

الجمهورية الجزائرية الديمقراطية الشعبية  
République algérienne démocratique et populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique  
جامعة عين تموشنت بلحاج بوشعيب  
Université –Ain Témouchent- Belhadj Bouchaib  
Faculté des Sciences et de Technologie  
Département Mathématiques et Informatique



Projet de Fin d'Etudes  
Pour l'obtention du diplôme de Master en Informatique  
**Domaine** : Mathématiques et Informatique  
**Filière** : Informatique  
**Spécialité** : Réseaux et Ingénierie des Données  
**Thème**

**Développement d'un système de reconnaissance de caractères arabes manuscrits**

**Présenté Par :**

- 1) M. Mohamed Amine HAMMOUTI
- 2) M. Abderrahim HADJ SAFI

**Devant le jury composé de :**

Mme Khadidja MEGAGUE	UAT.B.B (Ain Témouchent)	Président
Mme Amina BENOSMAN	UAT.B.B (Ain Témouchent)	Examinateur
Mme Fatima Zahra BERRAKEM	UAT.B.B (Ain Témouchent)	Encadrant

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# *Dédicaces*

**Je dédie ce travail :**

**A mon père que Dieu lui fasse miséricorde; qui m'a toujours soutenu dans mes études,**

**A ma chère mère qui m'a soutenue et encouragé durant toutes mes années d'études,**

**A mes chères soeurs et chère frère,**

**A toute ma chère famille,**

**A mes enseignants,**

**A mes chers amis,**

**A tous ceux qui m'aiment,**

**A tous ceux que j'aime,**

**A tout la promotion 2ème année master 2021/2022 .**

*Mohamed Amine*

# *Dédicaces*

**Je dédie ce travail à ma famille et mes amis,  
Un sentiment particulier de gratitude envers mes parents pour leurs encouragements,  
A mes soeurs. J'apprécierai toujours tout ce qu'elles ont fait,  
Je dédie ce travail et remercie tout particulièrement mon partenaire HAMMOUTI  
Mohamed Amine.**

*Abderrahim*

# ***Remerciement***

**En premier lieu, nous remercions ALLAH qui nous a permis d'arriver jusque-là,  
Grand merci à notre encadrante Mme Fatima Zahra BERRAKEM, pour l'orientation, la  
patience qui ont constitué un apport considérable.**

**Nos vifs remerciements vont également aux membres du jury  
Mme Amina BENOSMAN et Mme Khadidja MEGAGUE  
pour l'intérêt qu'ils ont porté à notre travail en acceptant d'examiner ce mémoire.**

**A tous les enseignants du département d'informatique.**

# Table des matières

<b>Dédicaces</b>	<b>i</b>
<b>Remerciement</b>	<b>iii</b>
<b>Table des matières</b>	<b>iv</b>
<b>Table des figures</b>	<b>x</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Liste des abréviations</b>	<b>xii</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Chapitre 1 Généralités sur la reconnaissance de l'écriture</b>	<b>3</b>
1.1 Introduction . . . . .	4
1.2 La reconnaissance automatique de l'écriture . . . . .	4
1.2.1 Définition . . . . .	4
1.3 Les différents aspects de la reconnaissance d'écriture . . . . .	5
1.3.1 Types d'écriture . . . . .	5
1.3.1.1 Écriture Imprimée . . . . .	5
1.3.1.2 Écriture Manuscrite . . . . .	5
1.3.2 Les modes d'acquisition des données . . . . .	6
1.3.2.1 Reconnaissance en ligne . . . . .	6
1.3.2.2 Reconnaissance hors ligne . . . . .	6
1.4 Reconnaissance de documents de caractères . . . . .	6

1.4.1	Processus de reconnaissance . . . . .	6
1.4.2	Saisie des documents . . . . .	7
1.4.3	Identification de la langue et de la fonte . . . . .	7
1.4.4	Reconnaissance de caractères . . . . .	8
1.5	Système de reconnaissance de l'écriture manuscrite hors ligne . . . . .	8
1.5.1	Architecture d'un système de reconnaissance d'écriture manuscrite .	9
1.5.2	Critères d'influences . . . . .	10
1.5.3	Approches de reconnaissance . . . . .	11
1.5.3.1	Reconnaissance de caractères isolés . . . . .	11
1.5.3.2	Reconnaissance de mots . . . . .	12
1.5.3.3	L'approche globale . . . . .	12
1.5.3.4	L'approche analytique . . . . .	12
1.5.4	Acquisition . . . . .	12
1.5.5	Techniques de prétraitement . . . . .	12
1.5.5.1	Réalignement (de-skew) . . . . .	12
1.5.5.2	Déparasitage (Despeckle) . . . . .	13
1.5.5.3	La binarisation . . . . .	13
1.5.5.4	Le redressement . . . . .	13
1.5.5.5	Suppression de la ligne . . . . .	13
1.5.5.6	Analyse de mise en page ou « zonage » . . . . .	13
1.5.5.7	Détection de lignes et de mots . . . . .	13
1.5.5.8	Reconnaissance du script . . . . .	13
1.5.5.9	Le zonage ou segmentation . . . . .	14
1.5.5.10	Normalisation . . . . .	14
1.5.6	Techniques de Segmentation . . . . .	14
1.5.6.1	Segmentation explicite . . . . .	15
1.5.6.2	Segmentation implicite . . . . .	15
1.6	La reconnaissance d'écriture et l'apprentissage automatique . . . . .	16
1.7	Domaines d'applications de la reconnaissance de l'écriture . . . . .	18
1.8	Travaux connexes . . . . .	18

1.8.1	Les travaux sur la base de données IFN/ENIT . . . . .	19
1.8.2	Les travaux sur la base de données ADAB . . . . .	20
1.8.3	Tableau récapitulatif . . . . .	21
1.9	Conclusion . . . . .	22
<b>2</b>	<b>Chapitre 2 Les différentes techniques pour la reconnaissance de l'écriture</b>	<b>23</b>
2.1	Introduction . . . . .	24
2.2	Classification . . . . .	24
2.2.1	Classificateur . . . . .	24
2.2.2	Modèle de classification . . . . .	24
2.2.3	Caractéristique . . . . .	24
2.2.4	Classification binaire . . . . .	25
2.2.5	Classement multi-classes . . . . .	25
2.2.6	Classification multi-étiquettes . . . . .	25
2.3	Réseaux de neurones artificiels . . . . .	25
2.3.1	Réseau de neurone monocouche Perceptron . . . . .	28
2.3.2	Réseau de neurone multicouche Perceptron (MLP) . . . . .	28
2.3.3	Avantages . . . . .	29
2.3.4	Désavantages . . . . .	30
2.3.4.1	Dépendance matérielle . . . . .	30
2.3.4.2	Fonctionnement inexplicé du réseau . . . . .	30
2.3.4.3	Assurance de la bonne structure du réseau . . . . .	30
2.3.4.4	La difficulté de montrer le problème au réseau . . . . .	30
2.3.4.5	La durée du réseau est inconnue . . . . .	31
2.4	Réseau neuronal convolutif . . . . .	31
2.4.1	Couche de convolution . . . . .	33
2.4.1.1	Filtre . . . . .	33
2.4.1.2	Foulée (Stride) . . . . .	33
2.4.2	Couche de POOLING . . . . .	33
2.4.3	Avantages . . . . .	33



2.4.4	Désavantages . . . . .	34
2.5	Réseau neuronal récurrent . . . . .	34
2.5.1	Réseau de mémoire à long court terme (LSTM) . . . . .	35
2.5.2	Avantages . . . . .	35
2.5.3	Désavantages . . . . .	36
2.6	Classification temporelle connexionniste (CTC) . . . . .	36
2.6.1	Encodage du texte . . . . .	36
2.6.2	Calcul des pertes . . . . .	37
2.6.3	Décodage . . . . .	38
2.7	Modèle de Markov caché (HMM) . . . . .	38
2.7.1	Avantages . . . . .	41
2.7.2	Désavantages . . . . .	41
2.8	Machine à vecteurs de support (SVM) . . . . .	42
2.8.1	Avantages . . . . .	44
2.8.2	Désavantages . . . . .	44
2.9	K-means . . . . .	44
2.9.1	Avantages . . . . .	46
2.9.2	Disavantages . . . . .	46
2.10	Algorithme génétique . . . . .	46
2.10.1	Notion de sélection naturelle . . . . .	47
2.10.2	Avantages . . . . .	47
2.10.3	Désavantages . . . . .	48
2.11	Conclusion . . . . .	48
<b>3</b>	<b>Chapitre 3 Modélisation et Implémentation d'un système de reconnaissance de mot arabe écrit manuscritement</b>	<b>49</b>
3.1	Introduction . . . . .	50
3.2	Environnement du développement . . . . .	50
3.2.1	Outils matériels . . . . .	50
3.2.2	Outils Logiciels . . . . .	50

3.2.3	Bibliothèques . . . . .	51
3.2.4	Développement WEB . . . . .	51
3.2.4.1	Flask . . . . .	51
3.3	Base de données . . . . .	52
3.4	L'approche proposée . . . . .	54
3.4.1	L'entrée de système . . . . .	55
3.4.2	La première phase . . . . .	56
3.4.3	La deuxième phase . . . . .	57
3.4.4	La troisième phase . . . . .	59
3.4.4.1	Décodage . . . . .	59
3.4.4.2	Le calcul de perte . . . . .	61
3.4.5	L'entraînement de modèle . . . . .	63
3.4.6	La validation de modèle . . . . .	64
3.5	Présentation de l'application . . . . .	64
3.5.1	Interface « Accueil » . . . . .	64
3.5.2	Interface « Help » . . . . .	66
3.5.3	Exemple . . . . .	67
3.6	Résultats . . . . .	68
3.7	Discussion des résultats . . . . .	69
3.8	Conclusion . . . . .	70
	<b>Conclusion générale</b>	<b>71</b>
	<b>Bibliographie</b>	<b>73</b>
	<b>Résumé</b>	<b>78</b>

# Table des figures

## 1 Chapitre 1 Généralités sur la reconnaissance de l'écriture

1.1 Architecture d'un système de reconnaissance d'écriture manuscrite . . . . .	9
1.2 Allographes de la lettre f . . . . .	11
1.3 Les segmentation possibles d'un mot en lettres ou entités plus fines . . . . .	16

## 2 Chapitre 2 Les différentes techniques pour la reconnaissance de l'écriture

2.1 Schéma de réseaux de neurones artificiels . . . . .	25
2.2 La démarche de modélisation de RNA . . . . .	26
2.3 L'organigramme de RNA . . . . .	27
2.4 Schéma de Réseau de neurone monocouche perceptron . . . . .	28
2.5 Schéma de MLP . . . . .	29
2.6 L'architecture standard de CNN . . . . .	32
2.7 Schéma de Réseau neuronal récurrent . . . . .	35
2.8 La matrice de sortie du réseau de neurones . . . . .	37
2.9 Exemples de HMM. . . . .	40
2.10 L'algorithme d'apprentissage de HM . . . . .	40
2.11 Exemple de SVM [20] . . . . .	42
2.12 Diagramme schématique du processus de classification des SVM . . . . .	43
2.13 Exemple de K-means . . . . .	45

## 3 Chapitre 3 Modélisation et Implémentation d'un système de reconnaissance de mot arabe écrit manuscritement

3.1 Structure de notre fichier créé "words.txt" . . . . .	53
3.2 Structure de la base de données utilisée . . . . .	53
3.3 L'architecture de notre système . . . . .	54
3.4 Exemple d'entrée de notre système . . . . .	55

3.5 La sortie de la première phase (CNN) . . . . .	56
3.6 La sortie de CNN . . . . .	57
3.7 Le schéma de RNN (Bi-LSTM) . . . . .	58
3.8 Exemple de prédiction . . . . .	59
3.9 La fusion des répétitions . . . . .	60
3.10 Exemple d'entrée d'un mot contenant des lettres répétées . . . . .	60
3.11 La sortie de CTC s'il n'y a pas des blancs . . . . .	61
3.12 La sortie de CTC s'il y a des blancs . . . . .	61
3.13 Exemple d'entrée après l'ajoute de 3 pas de temps . . . . .	62
3.14 Résultats obtenus lors d'entraînement le modèle . . . . .	63
3.15 Résultats obtenus lors de validation le modèle . . . . .	64
3.16 Interface d'Accueil . . . . .	65
3.17 Interface de choix d'images . . . . .	65
3.18 Interface principale après la sélection d'image . . . . .	65
3.19 Interface principale après l'affichage du résultat . . . . .	66
3.20 Interface d'Accueil . . . . .	66
3.21 Interface Help . . . . .	67
3.22 Exemple d'image contenant le mot "الناصور" . . . . .	67
3.23 Exemple d'image contenant la lettre "و" . . . . .	68
3.24 Exemple depuis l'invite de commande d'image contenant le mot "المزونة" . . . . .	68

# Liste des tableaux

## **1 Chapitre 1 Généralités sur la reconnaissance de l'écriture**

1.1 Tableau récapitulatif sur les travaux connexes . . . . .	22
--	----

## **3 Chapitre 3 Modélisation et Implémentation d'un système de reconnaissance de mot arabe écrit manuscritement**

3.1 Structure de fichier "words.txt" . . . . .	52
3.2 Structure de notre système . . . . .	55
3.3 La sortie de RNN matrice softmax . . . . .	58
3.4 La sortie de Bi-LSTM . . . . .	60
3.5 Prédiction de (best path) . . . . .	60
3.6 La sortie de Bi-LSTM . . . . .	62
3.7 Tout les possibilités de mot . . . . .	62
3.8 Les résultats obtenus . . . . .	69

# Liste des abréviations :

**ANN** : Artificial Neural Network.

**Bi-LSTM** : Bidirectional Long-Short Term Memory.

**CNN** : Convolutional Neural Network.

**CRNN** : Convolutional Recurrent Neural Networks.

**CTC** : Classification Temporelle Connexionniste .

**ELM** : Extreme Learning Machine.

**HMM** : Hidden Markov Models .

**LSTM** : Long-Short Term Memory.

**MLP** : Multi-Layer Perceptron.

**NN** : Neural Network.

**OCR** : Optical Character Recognition.

**RNA** : Réseaux de Neurones Artificiels.

**RNN** : Recurrent Neural Network.

**SVM** : Support Vector Machine.

# Introduction générale

L'écriture est l'une des méthodes de base utilisées dans la communication humaine. Par l'écriture, les gens peuvent concrétiser et enregistrer leurs propres opinions, activités, impressions, etc.

Aujourd'hui, la reconnaissance d'écriture est un domaine de recherche très actif, appartenant à un axe dans le domaine de la reconnaissance de formes, dans le but de développer des systèmes automatisés capables de lire un texte aussi efficacement que l'homme.

Il existe de nombreuses applications utiles et intéressantes de la reconnaissance de texte, allant de la numérisation et de l'indexation des manuscrits historiques à la lecture automatique des plaques d'immatriculation des voitures, le traitement automatique des chèques bancaires, la classification automatique des enveloppes, le traitement automatique des formulaires et tâches impliquant la reconnaissance de mots. De plus, en raison de l'utilisation généralisée du papier dans la vie quotidienne et de la nécessité de les stocker électroniquement pour une sauvegarde et une récupération efficace, des systèmes de reconnaissance de texte manuscrit hautement fiables et robustes sont nécessaires.

La reconnaissance de texte manuscrit implique la reconnaissance de texte à partir d'images contenant du texte manuscrit saisi par des humains.

De plus, la reconnaissance de texte imprimé soit beaucoup plus facile que la reconnaissance de texte manuscrit, Car la reconnaissance de texte manuscrit fait face à de nombreux défis tels que la reconnaissance de documents tachés, la reconnaissance de texte dans des polices multiples ou inhabituelles et l'alignement de texte et de documents avec une orientation non uniformes, la grande variation de l'écriture humaine entre différents écrivains ou le même écrivain.

Malgré que la langue arabe est considérée comme l'une des langues les plus importantes au monde, car elle se classe au sixième rang mondial en termes de nombre de locuteurs. C'est la langue officielle de 25 pays et compte environ 437 millions de locuteurs. les systèmes de reconnaissance de texte arabe malheureusement n'ont pas encore atteint leur maturité par rapport à d'autres langues étrangères, en raison du manque de

ressources linguistiques telles que les bases de données, les outils d'agrégation, d'annotations et de prétraitement des ensembles de données. Ainsi, elle est considérée comme une langue difficile par rapport aux d'autres langues, en raison du grand nombre de règles lexicales et de conjugaison. De plus, la façon dont les lettres sont écrites au début, au milieu et à la fin du mot est différente, et l'écriture diffère d'une personne à l'autre.

Donc, L'objectif de notre projet de fin d'étude est la mise en œuvre d'un système pour la reconnaissance d'écriture manuscrite en langue arabe à partir d'une image scannée (système hors ligne). Où, nous allons basé sur les réseaux de neurones récurrents convolutifs (CRNN) qui combine les réseaux de neurones convolutifs (CNN) et les réseaux de neurones récurrents (RNN); Dont, le CNN permet la vision par ordinateur en profondeur et le RNN traite le langage naturel. De plus, nous avons utilisé la classification temporelle connexionniste (CTC) pour le codage, nous avons choisi ces classificateurs car ils sont les plus populaires en apprentissage profond et ils nous donnent des résultats fiables. Ainsi que, nous allons basé sur les réseaux de neurones récurrents convolutifs (CRNN) utilisé les bases de données de référence les plus connue (IFN/ENIT)[5] et (ADAB)[24], comme nous avons ajouté 714 images de caractères extraites de la base de données introduite par El-Sawy et al [26], afin d'améliorer nos résultats. Le travail présenté dans ce mémoire est organisé en trois chapitres structurés comme suit :

Dans le premier chapitre nous allons présenter une description générale des différents éléments et aspects du processus de reconnaissance de l'écriture.

Dans le deuxième chapitre, nous allons présenter une série de classifieurs couramment utilisés dans les systèmes des reconnaissances d'écriture et montrerons leurs avantages et leurs inconvénients.

Dans le dernier chapitre, nous allons réaliser un système pour la reconnaissance de l'écriture manuscrit en langue arabe en utilisant une approche de classification basée sur les réseaux de neurones récurrents convolutifs (CRNN), nous allons créer ce système sous forme d'une application WEB pour être accessible pour tout le monde, et nous allons présenter et discuter les résultats obtenus.

Et enfin, nous terminons notre manuscrite par une conclusion générale et quelques travaux futurs qui peuvent être ajoutés.



# **Chapitre 1**

## **Généralités sur la reconnaissance de l'écriture**

## 1.1 Introduction

La reconnaissance automatique de l'écriture est une discipline en plein essor depuis deux décennies, consistant en développement et la création des systèmes capables de reconnaître toutes sortes d'écritures dans ses formes manuscrites et imprimées. Nous présentons dans ce chapitre une description générale des différents éléments et aspects du processus de reconnaissance de l'écriture manuscrite. En fait, dans un premier temps, nous présentons un ensemble de concepts préliminaires qui introduisent le domaine. Deuxièmement, nous décrivons techniquement le système de reconnaissance de l'écriture manuscrite à travers un graphe fonctionnel de référence en plusieurs étapes. Nous donnons ensuite des exemples de domaines d'application pour la reconnaissance de l'écriture manuscrite. Enfin, nous terminerons ce chapitre en présentant quelques travaux connexes.

## 1.2 La reconnaissance automatique de l'écriture

### 1.2.1 Définition

L'écriture est un ensemble de marques graphiques artificielles ; ces marques se combinent avec les relations conventionnelles des marques, généralement définies dans le cadre du langage naturel. Plus précisément, ces jetons représentent des caractères de langage ou des lettres.

La reconnaissance automatique de l'écriture également connue sous le nom de reconnaissance de texte (manuscrits, imprimés), est la capacité d'un ordinateur à recevoir et à interpréter des entrées manuscrites intelligibles provenant de sources telles que des documents papier, des photographies, des écrans tactiles et d'autres appareils.

La reconnaissance automatique de l'écriture (imprimés et manuscrits) est un domaine qui implique le développement de systèmes capables de reconnaître du texte manuscrit ou imprimé. Ces systèmes de reconnaissance dépendent de plusieurs facteurs, à savoir le nombre de scripteurs, le type ou le style d'écriture, la taille et le type de vocabulaire.

Ces facteurs sont des paramètres fondamentaux de sa complexité de mise en œuvre et de conception. Ils sont également différenciés en fonction de différents critères, tels que la manière dont ils sont identifiés, la localisation des unités d'écriture de base dans les mots et la stratégie suivie par la nature de la langue. [20]

## **1.3 Les différents aspects de la reconnaissance d'écriture**

### **1.3.1 Types d'écriture**

#### **1.3.1.1 Écriture Imprimée**

Le procédé d'impression inventé par Gutenberg au milieu du quinzième siècle a transformé nos vies grâce à une diffusion plus large et plus rapide des connaissances, et une grande partie du patrimoine culturel et technologique de l'humanité n'est encore disponible que sous forme de documents papier; ainsi, les entreprises et les autorités locales sont confrontées à la nécessité d'un retraitement, c'est-à-dire d'une conversion rétroactive au format électronique.[19]

#### **1.3.1.2 Écriture Manuscrite**

L'écriture manuscrite est l'une de nos formes d'expression les plus évoluées, elle est de toute façon très complexe, c'est une manière particulière d'encoder les informations liées à nos pensées, nos sentiments, en d'autres termes, la reconnaissance de l'écriture manuscrite a généralement deux caractéristiques :

- Un exemple de script pour vérifier la signature de l'auteur.
- Un autre concerne le sens de ce qui est écrit (contexte). [29]

## 1.3.2 Les modes d'acquisition des données

### 1.3.2.1 Reconnaissance en ligne

L'écriture en ligne (ou dynamique), Ce mode de reconnaissance fonctionne en temps réel (pendant l'écriture) et les caractères sont reconnus comme de l'écriture manuscrite. Ce modèle est généralement réservé à l'écriture manuscrite, et c'est une méthode "signal" pour identifier les données unidimensionnelles. en prenant en compte les informations relatives au mécanisme d'écriture telles la position des points, la vitesse et l'accélération qui sont des fonctions du temps. L'acquisition de l'écriture est généralement garantie par une tablette graphique équipée d'un stylet électronique. [18]

### 1.3.2.2 Reconnaissance hors ligne

L'écriture hors-ligne (ou en différé, ou encore statique), elle convient aux documents imprimés et les manuscrits déjà rédigés. Les données en entrée sont acquises via un scanner. Elle a comme tâche de déterminer quelles lettres ou mots sont présents dans l'image du texte scanné. [37]

## 1.4 Reconnaissance de documents de caractères

La reconnaissance de documents (contenant des caractères) ou l'analyse d'images de documents implique l'ensemble du processus de conversion d'images. Ce processus couvre l'ensemble des problématiques liées à la langue écrite et à sa transformation numérique : reconnaissance des caractères, mise en forme du texte, structure du contenu, accès aux informations indexées de l'application.

### 1.4.1 Processus de reconnaissance

Les principales étapes de la chaîne de reconnaissance sont :

**L'acquisition** qui permet de convertir des documents papier en image numérique (bitmap).

Le **prétraitement** dont le rôle est de préparer l'image du document pour le traitement. Les opérations de prétraitement impliquent la correction de l'image, la suppression du bruit et des informations redondantes, et enfin la sélection des régions de traitement utiles.

La **reconnaissance de contenu** conduit le plus souvent à la reconnaissance de texte et à l'extraction de la structure logique.

La **correction** des résultats de la reconnaissance en vue de valider l'opération de numérisation.

### 1.4.2 Saisie des documents

La capture des documents est réalisée par numérisation optique. Les résultats sont stockés dans un fichier de points, appelés pixels, dont la taille dépend de la résolution. Un pixel peut avoir les valeurs suivantes : 0 (éteint) ou 1 (actif) pour les images binaires, 0 (blanc) à 255 (noir) pour les images en niveaux de gris, et trois canaux avec des valeurs de couleur comprises entre 0 et 255 pour les images en couleur.

### 1.4.3 Identification de la langue et de la fonte

Pour la reconnaissance du langage, Spitz, [43] l'un des pionniers en la matière, a proposé une méthodologie permet la classification de cinq langues différentes au sein d'un même document. Premièrement, il distingue les langues latines des langues asiatiques utilisent l'écart type de la position verticale du concave par rapport à la ligne de base. Ces concavités se situent aux confins de la ligne de base latine, tandis que Le chinois, le japonais et le coréen sont répartis uniformément. Ensuite, les trois langues asiatiques ont été séparées en examinant les histogrammes de leurs distributions ponctuelles. Zramdini [46] a proposé le système ApOFIS, capable de distinguer plus de 280 polices différentes en combinant 10 polices, 7 tailles et 4 styles. Les polices ont été reconnues avec une précision de 97%, tandis que les styles, le corps et la pente ont été reconnus avec une précision de 97,5 à 99,9%.

#### 1.4.4 Reconnaissance de caractères

Le texte est une association de caractères appartenant à un alphabet, regroupés dans des mots d'un vocabulaire donné. Les système de reconnaissance de caractères doit trouver ces caractères, d'abord les identifier individuellement, puis les vérifier en effectuant une reconnaissance lexicale des mots qui les contiennent. Un système de reconnaissance de caractères est composé de plusieurs modules : segmentation, Apprentissage, identification et vérification du vocabulaire [23].

La segmentation permet de séparer les éléments de texte, les mots et les caractères pour la reconnaissance.

La reconnaissance de caractères peut déterminer l'identité d'un caractère en apprenant sa forme.

L'un des principaux objectifs est d'améliorer le taux de reconnaissance en utilisant des dictionnaires de di-grammes, tri-grammes ou n-grammes pour la correction orthographique ou morphologique. Lorsqu'il s'agit de reconnaissance de phrases entières, nous introduisons des contraintes à des niveaux successifs : lexical, syntaxique ou sémantique.

### 1.5 Système de reconnaissance de l'écriture manuscrite hors ligne

Dans cette partie, nous intéressons essentiellement aux systèmes de la reconnaissance automatique de l'écriture manuscrite hors ligne. En effet, pour mieux cerner ce système, nous allons les introduire à travers un modèle général à plusieurs étapes qui reflète globalement le processus de reconnaissance de l'écriture manuscrite. Les différentes techniques et méthodes développées et utilisées à chaque étape du modèle sont discutées et analysées. [20]

### 1.5.1 Architecture d'un système de reconnaissance d'écriture manuscrite

La tâche fondamentale des systèmes de reconnaissance d'écriture manuscrite hors ligne est de convertir les signaux d'écriture manuscrite sous différentes formes, imprimées ou manuscrites, en représentations symboliques, en tenant compte de leurs différentes propriétés spatiales. Pour ce faire, il nécessite un certain nombre d'étapes à mettre en œuvre. Toutes ces étapes forment généralement la structure d'un système de reconnaissance d'écriture manuscrite, qui peut se résumer par le schéma de la Figure 1.1 [20].

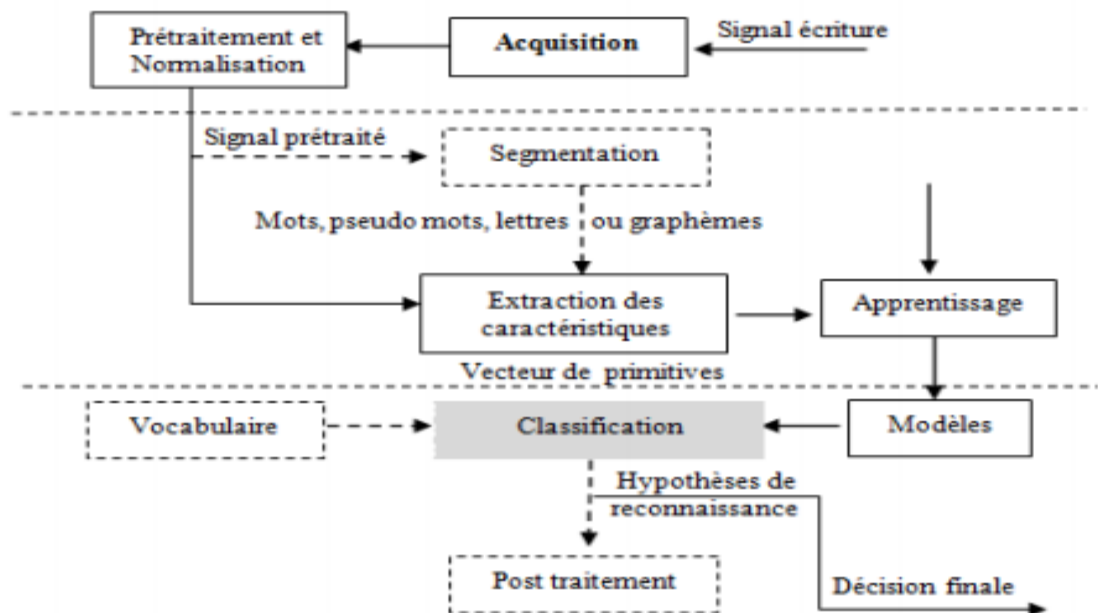


FIGURE 1.1 – Architecture d'un système de reconnaissance d'écriture manuscrite [20]

Le but de la phase d'acquisition est de numériser l'entrée à l'aide d'un scanner.

Le but des étapes de prétraitement et de normalisation est de réduire le bruit et la variabilité d'écriture afin que seules les informations utiles soient conservées.

La phase d'extraction des primitives permet une description complète des formes à reconnaître dans un espace à dimensions multiples à partir de la représentation d'image.

L'étape de segmentation permet de décomposer des images de texte en entités (mots, caractères ou graphèmes) pour réduire la complexité des modules de traitement ultérieurs.

La phase d'apprentissage consiste à trouver le modèle qui correspond le mieux à l'entrée du problème. Le résultat est une base de données de formation qui constitue la ligne de base du système.

Pour l'étape de classement des caractères ou des mots, les descriptions des caractères à reconnaître dans la base de test sont comparées aux descriptions des caractères dans la base de référence.

En plus de ces étapes, on peut également trouver une étape de post-traitement, qui est une opération optionnelle dont le rôle est de vérifier et éventuellement de corriger les hypothèses de reconnaissance générées par le module de classification en utilisant des informations contextuelles telles que lexicales, syntaxiques, sémantiques, pragmatiques, etc.

Par la suite, nous passerons en revue des techniques de mise en œuvre de chaque phase du système.

### 1.5.2 Critères d'influences

Sur le plan méthodologique, l'évaluation de la complexité de certains enjeux doit tenir compte plusieurs critères orthogonaux : [36]

**Disposition spatiale du texte :** La classification de Tappert [45] Indique que la présentation du texte peut être soumise à deux types de contraintes : externes conduisant à une écriture pré-encadrée, partitionnée, en tête ou générale; et internes provenant des habitudes spécifiques de chaque rédacteur et conduisant à une écriture séparée, groupée, scénarisée (bâton), pure Cursive ou écriture mixte.

**Nombre d'écrivains :** La difficulté de reconnaissance augmente avec ce nombre, divisant les gammes en trois : mono, multi et omni-scripteurs. En multi-écrivain, le système doit s'adapter à l'écriture par plusieurs écrivains, alors qu'en full-writer le système doit pouvoir généraliser son apprentissage à tout type d'écriture.



**Taille du Vocabulaire :** Il y a une distinction entre les applications avec un vocabulaire limité (< 100 mots) et les applications avec un vocabulaire très large (> 10 000 mots). Il est clair que dans le premier cas la complexité est moins, car le nombre réduit limite l'empreinte mémoire et facilite l'utilisation d'une méthode de reconnaissance simple et donc rapide, en scannant tous les mots du dictionnaire par le système.

D'autres types de normes peuvent affecter la complexité d'un système de reconnaissance de l'écriture manuscrite. Ils sont liés aux changements intrinsèques de l'écriture, dans le contexte de l'écriture cursive. Parmi ces changements, on peut remarquer :

**Spécifique à l'auteur :** style personnel en termes de rapidité, de continuité et de régularité. Tous ces éléments affectent la forme des lettres (écriture inclinée, recourbée, arrondie, linéaire, etc.) et bien sûr la forme des ligatures, et parfois la reconnaissance des limites entre les lettres.

**propres à l'écriture manuscrite :** La forme d'une lettre dépend de sa position dans le mot (début, milieu, fin) et des lettres adjacentes. Tous ces changements conduisent à différentes formes morphologiques (images) de la même lettre, appelées allographes.[23]

La Figure 1.2 donne des exemples d'allographes de la lettre f :

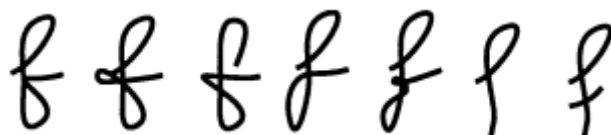


FIGURE 1.2 – Allographes de la lettre f [23]

### 1.5.3 Approches de reconnaissance

#### 1.5.3.1 Reconnaissance de caractères isolés

il s'agit de la tâche la plus élémentaire d'un système de reconnaissance de l'écriture manuscrite. Le travail analytique porte sur un élément du vocabulaire à la fois (considéré comme une forme globale).

### 1.5.3.2 Reconnaissance de mots

Il existe deux approches opposées dans la reconnaissance globale et analytique des mots :

### 1.5.3.3 L'approche globale

L'approche globale a une vision générale du mot, elle s'appuie sur une description unique du mot image, traité comme une entité indivisible. Disposant de beaucoup d'informations, il lui est plus facile d'absorber les changements dans son écriture.

### 1.5.3.4 L'approche analytique

L'approche analytique peut surmonter ces limitations, mais nécessite une décomposition basée sur les mots (segmentation) pour une interprétation locale.

## 1.5.4 Acquisition

Dans les systèmes hors ligne, l'acquisition est l'étape de numérisation de l'écriture à l'aide d'un scanner. Les résultats de cette phase représentent les entrées du système. Cela dépend de deux paramètres importants, la résolution et le niveau d'éclairage. L'acquisition est très simple mais très importante car elle influence fortement les prochaines étapes [20].

## 1.5.5 Techniques de prétraitement

La majorité des logiciels des systèmes de reconnaissance de l'écriture manuscrite prétraitent les images pour augmenter les chances de reconnaissance.

### 1.5.5.1 Réalignement (de-skew)

Si le document n'a pas été correctement aligné lorsqu'il est numérisé, il peut avoir besoin d'être tourné de quelques degrés dans le sens horaire ou antihoraire pour s'assurer que les lignes de texte soient parfaitement horizontales ou verticales. [2]

### 1.5.5.2 Déparasitage (Despeckle)

Déparasitage (Despeckle) consiste à enlever les taches sur le document ou lisser ses bords. [2]

### 1.5.5.3 La binarisation

La binarisation permet de passer d'une image de niveaux de gris à une image binaire composée de 2 valeurs 0 et 1. [23]

### 1.5.5.4 Le redressement

Le redressement est une opération fréquente en analyse de documents, souvent due à un mauvais positionnement du document sur le scanner, conduisant à une inclinaison de l'image. [23]

### 1.5.5.5 Suppression de la ligne

La suppression de la ligne consiste à nettoyer les boîtes et les lignes non glyphes. [2]

### 1.5.5.6 Analyse de mise en page ou « zonage »

L'analyse de mise en page ou « zonage » identifie les colonnes, les paragraphes et les annotations dans les blocs. Particulièrement utile dans les mises en page et les tables multicolonnées [2].

### 1.5.5.7 Détection de lignes et de mots

Établir des formes avec les mots et les caractères de base, diviser les mots au besoin [2].

### 1.5.5.8 Reconnaissance du script

Dans plusieurs documents linguistiques, le script peut se transformer au niveau des mots. Donc l'identification du script est essentielle avant que le système de reconnaissance de l'écriture manuscrite puisse être utilisé pour gérer un script en particulier. [2]

### 1.5.5.9 Le zonage ou segmentation

Le zonage ou segmentation physique permet de localiser les blocs d'information d'homogènes et de les classer en fonction de leur contenu : texte ou non texte. [23]

### 1.5.5.10 Normalisation

Normaliser le ratio des dimensions de l'image (aspect ratio) et l'échelle (scale ratio) [2].

## 1.5.6 Techniques de Segmentation

Dans les systèmes de reconnaissance la segmentation est une opération très critique.

La segmentation est le processus consistant à décomposer l'image d'un texte en entités (mots, caractères ou graphèmes) qui font parties d'un alphabet prédéfini selon le but visé. Par ailleurs, elle permet de réduire la complexité des modules de traitements utilisés par la suite.

En effet, la séparation des lignes, des mots, des pseudo-mots, des caractères et des graphèmes constituent des opérations difficiles et coûteuses, tant les écritures sont variées, les lignes sont parfois enchevêtrées et les caractères généralement liés (cas de l'arabe : l'écriture est semi-cursive) les unes aux autres.

Selon la littérature, le problème le plus ardu c'est le cas de la segmentation de l'écriture cursive. Dans le but de résoudre cette problématique plusieurs algorithmes de segmentation existent mais posent souvent une polémique autour de choix de l'un par rapport à l'autre, de ce fait, l'utilisation d'un algorithme est conditionnée par son efficacité et le type de la graphie étudiée. Les solutions proposées se basent sur deux stratégies de segmentation différentes ci-dessous : [20]

### 1.5.6.1 Segmentation explicite

La segmentation explicite, s'appuie sur un découpage à priori de l'image en sous-unités qui peuvent être des lettres ou des graphèmes (cf. Figure 1.3). Cette décomposition se base directement sur une analyse morphologique du texte ou de mot, ou sur la détection des points caractéristiques tels que les points d'intersection, les points d'inflexion, les boucles à l'intérieur du texte ou de mot pour localiser les points de segmentation potentiels. Dans le cas de segmentation en graphèmes, les textes ou les mots sont alors reconnus non comme une suite de lettres reconnues indépendamment, mais comme une suite de graphèmes globalement comparés à l'entrée.

L'avantage de cette segmentation c'est que l'information est localisée explicitement et ça va dans le sens de delimme de Sayre, puisque on sépare les lettres non pas d'après leur reconnaissance, mais d'après des critères topologiques ou morphologiques. Le défaut majeur de cette segmentation vient en premier lieu de choix des limites indépendant des critères des modèles : les limites sous-optimales pour les modèles et modélisation sousoptimale. En deuxième lieu, il n'existe pas de méthode de segmentation fiable à 100%, toute erreur de segmentation pénalise les performances de système. [20]

### 1.5.6.2 Segmentation implicite

La segmentation est dite implicite lorsque celle-ci est basée sur un moteur de reconnaissance pour valider et classer les hypothèses de segmentation (recherche de chemin des points de segmentation possibles) (cf. Figure 1.3). Dans ce cas, la segmentation et la reconnaissance sont réalisées conjointement, d'où le nom parfois employé de "segmentation-reconnaissance intégrée".

Contrairement à la segmentation explicite, on ne procède pas à une segmentation à priori en entrée, mais à une segmentation aveugle du mot dans le sens où elle ne dépend en aucun cas d'une analyse de l'image à segmenter et qui dépend d'une compétition des classes des lettres ou graphèmes en sortie du classifieur. L'avantage de cette segmentation c'est que l'information est localisée par les modèles des lettres et la validation se fait par ses modèles. Il n'y aura pas d'erreur de segmentation car en connaissant les lettres, on

n'engendre pas d'erreur de segmentation.

Dans les approches à segmentation implicite, la tâche de segmentation est accomplie par le système. Elle est soutenue simultanément par un processus de reconnaissance, en évitant la pré segmentation d'un mot en lettres ou entités plus fines . [20]

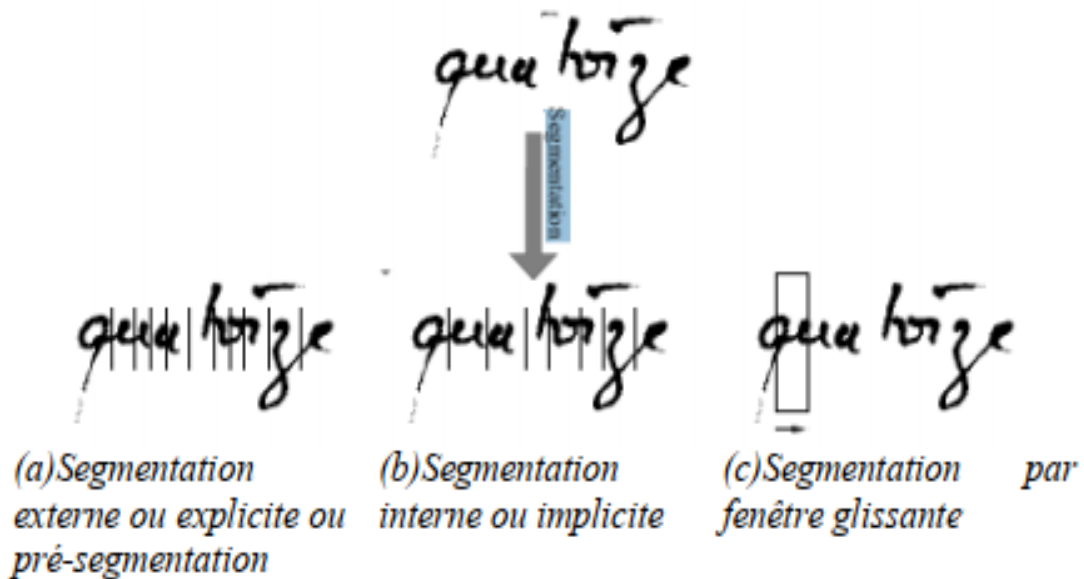


FIGURE 1.3 – Les segmentation possibles d'un mot en lettres ou entités plus fines [20]

## 1.6 La reconnaissance d'écriture et l'apprentissage automatique

Dans les systèmes de reconnaissance d'écriture manuscrite, il est difficile d'apprendre par généralisation comme un humain. Par exemple, il leur est difficile de construire un bon modèle d'un mot ou d'un caractère et de pouvoir ensuite le reconnaître efficacement dans une nouvelle image. Dans la littérature, il existe plusieurs études centrées sur le problème de l'apprentissage automatique. Selon ces travaux, l'apprentissage peut être divisé selon les principaux types suivants : [20]

**Apprentissage supervisé**

Elle suppose que chaque donnée d'apprentissage est étiquetée, par exemple lorsque l'on dispose d'un ensemble de données dont les catégories d'appartenance sont connues à priori. Il s'agit alors de chercher les surfaces de décision séparant au mieux les classes. Il s'agit donc d'un ensemble de techniques conçues pour deviner qu'un individu appartient à une certaine catégorie [20]

**Apprentissage non-supervisé**

Contrairement à l'apprentissage supervisé, dans l'apprentissage non supervisé, il n'y a pas de données étiquetées. Sinon, lorsque nous avons une base de données sans connaître leurs classes membres. Il s'agit de découvrir la structure derrière les données sans imposer de modèles [20].

**Apprentissage semi-supervisé**

Cet apprentissage est un bon compromis entre apprentissage supervisé et non supervisé. Parce que cela suppose que nous avons très peu de données étiquetées et beaucoup de données non étiquetées. Ensuite, apprenez à partir de deux sources de données [20].

**Apprentissage par renforcement**

Fait référence à toute méthode adaptative de résolution de problèmes de décision séquentielle. Ce type d'apprentissage est utile dans le cadre de l'apprentissage interactif. Au cours de ce processus d'apprentissage, le système adaptatif interagit avec son environnement et reçoit à son tour des signaux de renforcement [20].

Récemment, des méthodes d'apprentissage croisé basées sur HMMs ont été proposées pour résoudre le problème de la réestimation des modèles de lettres dans les méthodes analytiques. Ces méthodes comprennent la formation de modèles alphabétiques globalement par mots. Cela se traduira par une maximisation de la probabilité des mots dans les modèles de lettres.

## 1.7 Domaines d'applications de la reconnaissance de l'écriture

La reconnaissance automatique de l'écriture manuscrite hors ligne présente un intérêt indéniable pour permettre de nombreuses applications à grande échelle, telles que celles rencontrées dans certains domaines d'activité parmi lesquelles on peut citer :[20]

-L'authentification et la lecture de chèques bancaires par la reconnaissance des montants littéraux et des montants numériques manuscrits.

-La lecture automatique de formulaires et de documents administratifs.

-La lecture des adresses postales et le tri automatique du courrier.

-L'échange de fichiers informatisés à distance dans le domaine de télécommunications.

-L'indexation et l'archivage automatique de documents.

-La reconnaissance de documents techniques (schémas électronique, dessins techniques, plans architecturaux, plans cartographiques, etc.).

-La transcription assistée par ordinateur appliqué à la reconnaissance du manuscrit mono-scripteur qui permettra à la rétro-conversion de manuscrits anciens.

-La reconnaissance de numéros minéralogiques pour le contrôle routier, l'authentification et l'identification de manuscrits et l'identification du scripteur.

-La recherche d'information dans une base de documents manuscrits telle que l'identification du scripteur.

-Les applications de reconnaissance de l'écriture en ligne à travers les PDA, Tablet- PC, Ordinateurs sans clavier, ou stylo caméra.

## 1.8 Travaux connexes

Afin de concevoir un système de reconnaissance de caractères arabes, il est nécessaire d'étudier plusieurs travaux connexes, Voici quelques travaux que nous avons étudié, ces travaux sont appliqués à différentes bases de données (IFN/ENIT, ADAB) :



### 1.8.1 Les travaux sur la base de données IFN/ENIT

Azeem et al présentent un nouveau système de reconnaissance de l'écriture manuscrite arabe sans segmentation basé sur le modèle de Markov caché (HMM). Deux apports principaux sont introduits : une nouvelle technique de division de l'image en segments horizontaux non uniformes pour en extraire les traits et une nouvelle technique pour résoudre les problèmes d'inclinaison des caractères en fusionnant plusieurs HMM. Le taux de reconnaissance atteignent 93.44%. [21]

Khmiri et al proposent un nouveau système de reconnaissance de mots arabes manuscrits. Il est développé sur la base du modèle de Markov caché horizontal-vertical et du modèle de réseau bayésien dynamique. Les taux de reconnaissance atteignent 92,19% avec le modèle de Markov caché horizontal-vertical et 88,82% avec un réseau bayésien dynamique. [32]

Maalej et Monji proposent un nouveau système basé sur l'intégration de deux réseaux de neurones profonds. D'abord un réseau neuronal convolutif (CNN) pour extraire automatiquement les caractéristiques des images brutes, puis la mémoire bidirectionnelle à long-court terme (BLSTM) suivie d'une couche de classification temporelle connexionniste (CTC) pour l'étiquetage des séquences. le taux de reconnaissance atteint 92,21%. [38]

Khémiri et al ont analysé les réseaux de neurones bayésiens et convolutionnels (CNN), en tant que deux méthodes d'apprentissage automatique différentes, pour reconnaître les mots manuscrits arabes. Notre apport est triple. Tout d'abord, ils décrivons les principaux points forts de l'architecture de réseau bayésien dynamique (DBN), en particulier par rapport aux réseaux bayésiens standard. Pour cela, certaines caractéristiques structurelles sont extraites de l'image du mot et considérées comme des entrées pour différentes architectures de réseaux bayésiens (BN) telles que Naïve Bayes (NB), Tree Augmented Naïve Bayes (TAN), Forest Augmented Naïve Bayes (FAN) et Hidden Markov. modèle (HMM). Deuxièmement, nous modélisons une architecture d'apprentissage en profondeur, Les résultats obtenus montrent la précision relativement élevée de la combinaison DBN et CNN : 95,20% par rapport aux modèles restants. [31]

Jayech et al proposent un système de reconnaissance manuscrite arabe hors ligne

pour les noms de villes tunisiennes. Cet proposition basée sur les modèles de Markov cachés. Le taux de reconnaissance atteignent 91.10%. [30]

Le travail proposé par Rabi et al rapporte une méthode efficace d'amélioration des systèmes qui prend en considération le contexte de caractère en appliquant un HMM basé sur la formation intégrée. Ce HMM est amélioré par un réseau de neurones artificiels qui sont incorporés dans le processus de classification pour estimer. les probabilités d'émission. Le taux de reconnaissance atteint 89,03%. [39]

Elleuch et al ont proposé un nouveau modèle axé sur l'intégration de deux classificateurs; Convolutional Neural Network (CNN) et Support Vector Machine (SVM) pour la reconnaissance de l'écriture arabe hors ligne (OAHR) sur laquelle la technique de décrochage a été appliquée. Le taux de reconnaissance atteignent 92.95%. [27]

Tamen et al proposent un système efficace de classification multiple pour la reconnaissance de mots manuscrits arabes. Tout d'abord, ils utilisent des moments de Chebyshev (CM) améliorés avec certaines caractéristiques statistiques et basées sur les contours (SCF) pour décrire des images de mots. Ensuite, ils combinent plusieurs classifieurs intégrés au niveau décisionnel. ils considèrent le perceptron multicouche (MLP), la machine à vecteurs de support (SVM) et les classificateurs Extreme Learning Machine (ELM). Ils proposent plusieurs règles de combinaison entre les classificateurs MLP, SVM et ELM entraînés avec les fonctionnalités CM et SCF. Les résultats numériques sont compétitifs et montrent que leur système est capable d'atteindre un taux de reconnaissance global égal à 96,82% pour le jeu de données considéré. [44]

Rabi et al présentent un système de reconnaissance hors ligne de texte manuscrit arabe cursif basé sur des modèles de Markov cachés (HMM). Le système est analytique sans segmentation explicite et utilise une formation intégrée pour exécuter et améliorer les modèles de caractères. Le taux de reconnaissance atteignent 87.93%. [40]

### 1.8.2 Les travaux sur la base de données ADAB

Abdelazeem et al ont proposé un nouveau système de reconnaissance d'écriture manuscrite en ligne pour les noms de personnes arabes basé sur le modèle de Markov caché (HMM). Le taux de reconnaissance atteint 92.5%. [16]

Eraqi et Abdelazeem ont proposé un nouveau système de reconnaissance de l'écriture arabe en ligne basé sur une nouvelle technique de segmentation des graphèmes en ligne qui dépend de la direction d'écriture locale. Le taux de reconnaissance atteint 87%[28].

### 1.8.3 Tableau récapitulatif

Le tableau 1.1 résume les travaux connexes :

Auteurs	Méthode	Mode	Base de données	Précision
Azeem et al 2013. [21]	Multiple HMMs	Hors ligne	IFNIENIT Base de données	93.44%
Khmiri et al 2014. [32]	Horizontal-vertical Hidden Markov Model, Dynamic Bayesian Network	Hors ligne	IFNIENIT Base de données	92.19% with HMM and 88.82% DBN
Khémiri et al 2019. [31]	Dynamic Bayesian network (DBN) and Hidden Markov model (HMM)	Hors ligne	IFNIENIT Base de données	95.20%
Maalej and Monji 2018. [38]	Bidirectional Long Short-Term Memory (BLSTM), Connectionist Temporal Classification layer (CTC)	Hors ligne	IFNIENIT Base de données	92.21%
Jayech et al 2016. [30]	MSHMM	Hors ligne	IFNIENIT Base de données	91.10%
Rabi et al 2017. [39]	HMM/MLP	Hors ligne	IFNIENIT Base de données	89.03%

Auteurs	Méthode	Mode	Base de données	Précision
Elleuch et al 2016. [27]	Convolutional Neural Network (CNN), Support Vector Machine (SVM)	Hors ligne	IFNIENIT Base de données	92.95%
Tamen et al 2017. [44]	Multilayer perceptron, support vector machine and the Extreme Learning Machine	Hors ligne	IFNIENIT Base de données	96.82%
Rabi et al 2016. [40]	Hidden Markov Models (HMMs)	Hors ligne	IFNIENIT Base de données	87.93%
Abdelazeem et al 2011. [16]	Geometric features + holistic approach for delayedstroke detection + HTK	En ligne	ADAB for training 300 Arabic personal names for test	92.5%
Eraqi and Abdelazeem 2011. [28]	Grapheme segmentation + offline features + Fuzzy SVM	En ligne	ADAB Base de données	87%

TABLE 1.1 – Tableau récapitulatif sur les travaux connexes

## 1.9 Conclusion

Ce chapitre nous a permis de détailler le cadre général de notre travail en présentant une description générale des différents éléments et aspects du processus de reconnaissance de l'écriture manuscrite. Dans ce qui suit nous allons présenter une série de classificateurs couramment utilisés dans les systèmes de reconnaissance d'écriture et montrerons leurs avantages et leurs inconvénients.

## **Chapitre 2**

# **Les différentes techniques pour la reconnaissance de l'écriture**

## 2.1 Introduction

Dans l'ensemble du processus d'un système de reconnaissance des formes, la classification joue un rôle important pour décider si un motif appartient à une classe. L'idée principale de la classification est d'attribuer des exemples inconnus (formes) à des classes prédéfinies en fonction de la description dans le paramètre de forme. Dans ce chapitre, nous présenterons une série de classifieurs couramment utilisés dans les systèmes de reconnaissance d'écriture et montrerons leurs avantages et leurs inconvénients.

## 2.2 Classification

La classification est une technique dans laquelle nous catégorisons les données en un nombre donné de classes. L'objectif principal d'un problème de classification est d'identifier la catégorie/classe à laquelle une nouvelle donnée appartiendra. La classification peut être effectuée sur des données structurées ou non structurées.

Les terminologies rencontrées en machine learning – classification :

### 2.2.1 Classificateur

algorithme qui mappe les données d'entrée à une catégorie spécifique.

### 2.2.2 Modèle de classification

un modèle de classification tente de tirer des conclusions à partir des valeurs d'entrée fournies pour la formation. Il prédira les étiquettes/catégories de classe pour les nouvelles données.

### 2.2.3 Caractéristique

une caractéristique est une propriété individuelle mesurable d'un phénomène observé.

### 2.2.4 Classification binaire

tâche de classification avec deux résultats possibles. Ex : Classification par sexe (Homme / Femme).

### 2.2.5 Classement multi-classes

Classement à plus de deux classes. Dans la classification multi-classes, chaque échantillon est affecté à une et une seule étiquette cible. Ex : Un animal peut être chat ou chien mais pas les deux à la fois

### 2.2.6 Classification multi-étiquettes

Tâche de classification dans laquelle chaque échantillon est associé à un ensemble d'étiquettes cibles (plus d'une classe). Par exemple : un article de presse peut porter à la fois sur le sport, une personne et un lieu.

Ci-dessous les classificateurs les plus utilisés pour la reconnaissance de caractères :

## 2.3 Réseaux de neurones artificiels

Réseaux de neurones artificiels (RNA), un exemple est donné dans la figure 2.1 ci-dessous, ces moyens sont appelés réseaux connexionnistes. Ce sont des moyens hautement connectés de processeurs élémentaires distribués fonctionnant en parallèle pour l'apprentissage et la classification [20].

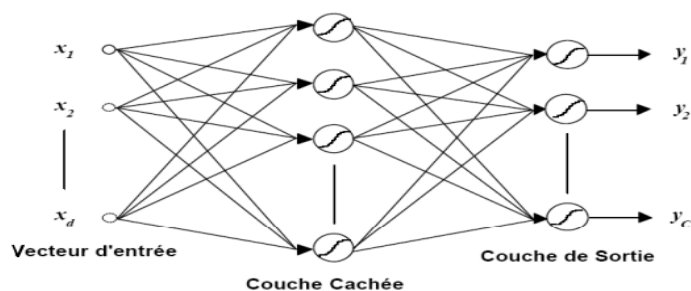


FIGURE 2.1 – Schéma de réseaux de neurones artificiels [20]

L'idée principale est que chaque processeur calcule une sortie à partir des informations qu'il reçoit en entrée. Le processeur est physiquement représenté par un neurone

formel, un modèle mathématique inspiré des modèles des biologistes. (voir la figure 2.2)

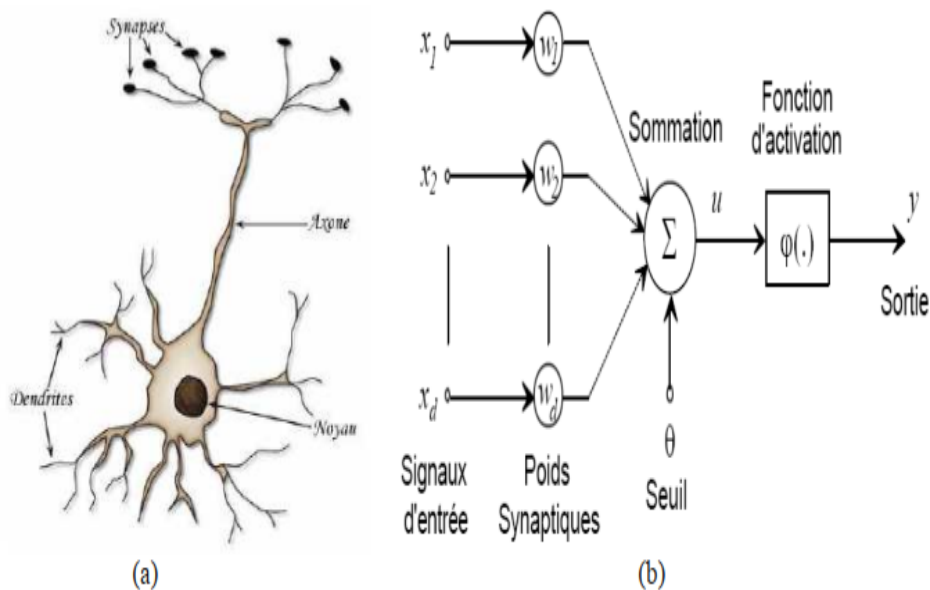


FIGURE 2.2 – La démarche de modélisation de RNA [20]

Les neurones formels sont des automates à  $n$  entrées réelles. Son traitement consiste à effectuer en sortie le résultat d'une fonction de seuil d'une somme pondérée de ses entrées. Récemment, les réseaux de neurones ont connu un grand succès dans plusieurs domaines, notamment la reconnaissance, grâce à la simplicité et à l'efficacité de leurs algorithmes d'apprentissage.

Plusieurs méthodes de formation sont utilisées pour apprendre les ANN, telles que l'apprentissage par correction d'erreurs, l'apprentissage Hebb et l'apprentissage compétitif. L'objectif de l'apprentissage d'un réseau de neurones est de lui apprendre à fournir une sortie souhaitée pour un ensemble de valeurs d'entrée. Pour ce faire, nous nous appuyons sur un grand nombre d'exemples qui peuvent ajuster les paramètres - les poids des connexions - afin d'obtenir la sortie souhaitée en fonction de l'entrée. Le réseau de neurones artificiels peut être divisé en deux catégories selon le flux d'informations et sa structure topologique : réseau de neurones à propagation directe dans lequel la direction des données est directe sans retour en arrière, et réseau de neurones récurrent.



Actuellement, les types les plus couramment utilisés dans les systèmes de reconnaissance de l'écriture manuscrite sont le Perceptron multicouche (MLP), la propagation directe, la mémoire associative ou les "cartes auto-organisatrices" de Kohonen Map (SOM), qui sont des RNA récurrents. Ils permettent de détecter automatiquement des prototypes de personnages dans un ensemble d'exemples d'apprentissage. Il existe également des réseaux à fonction de base radiale, des réseaux à délai, des réseaux convolutifs, des réseaux profonds (DBN) et des réseaux Diabolo. Bien que très discriminants, ils présentent également certains inconvénients, tels que :

- la difficulté à contrôler le comportement du réseau lors de l'apprentissage;
- les problèmes de dimensionnalité (on ne peut pas connaître a priori les dimensions du réseau pour un problème donné);
- le risque de sur apprentissage et le problème de convergence vers des minima locaux;
- la détermination des paramètres de réglage du poids de connexion est très fine.

La figure 2.3 représente l'organigramme de RNA. Le processus utilisé pour former le réseau est appelé un algorithme d'apprentissage, dont le travail consiste à modifier les poids de jonction du réseau pour obtenir l'objectif souhaité. [42]

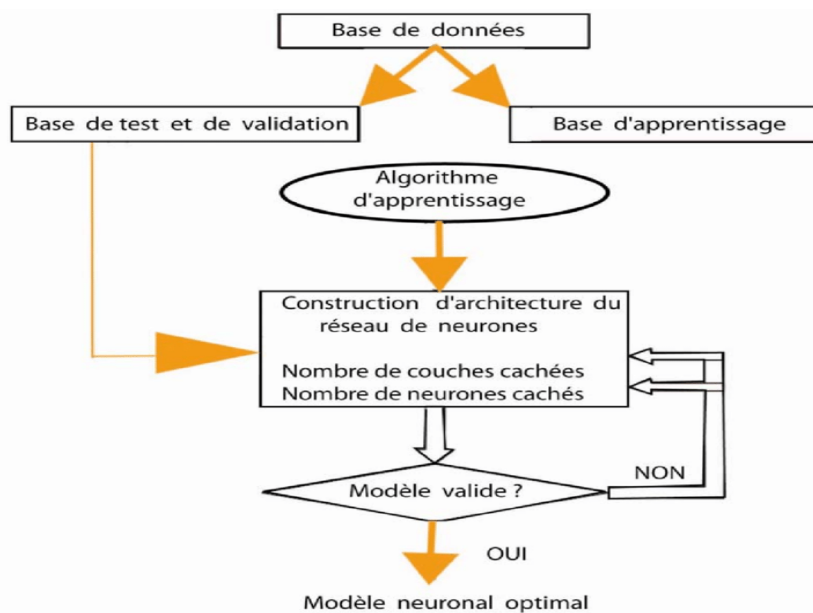


FIGURE 2.3 – L'organigramme de RNA [42]

### 2.3.1 Réseau de neurone monocouche Perceptron

le perceptron simple est dit simple parce qu'il ne dispose que de deux couches ; la couche en entrée et la couche en sortie. Le réseau est déclenché par la réception d'une information en entrée. Le traitement de la donnée dans ce réseau se fait entre la couche d'entrée et la couche de sortie qui sont toutes reliées entre elles. Le réseau intégral ne dispose ainsi que d'une matrice de poids. Le fait de disposer d'une seule matrice de poids limite le perceptron simple à un classificateur linéaire permettant de diviser l'ensemble d'informations obtenues en deux catégories distinguées voir la figure 2.4.

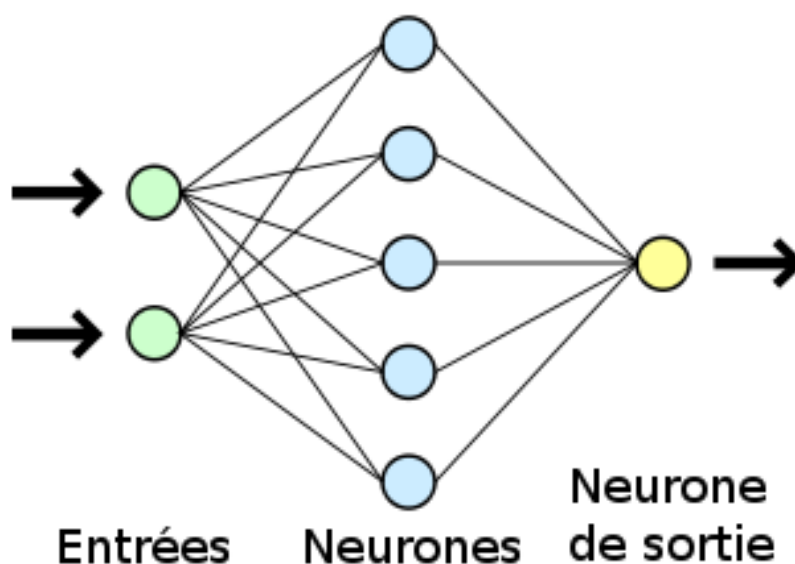


FIGURE 2.4 – Schéma de Réseau de neurone monocouche perceptron [42]

### 2.3.2 Réseau de neurone multicouche Perceptron (MLP)

Le perceptron multicouche se structure de la même façon. L'information entre par une couche d'entrée et sort par une couche de sortie. À la différence du perceptron simple, le perceptron multicouche dispose entre la couche en entrée et la couche en sortie une ou plusieurs couches dites « cachées ». Le nombre de couches correspond aux nombres de matrices de poids dont disposent le réseau. Un perceptron multicouche est donc mieux adapté pour traiter les types de fonctions non-linéaires voir la figure 2.5.

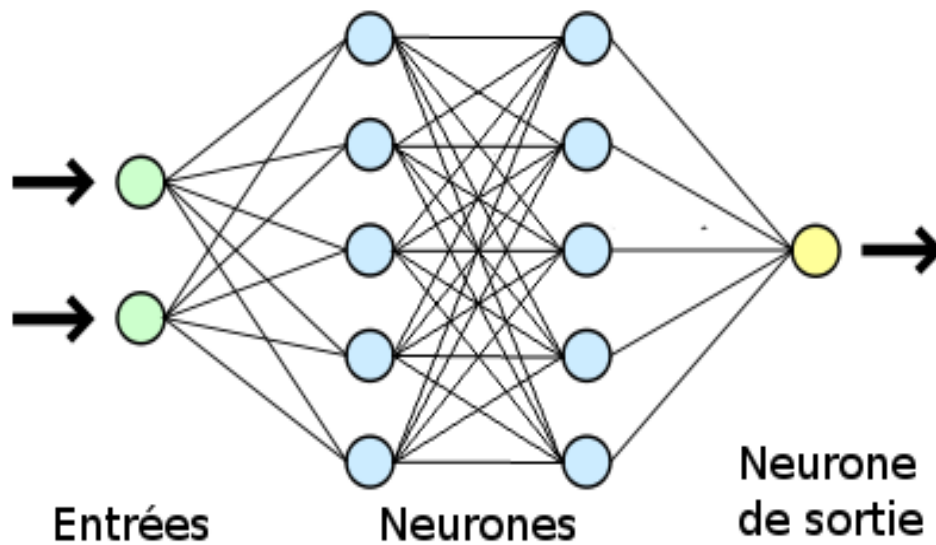


FIGURE 2.5 – Schéma de MLP [42]

### 2.3.3 Avantages [1]

- Les problèmes dans RNA sont représentés par des paires attribut-valeur.
- Les RNA sont utilisés pour les problèmes ayant la fonction cible, la sortie peut être à valeur discrète, à valeur réelle ou un vecteur de plusieurs attributs à valeur réelle ou discrète.
- Les méthodes d'apprentissage RNA sont assez robustes au bruit dans les données d'apprentissage. Les exemples de formation peuvent contenir des erreurs qui n'affectent pas la sortie finale.
- Il est utilisé lorsque l'évaluation rapide de la fonction cible apprise est requise.
- Les RNA peuvent supporter de longs temps de formation en fonction de facteurs tels que le nombre de poids dans le réseau, le nombre d'exemples de formation pris en compte et les réglages de divers paramètres d'algorithme d'apprentissage.

### 2.3.4 Désavantages [1]

#### 2.3.4.1 Dépendance matérielle

Les réseaux de neurones artificiels nécessitent des processeurs avec une puissance de traitement parallèle, de par leur structure.

Pour cette raison, la réalisation de l'équipement est dépendante.

#### 2.3.4.2 Fonctionnement inexplicé du réseau

C'est le problème le plus important d'RNA.

Lorsque RNA donne une solution de sondage, il ne donne aucun indice sur le pourquoi et le comment.

Cela réduit la confiance dans le réseau.

#### 2.3.4.3 Assurance de la bonne structure du réseau

Il n'y a pas de règle spécifique pour déterminer la structure des réseaux de neurones artificiels.

La structure de réseau appropriée est obtenue grâce à l'expérience et aux essais et erreurs.

#### 2.3.4.4 La difficulté de montrer le problème au réseau

Les RNA peuvent travailler avec des informations numériques.

Les problèmes doivent être traduits en valeurs numériques avant d'être introduits dans RNA.

Le mécanisme d'affichage à déterminer influencera directement les performances du réseau.

Cela dépend de la capacité de l'utilisateur.

#### 2.3.4.5 La durée du réseau est inconnue

Le réseau est réduit à une certaine valeur de l'erreur sur l'échantillon signifie que la formation est terminée.

La valeur ne nous donne pas des résultats optimaux.

## 2.4 Réseau neuronal convolutif

Les réseaux de neurones convolutifs sont apparus avec Alex Krizhevsky en 2012 [33], cette méthode était difficile pour un ordinateur. Heureusement, l'approche de ces réseaux inspirés de notre œil (en particulier, certains neurones de notre espace visuel ne répondent qu'à des frontières verticales, alors que d'autres répondent à des frontières horizontales/diagonales) a ouvert de nombreuses applications, que ce soit en imagerie médicale, voitures autonomes, visage reconnaissance et même l'analyse de texte.

Réseau de neurones convolutifs (CNN). C'est un algorithme d'apprentissage en profondeur qui est bon pour l'analyse d'images. En fait, il s'appuie sur des filtres et les réseaux convolutifs ont remporté un succès considérable dans les applications pratiques. Le nom "réseau neuronal convolutif" indique que le réseau emploie une opération mathématique appelée convolution. La convolution est un type particulier d'opération linéaire. Les réseaux convolutifs sont de simples réseaux de neurones qui utilisent des convolutions au lieu de la multiplication matricielle dans au moins une de leurs couches.

Ils ont de nombreuses applications dans la reconnaissance d'images et de vidéos, les systèmes de recommandation, le traitement du langage naturel et la reconnaissance d'écriture.

Les neurones dans les CNN fonctionnent en considérant une petite partie de l'image, appelons la sous-image. Les sous-images sont ensuite inspectées pour rechercher les fonctionnalités pouvant être reconnues par le réseau. À titre d'exemple simple, une aractéristique peut être une ligne verticale, une arche ou un cercle. Ces aractéristiques sont ensuite capturées par les cartes d'entités respectives du réseau.

Une combinaison de fonctionnalités est ensuite utilisée pour classer l'image. En outre, de multiples cartes de caractéristiques différentes sont utilisées pour rendre le réseau robuste à différents niveaux de contraste, de luminosité, de niveaux de saturation des couleurs, de bruit, etc.

Il existe deux types de couches, tous deux composés de cartes de caractéristiques. Le but de la couche convolutionnelle est de reconnaître les caractéristiques de l'image d'entrée.

Le réseau se compose généralement de plusieurs cartes de caractéristiques, chacune identifiant certaines entités. Les sous-images recouvrent intentionnellement les zones qui se chevauchent de l'image d'origine. Une telle conception est importante pour tolérer les distorsions d'image telles que la translation, la rotation, l'inclinaison, etc.

Un autre type de couche, appelée couche de sous-échantillonnage, suit toujours une couche convolutive. Il se compose du même nombre de cartes d'entités, et chaque carte de la couche convolutive est utilisée comme entrée de la carte d'entités correspondante dans la couche de sous-échantillonnage suivante. Les sous-images couvertes par les neurones de la couche de sous-échantillonnage ne se chevauchent généralement pas.

En fonction de la profondeur du réseau, les couches de convolution et de sous-échantillonnage alternent jusqu'à ce que la dernière couche de sous-échantillonnage soit atteinte.

Après la dernière couche de sous-échantillonnage, il peut y avoir un nombre quelconque de couches entièrement connectées, la dernière d'entre elles étant la couche de sortie. La figure 2.6 représente l'architecture standard de CNN.

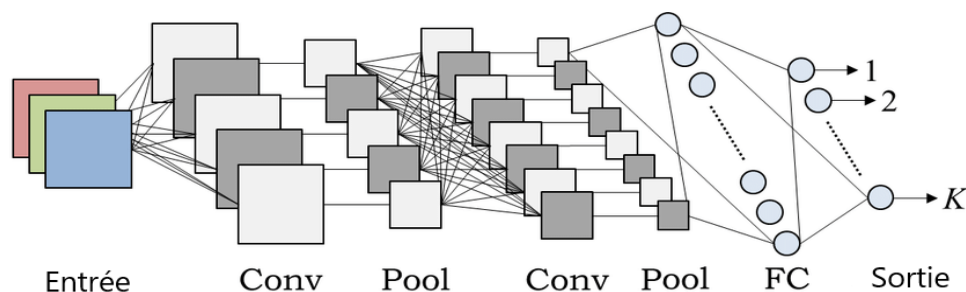


FIGURE 2.6 – L'architecture standard de CNN

### 2.4.1 Couche de convolution

L'objectif de l'opération de convolution est d'extraire les caractéristiques de haut niveau telles que les bords, à partir de l'image d'entrée. [41]

#### 2.4.1.1 Filtre

L'élément impliqué dans la réalisation de l'opération de convolution dans la première partie d'une couche convolutive est appelé le noyau/filtre. Il se déplace à travers la matrice en effectuant une opération de multiplication. [41]

#### 2.4.1.2 Foulée (Stride)

la foulée est la valeur des décalages du filtre à travers la matrice d'entrée. [41]

### 2.4.2 Couche de POOLING

la couche de pooling est responsable de la réduction de la taille spatiale de l'entité convoluée. [41]

### 2.4.3 Avantages [11]

- Il convient aux applications d'apprentissage automatique et d'Intelligence artificielle avec de grandes quantités de données d'entrée telles que la reconnaissance d'images.
- Le réseau fonctionne de manière robuste et est insensible à la distorsion ou à d'autres changements optiques.
- Il peut traiter des images enregistrées dans différentes conditions d'éclairage et dans différentes perspectives. Les caractéristiques typiques d'une image sont ainsi facilement identifiées.
- Il nécessite beaucoup moins d'espace de stockage que les réseaux de neurones entièrement maillés. Le CNN est divisé en plusieurs couches locales partiellement maillées. Les couches de convolution réduisent considérablement les besoins de stockage.
- Le temps de formation d'un CNN est également considérablement réduit. Grâce à l'utilisation de processeurs graphiques modernes, les CNN peuvent être formés de manière très efficace.

- Il est la technologie de pointe pour le Deep Learning et la classification dans la reconnaissance d'images (image recognition).

#### 2.4.4 Désavantages

- CNN n'encode pas la position et l'orientation de l'objet.
- Manque de capacité à être spatialement invariant aux données d'entrée.
- De nombreuses données d'entraînement sont nécessaires.
- Il faut beaucoup de temps pour former un réseau de neurones convolutifs, en particulier avec de grands ensembles de données. Vous avez généralement besoin de matériel spécialisé (comme un GPU) pour accélérer le processus de formation.

## 2.5 Réseau neuronal récurrent

Les réseaux de neurones récurrents (RNN) sont un type de NN largement utilisé pour effectuer le processus d'analyse de séquence car le RNN est conçu pour extraire les informations contextuelles en définissant les dépendances entre les différents horodatages. RNN se compose de nombreuses couches récurrentes successives, et ces couches sont séquentiellement modélisées afin de cartographier la séquence avec d'autres séquences. RNN a une forte capacité à capturer les données contextuelles de la séquence. Cependant, les indices contextuels dans la structure du réseau sont stables et sont efficacement utilisés pour réaliser le processus de classification des données.[34]

RNN est l'extension de feedforward NN avec la présence de boucles dans les couches cachées. RNN prend l'entrée avec la séquence d'échantillons et identifie la relation temporelle entre les échantillons voir la figure 2.7.

Un réseau neuronal récurrent est un type de réseau neuronal artificiel (RNA) et est utilisé dans les domaines d'application du traitement du langage naturel (TAL) et de la reconnaissance vocale. Un modèle RNN est conçu pour reconnaître les caractéristiques séquentielles des données et ensuite utiliser les modèles pour prédire le scénario à venir.



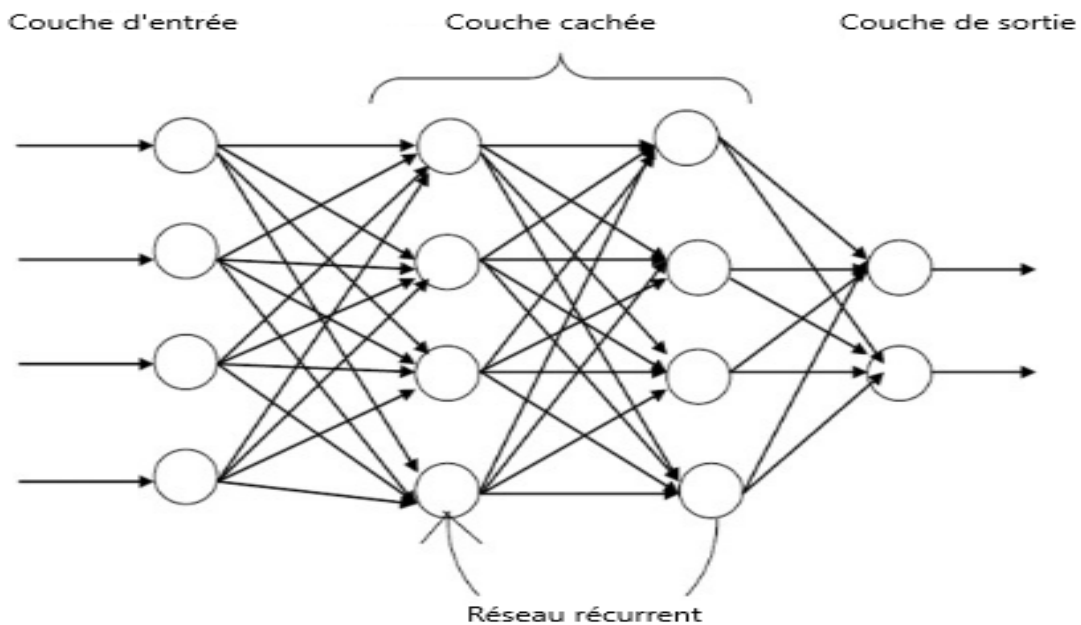


FIGURE 2.7 – Schéma de Réseau neuronal récurrent [34]

### 2.5.1 Réseau de mémoire à long court terme (LSTM) :

Long Short Term Memory Network (LSTM) est un RNN avancé, un réseau séquentiel, qui permet aux informations de persister. Il est capable de gérer le problème de gradient de fuite auquel est confronté RNN. Un réseau de neurones récurrent est également connu sous le nom de RNN et est utilisé pour la mémoire persistante. [8]

### 2.5.2 Avantages [12]

- RNN peut traiter des entrées de n'importe quelle longueur.
- Un modèle RNN est modélisé pour mémoriser chaque information tout au long du temps, ce qui est très utile dans tout prédicteur de série chronologique.
- Même si la taille d'entrée est plus grande, la taille du modèle n'augmente pas.
- Les poids peuvent être répartis sur les pas de temps.
- Les RNN peuvent utiliser leur mémoire interne pour traiter la série arbitraire d'entrées, ce qui n'est pas le cas avec les réseaux de neurones à anticipation.

### 2.5.3 Désavantages [12]

- En raison de sa nature récurrente, le calcul est lent.
- La formation des modèles RNN peut être difficile.
- Si nous utilisons relu ou tanh comme fonctions d'activation, il devient très difficile de traiter des séquences très longues.
- Sujet à des problèmes tels que l'explosion et la disparition du gradient.

## 2.6 Classification temporelle connexionniste (CTC)

La classification temporelle connexionniste (CTC) est un type de sortie de réseau de neurones utile pour résoudre les problèmes de séquence tels que l'écriture manuscrite et la reconnaissance vocale où la synchronisation varie. L'utilisation de CTC garantit que l'on n'a pas besoin d'un ensemble de données aligné, ce qui rend le processus de formation plus simple. [15]

CTC travaille sur les trois grands concepts suivants : [15]

### 2.6.1 Encodage du texte

CTC fusionne tous les caractères répétés en un seul caractère. Par exemple, si le mot dans l'image est "hey" où "h" prend trois pas de temps, "e" et "y" prennent un pas de temps chacun. Ensuite, la sortie du réseau utilisant CTC sera "hhhey", qui, selon notre schéma d'encodage, est réduite à "hey".

Pour les mots avec des lettres répétées, CTC introduit un pseudo-caractère appelé blanc noté « - » dans les exemples suivants. Lors de l'encodage du texte, si un caractère se répète, un espace est placé entre les caractères dans le texte de sortie. Considérons le mot 'meet', les encodages possibles seront, 'mm-ee-ee-t', 'mmm-e-e-ttt', le mauvais encodage sera 'mm-eee-tt', car il en résultera dans 'met' lorsqu'il est décodé. Le CRNN est formé pour produire le texte codé.

### 2.6.2 Calcul des pertes

Pour former le CRNN, on devons calculer la perte en fonction de l'image et de son étiquette. Une matrice du score est obtenue pour chaque caractère à chaque pas de temps du CRNN. La figure 2.8 montre un exemple de matrice de sortie du CRNN. Il y a trois pas de temps et trois caractères (dont un blanc). A chaque pas de temps, le score du caractère totalise 1.

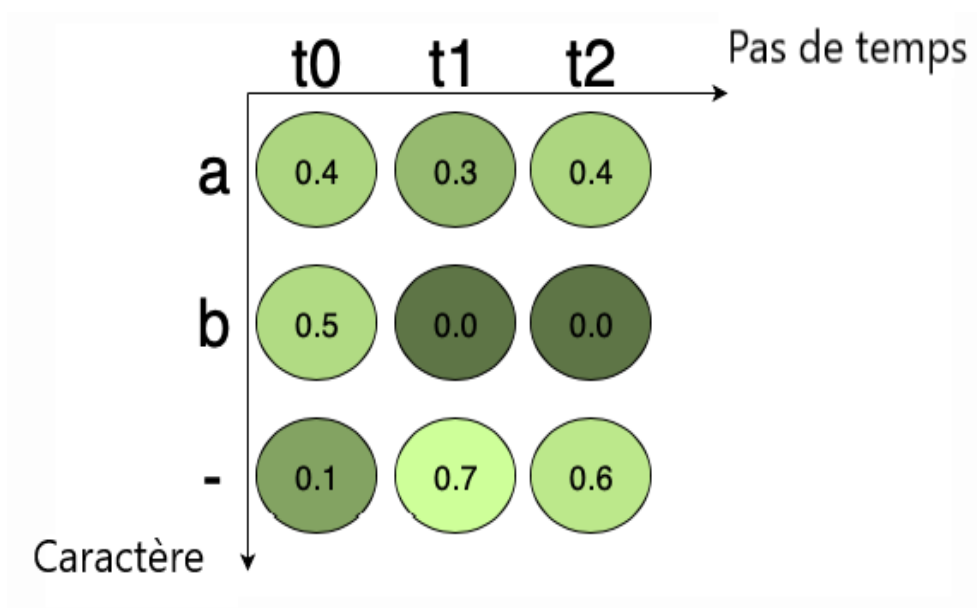


FIGURE 2.8 – La matrice de sortie du réseau de neurones [15]

Pour calculer la perte, tous les scores des alignements possibles de la vérité terrain sont additionnés. De cette manière, l'emplacement du caractère dans l'image n'est pas significatif.

Les scores des caractères correspondants sont multipliés ensemble pour obtenir le score d'un chemin. Dans la Fig.3, le score pour le chemin "a-" est  $0,4 \times 0,7 \times 0,6 = 0,168$ , et pour le chemin "aaa" est  $0,4 \times 0,3 \times 0,4 = 0,048$ . Pour obtenir le score correspondant à la vérité terrain donnée, les scores de tous les chemins vers le texte correspondant sont additionnés.

### 2.6.3 Décodage

Une fois que CRNN est formé, nous voulons qu'il nous donne une sortie sur des images de texte invisibles. En d'autres termes, nous voulons le texte le plus probable étant donné une matrice de sortie du CRNN. Une méthode peut consister à examiner chaque sortie de texte potentielle, mais ce ne sera pas très pratique du point de vue du calcul. L'algorithme du meilleur chemin est utilisé pour surmonter ce problème.

Il se compose des deux étapes suivantes :

1. Calcule le meilleur chemin en considérant le caractère avec la probabilité maximale à chaque pas de temps.
2. La deuxième étape consiste à supprimer les blancs et les caractères en double, ce qui donne le texte réel.

## 2.7 Modèle de Markov caché (HMM)

Le modèle de Markov caché (HMM) est un modèle statistique qui a été proposé pour la première fois par Baum L.E. (Baum et Petrie, 1966) [22] et utilise un processus de Markov qui contient des paramètres cachés et inconnus. Dans ce modèle, les paramètres observés sont utilisés pour identifier les paramètres cachés. Ces paramètres sont ensuite utilisés pour une analyse plus approfondie. Le HMM est un type de chaîne de Markov. Son état ne peut pas être observé directement mais peut être identifié en observant la série de vecteurs. Depuis les années 1980, HMM a été utilisé avec succès pour la reconnaissance vocale, la reconnaissance de caractères et les techniques de communication mobile. Il a également été rapidement adopté dans des domaines tels que la bio-informatique et le diagnostic de pannes. Le principe de base du HMM est que les événements observés n'ont pas de correspondance univoque avec les états, mais sont liés aux états par le biais de la distribution de probabilité.

Il s'agit d'un processus doublement stochastique, qui inclut une chaîne de Markov comme processus stochastique de base, et décrit les transitions d'état et les processus stochastiques qui décrivent la correspondance statistique entre les états et les valeurs observées. Du point de vue des observateurs, seule la valeur observée peut être visualisée, contrairement aux états. Un processus stochastique est utilisé pour identifier l'existence d'états et leurs caractéristiques. On parle alors de modèle de Markov « caché ».

Des méthodes statistiques sont utilisées pour construire des changements d'état dans HMM afin de comprendre les tendances les plus possibles dans les données de surveillance. HMM peut ajuster automatiquement et de manière flexible les tendances, les éléments saisonniers, covariants et distributionnels. Le HMM a été utilisé dans de nombreuses études sur les données de surveillance de séries chronologiques.

Les modèles de Markov cachés (HMM) sont des modèles de séquence. Autrement dit, étant donné une séquence d'entrées, telles que des mots, un HMM calculera une séquence de sorties de même longueur. Un modèle HMM est un graphe où les nœuds sont des distributions de probabilité sur les étiquettes et les arêtes donnent la probabilité de transition d'un nœud à l'autre. Ensemble, ceux-ci peuvent être utilisés pour calculer la probabilité d'une séquence d'étiquettes compte tenu de la séquence d'entrée.

C'est pour cela les méthodes de Markov sont bien adaptées à la modélisation de données séquentielles. Par conséquent, ils sont principalement utilisés pour la reconnaissance automatique de l'écriture manuscrite.

Les HMM présentent plusieurs avantages dans la reconnaissance de l'écriture manuscrite. En effet, ils tiennent compte de la variabilité des formes et des bruits qui interfèrent avec l'écriture, notamment dans les manuscrits. De plus, ils peuvent également prendre en compte des séquences de longueur variable. Cette qualité est particulièrement importante dans la reconnaissance de l'écriture manuscrite, car la longueur des lettres et la longueur des mots peuvent varier considérablement en fonction du style d'écriture et des habitudes de l'auteur. Deux séquences de la même classe ne sont pas forcément de longueurs identiques. Finalement ils s'appuient sur un ensemble d'algorithmes standards et éprouvés.

L'implémentation de ces algorithmes s'appuie sur des techniques bien maîtrisées de la programmation dynamique. En plus, un certain nombre de bibliothèques HTK, GHMM et ESMEALDA et de modules sont également publiquement accessibles pour l'apprentissage et le décodage de Modèles de Markov Cachés.

En reconnaissance de l'écriture, les graphèmes, les caractères ou les mots sont souvent modélisés par des états cachés dont leur succession est donnée par une chaîne de Markov pour laquelle quelques exemples sont donnés dans la Figure 2.9 ci-dessous.

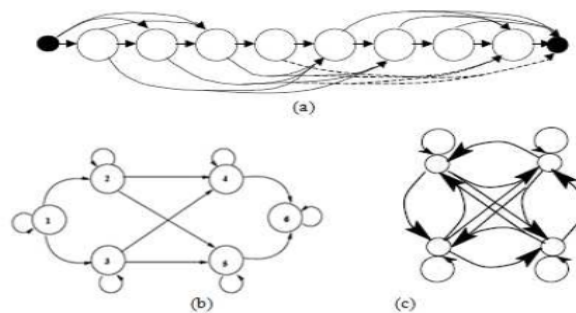


FIGURE 2.9 – Exemples de HMM [20]

Le schéma fonctionnel de l'algorithme d'apprentissage est donné sur la figure 2.10 où N représente le nombre d'états des modèles et M le nombre de mélanges par état. [35]

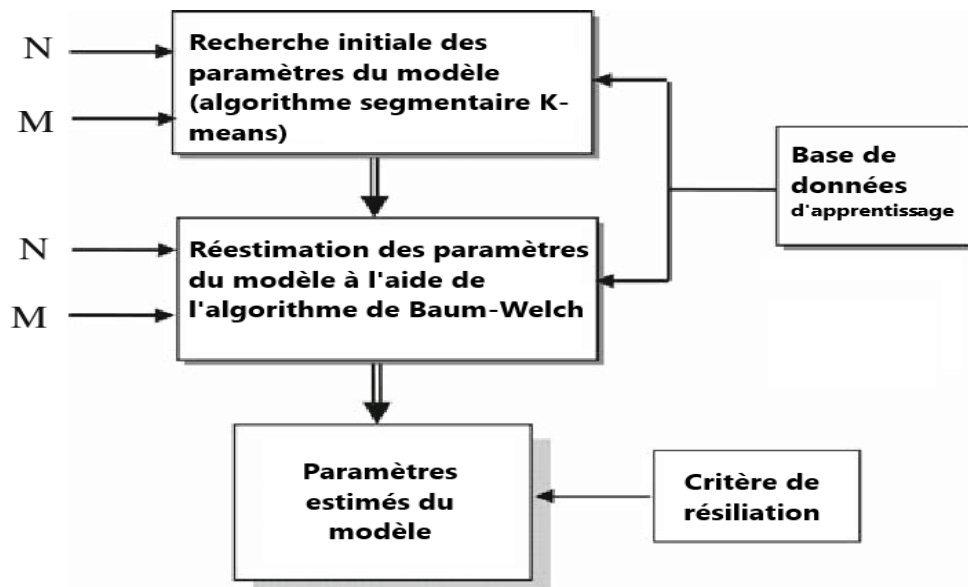


FIGURE 2.10 – L'algorithme d'apprentissage de HMM [35]

### 2.7.1 Avantages

- La base théorique sous-jacente est beaucoup plus solide, élégante et facile à comprendre.
- Il est plus facile à mettre en œuvre et à analyser.
- Les étiqueteurs HMM sont très simples à former (il suffit de compiler les décomptes à partir du corpus d'apprentissage).
- Performances relativement bonnes (plus de 90 % de performances sur les entités nommées).
- Les statisticiens sont à l'aise avec la base théorique du HMM.
- Liberté de manipuler les processus de formation et de vérification.
- Analyse mathématique/théorique des résultats et des processus.
- Intègre les connaissances préalables dans l'architecture avec une bonne conception.
- Initialiser le modèle près de quelque chose que l'on croit être correct.
- Il élimine le problème de biais d'étiquette.
- Il s'est également avéré efficace pour un certain nombre d'autres tâches, telles que la reconnaissance vocale, la reconnaissance de l'écriture manuscrite et la reconnaissance de la langue des signes.
- Parce que chaque HMM n'utilise que des données positives, ils évoluent bien ; puisque de nouveaux mots peuvent être ajoutés sans affecter les HMM appris.

### 2.7.2 Désavantages

- Afin de définir la probabilité conjointe sur l'observation et la séquence d'étiquettes, HMM doit énumérer toutes les séquences d'observation possibles.
- La difficulté principale est la modélisation de la probabilité d'attribuer une étiquette à un mot qui peut être très difficile si les « mots » sont complexes.
- Il n'est pas pratique de représenter plusieurs entités qui se chevauchent et des dépendances à long terme.
- Le nombre de paramètres à évaluer est énorme. Il a donc besoin d'un grand ensemble de données pour la formation.

- Il nécessite énormément de formation afin d'obtenir de meilleurs résultats.
- Les HMM n'utilisent que des données positives pour s'entraîner. En d'autres termes, la formation HMM consiste à maximiser les probabilités observées pour les exemples appartenant à une classe. Mais cela ne minimise pas la probabilité d'observation d'instances d'autres classes.
- Il adopte l'hypothèse Markovienne : que les probabilités d'émission et de transition ne dépendent que de l'état actuel, qui ne correspond pas bien à de nombreux domaines du monde réel .

## 2.8 Machine à vecteurs de support (SVM)

Ces méthodes sont des classificateurs à deux classes introduites par le mathématicien Vladimir Vapnik [25] avec de fortes capacités de généralisation. SVM est un ensemble d'algorithmes d'apprentissage capables de distinguer des formes. L'idée principale est que deux classes peuvent être séparées linéairement dans un espace de grande dimension, comme le montre la figure 2.11 ci-dessous. Dans le cas où les points sont séparables, on trouve souvent une infinité d'hyperplans séparants [20].

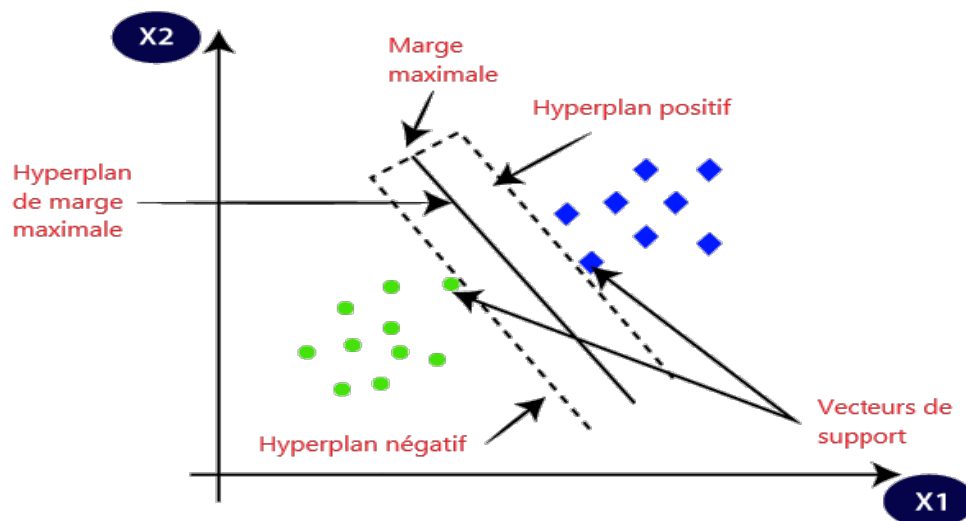


FIGURE 2.11 – Exemple de SVM [20]



Les SVM sont des modèles discriminants qui tentent de minimiser les erreurs d'apprentissage tout en maximisant la marge entre les classes, l'espace sans exemple autour de la frontière de décision. Pour ce faire, l'algorithme d'apprentissage choisit judicieusement un certain nombre de « vecteurs supports » parmi les exemples de la base d'apprentissage, qui définissent la frontière de décision optimale.

La figure 2.12 montre un diagramme schématique du processus de classification utilisant la méthode SVM, qui comprend un processus de formation et de test. Les noyaux SVM utilisent des polynômes et une fonction de base radiale (RBF). Ces choix sont considérés comme un mécanisme de classification efficace puisque ces noyaux montrent une séparation non linéaire entre les classes. Pour garantir une grande précision de prédiction, le processus de validation croisée est effectué. Cela permet d'obtenir les meilleurs paramètres du noyau. [17]

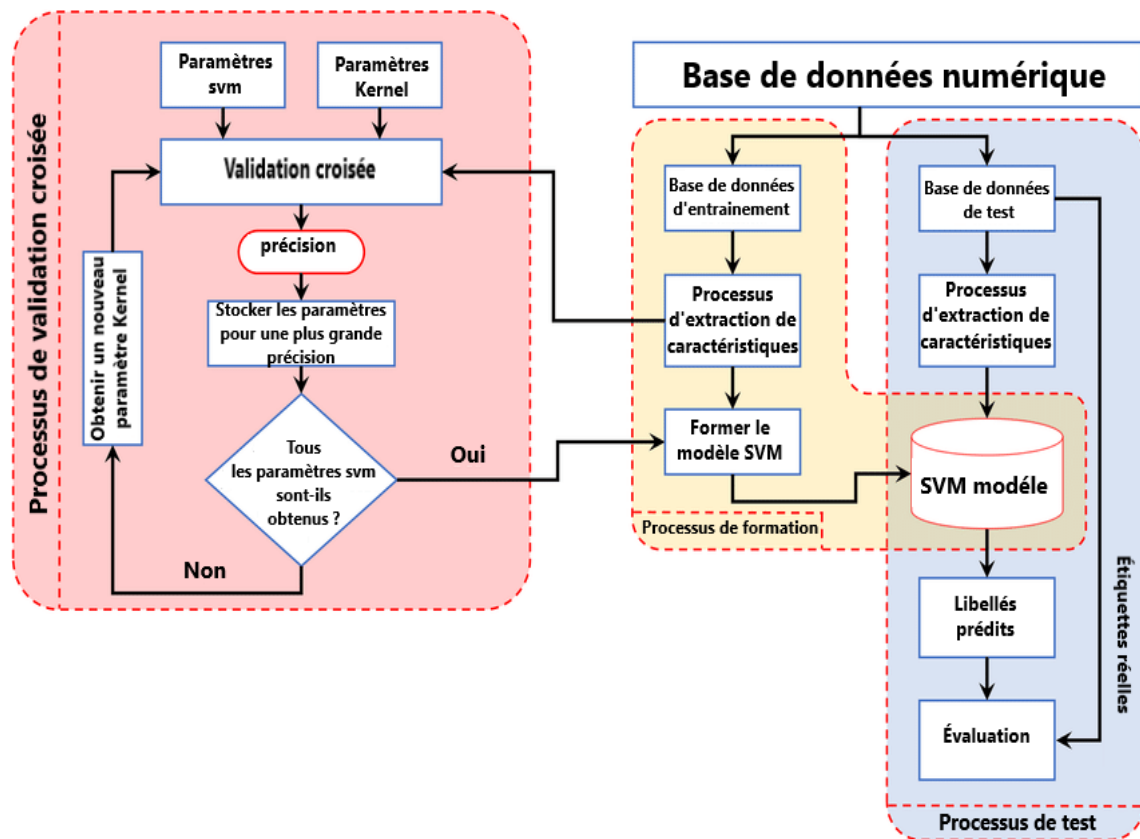


FIGURE 2.12 – Diagramme schématique du processus de classification des SVM [17]

### 2.8.1 Avantages

- SVM fonctionne relativement bien lorsqu'il existe une marge de séparation claire entre les classes.
- SVM est plus efficace dans les espaces de grande dimension.
- SVM est efficace dans les cas où le nombre de dimensions est supérieur au nombre d'échantillons.
- SVM est relativement économe en mémoire.

### 2.8.2 Désavantages

- L'algorithme SVM n'est pas adapté aux grands ensembles de données.
- SVM ne fonctionne pas très bien lorsque l'ensemble de données a plus de bruit, c'est-à-dire que les classes cibles se chevauchent.
- Dans les cas où le nombre de caractéristiques pour chaque point de données dépasse le nombre d'échantillons de données d'apprentissage, la SVM sera sous-performante.
- Comme le classificateur de vecteur de support fonctionne en plaçant des points de données, au-dessus et en dessous de l'hyperplan de classification, il n'y a pas d'explication probabiliste pour la classification.

## 2.9 K-means

L'algorithme Kmeans est un algorithme itératif qui tente de partitionner l'ensemble de données en K sous-groupes (clusters) distincts et non superposés prédéfinis, où chaque point de données appartient à un seul groupe. Il essaie de rendre les points de données intra-cluster aussi similaires que possible tout en gardant les clusters aussi différents (loin) que possible.

Il attribue des points de données à un cluster de sorte que la somme de la distance au carré entre les points de données et le centroïde du cluster (moyenne arithmétique de tous les points de données appartenant à ce cluster) soit au minimum. Moins nous avons de variation au sein des clusters, plus les points de données sont homogènes (similaires) au sein du même cluster. voir la figure 2.13

Le fonctionnement de l'algorithme kmeans est le suivant :

- 1-Spécifiez le nombre de clusters K.
- 2-Initialisez les centroïdes en mélangeant d'abord l'ensemble de données, puis en sélectionnant au hasard K points de données pour les centroïdes sans remplacement.
- 3-Continuez à itérer jusqu'à ce qu'il n'y ait aucun changement dans les centroïdes. c'est-à-dire que l'affectation des points de données aux clusters ne change pas.
- 4-Calculez la somme de la distance au carré entre les points de données et tous les centroïdes.
- 5-Attribuez chaque point de données au cluster le plus proche (centre de gravité).
- 6-Calculez les centroïdes des clusters en prenant la moyenne de tous les points de données appartenant à chaque cluster. [7]

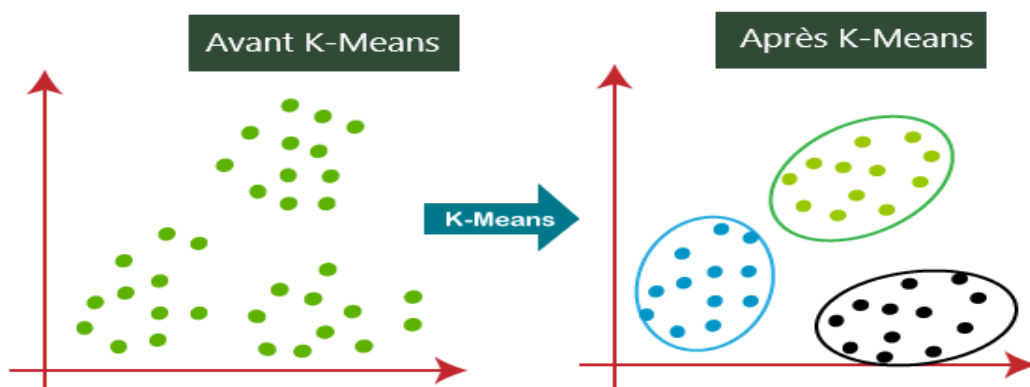


FIGURE 2.13 – Exemple de K-means [7]

### 2.9.1 Avantages [6]

- Relativement simple à mettre en œuvre.
- S'adapte à de grands ensembles de données.
- Garantit la convergence.
- Peut démarrer à chaud les positions des centroïdes.
- S'adapte facilement aux nouveaux exemples.
- Se généralise aux clusters de différentes formes et tailles, tels que les clusters elliptiques.

### 2.9.2 Disavantages [6]

- Choisir k manuellement.
- Être dépendant des valeurs initiales.
- Regroupement de données de différentes tailles et densités.
- Mise à l'échelle avec nombre de dimensions.

## 2.10 Algorithme génétique

Les algorithmes génétiques (GA) appartiennent à la famille des algorithmes évolutionnistes. Les principaux opérateurs de GA sont la sélection, croisement et mutation, GA s'inspire de la théorie darwinienne de l'évolution, dans laquelle la survie de créatures plus en forme et leurs gènes ont été simulés. Le but des GA est d'obtenir une solution approchée à un problème d'optimisation, lorsqu'il n'existe pas de méthode exacte (ou que la solution est inconnue) pour le résoudre en un temps raisonnable. Les algorithmes génétiques utilisent la notion de sélection naturelle et l'appliquent à une population de solutions potentielles au problème donné. La solution est approchée par « bonds » successifs, comme dans une procédure de séparation et évaluation (branch & bound), à ceci près que ce sont des formules qui sont recherchées et non plus directement des valeurs[4].

### 2.10.1 Notion de sélection naturelle

Le processus de sélection naturelle commence par la sélection des individus les plus aptes à partir d'une population. Ils produisent une progéniture qui hérite des caractéristiques des parents et sera ajoutée à la génération suivante. Si les parents ont une meilleure forme physique, leur progéniture sera meilleure que les parents et aura de meilleures chances de survivre. Ce processus continue d'itérer et à la fin, une génération avec les individus les plus aptes sera trouvée.

Cette notion peut être appliquée à un problème de recherche. Nous considérons un ensemble de solutions pour un problème et sélectionnons l'ensemble des meilleures d'entre elles.

Il y a cinq étapes dans l'algorithme génétique :

- Population initiale.
- Fonction de remise en forme.
- Sélections.
- Croisement.
- Mutation.

### 2.10.2 Avantages [4]

- le concept est facile à comprendre.
- AG recherche à partir d'une population de points, pas d'un seul point.
- AG utiliser une information de gain (fonction objective) et non des dérivés.
- AG prend en charge l'optimisation multi-objectifs.
- AG utiliser des règles de transition probabilistes, et non des règles déterministes.
- AG est bon pour les environnements bruyants.
- AG est facilement parallélisé.
- AG peut fonctionner sur diverses représentations.
- AG est stochastique.
- AG fonctionne bien sur un problème mixte discret/continu.

### 2.10.3 Désavantages [4]

- AG nécessite moins d'informations sur le problème, mais concevoir une fonction objective et obtenir la représentation et les opérateurs corrects peut être difficile.
- AG peut prendre du temps.

## 2.11 Conclusion

A la fin de ce chapitre et après avoir eu un aperçu du comportement des classifieurs (RNA, CNN, RNN, CTC, SVM, K-means, HMM, AG), nous avons pu comprendre leur fonctionnement et cela nous a permis d'entamer la phase d'implémentation. Dans ce qui suit, nous présenterons le travail que nous avons réalisé.

## **Chapitre 3**

# **Modélisation et Implémentation d'un système de reconnaissance de mot arabe écrit manuscritement**

## 3.1 Introduction

Dans ce projet, nous avons réalisé une application WEB pour la reconnaissance de l'écriture manuscrite arabe en utilisant le langage de programmation (python), dans ce chapitre nous allons expliquer le principe de fonctionnement et comment utiliser notre système, Puis on va présenter les différentes interfaces qui la composent pour avoir une vision générale sur le comportement du système avec quelques exemples.

## 3.2 Environnement du développement

Pour le développement de notre applicatio; nous avons utilisé les outils matériels et logiciels suivants :

### 3.2.1 Outils matériels

On a utilisé un ordinateur personnel pour la réalisation de l'application.

#### Caractéristiques de la machine utilisée

- Mémoire centrale RAM : 8GO.
- Processeur : Intel(R) Core(TM) i5-2450M CPU @ 2.50 GHz 2.50 GHz.
- Disque dur : 1TO.
- Système d'exploitation : Windows 10 Professionnel.

### 3.2.2 Outils Logiciels

- Environnement de développement intégré : visual Studio.
- Langage de programmation : Pythone 3.10 .
- Développement WEB : Flask 2.1.1.
- La rédaction : Latex.
- Bibliographie : Zotero.



### 3.2.3 Bibliothèques

Pour le développement de notre système, nous avons utilisé, les Bibliothèques suivants :

#### Tensorflow

On a utilisé tensorflow 2.8.0, ce dernier est une plate-forme Open Source de bout en bout dédiée au machine learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie. [13]

#### Numpy

On a utilisé NumPy 1.21.5, ce dernier est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que des tableaux masqués et des matrices) et un assortiment de routines pour des opérations rapides sur des tableaux, y compris mathématiques, logiques, manipulation de forme, tri, sélection, E/S, transformées de Fourier discrètes, algèbre linéaire de base, opérations statistiques de base, simulation aléatoire et bien plus encore. [9]

#### Editdistance

On a utilisé editdistance 0.6.0, Cette bibliothèque est utilisée pour ajuster la distance (distance de Levenshtein) implémente simplement avec C++ et python. [3]

#### Opencv

On a utilisé OpenCV 4.5.5, cette bibliothèque (Open Source Computer Vision Library) est une open-source qui comprend plusieurs centaines d'algorithmes de vision par ordinateur.[10]

### 3.2.4 Développement WEB

#### 3.2.4.1 Flask

Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger. [14]

### 3.3 Base de données

Il existe de nombreuses bases de données utilisées pour entraîner les systèmes de reconnaissance de texte arabe, dont les plus célèbres sont (ADAB, KHAT, IFN/ENIT), les versions complètes de ces bases de données ne sont pas disponibles. Nous avons donc créé une base de données composée de (IFN/ENIT version démo [5] avec ADAB [24]) notre base de données contient 569 images de (IFN/ENIT) et 15028 images de (ADAB) nous avons donc 15 597 noms de villes/villages tunisiens manuscrits toute ces images sont des images de deux dimontions. De plus, nous avons ajouté un ensemble de 714 caractères extraits de la base de données fournie par El-Sawy et al [26]. Nous avons créé un fichier (Words.txt) grâce à une fonction que nous avons développé dans laquelle on extrait les noms des images, la taille de chaque image et l'écriture écrits dans chaque image.

Voici la structure de fichier words.txt voir le tableau 3.1 que nous avons créé :

Identifiant de mot pour la ligne 00 dans le fichier words	La taille de l'image (Largeur, Hauteur)	La transcription de mot
263	453x90	الخليج
1232444119357	142x76	قرطاج
a-char98	32x32	ج

TABLE 3.1 – Structure de fichier "words.txt"

Voici une capture de fichier words.txt voir la figure 3.1 :

words.txt - Bloc-notes

Fichier	Edition	Format	Affichage	Aide
263			453 90	الخليج
1232444119357			142 76	قرطاج
a-char98			32 32	ج

FIGURE 3.1 – Structure de notre fichier créé "words.txt"

Voici la structure de notre base de données voir la figure 3.2 :

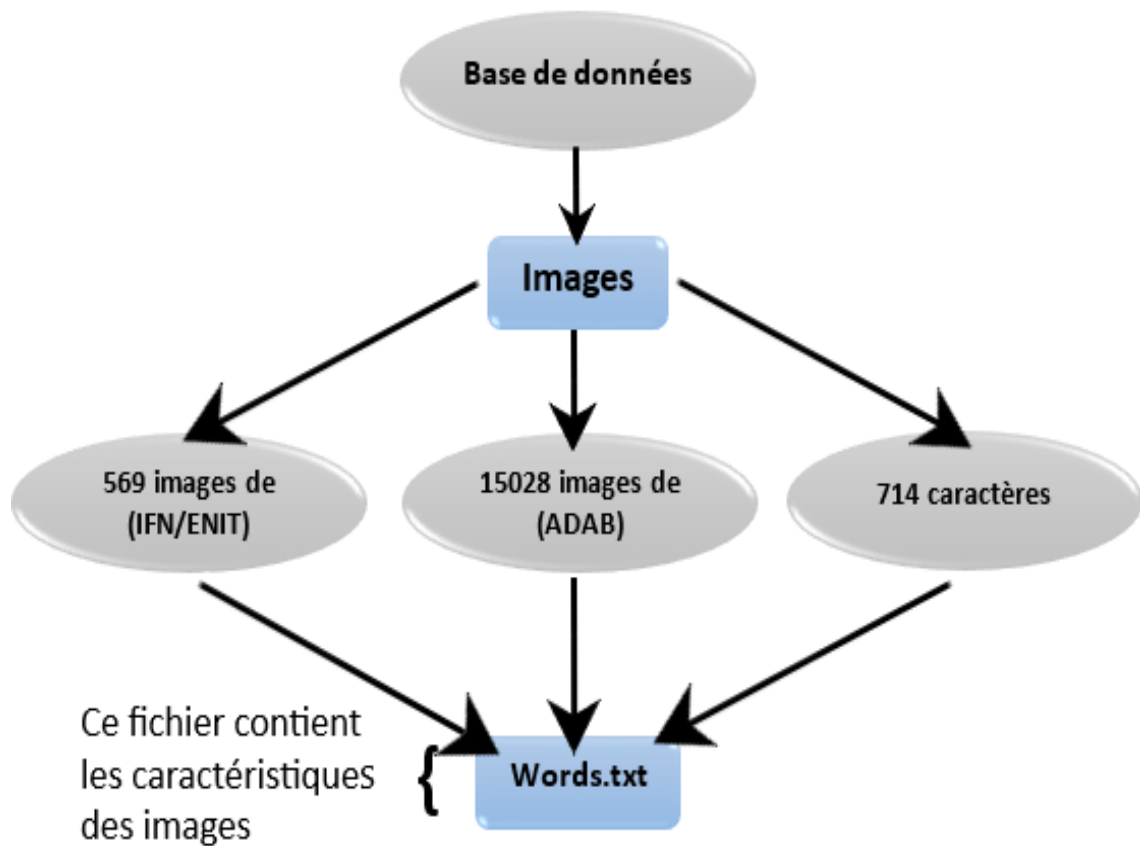


FIGURE 3.2 – Structure de la base de données utilisée

### 3.4 L'approche proposée

Dans notre travail; pour construire un système de reconnaissance de texte arabe a écriture manuscrite, nous avons utilisé un réseau de neurones constitué d'un réseau de neurones convolutifs (CNN) pour la vision par ordinateur en profondeur et d'un réseau de neurones récurrent (RNN) pour le traitement du langage naturel et d'une couche finale de classification temporelle connexionniste (CTC) voir la figure 3.3.

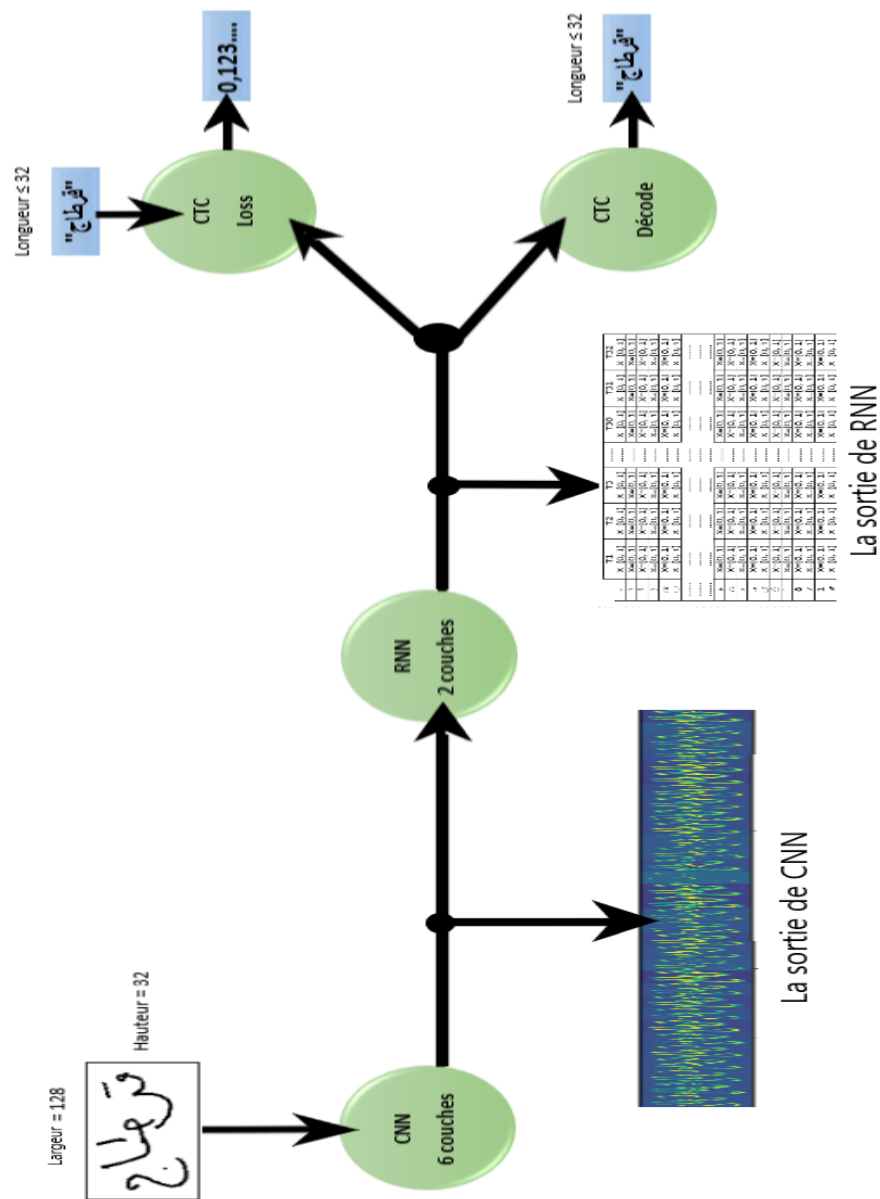


FIGURE 3.3 – L'architecture de notre système

Le tableau suivant 3.2 présente les étapes de notre système.

Type	Description
Entrée	128x32 niveaux de gris image
Convolution+BN	Kernel 5x5
MaxPooling	Pool 2x2
Convolution+BN	Kernel 5x5
MaxPooling	Pool 2x2
Convolution+BN	Kernel 5x5
MaxPooling	Pool 1x2
Convolution+BN	Kernel 3x3
MaxPooling	Pool 1x2
Convolution+BN	Kernel 3x3
MaxPooling	Pool 1x2
Convolution+BN	Kernel 3x3
MaxPooling	Pool 1x1
Bidirectional-LSTM	unités cachées :512
CTC	Décodage ou Perte $\leq 1$

TABLE 3.2 – Structure de notre système

### 3.4.1 L'entrée de système

Les images d'entrée doivent avoir un fond blanc avec une écriture noire, nous remodelons la taille de l'image à 128x32 (LargeurxHauteur) puis nous la saisissons dans le CNN.



FIGURE 3.4 – Exemple d'entrée de notre système

### 3.4.2 La première phase

Une fois que nous avons fini de remodeler les images d'entrée, elles sont introduites dans les couches CNN, tout d'abord, nous entrons les images dans les couches CNN pour extraire les caractéristiques des images. La couche de convolution utilise deux filtres de taille 5x5 dans les trois premières couches et 3x3 dans les trois dernières couches. Ensuite, une fonction RELU est appliquée. Après cela, une couche de POOLING résume les régions d'image et génère une version réduite de l'entrée.

Ceci est un exemple de la sortie CNN voir la figure 3.5 :

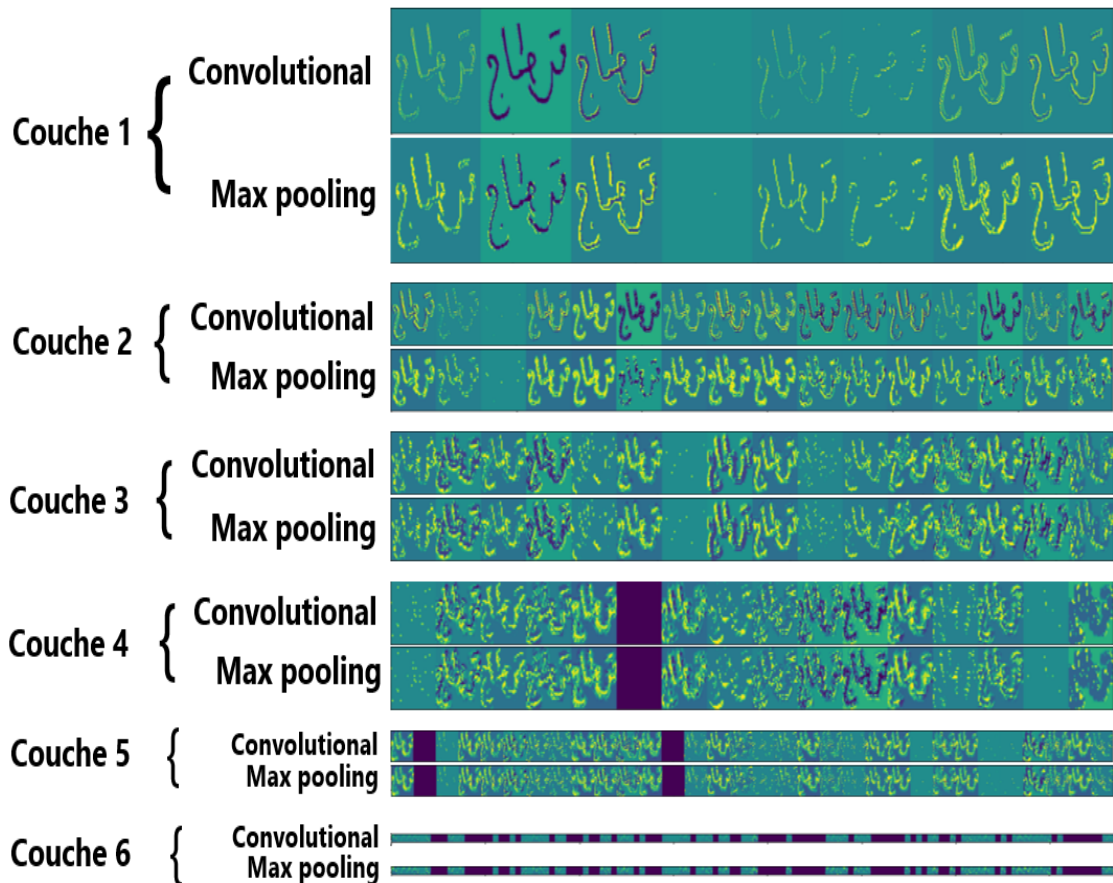


FIGURE 3.5 – La sortie de la première phase (CNN)

Nous obtenons des séquence de caractéristiques de la forme (None,32, 512), 32 correspond au nombre de pas de temps, et 512 n'est rien d'autre que le nombre de caractéristiques à chaque pas de temps voir la figure 3.6.

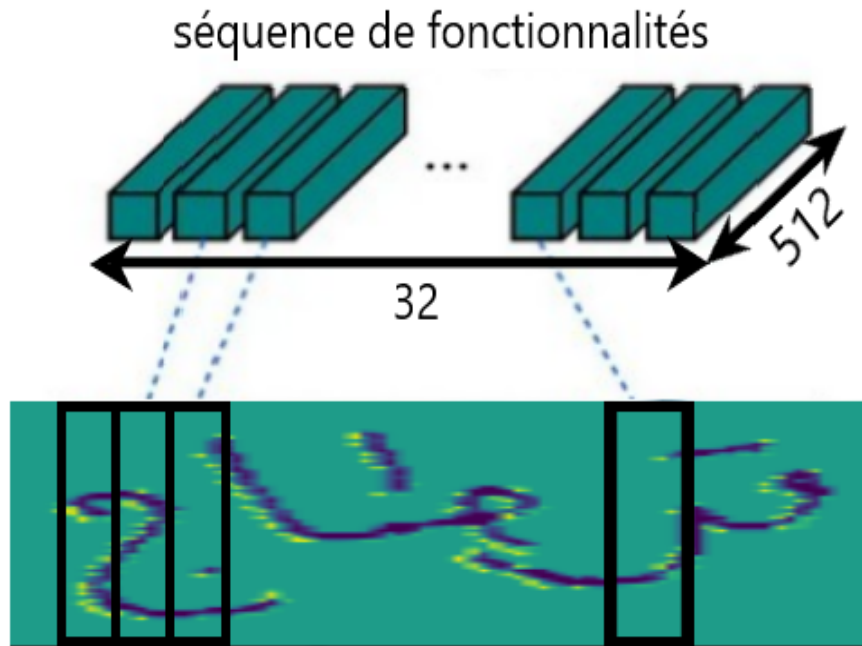


FIGURE 3.6 – La sortie de CNN

Une fois que l'image d'entrée est passée à travers des couches convolutives pour extraire des caractéristiques des images, la sortie est ensuite envoyée aux couches RNN.

### 3.4.3 La deuxième phase

La séquence de caractéristiques contient 512 caractéristiques par pas de temps, et le RNN propage les informations pertinentes à travers cette séquence. La célèbre implémentation de mémoire à long terme (Bi-LSTM) des RNN est utilisée, car elle est capable de diffuser des informations sur de plus longues distances et offre des propriétés d'entraînement plus robustes que RNN voir la figure 3.7. La séquence de sortie du RNN est mappée sur un tableau 32x44. L'ensemble de données que nous avons utilisé se compose de 43 caractères différents, de plus, le caractère supplémentaire ajouté par CTC, il y a donc 44 entrées pour chacune des 32 pas de temps voir le tableau 3.3.

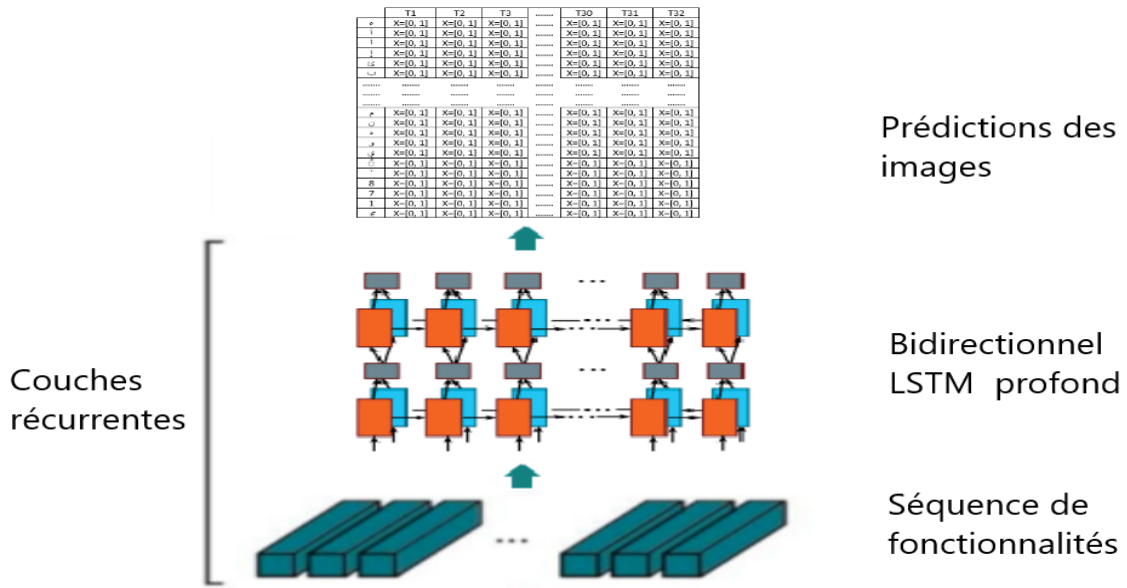


FIGURE 3.7 – Le schéma de RNN (Bi-LSTM)

Le RNN produit une probabilité softmax sur la matrice de vocabulaire voir le tableau 3.3, les colonnes correspondent aux 32 pas de temps, les lignes correspondent aux caractères suivants : " 178' ".

	T1	T2	T3	.....	T30	T31	T32
ء	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
آ	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
إ	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
!	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
ع	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
ب	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....
م	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
و	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
ه	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
و	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
ي	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
ي	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
و	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
8	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
7	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
1	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]
€	X=[0, 1]	X=[0, 1]	X=[0, 1]	.....	X=[0, 1]	X=[0, 1]	X=[0, 1]

TABLE 3.3 – La sortie de RNN matrice softmax



Ces sorties de différents pas de temps voir le tableau 3.3 sont transmises au décodeur CTC pour finalement obtenir le texte brut des images.

### 3.4.4 La troisième phase

Après la transmission de séquence de caractéristique au modèle LSTM, le CTC reçoit une matrice de probabilité softmax sur le vocabulaire et le texte de la transcription de mot et calcule la valeur de perte. Lors de l'inférence, le CTC ne reçoit que la matrice et il la décode dans le texte final. Le texte de la transcription de mot et le texte reconnu peuvent comporter au maximum 32 caractères.

Puisque nous n'avons pas de séquence de caractéristique pour tous les pas de temps c'est-à-dire, 32 nous utilisons CTC pour reconnaître le texte, avec CTC nous n'avons pas à nous soucier de l'alignement de chaque caractère sur l'emplacement de l'image d'entrée.

Dans notre exemple la longueur de texte de la transcription de mot est de 5 caractères voir la figure 3.8, ce qui n'est pas égal à la longueur de la prédiction, c'est-à-dire 32.

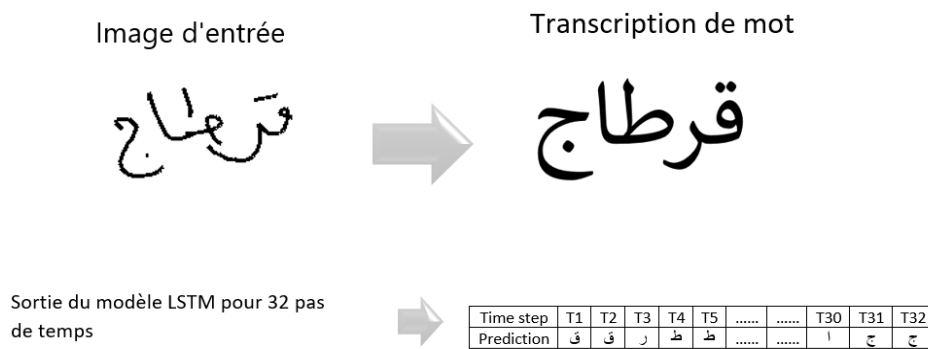


FIGURE 3.8 – Exemple de prédiction

#### 3.4.4.1 Décodage

Maintenant, l'opération de décodage se compose de deux étapes :

##### a. Le calcul de bestpath

La première étape consiste à calculer le meilleur chemin en considérant le caractère avec la probabilité maximale à chaque pas de temps voir les tableaux 3.4 et 3.5.

	T1	T2	T3
ا	0	0	0.9
ب	1	0.6	0
ε	0	0.4	0.1

TABLE 3.4 – La sortie de Bi-LSTM

Pas de temps	T1	T2	T3
Prédiction	ب	ب	ا

TABLE 3.5 – Prédiction de (best path)

**b. La fusion des répétitions et la suppression des caractères vides**

La deuxième étape consiste à supprimer les blancs et les caractères en double voir la figure 3.9.

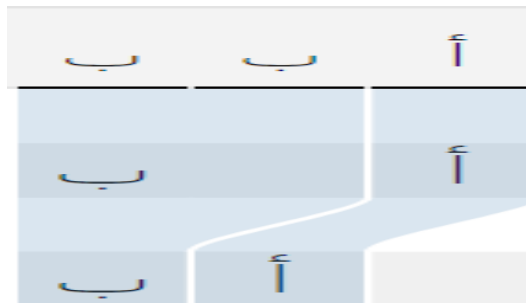


FIGURE 3.9 – La fusion des répétitions

Dans ce cas il y a un autre problème, parfois il y a des caractères qui se répètent successivement voir la figure 3.10.

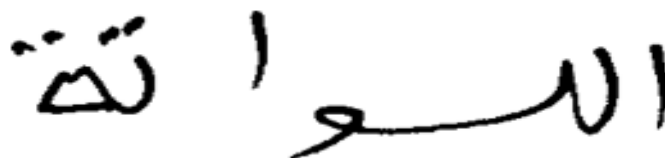


FIGURE 3.10 – Exemple d'entrée d'un mot contenant des lettres répétées

Si nous fusionnons les caractères répétés, nous allons perdre un caractère qui est censé être répété voir la figur 3.11.

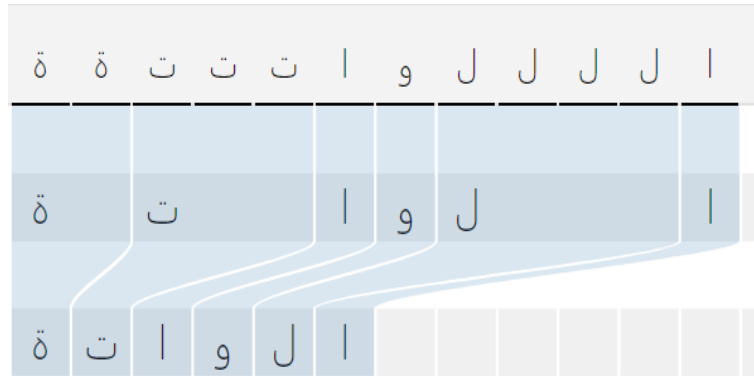


FIGURE 3.11 – La sortie de CTC s’il n’y a pas des blancs

Donc le modèle Bi-LSTM ajoute un caractère vide pour séparer les caractères et nous arrivons à garder tous les caractères voir la figure 3.12.

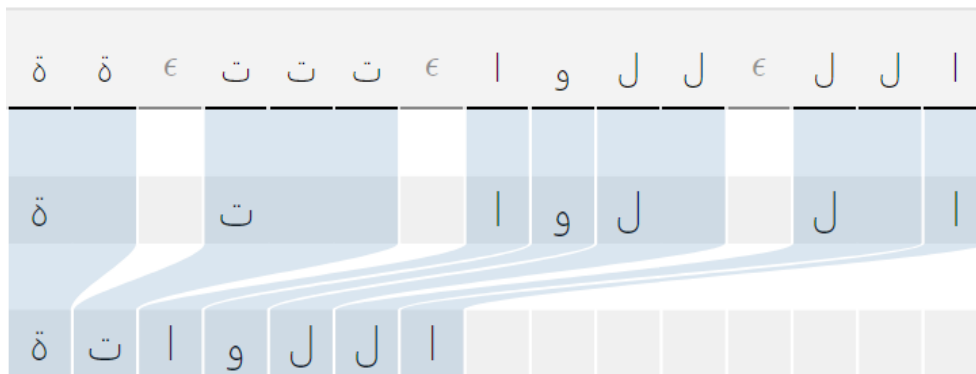


FIGURE 3.12 – La sortie de CTC s’il y a des blancs

### 3.4.4.2 Le calcul de perte

Pour former le CRNN, nous devons calculer la perte en fonction de l’image et de son étiquette.

Le tableau 3.6 montre un exemple de matrice de sortie du Bi-LSTM pour l’image montre dans la figure 3.13. Il y a trois pas de temps et trois caractères (dont un blanc).

A chaque pas de temps, le score du caractère totalise 1. Pour calculer la perte, tous les scores des alignements possibles de texte de la transcription de mot sont additionnés voir le tableau 3.7. De cette manière, l'emplacement du caractère dans l'image n'est pas significatif.



FIGURE 3.13 – Exemple d'entrée après l'ajoute de 3 pas de temps

Pour calculer la probabilité d'obtenir un texte correct, nous multiplions les scores de probabilité des caractères dans chaque ligne. puis nous calculons la somme de tous les probabilités voir l'exemple suivant voir les tableaux 3.6 et 3.7 :

\	T1	T2	T3
ا	0	0	0.9
ب	1	0.6	0
ε	0	0.4	0.1

TABLE 3.6 – La sortie de Bi-LSTM

\	<b>T1</b>	<b>T2</b>	<b>T3</b>
X1	P(ε)=0	P(ب)=0.6	P(ا)=0.9
X2	P(ب)=1	P(ا)=0	P(ε)=0.1
X3	P(ب)=1	P(ε)=0.4	P(ا)=0.9
X4	P(ب)=1	P(ا)=0	P(ا)=0.9
X5	P(ب)=1	P(ب)=0.6	P(ا)=0.9

TABLE 3.7 – Toute les possibilités de mot

Chapitre 3

$$P(X1) = 0 \times 0.6 \times 0.9 = 0$$

$$P(X2) = 1 \times 0 \times 0.1 = 0$$

$$P(X3) = 1 \times 0.4 \times 0.9 = 0.36$$

$$P(X4) = 1 \times 0 \times 0.9 = 0$$

$$P(X5) = 1 \times 0.6 \times 0.9 = 0.54$$

$$P = P(X1) + P(X2) + P(X3) + P(X4) + P(X5)$$

$$P = 0.90.$$

Pour calculer la perte facilement nous appliquons l'équation suivant :  $Loss = -\log(P)$ .

Cette perte peut être rétro-propagée et le réseau peut être formé.

$$Loss = -\log(P) = -\log(0.90) = 0.04575.$$

### 3.4.5 L'entraînement de modèle

Pour entraîner le modèle à partir de rien :

- Il faut accéder au répertoire src et exécutez la commande `python main.py -train` voir la figure 3.14.

Après chaque epoch d'apprentissage, la validation est effectuée sur un ensemble de validation l'ensemble de données est divisé en 95% des échantillons utilisés pour l'apprentissage et 5% pour la validation.

```
Epoch: 1
Train NN
Batch: 1 / 987 Loss: 0.053096976
Batch: 2 / 987 Loss: 0.23889847
Batch: 3 / 987 Loss: 6.6904097
Batch: 4 / 987 Loss: 77.70292
Batch: 5 / 987 Loss: 147.93971
Batch: 6 / 987 Loss: 79.928375
Batch: 7 / 987 Loss: 92.8
Batch: 8 / 987 Loss: 88.896576
Batch: 9 / 987 Loss: 65.581116
Batch: 10 / 987 Loss: 51.85073
```

FIGURE 3.14 – Résultats obtenus lors d'entraînement le modèle

### 3.4.6 La validation de modèle

Pour effectuer uniquement une validation avec un NN formé, nous avons exécuté la commande python **main.py -validate**. La formation sur le CPU prend 26 heures sur notre pc.

La sortie attendue est indiquée ci-dessous voir la figure 3.15.

```
[OK] "عيني ابي ة" -> "عيني ابي ة"  
[OK] "سوسة إبن خلدون" -> "سوسة إبن خلدون"  
[OK] "بوج ابر" -> "بوج ابر"  
[OK] "بودري اس" -> "بودري اس"  
[OK] "الرحم ات" -> "الرحم ات"  
[OK] "تركّي" -> "تركّي"  
[OK] "سبببة" -> "سبببة"  
[OK] "تل ابّت" -> "تل ابّت"  
Character error rate: 0.769112%. Word accuracy: 96.000000%.
```

FIGURE 3.15 – Résultats obtenus lors de validation le modèle

## 3.5 Présentation de l'application

### 3.5.1 Interface « Accueil »

Au démarrage de l'application, la page <Accueil> affiche une zone de téléchargement des images (A) qui accepte les extensions suivantes ('.bmp', '.tif', '.png', '.jpg'), La zone (B) affiche le temps de réponse, calculé à partir du moment où l'utilisateur clique sur le bouton <Recognize> jusqu'à l'obtention du caractère reconnu, l'unité de ce temps en secondes et une zone d'affichage du résultat de la reconnaissance (C), où l'écriture n'est pas autorisée dans cette zone voir la figure 3.16.

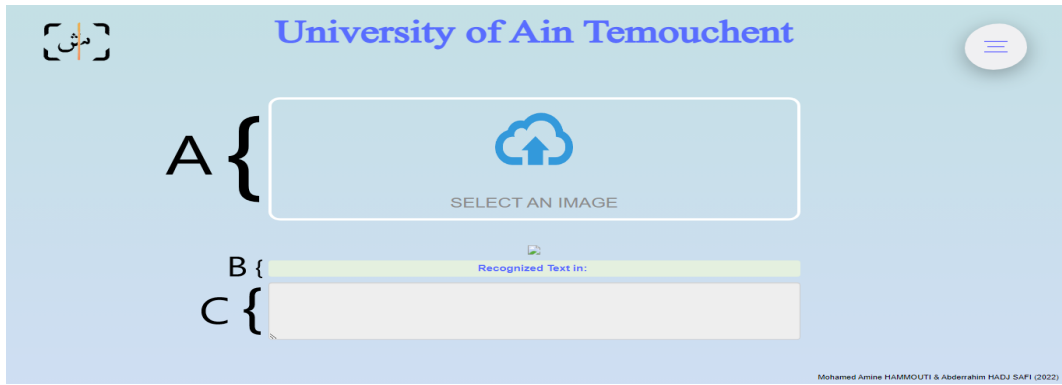


FIGURE 3.16 – Interface d’Accueil

Lorsque l'utilisateur clique sur la zone (A), l'interface illustrée à la figure 3.17 apparaît, l'utilisateur choisit l'image.



FIGURE 3.17 – Interface de choix d’images

Lors de la sélection de l'image, l'interface 3.18 apparaît, où se trouvent deux boutons <Recognize> pour la reconnaissance de l'écriture, <Change> pour changer le choix «permettre au utilisateur de modifier son choix d'image qu'il veut ouvrir».

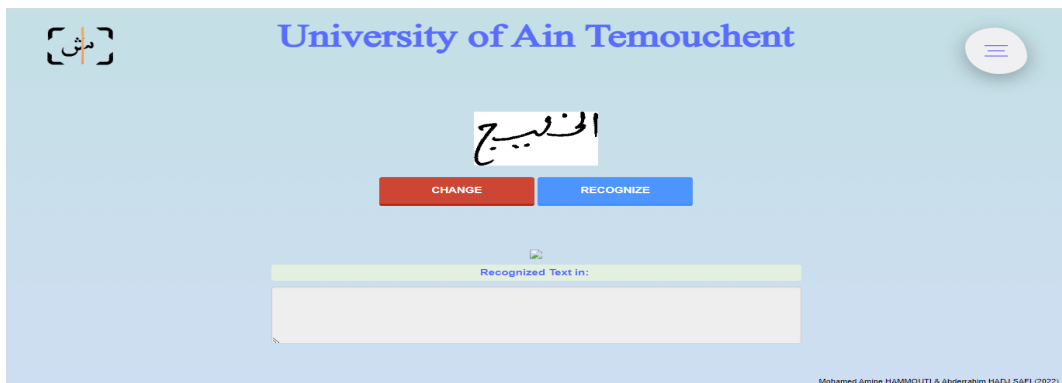


FIGURE 3.18 – Interface principale après la sélection d’image

Lorsque <Recognize> est pressé, l'interface 3.19 apparaît, où la région b montre le résultat de la reconnaissance de l'écriture.

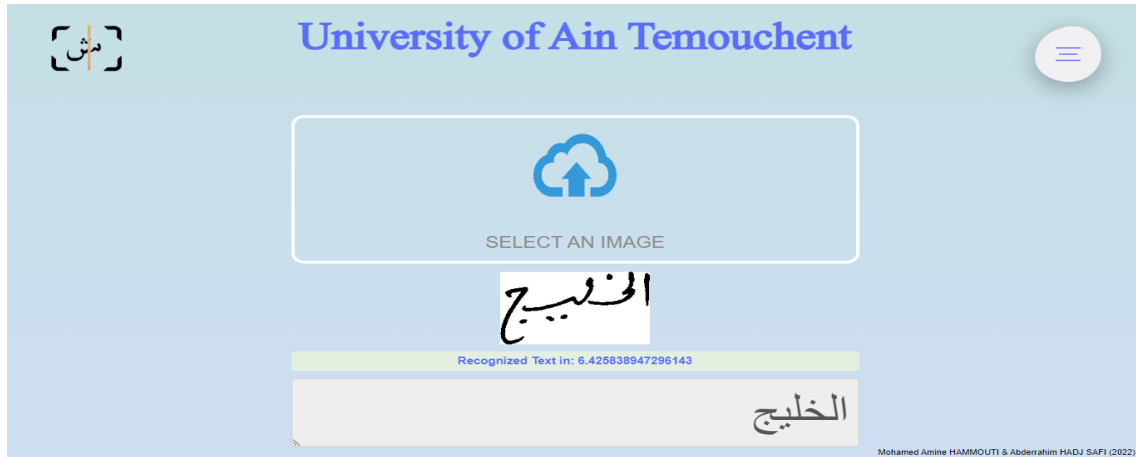


FIGURE 3.19 – Interface principale après l’affichage du résultat

### 3.5.2 Interface « Help »

Cette fenêtre s’affiche lorsque l’utilisateur clique sur le bouton (D) voir la figure 3.20.

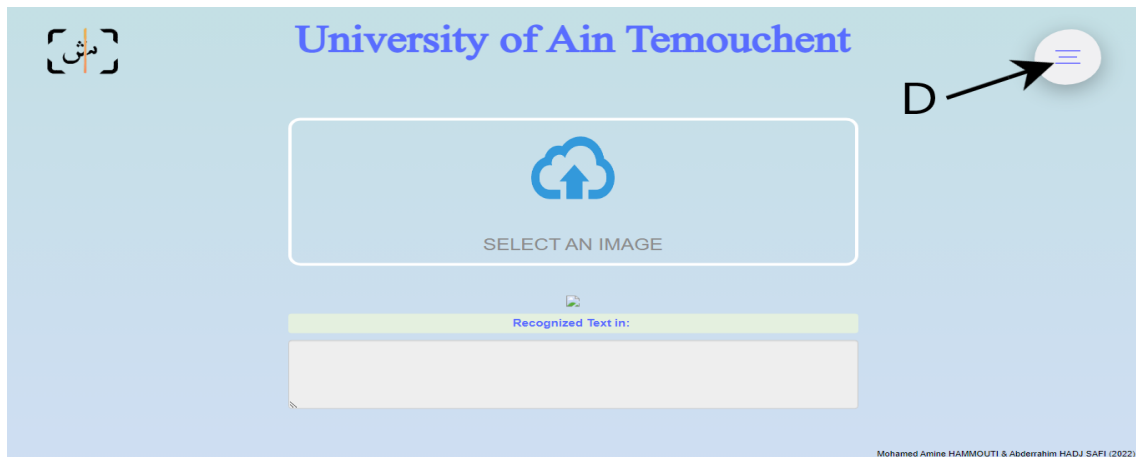


FIGURE 3.20 – Interface d’Accueil

Cette fenêtre contient une explication du fonctionnement de notre système, de l’utilisation de ce site et des résultats obtenus par ce système. Elle contient aussi notre contact voir la figure 3.21.



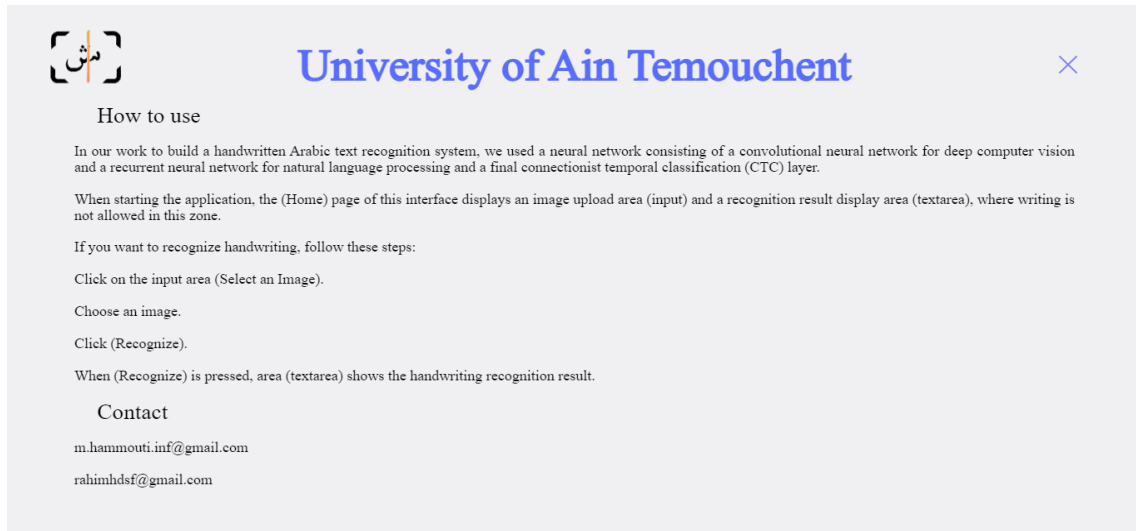


FIGURE 3.21 – Interface Help

### 3.5.3 Exemple

Voici quelques exemples de notre système voir les figures ( 3.22 et 3.23) et un exemple depuis l'invite de commande voir la figure 3.24 :



FIGURE 3.22 – Exemple d'image contenant le mot "الناضور"

La figure 3.22 contenant un mot et la figure contenant un caractère 3.23.



FIGURE 3.23 – Exemple d'image contenant la lettre "و"

Ci-dessous la figure 3.24 qui représente un exemple d'image contenant un mot depuis l'invite de commande.

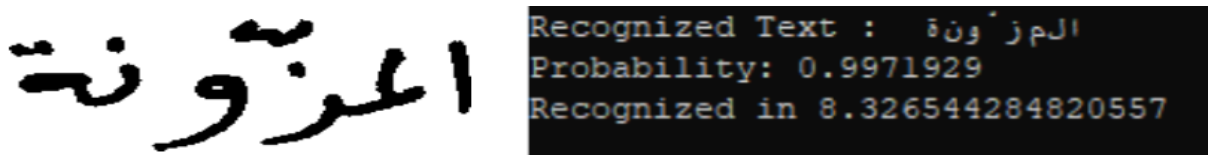


FIGURE 3.24 – Exemple depuis l'invite de commande d'image contenant le mot "المزونة"

### 3.6 Résultats

Après validation, nous avons obtenu un taux de correction de 96% et taux d'erreur de 0.769112% résultat que nous avons obtenu après avoir entraîné ce modèle par (IFN/ENIT+14258 images d'ADAB). Mais lorsque nous avons entraîné ce modèle par (ADAB) nous avons obtenu un taux de correction de 61,4265% et taux d'erreur de 9.813597%.

Le tableau suivant 3.8 montre les résultats obtenus.

Entraînement	Validation	Taux de précision	Taux d'erreur
IFN/ENIT+714 caractères +14258 images d'ADAB	ADAB	96%	0.769112%
ADAB	IFN/ENIT	61,4265%	9.813597%

TABLE 3.8 – Les résultats obtenus

### 3.7 Discussion des résultats

Dans notre projet nous avons travaillé sur la détection de mots arabes manuscrits avec un apprentissage supervisé à l'aide d'une base de données avec un fichier texte qui contient une description de la base de données.

Voici les facteurs affectant la précision de la reconnaissance :

Nombre de données utilisées.

Nombre de couches CNN utilisées.

Modèle de LSTM utilisé.

La précision des mots détectés peut varier et dépend de la clarté de l'écriture et de l'auteur la précision des mots détectés peut être moins précise dans les mots écrits manuscrits que dans les mots imprimés car parfois les mots peuvent être mal écrits ou pas assez clairs, aussi certaines lettres arabes sont de forme similaire, donc parfois elles peuvent être confondues les unes avec les autres.

### **3.8 Conclusion**

Dans ce dernier chapitre de notre mémoire; nous avons expliqué le principe de fonctionnement de notre système et comment nous appliquons CNN, RNN et CTC dans le processus de reconnaissance de l'écriture manuscrite de mot arabe, et nous avons expliqué comment utiliser notre système, autrement dit, les avantages offerts par notre système.

## Conclusion générale

Dans ce mémoire nous avons réalisé un système hors ligne de reconnaissance de l'écriture manuscrite arabe. Pour ce faire, nous avons utilisé les réseaux de neurones récurrents convolutifs (CRNN). Nous avons discuté les notions d'une série de classifieurs couramment utilisés dans les systèmes de reconnaissance d'écritures et nous avons présenté leurs avantages et leurs inconvénients.

Ainsi, après avoir eu un aperçu du comportement des classifieurs, nous avons expliqué la façon d'utilisation du CNN, RNN et CTC dans notre système, comme nous l'avons décrit, aussi que ses avantages. Nous avons implémenté notre système proposée en utilisant le langage de programmation (PYTHON). Concernant la base de données, nous avons combiné une partie de la base de données (IFN/ENIT) [5], la base de données (ADAB) [24], et un ensemble de 714 images qui représentent caractères de la langue arabe; ces images sont extraites de la base de données introduite par El-Sawy et al [26], afin d'améliorer nos résultats.

Après l'implémentation du CRNN on a entraîné et validé notre modèle, nous avons obtenu des résultats très intéressants avoisinant les 96% pour la reconnaissance des mots arabes et un taux d'erreur de 0.77%. Enfin, nous avons élaboré quelques tests sur des images de la base donnée avec un index au hasard nous avons obtenu des résultats fiables et étonnants.

Comme travaux futurs, nous pouvons enrichir notre système avec plus de fonctionnalités telles qu'il peut reconnaître des phrases et des textes entiers (il nous suffira juste de reconnaître le caractère espace), nous allons enrichir notre système avec plus de données afin de reconnaître :

- Les nombres,
- Des textes arabe imprimés,
- Les images en couleurs,
- Les images tachées.

Et nous augmenterons le taux de précision et réduirons le taux d'erreur en essayant d'ajouter quelques techniques dans notre système comme :

- Les algorithmes génétiques,
- Logique floue pour plus d'interprétabilité,
- Ajouter plus de couches CNN,
- Remplacer LSTM par 2D-LSTM.

# Bibliographie

- [1] Advantages and disadvantages of ann. <https://www.asquero.com/article/advantages-and-disadvantages-of-artificial-neural-networks>. [Online accessed : 2022-03-10], [language : English], [journal : Asquero].
- [2] Comprendre la reconnaissance optique de caractères : le guide moov ai. <https://moov.ai/fr/blog/reconnaissance-optique-de-caracteres-ocr/>. [Online accessed : 2022-02-07].
- [3] editdistance pypi. <https://pypi.org/project/editdistance/0.3.1/>. [Online accessed : 2022-04-18].
- [4] Genetic algorithm advantages disadvantages electricalvoice. <https://electricalvoice.com/genetic-algorithm-advantages-disadvantages/>. [Online accessed : 2022-03-12].
- [5] IFN/ENIT - database Arabic OCR handwritten arabic word recognition arabic database. <http://www.ifnenit.com/download.htm>. [Online accessed : 2022-04-29].
- [6] k-means advantages and disadvantages, google developers. <https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages>. [Online accessed : 2022-03-12].
- [7] K-means clustering : Algorithm, applications, evaluation methods, and drawbacks, data science. <https://towardsdatascience.com/>.
- [8] LSTM | Introduction to LSTM | Long Short Term Memor | <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>. [Online accessed : 2022-05-09].

- 
- [9] Numpy documentation numpy v1.22 manual. <https://numpy.org/doc/stable/>. [Online accessed : 2022-04-18].
- [10] Opencv : Introduction. <https://docs.opencv.org/4.5.5/d1/dfb/intro.html>. [Online accessed : 2022-04-18].
- [11] Qu'est-ce qu'un convolutional neural network? <https://www.jeuxetredatascientist.fr/convolutional-neural-network/>. [language : fr-FR], [Online accessed : 2022-03-10], [journal : Devenir Data Scientist], [month : jun], [year : 2021].
- [12] Recurrent neural networks (rnn) advantages. <https://www.educba.com/recurrent-neural-networks-rnn/>. [Online accessed : 2022-03-12].
- [13] Tensorflow. <https://www.tensorflow.org/?hl=fr>. [Online accessed : 2022-04-18], [language : fr], [journal : TensorFlow].
- [14] Welcome to flask documentation (2.1.x). <https://flask.palletsprojects.com/en/2.1.x/>. [Online accessed : 2022-04-18].
- [15] Explanation of Connectionist Temporal Classification, October 2019.
- [16] Sherif Abdelazeem and Hesham M Eraqi. On-line arabic handwritten personal names recognition system based on hmm. In *2011 International Conference on Document Analysis and Recognition*, pages 1304–1308. IEEE, 2011.
- [17] Sadiq H Abdulhussain, Basheera M Mahmmmod, Marwah Abdulrazzaq Naser, Muntadher Qasim Alsabah, Roslilah Ali, and SAR Al-Haddad. A robust handwritten numeral recognition using hybrid orthogonal polynomials and moments. *Sensors*, 21(6) :1999, 2021.
- [18] Badr Al-Badr and Sabri A Mahmoud. Survey and bibliography of arabic optical text recognition. *Signal processing*, 41(1) :49–77, 1995.
- [19] NAJOUA BEN AMARA. Utilisation des modèles de markov cachés planaires en reconnaissance de l'écriture arabe imprimée. In *Ecrit et multimedia (Tours, 9 septembre 1999)*, pages 5–6, 1999.
- [20] Mustapha Amrouch. Reconnaissance de caractères imprimés et manuscrits, textes et documents basée sur les modèles de markov cachés. 2012.



- 
- [21] Sherif Abdel Azeem and Hany Ahmed. Effective technique for the recognition of offline arabic handwritten words using hidden markov models. *International Journal on Document Analysis and Recognition (IJ DAR)*, 16(4) :399–412, 2013.
- [22] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6) :1554–1563, 1966.
- [23] Abdel Belaïd. Reconnaissance automatique de l'écriture et du document. *Campus scientifique, Vandoeuvre-Lès-nancy*, 2001.
- [24] Houcine Boubaker, Abdelkarim Elbaati, Najiba Tagougui, Haikal El Abed, Monji Kherallah, Volker Märgner, and Adel M. Alimi. Adab database, 2021.
- [25] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3) :273–297, 1995.
- [26] Ahmed El-Sawy, Mohamed Loey, and Hazem El-Bakry. Arabic handwritten characters recognition using convolutional neural network. *WSEAS Transactions on Computer Research*, 5(1) :11–19, 2017.
- [27] Mohamed Elleuch, Rania Maalej, and Monji Kherallah. A new design based-svm of the cnn classifier architecture with dropout for offline arabic handwritten recognition. *Procedia Computer Science*, 80 :1712–1723, 2016.
- [28] Hesham M Eraqi and Sherif Abdel Azeem. An on-line arabic handwriting recognition system : Based on a new on-line graphemes segmentation technique. In *2011 international conference on document analysis and recognition*, pages 409–413. IEEE, 2011.
- [29] Maged Mohamed Mahmoud Fahmy and S Al Ali. Automatic recognition of handwritten arabic characters using their geometrical features. *Studies in informatics and Control*, 10(2) :81–98, 2001.
- [30] Khaoula Jayech, Mohamed Ali Mahjoub, and Najoua Essoukri Ben Amara. Synchronous multi-stream hidden markov model for offline arabic handwriting recognition without explicit segmentation. *Neurocomputing*, 214 :958–971, 2016.

- 
- [31] Akram Khémiri, Afef Kacem Echi, and Mourad Elloumi. Bayesian versus convolutional networks for arabic handwriting recognition. *Arabian Journal for Science and Engineering*, 44(11) :9301–9319, 2019.
- [32] Akram Khémiri, Afef Kacem, and Abdel Belaïd. Towards arabic handwritten word recognition via probabilistic graphical models. In *2014 14th International conference on frontiers in handwriting recognition*, pages 678–683. IEEE, 2014.
- [33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [34] Balachandra Kumaraswamy. Neural networks for data classification. In D. Binu and B.R. Rajakumar, editors, *Artificial Intelligence in Data Mining*, pages 109–131. Elsevier, 2021.
- [35] Dror Lederman, Ehud Zmora, Stephanie Hauschildt, Angelika Stellzig-Eisenhauer, and Kathleen Wermke. Classification of cries of infants with cleft-palate using parallel hidden markov models. *Medical & biological engineering & computing*, 46(10) :965–975, 2008.
- [36] G Lorette and Y Lecourtier. Reconnaissance et interprétation de textes manuscrits hors-ligne : un problème d’analyse de scène. *CNED*, 92 :109–135, 1992.
- [37] Liana M Lorigo and Venugopal Govindaraju. Offline arabic handwriting recognition : a survey. *IEEE transactions on pattern analysis and machine intelligence*, 28(5) :712–724, 2006.
- [38] Rania Maalej and Monji Kherallah. Convolutional neural network and blstm for offline arabic handwriting recognition. In *2018 International Arab conference on information technology (ACIT)*, pages 1–6. IEEE, 2018.
- [39] Mouhcine Rabi, Mustapha Amrouch, and Zouhir Mahani. Hybrid hmm/mlp models for recognizing unconstrained cursive arabic handwritten text. In *International Conference on Advanced Information Technology, Services and Systems*, pages 438–448. Springer, 2017.

- [40] Mouhcine Rabi, Mustapha Amrouch, Zouhir Mahani, and Driss Mammass. Recognition of cursive arabic handwritten text using embedded training based on hmms. In *2016 International Conference on Engineering & MIS (ICEMIS)*, pages 1–5. IEEE, 2016.
- [41] Sumit Saha. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*, December 2018.
- [42] S Sakunthala, R Kiranmayi, and P Nagaraju Mandadi. A review on artificial intelligence techniques in electrical drives : Neural networks, fuzzy logic, and genetic algorithm. In *2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon)*, pages 11–16. IEEE, 2017.
- [43] A Lawrence Spitz. Determination of the script and language content of document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3) :235–245, 1997.
- [44] Zahia Tamen, Habiba Drias, and Dalila Boughaci. An efficient multiple classifier system for arabic handwritten words recognition. *Pattern Recognition Letters*, 93 :123–132, 2017.
- [45] Charles C Tappert, Ching Y Suen, and T Wakahara. *On-line handwriting recognition : A survey*. IBM Thomas J. Watson Research Division, 1988.
- [46] Abdelwahab Zramdini and Rolf Ingold. Optical font recognition using typographical features. *IEEE Transactions on pattern analysis and machine intelligence*, 20(8) :877–882, 1998.

**Résumé:** Le domaine de la reconnaissance automatique de l'écriture (imprimée ou manuscrite) est important en raison de sa polyvalence.

Par conséquent, l'objectif de notre projet de fin d'études est de mettre en place un système d'identification de l'écriture arabe à partir d'images numérisées (système hors ligne). Où, nous nous basons sur le réseau neuronal récurrent convolutif (CRNN), qui combine le réseau neuronal convolutif (CNN) et le réseau neuronal récurrent (RNN). Tandis que, CNN prend en charge la vision par ordinateur approfondie et RNN traite le langage naturel. De plus, nous utilisons la classification temporelle connexionniste (CTC) pour l'encodage. Par ailleurs, nous avons utilisé les bases de données les plus connues (IFN/ENIT), (ADAB) et nous avons ajouté 714 images de caractères pour améliorer nos résultats.

Après avoir appliqué le CRNN, nous avons formé et validé notre modèle, et nous avons obtenu des résultats très intéressants autour de 96% sur la reconnaissance des mots arabes avec un taux d'erreur de 0,77%.

**Mots clés:** Intelligence Artificielle (IA), Traitement automatique de la langue naturelle (TALN), Reconnaissance automatique de l'écriture arabe manuscrite, Réseaux de Neurones Artificiels (RNA), Apprentissage en profondeur.

**Abstract:** The field of automatic handwriting recognition (printed or handwritten) is important because of its versatility.

Therefore, the objective of our graduation project is to set up a system for identifying Arabic script from digitized images (offline system). Where, we rely on Convolutional Recurrent Neural Network (CRNN), which combines Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Whereas, CNN supports deep computer vision and RNN processes natural language. Additionally, we use connectionist temporal classification (CTC) for encoding. Moreover, we used the most known databases (IFN/ENIT), (ADAB) and we added 714 images of characters to improve our results.

After applying the CRNN, we trained and validated our model, and we obtained very interesting results around 96% on the recognition of Arabic words with an error rate of 0.77%.

**Keywords:** Artificial Intelligence (AI), Natural language processing (NLP), Automatic recognition of handwritten, Artificial Neural Networks (ANN), Deep Learning.

**ملخص:** يعد مجال التعرف التلقائي على خط اليد (المطبوع أو المكتوب بخط اليد) مهمًا نظرًا لتعدد استخداماته.

لذلك ، فإن الهدف من مشروع تخرجنا هو إنشاء نظام لتعرف على النص العربي من الصور الرقمية (نظام غير متصل بالشبكة). حيث نعتمد على الشبكة العصبية التلافيفية المتكررة ، التي تجمع بين الشبكة العصبية التلافيفية والشبكة العصبية المتكررة . وحيث تدعم الرؤية العميقة للكمبيوتر وعمليات المعالجة للغة الطبيعية. بالإضافة إلى ذلك ، نستخدم التصنيف الزمني الوصلي للتشفير. علاوة على ذلك، استخدمنا قواعد البيانات الأكثر شهرة (افن / وُنيت) ، (أداب) وأضفنا 714 صورة للأحرف لتحسين نتائجنا.

بعد تطبيق الشبكة العصبية التلافيفية المتكررة، قمنا بتدريب نموذجنا والتحقق من صحته ، وحصلنا على نتائج مثيرة جدًا حول 96٪ في التعرف على الكلمات العربية بمعدل خطأ 0.77٪.

**الكلمات المفتاحية:** الذكاء الاصطناعي، المعالجة الآلية للغة الطبيعية، التعرف التلقائي على النص العربي المكتوب بخط اليد، الشبكة العصبية الاصطناعية، التعلم العميق.