	97%	3%

Tableau III.3 : Comparaison entre les différentes architectures et époques cas train 2/3, test 1/3.

Commentaire :

Après l'exécution de notre cas « 1/3 pour tester et 2/3 pour l'apprentissage » on remarque que le meilleur résultat obtenu est dans le modèle de 4 couches avec 4 couches de convolutions, 2 couches de pooling et 3 couches de fully connected par contre dans le dernier modèle de 6 couches on a eu le sur apprentissage après avoir augmenté le nombre de couches utilisées.

✓ **2ème cas : « train 1/3, test 2/3 » :**

Nous avons utilisé un 2/3 pour le test avec 520 images insérées et 1/3 pour le train avec 260 images pour l'apprentissage parmi 780 images existantes.

	Architecture utiliser			Nombre d'époque	Precision	Erreur
	Couche de convolution	Couche de pooling	Couche de Fully connected			
Modèle 1 2 couches	2	2	2	10	91%	9%
				20	93%	7%
				30	94%	6%

Modèle 2 4 couches	4	2	3	10	94%	6%
				20	96%	4%
				30	97%	3%
Modèle 3 6 couches	6	3	3	10	93%	7%
				20	96%	4%
				30	96%	4%

Tableau III.4 : Comparaison entre les différentes architectures et époques cas train 1/3,

test 2/3

Commentaire :

Da le cas de « 2/3 pour le test et 1/3 pour le training » on remarque la diminution des pourcentages d'apprentissage tous en comparons avec les résultats obtenus dans le cas précédent (le cas inverse) alors on constate que pour avoir bon résultats parmi les deux cas utiliser on doit augmenter le nombre des caractères utiliser dans le train pour bien faire son apprentissage et mieux apprendre pour bien tester.

✓ **3ème cas : « train 1/2, test 1/2 » :**

Nous avons utilisé un 1/2 pour le test avec 390 images insérées et 1/2 pour le train avec le même nombre d'images pour l'apprentissage parmi 780 images existantes.

	Architecture utiliser			Nombre d'époque	Precision	Erreur
	Couche de convolution	Couche de pooling	Couche de Fully connected			
Modèle 1 2 couches	2	2	2	10	92%	8%
				20	95%	5%
				30	94%	6%
Modèle 2 4 couches	4	2	3	10	96%	4%
				20	97%	3%
				30	97%	3%
Modèle 3 6 couches	6	3	3	10	96%	4%
				20	95%	5%
				30	97%	3%

Tableau III.5 : Comparaison entre les différentes architectures et époques cas test 1/2,

train 1/2

Commentaire :

Dans ce cas ou on a le même nombre de caractère dans le test que dans le train alors on remarque après l'exécution qu'on a obtenus des résultats moyen après la comparaison avec les deux cas utilisés précédemment avec un meilleur pourcentage dans le modèle de 4 couches.

III.10.2 Avec binarisation et marquage :

✓ 1 er cas : « train 2/3, test 1/3 » :

Nous avons utilisé un 1/3 pour le test avec 260 images insérées et 2/3 pour le train avec 520images pour l'apprentissage parmi 780 images existantes.

	Architecture utiliser			Nombre d'époque	Precision	Erreur
	Couche de convolution	Couche de pooling	Couche de Fully connected			
Modèle 1 2 couches	2	2	2	10	97%	3%
				20	98%	2%
				30	98%	2%
Modèle 2 4 couches	4	2	3	10	98%	2%
				20	99%	1%
				30	99%	1%

Modèle 3 6 couches	6	3	3	10	98%	2%
				20	97%	3%
				30	98%	2%

Tableau III.6 : Comparaison entre les différentes architectures et époques cas train 2/3, test 1/3

✓ **2ème cas : « train 1/3, test 2/3 » :**

Nous avons utilisé un 2/3 pour le test avec 520 images insérées et 1/3 pour le train avec 260 images pour l'apprentissage parmi 780 images existantes.

	Architecture utiliser			Nombre d'époque	Precision	Erreur
	Couche de convolution	Couche de pooling	Couche de Fully connected			
Modèle 1 2 couches	2	2	2	10	91%	9%
				20	93%	7%
				30	93%	7%
Modèle 2 4 couches	4	2	3	10	92%	8%
				20	94%	6%
				30	94%	6%

Modèle 3 6 couches	6	3	3	10	91%	9%
				20	92%	8%
				30	93%	7%

Tableau III.7 : Comparaison entre les différentes architectures et époques cas train 1/3, test 2/3

✓ **3ème cas : « train 1/2, test 1/2 » :**

Nous avons utilisé un 1/2 pour le test avec 390 images insérées et 1/2 pour le train avec le même nombre d'images pour l'apprentissage parmi 780 images existantes.

	Architecture utiliser			Nombre d'époque	Precision	Erreur
	Couche de convolution	Couche de pooling	Couche de Fully connected			
Modèle 1 2 couches	2	2	2	10	95%	5%
				20	96%	4%
				30	97%	3%
Modèle 2 4 couches	4	2	3	10	97%	3%
				20	98%	2%
				30	98%	2%

Modèle 3 6 couches	6	3	3	10	96%	4%
				20	98%	2%
				30	97%	3%

Tableau III.8 : Comparaison entre les différentes architectures et époques cas test 1/2, train1/2

Commentaire :

Pour les trois résultats des modèles (2couches, 4couches et le 6couches) avec les différents cas on remarque que avec la binarisation et marquage on à pu avoir de le meilleur résultat souhaité avec un pourcentage aller jusqu'à 99% pour les 4couches de convolution, 2couches de pooling et 3couches pour la fully connected dans le cas ou en a « 1/3 pour le test et 2/3 de train ».

Après avoir exécuté tous les cas possible dans la binarisation et marquage et sans binarisation et marquage en remarque que le meilleur cas c'est ou le trining contient un nombre d'image supérieure par rapport au test (2/3 train - 1/3 test), alors en a opter pour la combinaison des deux base d'images avec « binarisation et marquage » et « sans binarisation et marquage » pour avoir le meilleur et le plus précis des résultats. Le tableau III.9 montre les résultats qu'on a obtenus :

	Architecture utiliser			Nombre d'époque	Precision	Erreur
	Couche de convolution	Couche de pooling	Couche de Fully connected			
Modèle 1 2 couches	2	2	2	10	95%	5%
				20	95%	5%
				30	97%	3%
Modèle 2 4 couches	4	2	3	10	98%	2%
				20	98%	2%
				30	98%	2%
Modèle 3 6 couches	6	3	3	10	97%	3%
				20	96%	4%
				30	98%	2%

Tableau III.9 Comparaison entre les différentes architectures et époques pour le cas binarisation et marquage sans binarisation et marquage.

Le tableau III.10 illustre la comparaison des précisions entre les différentes architectures et époques pour le cas (train2/3-test 1/3) :

	Nombre d'époque	Précision Sans marquage et binarisation	Précision avec marquage et binarisation	Précision Combinaisons entre marquage et binarisation et sans marquage et binarisation
		train2/3-test 1/3	train2/3-test 1/3	train2/3-test 1/3
2 couches	10	95%	97%	95%
	20	96%	98%	95%
	30	96%	98%	97%
4 couches	10	98%	98%	98%
	20	98%	99%	98%
	30	98%	99%	98%
6 couches	10	97%	98%	97%
	20	98%	97%	96%
	30	97%	98%	98%

Tableau III.10 Comparaison de différentes précisions dans le cas train 2/3 – test 1/3.

D'après nos résultats obtenus on remarque qu'après l'utilisation des attributs HMB, les résultats sont améliorés par rapport aux autres et le taux de précision atteindre 99%.

III.11 Résultats et discussions pour la combinaison de la base AMHCD (sans binarisation et marquage et avec binarisation et marquage) :

III.11.1 Résultats obtenus pour le modèle à 2 couches :

- Nombre d'époques = 10

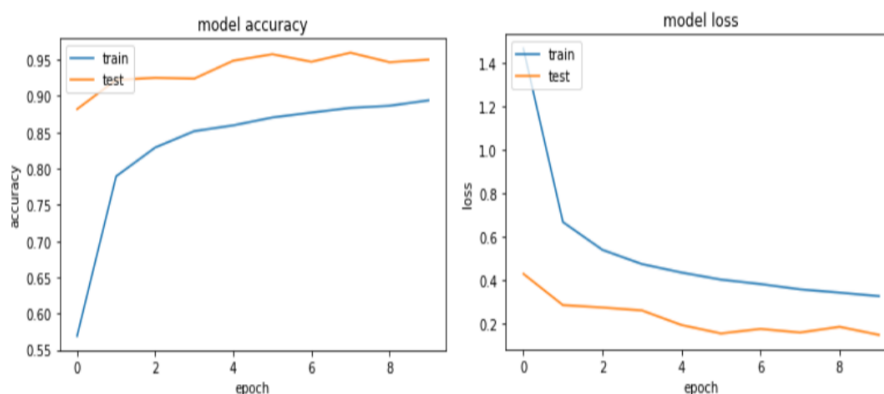


Figure III.9 Précision et erreur du modèle à 2 couches (10 époques).

D'après la figure III.9, la précision d'apprentissage (train) et celle du test augmente avec le nombre d'époque. Contrairement à l'erreur qui diminue avec l'augmentation du nombre d'époques. un taux de précision de 95% qui correspond à un total de 16302 caractères bien reconnus, et un taux d'erreur de 5% qui correspond à un total de 858 caractères mal reconnus.

- **Nombre d'époques = 20**

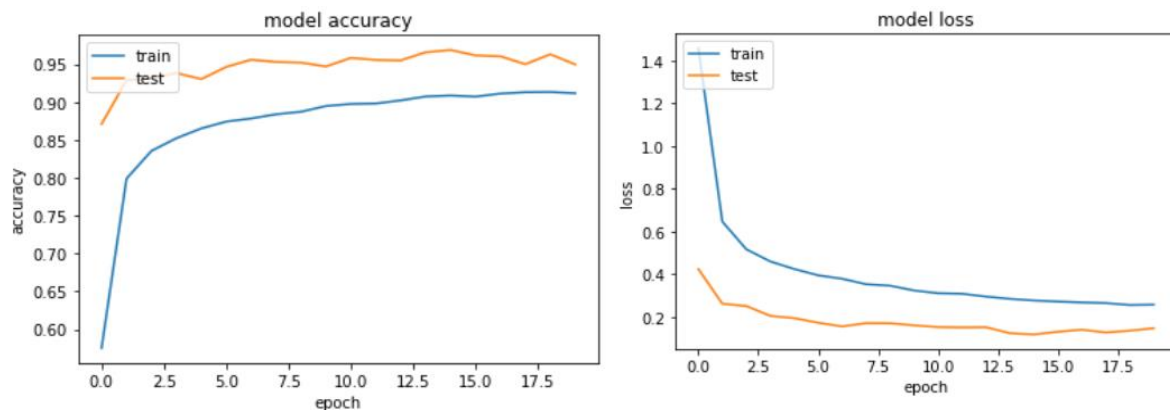


Figure III.10 Précision et erreur du modèle à 2 couches (20 époques).

D'après la figure III.10, on remarque que la précision de l'apprentissage et du test augmente avec le nombre d'époques dans le model accuracy. Ainsi, on voit que pour les deux époques 10 et 20 à reconnu les caractères avec une même précision 95%.

- **Nombre d'époques = 30**

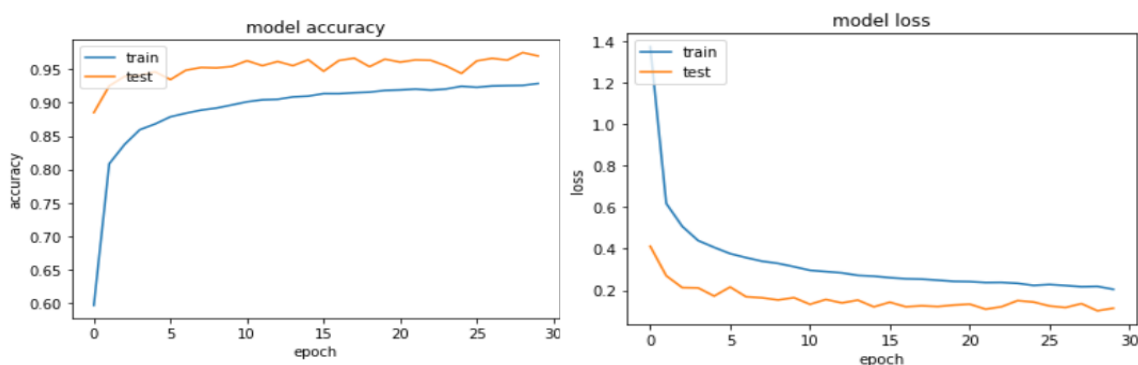


Figure III.11 Précision et erreur du modèle à 2 couches (30 époques).

D'après la figure III.11, on remarque que la précision de l'apprentissage et du test dans le model accuracy augmente avec le nombre d'époques. Ainsi, on voit que pour 30 époques le modèle a reconnu les caractères d'une manière plus précise par rapport à l'époque 10 et 20. Dans l'intervalle [10-30] époques, la précision d'apprentissage (train) ne cesse d'augmenter et la validation (test) se stabilise au niveau de 0.97. De même, l'erreur d'apprentissage et du test diminue avec le nombre d'époques et on remarque que les pertes ne sont pas conséquentes dès les premières époques pour le test jusqu'à stabilisation.

Un taux de précision de 97% qui correspond à un total de 16645 caractères bien reconnus, et un taux d'erreur de 3% qui correspond à un total de 515 caractères mal reconnus.

III.11.2 Résultats obtenus pour le modèle à 4 couches :

- Nombre d'époque 10,20 et 30

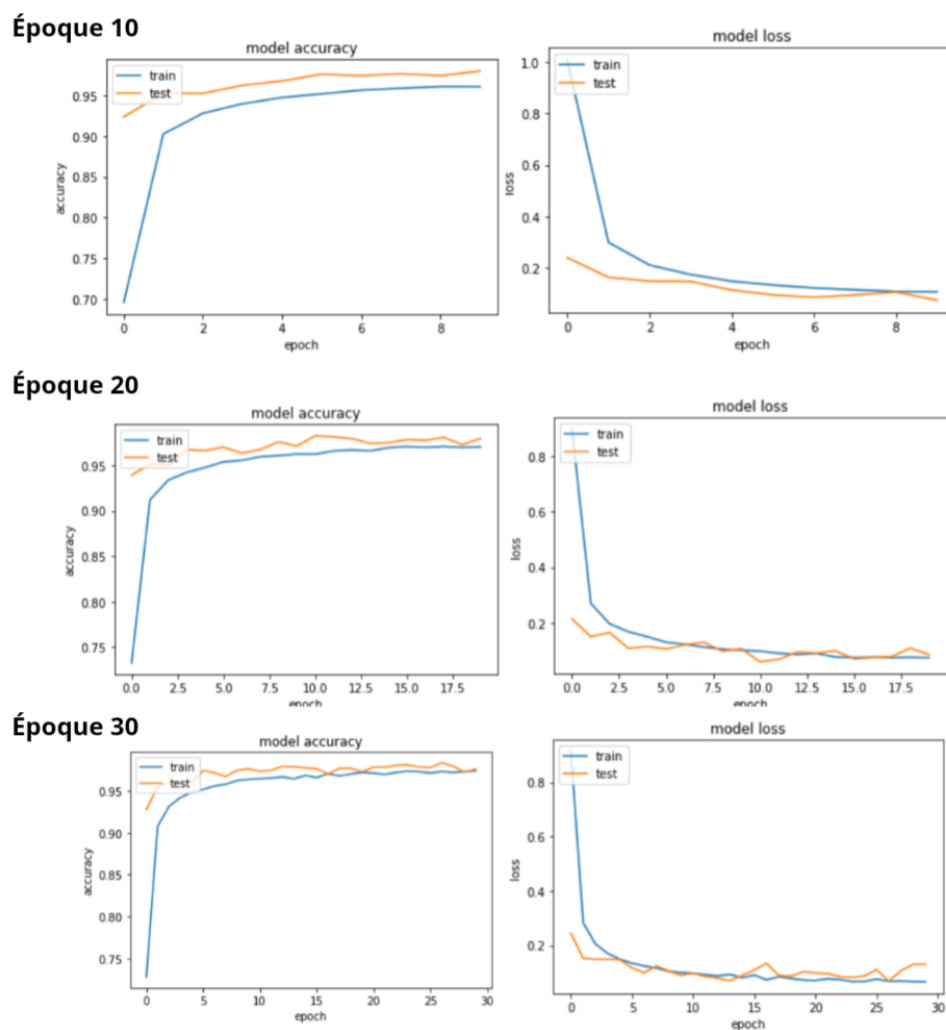


Figure III.12 Précision et erreur du modèle à 4 couches (10,20 et 30 époques).

Pour le réseau neuronal convolutif à 4 couches on voit que pour les trois époques 10, 20 et 30 à reconnu les caractères avec une même précision 98%. qui correspond à un total de 16817 caractères bien reconnus, et un taux d'erreur de 2% qui correspond à un total de 343 caractères mal reconnus.

III.11.3 Résultats obtenus pour le modèle à 6 couches

- Nombre d'époques = 10

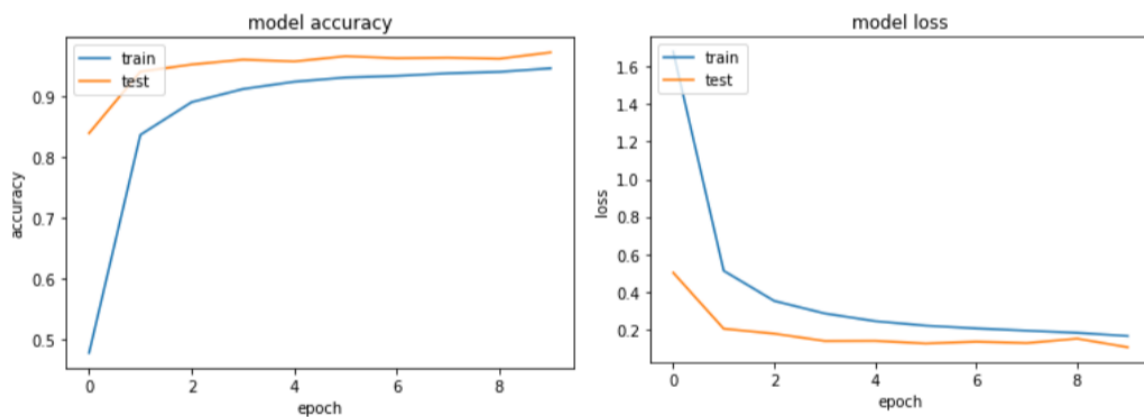


Figure III.13 Précision et erreur du modèle à 6 couches (10 époques).

D'après la figure III.13, la précision du train et celle de test augmente d'une manière successive durant les 10 époques, contrairement le modèle d'erreur il diminue consécutive.

Un taux de précision de 97% qui correspond à un total de 16645 caractères bien reconnus, et un taux d'erreur de 3% qui correspond à un total de 515 caractères mal reconnus.

- Nombre d'époques = 20

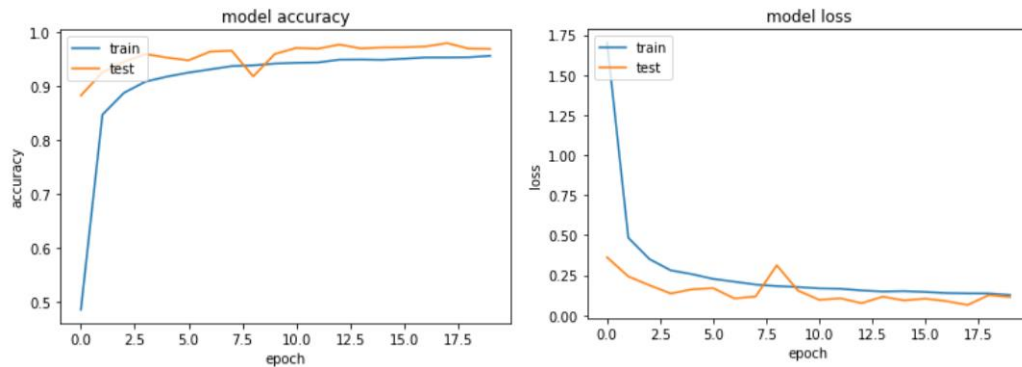


Figure III.14 Précision et erreur du modèle à 6 couches (20 époques).

D'après la figure III.14, on remarque que la précision du test sur l'intervalle [7.5-8.5] époques, elle diminue jusqu'à 0.90 ensuite elle s'augmente et elle se stabilise à 0.96 (96%). Pour le modèle d'erreur, on remarque que la précision du test sur l'intervalle [7.5-8.5] elle augmente jusqu'à 0.3.

Un taux de précision de 96% qui correspond à un total de 16474 caractères bien reconnus, et un taux d'erreur de 4% qui correspond à un total de 686 caractères mal reconnus.

- Nombre d'époques = 30

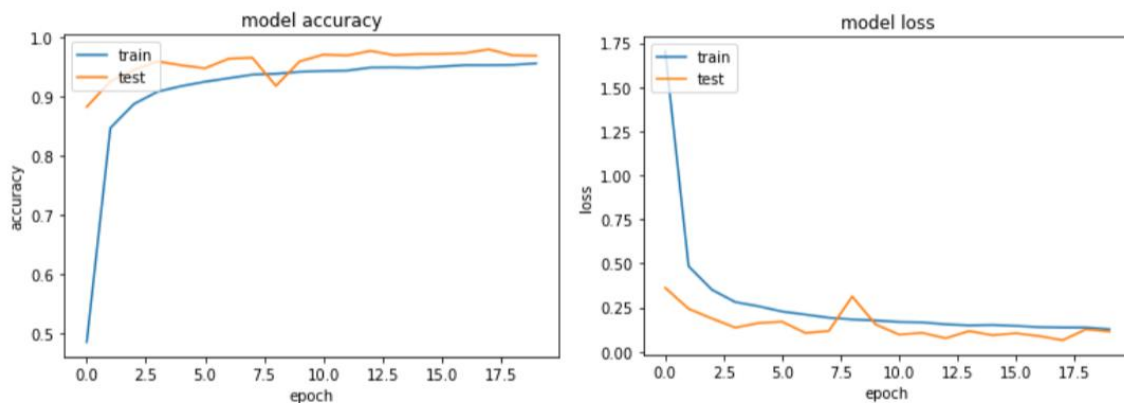


Figure III.15 Précision et erreur du modèle à 6 couches (30 époques).

D'après la figure III.15, la précision de train ne cesse d'augmenter jusqu'à la dernière époque, pareillement pour le test qui commence quant à lui avec une précision de 0.83,

augmente petit à petit jusqu'à stabilisation sur 0.98. En ce qui concerne l'erreur, la précision, du train elle diminue successivement, par contre le test elle diminue et elle s'augmente jusqu'à la stabilisation.

Un taux de précision de 98% qui correspond à un total de 16817 caractères bien reconnus, et un taux d'erreur de 2% qui correspond à un total de 343 caractères mal reconnus.

III.12 Conclusion :

Dans ce chapitre, nous avons élaboré une implémentation pour la reconnaissance des lettres amazigh (tfinagh) manuscrites en utilisant une approche de classification basée sur le réseau de neurone convolutionnel.

Les résultats obtenus sont très encourageant après l'utilisation de l'attribut HMB avec un taux d'exactitude de classification de 99% pour la reconnaissance des lettres Amazigh manuscrite « tfinagh » malgré la complexité de chaque caractère et on remarque que pour avoir un meilleur apprentissage on doit augmenter le nombre d'images dans le train comparons aux images insérer dans test qui aura un taux d'un 1/3 par rapport au totale. Le meilleur résultat obtenus c'est le cas de la binariation avec marque pour 1/3 pour le test et 2/3 pour le train dans le modèle de 4couches.

Conclusion générale

Au cours de ce mémoire, Nous avons cités les différentes techniques de reconnaissance de l'écriture manuscrite, nous avons pu observer que cette dernière a connu des progrès très intéressantes, durant ces dernières années , permettant de faire face à la variabilité de l'écriture, aussi de reconnaître de manière convenable des entités manuscrites.

Dans ce travail, nous nous sommes basée sur la reconnaissance des lettres tiffinagh par les réseaux de neurones convolutifs.

Nous avons parlé sur les notions fondamentales de Deep Learning, les réseaux de neurones en générale et les réseaux de neurones convolutionnels en particulier, on a opté à bien comprendre les différentes couches utilisées dans la classification : la couche convolutionnels, la couche de rectification, la couche de pooling et la couche fullyconnected. Nous avons également parlé du problème de sur apprentissage et de la manière de le résoudre au moyen des méthodes de régularisation (dropout). En phase d'implémentation, on a compté sur le langage reconnu dans le domaine de deep Learning (python), avec les différentes bibliothèques dédiées à ce type de recherche tel que (Tensorflow, Keras...), pour faciliter la tâche de création de nos modèles et pour l'accélération du training.

On a implémenté trois modèles de réseaux neurones convolutifs avec différentes architectures pour la base AMHCD, et pour éviter l'inconvénient principal de l'utilisation du l'apprentissage profond on a opté a la combinaison de l'attribut HMB et le CNN, puis on a appliqué pour chacun de ces modèles des tests qui consistent à changer le nombre d'époques jusqu'à l'obtention d'un taux de réussite important.

Enfin on a terminé avec des tableaux récapitulatifs et comparatifs des résultats obtenus qui s'avèrent être très satisfaisants et encourageants.

Bibliographie

- [1]- N. Benamara : « Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée ». Thèse de doctorat, spécialité Génie Electrique, Université des sciences, des Techniques et de médecine de Tunis II, 1999.
- [2] – Y.ES-SAADY : « Contribution au développement d'approches de reconnaissance automatique de caractères imprimés et manuscrits, de textes et de documents Amazighes. » ». Thèse de doctorat, spécialité mathématiques et informatique appliquées.
- [3]- J. Anigbogu : « Reconnaissance de textes imprimés mutifontes à l'aide de modèles stochastiques et métriques ». thèse de doctorat, Université de Nancy I, 1992.
- [4]- B. Al-Badr , S.A. Mahmoud : « Survey and bibliography of Arabic optical text recognition ». Signal processing , vol. 41, pp. 49-77, 1995.
- [5]- E.Lecolinet, O. Barrett : « Cursive word recognition : Methods and strategies ». In NATO/ASI , Fundamentals in handwriting recognition, Bonas, France June 21-july 3, 1993.
- [6]- R.G. Casey, E. Lecolinet : « Strategies in character segmentation : A survey ». IEEE. Proc. 3rd international conference on document Analysis and recognition (ICDAR'95), pp. 1028-1033, Montreal, Canada, 1995.
- [7]- B. Al-Badr , R.M. Haralick : « Symbol recognition without prior segmentation ». Conference SPIE-EI 1994.
- [8]- A.Belaid, «Reconnaissance automatique de l'écriture et du document» , Pour la science, disponible sur le lien web :<http://webloria.loria.fr/~abelaid/Publications.html>, 2001
- [9]- N. Ben Amara, S. Gazzah, «Une approche d'identification des fontes arabes» CIFED'04, pp.273-278, La Rochelle, France, Juin 2004.
- [10]- S. Chevalier, M. Lemaître, E. Geoffrois, «Étude de primitives spectrales pour la reconnaissance de caractères manuscrits dans le cadre d'une approche markovienne 2D» , Actes 15ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle (RFIA'2006), Tours, France (CDROM), 2006.
- [11]- H. Miled, «Stratégies de reconnaissance de l'écriture semi cursive : application aux mots manuscrits arabes», Thèse de doctorat, Université de Rouen, 1998.
- [12]- F. Grandidier, R. Sabourin, C.Y. Suen, «Quelques techniques pour l'amélioration du pouvoir discriminant de primitives discrètes», Conférence Internationale Francophone sur l'Ecrit et le Document CIFED'04, 2004.
- [13]- R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Second edition. New York: Wiley Interscience, 2001.

Bibliographie

- [14]- C. Saint-Jean, «Classification paramétrique robuste partiellement supervisée en reconnaissance des formes », Thèse de doctorat, Université de la Rochelle, 2001
- [28] -M.Khoscheed : «off-line arabic character recognition areview»,patter analysis application,vol 5,(2002),pp.31-45 .
- [15] - <https://atcold.github.io/pytorch-Deep-Learning/fr/week01/01-1/>
- [16]- Thierry Deneux Université de Technologie de Compiègne Département Génie Informatique Heudiasyc ,14 septembre 2018
- [17]- <https://blog.varonis.fr/classification-des-donnees-kesako/>
- [18]- Classification avec les réseaux de neurones. Réalisé par : HADJ SAID Nadine, REKOUANE Hichem 2019-2020
- [19]- McCulloch, W. S., Pitts, W. A Logical Calculus of the Ideas Immanent in Nervous Activity Bulletin of Mathematical Biophysics. 1943.
- [20]-<https://www.juripredis.com/fr/blog/id-19-demystifier-le-machine-learning-partie-2-les-reseaux-de-neurones-artificiels>.
- [21]- <https://www.racinely.com/post/les-fonctions-d-activation-part>
- [22]- D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT Press, 1986.
- [23]- MALKI NAR IMENE, Classification automatique des textes par Les réseaux de neurones à convolution, thèse MASTER ACADEMIQUE, Domaine : Mathématique et Informatique, Filière : Informatique, Spécialité : Vision Artificielle, Faculté sciences exactes et sciences de la nature et de la vie, Département de mathématiques et d'informatique, Université LARBI ben M'HIDI O.E.B, 2018/2019.
- [24]- Houacine Noura, Khelifa Nadia, Classification des textures par les réseaux de neurones convolutif, thèse MASTER ACADEMIQUE, Domaine : Sciences et Technologies, Filière : Génie électrique, Spécialité : Commande des systèmes, FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE DEPARTEMENT D'AUTOMATIQUE, UNIVERSITE MOULOUDE MAMMERI DE TIZI-OUZOU, 27/09/2018.
- [25]- https://penseeartificielle.fr/focus-reseau-neuronesconvolutifs/#4_Le_flattening_ou_mise_a_plat
- [26] -D.P. Kingma and J.L. Ba. ADAM: A method for stochastic optimization. ICLR 2015. juillet 2017.

Bibliographie

- [27]-Moualek Djaloul Youcef, «Deep Learning pour la classification des images», Master, Université Abou Bakr Belkaid Tlemcen Faculté des Sciences Département d'informatique, juillet 2017.
- [28]-<https://deepai.org/machine-learning-glossary-and-terms/tensorflow>.
- [29]- <https://deepai.org/machine-learning-glossary-and-terms/keras>.
- [30]- <https://www.kaggle.com/datasets/benaddym/amazigh-handwritten-character-database-amhcd>
- [31]- GAGAOUA, Meriem, GHILAS, Hamza, TARI, Abdelkamel, *et al.* Histogram of marked background (HMB) feature extraction method for Arabic handwriting recognition. *International Journal of Image and Graphics*, 2022, vol. 22, no 02, p. 2250015.
- [32]- Lamtougui, H., et al. Offline Arabic Handwriting Recognition Using Deep Learning: Comparative Study. in 2020 International Conference on Intelligent Systems and Computer Vision (ISCV). 2020. IEEE.
- [33]-Korichi, A., et al. Arabic handwriting recognition: Between handcrafted methods and deep learning techniques. in 2020 21st International Arab Conference on Information Technology (ACIT). 2020. IEEE.
- [34] - Altwaijry, N. and I. Al-Turaiki, Arabic handwriting recognition system using convolutional neural network. *Neural Computing and Applications*, 2020: p. 1-13.
- [35] -M. Ameer, A. Bouhjar, F. Boukhris, A. Boukouss, A. Boumalk, M.Elmedlaoui, E. Iazzi, H. Souifi, "Initiation à la langue Amazighe", Publications de l'IRCAM, CAL, Rabat, 2004.

Résumé :

Dans ce travail, on a utilisée la combinaison de deep learning et l'attribut HMB (Histogram of marked background) avec ses réseaux de neurones plus précisément le réseau de neurone convolutif (CNN) pour la reconnaissance des lettres berbères afin de construire une nouvelle représentation d'un jeu de données en utilisant deux types de fichier principale le training et le test aussi pour apprendre à la machine de faire l'apprentissage des lettres manuscrites berbère de type « tafinagh » avec plusieurs cas d'apprentissage.

Nous avons obtenu de bons résultats, et nous avons apporté plusieurs améliorations en fusionnent deep learning et l'attribut HMB pour obtenir les meilleurs résultats et un excellent pourcentage possible de la reconnaissance souhaitée.

Les résultats obtenus ont montré que le choix du nombre d'époque, la taille de la base de données ainsi que la profondeur du réseau ont une grande influence pour avoir les résultats meilleurs.

Mots clés: Deep Learning, réseaux de neurones convoltonnel (CNN), lettres berbère, L'apprentissage profond, Tafinagh.

Abstract:

In this work, we used the combination of deep learning and the attribute Histogram of marked background with its neural networks, more precisely the convolutional neural network (CNN) for the recognition of Berber letters in order to build a new representation of a data set using two types of main file the training and the test also to teach the machine to learn Berber handwritten letters of the "tafinagh" type with several learning cases.

We have obtained good results, and we have made several improvements by merging deep learning and the attribute to obtain the best results and an excellent possible percentage of the desired recognition.

The results obtained showed that the choice of the number of epochs, the size of the database as well as the depth of the network have a great influence to have the best results.

Keyword: Deep Learning, convoltonal neural networks (CNN), Berber letters, Tafinagh.