

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLAB BLIDA



Département d'informatique

MEMOIRE DE FIN D'ETUDE

*En Vue De L'obtention Du Diplôme master académique en  
Sciences et technologie*

*Spécialité : Informatique*

*Option : Traitement  
automatique des langues*

**THEME**

La reconnaissance des caractères arabes  
manuscrits par les réseaux des neurones  
convolutionnels

*Présenté et soutenu publiquement par :*

Daoud Fouad et Louali Farouk

*Le :*

**Membres du jury:**

**President : Mme. BOUSTIA NARHIMENE**

**Examineurs : Mme. BACHA SIHAM**

**Encadreur : Mme ALIANE HASSINA**

**Année universitaire : 2018/2019**

---

## Remerciements

Tout d'abord, nous tenons à remercier le bon Dieu le tout Puissant de nous avoir donné la force et le courage de mener à bien ce modeste travail, également nous remercions infiniment nos parents, qui nous ont encouragés et aidés à arriver à ce stade de notre formation. Nous tenons à remercier tous ceux et celles qui ont contribué à finaliser ce modeste travail. Nos remerciements vont au professeur Mme ALIANE HASSINA notre encadreur de nous avoir guidés pour la réalisation de ce projet et pour tout ce qu'on a appris et soutenu lors de notre stage au niveau du centre de recherche CERIST.

Nous remercions vivement notre promoteur MR CHERIF ZAHAR pour ces enseignements et ces conseils décisifs dans l'élaboration de notre modeste travail et aussi pour son aide morale durant toute la période de stage.

Enfin, nous tenons à remercier tous ceux qui nous ont aidé et assisté durant notre cursus et nous exprimons toute notre gratitude à Mme BENBLIDIA, Mr KAMECHE et tout le département informatique de l'université de Blida SAAD DAHLAB.

---

# Table des matières

<b>Chapitre 1. La reconnaissance des caractères arabes .....</b>	<b>12</b>
1.1 Introduction : .....	13
1.2 Historique : .....	14
1.3 Reconnaissance en ligne et hors ligne : .....	15
1.3.1 Systèmes en ligne : .....	15
1.3.2 Systèmes hors ligne : .....	16
1.3.3 Comparaison de la reconnaissance des caractères en ligne et hors ligne : .....	17
1.4 Caractéristiques morphologique du texte : .....	17
1.5 Contraintes spécifiques au texte arabe: .....	19
1.6 Architecture d'un système de reconnaissance des caractères arabes : .....	23
1.6.1 Prétraitement : .....	26
1.6.1.1 Binarisation : .....	26
1.6.1.2 Suppression de bruit : .....	26
a. Filtrage : .....	26
b. Opérations morphologiques mathématiques : .....	27
1.6.1.3 Détection de la ligne de base : .....	27
1.6.1.4 Seuillage : .....	28
1.6.1.5 Squelettisation : .....	28
1.6.1.6 Normalisation : .....	30
1.6.2 Segmentation : .....	31
1.6.2.1 Types de segmentation : .....	31
a. Segmentation d'une page en lignes : .....	31
b. Segmentation d'une ligne en mots : .....	32
c. Segmentation d'un mot en caractères : .....	32
1.6.2.2 Différentes techniques appliquées pour la segmentation : .....	32
A. Technique de segmentation basée sur la projection verticale : .....	32
B. Technique de segmentation basée sur l'éclaircissement : .....	33
C. Technique de segmentation basée sur le traçage des contours : .....	33
D. Technique de Segmentation basée sur les réseaux neuronaux artificiels : .....	33
E. Technique de segmentation basée sur des opérateurs morphologiques : .....	33
1.6.3 Extraction des caractéristiques : .....	34
1.6.3.1 Caractéristiques structurelles : .....	34

1.6.3.2	Caractéristiques statistiques :	35
1.6.3.3	Transformation globale :	35
1.6.4	Classification :	35
1.7	Conclusion :	36
<b>Chapitre 2. Les réseaux des neurones convolutionnels.....</b>		<b>37</b>
2.1	Introduction :	38
2.2	Algorithmes de Deep Learning :	39
2.3	Neurone biologique :	39
2.4	Réseaux de neurones artificiels :	40
2.5	Perceptron :	41
2.6	Techniques utilisées par les réseaux de neurones :	41
2.6.1	Apprentissage supervisé :	42
2.6.2	Classification :	42
2.6.3	Régression :	42
2.6.4	Reconnaissance de formes :	42
2.6.5	Apprentissage non supervisé :	43
2.6.6	Clustering :	43
2.7	Fonctions d'activation :	43
2.7.1	Sigmoïde :	44
2.7.2	Tangente hyperbolique :	44
2.7.3	Unité de rectification linéaire (ReLU) :	45
2.7.4	Softmax :	45
2.8	Perceptron multicouches :	46
2.8.1	Le processus d'apprentissage :	47
2.8.2	Algorithme du gradient:	49
2.9	Réseaux des neurones convolutionnels (RNC) :	52
2.9.1	Architecture de réseaux de neurone convolutionnel:	53
2.9.1.1	la convolution :	53
2.9.1.2	Couche de mise en commun ( pooling ) :	54
2.9.1.3	la couche ReLU :	55
2.9.1.4	Couche fully-connected :	55
2.10	Conclusion :	56
<b>Chapitre 3. Conception de l'architecture CNN pour la reconnaissance des caractères arabes</b>		<b>58</b>
3.1	Introduction :	59
3.2	Conception de l'architecture CNN:	59

3.3	Logiciels et bibliothèques Utilisés dans l'implémentation :.....	59
3.3.1	Dataset : .....	59
3.3.2	TensorFlow : .....	61
3.3.3	Keras : .....	62
3.3.4	Module keras utiliser : .....	63
3.3.5	Python : .....	63
3.4	Configuration Utilisé dans l'implémentation :.....	64
3.5	Architecture du réseau :.....	64
3.6	Résultat : .....	69
3.7	Conclusion :.....	70

## Liste des figures

FIGURE 1-0 1: Exemple de reconnaissance en ligne.....	16
FIGURE 1-0 2: LES DIFFERENTS TYPES D'ECRITURES.....	18
FIGURE 1-0 3: Exemple du texte contraint et non contraint. ....	19
FIGURE 1-0 4: Ligne de base, ascendeurs et descendeurs comme indiqué dans un mot. ....	20
FIGURE 1-0 5: Mots avec différents sous-mots. ....	21
FIGURE 1-0 6: Des styles et des tailles différents pour le même mot.....	21
FIGURE 1-0 7: Différents ligatures des mots. ....	22
FIGURE 1-0 8: Similitude entre certains caractères. ....	23
FIGURE 1-0 9: Classification des systèmes de reconnaissance des caractères. ....	24
La figure 1-0 10 indique différentes voies pour la reconnaissance des caractères arabes avec ou sans segmentation : .....	25
FIGURE 1-0 10: Étapes générales pour la reconnaissance du texte manuscrit.....	25
FIGURE 1-1 1: Résultat d'une opération d'éclaircie.....	29
FIGURE 2-0 1: La relation entre l'intelligence artificielle, le ML et le deep learning.....	38
FIGURE 2-0 2: Modèle d'un neurone biologique.....	40
FIGURE 2-0 3: Modèle d'un neurone artificiel.....	41
FIGURE 2-0 4: Placement de la fonction d'activation dans le modèle de réseau neuronal. ....	44
FIGURE 2-0 5: fonction d'activation .....	46
FIGURE 2-0 6: Schéma d'un PMC.....	47
FIGURE 2-0 7: L'évolution de l'erreur d'apprentissage par rapport à l'erreur de validation. ....	49
FIGURE 2-0 8: Architecture standard d'un réseau de neurone convolutionnel.....	52
FIGURE 2-0 9: Schéma du parcours de la fenêtre de filtre sur l'image. ....	53
FIGURE 2-0 10 : Exemple d'une convolution dont la configuration est : Opération = Argument maximale, pas horizontale = 1 pixel, pas vertical = 1 pixel. ....	54
FIGURE 3-0 1: Exemple de dataset des caractères arabes.....	60
FIGURE 3-0 2: Architecture de notre RNC .....	65
FIGURE 3-0 3: Configuration de notre modèle rnc.....	65
FIGURE 3-0 4: Taux de Précision Du Modèle.....	69
FIGURE 3-0 5: Taux d'Erreur du Modèle.....	70

## Liste des tableaux

TABLEAU 1-0 1:FORMES DES CARACTERES ARABES.....	20
---	----

## Liste des abréviations

OCR : Reconnaissance optique des caractères.

RNA : Le réseau de neurone artificiel.

RNC : Le réseau de neurones convolutionnel.

CNN : Convolutional neural network.

GED : Gestion électronique des documents.

2D : Deux dimensions.

3D : trois dimensions.

PAW : Pieces of Arabic Words.

## Introduction générale

---

Les innovations technologiques, le fonctionnement digital des services, contribuent à modifier les habitudes et pratiques des acteurs dans tous les domaines.

Face à l'introduction de nouvelles techniques informatique et de traitement automatique des données, l'être humain voit son quotidien modifié et est obligé d'intégrer ces outils pertinents dans ses faits et gestes d'une manière optimisée.

La digitalisation constitue un événement d'actualité qui se développe avec une dynamique soutenue, c'est le processus moderne de management des entreprises, cette digitalisation s'appuie entre autre sur la dématérialisation des documents (GED) pour constituer l'entrepôt de donnée sur lequel se base les stratégies de développement des entreprises.

La dématérialisation de document est tributaire des techniques de reconnaissance des caractères des langues.

La langue arabe est considérée comme la 8eme langue la plus parlée au monde avec plus 400 millions de locuteurs (internet).

Cependant, nous constatons que les évolutions technologiques n'ont pas fait avancer la reconnaissance des caractères arabes ce qui a retardé la dématérialisation des documents contenant des textes exprimés en caractères arabes.

Nous savons que pour les caractères latins la problématique ne se pose pas , La comparaison des taux de reconnaissances des caractères Latins et Arabes est exprimée comme suit (internet) : Taux de reconnaissance moyen des caractères Latins : 90%

Taux de reconnaissance moyen des caractères Arabes :  $\pm 50\%$

Les analystes et experts en charge de traitement de la reconnaissance des caractères mettent en exergue la difficulté du traitement des caractères arabes compte tenu de la spécificité de ces caractères qui se distinguent par le rang ou l'emplacement du caractère dans un mot :

Le même caractère arabe s'écrit de manière différente : au début, au milieu et à la fin du mot.

Notons que la reconnaissance automatique des caractères arabes n'a pas bénéficié de recherches soutenues, mais quelques tentatives d'études ont été menées sans pour autant résoudre la problématique de la reconnaissance des caractères arabes d'une manière totale, et efficace.

Il est important de rappeler que le caractère arabe est cursif de par sa nature, ce qui rend sa reconnaissance automatique laborieuse au stade actuel des évolutions technologiques.

Ce qui rend encore la tâche ardue c'est la reconnaissance manuscrite objet de notre étude.

En effet un caractère arabe manuscrit prend plusieurs formes selon plusieurs utilisateurs.

La conception de système de reconnaissance des caractères manuscrits se base sur la segmentation des mots induisant ainsi un temps de traitement et de recherche conséquents.

Pour pallier à ces lacunes, des approches de réseau neuronal ont été proposées, ce processus se décline par la considération d'un nombre important de variabilité des caractères, ce qui a amené les chercheurs dans ce domaine à se pencher sur le réseau de neurones convolutionnels (convolutifs) pour extraire, classifier les caractéristiques des caractères arabes, et par conséquent permettre la reconnaissance automatique de l'écriture manuscrite.

Dans notre modeste mais ambitieuse étude et recherches assez instructives, nous avons conçu notre mémoire sur trois (3) grandes étapes (3 chapitres) à l'effet de :

- 1 --- s'informer, et se documenter sur les approches et expériences conduites par les experts dans le domaine de reconnaissance manuscrite des caractères arabes et d'aborder les jalons qui composent une chaîne de reconnaissance de l'écriture manuscrite, arabe.
- 2 --- d'étudier les possibilités offertes par les réseaux des neurones artificiels en axant l'explication sur la structuration d'un neurone, élément fondamental du réseau et sur les algorithmes des réseaux multicouches et réseaux convolutionnels pour la classification des caractéristiques.
- 3 --- Cette phase d'étude (chapitre 3) est consacré à la conception d'un algorithme et l'applicatif développé pour l'utilisation du réseau de neurones convolutionnels de

reconnaissance des caractères arabes manuscrits, nous aborderons les tests et simulations opérées et les résultats obtenus à chaque simulation.

Nous concluons cette étude en évoquant les futures mutations technologiques et le monde de digitalisation qui se développe, qui lèveront toute contrainte ou ambiguïté sur la problématique rencontrée de nos jours sur l'écriture arabe manuscrite .

## **Chapitre 1.**

# **La reconnaissance des caractères arabes**

## 1.1 Introduction :

Au cours des dernières décennies, la reconnaissance des caractères alphanumériques manuscrits en arabe est devenue un des domaines de recherche difficiles dans le domaine du traitement des images de documents. Alors que la reconnaissance du latin écrit à la main a fait l'objet de nombreuses recherches faisant appel à diverses techniques, peu de travaux ont été entrepris sur la reconnaissance arabe manuscrite, et aucune des techniques existantes n'est assez précise pour une application pratique

Reconnaître un caractère ou un texte arabe est une tâche compliquée en raison de la variation illimitée de l'écriture humaine, la grande variété de formes de caractères arabes, la présence de la ligature entre les caractères et le chevauchement des composants , La principale distinction entre l'arabe et les autres langues romaines est que les mots et les caractères arabes contenus dans les mots sont écrits à partir de de droite à gauche, par opposition aux mots anglais qui sont écrits de gauche a droite.

En outre, l'alphabet arabe est largement utilisé par de nombreuses personnes de différents pays, les pays arabes en tête, en plus d'être utilisés pour l'écriture du persan, de l' ourdou et du pachtou .Ce serait génial d'utiliser la reconnaissance de caractères manuscrite en arabe (AHCR) pour convertir de nombreux documents au format numérique format accessible électroniquement.

Les applications comprennent: la lecture d'adresses postales sur des enveloppes ,le tri automatiquement du courrier, l'aide aux non voyants, la lecture des formulaires remplis par les clients, les réclamations d'assurance, les formulaires de demande ,l'automatisation des bureaux, l'archivage et la récupération de texte .Ainsi on améliorera les interfaces homme-machine.

## 1.2 Historique :

Dans le domaine de la reconnaissance automatique de l'écriture manuscrite, les bases de données d'images annotées ne sont pas nombreuses.

Le domaine de la reconnaissance automatique de l'écriture arabe manuscrite a souffert pendant de nombreuses années de l'absence d'une base de données de référence, qui permet des comparaisons objectives entre les différents systèmes.

Des bases importantes mais payantes comme la base Kharma/Ahmed/Ward ou la base de chèques du CENPARMI sont apparues. Mais c'est la base IFN/ENIT, gratuite pour la recherche académique, qui s'est imposée comme la base de données de référence pour la comparaison des performances des systèmes de reconnaissance de l'écriture arabe manuscrite.

En effet en 1999, N. Kharma et al proposent une base de données, obtenue à partir des contributions de 500 étudiants : – 37000 mots arabes – 10000 chiffres.

Les images sont disponibles à la fois en niveaux de gris et sous forme binarisée.

En 2002 :

Almaadeed et al. présentent une base de données collectées à l'aide d'une centaine de scripteurs .(AHDB)

M. Pechwitz et al introduisent la base IFN/ENIT. Il s'agit d'une base de données d'images de noms de villes tunisiennes. Elle sert de support à l'organisation des compétitions internationales dans le domaine de la reconnaissance de l'écriture arabe manuscrite : ICDAR Arabic Handwritten Competition ( en 2005 et 2007)

En 2003, Y. Al-Ohali et al ont achevé la réalisation d'une base de données pour la reconnaissance de chèques arabes manuscrits. Cette base se compose de 7000 images de chèques saoudiens issus de la pratique bancaire, scannés en niveaux de gris avec une résolution de 300 dpi. (CENPARMI) .

.En 2006 Gregory R. Ball Sargur N. Srihari et Harish Srinivasan. Ont présenté La base de données CEDARABIC . Elle comprend 10 scripteurs différents. Chacun d'entre eux a écrit 10 pages complètes différentes de texte manuscrit. Chaque page comprend entre 150 et 200 mots, soit un total d'environ 20000 mots. Les documents sont scannés en 300 dpi.

En février 2007 la base de données HODA farsi digits a été rendue publique par H. Khosravi et E. Kabir. Il s'agit d'une base conséquente de chiffres farsi isolés, d'une taille comparable à la base MNIST. Elle est composée de 102352 chiffres extraits à partir d'environ 12000 formulaires, remplis ensemble d'apprentissage (60000 exemples), un ensemble de test (20000 exemples), plus un troisième ensemble qui contient des données restantes (22352 exemples).

En septembre 2007 la base de données Applied Media Analysis : Arabic-Handwriting- a été rendue publique. Ses principales caractéristiques sont : – 5000 pages manuscrites – plusieurs types de documents : diagrammes, mémos, formulaires, listes (comprenant des chiffres indiens et arabes), poèmes.

## **1.3 Reconnaissance en ligne et hors ligne :**

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

### **1.3.1 Systèmes en ligne :**

Ce mode de reconnaissance s'opère en temps réel (pendant l'écriture). Les symboles sont reconnus au fur et à mesure qu'ils sont écrits à la main. Ce mode est réservé généralement à l'écriture manuscrite. C'est une approche où la reconnaissance est effectuée sur des données à une dimension. L'écriture prend l'aspect d'un couple de signaux temporels numérisés [1]. La reconnaissance en ligne présente un avantage majeur c'est la possibilité de correction et de modification de l'écriture de manière interactive vu la réponse en continu du système[ 2]. L'acquisition de l'écrit est généralement assurée par une tablette graphique munie d'un stylo électronique.



**FIGURE 1-0 1: EXEMPLE DE RECONNAISSANCE EN LIGNE.**

Aujourd'hui d'autres applications se sont développées à partir de ces assistants personnels liés aux domaines de la médecine, management, marketing, l'éducation, l'industrie et la gestion.

### **1.3.2 Systèmes hors ligne :**

Démarre après l'acquisition. Elle convient aux documents imprimés et les manuscrits déjà rédigés. Ce mode peut être considéré comme le cas le plus général de la reconnaissance de l'écriture. Il se rapproche du mode de la reconnaissance visuelle. L'interprétation de l'information est indépendante de la source de génération [3].

La reconnaissance hors-ligne peut être classée en plusieurs types :

Reconnaissance de texte ou analyse de documents :

Dans le premier cas il s'agit de reconnaître un texte de structure limitée à quelques lignes ou mots. La recherche consiste en un simple repérage des mots dans les lignes, puis à un découpage de chaque mot en caractères. Dans le second cas (analyse de document), il s'agit de données bien structurées dont la lecture nécessite la connaissance de la typographie et de la mise en page du document. Ici la démarche n'est plus un simple prétraitement, mais une démarche experte

d'analyse de document il y'a localisation des régions, séparation des régions graphiques et photographique, étiquetage sémantique des zones textuelles à partir de modèles, détermination de l'ordre de lecture et de la structure du document [4].

Reconnaissance de l'imprimé ou du manuscrit :

Les approches diffèrent selon qu'il s'agisse de reconnaissance de caractères imprimés ou manuscrits. Les caractères imprimés sont dans le cas général alignés horizontalement et séparés verticalement, ce qui simplifie la phase de lecture [5]. La forme des caractères est définie par un style calligraphique (fonte) qui constitue un modèle pour l'identification. Dans le cas du manuscrit, les caractères sont souvent ligaturés et leur graphisme est inégalement proportionné provenant de la variabilité intra et inter scripteurs. Cela nécessite généralement l'emploi de techniques de délimitation spécifiques et souvent des connaissances contextuelles pour guider la lecture [4].

### **1.3.3 Comparaison de la reconnaissance des caractères en ligne et hors ligne :**

#### **Caractères en ligne :**

- Utilisation de stylos numériques
- Exigence d'échantillons
- Disponibilité de coup de stylo
- Taux de reconnaissance élevé
- Grande précision

#### **Caractères hors ligne :**

- Utilisation de papier
- Exigence de points
- Non disponible
- Taux de reconnaissance bas
- Faible précision

### **1.4 Caractéristiques morphologique du texte :**

Le texte manuscrit arabe dispose de plusieurs formes contraint ou non contraint avec des caractères continus ou isolés, cursif ou non cursif.

Le caractère isolé pose moins de contraintes et sa reconnaissance passe par la classification

des caractères sans avoir besoin de l'étape de segmentation .

Le texte continu par contre passe par une étape de segmentation de l'image d'entrée avant la phase de la classification.

La reconnaissance par le système d'un texte imprimé est toujours considérée comme une forme isolée.

Un cas particulier du texte continu est la reconnaissance de l'écriture cursive. La segmentation dans ce cas est beaucoup plus difficile. L'écriture non cursive est également reconnue en tant que caractères isolés [6].



**FIGURE 1-0 2: LES DIFFERENTS TYPES D'ECRITURES.**

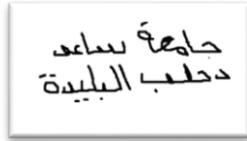
La contrainte quand elle caractérise un texte manuscrit ou imprimé engendre des situations où l'utilisateur est orienté dans son écriture pour le remplissage des emplacements prédéfinis .

Le système dispose donc d'informations précises quand à ses priorités ,les reconnaissances qu'il doit opérer et l'emplacement du dit texte.

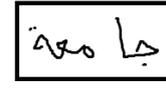
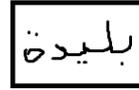
Dans l'autre cas c'est-à-dire du texte non contraint, l'utilisateur dispose de toutes latitudes à l'effet de renseigner la page comme bon lui semble sans contrainte aucune.

Un texte constitué de plus d'une ligne sans contrainte d'entrée ni de nombre de mots dans chaque ligne est considéré dans la pratique comme un texte non contraint.

Des exemples des textes contraints et non contraint sont illustrés dans la figure suivante :



Texte non contraint



Texte contraint

**FIGURE 1-0 3: EXEMPLE DU TEXTE CONTRAINT ET NON CONTRAINT.**

Par ailleurs, chaque langue possède ses propres caractéristiques tels que le nombre de lettres et la façon de constituer sémantiquement et graphiquement un mot.

On distingue quatre catégories de reconnaissance de l'écriture manuscrite selon que l'unité de reconnaissance est un caractère, un mot, une phrase, ou encore une partie d'un texte, comme nous allons montrer par la suite.

## **1.5 Contraintes spécifiques au texte arabe:**

L'arabe s'écrit horizontalement de droite à gauche et l'alphabet est composée de 28 caractères. C'est un alphabet consonantique, aussi appelé abjad, dont les graphèmes (unités de base) sont des consonnes. Les voyelles dans un alphabet consonantique sont *implicitement* dictées par la phonologie : le lecteur doit connaître la langue pour lire toutes les voyelles.

Chaque caractère a deux ou quatre formes, la forme du caractère dépendant de sa position dans le mot. Les points jouent aussi un rôle important.

Il y a similitude des formes de certains caractères. La différence vient de la position et du nombre de points, tels que (ب, ت, ث), qui peuvent apparaître au-dessus ou au-dessous des caractères. Dix caractères de l'alphabet ont un points, trois en ont deux et deux ont un.

Les points peuvent apparaître sous forme de deux points distincts ou peuvent être reliés en une ligne dans un texte manuscrit. En outre, des marques courtes telles qu'un " " hamza " " peuvent être placées au-dessus ou au-dessous de cinq caractères particuliers ou peuvent apparaître sous forme de caractères isolés. Certains caractères arabes ont une boucle, telle que (ص, ف, و). La nature cursive du texte arabe signifie que les caractères d'un mot sont reliés par une ligne horizontale imaginaire appelée ligne de base. De plus, il existe des lignes qui apparaissent au-dessus et en dessous de la ligne de base, appelées ascendeurs et descendeurs, comme illustré à la figure 1-04 [7].

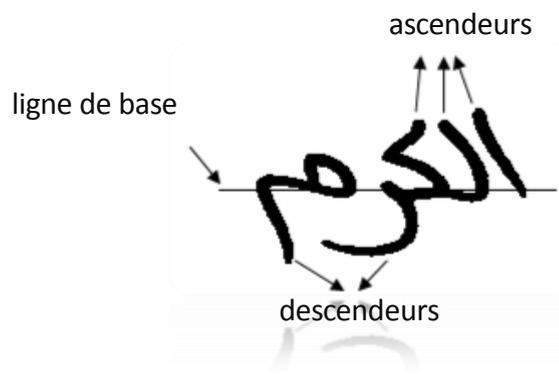
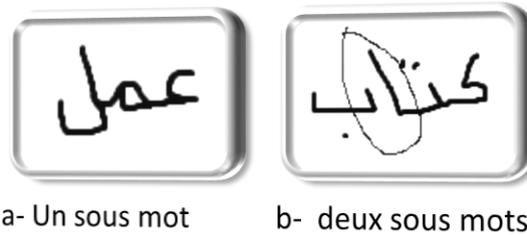


FIGURE 1-0 4: LIGNE DE BASE, ASCENDEURS ET DESCENDEURS COMME INDIQUE DANS UN MOT.

Lettre	Isolé	For.F/M/I	Lettre	Isolé	For.F/M/I
Alif	ا	ا ا ا	Dhad	ض	ض ض ض
Ba	ب	ب ب ب	Tad	ط	ط ط ط
Ta	ت	ت ت ت	Thad	ظ	ظ ظ ظ
Tha	ث	ث ث ث	Ayn	ع	ع ع ع
Jim	ج	ج ج ج	Gyn	غ	غ غ غ
Ha	ح	ح ح ح	Fa	ف	ف ف ف
Kha	خ	خ خ خ	Qaf	ق	ق ق ق
Dal	د	د د د	Kaf	ك	ك ك ك
Dhal	ذ	ذ ذ ذ	Lam	ل	ل ل ل
Ra	ر	ر ر ر	Mim	م	م م م
Zay	ز	ز ز ز	Nun	ن	ن ن ن
Sin	س	س س س	Ha	ه	ه ه ه
Shin	ش	ش ش ش	Waw	و	و و و
Sad	ص	ص ص ص	Ya	ي	ي ي ي

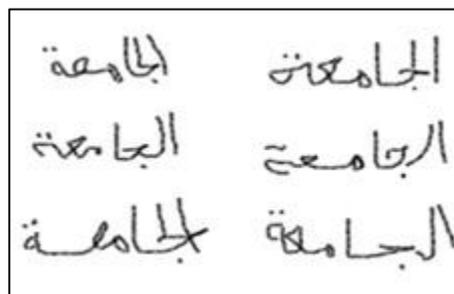
TABLEAU 1-0 1:FORMES DES CARACTERES ARABES.

Les caractères suivants (ا, د, ذ, ر, ز, و) ne se connectent pas à un caractère suivant dans un mot, ce qui entraîne une séparation du mot en plusieurs parties. Ces parties sont appelées sous-mots. Les espaces séparent les mots et les espaces courts séparent les sous-mots. Il est difficile de segmenter un mot arabe manuscrit en caractères, en raison de la nature cursive du texte arabe.



**FIGURE 1-0 5: MOTS AVEC DIFFERENTS SOUS-MOTS.**

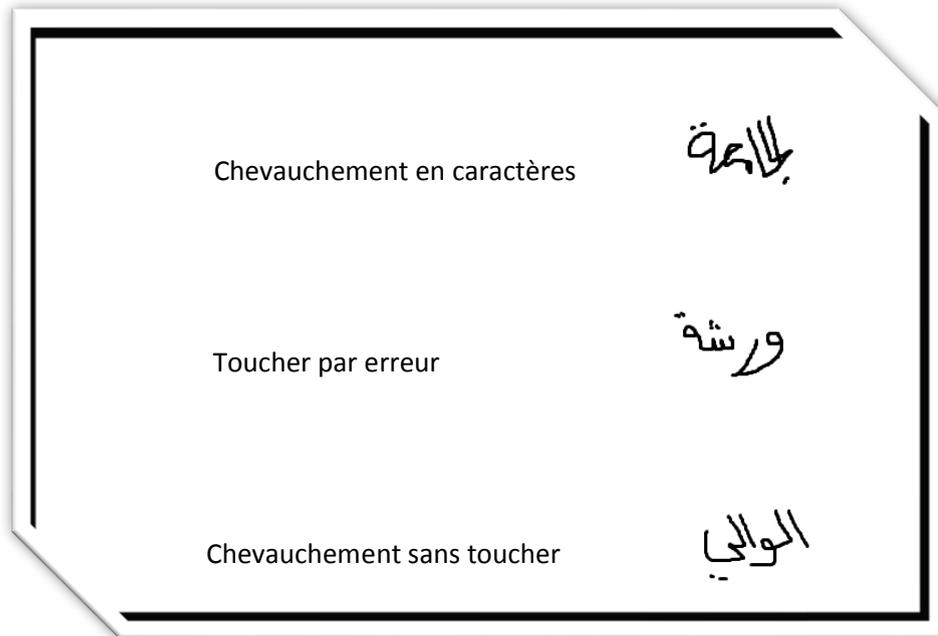
Le même mot dans un texte écrit à la main en arabe pourrait avoir des styles et des tailles différents pour le même auteur ainsi que pour différents auteurs, comme indiqué sur la figure 1-06:



**FIGURE 1-0 6: DES STYLES ET DES TAILLES DIFFERENTS POUR LE MEME MOT.**

Par ailleurs ,deux ou plusieurs caractères dans la langue arabe manuscrite peuvent être combinés verticalement et représentés par différentes formes. C'est une ligature c'est à dire un chevauchement entre les caractères voisins ,ce qui signifie que le second caractère peut apparaître avant le premier dans certains cas [7].Une ligature peut se produire lorsque des caractères tels que (ح, خ, ج, م, ل) apparaissent après certains autres caractères.

Dans le 1<sup>er</sup> cas de la Figure 1-07 il apparait que le chevauchement des trois premiers Caractères (ا, ل, ج). Dans certains cas, deux caractères peuvent se toucher par erreur, comme les caractères (و, ر) comme le montre le 2<sup>ème</sup> cas de la Figure 1-07. De plus, deux caractères peuvent être chevauchés verticalement sans toucher l'un l'autre, tels que les caractères (و, ا, د), dans le 3<sup>ème</sup> cas de la Figure 1-07.



**FIGURE 1-0 7: DIFFERENTS LIGATURES DES MOTS.**

Dans l'écriture arabe, deux lettres liées le sont par une liaison généralement horizontale , laquelle peut être plus ou moins allongée en fonction des besoins de la justification., il n'est pas tout à fait correct de décrire l'écriture arabe comme simplement formée de lettres qui peuvent être liées ou pas : entre deux lettres d'un mot il y a toujours un trait inter-lettre, cette liaison , qui peut être absente lorsque la première lettre ne se lie pas à la suivante, ou que la seconde lettre ne se lie pas à la précédente ; mais qui doit être tracée sinon.

À l'origine, l'alphabet arabe n'avait pas de telles variantes, qui sont nées des déformations impliquées par la graphie cursive, laquelle procède par des adaptations liées à la nécessité de ne pas lever le calame pour ne pas interrompre le trait. De simples variantes non pertinentes, les allographes ont ensuite acquis le statut de formes normées et obligatoires.

Les lettres de l'alphabet arabe peuvent prendre quatre formes souvent différentes, suivant qu'elles sont isolées, liées à celle qui précède, à celle qui suit, ou aux deux.

Dans les textes arabes manuscrits, certains caractères semblent similaires, bien qu'ils soient différents, et il est difficile même pour l'œil humain de trouver la différence [8]. Il existe des différences entre la longueur et la largeur des caractères arabes, par exemple (ب, ل). En outre, le même caractère peut sembler différent dans ses diverses formes, telles que (ف،غ) [8].

En outre, la grande similitude entre certains des caractères manuscrits rend la classification de ces caractères un autre défi, comme le montre la figure 1-08.

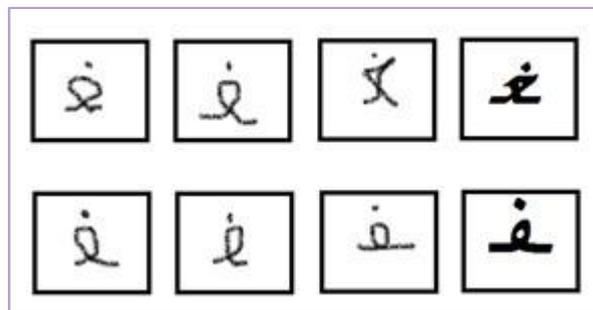
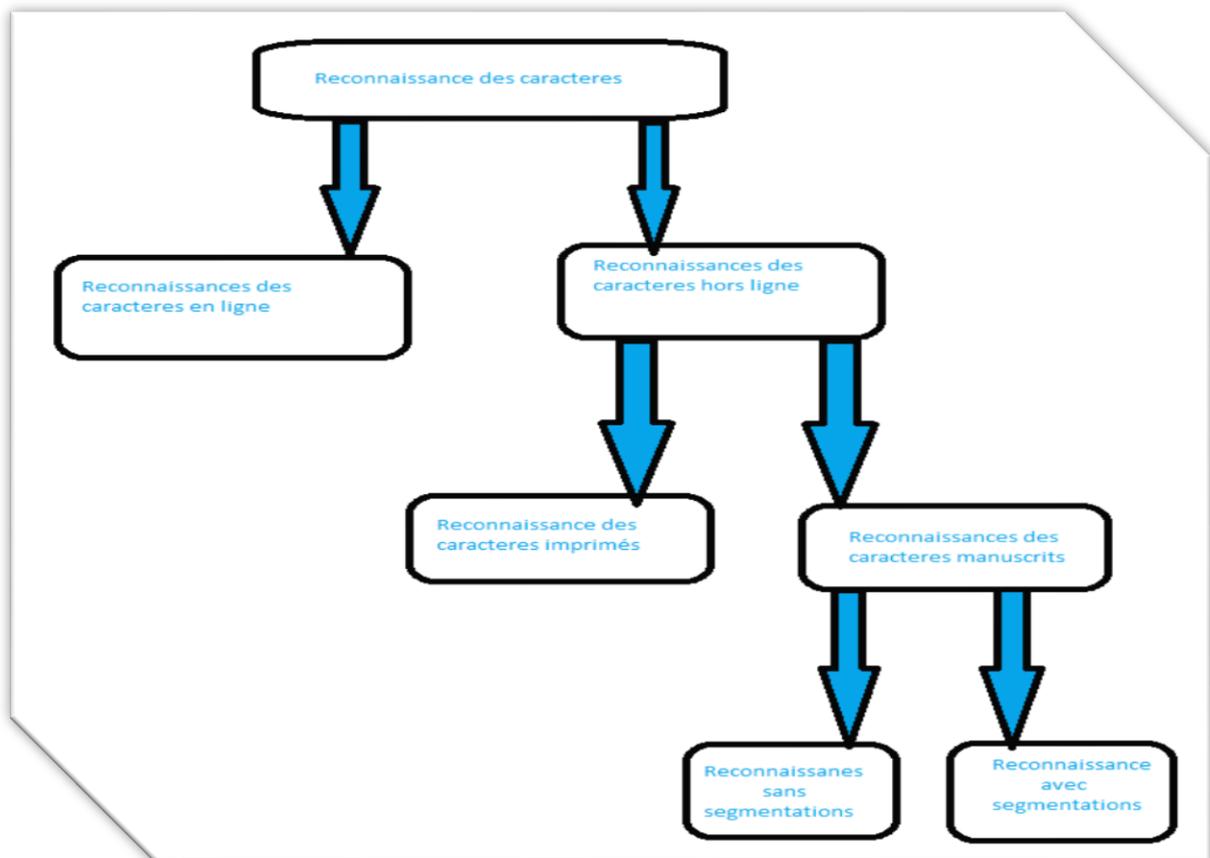


FIGURE 1-0 8: SIMILITUDE ENTRE CERTAINS CARACTERES.

## 1.6 Architecture d'un système de reconnaissance des caractères arabes :

La reconnaissance d'écriture manuscrite arabe reste toujours un problème ouvert. Ceci à cause des difficultés auxquelles sont confrontés les chercheurs et les développeurs, telles que la variabilité de la forme du style et l'inclinaison de l'écriture et de la nature cursive et connectée de l'écriture arabe manuscrite. La reconnaissance des caractères manuscrits hors ligne comporte de nombreux défis en raison de la complexité et de l'ambiguïté des styles d'écriture. Les systèmes de reconnaissance des caractères peuvent être classés en différentes catégories, comme le montre la figure 1-09.



**FIGURE 1-0 9: CLASSIFICATION DES SYSTEMES DE RECONNAISSANCE DES CARACTERES.**

La reconnaissance des caractères peut être divisée en deux types: reconnaissance des caractères en ligne et hors ligne .

Pour la reconnaissance en ligne des caractères la classification se fait alors que l'utilisateur écrit. Cette technique utilise un équipement tel qu'un crayon spécial et une tablette, la trace numérisée du stylo étant utilisée pour reconnaître le caractère.

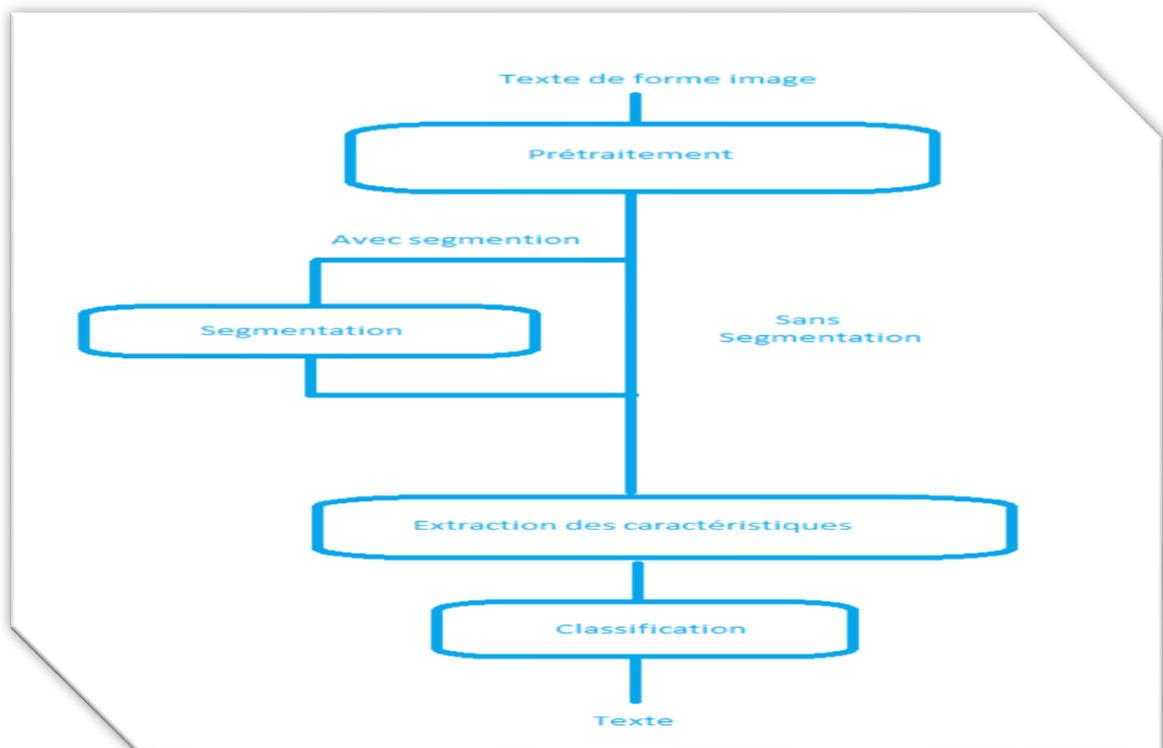
De ce fait les documents pré-écrits sont exclus et ne sont pas concernés par cette méthode. Pour plus de détail sur la différence entre la reconnaissance des caractères en ligne et hors ligne, les lecteurs sont référés à l'article [9]. D'autre part, les systèmes de reconnaissance hors ligne traitent des images numérisées des documents précédemment écrits.

La reconnaissance hors ligne des textes peut être divisée en deux catégories : la reconnaissance des caractères imprimés et des caractères manuscrits. Les caractères

imprimés ont un style et une taille pour n'importe quelle police donnée. Cependant, les caractères manuscrits ont des styles et des tailles qui varient, à la fois pour le même auteur et entre les différents écrivains.

Les mots manuscrits peuvent être reconnus de deux façons : la reconnaissance d'un mot entier sans segmentation, ou la reconnaissance basée sur la segmentation. En raison de la présence de la ligature et de la nature cursive de l'écriture arabe, plusieurs chercheurs ont présenté des techniques basées sur la reconnaissance du mot entier sans segmentation [7, 9]. La reconnaissance manuscrite des mots implique plusieurs étapes pour obtenir la classification en tant que fichier texte. La figure 1-10 s'illustre les étapes générales de reconnaissance de texte manuscrit. Puisque le processus de segmentation est la principale source d'erreurs dans la reconnaissance, la plupart des systèmes évitent cette étape et ne reconnaissent que le mot entier.

La figure 1-0 10 indique différentes voies pour la reconnaissance des caractères arabes avec ou sans segmentation :



**FIGURE 1-0 11: ÉTAPES GENERALES POUR LA RECONNAISSANCE DU TEXTE MANUSCRIT.**

## **1.6.1 Prétraitement :**

Le prétraitement est l'étape cruciale dans tout système de reconnaissance [10]. Le but de cette étape dans la reconnaissance de texte manuscrit est d'améliorer la lisibilité de l'image de texte et de supprimer les détails qui n'ont pas de pouvoir discriminatif dans le processus de reconnaissance [11].

L'étape de prétraitement comprend habituellement plusieurs tâches : binarisation, suppression du bruit, détection de référence et normalisation.

### **1.6.1.1 Binarisation :**

La binarisation est un processus qui permet de convertir une image de texte en format binaire: les valeurs des pixels d'arrière-plan comme 1 (blanc) et les valeurs des pixels d'avant-plan comme 0 (noir). Ce processus est réalisé en choisissant une valeur de méthode de seuillage efficace. Un des avantages de ce procédé est qu'il augmente la vitesse de traitement [12]. La distorsion de l'image se produit généralement pendant le processus de numérisation. Les petits objets qui ne font pas partie de l'écriture peuvent être considérés comme du bruit et doivent être supprimés.

### **1.6.1.2 Suppression de bruit :**

Le bruit est introduit lors de l'acquisition d'images d'entrée via des scanners optiques ou des dispositifs d'écriture et provoque une distorsion dans l'image d'entrée. Avant la reconnaissance du texte, il est essentiel de supprimer ces variantes. Bien qu'il existe des nombreuses techniques utilisées pour éliminer le bruit, deux approches principales sont largement utilisées [13] :

#### **A. FILTRAGE :**

Il existe plusieurs filtres de domaine spatial et de fréquence qui peuvent être conçus pour supprimer le bruit et réduire le nombre des points parasites. Les filtres peuvent être conçus pour le seuillage, l'affûtage, le lissage et pour le réglage du contraste. La mise en œuvre d'un filtre spatial linéaire implique la conception d'un masque et l'exécution d'une convolution de l'image et du masque comme suit :

$$Out(x, y) = \sum_i \sum_j w_{i,j} * inp(x - i, y - j)$$

Ou  $w_{i,j}$  est les poids des niveaux de gris des pixels de masque à l'emplacement (i,j).

## B. OPERATIONS MORPHOLOGIQUES MATHÉMATIQUES :

Des nombreux filtres spatiaux et de domaine fréquentiel peuvent être conçus pour différentes applications, comme le lissage [14] et l'élimination du bruit [15]. Le lissage réduit le bruit en utilisant des opérations de morphologie mathématique. Il est généralement réalisé de deux façons: en ouvrant les espaces entre les objets en contact ou en comblant des petits espaces [16]. Les deux méthodes, l'ouverture et la fermeture, appliquent respectivement les opérations morphologiques d'érosion et de dilatation. Mathématiquement [17], l'érosion de A par B est l'ensemble des tous les points z et peut être définie comme :

$$\text{Erosion de A par B : } A \ominus B = \{Z | (B)_z \cap A \neq \emptyset\}$$

$$\text{Dilatation de A par B : } A \oplus B = \{Z | (\hat{B})_z \cap A \neq \emptyset\}$$

### 1.6.1.3 Détection de la ligne de base :

La ligne de base est une ligne horizontale imaginaire qui relie les caractères d'un mot. La détection de la ligne de base est l'une des tâches importantes dans l'étape de prétraitement [11]. Il pourrait être utile dans la correction oblique ainsi que la segmentation des caractères [18]. Une méthode commune utilisée pour extraire la ligne de base utilise une projection horizontale et cela fonctionne bien, en particulier avec le texte imprimé [19]. La détection de la ligne de base permet de déterminer certaines caractéristiques structurelles, telles que les points et leurs positions, les ascendants et les descendants, ainsi que la correction d'inclinaison.

### 1.6.1.4 Seuillage :

Est le processus de conversion d'une image en noir et blanc, image binaire. Les principaux avantages du processus de seuillage sont qu'il réduit les exigences de stockage et augmente la vitesse de traitement. Par conséquent, il est recommandé de convertir des images en couleur ou en gris en images binaires en choisissant une valeur de seuil correcte.

L'opération de seuillage sépare le premier plan (encre) de son arrière-plan (papier) [6], en utilisant la fonction de mappage suivante : Supposons que l'image d'entrée est  $inp(x,y)$ , et de sortie est  $Out(x,y)$  donc :

$$Out(x,y) = \begin{cases} 0 & \text{si } inp(x,y) < seil \\ 1 & \text{si } inp(x,y) > seil \end{cases}$$

### 1.6.1.5 Squelettisation :

Il n'existe pas de définition unique du squelette. Le squelette doit seulement remplir trois conditions: · Il doit être aussi fin que possible (typiquement, 1 pixel d'épaisseur). · Il doit respecter la connexité. · Il doit être centré dans la forme qu'il représente. En effet, les définitions d'un squelette varient selon l'espace de travail (continu ou discret, image binaire ou en niveaux de gris) ainsi que les besoins requis par l'application. La première définition du squelette a été émise par Blum [1] par l'analogie du feu de prairie.

Un squelette doit vérifier un certain nombre de propriétés dont l'importance varie selon le domaine d'application. Ces propriétés sont indépendantes de la qualité et du type d'images en entrée. En effet, que l'image soit binaire ou en niveau de gris, 2D ou 3D, l'objectif derrière sa squelettisation est toujours le même. Par conséquent, les propriétés de cette opération restent inchangées. Nous les listons par ordre de priorité et expliquons l'importance de chacune pour les domaines d'application.

L'homotopie : C'est la propriété du squelette la plus importante. Elle consiste à conserver la topologie de l'image. En d'autres termes, le squelette doit conserver les relations de connexité : préserver le nombre de composantes connexes de l'objet ainsi que celles du fond (les trous) et par conséquent, préserver les caractéristiques structurelles telles que le nombre de nœuds, de segments et d'extrémités de l'objet initial. Cette propriété est

utile à l'appariement d'objet dans des domaines comme la reconnaissance de caractères, la reconnaissance de symboles et la biométrie...

L'amincissement : La squelettisation par l'amincissement. la préservation de la géométrie est un critère d'évaluation de la squelettisation répandu dans ce cadre applicatif, ce qui explique l'importance de cette propriété.

L'épaisseur unitaire : Cette propriété est utile aux applications qui exploitent le squelette pour extraire des paramètres structurels dans un processus d'appariement comme c'est le cas en reconnaissance de caractères

La reconstructibilité Certaines applications requièrent l'utilisation d'une squelettisation réversible pour la reconstruction de l'objet initial. Cette propriété n'est pas toujours respectée par les méthodes de squelettisation parce qu'elle ne présente pas d'intérêt dans les domaines d'application visés par ces méthodes.

L'invariance au bruit L'objectif ultime de la squelettisation est d'être appliqué sur des images réelles, dans des contextes applicatifs variés. Cependant, les images réelles sont dans la majorité des cas bruitées. C'est pourquoi, un des défis de la squelettisation est de produire un squelette qui reste inchangé face à l'apparition de bruit sur les images à traiter.

La plupart des systèmes de reconnaissance mettent en œuvre l'éclaircissage en tant que phase de prétraitement avant les phases de segmentation, d'extraction des caractéristiques et de reconnaissance. Grâce à l'opération consistant à réduire la largeur du trait du texte à un seul pixel, des approches d'amincissement sont utilisées pour obtenir des squelettes de texte imprimé et manuscrit . Les chercheurs en écriture arabe font face à un problème après avoir appliqué des algorithmes d'amincissement.



FIGURE 1-1 1: RESULTAT D'UNE OPERATION D'ECLAIRCIE.

Cela est dû à la forme similaire des lettres arabes avec différents points tels que, ش, ذ, ي. Pour surmonter un tel problème, les points sont extraits avant éclaircissage et sont reconnus séparément. La figure 1-11 montre les résultats d'une opération d'éclaircie.

#### **1.6.1.6 Normalisation :**

L'écriture manuscrite a des styles et des tailles différents. Par conséquent, le processus de normalisation est l'une des tâches les plus importantes dans le processus de reconnaissance du texte. Il permet de réduire la variation entre les images du texte et d'ajuster la taille du caractère ou du mot. La normalisation des tailles est couramment utilisée pour réduire la variation de taille et ajuster les tailles des caractères ou des mots à une taille identique [20]. Certains chercheurs [21] ont proposé de diviser le caractère en un certain nombre des zones, puis chaque zone de mise à l'échelle séparément. En raison d'inexactitudes dans le processus de numérisation, le document numérisé peut être légèrement tourné. La correction d'inclinaison est utilisée pour corriger l'angle d'orientation. La ligne de base du texte est habituellement utilisée pour détecter l'obliquité dans les textes arabes. En 2005, d'autres chercheurs ont proposé un algorithme pour corriger l'asymétrie et incliner le mot, basé sur la transformation du Radon []. Nous citerons ici trois techniques de normalisation :

La correction de l'inclinaison des lignes de texte (également appelée correction de "skew"), consiste à redresser horizontalement les lignes d'écriture obliques et l'inclinaison des lignes de texte .

La correction de l'inclinaison des lettres : Certains scripteurs écrivent leurs lettres de façon inclinée par rapport à l'axe vertical. Cette inclinaison de l'écriture est également appelée « slant ». Les lettres peuvent être inclinées vers la droite ou vers la gauche. Pour la même raison que dans le paragraphe précédent, il convient de corriger cette inclinaison de l'écriture pour la rendre la plus indépendante possible des spécificités d'écriture du scripteur

La ligne de base : Les lignes de base encadrent la bande des minuscules et délimitent les zones contenant les ascendants et descendants. Ces lignes sont importantes en reconnaissance de l'écriture. Cette information est utilisée pour détecter ascendants et

descendants, et également pour normaliser les primitives, les rendant ainsi moins dépendantes de la hauteur de l'écriture.

## 1.6.2 Segmentation :

Isolement des caractères ou "segmentation" c'est-à-dire segmenter le texte en sous unités tels que des lignes des mots ou des caractères.

Plusieurs caractères connectés en raison d'artefacts d'image doivent être séparés. Les caractères uniques qui sont divisés en plusieurs morceaux en raison d'artefacts doivent être connectés.

La segmentation des polices à pas fixe s'effectue simplement en alignant l'image sur une grille uniforme en fonction du point où les lignes de grille verticales intersectent le moins souvent les zones noires. Pour les polices proportionnelles, des techniques plus sophistiquées sont nécessaires car les espaces entre les lettres peuvent parfois être supérieurs à ceux entre les mots, et les lignes verticales peuvent croiser plus d'un caractère.

Un autre niveau de modélisation spécifique à l'écriture arabe est le niveau pseudomot (ou PAW : Pieces of Arabic Words). Un PAW est une séquence de lettres liées, ce qui donne l'aspect de cursivité à l'écriture arabe, notons qu'un caractère isolé peut constituer un PAW à lui seul. La notion de PAW introduit une segmentation naturelle de l'écriture arabe et fait apparaître une nouvelle approche de reconnaissance appelée approche pseudo analytique. Cette approche, peu exploitée dans l'écriture arabe, offre une solution intermédiaire aux limites de deux approches analytique et globale : d'une part, elle évite le problème délicat de la segmentation en lettres lié à l'approche analytique. D'autre part, l'interprétation des PAWs plutôt que les mots conduit à une réduction de la taille et de la complexité du vocabulaire et ouvre la voie à l'exploration des vocabulaires plus étendus que ceux abordés par l'approche globale.

### 1.6.2.1 Types de segmentation :

#### A. SEGMENTATION D'UNE PAGE EN LIGNES :

En général, le paragraphe ou la page se compose de plusieurs lignes. En reconnaissance de texte, le texte comporte plusieurs lignes se segmentant en lignes séparées. Il existe certaines méthodes utilisées à cet effet, telles que la projection horizontale [ 22 ].

## **B. SEGMENTATION D'UNE LIGNE EN MOTS :**

Après la segmentation des lignes, la ligne est segmentée en mots. Segmenter la ligne en mots dépend de l'espace entre les mots. Ce pendant, certains caractères ne se connectent pas à un caractère ultérieur d'un mot, ce qui entraîne une séparation du mot en sous-mots. Les espaces plus longs séparent les mots tandis que les espaces courts séparent les sous-mots. Par conséquent, la plupart des chercheurs supposent dans leur technique que l'espace entre les mots est plus grand que l'espace entre les sous-mots [23]. Certaines des méthodes analysent les distances entre les composants connectés pour segmenter les mots [24].

## **C. SEGMENTATION D'UN MOT EN CARACTERES :**

Il s'agit ici de la segmentation d'un mot en caractères individuels. Comme décrit dans [25], les points de segmentation sont identifiés à la fin d'un caractère et au début de la suivante. La nature cursive de la langue arabe rend la segmentation des mots en caractères individuels une tâche difficile.

### **1.6.2.2 Différentes techniques appliquées pour la segmentation :**

#### **A. Technique de segmentation basée sur la projection verticale :**

Cette technique est basée sur le fait que l'épaisseur de la course de connexion entre les caractères dans le mot est toujours inférieure aux autres parties.

La technique de projection verticale réduit les informations bidimensionnelles en une seule dimension..

De nombreux algorithmes ont été proposés dans cette technique pour la segmentation des caractères. La projection verticale est utile pour segmenter les mots, les sous-mots et les

caractères, tandis que la projection horizontale est habituellement utilisée pour la segmentation de ligne et l'extraction de ligne de base.

### **B. Technique de segmentation basée sur l'éclaircissement :**

Lors de la reconnaissance des caractères, le squelette d'un caractère fournit les informations essentielles sur la forme du caractère. Un certain nombre d'algorithmes ont été proposés pour extraire les squelettes.

### **C. Technique de segmentation basée sur le traçage des contours :**

Une autre technique basée sur le traçage des contours d'un mot est également utilisée pour la segmentation. Cette technique est obtenue en traçant le contour extérieur du corps principal du mot. Dans [26] ont proposé un algorithme de segmentation des caractères basé sur l'analyse de contour et les règles topologiques. Il commence par trouver le point minimum local du contour inférieur. Ensuite, il utilise les règles topologiques pour déterminer si le point de minimum local est un point de segmentation.

### **D. Technique de Segmentation basée sur les réseaux neuronaux artificiels :**

Les RNAs sont utilisés pour vérifier les points de segmentation valides. Dans [27], Hamid et Haraty ont proposé une technique pour segmenter des textes arabes manuscrits basés sur RNAs. Ils ont identifié des points de pré-segmentation en utilisant les caractéristiques topographiques de chaque bloc des caractères connecté, comme la densité des pixels noirs et les trous. Les RNA sont ensuite utilisées pour vérifier ces points de segmentation. Les points potentiels sont classés manuellement en points de segmentation valides et non valides, et ces points sont alimentés avec leurs caractéristiques, dans les RNA.

### **E. Technique de segmentation basée sur des opérateurs morphologiques :**

Des opérations morphologiques telles que la fermeture suivie de l'ouverture sont utilisées pour la segmentation des caractères. Dans [28] ont proposé un algorithme pour segmenter des mots arabes manuscrits. Ils supposent que les traits verticaux qui peuvent

représenter le début et la fin du caractère sont trouvés par singularités. Les régularités qui ont des informations sont nécessaires pour connecter un caractère à un caractère ultérieur. Par conséquent, l'image originale est balayée de droite à gauche afin d'identifier les régularités. Les singularités sont identifiées en appliquant une ouverture au mot. Les régularités sont extraites en soustrayant les singularités de l'image du mot. Ces régularités sont utilisées pour identifier les candidats des points de segmentation.

Cependant, la plupart des algorithmes de segmentation actuellement proposés ne résolvent pas le problème du chevauchement des caractères en écriture arabe. L'étape de segmentation est l'étape la plus difficile et la principale source d'erreurs dans la reconnaissance. Les techniques de segmentation représentent donc encore un défi dans la reconnaissance du texte et doivent être améliorées.

### **1.6.3 Extraction des caractéristiques :**

Dans les textes imprimés et manuscrits, les caractéristiques saisissent les informations extraites de l'image textuelle. Ces informations doivent avoir les caractéristiques essentielles du caractère ou du mot qui le différencient; En d'autres termes, filtrer tous les attributs et préserver les propriétés qui rendent un caractère ou un mot différent d'un autre [16]. Ces informations sont transmises au classificateur pour faciliter le processus de classification. Les techniques d'extraction des caractéristiques diffèrent d'une application à l'autre. Les techniques qui réussissent dans une application, peuvent ne pas être couronnées de succès pour d'autres applications [29]. Par conséquent, la sélection de la méthode d'extraction des caractéristiques reste l'étape la plus importante pour obtenir une précision de reconnaissance élevée. Les caractéristiques des textes manuscrits peuvent être classées dans les catégories suivantes :

#### **1.6.3.1 Caractéristiques structurelles :**

Les caractéristiques structurelles sont les caractéristiques les plus courantes utilisées par les chercheurs [30]. Ils illustrent les caractéristiques géométriques et topologiques de l'image textuelle en décrivant leurs propriétés locales et globales [16]. Les caractéristiques structurelles dépendent de la catégorie du modèle à classer. Pour un texte en arabe, par

exemple, les caractéristiques incluent les points et leur position, les traits, la largeur et la hauteur du trait, les directions, l'intersection des segments de ligne et des boucles [31].

### **1.6.3.2 Caractéristiques statistiques :**

Les caractéristiques statistiques analysent la répartition spatiale des pixels en comptant les caractéristiques locales à chaque pixel et en dérivant un ensemble des statistiques à partir des distributions des caractéristiques locales. Les caractéristiques statistiques majeures du caractère incluent le zonage, où le caractère se divise en zone se chevauchant ou non se chevauchant et la distribution de densité des pixels de caractère dans différentes régions est analysée. Mohiuddin<sup>0</sup> et Mao dans [32] ont mesuré la direction du contour du caractère en divisant le caractère d'image en zones. Ensuite, des histogrammes des codes de chaîne sont utilisés pour calculer la direction du contour dans ces régions.

### **1.6.3.3 Transformation globale :**

Les techniques de transformation globale ont la capacité de convertir la représentation des pixels en une forme plus compacte. Ces techniques peuvent représenter le signal par une combinaison linéaire d'une série des fonctions plus simples et bien définies.

L'expansion en série fournit un codage compact par les coefficients de la combinaison linéaire. Une des transformations les plus simples est celle qui représente le squelette ou le contour d'un caractère sous forme d'une chaîne de codes de directions. Parmi les méthodes de transformation, nous pouvons citer les transformées de Hough et de Fourier.

### **1.6.4 Classification :**

Il s'agit ici d'identifier un objet en comparant ses caractéristiques à ceux d'un ensemble donné des classes. Il suppose que les cours d'apprentissage et le modèle d'apprentissage ont été fournis. Le modèle d'apprentissage se compose d'un ensemble d'instances où les bonnes réponses(sorties) pour chaque entrée sont fournies. Par conséquent, un classificateur est utilisé pour identifier un objet en utilisant ses

caractéristiques, ceux-ci sont ensuite comparés et sauvegardés comme modèles pour les classes formées. Les caractéristiques d'un objet dans la phase de test seront extraites et comparées avec les caractéristiques des modèles d'apprentissage pour identifier l'objet inconnu. Les méthodes courantes dans la phase de classification sont basées sur:

▮ Réseaux des neurones artificiels.

▮ Modèle de Markov caché.

▮ k-voisin le plus proche.

## **1.7 Conclusion :**

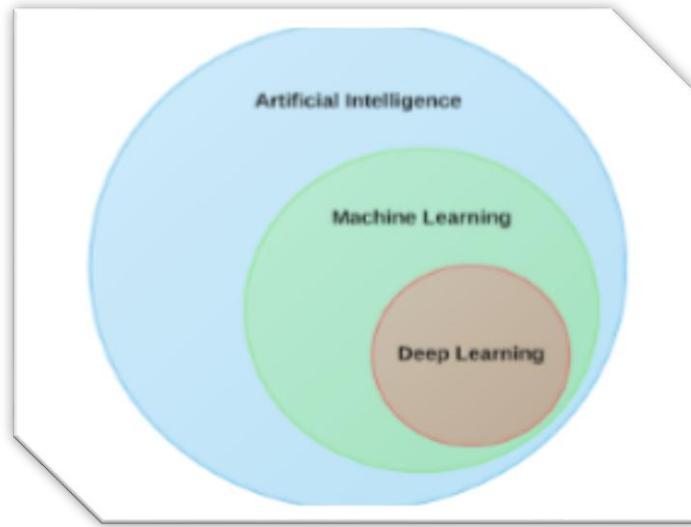
Dans ce chapitre nous avons traité la problématique de la reconnaissance des caractères arabes et les méthodes utilisées à cet effet. Le principe de reconnaissance est basé sur l'obtention d'une base de données contenant des images de chaque caractère puis en passant par trois étapes nécessaires qui sont : le prétraitement puis l'extraction des attributs ,puis la classification. Nous avons utilisé les réseaux de neurones comme classifieur qui seront détaillés ainsi que les algorithmes y afférents dans les chapitres qui suivent.

## **Chapitre 2.**

# **Les réseaux des neurones convolutionnels**

## 2.1 Introduction :

Le Deep Learning est un nouveau domaine de recherche du ML (Machine Learning) , qui a été introduit dans le but de rapprocher le ML de son objectif principal : l'intelligence artificielle .Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.



**FIGURE 2-0 1: LA RELATION ENTRE L'INTELLIGENCE ARTIFICIELLE, LE ML ET LE DEEP LEARNING**

Le Deep Learning est basé sur l'idée des réseaux de neurones artificielles et il est taillé pour gérer de larges quantités de données en ajoutant des couches au réseau. Un modèle de deep Learning a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composées de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petit à petit à travers chaque couche avec une intervention humaine minimale.

Sur les cinq dernières années, le deep learning est passé d'un marché de niche où seulement une poignée de chercheurs s'y intéressait au domaine le plus prisé par les chercheurs. Les recherches en relation avec le deep learning apparaissent maintenant dans les top journaux comme Science Nature et Nature Methods pour ne citer que ceux-là. Le deep learning a appris à conduire une voiture, diagnostiquer le cancer et l'autisme et même à devenir un artiste.

Le terme "Deep Learning" a été introduit pour la première fois au ML par Dechter (1986) , et aux réseaux neurones artificiels par Aizenberg et al (2000) .

## 2.2 Algorithmes de Deep Learning :

Il existe différents algorithmes de Deep Learning : Nous pouvons ainsi citer:

Les réseaux de neurones profonds (Deep Neural Networks). Ces réseaux sont similaires aux réseaux MLP mais avec plus de couches cachées. L'augmentation du nombre de couches, permet à un réseau de neurones de détecter de légères variations du modèle d'apprentissage, favorisant le sur-apprentissage ou sur-ajustement (« overfitting »).

Les réseaux de neurones convolutionnels (CNN ou Convolutional Neural Networks) : Le problème est divisé en sous parties, et pour chaque partie, un «cluster» de neurones sera créé afin d'étudier cette portion spécifique. Par exemple, pour une image en couleur, il est possible de diviser l'image sur la largeur, la hauteur et la profondeur (les couleurs).

La machine de Boltzmann profonde (Deep Belief Network): Ces algorithmes fonctionnent suivant une première phase non supervisée, suivi de l'entraînement classique supervisé. Cette étape d'apprentissage non-supervisée, permet, en outre, de faciliter l'apprentissage supervisé.

## 2.3 Neurone biologique :

Le neurone est une cellule nerveuse. Elle se compose d'un corps cellulaire appelé « Soma » qui contient le noyau (où se déroulent les activités cellulaires vitales) de prolongement appelé « Neurite » . Ces dernières sont de deux types, les Dendrites qui servent de canaux d'entrées et l'axone, unique qui est le canal de sortie [33]. Au point de vue fonctionnel, on considère le neurone comme une entité polarisée, c'est-à-dire que l'information ne se transmet que dans un seul sens : des dendrites vers l'axone. Le neurone va donc recevoir des informations, venant d'autres neurones, grâce à ses dendrites. Il va ensuite y avoir sommation, au niveau du corps cellulaire, de toutes ces informations et via un potentiel d'action (un signal électrique). Le résultat de l'analyse va transiter le long de l'axone jusqu'aux terminaisons synaptiques. A cet endroit, lors de l'arrivée du signal, des vésicules synaptiques vont venir fusionner avec la

membrane cellulaire, ce qui va permettre la libération des neurotransmetteurs (médiateurs chimiques) dans la fente synaptique. Le signal électrique ne pouvant pas passer la synapse (dans le cas d'une synapse chimique), les neurotransmetteurs permettent donc le passage des informations, d'un neurone à un autre. Les synapses possèdent une sorte de « mémoire » qui leur permet d'ajuster leur fonctionnement. En fonction de leur « histoire », c'est-à-dire de leur activation répétée ou non entre deux neurones, les connexions synaptiques vont donc se modifier. Ainsi, la synapse va faciliter ou non le passage des influx nerveux. Cette plasticité est à l'origine des mécanismes d'apprentissage.

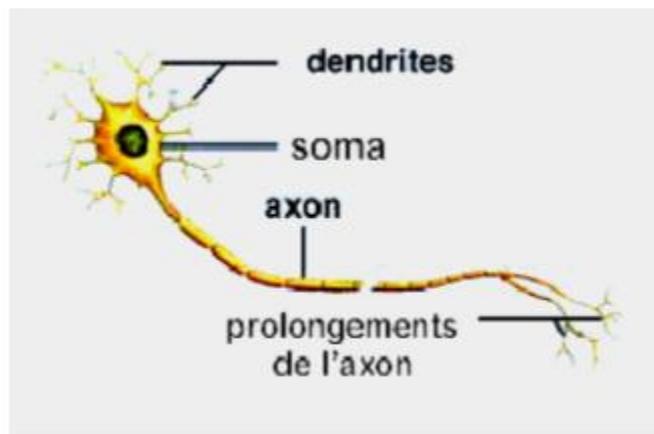


FIGURE 2-0 2: MODELE D'UN NEURONE BIOLOGIQUE.

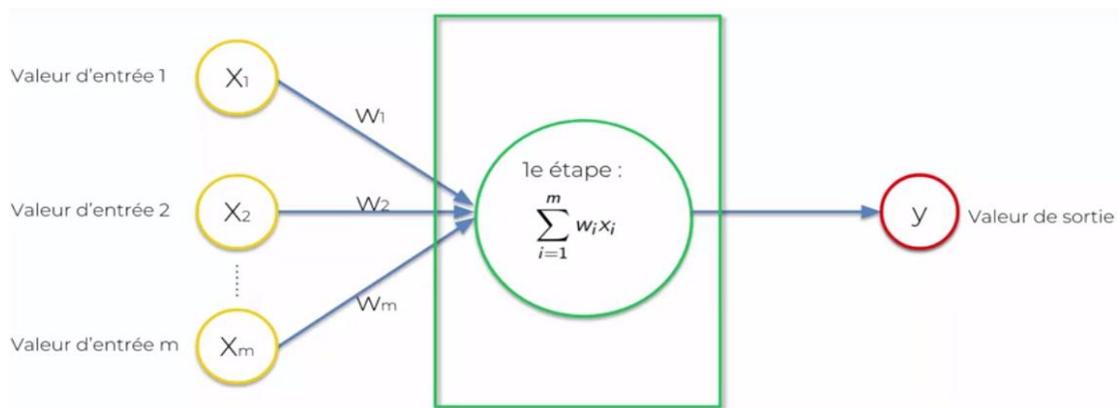
## 2.4 Réseaux de neurones artificiels :

Les réseaux de neurones artificiels sont simplement des systèmes inspirés du fonctionnement des neurones biologiques. Le plus célèbre d'entre eux est le perceptron multicouche, un système artificiel capable d'apprendre par... l'expérience ! Introduit en 1957 par Franck Rosenblatt, il n'est véritablement utilisé que depuis 1982 après son perfectionnement. Grâce à la puissance de calcul des années 2000, le perceptron s'est largement démocratisé et est de plus en plus utilisé.

La première description formelle des réseaux des neurones artificiels a été proposée par McCulloch & Pitts [34]. Ces systèmes sont des algorithmes prenant plusieurs valeurs en entrée, ces valeurs sont traitées par plusieurs fonctions et en sortit retourne une valeur. Ces fonctions passent d'abord par une phase d'apprentissage dans le but de calibrer les résultats en sortie. On donne à un réseau des valeurs d'entrée, on connaît le résultat de sortie et on vérifie que le

réseau retourne le résultat attendu, tant que ce n'est pas le cas on continu de faire des tests jusqu'à ce que le réseau soit correctement configuré et qu'il soit en mesure de répondre systématiquement le résultat attendu. Ensuite il agit comme une boîte noire, on lui donnera des valeurs d'entrée dont on ne connaît pas encore le résultat et il nous donnera une valeur en sortie sans qu'on ne sache ce qu'il a fait. Ainsi un réseau ne donne pas forcément de règle exploitable par un humain.

L'apprentissage par l'expérience permet ensuite d'utiliser les réseaux neuronaux pour de la reconnaissance d'image, de la prédiction boursière, la résolution de problèmes mathématiques, etc.



**FIGURE 2-0 3: MODELE D'UN NEURONE ARTIFICIEL.**

## 2.5 Perceptron :

Le réseau neuronal le plus simple est le perceptron qui se compose d'un seul neurone entièrement fonctionnel, est un cas particulier de réseau de neurones, mais très souvent, quelqu'un qui dit « réseau de neurones artificiels » (artificial neural network) pense au perceptron (cas le plus simple et le plus répandu).

## 2.6 Techniques utilisées par les réseaux de neurones :

Les techniques courantes de Machine Learning entrant dans la conception d'applications de réseaux de neurones comprennent l'apprentissage supervisé et non supervisé, la classification, la régression, la reconnaissance de formes et le clustering.

### **2.6.1 Apprentissage supervisé :**

Les réseaux de neurones supervisés sont formés pour générer les sorties voulues en réponse aux échantillons d'entrée, afin de les adapter tout particulièrement à la modélisation et au contrôle de systèmes dynamiques, à la classification de données bruyantes et à la prédiction des événements futurs. Deep Learning Toolbox™ inclut quatre types de réseaux supervisés : contrôleur boucle ouvert, base radiale, dynamique et quantification de vecteur d'apprentissage.

### **2.6.2 Classification :**

La classification est un type de Machine Learning supervisé dans lequel un algorithme « apprend » à classer de nouvelles observations à partir d'exemples de données étiquetées.

### **2.6.3 Régression :**

Les modèles de régression décrivent la relation entre une variable réponse (sortie) et une ou plusieurs variables prédictives (entrée).

### **2.6.4 Reconnaissance de formes :**

La reconnaissance de formes est une composante importante des applications des réseaux de neurones dans les domaines de la vision par ordinateur, du traitement radar, de la reconnaissance vocale et de la classification de texte. Elle fonctionne en classant les données d'entrée en objets ou en classes selon des caractéristiques clés, en utilisant une classification supervisée ou non supervisée.

Par exemple, en vision par ordinateur, les techniques non supervisées de reconnaissance de formes servent pour la reconnaissance optique des caractères (ROC), la détection des visages, la reconnaissance faciale, la détection d'objets et la classification d'objets. Dans le traitement d'image et la vision par ordinateur, les techniques non supervisées de reconnaissance de formes servent pour la détection d'objets et la segmentation d'images.

## 2.6.5 Apprentissage non supervisé :

La formation d'un réseau de neurones non supervisé s'effectue en le laissant s'ajuster continuellement aux nouvelles entrées. Il est utilisé pour tirer des conclusions à partir d'ensembles de données composés de données d'entrée sans réponses labellisées. Vous pouvez l'utiliser pour découvrir des distributions naturelles, des catégories et des relations de catégories au sein des données.

Deep Learning Toolbox inclut deux types de réseaux non supervisés : les couches cachées et les cartes auto-organisatrices de Kohonen.

## 2.6.6 Clustering :

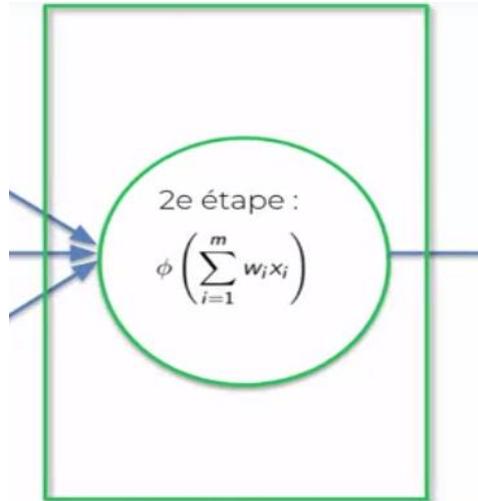
Le clustering est une approche d'apprentissage non supervisée dans laquelle les réseaux de neurones peuvent être utilisés pour effectuer une analyse exploratoire des données afin de trouver des modèles cachés ou des regroupements dans les données. Ce processus implique de grouper les données par similarité. L'analyse de séquence génomique, l'étude de marché et la reconnaissance d'objets sont des exemples d'applications d'analyse de cluster.

## 2.7 Fonctions d'activation :

On dit qu'une fonction d'activation est activée si sa sortie n'est pas nulle. Il est également dit avoir une forte activation si la sortie est relativement élevée et une activation faible si sa sortie est relativement faible. Une fonction d'activation doit être non linéaire, continuellement différentiable et monotone, il est en outre souhaité que la fonction  $f(x) \approx x$  lorsque  $x$  s'approche de 0. Les fonctions d'activation doivent être non linéaires car il s'agit d'une caractéristique nécessaire pour que le réseau neural soit une approximation universelle [35]. Des fonctions d'activation continuellement différentiables sont nécessaires pour les méthodes

d'optimisation basées sur les gradients. Les fonctions d'activation monotone garantissent une surface d'erreur convexe d'un modèle à une seule couche [35]. Et si  $f(x) \approx x$  lorsque  $x$  s'approche de 0, les réseaux peuvent s'entraîner plus efficacement. Il existe plusieurs fonctions d'activation à choisir.

La fonction sigmoïde, la tangente hyperbolique et la fonction d'activation unité de rectification linéaire (ReLU) sont introduites et définies dans cette section. Certaines fonctions de classification et de régression utilisées dans la couche finale des réseaux de neurones sont également introduites.



**FIGURE 2-0 4: PLACEMENT DE LA FONCTION D'ACTIVATION DANS LE MODELE DE RESEAU NEURONAL.**

### 2.7.1 Sigmoïde :

La fonction sigmoïde logistique (Equation 2.1) est une fonction d'activation largement utilisée biologiquement plus plausible que la tangente hyperbolique [36]. L'une des raisons pour lesquelles la fonction sigmoïde est largement utilisée est le fait, la fonction sigmoïde est différente à chaque point.

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.1)$$

### 2.7.2 Tangente hyperbolique :

L'une des fonctions d'activation les plus populaires est la fonction tangente hyperbolique (Equation 2.2). L'entrée est une combinaison linéaire pondérée des entrées du nœud. Cette fonction fonctionne le plus efficacement sur les entrées dans la plage (0 ; 1), produisant des sorties en intervalle (-1 ;1)

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (2.2)$$

### 2.7.3 Unité de rectification linéaire (ReLU) :

La fonction de l'unité de Rectification Linéaire (Equation 2.3) est utilisée dans le but d'augmenter la non-linéarité du réseau. Les neurones de rectification sont considérés comme biologiquement plus plausibles que les neurones tangentiels sigmoïdes ou hyperboliques tangents. Ils bénéficient de leur simplicité, entraînant une formation plus rapide et des améliorations de performance dans des cas particuliers, et donc souvent utilisés dans les RNC. ReLU est donné par l'équation:

$$f(x) = \max(0, x) \quad (2.3)$$

### 2.7.4 Softmax :

La fonction d'activation de softmax (équation 2.4) est habituellement utilisée dans la dernière couche du réseau, lorsque le problème de classification n'est pas binaire mais contient plusieurs classes, la fonction softmax est généralement utilisée, mais elle est également viable pour le cas binaire. Il s'agit d'une généralisation de la fonction logistique et n'est qu'un autre nom pour un modèle de classification multinomiale quand on suppose qu'il n'existe aucune hiérarchie parmi les classes. La fonction softmax est agréable car elle donne une approximation de la probabilité qu'une classe soit correcte. L'approche la plus simple consiste à simplement choisir la classe avec la probabilité la plus élevée et ignorer le reste. Mais étant donné qu'il s'agit d'une fonction probabiliste, il peut également être utilisé pour un modèle générateur. Les scores de softmax (probabilités) sont calculés par la fonction de normalisation.

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{K=1}^N e^{x_K}} \quad (2.4)$$

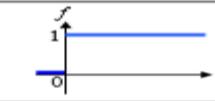
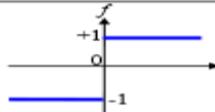
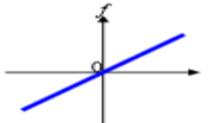
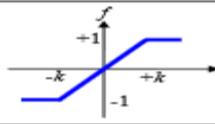
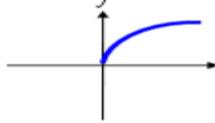
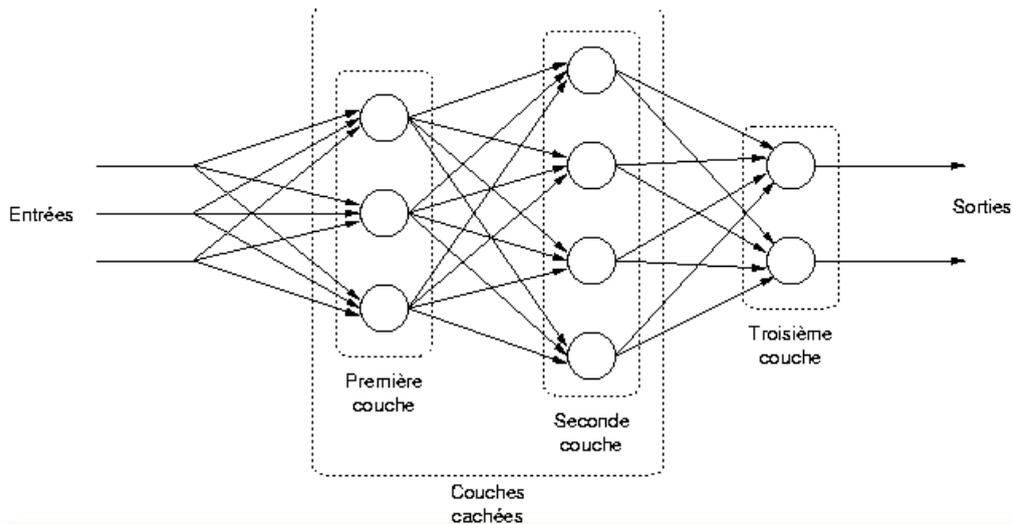
Catégorie	Type	Equation	Allure
Seuil	Binaire (Heaviside)	$f(x)=1$ si $x>0$ $f(x)=0$ si $x\leq 0$	
	Signe	$f(x)=1$ si $x>0$ $f(x)=-1$ si $x\leq 0$	
Linéaire	Identité	$f(x)=x$	
	Saturation	$f(k,x)=-1$ si $x<-1/k$ $f(k,x)=1$ si $x>=1/k$ $f(k,x)=kx$ sinon	
Sigmoïde	Positive (logistique)	$f(k,x) = \frac{1}{1+e^{-kx}}$	
	Symétrique (type tanch)	$f(k,x) = \frac{2}{1+e^{-kx}} - 1$	

FIGURE 2-0 5: FONCTION D'ACTIVATION

## 2.8 Perceptron multicouches :

Un perceptron simple couche est capable de classer les données uniquement linéairement séparables. Par exemple, il n'est pas capable de résoudre le problème XOR. Ce fait a été remarqué par Minsky et Papert dans leur célèbre livre "Perceptrons" en 1969. Le livre a contribué à la stagnation dans la recherche sur les réseaux des neurones pour un certain temps. On savait que le perceptron multicouche résoudrait des problèmes linéairement non séparables, même si des algorithmes efficaces pour l'apprentissage des PMC n'étaient pas connus à ce moment-là. Le premier algorithme réussi, appelé rétro-propagation, a été développé plusieurs années plus tard et depuis lors, le domaine des réseaux neuronaux s'est rapidement développé.

Un perceptron multicouche est un réseau qui consiste généralement en deux ou trois couches des neurones et d'une couche d'entrée supplémentaire. La couche d'entrée est comptée par certains auteurs comme une couche réseau distincte alors que par d'autres, elle ne l'est pas.



**FIGURE 2-0 6: SCHEMA D'UN PMC**

Les perceptrons sont disposés en  $k \geq 2$  couches. Considérons un réseau  $M$  avec des couches  $k$ . L'ensemble des neurones  $C$  est divisé en sous-ensembles disjoints mutuellement appelés couches  $L_1, \dots, L_k$ . Plus formellement il tient  $\forall i, j: 1 \leq i, j \leq k (L_i \neq \phi \wedge L_i \cap L_j \neq \phi) \Rightarrow i = j$ . Les couches de réseau sont empilées l'une sur l'autre,  $L_1$  étant la couche d'entrée,  $L_2, \dots, L_{K-1}$  étant les couches cachées et  $L_k$  étant la couche de sortie. Comme le montre la figure FIGURE 2-0 7, les bords sont tous orientés dans la direction de la couche d'entrée  $L_1$  vers la couche de sortie  $L_k$ . Chaque neurone dans la couche  $L_i$  est connecté à chaque neurone dans la couche  $L_{i+1}$ . En d'autres termes, toutes les couches voisines forment des graphiques bipartites complets.

La sortie du réseau est calculée séquentiellement, couche par couche. Nous commençons par la couche d'entrée en attribuant directement  $\vec{y}^0 = \vec{x}$ . Ensuite, le calcul se déroule en assignant l'entrée  $\vec{x}^i = \vec{y}^{i-1}$  pour la couche  $L_i$ . Les poids et la fonction d'activation sont donnés par le réseau, donc la sortie de chaque couche dépend uniquement de la sortie de la couche précédente. La sortie finale du réseau est ensuite produite comme  $\vec{y}^k$  dans la couche de sortie  $L_k$ .

### 2.8.1 Le processus d'apprentissage :

Pour entraîner un perceptron, c'est-à-dire apprendre les poids de connexion, nous allons chercher à minimiser l'erreur de prédiction sur le jeu d'entraînement. Nous pourrions faire ça de

manière explicite, cependant ce n'est vraisemblablement pas comme ça qu'un réseau de neurones biologiques fonctionne.

De plus, les réseaux de neurones biologiques sont supposés être plastiques, c'est-à-dire qu'ils s'adaptent constamment, en fonction des signaux qu'ils reçoivent. Ainsi, nous allons supposer que nos  $n$  observations  $x(1), x(2), \dots, x(n)$  ne sont pas observées simultanément mais *séquentiellement*, l'une après l'autre.

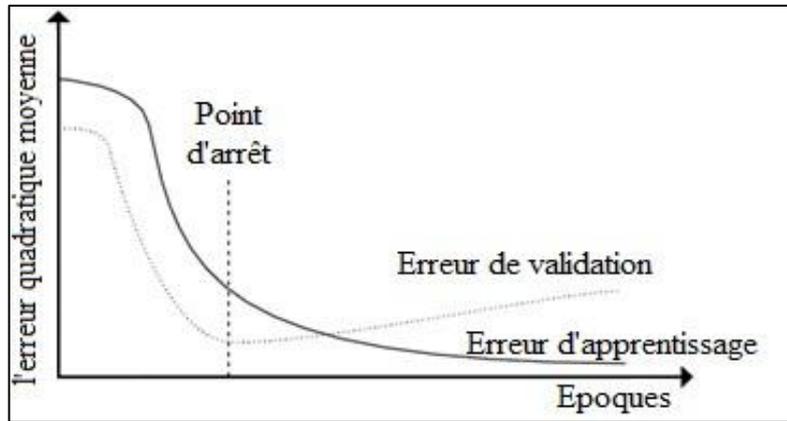
L'entraînement d'un perceptron est donc un processus **itératif**. Après chaque observation, nous allons ajuster les poids de connexion de sorte à réduire l'erreur de prédiction faite par le perceptron dans son état actuel. Pour cela, nous allons utiliser l'algorithme du gradient : le gradient nous donnant la direction de plus grande variation d'une fonction (dans notre cas, la fonction d'erreur), pour trouver le minimum de cette fonction il faut se déplacer dans la direction opposée au gradient. (Lorsque la fonction est minimisée localement, son gradient est égal à 0.)

Un problème courant rencontré dans le processus d'apprentissage est le sur-apprentissage. Il se produit généralement lorsque l'apprentissage se fait trop longtemps, et surtout lorsque l'ensemble d'apprentissage est trop faible pour représenter uniformément tous les types des modèles à partir du domaine des entrées possibles du réseau. Dans un tel cas, l'apprentissage peut ajuster le réseau aux fonctions aléatoires présentes dans les données d'apprentissage. L'accouplement est observé pendant le processus d'apprentissage, lorsque la performance prédictive du réseau s'améliore sur l'ensemble d'apprentissage, tout en aggravant les données de test précédemment non vues.

Pour lutter contre ce problème, les données marquées sont divisées en un ensemble d'apprentissage et un ensemble de validation. La principale raison pour laquelle on utilise l'ensemble de validation est qu'il montre les taux d'erreur sur les données indépendantes des données sur lesquelles nous nous entraînons. Une étude de Guyon suggère que le rapport optimal entre la taille de l'ensemble des données d'apprentissage et de validation dépend du nombre des classes reconnues et de la complexité des caractéristiques de classe.

Une estimation de la complexité des caractéristiques est cependant assez lourde. Un bon point de départ pour déterminer ce ratio est de mettre 80% des données disponibles dans l'ensemble d'apprentissage et 20% dans l'ensemble de validation. Une expérimentation

supplémentaire peut aider à se rapprocher du rapport optimal. Tout en apprenant, la performance de RNA est régulièrement examinée sur l'ensemble des données de validation. Lorsque les erreurs récupérées sur les données de validation atteignent un point d'arrêt, le processus d'apprentissage est arrêté (Figure2-07) et le réseau est considéré comme formé.



**FIGURE 2-0 8: L'EVOLUTION DE L'ERREUR D'APPRENTISSAGE PAR RAPPORT A L'ERREUR DE VALIDATION.**

### 2.8.2 Algorithme du gradient:

Cet algorithme permet de réaliser un apprentissage du réseau de neurones. On cherche à obtenir du réseau une réponse préétablie comme étant correcte. On dispose d'une base de connaissance de type entrée (p) - sortie attendue (d). On compare ensuite la sortie (a) à la sortie attendue. On introduit une fonction  $e = d - a$  qui est la fonction d'erreur que l'on va chercher à minimiser en modifiant les poids du réseau. Une fois les poids définis par cet algorithme à partir d'exemples connus de l'utilisateur, on va chercher à extrapoler le réseau, en lui fournissant des entrées inconnues.

Sachant évaluer les gradients, différents algorithmes, plus ou moins sophistiqués, sont implémentés. Le plus élémentaire est une utilisation itérative du gradient : en tout point de l'espace des paramètres, le vecteur gradient de Q pointe dans la direction de l'erreur croissante. Pour faire décroître Q il suffit donc de se déplacer en sens contraire. Il s'agit d'un algorithme itératif modifiant les poids de chaque neurone selon :

$$\beta_k^{(r+1)} = \beta_k^{(r)} - \tau \sum_{i=1}^n \frac{\partial Q_i}{\partial \beta_k^{(r)}}$$

$$\alpha_{kp}^{(r+1)} = \alpha_{kp}^{(r)} - \tau \sum_{i=1}^n \frac{\partial Q_i}{\partial \alpha_{kp}^{(r)}}$$

Le coefficient de proportionnalité  $\tau$  est appelé le taux d'apprentissage. Il peut être fixe, à déterminer par l'utilisateur, ou encore varier en cours d'exécution selon certaines heuristiques. Il paraît en effet intuitivement raisonnable que, grand au début pour aller plus vite, ce taux décroisse pour aboutir à un réglage plus fin au fur et à mesure que le système s'approche d'une solution.

Si l'espace mémoire est suffisant, une version accélérée de l'algorithme fait intervenir à chaque itération un ensemble (batch) d'observations pour moyenner les gradients et mises à jour des poids. Bien d'autres méthodes d'optimisation ont été adaptées à l'apprentissage d'un réseau : méthodes du gradient avec second ordre utilisant une approximation itérative de la matrice hessienne (algorithme BFGS, de LevenbergMarquardt) ou encore une évaluation implicite de cette matrice par la méthode dite du gradient conjugué.

La littérature sur le sujet propose quantités de recettes destinées à améliorer la vitesse de convergence de l'algorithme ou bien lui éviter de rester collé à une solution locale défavorable. D'autres heuristiques proposent d'ajouter un terme d'inertie afin d'éviter des oscillations de l'algorithme. D'autres algorithmes encore sont des versions adaptatives. Lorsque de nouvelles observations sont proposées une à une au réseau.

Dans ce dernier type d'algorithme, des propriétés de dynamique markovienne (processus ergodique convergeant vers la mesure stationnaire) impliquent une convergence presque sûre : la probabilité d'atteindre une précision fixée a priori tend vers 1 lorsque la taille de l'échantillon d'apprentissage tend vers l'infini.

On pourra se reporter à l'abondante littérature sur le sujet pour obtenir des précisions sur les algorithmes d'apprentissage et leurs nombreuses variantes. Il est important de rappeler la liste des choix qui sont laissés à l'utilisateur. En effet, même si les logiciels proposent des valeurs par défaut, il est fréquent que cet algorithme connaisse quelques soucis de convergence.

---

**Algorithm 1** Rétro propagation élémentaire du gradient

---

Initialisation des poids  $b_{jkl}$  par tirage aléatoire selon une loi uniforme sur  $[0, 1]$ .

Normaliser dans  $[0, 1]$  les données d'apprentissage.

**while**  $Q > \text{errmax}$  ou  $\text{niter} < \text{itermax}$  **do**

Ranger la base d'apprentissage dans un nouvel ordre aléatoire.

**for** chaque élément  $i = 1, \dots, n$  de la base **do**

Calculer  $\varepsilon(i) = y_i - f(x_i^1, \dots, x_i^p; (b)(i-1))$  en propageant les entrées vers l'avant.

L'erreur est rétro-propagée dans les différentes couches afin d'affecter à chaque entrée une responsabilité dans l'erreur globale.

Mise à jour de chaque poids  $b_{jkl}(i) = b_{jkl}(i-1) + \Delta b_{jkl}(i)$

**end for**

**end while**

---

Le but de l'algorithme de rétro-propagation est d'ajuster les poids synaptiques des neurones, de sorte que le réseau génère la sortie souhaitée. L'algorithme décrit le processus d'apprentissage. Le résultat de cet algorithme est un réseau neuronal configuré pour minimiser l'erreur lors de la résolution d'un problème.

L'apprentissage doit être effectué sur les données marquées et donc supervisée. Avant que l'algorithme ne commence, les poids doivent être initialisés à certaines valeurs. L'initialisation ne fait pas partie de la spécification de l'algorithme, car il peut y avoir des approches différentes, la plus courante étant l'initialisation la plus banale - aléatoire. Ensuite, l'algorithme d'apprentissage commence.

## 2.9 Réseaux des neurones convolutionnels (RNC) :

Nous allons présenter dans cette section les différents types de couches classiquement utilisés dans les CNN. Le but est d'exprimer leur connectivité indépendamment du modèle de neurone. Les réseaux de neurones convolutionnels sont à ce jour les modèles les plus performants pour classer des images.

Désignés par l'acronyme CNN, de l'anglais Convolutional Neural Network, ils comportent deux parties bien distinctes. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu].

La première partie d'un CNN est la partie convolutive à proprement parler. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN. [37]

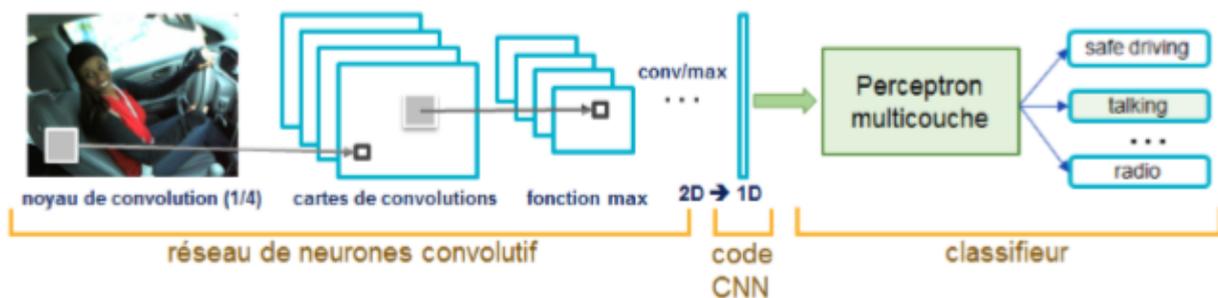


FIGURE 2-0 9: ARCHITECTURE STANDARD D'UN RESEAU DE NEURONE CONVOLUTIONNEL.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image.

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

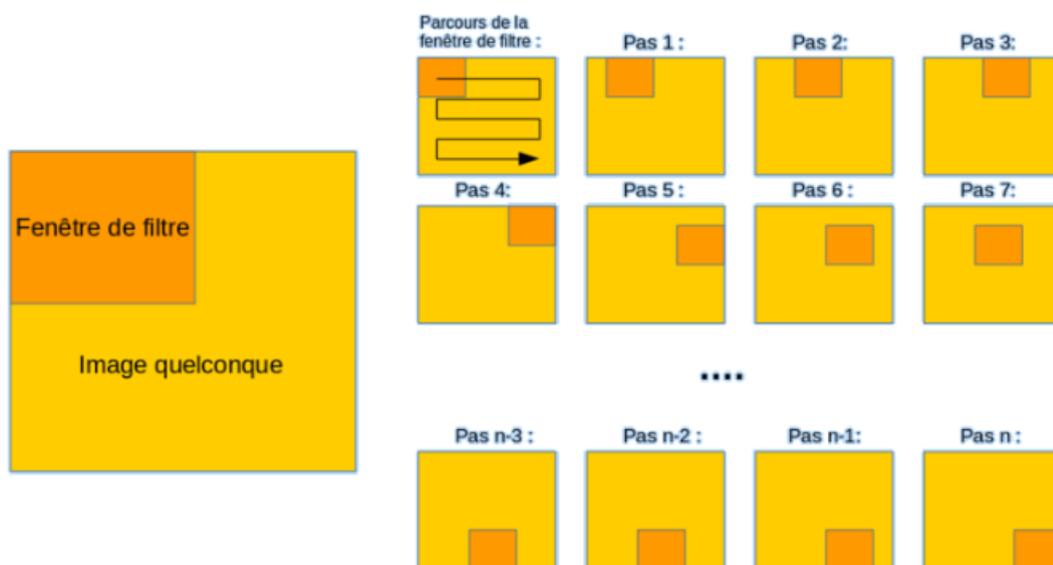
## 2.9.1 Architecture de réseaux de neurone convolutionnel:

### 2.9.1.1 la convolution :

La convolution est un outil mathématique simple qui est très largement utilisé pour le traitement d'image, ce qui explique que les réseaux de neurones à convolution soient particulièrement bien adaptés à la reconnaissance d'image.

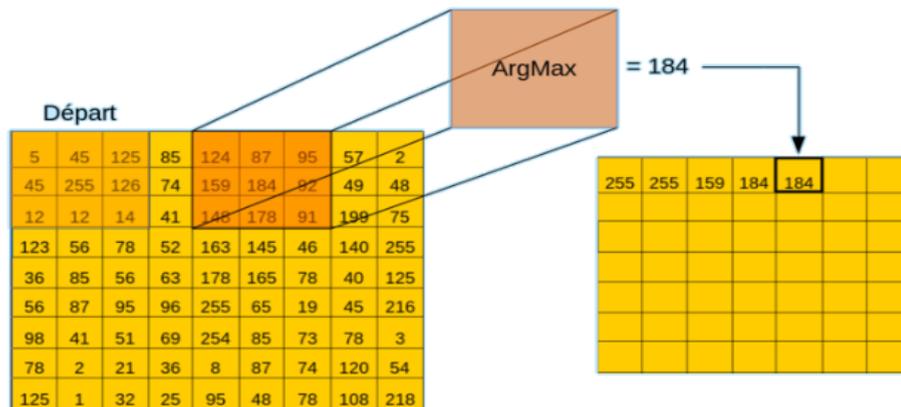
La convolution agit comme un filtrage. On définit une taille de fenêtre qui va se balader à travers toute l'image (rappelez-vous qu'une image peut être vue comme étant un tableau).

Au tout début de la convolution, la fenêtre sera positionnée tout en haut à gauche de l'image puis elle va se décaler d'un certain nombre de cases (c'est ce que l'on appelle le pas) vers la droite et lorsqu'elle arrivera au bout de l'image, elle se décalera d'un pas vers le bas ainsi de suite jusqu'à ce que le filtre est parcourue la totalité de l'image :



**FIGURE 2-0 10: SCHEMA DU PARCOURS DE LA FENETRE DE FILTRE SUR L'IMAGE.**

Le but est de se servir des valeurs présentes dans le filtre à chaque pas. Par exemple si l'on définit une fenêtre 3 par 3, cela représentera 9 cases du tableau (c'est à dire 9 pixels). La convolution va effectuer une opération avec ces 9 pixels. Il peut s'agir de n'importe quelle opération, par exemple on extrait la valeur la plus grande (soit le pixel avec la plus grande valeur).



**FIGURE 2-0 11 : EXEMPLE D'UNE CONVOLUTION DONT LA CONFIGURATION EST : OPERATION = ARGUMENT MAXIMALE, PAS HORIZONTALE = 1 PIXEL, PAS VERTICAL = 1 PIXEL.**

On fait glisser la **fenêtre** en orange et à chaque **pas** on récupère la valeur la plus grande parmi les 9 valeurs de **pixels**.

On remarque que la sortie de la **convolution**, que l'on peut appeler «**carte de caractéristiques**», à des dimensions plus petites que celle de l'**image en entrée**.

### 2.9.1.2 Couche de mise en commun ( pooling ) :

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs *feature maps*, et applique à chacune d'entre elles l'opération de *pooling*.

L'opération de pooling (ou sub-sampling) consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

Pour cela, on découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille 2 × 2 pixels qui ne se chevauchent pas, ou des cellules de taille 3 × 3 pixels, distantes les unes des autres d'un pas de 2 pixels (qui se chevauchent donc). On obtient en sortie le même nombre de *feature maps* qu'en entrée, mais celles-ci sont bien.

La couche de *pooling* permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage.

Il est courant d'insérer périodiquement une couche Pooling entre les couches Conv successives dans une architecture ConvNet. Sa fonction est de réduire progressivement la taille spatiale de

la représentation afin de réduire la quantité de paramètres et de calculs dans le réseau, et donc de contrôler également le sur-ajustement. La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne spatialement, en utilisant l'opération MAX.

Ainsi, la couche de *pooling* rend le réseau moins sensible à la position des *features* : le fait qu'une *feature* se situe un peu plus en haut ou en bas, ou même qu'elle ait une orientation légèrement différente ne devrait pas provoquer un changement radical dans la classification de l'image.

### 2.9.1.3 la couche ReLU :

Pour améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. Dans ce cadre on trouve 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. Souvent, la correction Relu est préférable, mais il existe d'autre forme

- La correction par *tangente hyperbolique*  $f(x)=\tanh(x)$ ,
- La correction par la *tangente hyperbolique saturante*:  $f(x)=|\tanh(x)|$ ,
- La correction par la *fonction sigmoïde*  $f(x)=\frac{1}{1+e^{-x}}$ .  $f(x)=\frac{1}{1+e^{-x}}$

### 2.9.1.4 Couche fully-connected :

La couche fully-connected constitue toujours la dernière couche d'un réseau de neurones, convolutif 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.

La couche *fully-connected* permet de classifier l'image en entrée du réseau : elle renvoie un vecteur de taille N, où N est le nombre de classes dans notre problème de classification d'images. Chaque élément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe.

Par exemple, si le problème consiste à distinguer les chats des chiens, le vecteur final sera de taille 2 : le premier élément (respectivement, le deuxième) donne la probabilité d'appartenir à la classe "chat" (respectivement "chien"). Ainsi, le vecteur [0.90.1] signifie que l'image a 90% de chances de représenter un chat.

Chaque valeur du tableau en entrée "vote" en faveur d'une classe. Les votes n'ont pas tous la même importance : la couche leur accorde des poids qui dépendent de l'élément du tableau et de la classe.

Pour calculer les probabilités, la couche *fully-connected* multiplie donc chaque élément en entrée par un poids, fait la somme, puis applique une fonction d'activation (logistique si  $N=2$ , *softmax* si  $N>2$ ) :

Il est possible de remplacer les couches entièrement connectées d'un CNN par des couches convolutionnelles, ce qui le rend *entièrement convolutionnel*. All-convolutional network est une bonne idée je vous invite à le découvrir.

## 2.10 Conclusion :

Dans ce chapitre, il était question des réseaux de neurones et particulièrement des réseaux de neurones convolutionnels. Nous avons détaillé leur conception et leur utilisation dans le traitement de la reconnaissance des caractères arabes. Nous allons procéder dans le chapitre qui suit à l'application de ces réseaux de neurone convolutionnels dans la reconnaissance des caractères arabes.

Ces réseaux sont capables d'extraire des caractéristiques d'images présentées en entrée et de classifier ces caractéristiques. Ils implémentent l'idée de partage des poids permettant ainsi de réduire d'une part le nombre de paramètres libres de l'architecture,

d'autre part de réduire les temps de calcul, l'espace mémoire nécessaire, et améliorer ainsi les capacités de généralisation du réseau.

## **Chapitre 3.**

# **Conception de l'architecture CNN pour la reconnaissance des caractères arabes**

### **3.1 Introduction :**

Les systèmes de reconnaissance des caractères arabes manuscrits font face à plusieurs défis, y compris la variation illimitée de l'écriture humaine et des grandes bases des données publiques. Dans ce chapitre, nous utilisons les réseaux des neurones convolutionnels qui peut être appliqué efficacement à la reconnaissance des caractères arabes manuscrits. Donc on va voir dans ce chapitre l'utilisation du RNC dans un mode de reconnaissance de caractères.

### **3.2 Conception de l'architecture CNN:**

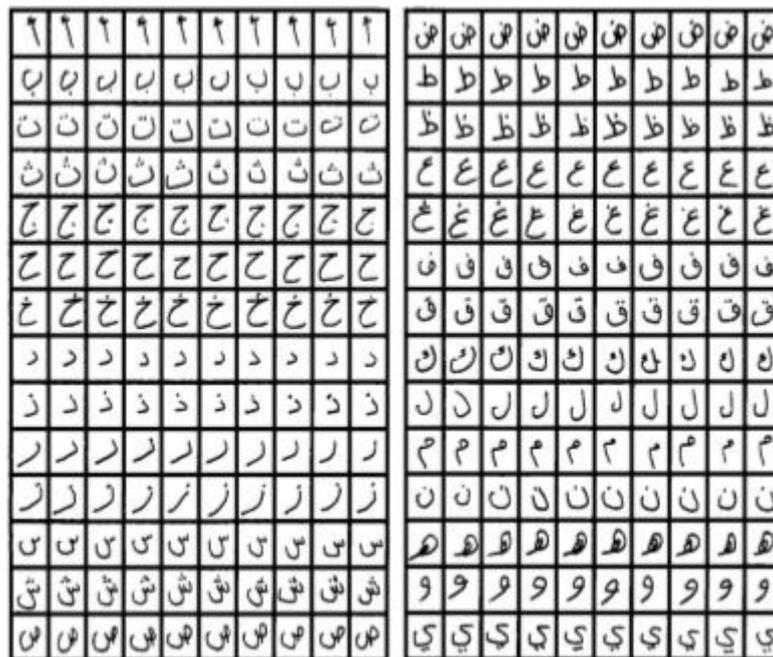
Le réseau de neurone convolutionnel nécessite une grande donnée de formation des images à des caractères manuscrits pour obtenir un bon résultat. Les données disponibles pour la formation sont divisées en deux ensembles différents: Ensemble d'apprentissage et Ensemble de validation. Il ne devrait pas y avoir de chevauchement entre ces deux ensembles des données afin d'améliorer la capacité de généralisation d'un réseau neuronal. Cette technique s'appelle la validation croisée [37]. Les performances réelles d'un réseau ne sont révélées que lorsque le réseau est testé avec des données de test pour mesurer le rendement du réseau sur les données qui n'ont pas été vues pendant l'apprentissage. Le test est conçu pour accéder à la capacité de généralisation de réseau. Une bonne généralisation signifie que le réseau fonctionne correctement sur des données similaires, mais différentes des données d'apprentissage.

### **3.3 Logiciels et bibliothèques Utilisés dans l'implémentation :**

#### **3.3.1 Dataset :**

Les réseaux de neurones convolutifs ont besoin d'une grande quantité de données d'entraînement sur les images de caractères manuscrits pour obtenir un bon résultat. Nous avons donc utilisé le jeu de données qui est composé de 16 800 caractères écrits par 60 participants, la tranche d'âge va de 19 à 40 ans et 90% des participants sont à droite. Chaque

participant a écrit chaque caractère (de "alef" à "yeh") dix fois sur deux formulaires. La base de données est divisée en deux ensembles: un ensemble d'apprentissage (13 440 caractères pour 480 images par classe) et un ensemble de test (3 360 caractères pour 120 images par classe). L'ordre dans lequel les rédacteurs inclus dans l'ensemble de tests sont randomisés et vérifiés afin de s'assurer que les rédacteurs de l'ensemble de tests ne proviennent pas d'une seule institution (afin de garantir la variabilité de l'ensemble de tests).



(a)

(b)

**FIGURE 3-0 1: EXEMPLE DE DATASET DES CARACTERES ARABES**

Tout d'abord on fixe notre répertoire de travail pour que python reconnaisse les dossiers automatiquement.

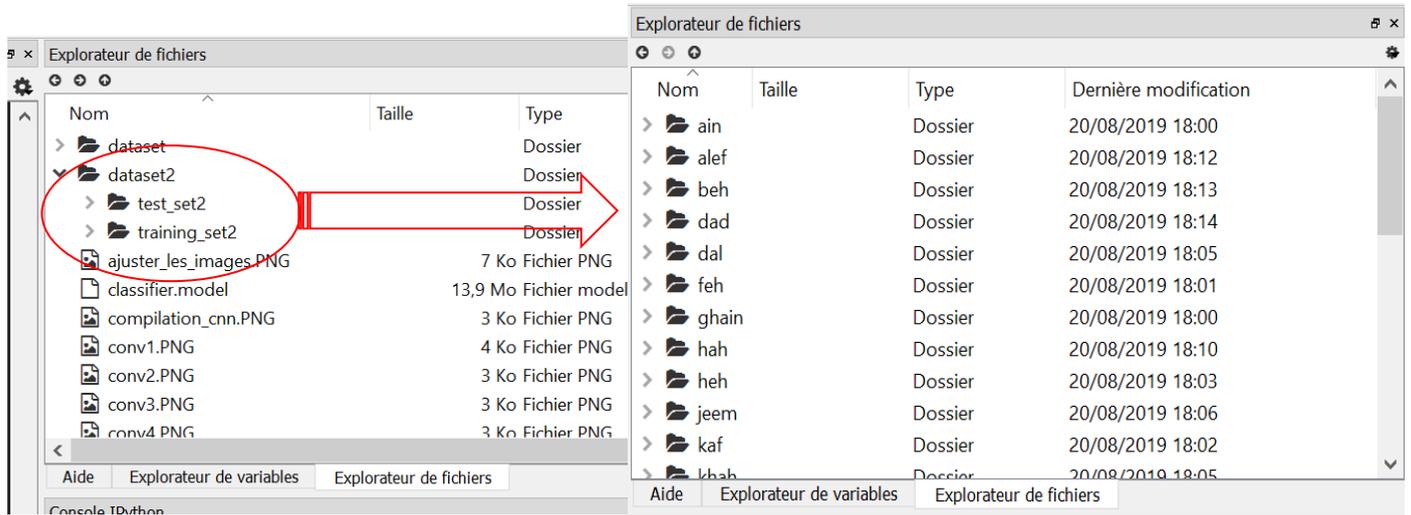
On commence par structurer nos données.

Pour lire les données il existe une solution qui est la librairie Keras qui contient des outils pour extraire les images facilement.

Peu importe les images avec Keras ce qu'il faut faire c'est préparer les dossiers correctement avec la bonne structure.

D'abord, on sépare le jeu de test et le jeu d'entraînement en deux dossiers différents comme le montre l'image, ensuite on sépare encore chaque dossier pour chaque caractère et à l'intérieur des dossiers on a nos images.

Pour finaliser on différencie chaque fichier (image) par des noms par exemple (ba1 - ba2) c-a-d un dossier par classe, comme le montre la Figure 3-



### 3.3.2 TensorFlow :

TensorFlow est une bibliothèque open source développée par Google pour l'apprentissage par la machine et la recherche sur les réseaux de neurones profonds (deep neural network).

Mais le système est suffisamment général pour pouvoir s'appliquer à une grande variété d'autres domaines également, il est aussi multiplateformes. Il fonctionne sur Presque tout, les processeurs graphiques GPUs et les processeurs CPUs , y compris les plates-formes mobiles et intégrées, et même les unités de traitement de tenseurs (TPU), qui sont du matériel spécialisé permettant d'effectuer des calculs de tenseurs.

L'apprentissage automatique peut devenir complexe rapidement et les modèles d'apprentissage en profondeur peuvent devenir volumineux. Pour de nombreux modèles de graphiques, vous avez besoin d'une formation distribuée pour pouvoir effectuer une itération dans un délai raisonnable. Et vous souhaitez généralement que les modèles que vous développez soient déployés sur plusieurs plates-formes.

Avec la version actuelle de TensorFlow, vous écrivez du code pour créer un graphe de calcul, puis vous l'exécutez. Le graphique est une structure de données qui décrit complètement le calcul que vous souhaitez effectuer.

Cela présente de nombreux avantages: Il est portable, car le graphique peut être exécuté immédiatement ou enregistré pour une utilisation ultérieure, et il peut être exécuté sur plusieurs plates-formes: CPU, GPU, TPU, mobile, embarqué.

En outre, il peut être déployé en production sans avoir à dépendre du code qui a construit le graphe, mais uniquement du temps d'exécution nécessaire à son exécution.

Il est transformable et optimisable, car le graphique peut être transformé pour produire une version plus optimale pour une plate-forme donnée.

En outre, des optimisations de mémoire ou de calcul peuvent être effectuées et des compromis entre elles, Ceci est utile, par exemple, pour prendre en charge une inférence mobile plus rapide après une formation sur des machines plus grandes, Prise en charge de l'exécution distribuée.

Les API de haut niveau ( Keras ) de TensorFlow, associées aux graphes de calcul, permettent un environnement de développement riche et flexible et des capacités de production puissantes dans le même cadre.

### **3.3.3 Keras :**

Bien que les réseaux de neurones profonds fassent fureur, la complexité des principaux cadres constitue un obstacle à leur utilisation par les développeurs novices en apprentissage automatique.

Plusieurs propositions d'améliorations et de simplification des API de haut niveau pour la construction de modèles de réseaux neuronaux ont été proposées, Toutes ont tendance à ressembler de loin, mais montrent des différences à l'examen.

Keras est l'un des principaux API de réseaux de neurones de haut niveau, Il est écrit en Python et prend en charge plusieurs moteurs de calcul de réseau neuronal.

Le projet TensorFlow a adopté Keras comme API de haut niveau .

Keras a été créé pour être convivial, modulaire, facile à étendre et pour travailler avec Python. L'API a été «conçu pour les êtres humains et non pour les machines» et «suit les meilleures pratiques pour réduire la charge cognitive».

Les couches neuronales, les fonctions de coût, les optimiseurs, les schémas d'initialisation, les fonctions d'activation et les schémas de régularisation sont tous des modules autonomes que vous pouvez combiner pour créer de nouveaux modèles.

Les nouveaux modules sont simples à ajouter, en tant que nouvelles classes et fonctions, les modèles sont définis dans le code Python et non dans des fichiers de configuration de modèle distincts.

### **3.3.4 Module keras utiliser :**

Les modules de Keras qu'on a utiliser sont :

-Keras.models :

Sequential

-Keras.layers :

Conv2D

MaxPooling2D

Flatten

Dense

Dropout

-Keras.preprocessing.image :

ImageDataGenerator

image

### **3.3.5 Python :**

Python est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple, facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes. Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement.

### 3.4 Configuration Utilisé dans l'implémentation :

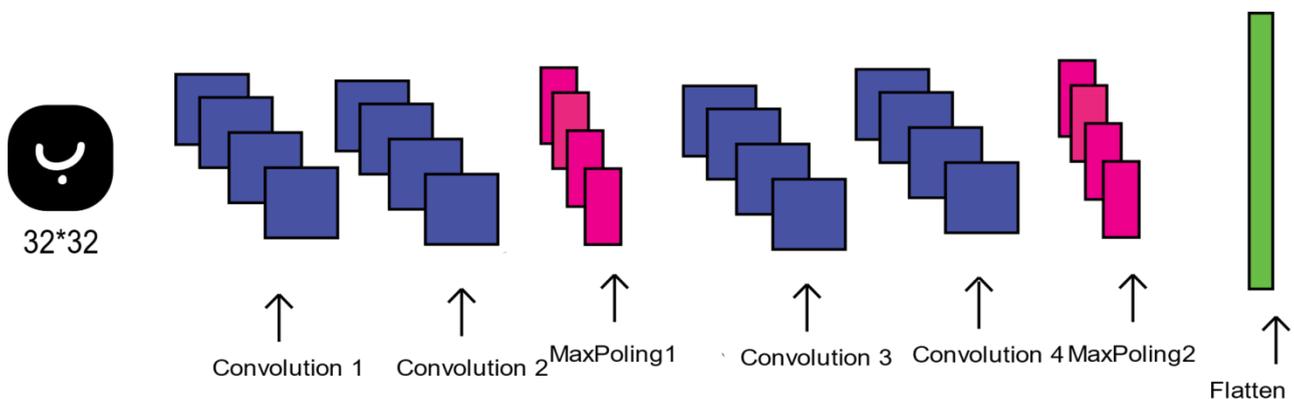
La configuration du matériel utilisé dans notre implémentation est :

- PC portable Asus i7 CPU 2.40 GHZ
- Carte graphique Nvidia GeForce GTX 950M
- RAM de taille 8 GO
- Disque dur de taille 500 GO SSD
- Système d'exploitation windows 10

### 3.5 Architecture du réseau :

L'architecture de notre réseau pour la reconnaissance des caractères arabes manuscrit est basé sur le réseau de neurones convolutionnel dont la structure est donné ci-dessous

Figure 3-02



**FIGURE 3-0 2:ARCHITECTURE DE NOTRE RNC**

Nous présentons dans la figure 3-02 présente l’architecture de notre RNC qui est composé de quatre couches de convolution et deux couches de maxpooling et trois couches de fully connected.

L’image en entrée est de taille 32\*32, l’image passe d’abord à la première couche de convolution. Cette couche est composée de 32 filtres de taille 5\*5, Chacune de nos couches de convolution est suivie d’une fonction d’activation ReLU cette fonction force les neurones à retourner des valeurs positives, après cette convolution 32 features maps de taille 32\*32 seront créés.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 32)	2432
conv2d_2 (Conv2D)	(None, 32, 32, 32)	25632
max_pooling2d_1 (MaxPooling2)	(None, 16, 16, 32)	0
dropout_1 (Dropout)	(None, 16, 16, 32)	0
conv2d_3 (Conv2D)	(None, 16, 16, 64)	18496
conv2d_4 (Conv2D)	(None, 16, 16, 64)	36928
max_pooling2d_2 (MaxPooling2)	(None, 8, 8, 64)	0
dropout_2 (Dropout)	(None, 8, 8, 64)	0
flatten_1 (Flatten)	(None, 4096)	0
dense_1 (Dense)	(None, 256)	1048832
dense_2 (Dense)	(None, 256)	65792
dropout_3 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 28)	7196
=====		
Total params: 1,205,308		
Trainable params: 1,205,308		
Non-trainable params: 0		

**FIGURE 3-0 3: CONFIGURATION DU NOTRE MODELE RNC**

Les 32 feature maps qui sont obtenus auparavant ils sont donnés en entrée de la deuxième couche de convolution qui est composée aussi de 32 filtres, une fonction d'activation RELU est appliquée sur la couche de convolution, ensuite on applique Maxpooling pour réduire la taille de l'image ainsi la quantité de paramètres et de calcul. À la sortie de cette couche, nous aurons 32 feature maps de taille 16\*16.

On répète la même chose avec les couches de convolutions trois, quatre, ces couches sont composées de 64 filtres, la fonction d'activation ReLU est appliquée toujours sur chaque convolution. Une couche de Maxpooling est appliquée après la couche de convolution quatre. À la sortie de cette couche, nous aurons 64 feature maps de taille 8\*8. Le vecteur de caractéristiques issu des convolutions a une dimension de 4096.

Après ces quatre couches de convolution, nous utilisons un réseau de neurones composé de trois couches fully connected. Les deux premières couches ont chacune 256 neurones où la fonction d'activation utilisée est le ReLU, et la troisième couche est un softmax qui permet de calculer la distribution de probabilité des 28 classes (nombre de classe dans la base d'image).

#### **Code Source de notre model proposé :**

```
filtre = 5          #taille de filtre
strides=1          #le pas de convolution
convolution1=32    # nombre de filtre (32 features maps)
convolution2=32    # nombre de filtre (32 features maps)
convolution3 = 64  # nombre de filtre (64 features maps)
convolution4 = 64  # nombre de filtre (64 features maps)
pool_size=2        # taille de pool
dropout1 = 0.25    #appliquer un dropout avec probabilité de 25 %
dropout2=0.50     #appliquer un dropout avec probabilité de 50 %
couche1= 256      #créer une première couche de 256 neurones
```

couchecache2=256 # créer une deuxième couche de 256 neurones

```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

training_set = train_datagen.flow_from_directory('dataset2\\training_set2',
                                                target_size = (32, 32),
                                                batch_size = 32,
                                                class_mode = 'categorical')

test_set = test_datagen.flow_from_directory('dataset2\\test_set2',
                                            target_size = (32, 32),
                                            batch_size = 32,
                                            class_mode = 'categorical')
```

# charger et ajuster la base donnée d'image

#13440 images d'apprentissages

#3360 images de test

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

# Compiler le model.

```
classfier.fit_generator(training_set,
                       steps_per_epoch = 420,
                       epochs = 30,
                       validation_data = test_set,
                       validation_steps = 105)
```

Lancement de l'apprentissage du modèle sur 30 epoch

num\_classes = 28 classes

image d'entrée de taille 3\*32\*32

```
classfier.add(Conv2D(32, (5, 5),padding='same', input_shape = (32, 32, 3), activation = 'relu'))
```

1 : Cette commande permet de créer 32 features maps en utilisant un filtre de taille 5 par 5 pixels et avec un mode de bordure égal à la taille de l'image précédente et une fonction d'activation de type RELU.

```
classfier.add(Conv2D(32, (5, 5),padding='same', activation = 'relu'))
```

2 : Cette commande permet de créer 32 features maps en utilisant un filtre de taille 5 par 5 pixels et avec un mode de bordure égal à la taille de l'image précédente et une fonction d'activation de type RELU.

```
classfier.add(MaxPooling2D(pool_size = (2, 2)))
```

3 : Cette ligne de commande permet de réduire la taille de l'image, La méthode Max Pooling est utilisée et la taille de l'image sera divisée sur 2.

```
classfier.add(Conv2D(64, (3, 3),padding='same', activation = 'relu'))
```

4 : Cette commande permet de créer 64 features maps en utilisant un filtre de taille 3 par 3 pixels et avec un mode de bordure égal à la taille de l'image précédente et une fonction d'activation de type RELU.

```
classfier.add(Conv2D(64, (3, 3),padding='same', activation = 'relu'))
```

5 : Cette commande permet de créer 64 features maps en utilisant un filtre de taille 3 par 3 pixels et avec un mode de bordure égal à la taille de l'image précédente et une fonction d'activation de type RELU.

```
classfier.add(MaxPooling2D(pool_size = (2, 2) , strides=(2,2)))
```

6 : Cette ligne de commande permet de réduire la taille de l'image, La méthode Max Pooling est utilisée et la taille de l'image sera divisée sur 2.

```
classfier.add(Flatten())
```

7 : Cette commande permet de créer un seul vecteur 1D puis connecter avec la première couche cachée pour commencer la classification.

```
classfier.add(Dense(units = 256, input_dim = 1024, activation = 'relu'))
```

8 : Cette commande permet de créer une couche cachée avec une taille de 256 neurones, la fonction RELU est utilisé comme fonction d'activation.

```
classfier.add(Dropout(0.25))
```

9: Pour ne pas tomber dans le problème de sur apprentissage il faut utiliser dropout elle est très efficace pour les réseaux de neurones pour le régulariser et elle n'a besoin que de deux paramètres pour être défini, dont :

- Le paramètre type avec pour valeur 'dropout'
- Le paramètre rate avec pour valeur 0.25 et 0.5.

```
classfier.add(Dense(units = 256, activation = "relu"))
```

10 : Cette commande permet de créer une couche cachée avec une taille de 1024 neurones, la fonction RELU est utilisée comme fonction d'activation.

```
classfier.add(Dropout(0.5))
```

11 : Appliquer dropout sur la couche cachée 2

```
classfier.add(Dense(28, activation = "softmax"))
```

12 : Cette commande permet de créer une couche de sortie composée de 28 neurones (nombre de classes) la fonction softmax est utilisée pour calculer la probabilité de chaque classe.

### 3.6 Résultat :

Après l'analyse des résultats obtenus, On constate les remarques suivantes :

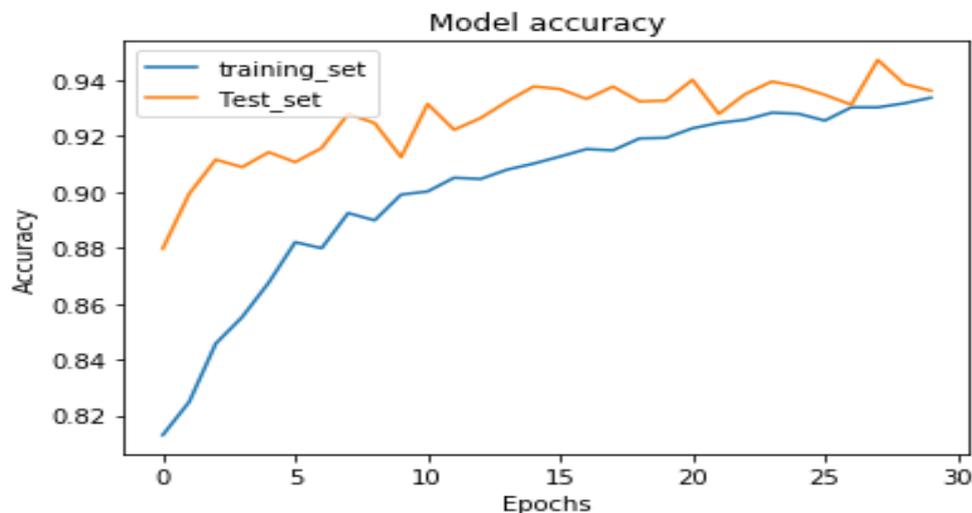
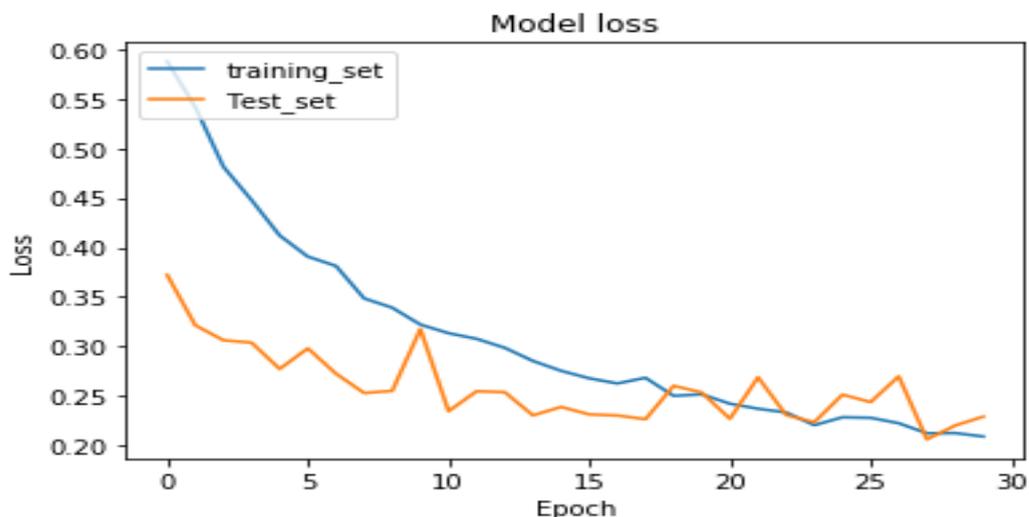


FIGURE 3-0 4: TAUX DE PRECISION DU MODELE.



**FIGURE 3-0 5: TAUX D'ERREUR DU MODELE.**

D'après la Figure 3-04 La précision de l'apprentissage et de test augmente avec le nombre d'époque, ceci reflète qu'à chaque époque le modèle apprend plus d'informations. Si la précision est diminuée alors on aura besoin de plus d'information pour faire apprendre notre modèle et par conséquent on doit augmenter le nombre d'époque.

De même, l'erreur d'apprentissage et de validation diminue avec le nombre d'époque.

Dans cette section, la performance de CNN a été étudiée pour la formation et la reconnaissance des caractères arabes. Les expériences sont réalisées dans un environnement de programmation python avec la bibliothèque TensorFlow et API keras.. Au début, pour évaluer les performances de CNN sur les caractères arabes, une approche de formation progressive a été utilisée pour l'approche proposée avec le mode mini-batch. Le taux de classification des données d'apprentissage a atteint 94% pour les époques de 25 à 30. Notre approche est conçue pour 30 époques, mais à partir de 25, le CNN affiche un taux d'erreur de classification faible.

### 3.7 Conclusion :

Nous avons montré dans ce chapitre l'utilisation des réseaux de neurone pour la reconnaissance des caractères arabe manuscrits.

Les résultats obtenus sont très encourageants avec un taux d'exactitudes de classification de 94% et ce malgré la complexité de chaque caractère et la similitude entre les caractères.

## Conclusion générale

La reconnaissance automatique de l'écriture Arabe manuscrite demeure un domaine de recherche et d'exploration de méthodes et techniques modernes pour la conception de procédures, logiciels modernes optimisés à l'instar des autres langues, et ce en dépit de la portée de cette Langue au travers le monde (400 millions de locuteurs /internet).

Nous constatons que les avancées technologiques n'ont pas pour autant fait avancer les recherches sur cette problématique qui nécessite plus d'effort et peut être plus de volonté.

Notre étude a consisté d'une part, à comprendre la problématique, d'analyser les différentes solutions proposées par les chercheurs pour la reconnaissance des caractères arabes manuscrits, d'explorer d'autres part les techniques et méthodes de traitement utilisées pour cette reconnaissance et de concevoir un système relativement fiable, optimisé pouvant contribuer à mieux cerner les difficultés relatives à la reconnaissance de l'écriture arabe compte tenu de la spécificité de ses caractères, tel qu'expliqué dans le document (caractère, au début d'un mot, au milieu du mot, à la fin d'un mot).

Nous rappelons les différentes méthodes et processus utilisés pour atteindre notre objectif avec une marge d'erreur acceptable, en référence aux moyens et disponibilité d'information.

La première phase de notre étude s'est articulée sur la compréhension des caractéristiques des caractères arabes en prenant connaissance des différentes méthodes de classification et processus de reconnaissance évoquées dans les différentes expériences y afférentes.

Nos différentes recherches et tests ont conclu en la méthode la plus efficace soit : le réseau des neurones artificiels.

Nous avons détaillé la méthodologie d'utilisation de ce processus en mettant en exergue les avantages liés à la reconnaissance de l'écriture arabe manuscrite ; cependant tel qu'indiqué sur les recherches ayant trait à notre sujet, il est recommandé d'examiner et de déterminer les règles et algorithmes de recherche sur les réseaux des neurones convolutionnels, considérés assez importants pour le domaine de reconnaissance des caractères arabes.

Partant de ce constat, nous avons expliqué l'architecture de ce réseau et conçu la logique de notre logiciel et procédé aux tests nécessaires et à différentes simulations de cas pour

aboutir à des résultats satisfaisants (objet de la démonstration) ; néanmoins , la comparaison de fiabilité entre la reconnaissance des caractères est assez mitigée , du fait de la spécificité des caractères arabes , tel que précisé plus haut :

Les lettres sont 28 et ils prennent des positions différentes

La base de données des caractères est (13 440 = training +3 360 = test)

Considérant ces observations et les marges d'erreurs susceptibles de fausser les résultats, nous pouvons tirer les enseignements suivants :

Le réseau de neurone convolutionnel est basé sur une importante base de données pour l'expérimentation et l'apprentissage qui nécessite des techniques de pointe et des procédés efficaces pour être conforme à son exploitation ; il reste certain que dans un avenir proche ,les évolutions techniques et scientifiques vont découdre de ces lacunes et permettre un niveau de reconnaissance des caractères arabes assez élevé .

Ouverture vers l'avenir :

Ces recherches et études , très intéressantes ,réalisées pour ce mémoire nous laissent envisager des perspectives de travaux à accomplir et des domaines a explorer pour mieux appréhender le processus le plus adéquat , optimisé et rationnel pour manager cette reconnaissance des caractères arabes .

Nous affirmons que sommes aptes à développer davantage cette application par un apport qualitatif et quantitatif des bases de données .Il faut aussi s'accorder du temps pour permettre un bon apprentissage du réseau de neurones convolutionnel. Sans oublier un matériel informatique performant.

## Bibliographie

- [1] B. Al-Badr , S.A. Mahmoud : « Survey and bibliography of Arabic optical text recognition ». *Signal processing* , vol. 41, pp. 49-77,1995.
- [2] P.M. Lallican, C. Viarp-Gaudin, S. Knerr : « From off-line to on-line handwriting recognition ». *Proc. 7th workshop on frontiers in handwriting recognition*, pp. 303-312, Amsterdam 2000.
- [3] I.R. Tsang : «Pattern recognition and complex systems». Thèse de doctorat, université d'Anterwerpen, 2000.
- [4] J. Trenkle, A. Gillies, S.Schlosser : « An off-line Arabic recognition system for machine printed documents ». *Proc. Of the symposium on document image understanding technology (SDIUT'97)*, pp. 155-161 1997.
- [5] 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.
- [6] Salima Nebti, « Reconnaissance de Caractères Manuscrits par Intelligence Collective », Thèse de doctorat Université Ferhat Abass-Sétif, Algeria, 2013.
- [7] L. M. Lorigo and V. Govindaraju, « Offline Arabic Handwriting Recognition », *A Survey*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, (2006), pp. 712-725.
- [8] H. A. Al-Hamad and R. Abu Zitar, « Development of an Efficient Neural-Based Segmentation Technique for Arabic Handwriting Recognition », *Pattern Recognition*, vol. 43, no. 8, (2010), pp. 2773– 2800.
- [9] R. Plamodon and S. N. Srihari, « Online and Offline Handwriting Recognition », *A Comprehensive Survey*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, (2000), pp. 6384.
- [10] H. El-Abed and V. Margner, « Comparison of Different Preprocessing and Feature Extraction Methods for Offline Recognition of Handwritten Arabic Words », *In Ninth International Conference on Document Analysis and Recognition*, vol. 2, (2007), pp. 974-978.

- [11] H. Al-Rashaideh, « Preprocessing Phase for Arabic Word Handwritten Recognition », Russian Academy of Sciences, Russian Federation, vol. 6, no. 1, (2006), pp. 11-19.
- [12] J. Alkhateeb, « Word Based Off-line Handwritten Arabic Classification and Recognition », PhD Thesis, School of Computing, Informatics and Media, University of Bradford, (2010).
- [13] GONZALEZ, R. C. & WOODS, R. E. (2002) Digital Image Processing.
- [14] R. Legault and C. Y. Suen, « Optimal Local Weighted Averaging Methods in Contour Smoothing », IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, (1997), no. 8, pp. 801 –817.
- [15] W. L. Lee and K. C. Fan, « Document Image Pre-processing Based on Optimal Boolean Filter », Signal Processing, vol. 8, no. 1, (2000), pp. 45–55.
- [16] M. Khorsheed, « Off-line Arabic Character Recognition - A Review », Pattern Analysis Applications, vol. 5, (2002), pp. 31–45.
- [17] R. C. Gonzalez and R. E. Woods, “Digital Image Processing”, Dorling Kindersley Pvt. Ltd, (2009).
- [18] A. AL-Shatnawi and K. Omar, « A Comparative Study Between Methods of Arabic Baseline Detection », In International Conference on Electrical Engineering and Informatics, vol. 1, (2009), pp. 73–77.
- [19] M. Pechwitz and V. Margner, « Baseline Estimation for Arabic Handwritten Words, In Eighth International Workshop on Frontiers» ,in Handwriting Recognition, (2002), pp. 479–484.
- [20] Z. Q. Liu, J. Cai and R. Buse, «Handwriting Recognition Soft Computing and Probabilistic Approaches », Springer, (2010).
- [21] B. A. Yanikoglu and P. A. Sandon, « Recognizing Off-line Cursive Handwriting », In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (1994), pp. 397 –403.
- [22] Z. Al-Aghbari and S. Brook, « Hah Manuscripts: A Holistic Paradigm for Classifying and Retrieving Historical Arabic Handwritten Documents », Expert Systems with Applications, vol. 36, no. 8, (2009), pp. 10942-10951.

- [23] G. Kim, V. Govindaraju and S. N. Srihari, « An Architecture for Handwritten Text Recognition Systems », *International Journal on Document Analysis and Recognition*, vol. 2, no. 1, (1999), pp. 37–44.
- [24] J. Alkhateeb, J. Ren, S. Ipson and J. Jiang, « Component-based Segmentation of Words from Handwritten Arabic Text », *International Journal of Computer Systems Science and Engineering*, vol. 5, no. 1, (2009), pp. 54–58.
- [25] L. Lorigo and V. Govindaraju, « Segmentation and Pre-recognition of Arabic Handwriting », In *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, vol. 2, (2002), pp. 605-609.
- [26] T.Sari, L. Souici and M. Sellami, « Off-line Handwritten Arabic Character Segmentation Algorithm: ACSA », In *Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition*, (2002), pp. 452–457.
- [27] A. Hamid and R. Haraty, « A Neuro-heuristic Approach for Segmenting Handwritten Arabic Text », In *International Conference on Computer Systems and Applications*, (2001), pp. 110-113.
- [28] D. Motawa, A. Amin and R. Sabourin, « Segmentation of Arabic Cursive Script », In *Proceedings of The 4th International Conference on Document Analysis and Recognition*, (1997), pp. 625–628.
- [29] S. Al-Ma`adeed, « Recognition of Off-line Handwritten Arabic Words », PhD Thesis, University of Nottingham, (2004).
- [30] V. K. Govindan and A. P. Shivaprasad, « Character Recognition A Review », *Pattern Recognition*, vol. 23, no. 7, (1990), pp. 671–683.
- [31] A. Amin and J. Mari, « Machine Recognition and Correction of Printed Arabic Text », *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 5, (1989), pp. 1300 –1306.
- [32] K. M. Mohiuddin and J. Mao, « A Comparative Study of Different Classifiers for Handprinted Character Recognition », In *Pattern Recognition in Practice IV*, (1994), pp. 437–448.

[33] K. Fukushima. Neocognitron, « A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position » Biological Cybernetics, vol. 36 ,(1980) pp. 193–202.

[34] K. Fukushima. « A neural network model for selective attention in visual pattern recognition», Biological Cybernetics, vol.,55,(1986) pp.5–15.

[35] K. Fukushima. Analysis of the process of visual pattern recognition by the neocognitron. Neural Networks, 2(6) :413–420, 1989.

[36] X. Glorot, A. Bordes, and Y. Bengio, « Deep sparse rectifier neural networks » , in JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011), Apr. 2011.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, « Gradient-based learning applied to document recognition » ,Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

## Résumé

Dans notre travail on a utilisé Les réseaux de neurones convolutionnels pour la reconnaissance des caractères manuscrits arabes. Ce sont des réseaux de neurones multicouches qui sont spécialisés dans des tâches de reconnaissance de forme. 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, améliore les performances et permet une invariance du traitement par translation.

Comparés à d'autres algorithmes de classification d'image, les réseaux de neurones convolutifs utilisent relativement peu de pré-traitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres (apprentissage sans supervision), ce qui n'est pas le cas d'autres algorithmes plus traditionnels.

On a proposé un modèle avec une architecture (des couches de convolutions, des couches de pooling, des couches entièrement connectées et le nombre d'époque). Les résultats obtenus ont montré que le choix du nombre d'époque et la taille de la base d'images ainsi que la profondeur du réseau ont une grande influence pour avoir des meilleurs résultats.

## ملخص

متعددة عصبية شبكات وهي ، العربية المخطوطات حروف على للتعرف التلافيفية العصبية الشبكات استخدمنا ، عملنا في يرتبط فريد وزن استخدام في التلافيفية للشبكات الرئيسية الميزة تتمثل. الأنماط على التعرف مهام في متخصصة الطبقات الذاكرة حجم تقليل على الطريقة هذه تعمل. نفسه الالتواء قلب في توجد التي العصبية الخلايا جميع تدخل التي بالإشارات الترجمة معالجة معالجة ونتيح الأداء وتحسين

وهذا نسبيًا المسبقة المعالجة من القليل التلافيفية العصبية الشبكات تستخدم ، الأخرى الصور تصنيف بخوارزميات مقارنةً الأخرى الخوارزميات على ينطبق لا ما هو ، ذاتها حد في بها الخاصة التصفية عوامل تغيير عن مسؤولية الشبكة أن يعني تقليدية الأكثر

تم التي النتائج أظهرت. مرات وعدد بالكامل المتصلة الطبقات ، التجميع طبقات ، الالتفاف طبقات : بهيكل نموذج اقتراح تم أفضل نتائج على كبير تأثير لها الشبكة عمق وكذلك الصورة قاعدة وحجم الحقة عدد اختيار أن عليها الحصول

## **Abstract**

In our work we used the convolutional neural networks for the recognition of Arabic manuscript characters. These are multilayer neural networks that are specialized in pattern recognition tasks. A major advantage of convolutional networks is the use of a unique weight associated with the signals entering all the neurons of the same convolution core. This method reduces the memory footprint, improves performance<sup>3</sup> and allows translation processing invariance.

Compared to other image classification algorithms, convolutional neural networks use relatively little pre-processing. This means that the network is responsible for changing its own filters by itself (learning without supervision), which is not the case for other, more traditional algorithms.

We proposed a model with an architecture (convolution layers, pooling layers, fully connected layers and the number of epochs). The results obtained have shown that the choice of the number of epochs and the size of the image base as well as the depth of the network have a great influence to have better results.