



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la

Recherche Scientifique

Université 20 Août 1955- Skikda

Faculté des Sciences

Département d'Informatique

Mémoire

En vue de l'obtention du diplôme de

Master en Informatique

Spécialité : Systèmes Informatiques

Thème

Un Modèle d'Apprentissage Profond pour la Reconnaissance des Caractères Manuscrits

Présenté Par :

Selma SAADI

Encadré Par :

Dr. Samira HAZMOUNE

Soutenue publiquement le 03 Juillet 2022 devant le jury composé de :

Président : Dr Rami Soumia

Examineur : Dr Bendana Rokia

Année universitaire 2021/2022

DEDICACES

Je dédie ce mémoire à mes chers parents, ce travail leur doit beaucoup, qu'il soit pour eux le témoignage de mon infinie reconnaissance pour leurs accompagnements tout au long de mon parcours scolaire.

REMERCIEMENTS

Tout d'abord, je tiens à remercier le Dieu le miséricordieux de m'accorder sa bénédiction, sans laquelle ce mémoire n'aurait pas avancé ni vu le jour.

Mes sincères remerciements sont exprimés à Dr. HAZMOUNE Samira, pour son encadrement de qualité, ses remarques ainsi que ses suggestions visant un travail de haute qualité et rien de moins m'ont profondément impressionné.

Je tiens également à exprimer mes remerciements aux membres de jury, qui m'ont fait l'honneur d'examiner mon travail.

Je remercie profondément mes chers parents, à qui je dois ce que je suis et ce que je serais, qui n'ont jamais cessé de m'encourager et me conseiller, qui m'ont beaucoup aidé tout en long de mon parcours scolaire, grâce à leur amour, leur compréhension et leur patience sans jamais me quitter des yeux ni baisser les bras. C'est leur soutien inconditionnel qui m'a relevé quand je me suis lassé. Aucun remerciement ne peut exprimer ma reconnaissance que vous soyez mes parents.

Mes remerciements vont également à mes chères sœurs. A Insaf, ma sœur et ma meilleure amie, qui m'a aidée à traverser cette période angoissante de la manière la plus positive. Et Assia, qui a été généreuse avec son amour et son encouragement malgré la longue distance qui nous sépare.

Je ne peux pas clore cette liste sans citer mon petit neveu Iyad, qui sait toujours comment procurer la joie et le bonheur pour toute la famille.

RESUME

Ce mémoire s'inscrit dans le cadre de la reconnaissance automatique de l'écriture manuscrite, qui a suscité beaucoup d'intérêt de la part des chercheurs au cours des dernières décennies, en raison de son omniprésence dans les domaines où les humains interagissent, communiquent et effectuent des transactions.

Le Deep Learning (Apprentissage Profond) et plus particulièrement, le réseau de neurones convolutif CNN (Convolutional Neural Network) a montré beaucoup de succès dans le domaine de reconnaissance de l'écriture manuscrite. Le succès est largement lié à son architecture profonde qui comporte plusieurs couches cachées permettant l'extraction automatique des caractéristiques. La modélisation des couches CNN ainsi que les hyper paramètres ont un effet profond sur la précision du modèle de reconnaissance.

Dans ce mémoire, nous présentons un modèle de Deep Learning basé sur les CNN pour la reconnaissance des caractères manuscrits. Afin d'améliorer la précision du modèle, une série d'expérimentations sur l'ensemble de données EMNIST a été réalisée pour déterminer les hyper paramètres optimaux de l'architecture CNN. Le réseau profond CNN a été entraîné et évalué sur les sous-ensembles de données EMNIST Letters, EMNIST Balanced et EMNIST MNIST. La précision de la classification pour les trois ensembles de données était de 94,28 %, 88,93 %, et 99,32 % respectivement. Les résultats comparatifs avec certains algorithmes d'apprentissage et quelques travaux précédents sur la même base de données sont très encourageants.

Mots clés : Reconnaissance de l'écriture manuscrite, CNN, Deep Learning, EMNIST.

ملخص

تندرج هذه المذكرة في إطار التعرف التلقائي على خط اليد الذي جذب اهتمامًا كبيرًا من الباحثين على مدار العقود الماضية ، نظرًا لانتشاره في كل مكان في المجالات التي يتفاعل فيها البشر ويتواصلون وينفذون المعاملات.

أظهر التعلم العميق وبشكل أكثر تحديدًا الشبكة العصبية التلافيفية (CNN) نجاحًا كبيرًا في مجال التعرف على خط اليد. يرتبط النجاح إلى حد كبير بهندسته العميقة التي تحتوي على عدة طبقات مخفية تسمح باستخراج الميزة تلقائيًا. نمذجة طبقة CNN وكذلك المعلمات الفائقة لها تأثير عميق على دقة نموذج التعرف.

في هذه المذكرة ، نقدم نموذج التعلم العميق المستند إلى CNN للتعرف على الأحرف المكتوبة بخط اليد. من أجل تحسين دقة النموذج ، تم إجراء سلسلة من التجارب على مجموعة بيانات EMNIST لتحديد المعلمات الفائقة المثلى لهندسة CNN. تم تدريب شبكة CNN العميقة وتقييمها على مجموعات بيانات EMNIST و EMNIST Balanced و EMNIST MNIST. كانت دقة التصنيف لمجموعات البيانات الثلاث 94.28% و 88.93% و 99.32% على التوالي. النتائج المقارنة مع بعض خوارزميات التعلم وبعض الأعمال السابقة على نفس قاعدة البيانات مشجعة للغاية.

الكلمات المفتاحية: التعرف على خط اليد ، الشبكات العصبية التلافيفية (CNN) ، التعلم العميق ، EMNIST.

ABSTRACT

This thesis is part of automatic handwriting recognition, which has attracted a lot of interest from researchers over the past decades, due to its ubiquity in areas where humans interact, communicate and perform transactions.

Deep Learning and more specifically the Convolutional Neural Network (CNN) has shown great success in the field of handwriting recognition. The success is largely related to its deep architecture which has several hidden layers allowing automatic feature extraction. CNN layer modeling as well as hyper parameters have a profound effect on the accuracy of the recognition model.

In this thesis, we present a CNN-based Deep Learning model for handwritten character recognition. In order to improve the accuracy of the model, a series of experiments on the EMNIST dataset were performed to determine the optimal hyper parameters of the CNN architecture. The CNN deep network was trained and evaluated on the EMNIST Letters, EMNIST Balanced and EMNIST MNIST datasets. The classification accuracy for the three datasets was 94.28%, 88.93% and 99.32%, respectively. The comparative results with some learning algorithms and some previous work on the same database are very encouraging.

Keywords: handwriting recognition, convolutional neural networks (CNN), Deep Learning, EMNIST.

Table des matières

Introduction Générale	1
I. Contexte.....	2
II. Problématique et objectifs	2
III. Plan de mémoire	3
Chapitre 01 : Etat de l'art sur la reconnaissance de l'écriture manuscrite	6
1.1. Introduction	6
1.2. Définition de la reconnaissance de l'écriture manuscrite	6
1.3. Classifications des systèmes de reconnaissance de l'écriture	7
1.3.1. Le mode d'acquisition.....	7
1.3.2. Le type d'écriture	8
1.3.3. L'utilisateur du système	9
1.3.4. Le mode de reconnaissance.....	9
1.4. La reconnaissance de l'écriture manuscrite hors ligne	10
1.4.1. Architecture générale d'un système de reconnaissance de l'écriture manuscrite hors ligne	10
1.4.2. Les approches de classification	16
1.5. Domaines d'applications	18
1.6. Quelques travaux récents	19
1.7. Conclusion	22
Chapitre 02 : L'apprentissage automatique et l'apprentissage profond	26
2.1. Introduction	26
2.2. Apprentissage automatique et apprentissage profond	26
2.2.1. Apprentissage automatique (Machine Learning)	26
2.2.2. Apprentissage profond (Deep Learning).....	30
2.2.3. Différence entre l'apprentissage automatique et l'apprentissage profond	30
2.3. Réseaux de neurones profonds	32
2.3.1. Concept de réseaux de neurones	32
2.3.2. Type des réseaux de neurones	33
2.4. Réseaux de neurones convolutifs	37
2.4.1. Architecture globale	37

2.4.2. Évolution architecturale des réseaux de neurones convolutifs.....	39
2.5. Quelques systèmes existants de reconnaissance de caractères manuscrits basé CNN	43
2.6. Conclusion	45
Chapitre 03 : Un système de reconnaissance des caractères manuscrits : Conception, Expérimentation et Résultats	48
3.1. Introduction	48
3.2. Motivation et objectif	48
3.3. Architecture et description du système de reconnaissance.....	49
3.3.1. Présentation détaillée du système.....	49
3.4. Expérimentations et résultats	53
3.4.1. Bases de données utilisées.....	53
3.4.2. Détermination de l'architecture du modèle CNN	57
3.4.3. Evaluation des performances	62
3.4.4. Comparaison avec les algorithmes d'apprentissage automatique	68
3.4.5. Comparaison avec des travaux de la littérature.....	71
3.5. Conclusion	76
Conclusion et perspectives.....	78
I. Bilan	78
II. Perspectives	78
Bibliographie.....	80
Webographie	86
Annexe : Outils d'implémentation et application	88

Liste des figures

Chapitre 01 : Etat de l'art sur la reconnaissance de l'écriture manuscrite

Figure 1.1: Types de communication Homme-Machine par l'écriture (Bendjeddou, 2008).....	6
Figure 1.2: Représentation de l'information en ligne et hors ligne [Web 02]	6
Figure 1.3: Écriture en ligne et hors ligne	8
Figure 1.4: Les deux types d'écritures (Pluchot, 2021).....	8
Figure 1.5: Structure générale d'un système de reconnaissance d'écriture manuscrite.....	11
Figure 1.6: Image de départ en niveau de gris et son Image binarisée [Web 01]	12
Figure 1.7 : Exemple de correction de pente sur une image, la ligne de base est indiquée en orange (Mioulet, 2015).....	12
Figure 1.8: Exemple de correction de l'inclinaison sur une image (Mioulet, 2015).....	13
Figure 1.9: Exemple de Squelettisation sur une image (Mioulet, 2015).....	13
Figure 1.10: Exemple de segmentation sur une image [Web 03]	14
Figure 1.11: Les 3 architectures de combinaison des classifieurs (Es Saady, 2012)	18

Chapitre 02 : L'apprentissage automatique et l'apprentissage profond

Figure 2.1: Domaines en Intelligence Artificielle (Darwish et al., 2020).....	29
Figure 2.2: Classification des approches d'apprentissage automatique (Chauhan et Singh, 2018).....	29
Figure 2.3: Exemple d'un HMM gauche-droite (Mioulet, 2015)	29
Figure 2.4: Machine à vecteurs de support (Kumar, 2021).....	29
Figure 2.5: K plus proches voisins [Web 04].....	29
Figure 2.6: Machine Learning vs Deep Learning (Wolfewicz, 2021)	31
Figure 2.7: Fonctionnement de l'apprentissage automatique et l'apprentissage profond (Alom et al., 2019).....	31
Figure 2.8: Du neurone biologique au neurone artificiel (Hossam, 2020).....	34
Figure 2.9: Perceptron multi-couches (Mioulet, 2015)	34
Figure 2.10: Neurone artificiel (Sueiras, 2021).....	35
Figure 2.11: Exemples de fonctions d'activation d'un perceptron Multi-Couches (à gauche :, à droite.....) [Web 05]	35
Figure 2.12: Architecture convolutive standard (Chauhan et Singh, 2018).....	35
Figure 2.13: Neurone récurrent (Mioulet, 2015).....	36
Figure 2.14: Cellule Long Short-Term Memory (LSTM) (Sueiras, 2021)	37
Figure 2.15: Un exemple de l'opération de convolution (Rezgui, 2019).....	38
Figure 2.16: Un exemple de l'opération de Max pooling (Chauhan et Singh, 2018)	39
Figure 2.17: Couche Fully-Connected [Web 06]	39

Chapitre 03 : Un système de reconnaissance des caractères manuscrits : Conception, Expérimentation et Résultats

Figure 3.1: Architecture globale de notre système.....	49
Figure 3.2: Architecture CNN à deux couches de convolution (Alom et al., 2019).....	51
Figure 3.3: L'architecture du CNN utilisé dans notre modèle (Modèle Lettre).....	51
Figure 3.4: Échantillons de toutes les lettres et chiffres de l'ensemble de données EMNIST (Cohen et al., 2017).....	53
Figure 3.5: Processus de conversion utilisé pour convertir l'ensemble de données NIST (Cohen et al., 2017).....	54
Figure 3.6: La distribution des échantillons dans chaque classe pour l'ensemble de données EMNIST Lettres (Cohen et al., 2017).	55
Figure 3.7: La distribution des échantillons dans chaque classe pour l'ensemble de données Balanced (Cohen et al., 2017).	56
Figure 3.8: La distribution des échantillons dans chaque classe pour l'ensemble de données EMNIST MNIST (Cohen et al., 2017).....	56
Figure 3.9: Matrice de confusion pour la base EMNIST Letters	63
Figure 3.10: Matrice de confusion pour la base EMNIST MNIST	64
Figure 3.11: Matrice de confusion pour la base EMNIST Balanced	66
Figure 3.12: Similarité de la lettre G et la lettre Q.....	67
Figure 3.13: Similarité de la lettre I majuscule et la lettre L minuscule	67
Figure 3.14: Précision des modèles CNNs conçu	68
Figure 3.15: Graphique à barres illustrant la comparaison de précision sur la base EMNIST Letters	69
Figure 3.16: Graphique à barres illustrant la comparaison de précision sur la base EMNIST Balanced.....	69
Figure 3.17: Graphique à barres illustrant la comparaison de précision sur la base EMNIST MNIST	70
Figure 3.18: Graphique à barres illustrant la comparaison de précision sur la base EMNIST Letters.....	74
Figure 3.19: Graphique à barres illustrant la comparaison de précision sur la base EMNIST Balanced.....	74
Figure 3.20: Graphique à barres illustrant la comparaison de précision sur la base EMNIST MNIST	75

Liste des tableaux

Chapitre 01 : Etat de l'art sur la reconnaissance de l'écriture manuscrite

Tableau.1.1 : Résumé des résultats de quelques systèmes de reconnaissance de l'écriture manuscrite	19
--	----

Chapitre 02 : L'apprentissage automatique et l'apprentissage profond

Tableau.2.1 : Comparaison des performances des architectures convolutives (Khan et al., 2020), (OpenGenus, 2019)	43
Tableau.2.2 : Résumé des résultats de quelques systèmes de reconnaissance de l'écriture manuscrite basé CNN.....	45

Chapitre 03 : Un système de reconnaissance des caractères manuscrits : Conception, Expérimentation et Résultats

Tableau 3. 1: Résumé des sous-ensembles de données EMNIST (Cohen et al., 2017).....	55
Tableau 3. 2: Résultats expérimentaux pour déterminer le nombre de paires de couches Convolution – MaxPooling	58
Tableau 3. 3: Résultats expérimentaux pour déterminer la taille des cartes de caractéristiques	59
Tableau 3.4: Résultats expérimentaux pour différentes tailles de la couche dense	60
Tableau 3.5: Résultats expérimentaux pour déterminer le taux de Dropout	61
Tableau 3.6: Résultats de reconnaissance sur la base EMNIST Lettres.	63
Tableau 3.7: Résultats de reconnaissance sur la base EMNIST MNIST... ..	64
Tableau 3.8: Résultats de reconnaissance sur la base EMNIST Balanced.....	65
Tableau 3.9: Comparaison des performances avec l'état de l'art.....	72

Liste des abréviations

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
EMNIST	Extended Modified NIST
HMM	Hidden Markov Models
IA	Intelligence Artificielle
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
K-NN	K Nearest Neighbors
K-PPV	K Plus Proche Voisins
LSTM	Long short-term memory
MLP	Multi-layer Perceptron
MNIST	Modified National Institute of Standards and Technology
NIN	Network in Network
NIST	National Institute of Standards and Technology.
NN	Neural Network
PAPI	Paper and Pencil Interviewing
PMC	Perceptron Multi-Couches
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Networks
SVM	Machines à vecteurs de support
VGG	Visual Geometry Group

INTRODUCTION GENERALE

Sommaire

I.	Contexte.....	2
II.	Problématique et objectifs	3
III.	Plan de mémoire	3

I. Contexte

La reconnaissance des caractères manuscrits est devenue de plus en plus importante dans le monde numérisé d'aujourd'hui en raison de leurs applications pratiques dans diverses activités quotidiennes. Cela peut être prouvé par le fait que ces dernières années, différents systèmes de reconnaissance ont été développés ou proposés pour être utilisés dans différents domaines où une précision élevée est nécessaire.

Les systèmes utilisés pour reconnaître les lettres, les caractères et les chiffres de l'écriture manuscrite aident les gens à résoudre des tâches plus complexes qui, autrement, prendraient du temps et seraient coûteuses. Un bon exemple est l'utilisation de systèmes de traitement automatique dans les banques pour traiter les chèques bancaires. Sans systèmes automatisés de traitement des chèques bancaires, la banque serait obligée d'employer de nombreux employés qui pourraient ne pas être aussi efficaces que le système de traitement informatisé. Les systèmes de reconnaissance de l'écriture manuscrite peuvent s'inspirer des réseaux de neurones biologiques, qui permettent aux humains d'apprendre et de modéliser des relations non linéaires et complexes. Cela signifie qu'ils peuvent être développés à partir des réseaux de neurones artificiels, notamment, le Deep Learning. Le cerveau humain permet aux individus de reconnaître différents objets d'écriture manuscrite tels que des chiffres, des lettres et des caractères. Pour cela, Il est nécessaire de comprendre comment la souscription humaine lit, afin de développer des systèmes capables d'interpréter les lettres et les chiffres de l'écriture manuscrite.

Le système visuel humain est principalement impliqué chaque fois que les individus lisent des caractères manuscrits, des lettres, des mots ou des chiffres. Cela semble sans effort chaque fois que l'on lit l'écriture manuscrite, mais ce n'est pas aussi facile que les gens le croient. Un humain peut donner un sens à ce qu'il voit en se basant sur ce que son cerveau a appris, bien que tout soit fait inconsciemment. Un être humain peut ne pas apprécier à quel point il est difficile de résoudre l'écriture manuscrite. Le défi de la reconnaissance visuelle des formes n'est apparent que pour développer un système informatique pour lire l'écriture manuscrite. L'approche des réseaux de neurones artificiels est considérée comme le meilleur moyen de développer des systèmes de reconnaissance de l'écriture manuscrite.

Les réseaux de neurones aident à simuler le fonctionnement du cerveau humain lors de la lecture de l'écriture manuscrite sous une forme plus simplifiée. Il permet aux machines d'égaliser et même de dépasser les capacités humaines de lecture de l'écriture manuscrite. Les humains ont des styles d'écriture différents, dont certains sont difficiles à lire. En outre, la lecture de l'écriture manuscrite peut être longue et fastidieuse, en particulier lorsque les individus doivent lire plusieurs documents écrits par différents scribes.

II. Problématique et objectifs

La reconnaissance des chiffres manuscrits était le premier problème de recherche de la communauté de l'analyse et de la reconnaissance de documents depuis plus de trois décennies. De nombreux algorithmes de classification fondée sur l'apprentissage profond ont

été étudiés. À cette fin, un jeu de données a été largement utilisé : le jeu de données MNIST. Qui est une base de données de chiffres manuscrits étiquetés. Comme, il existe une quantité importante de travaux testés sur MNIST signalant un taux d'erreur de test inférieur à 1 %, MNIST est considéré comme déjà résolu. En conséquence, une version étendue de MNIST, à savoir l'ensemble de données EMNIST, a été publiée en avril 2017. EMNIST comprend à la fois des chiffres manuscrits et des lettres manuscrites, et il est nettement plus grand que MNIST et les données proviennent d'une source différente. EMNIST est une base de données récente qui représente jusqu'à présent une tâche de classification plus difficile pour les chercheurs.

L'objectif principal de cette recherche est de concevoir un modèle basé sur les réseaux de neurones convolutifs (CNN), qui vise à améliorer les performances des systèmes de reconnaissance automatique des caractères manuscrits pour la base de données EMNIST.

Cet objectif général peut être détaillé dans les sous-objectifs spécifiques suivants :

- Concevoir un modèle CNN pour reconnaître les lettres, les chiffres et les caractères manuscrits à partir des entrées.
- Une étude expérimentale pour déterminer les hyper paramètres optimaux, ainsi que l'effet important des hyper paramètres sur la précision du modèle.
- Une étude comparative afin de démontrer l'utilité de la technologie des réseaux de neurones convolutifs (CNN) dans le développement de systèmes efficaces de reconnaissance de caractères manuscrits.

III. Plan de mémoire

Outre cette introduction et la conclusion générale, ce mémoire est divisé en trois chapitres principaux :

Chapitre 01 : Il présente l'état de l'art sur la reconnaissance automatique de l'écriture. En effet, ce chapitre décrit le principe, l'architecture d'un système de reconnaissance de l'écriture et les différentes techniques utilisées dans ce domaine.

Chapitre 02 : Il donne un aperçu général des méthodes de l'apprentissage automatique ainsi que celles de l'apprentissage profond. Nous y présentons en particulier, les réseaux de neurones convolutifs (CNN), la technique utilisée dans cette étude. Nous y dressons, enfin l'état de l'art sur la reconnaissance automatique de l'écriture par les CNNs.

Chapitre 03 : Il regroupe la description de notre système, ainsi que les différentes expérimentations réalisées et l'analyse des résultats obtenus. Nous présentons dans un premier temps l'architecture de notre système de reconnaissance d'écriture manuscrite. Les performances du système proposé seront évaluées sur trois sous-ensembles de données : EMNIST Letters, EMNIST Balanced et EMNIST MNIST. Enfin, une comparaison est effectuée du modèle conçu avec certains travaux de la littérature afin d'évaluer notre système.

ETAT DE L'ART SUR LA RECONNAISSANCE DE L'ECRITURE MANUSCRITE

Sommaire

1.1.	Introduction	6
1.2.	Définition de la reconnaissance de l'écriture manuscrite	6
1.3.	Classifications des systèmes de reconnaissance de l'écriture	7
1.3.1.	Le mode d'acquisition	7
1.3.2.	Le type d'écriture	8
1.3.3.	L'utilisateur du système	9
1.3.4.	Le mode de reconnaissance	9
1.4.	La reconnaissance de l'écriture manuscrite hors ligne	10
1.4.1.	Architecture générale d'un système de reconnaissance de l'écriture manuscrite hors ligne	10
1.4.2.	Les approches de classification	16
1.5.	Domaines d'applications	18
1.6.	Quelques travaux récents	19
1.7.	Conclusion	22

1.1. Introduction

En raison de son omniprésence dans les domaines où les humains interagissent, communiquent et effectuent des transactions, la reconnaissance de l'écriture manuscrite a rassemblé une grande partie de l'intérêt des chercheurs dans ces dernières décennies. Elle permet le tri automatique du courrier postal, la lecture des chèques bancaires, le traitement automatique des dossiers administratifs, etc. Elle offre aujourd'hui un surcroît d'intérêt avec le développement des nouvelles méthodes permettant de communiquer directement avec la machine de manière plus naturelle.

La reconnaissance de l'écriture manuscrite peut être définie comme une capacité et une technique qui permet le passage d'une image ou d'un signal, contenant une information textuelle à un texte en format électronique (Mioulet, 2015). Ce processus est considéré comme un domaine difficile de reconnaissance de formes, car l'écriture possède une infinité de représentations dues aux variations dans l'écriture manuscrite de différents individus (Altwaijry et Al-Turaiki, 2021). De plus, l'écriture manuscrite d'un même scripteur peut être légèrement différente à chaque fois. Il est donc très difficile de reconnaître les bons caractères et chiffres manuscrits.

Dans ce chapitre, nous décrivons tout d'abord, le domaine de la reconnaissance de l'écriture manuscrite et les approches utilisées en général. Puis, nous nous intéresserons plus particulièrement à la reconnaissance de l'écriture manuscrite hors ligne.

1.2. Définition de la reconnaissance de l'écriture manuscrite

L'écriture manuscrite est l'un des modes classiques de communication entre les êtres humains, qui présente l'avantage d'être familier à la plupart des gens. La reconnaissance de l'écriture est l'utilisation de la technologie pour distinguer les caractères de texte imprimés ou manuscrits à l'intérieur des images ou signaux numériques de documents physiques, tels qu'un document papier numérisé.

La reconnaissance de l'écriture manuscrite est la transformation d'un texte manuscrit, sur tout type de support, vers une représentation numérique compréhensible par la machine et permet à cette dernière de reconnaître le texte comme les humains. La figure 1.1 montre les différents types de communication Homme-Machine en se basant sur la reconnaissance de l'écriture manuscrite.

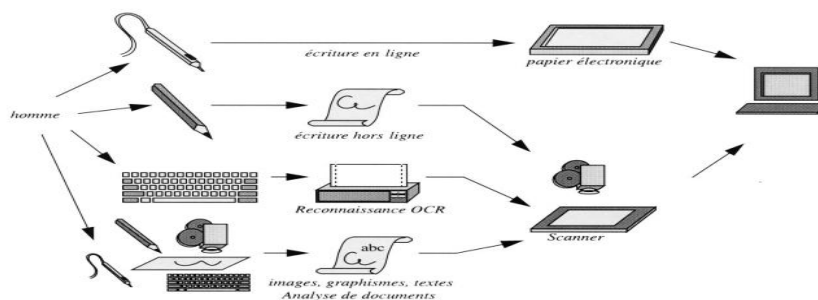


Figure 1.1: Types de communication Homme-Machine par l'écriture (Bendjeddou, 2008)

1.3. Classifications des systèmes de reconnaissance de l'écriture

L'appellation reconnaissance de l'écriture se réfère à plusieurs types de systèmes dont la mission est de décoder l'information écrite. On distingue plusieurs types de système de reconnaissance de l'écriture. Cette diversité des systèmes dépend du type de données traitées et de besoins de l'application visée. Il existe plusieurs critères de classification des systèmes de reconnaissance de l'écriture, parmi lesquels on peut citer : Le mode d'acquisition, le type de l'écriture, l'utilisateur du système et le mode de reconnaissance.

1.3.1. Le mode d'acquisition

Dans la reconnaissance automatique de l'écriture manuscrite, l'entrée est numérisée à partir d'images, de documents et d'appareils en temps réel tels que des tablettes, des tabloïds, des numériseurs, etc., qui sont ensuite interprétés en texte numérique (Priya et al., 2016), de ce fait, les modes d'acquisition de caractères sont de deux types : Saisie en ligne et saisie hors ligne (Figure 1.2). Ainsi, les systèmes de reconnaissance de l'écriture manuscrite peuvent être classés en deux grandes classes : Les systèmes de reconnaissance en ligne, qui utilisent la trace d'un stylo pour la classification et la reconnaissance des informations d'entrée. Et les systèmes de reconnaissance hors ligne, qui utilisent les images comme entrée pour les étapes de reconnaissance.

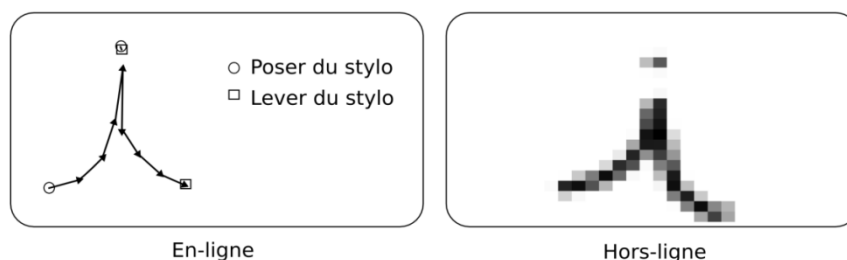


Figure 1.2 : Représentation de l'information en ligne et hors ligne [Web 02]

a. La reconnaissance de l'écriture manuscrite en ligne

La reconnaissance de l'écriture manuscrite en ligne (dite dynamique ou temps-réel) signifie que la machine reconnaît l'écriture pendant que l'utilisateur écrit. Il s'agit d'une représentation numérisée des mouvements du stylo décrivant les informations sur la position, la vitesse, l'accélération et les angles du stylo de manière séquentielle dans les deux coordonnées (Priya et al., 2016). Ce type de reconnaissance nécessite un transducteur qui capture l'écriture au fur et à mesure qu'elle est écrite. Le plus courant de ces appareils est la tablette électronique ou le numériseur.

b. La reconnaissance de l'écriture manuscrite hors ligne

La reconnaissance de l'écriture manuscrite hors ligne (dite statique) est effectuée une fois l'écriture terminée. Ce type de reconnaissance nécessite un scanner optique qui implique la conversion automatique de l'image de l'écriture en un motif binaire (Tappert et al., 1990).

Il est généralement observé qu'elle est plus difficile que la reconnaissance de l'écriture manuscrite en ligne (Azeem et Ahmed, 2013), pour les raisons suivantes (Aries, 2009) :

- A cause de l'absence de l'ordre d'écriture et la dynamique des traits, la détection des traits d'écriture ainsi que sa reconnaissance ne sont pas toujours triviales. Cette difficulté est encore aggravée par, l'état du document, le bruit qui l'entache et les conditions de numérisation (résolution limitée, inclinaison, transaction, élargissement, etc.).
- La variabilité et la non-homogénéité de l'épaisseur des traits d'écriture constituent une charge supplémentaire à prendre en compte dans le processus de reconnaissance.

La figure 1.3 montre un signal d'entrée typique qui peut être analysé dans les deux cas.

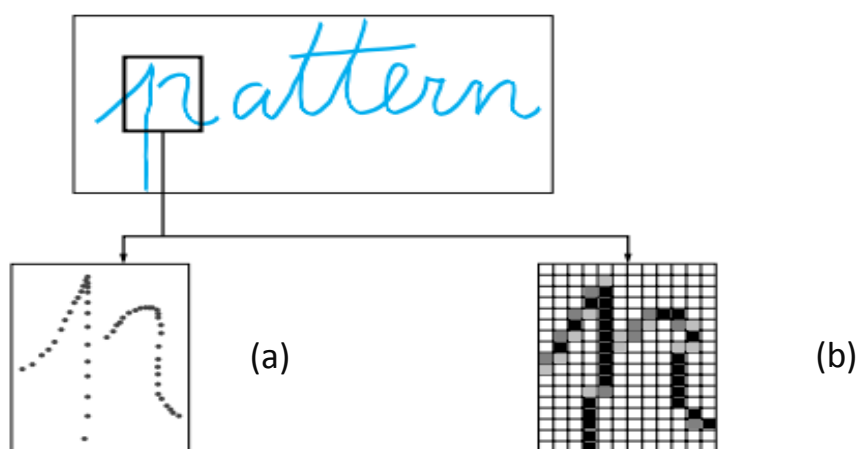


Figure 1.3 : Écriture en ligne et hors ligne (Crettez et Lorette, 1998)

- (a) Informations en ligne séquences de coordonnées (X,Y) de points, ordonnés dans le temps,
- (b) Informations hors ligne : image (matrice de pixels)

1.3.2. Le type d'écriture

La reconnaissance de l'écriture signifie le décodage de n'importe quelle information écrite. D'une manière générale, il existe deux types d'écriture, selon le support de saisie, l'écriture manuscrite (Figure 1.4 à gauche) et l'écriture imprimée (Figure 1.4 à droite).



Figure 1.4: Les deux types d'écritures (Pluchot, 2021)

a. L'écriture manuscrite

L'écriture manuscrite, est le processus d'écriture fait à la main par une personne à l'aide d'un matériel d'écriture tel qu'un crayon ou un stylo. L'écriture manuscrite comprend à la fois le style cursif ainsi que le style imprimée. Ce type d'écriture est très complexe et varié car l'écriture de chacun est unique et différente.

b. L'écriture imprimée

L'écriture imprimée (dite aussi, impression), est le processus de production d'une ou plusieurs copies d'un texte, avec de l'encre sur papier à l'aide d'une presse à imprimer. Ce type d'écriture, est réservé seulement au style d'impression c'est-à-dire les lettres sont écrites séparément plutôt que jointes. L'histoire de cette technologie remonte à l'invention de procédés d'impression par Johannes Gutenberg au 15ème siècle. De nos jours, l'impression peut être vue dans les médias imprimés tels que les journaux et les livres ainsi que sous des formes électroniques telles que les documents Word ou les présentations.

1.3.3. L'utilisateur du système

Les systèmes de reconnaissance de l'écriture manuscrite, peuvent être classés selon le nombre de scripteurs, en trois catégories : Les systèmes mono, multi et omni-scripteurs.

a. Système mono-scripteur

Le système mono-scripteur (propre au scripteur), est un système qui permet d'identifier et reconnaître qu'une seule écriture d'un seul scripteur.

b. Système multi-scripteur

Le système multi-scripteur (propre à l'écriture manuscrite), est un système qui permet d'identifier et reconnaître l'écriture pour un groupe restreint de scripteurs.

c. Système omni-scripteurs

Le système omni-scripteurs (propres à n'importe quel scripteur), est un système qui est censé reconnaître l'écriture de n'importe qui.

1.3.4. Le mode de reconnaissance

Le choix entre les différentes approches est fortement dépendant de l'application où s'exerce la reconnaissance de l'écriture. La première approche consiste en une reconnaissance globale du mot. La deuxième approche correspond à une décomposition du mot en unités élémentaires telles que les lettres ou les graphèmes et la reconnaissance du mot via une identification des unités qui le composent.

a. Approche globale

L'approche globale est dédiée principalement à la reconnaissance de mots isolés, elle est davantage adaptée aux lexiques statiques réduits. Son principe consiste à considérer le mot comme une entité global unique et indivisible. Ce type d'approche est plus largement utilisé dans le domaine hors ligne que dans le domaine en ligne, notamment pour la robustesse de l'approche face aux bruits et aux déformations de l'écriture et les applications spécifiques telles que le traitement des chèques (Poisson, 2005).

b. Approche analytique

Contrairement aux approches globales ou on ne peut pas reconnaître de mot non modélisé, l'approche analytique donne une plus grande souplesse et généralité au système de reconnaissance de l'écriture manuscrite. Elle s'appuie sur une description plus fine souvent liée à la structure alphabétique du vocabulaire : Modèles lettres, modèles mots ou de phrases. Ainsi un modèle mot sera construit par concaténation des modèles lettres qui le composent. Une lettre pourra elle-même être décomposée en sous entités, graphèmes ou début, corps et fin de lettre. Une étape de segmentation est donc nécessaire afin de localiser les lettres du mot. Cette tâche est particulièrement délicate du fait de l'absence de segmentation idéale : Les limites entre caractères sont parfois difficiles à déterminer même pour un être humain (Aries, 2009).

1.4. La reconnaissance de l'écriture manuscrite hors ligne

1.4.1. Architecture générale d'un système de reconnaissance de l'écriture manuscrite hors ligne

La reconnaissance de l'écriture manuscrite hors ligne est une application classique de reconnaissance des formes. Elle s'articule autour des principes habituels : Acquisition d'informations par des organes sensoriels (capteurs), prétraitement, extraction des caractéristiques et classification, suivis éventuellement d'une phase de post-traitement.

La figure 1.5 Montre la structure générale d'un système de reconnaissance de l'écriture manuscrite hors ligne.

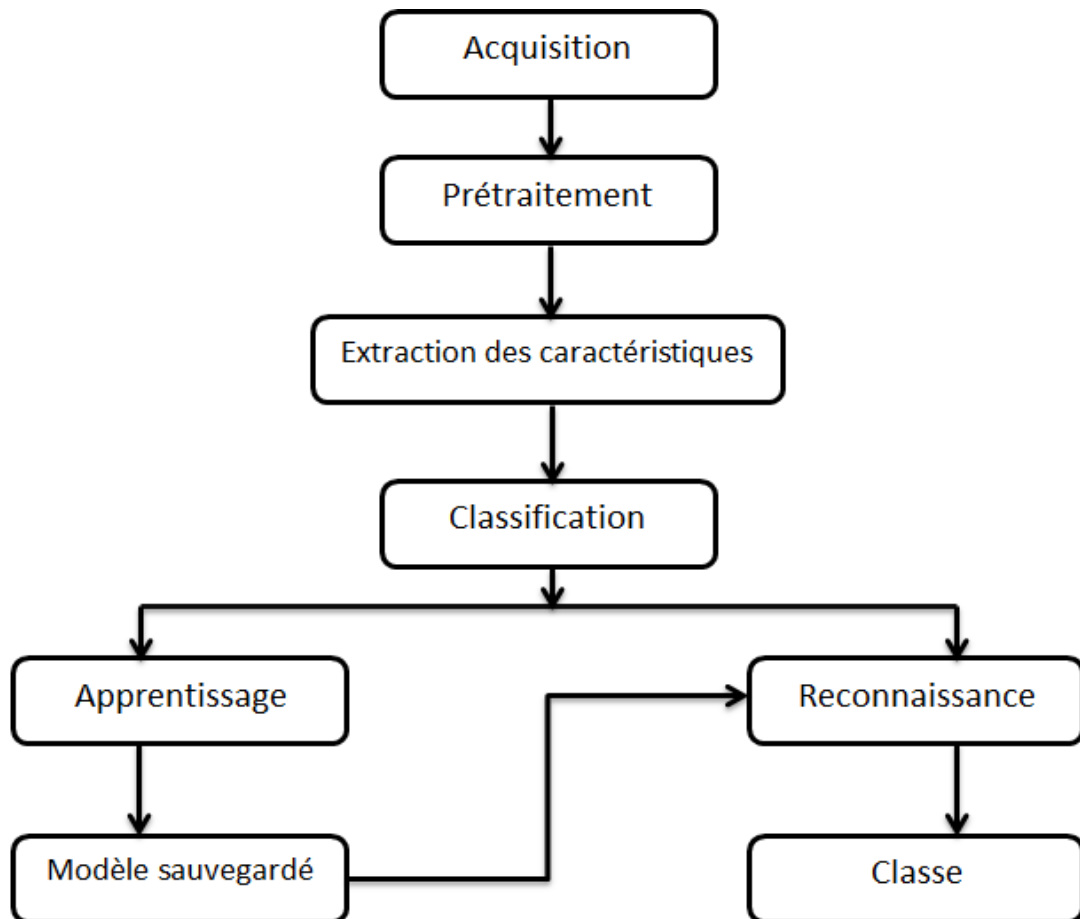


Figure 1.5 : Structure générale d'un système de reconnaissance d'écriture manuscrite

1.4.1.1. Acquisition de l'image

Dans la reconnaissance hors ligne, l'écriture est généralement capturée optiquement par le scanner ou tout autre périphérique d'entrée numérique qui est ensuite disponible en tant qu'image d'entrée de formats tels que JPEG, PNG, etc.

A ce stade, bien que la qualité du système d'acquisition soit bonne, des bruits parasites peuvent apparaître et provoquer une hétérogénéité du fond ceci est dû à la nature de la texture, l'aire de travail et de son éclairage (Djerouni, 2021).

1.4.1.2. Prétraitement

Comme mentionnée dans la section précédente, Il est possible que du bruit soit généré lors de la phase d'acquisition de l'image. Pour cela, il est habituel de prétraiter les images de texte manuscrit avant de les utiliser dans des modèles et des algorithmes de reconnaissance, ce qui permet d'éliminer le bruit et de normaliser au maximum les images à analyser. Il n'est pas nécessaire d'effectuer cette étape, mais selon les applications recherchées, il a été montré que l'utilisation de prétraitement améliore la reconnaissance. Plusieurs opérations de prétraitements peuvent être effectuées après l'étape d'acquisition, les plus généralement

utilisée sont : La binarisation et la squelettisation, la correction de pente, la correction de l'inclinaison et la normalisation.

Dans ce qui suit nous décrivons chacune de ces opérations dans le cadre de la reconnaissance de l'écriture manuscrite.

a. Binarisation

La binarisation a pour but de transformer une image en niveau de gris en une image binaire. Cette étape est effectuée pour diminuer la quantité d'informations présentes dans l'image, et de ne garder que les informations pertinentes. La binarisation peut s'effectuer de manière simple par un seuillage statique, les valeurs supérieures à un seuil T donné prennent une valeur binaire, les valeurs inférieures prennent l'autre valeur, autrement dit, les pixels de l'image sont partagés par un seuil T en deux classes. En général, ils sont représentés par une classe de pixels blancs et une autre classe de pixels noirs. Figure 1.6 illustre le passage d'une image de niveaux de gris à une image binaire.

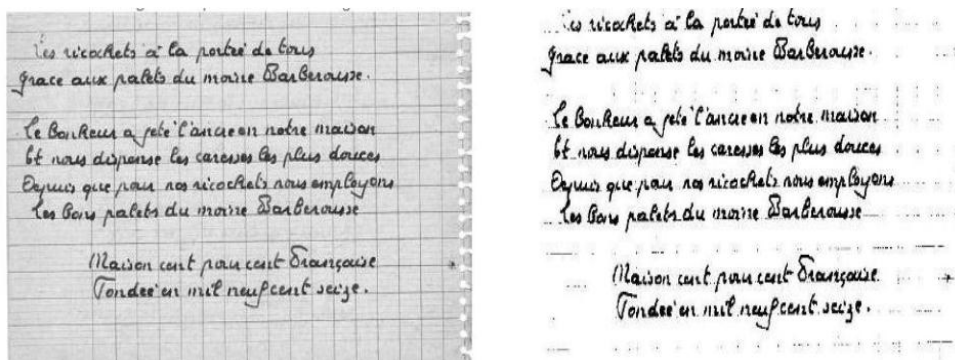
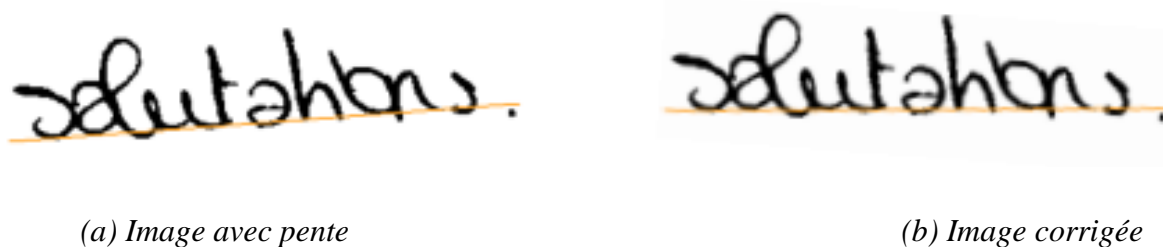


Figure 1.6: Image de départ en niveau de gris et son Image binarisée [Web 01]

b. Correction de pente

La pente est l'inclinaison des lignes de texte par rapport à une ligne de base complètement horizontale. Elle apparaît généralement lorsque le texte est écrit sur une page vierge sans motif réimprimé de lignes ou de cases (Sueiras, 2021). L'objectif de la correction de la pente est de modifier une image pour positionner la ligne de base à l'horizontale.



(a) Image avec pente

(b) Image corrigée

Figure 1.7 : Exemple de correction de pente sur une image, la ligne de base est indiquée en orange (Mioulet, 2015)

c. Correction de l'inclinaison

L'inclinaison est une caractéristique commune de la calligraphie de type cursif présente dans plusieurs textes manuscrits, en particulier dans les traits ascendants et descendants (Sueiras, 2021). Elle constitue l'une des principales sources de variabilité du texte manuscrit, ce qui rend difficile la reconnaissance de l'écriture à l'aide de systèmes automatiques. La correction de l'inclinaison redresse les lettres (Figure 1.8), cette correction permet donc de réduire les différences intra-caractères, en fonction du style de l'écriture (Mioulet, 2015).



(a) Image originale



(b) Image redressée

Figure 1.8 : Exemple de correction de l'inclinaison sur une image (Mioulet, 2015)

d. Normalisation de la ligne de base

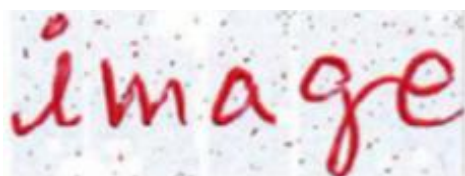
L'un des problèmes rencontrés lors de la reconnaissance de l'écriture manuscrite est l'inclinaison du tracé de l'écriture. Cette inclinaison peut provenir du processus de l'acquisition. L'objectif de la normalisation de la position est de placer la ligne de base au centre de l'image de manière à réduire la variabilité intra-caractères.

e. Filtrage et lissage

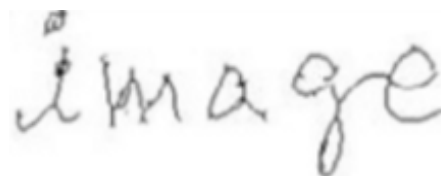
Il s'agit de l'application des différents filtres sur l'image binaire afin d'éliminer le bruit résultant de fonds texturés / colorés, de lisser les contours des formes à reconnaître, de reconnecter certains traits déconnectés, etc.

f. Squelettisation

La squelettisation est une opération qui permet de passer d'une image aux traits épais en une image à ligne plus facile à traiter appelée squelette, comme le montre la Figure 1.9.



(a) Image originale



(b) Squelettisation

Figure 1.9 : Exemple de Squelettisation sur une image (Mioulet, 2015)

g. Segmentation

La segmentation est une étape cruciale dans les systèmes de reconnaissance de l'écriture, et son objectif, comme son nom indique est de découper un mot ou pseudo-mot en une suite de caractères ou graphèmes (Figure 1.10), afin de réduire la complexité des modules de traitements utilisés par la suite (Aries, 2009).



(a) Image originale



(b) Image segmentée

Figure 1.10 : Exemple de segmentation sur une image [Web 03]

1.4.1.3. Extraction des caractéristiques

L'extraction des caractéristiques, autrement dit, l'analyse est une étape très importante dans le système de reconnaissance de l'écriture manuscrite (Aries, 2009). La sélection des caractéristiques présentées au système de reconnaissance est un problème complexe, parce que les résultats finaux ainsi que l'efficacité du système dépendent essentiellement de la relation entre ces derniers, du nombre d'échantillons et de la méthode de classification utilisée. Donc, l'extraction des caractéristiques, peut être défini comme étant l'extraction de l'information la plus pertinente pour un problème de classification donnée, c'est-à-dire celle qui minimise la variabilité intra-classes et qui maximise la variabilité inter-classes.

L'objectif de cette étape est d'analyser le segment d'écriture (mot, caractères, etc.) Afin d'extraire des caractéristiques (primitives) permettant de l'identifier préférentiellement sans ambiguïté. Les paramètres extraits doivent être codifiés dans une représentation appropriée sur laquelle va s'appuyer la méthode de classification (Aries, 2009). Les types de caractéristiques peuvent être classés en deux catégories principales de primitives selon la méthode de définition, c'est-à-dire, celles définies par le concepteur et celles définies par la machine (les caractéristiques automatiques).

a. Caractéristiques définies par le concepteur

Partant d'une image en niveau de gris, le plus bas niveau de représentation est la valeur du pixel. Aucune caractéristique n'est extraite, le vecteur image est fourni tel qu'il est au système (Mioulet, 2015). Des caractéristiques simples qui proviennent de l'information des pixels sont utilisées pour la reconnaissance de l'écriture manuscrite : Pourcentage de pixels noirs dans une zone (Zonage), histogramme de densités horizontal ou vertical (Histogramme), moments invariants, etc.

b. Caractéristiques automatiques

Une façon différente d'obtenir des caractéristiques est de laisser le système les créer pour les besoins de la tâche. De nombreux algorithmes produisent en interne une

représentation des données qui leur sont fournies. Cette représentation interne des données peut tenir lieu de phases de génération de caractéristiques. (Mioulet, 2015). C'est le cas par exemple des réseaux de neurones profonds qui nous intéressent dans le cadre de ce travail.

1.4.1.4. Classification

La phase de classification est la partie décisionnelle du système de reconnaissance. Il s'agit d'utiliser des classifieurs performants optimisés pour la reconnaissance de l'écriture manuscrite (Mioulet, 2015). Lorsque l'image d'entrée est présentée au système de reconnaissance de l'écriture manuscrite, ses caractéristiques sont extraites et transmises en tant qu'entrée au classifieur formé. Les classifieurs comparent l'entité en entrée avec le modèle stocké et trouvent la meilleure classe correspondante pour l'entrée (Dhaka, 2015).

Le processus de classification est réalisé en deux phases : La première phase correspond à la conception du classifieur (modèle de reconnaissance). Tandis que la deuxième phase consiste à utiliser le modèle générée pour classifier un nouvel exemple.

a. Phase d'apprentissage

Le mécanisme d'apprentissage a pour objectif d'apprendre au système les propriétés pertinentes du vocabulaire utilisé et de les organiser en modèles de références. Il s'agit, lors de cette étape, d'apprendre au système autant d'échantillons que de formes d'écritures différents appelées base d'apprentissage, un ensemble de connaissances nécessaire à la modélisation du problème est extrait à partir de la base d'apprentissage. Cet ensemble permet de reconnaître ultérieurement des formes de classes inconnues. Les procédés d'apprentissage sont différents selon le type du système de reconnaissance. D'une manière générale, il existe deux types d'apprentissage : Supervisé et non supervisé.

- **Apprentissage supervisé**

L'apprentissage est dit supervisé si les différentes classes des exemples sont connues a priori et la tâche d'apprentissage est guidée par un superviseur ou un professeur, c'est-à-dire, le concepteur. Le processus d'apprentissage se déroule de la manière suivante : Pour chaque exemple en entrée, le professeur indique dans quelle classe doit se trouver ce dernier. Le nombre d'exemples varie de quelques unités à quelques dizaines, voire même quelques centaines par caractère. Ce type d'apprentissage est effectué, en général, dans le contexte de la classification et de la régression.

- **Apprentissage non supervisé**

À l'inverse de l'apprentissage supervisé, l'apprentissage non supervisé (sans professeur) ne s'appuie sur aucune cible prédéfinie de la sortie. Il consiste à doter le système d'un mécanisme automatique, qui s'appuie sur des règles précises de regroupement afin de trouver les classes de référence avec une assistance minimale. Dans ce cas, les exemples en entrées sont introduits en un grand nombre par l'utilisateur sans indiquer leurs classes.

b. Phase de reconnaissance et décision

Dans un système de reconnaissance de l'écriture manuscrite, la décision est l'ultime étape de reconnaissance. L'idée principale est de déterminer l'appartenance d'une forme non connue à une classe prédéfinie à partir de la description en paramètres de la forme (à partir des vecteurs de caractéristiques extraits de cette forme). La reconnaissance peut conduire à un succès, dans le cas où la réponse est unique, c'est-à-dire, un seul modèle répond à la description de la forme du caractère. Comme elle peut conduire à une confusion si la réponse est multiple, dans le cas où plusieurs modèles correspondent à la description.

1.4.1.5. Post-traitement

L'étape de post-traitement est la dernière étape du système de reconnaissance proposé. Les résultats du système contiennent généralement des erreurs en raison de problèmes de classification et de segmentation des caractères (Dhaka, 2015). Cette phase a pour but de corriger les erreurs effectuées lors de la reconnaissance. Bien qu'elle soit facultative, l'utilisation du post-traitement est nécessaire, car l'écriture manuscrite est un processus qui engendre beaucoup de bruit lors de la reconnaissance. Elle permet de corriger des erreurs minimales qui n'ont pas de sens au vu de la langue traitée (Mioulet, 2015), dans le but d'améliorer le taux de reconnaissance. Pour la correction des erreurs de reconnaissance, les systèmes de reconnaissance de l'écriture manuscrite appliquent des techniques de post-traitement contextuel. Les deux techniques de post-traitement les plus courantes pour la correction d'erreurs sont la recherche dans le dictionnaire et l'approche statistique.

1.4.2. Les approches de classification

Diverses approches de reconnaissance sont utilisées dans le domaine de reconnaissance de formes qui, à leur tour, s'adaptent à la reconnaissance de l'écriture. Ces approches de reconnaissance peuvent être regroupées en quatre groupes principaux: Template Matching, approche statistique, approche structurelle et les approches connexionnistes.

1.4.2.1. Template Matching (ou appariement de formes)

La méthode de reconnaissance de caractères la plus simple consiste à faire correspondre les prototypes stockés avec le caractère ou le mot à reconnaître. D'une manière générale, l'opération d'appariement détermine le degré de similitude entre deux vecteurs (groupe de pixels, courbure, formes, etc.) dans l'espace des caractéristiques. Les techniques d'appariement peuvent être étudiées en trois classes Direct Matching, Deformable Templates et Elastic Matching et Relaxation Matching. (Dhaka, 2015).

1.4.2.2. Approche statistique

Dans cette approche la reconnaissance est basée sur l'étude statistique des mesures que l'on effectue sur les formes à reconnaître (Djerouni, 2021). La théorie de la décision statistique concerne les fonctions de décision statistique et un ensemble de critères d'optimalité, qui maximise la probabilité du modèle observé compte tenu du modèle d'une

certaine classe, en d'autre terme, prendre une décision du type plus forte probabilité d'appartenance à une classe.

Cette approche bénéficie des méthodes d'apprentissage automatique qui s'appuient sur des bases théoriques connues telles que : Le classifieur bayésien, la méthode de k plus proches voisins (K-PPV), Les machines à vecteurs de support (SVM), les modèles de Markov cachés, etc.

1.4.2.3. Approche structurelle

Les approches structurelles ou syntaxiques cherchent à structurer l'information en décrivant l'organisation géométrique (ou la structure) d'une forme complexe à partir des composantes les plus élémentaires appelées primitives. Elles reposent sur une représentation hiérarchique des formes. Chaque forme est vue comme un ensemble de sous-formes qu'on appelle "pattern", elles-mêmes composées de patterns plus petits. Les plus petits patterns (i.e. primitives) sont des caractéristiques, par exemple un segment de droite, une occlusion ou un arc pour les caractères manuscrits (Aries, 2009). La différence que l'on peut distinguer entre les méthodes statistiques et ces méthodes est que ces caractéristiques sont des formes élémentaires (les primitives sont de type topologiques) et non pas des mesures (Djerouni, 2021).

1.4.2.4. Approches connexionnistes

L'approche connexionniste est basée sur l'utilisation des réseaux de neurones, ces derniers sont définis comme une architecture informatique qui consiste en une interconnexion massivement parallèle de processeurs « neuronaux » adaptatifs. En raison de sa nature parallèle, il peut effectuer des calculs à un rythme plus élevé par rapport aux techniques classiques. Il peut facilement s'adapter aux changements dans les données et apprendre les caractéristiques des signaux d'entrée, en raison de sa nature adaptative. Un réseau de neurones est composé de plusieurs nœuds. La sortie d'un nœud est transmise à un autre dans le réseau et la décision finale dépend de l'interaction complexe de tous les nœuds (Dhaka, 2015). Le réseau de neurones le plus familier utilisé dans les systèmes de reconnaissance de l'écriture est le Perceptron Multi-Couches (PMC).

1.4.2.5. Combinaison des classifieurs

Cette approche est basée sur la combinaison des classifieurs pour bénéficier de leur éventuelle complémentarité. L'objectif principal de la combinaison des classifieurs est d'améliorer la qualité de reconnaissance du système par rapport à chacun des classifieurs individuels, c'est-à-dire obtenir des décisions plus robustes et compenser les faiblesses de chaque classifieur. Cette approche a été utilisée avec succès en reconnaissance de formes et en particulier en reconnaissance de caractères manuscrits. Il existe trois architectures de combinaison de classifieurs (Es Saady, 2012) (voir Figure 1.11) :

- La combinaison parallèle où la forme à reconnaître est présentée à plusieurs classifieurs indépendants dont les sorties sont combinées pour fournir la décision finale.
- La combinaison séquentielle consiste à organiser les classifieurs en niveaux successifs de décision permettant le filtrage progressif des décisions et la réduction de l'ambiguïté.
- Les approches hybrides consistent à combiner les architectures séquentielles et parallèles.

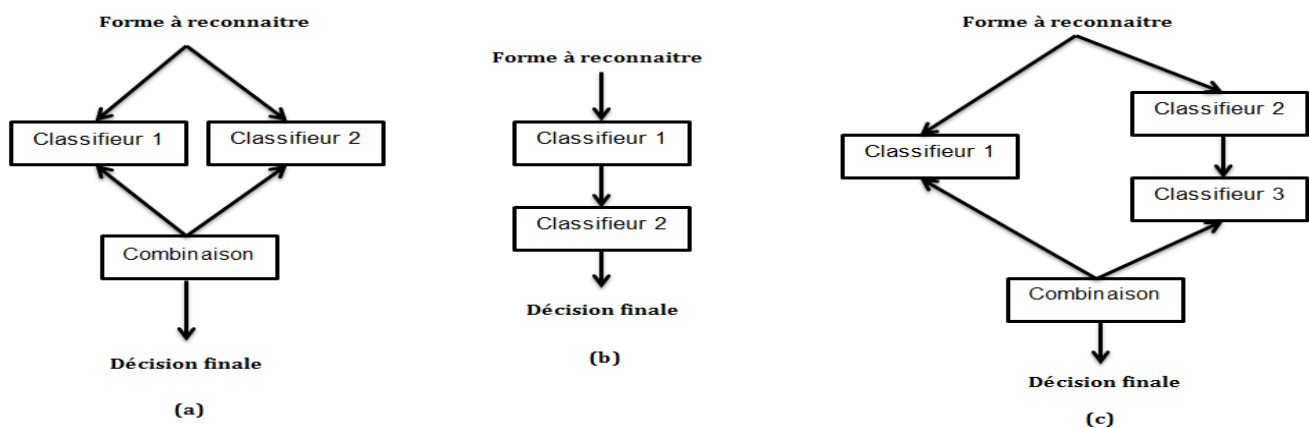


Figure 1.11 : Les 3 architectures de combinaison des classifieurs (Es Saady, 2012)

(a) approche parallèle, (b) approche séquentielle, (c) approche hybride

1.5. Domaines d'applications

Il y a eu une croissance significative dans l'application de la reconnaissance de l'écriture manuscrite hors ligne au cours des dernières décennies. Cela est dû à son essor important dans les domaines liés au développement des intérêts économiques. Dans ce qui suit, nous présentons peu d'entre eux.

1.5.1. Domaine bancaire

Les gens font des chèques régulièrement et les chèques jouent toujours un rôle majeur dans la plupart des transactions autres qu'en espèces. La procédure actuelle de traitement des chèques nécessite qu'un employé de banque lise et entre manuellement les informations présentes sur un chèque et vérifie également les entrées telles que la signature et la date. Comme un grand nombre de chèques doivent être traités chaque jour dans une banque, un système de reconnaissance de texte manuscrit peut économiser des coûts et des heures de travail humain.

1.5.2. Assurances et administrations

Il s'agit principalement des applications de reconnaissance des caractères précaisés manuscrits à la lecture et traitement des documents manuscrits tels que : Analyse des

formulaires, des bordereaux de commande, des feuilles de déclaration URSSAF, des constats d'assurance, etc., pour les grandes administrations (Caisses d'allocations familiales, Trésor Public, etc.)

1.5.3. La numérisation des documents

D'énormes quantités de connaissances historiques sont numérisées en téléchargeant les images numérisées pour un accès au monde entier. Mais cet effort n'est pas très utile jusqu'à ce que le texte des images soit identifié et puisse être indexé, interrogé et parcouru. La reconnaissance de l'écriture manuscrite joue un rôle clé dans la revitalisation des documents médiévaux et du XXe siècle, des cartes postales, des études de recherche, etc.

1.5.4. Santé et pharmacie

La numérisation des ordonnances des patients, ainsi que l'inscription des patients et la numérisation des formulaires. En ajoutant la reconnaissance de l'écriture manuscrite à leur boîte à outils de services, les hôpitaux/pharmacie peuvent améliorer considérablement l'expérience utilisateur.

1.5.5. La police et la sécurité

La reconnaissance des numéros minéralogiques pour le contrôle routier, l'authentification de manuscrits et l'identité des scripteurs.

1.6. Quelques travaux récents

Plusieurs travaux ont été proposés dans le domaine de la reconnaissance de l'écriture manuscrite. Cette section est consacrée à réaliser un état de l'art sur les différents travaux proposés par des chercheurs afin de réaliser des systèmes de reconnaissance de l'écriture manuscrite. Les systèmes présentés utilisent des algorithmes de reconnaissance différents, et travaillent sur différentes bases.

Le tableau suivant présente un récapitulatif précisant les performances de certain systèmes de reconnaissance de l'écriture manuscrite:

Tableau.1.1 : Résumé des résultats de quelques systèmes de reconnaissance de l'écriture manuscrite

Systeme	Classifieur	Base de données	Taux de reconnaissance (%)
(Peng et Yin, 2017)	HMM-CNN	EMNIST	87.77
(Singh et al., 2017)	CNN	EMNIST Digits	99.62
(Amrouch et al., 2018)	HMM-CNN	IFN / ENIT	89.23
(Arbain et al., 2018)	SVM	MNIST	93.18
	MLP		96.51
(Chen et al., 2018)	KNN	MNIST	91.7
	NN		91.8
(Prasad et Sanyal., 2018)	KNN-SVM	MNIST	99.26
(Shawon et al., 2018)	CNN	EMNIST Balanced	90.59
		EMNIST Digits	99.79
(Ali et Suresha, 2019)	CNN	AHDB - IFN/ENIT	97.7
	KNN		95.3
	SVM		96.3

	KNN-SVM		98.4
(Assegie et Nair, 2019)	Decision Tree	KAGGLE	83.4
(Baldominos et al., 2019)	HMM-CNN	EMNIST	95,44
(dos Santos et al., 2019)	CNN	EMNIST Digits	99.775
(Hassan et al., 2019)	SVM	AHDB	99.08
(Jayasundara et al., 2019)	CNN	EMNIST Letters	95.36
		EMNIST Digits	99.79
(Siddhu et al., 2019)	SVM	IFN / ENIT	Phase 01 : 96,71
			Phase 02 : 94,52
(Mishra et al., 2020)	Random Forest	MNIST	92.84
	KNN		92.60
	SVM		94,75
(Ali et Mallaiah, 2021)	CNN-SVM	AHDB	99.00
		AHCD	99.71

		HACDB	99.85
		IFN/ENIT	98.58
(Erdoğan et Tümer, 2021)	CNN	EMNIST	87.81
(Mustafa et al., 2021)	CNN	EMNIST Balanced	99
		EMNIST Letters	95
		EMNIST MNIST	89
(Pashine et al., 2021)	SVM	MNIST	94.005
	MLP		98.85
	CNN		99.31
(Ramesh et al., 2021)	Foret Aléatoire – CNN	Kannada-MNIST	99.66
(Shetty et al., 2021)	CNN-LSTM	MNIST	98.23
		EMNIST	85.17

1.7. Conclusion

Dans ce chapitre, nous avons présenté, dans un premier temps, un aperçu du domaine de reconnaissance de l'écriture en général, et ce, à travers une taxonomie des systèmes de reconnaissance selon plusieurs critères. Par la suite, nous avons décrit les différentes étapes constituant le processus de reconnaissance de l'écriture manuscrite hors-ligne. Pour chacune de ces étapes, nous avons abordé les différentes méthodes et techniques utilisées dans le cadre de la reconnaissance de l'écriture manuscrite. Et pour finir, nous avons présenté un état de

l'art récent et les différentes approches développées dans le domaine de l'écriture manuscrite hors-ligne et notamment celles évaluées sur les bases de données MNIST et EMNIST.

Dans le chapitre suivant, nous présenterons les techniques d'apprentissage automatique et d'apprentissage profond et plus précisément les réseaux de neurones convolutifs.

L'APPRENTISSAGE AUTOMATIQUE ET L'APPRENTISSAGE PROFOND

Sommaire

2.1.	Introduction	26
2.2.	Apprentissage automatique et apprentissage profond	26
2.2.1.	Apprentissage automatique (Machine Learning)	26
2.2.2.	Apprentissage profond (Deep Learning)	30
2.2.3.	Différence entre l'apprentissage automatique et l'apprentissage profond	30
2.3.	Réseaux de neurones profonds	32
2.3.1.	Concept de réseaux de neurones	32
2.3.2.	Types des réseaux de neurones	33
2.4.	Réseaux de neurones convolutifs	37
2.4.1.	Architecture globale	37
2.4.2.	Évolution architecturale des réseaux de neurones convolutifs.....	39
2.5.	Quelques systèmes existants de reconnaissance de caractères manuscrits basé CNN 43	
2.6.	Conclusion	46

2.1. Introduction

Jusqu'aux années 2012-2013, les approches du problème de la reconnaissance de l'écriture manuscrite reposaient sur l'extraction de caractéristiques à partir des images qui nécessitaient habituellement une expertise du domaine (Sueiras, 2021). Une amélioration par rapport aux approches précédentes a été l'introduction des réseaux de neurones (NN) qui permettent l'extraction automatique de caractéristiques. Ces modèles ont apporté des améliorations significatives dans les résultats de la reconnaissance de l'écriture manuscrite qui n'étaient pas considérés comme possibles auparavant. Depuis lors, les réseaux de neurones sont devenus les méthodes les plus couramment utilisées pour résoudre le problème de la reconnaissance de l'écriture manuscrite hors ligne. Différents auteurs proposent plusieurs architectures et utilisent différents types de réseaux pour faire face à ce problème.

Dans ce chapitre, nous aborderons dans un premier temps les différents concepts liés au Machine Learning et au Deep Learning. Ensuite, nous nous concentrerons sur les réseaux de neurones et plus particulièrement aux réseaux de neurones convolutifs (CNN) que nous avons choisis pour la reconnaissance de l'écriture manuscrite et qui sera aussi utilisé comme module important dans notre système de reconnaissance de caractères manuscrits. Nous présenterons, les bases des réseaux de neurones profonds. Par la suite, nous présenterons les différentes architectures de réseaux de neurones utilisées pour résoudre le problème de la reconnaissance de l'écriture manuscrite. Finalement, nous décrirons en détail les réseaux de neurones convolutifs (CNN) et leurs applications pour la reconnaissance de l'écriture manuscrite.

2.2. Apprentissage automatique et apprentissage profond

L'apprentissage profond (Deep Learning) est un sous-ensemble de l'apprentissage automatique (Machine Learning) qui, à son tour, est un sous-ensemble de l'intelligence artificielle (IA). C'est-à-dire l'apprentissage automatique et l'apprentissage profond relèvent tous les deux de la vaste classe de l'intelligence artificielle.

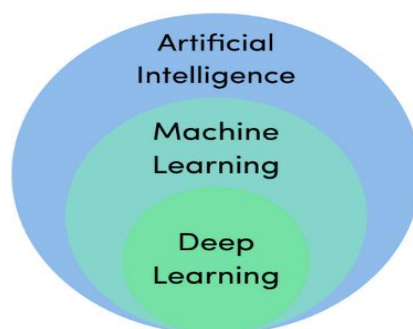


Figure 2.1 : Domaines en Intelligence Artificielle (Darwish et al., 2020)

2.2.1. Apprentissage automatique (Machine Learning)

L'apprentissage automatique est le terme général désignant d'après Arthur Samuel (1959), la capacité d'un ordinateur à apprendre sans être explicitement programmés. La théorie computationnelle de l'apprentissage dans l'intelligence artificielle a développé

l'apprentissage automatique qui explore l'analyse et la construction d'algorithmes qui apprennent à partir des données brutes, entraînent le système et font des prédictions basées sur ces données d'entraînement (Chauhan et Singh, 2018). L'apprentissage automatique comprend deux types de méthodologie d'apprentissage, l'apprentissage supervisé et non supervisé. Dans l'apprentissage supervisé, l'étiquette ou la valeur cible est fournie pour chaque échantillon. Tandis que dans l'apprentissage non supervisé, la valeur cible n'est pas fournie.

2.2.1.1. Apprentissage supervisé

L'apprentissage supervisé est l'approche d'apprentissage automatique définie par son utilisation d'ensembles de données étiquetés pour former des algorithmes afin de classer les données et de prédire les résultats. L'ensemble de données étiqueté a une sortie étiquetée correspondant aux données d'entrée pour que la machine comprenne ce qu'il faut rechercher dans les données invisibles. Il existe deux principaux domaines dans lesquels l'apprentissage automatique supervisé est utile : les problèmes de classification et les problèmes de régression. La classification consiste à prendre une valeur d'entrée et à l'affecter à une valeur discrète. Dans les problèmes de classification, la sortie est généralement une classe ou une catégorie. Tandis que la régression est liée aux données continues. En la régression, les valeurs de sortie prédites sont des nombres réels.

2.2.1.2. Apprentissage non supervisé

L'apprentissage non supervisé est un type d'apprentissage automatique dans lequel les algorithmes reçoivent des données qui ne contiennent aucune étiquette ou instruction explicite sur ce qu'il faut en faire. L'objectif est que l'algorithme d'apprentissage trouve lui-même une structure dans les données d'entrée. Les modèles d'apprentissage non supervisé peuvent effectuer des tâches plus complexes que les modèles d'apprentissage supervisé. L'apprentissage non supervisé est effectué dans le contexte de regroupement et d'association. Le clustering est le type d'apprentissage non supervisé qui consiste à trouver des modèles cachés dans les données en fonction de leurs similitudes ou de leurs différences. Ces modèles peuvent être liés à la forme, à la taille ou à la couleur et sont utilisés pour regrouper des éléments de données ou créer des clusters. Alors que, l'association est le type d'apprentissage non supervisé consiste à découvrir les relations entre les éléments de données.

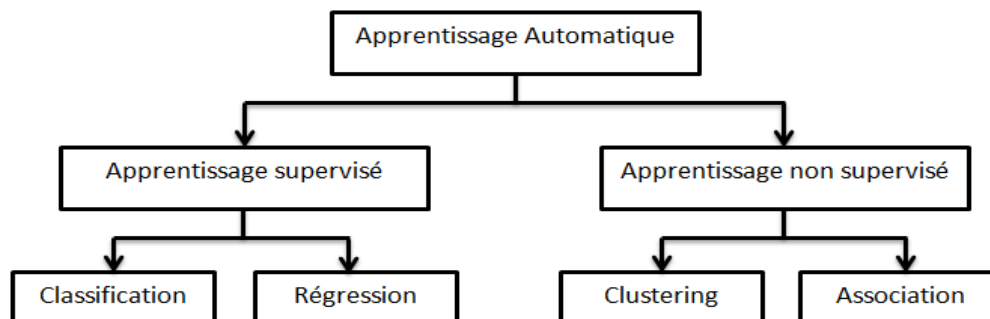


Figure 2.2 : Classification des approches d'apprentissage automatique (Chauhan et Singh, 2018)

2.2.1.3. Méthodes d'apprentissage automatique

Il existe plusieurs méthodes d'apprentissage automatique. Dans cette section, nous décrirons brièvement les modèles de classification les plus populaires et les plus utilisées.

a. Les modèles de Markov cachés

Les modèles de Markov cachés (HMM), sont utilisés pour reconnaître les caractères, les mots et les lignes. Les HMM sont largement utilisés dans le domaine de la reconnaissance de texte. Les HMM sont des modèles graphiques probabilistes, qui modélisent une séquence en intégrant des connaissances a priori : modèle de langage, lexicale. Ces modèles statistiques sont utilisés à l'origine pour la reconnaissance vocale de manière efficace. En raison du succès des HMM dans la reconnaissance vocale et des similitudes entre la reconnaissance de la parole et l'écriture manuscrite cursive, les HMM ont été étendus pour la reconnaissance de l'écriture manuscrite en ligne et hors ligne. Dans les systèmes de reconnaissance hors ligne basés sur les HMM, le concept principal est transformer le mot image en une séquence d'observations (AlKhateeb et al., 2011).

La figure 2.3, illustre le cas d'un HMM gauche-droite, qui est le modèle référence pour la reconnaissance de l'écriture.

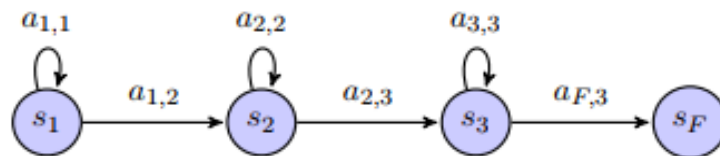


Figure 2.3: Exemple d'un HMM gauche-droite (Mioulet, 2015)

b. Machines à vecteurs de support

Les machines à vecteurs de support (SVM) sont des classificateurs particuliers basés sur le principe de maximisation des marges et destinées à résoudre des problèmes de la classification binaire, qui ont été introduit dans l'apprentissage automatique par le mathématicien Vladimir Vapnik (Adankon et al., 2009). Les classificateurs SVMs sont des algorithmes qui maximisent la marge entre les classes du problème à résoudre et réduisent au minimum l'erreur de classification. L'objectif de la marge maximale est de faire séparer deux classes par un hyperplan de telle sorte que la distance par rapport aux vecteurs supports soit maximale. Les SVMs sont des classificateurs puissants qui ont été utilisées avec succès dans de nombreux problèmes de reconnaissance de formes. Ils se sont également avérés performants dans le domaine de la reconnaissance des caractères manuscrits, mais peu applicables à la reconnaissance de mots manuscrits.

La figure 2.4, illustre le principe d'une machine à vecteurs de support

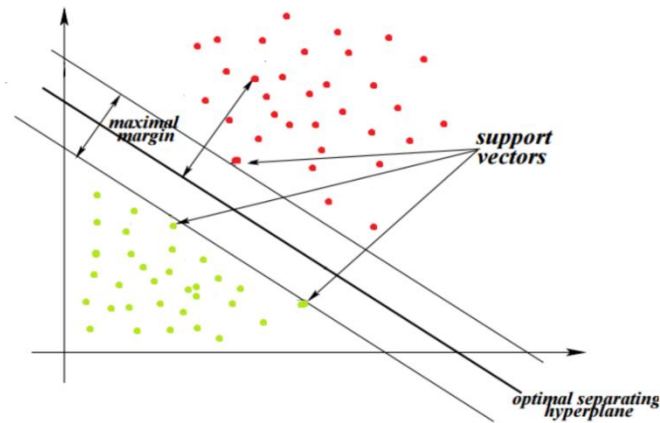


Figure 2.4 : Machine à vecteurs de support (Kumar, 2021)

c. Le classifieur du K plus proches voisins

L'algorithme K Plus Proche Voisins (KPPV) est un algorithme d'apprentissage automatique qui est généralement utilisé dans la reconnaissance de formes. Cet algorithme est une approche statistique sans phase d'apprentissage. L'idée principale est d'affecter une forme inconnue à la classe de son plus proche voisin en la comparant aux formes stockées dans une base d'apprentissage. Le classement de nouvelles formes opère en calculant la distance euclidienne entre la représentation vectorielle de la forme à classer et celles des formes déjà classées pour trouver les K formes les plus proches, les K formes les plus proches sont sélectionnées et la forme est assignée à la classe majoritaire.

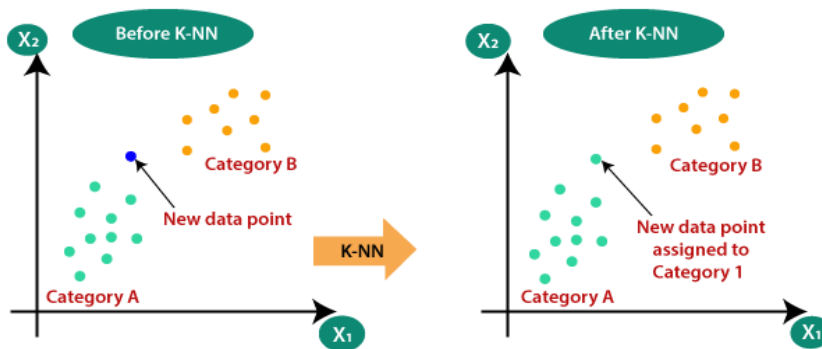


Figure 2.5: K plus proches voisins [Web 04]

d. Le classifieur Bayes naïf

La classification bayésienne est une approche fondamentale en reconnaissance de formes. Le classificateur bayésien est un algorithme de l'apprentissage automatique utilisé pour la classification. Il est particulièrement utilisé dans la catégorisation de textes. Ce dernier est basé sur le théorème de Bayes, qui vise à estimer la probabilité conditionnelle d'une catégorie sachant un texte et affecte au texte la catégorie la plus probable.

e. Arbre de décision

L'approche par arbre de décision est un autre algorithme d'apprentissage supervisé dans lequel tout l'ensemble de données est étiqueté. Les arbres de décision sont utilisés pour la classification en triant les classes en fonction des valeurs des paramètres. ID3, C4.5, CHAID

et CART sont quelques-uns des algorithmes appartenant à l'arbre de décision. L'avantage majeur de cette approche est qu'elle est capable de gérer des attributs numériques aussi bien que catégoriques. Cette méthode est valable pour les petits ensembles de données, mais entraîne un retard pour les grands ensembles de données.

2.2.2. Apprentissage profond (Deep Learning)

L'apprentissage profond est un sous-domaine de l'apprentissage automatique. L'apprentissage profond est une approche avancée d'apprentissage automatique qui est utilisée pour rendre les machines capables d'extraire, d'analyser et de comprendre automatiquement les informations utiles à partir des données brutes. L'apprentissage profond utilise les réseaux de neurones qui simulent la prise de décision humaine. Par conséquent, l'apprentissage profond entraîne la machine à faire ce que le cerveau humain fait naturellement (Boesch, 2021). Les résultats obtenus à partir de l'apprentissage profond sont bien meilleurs que l'approche d'apprentissage automatique (Chauhan et Singh, 2018). Le réseau neurones profond utilise un modèle non linéaire d'architecture à plusieurs couches cachées qui permet au système d'apprendre la relation complexe entre l'entrée et la sortie. L'avantage de l'apprentissage en profondeur par rapport à l'apprentissage automatique est qu'il ne nécessite pas de fonctionnalités extraites manuellement ou fabriquées à la main comme dans l'apprentissage automatique. L'apprentissage en profondeur extrait automatiquement les caractéristiques des données brutes, les traite et prend la décision en fonction de cela. Il existe de nombreux types de réseaux de neurones disponibles qui peuvent être classés en fonction de leur : Structure, Flux de données, Neurones utilisés et leur densité, Couches et leurs filtres d'activation en profondeur etc. Dans les sections suivantes, nous examinerons les réseaux profonds les plus couramment utilisés : Les perceptrons Multi-Couches (MLP), les réseaux de neurones convolutifs (CNN) et également les réseaux de neurones récurrents (RNN), mais nous nous concentrerons principalement sur les neurones convolutifs (CNN).

2.2.3. Différence entre l'apprentissage automatique et l'apprentissage profond

Les algorithmes d'apprentissage profond sont des algorithmes d'apprentissage automatique et fonctionne de manière similaire. Cependant, ses capacités sont différentes. Les résultats obtenus à partir de l'apprentissage en profondeur sont bien meilleurs que l'approche d'apprentissage automatique, et cela est principalement dû à la structure de l'algorithme réseau de neurones artificiels (ANN), à l'extraction automatique des caractéristiques et au besoin d'une quantité importante de données (Wolfewicz ,2021).

a. La structure de l'algorithme ANN

L'apprentissage automatique utilise des algorithmes avec une structure assez simple, comme la régression linéaire ou un arbre de décision pour analyser les données, apprendre de ces données et prendre des décisions éclairées en fonction de ce qu'il a appris, tandis que l'apprentissage en profondeur structure les algorithmes en couches pour créer un réseau de neurones artificiels complexe et entrelacé qui peut apprendre et prendre des décisions intelligentes par lui-même.

b. Extraction automatique des caractéristiques

Les algorithmes d'apprentissage en profondeur nécessitent beaucoup moins d'intervention humaine. En tant qu'algorithme d'apprentissage profond, les fonctionnalités sont extraites automatiquement via son réseau neuronal., l'algorithme n'a pas besoin d'un ingénieur logiciel pour identifier les fonctionnalités, contrairement aux algorithmes d'apprentissage automatique (voir Figure 2.6 ci-dessous)

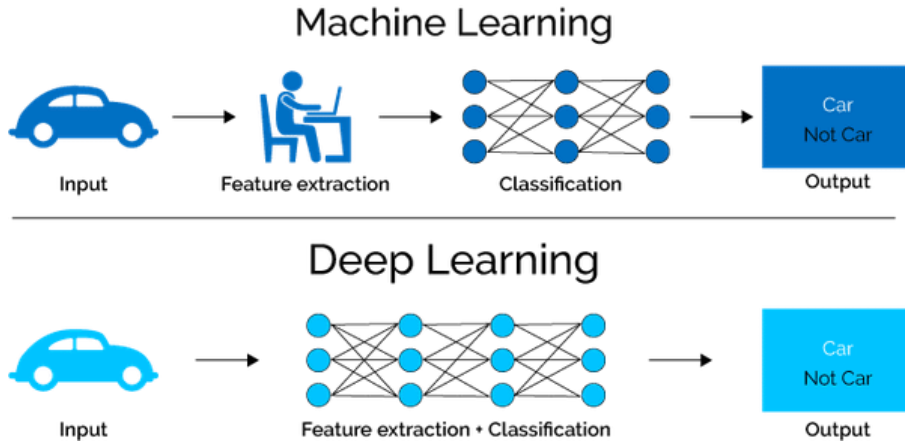


Figure 2.6: Machine Learning vs Deep Learning (Wolfewicz, 2021)

c. Quantité de données

L'apprentissage en profondeur nécessite beaucoup plus de données qu'un algorithme d'apprentissage automatique traditionnel pour fonctionner correctement, en raison de la structure multicouche complexe. L'apprentissage profond fonctionne efficacement si la quantité de données augmente rapidement alors que l'apprentissage automatique cesse de s'améliorer après un point de saturation. La figure 2.7 montre le fonctionnement de l'apprentissage automatique et l'apprentissage profond avec la quantité de données.

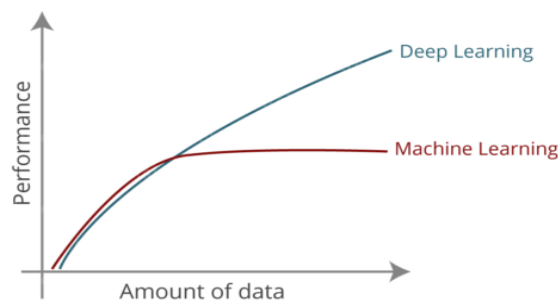


Figure 2.7: Fonctionnement de l'apprentissage automatique et l'apprentissage profond (Alom et al., 2019)

2.3. Réseaux de neurones profonds

Inspirés des systèmes nerveux biologiques, les réseaux de neurones artificiels (ANN) visent à atteindre leur polyvalence par l'apprentissage. Les ANN sont couramment utilisées dans l'intelligence artificielle, l'apprentissage automatique et la reconnaissance de formes. Il y a eu des recherches substantielles sur la façon dont la structure du cerveau humain atteint un tel niveau de polyvalence. Ces recherches ont fourni des informations importantes, mais les conclusions sont loin d'expliquer complètement le fonctionnement complexe du cerveau. Bien que la structure interne détaillée du cerveau humain soit connue, les chercheurs sont toujours incapables de la simuler avec la technologie actuelle en raison de sa robustesse. Leur but est donc de construire un réseau de neurones avec un bon rapport entre sa taille et son efficacité.

2.3.1. Concept de réseaux de neurones

Les modèles de réseaux de neurones profonds ont été inspirés à l'origine par la neurobiologie. À un niveau élevé, un neurone biologique reçoit plusieurs signaux à travers les synapses en contact avec ses dendrites et envoie un seul flux de potentiels d'action à travers son axone. La complexité des entrées multiples est réduite en catégorisant ses modèles d'entrée. Inspirés par cette intuition, les modèles de réseaux de neurones artificiels sont composés d'unités qui combinent plusieurs entrées et produisent une seule sortie. Les réseaux de neurones ciblent des fonctionnalités semblables à celles du cerveau et sont basés sur un simple neurone artificiel : une fonction non linéaire d'une somme pondérée des entrées. Ces pseudo-neurones sont rassemblés en couches, et les sorties d'une couche deviennent les entrées de la suivante dans la séquence.

La figure 2.8 illustre une comparaison simple des éléments constitutifs des neurones biologiques avec ceux des neurones artificiels. (Boesch, 2021)

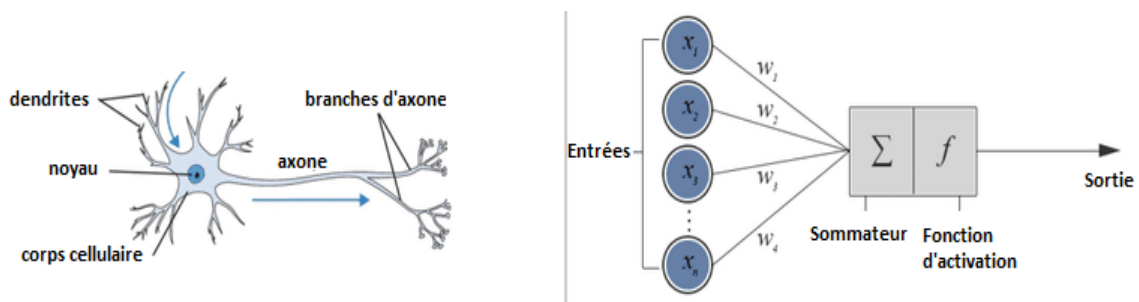


Figure 2.8 : Du neurone biologique au neurone artificiel (Hossam, 2020)

En bref, chaque neurone reçoit une version multipliée des entrées et des poids aléatoires qui est ensuite ajoutée avec une valeur de biais statique (unique à chaque couche de neurones), celle-ci est ensuite transmise à une fonction d'activation appropriée qui décide de la valeur finale à donner du neurone. Différentes fonctions d'activation sont disponibles selon la nature des valeurs d'entrée. Une fois que la sortie est générée à partir de la couche de réseau neuronal finale, la fonction de perte est calculée et la rétropropagation est effectuée là où les poids sont

ajustés pour rendre la perte minimale. L'opération globale se concentre sur la recherche de valeurs optimales de poids.

2.3.2. Type des réseaux de neurones

2.3.2.1. Le perceptron Multi-Couches

Un perceptron multicouche (MLP) est une classe d'un réseau de neurones artificiels (ANN). Les modèles MLP sont le réseau neuronal profond le plus élémentaire, composé d'une série de couches entièrement connectées (Boesch, 2021). Chaque neurone d'une couche est connecté en entrée à tous les neurones de la couche précédente, et en sortie aux neurones de la couche suivante. Dans un perceptron Multi-Couches, le réseau reçoit un signal d'entrée x à travers un ensemble de nœuds initiaux appelés couche d'entrée et il produit un signal de sortie y à travers d'autres nœuds appelés couche de sortie. Les couches intermédiaires, à travers lesquelles le signal circule et se transforme, sont appelées couches cachées h . La figure 2.9 montre un schéma de réseau neuronal de base avec les composants ci-dessus. Dans ce cas, La couche d'entrée n'est pas réellement une couche, il s'agit du vecteur de données, elle n'est pas comptée dans le nombre de couches d'un réseau de neurones bien que souvent représentée dans les illustrations. La couche de sortie réalise l'association entre les classes de l'entrée et la représentation produite par les couches cachées, en effet chaque neurone sur la couche de sortie représente une classe. Les couches cachées, relient la couche d'entrée à la couche de sortie, elles réalisent un ensemble de transformations des données d'entrée permettant à la couche de sortie d'effectuer une décision (Mioulet, 2015).

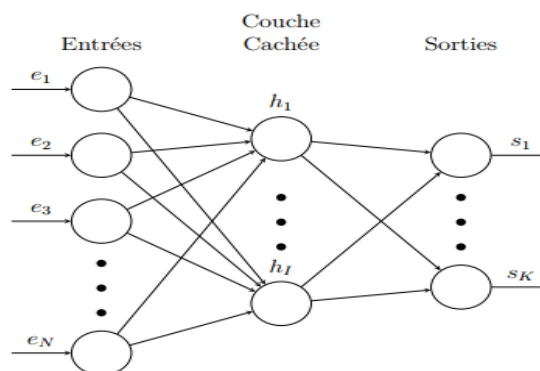


Figure 2.9: Perceptron multi-couches (Mioulet, 2015)

Les perceptrons multicouches sont appelés feed-forward (propagation avant), parce que les informations d'entrée x suivent un seul chemin vers l'avant jusqu'à ce qu'elles soient transformées en informations de sortie y sans aucune rétroaction.

La figure 2.10 montre le fonctionnement d'un nœud particulier du réseau (Sueiras, 2021). Chaque nœud reçoit plusieurs signaux d'entrée x_i à travers ses fronts entrants, modulés par leurs poids respectifs w_i . Il les fusionne au moyen d'une fonction linéaire de combinaison, qui est généralement la somme, et génère un signal de sortie y qui est un nombre réel. Cette sortie est modulée par une fonction d'activation non linéaire, avant d'être envoyée aux nœuds suivants du réseau. Cette activation est une fonction monotone qui limite généralement la

plage de variation du signal de sortie, et donc permet de répondre à un problème de classification avec des frontières non linéaires dans des espaces de grande dimension. Formellement, les signaux d'entrée x_1, \dots, x_n provenant de la couche précédente sont multipliés par leurs poids respectifs w_1, \dots, w_n et combiné au nœud, dans ce cas avec la fonction somme (sum). Au résultat précédent un biais b est ajouté et à cela la fonction d'activation f est appliquée.

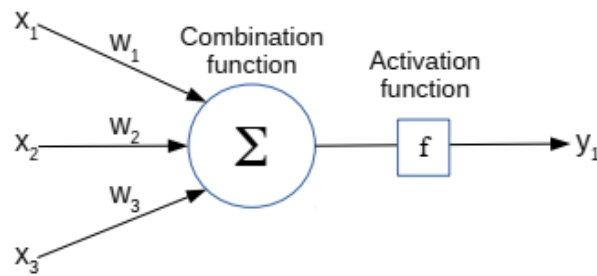


Figure 2.10: Neurone artificiel (Sueiras, 2021)

Le choix de la fonction d'activation pour les couches cachées réside dans sa dérivabilité et sa simplicité de calcul pour l'apprentissage comme par exemple la fonction sigmoïde ou la fonction tangente hyperbolique (Poisson, 2005).

- Sigmoïde (v_i) = $\frac{1}{1+e^{-v_i}}$ entre [0 et 1]
- Tangente hyperbolique (v_i) = $\frac{\sinh(v_i)}{\cosh(v_i)} = \frac{e^{2v_i}-1}{e^{2v_i}+1}$

Les représentations graphiques de ces deux fonctions sont indiquées dans la figure 2.11.

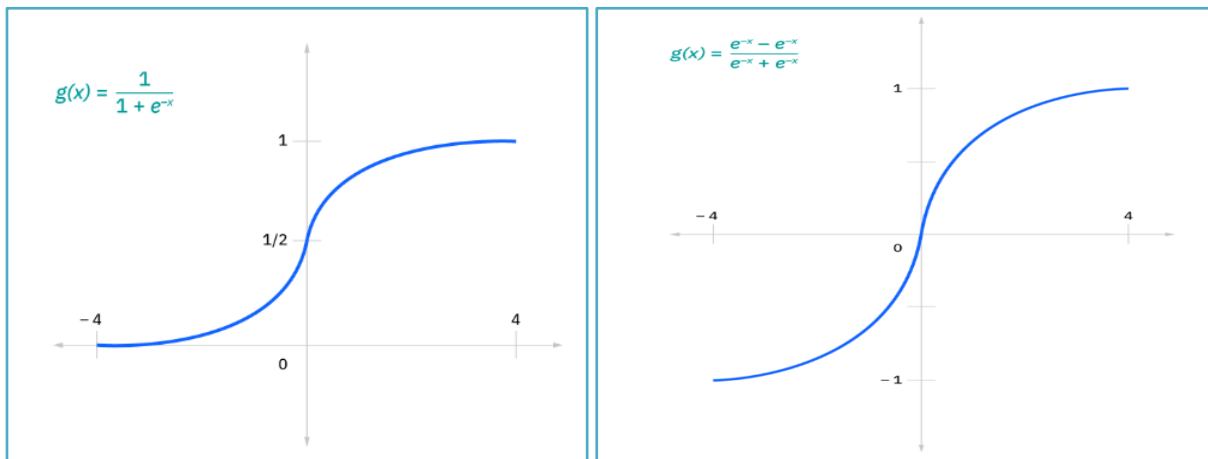


Figure 2.11 : Exemples de fonctions d'activation d'un perceptron Multi-Couches [Web 05]

La décision finale en reconnaissance de l'écriture se caractérise en général par un vecteur de probabilités d'une classe par rapport à l'entrée. Pour obtenir des sorties de type probabilités (entre 0 et 1), on utilise en sortie comme fonction de transfert la fonction dite Softmax (Poisson, 2005).

$$\text{Softmax}(v_i) = P(\text{Classe} = C_i | X = x_i) = \frac{e^{v_i}}{\sum_k e^{v_k}}$$

Avec :

- i : est l'indice du neurone correspondant à la classe C_i .
- k : est l'indice balayant tous les neurones de la même couche.
- v : est le potentiel synaptique du neurone.
- x : est l'entrée des neurones.

2.3.2.2. Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN) sont des algorithmes d'apprentissage en profondeur conçus pour le traitement d'images. Les CNN sont spécialisés dans la modélisation de l'information avec une structure en forme de grille, telle que les images sont considérées comme une grille de pixels bidimensionnelle. Ces réseaux utilisent l'opération mathématique convolutive au lieu de la multiplication matricielle standard pour calculer la sortie du réseau.

Les réseaux convolutifs (CNN) sont basés sur des neurones organisés en couches et peuvent donc apprendre des représentations hiérarchiques. Les neurones entre les couches sont connectés par des poids et des biais. La couche initiale est la couche d'entrée, et la dernière couche est la sortie. Entre les deux se trouvent des couches cachées transformant l'espace des caractéristiques de l'entrée de manière à ce qu'il corresponde à la sortie (Kattenborn et al., 2021). Une architecture convolutive standard se compose généralement de couches convolutionnelles, de couches de Pooling (couches de regroupement) et de couches Fully-Connected (couches entièrement connectées) comme le montre la figure 2.12 ci-dessous.

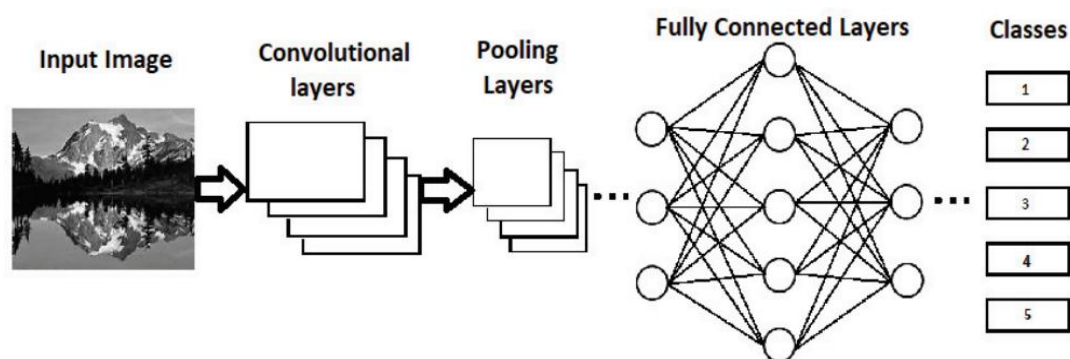


Figure 2.12 : Architecture convolutive standard (Chauhan et Singh, 2018)

Les architectures convolutive jouent un rôle fondamental dans tous les modèles travaillant avec des données de nature spatiale, en particulier des images 2D, mais aussi des séries temporelles 1D ou des vidéos 3D. Au fil des ans, plusieurs architectures convolutive ont été proposées pour différents problèmes liés à l'analyse d'images. Une revue complète de ces architectures sera présentée par la suite dans la section 2.4.

2.3.2.3. Réseaux de neurones récurrents (RNN)

Les réseaux de neurones récurrents (RNN) sont des techniques d'apprentissage profond bien conçues pour résoudre les problèmes des séries temporelles. Les RNN se distinguent par leur qualité dans différents domaines d'activité, tels que le traitement du signal, le traitement du langage naturel et la reconnaissance de la parole. Contrairement aux réseaux de neurones convolutifs, les RNN gèrent clairement les dépendances temporelles des données, puisque la sortie du neurone à l'instant $t - 1$ est utilisée avec l'entrée suivante, pour alimenter le neurone lui-même à l'instant t , cela est dû au fait que les réseaux de neurones récurrents sont des réseaux de neurones intégrant une mémoire. Cette prise en compte du contexte temporel permet aux réseaux de neurones une meilleure performance sur des problèmes pour lesquels l'aspect temporel est important, comme par exemple la reconnaissance de l'écriture manuscrite. Par conséquent, les RNN standard ne parviennent pas à apprendre en présence de décalages temporels supérieurs à 5 à 10 pas de temps discrets entre les événements d'entrée pertinents et les signaux cibles (Brownlee, 2017). Le problème de l'erreur de disparition (dit aussi, la perte du gradient) jette un doute sur la capacité des RNN standard à présenter des avantages pratiques significatifs par rapport aux réseaux de neurones que nous avons présentés précédemment dans la section 2.3.2.1 et 2.3.2.2 (MLP, CNN). Dans ce qui suit, nous présenterons brièvement le Long Short-Term Memory (LSTM) qui a été introduit comme solution à ce problème.

La figure 2.13, illustre un neurone récurrent isolé utilise en entrée à l'instant t sa sortie à $t - 1$.

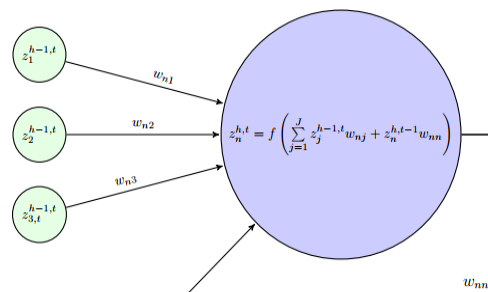


Figure 2.13 : Neuron recurrent (Mioulet, 2015)

a. Le Long Short Term Memory (LSTM)

Le Long Short Term Memory (LSTM) est un type particulier de réseau neuronal récurrent, capable de capturer efficacement les informations contextuelles des séquences d'entrée en raison de leur grande puissance de mémoire. Ce modèle a été introduit par Sepp Hochreiter et Juergen Schmidhuber comme solution au problème de la perte du gradient. LSTM peut apprendre à combler des décalages temporels minimaux supérieurs à 1000 pas de temps discrets en utilisant des cellules dans les couches cachées du réseau neuronal (Brownlee, 2017), qui ont trois portes : une porte d'entrée, une porte de sortie et une porte d'oubli. Ces portes contrôlent le flux d'informations nécessaires pour prédire la sortie dans le réseau.

La figure 2.14 illustre un schéma de cellule LSTM.

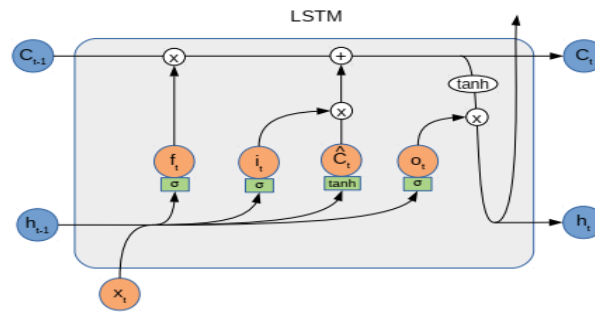


Figure 2.14: Cellule Long Short-Term Memory (LSTM) (Sueiras, 2021)

La porte i , nommée porte d'entrée, apprend quelles valeurs seront mises à jour dans le vecteur d'état. La porte f , appelée porte d'oubli, apprend les informations de l'état précédent qui peuvent être rejetées. Avec la sortie de ces deux couches, le réseau crée un vecteur de nouvelles valeurs d'état candidates c . Enfin, la porte o , nommée porte de sortie, apprend quelles informations iront à la sortie h .

2.4. Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNN) sont des réseaux feed-forward multicouche principalement utilisés pour résoudre des tâches difficiles de reconnaissance de formes basées sur l'image, grâce à leur architecture précise mais simple. Leur architecture s'inspire de l'étude de Hubel et Wiesel sur le traitement du signal neurobiologique dans le cortex visuel du chat (Hubel and Wiesel, 1968). Les CNN a fait preuve d'une performance exemplaire la classification et la segmentation des images, la détection d'objets, le traitement vidéo, le traitement du langage naturel et la reconnaissance vocale. La puissante capacité d'apprentissage de CNN est principalement due à l'utilisation de plusieurs étapes d'extraction de caractéristiques qui peuvent automatiquement apprendre des représentations à partir des données. Cette section se concentre sur les architectures des réseaux de neurones convolutifs (CNN), dans un premier temps nous allons illustrer l'architecture globale des CNN, par la suite nous présenterons l'évolution architecturale des réseaux de neurones convolutifs (CNN).

2.4.1. Architecture globale

Une architecture CNN typique comprend généralement des couches alternées de convolution et de Pooling suivies d'une ou plusieurs couches Fully-Connected à la fin. Un réseau convolutif est architecturalement divisé en deux parties importantes. Chaque partie est conçue pour remplir un objectif différent. La première partie du réseau effectue l'extraction des caractéristiques et est constituée de couches convolutionnelles et de pooling. La deuxième partie effectue une classification basée sur les fonctionnalités extraites en utilisant des couches Fully-Connected (Khan et al., 2020). L'agencement des composants CNN joue un rôle fondamental dans la conception de nouvelles architectures et donc dans l'amélioration des performances. Cette section décrit brièvement le rôle de ces couches dans une architecture CNN.

2.4.1.1. Couche de convolution (CONV)

Cette couche est l'unité principale de CNN dans laquelle la plupart des calculs sont effectués. Les couches convolutionnelles comprennent plusieurs filtres optimisables qui transforment l'entrée X en une sortie y communément appelée carte de caractéristiques (feature map) (Chauhan et Singh, 2018). Le nombre de filtres définit la profondeur d'une couche convolutionnelle. Cette opération est effectuée localement de manière itérative par convolution, qui est essentiellement le glissement du filtre sur la couche et le calcul du produit scalaire du filtre et des valeurs de la couche. Le résultat est une nouvelle couche de produits scalaires pour chaque filtre, également appelée carte de caractéristiques (voir figure 2.15). Les premières cartes de caractéristiques d'un CNN peuvent inclure des modèles à échelle simple et fine, tels que des coins, des cercles ou des bords, tandis que dans les couches plus profondes, la convolution révèle généralement des modèles plus abstraits et des concepts de niveau supérieur (Kattenborn et al., 2021). Les cartes de caractéristiques dérivées servent ensuite d'entrée pour la couche suivante.

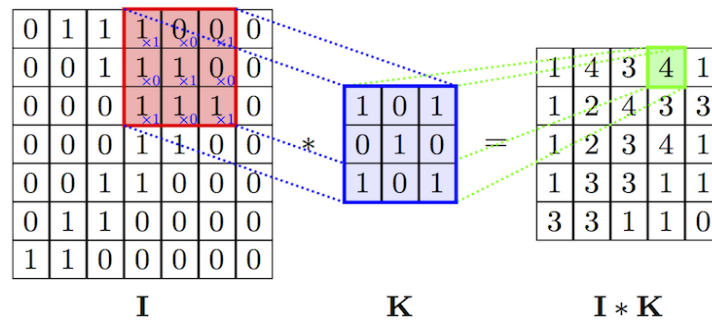


Figure 2.15 : Un exemple de l'opération de convolution (Rezgui, 2019)

2.4.1.2. Couche de Pooling (POOL)

Entre les séquences de plusieurs couches convolutionnelles, les cartes de caractéristiques sont généralement sous-échantillonnées spatialement à l'aide d'opérations de Pooling. Le Pooling décrit la transformation de plusieurs cellules en une seule cellule (voir Figure. 2), similaire au rééchantillonnage d'une image à une résolution spatiale plus grossière. La couche de Pooling est utilisée pour réduire la dimension spatiale des cartes de caractéristiques d'image et le nombre d'entités tout en préservant les informations discriminantes, ce qui à son tour entraîne une réduction de la complexité de calcul globale ainsi que, le risque de sur-ajustement. Le Pooling est définie par une taille de filtre, une foulée (la distance entre les opérations de Pooling consécutives) et une opération de réduction. Plusieurs opérations de Pooling sont utilisées tels que Average Pooling, Max Pooling, Stochastic Pooling, Spatial Pyramid Pooling, Spectral Pooling Et Multiscale Pooling. L'opération de Pooling la plus courante est le Max Pooling (Sueiras, 2021). L'idée de cette opération est de conserver les activations fortes (la valeur la plus grande parmi toutes les valeurs) dans le réseau.

La figure 2.16 ci-dessous représente une opération Max Pooling avec une taille de filtre 2x2 et une foulée de 2 c'est-à-dire la fenêtre de filtre se déplace de deux pixels, cette dernière

réduit la taille de la carte des caractéristiques d'entrée d'un facteur 4, créant une nouvelle matrice de sortie où chaque élément contiennent la valeur maximale des 4 cellules d'entrée selon filtres 2x2.

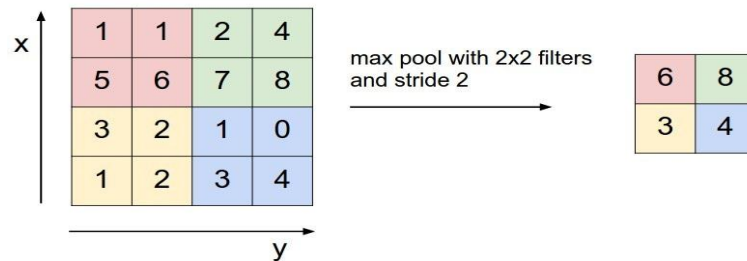


Figure 2.16 : Un exemple de l'opération de Max pooling (Chauhan et Singh, 2018)

2.4.1.3. Couche Fully-Connected (FC)

La couche Fully-Connected est similaire à la manière dont les neurones sont disposés dans un réseau neuronal traditionnel (NN). Par conséquent, chaque nœud d'une couche Fully-Connected est directement connecté à chaque nœud de la couche précédente et de la couche suivante (Albawi et al., 2017), comme illustré dans la Figure 2.17 Elle convertit une carte de caractéristiques de haute dimension en une seule dimension (1D) par un raisonnement de haut niveau qui donne la probabilité que la caractéristique appartienne à une classe spécifique.

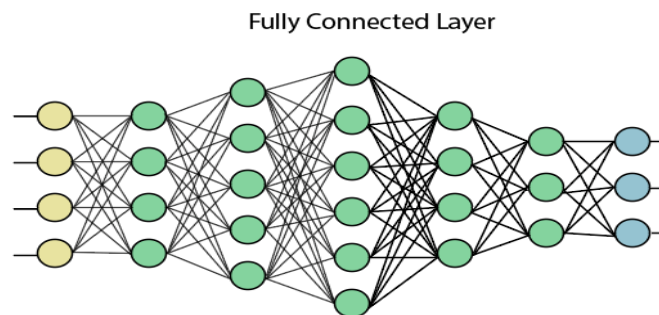


Figure 2.17 : Couche Fully-Connected [Web 06]

2.4.2. Évolution architecturale des réseaux de neurones convolutifs

De la fin des années 1990 jusqu'en 2000, diverses améliorations de la méthodologie et de l'architecture d'apprentissage de CNN ont été apportées pour rendre CNN évolutif aux problèmes de grande taille, hétérogènes, complexes et multi-classes. Les innovations dans les CNN incluent différents aspects tels que la modification des unités de traitement, les stratégies d'optimisation des paramètres et des hyper-paramètres, les modèles de conception et la connectivité des couches, etc. Les applications basées sur CNN sont devenues courantes après les performances exemplaires d'AlexNet en remportant le défi de ImageNet Large Scale Visual Recognition Challenge (ILSVRC), dans lequel les logiciels s'affrontent pour classer et détecter correctement des objets et des scènes (Krizhevsky et al. 2012). Des innovations importantes dans CNN ont été proposées depuis lors et sont largement attribuées à la restructuration des unités de traitement et à la conception de nouveaux blocs. Zeiler et Fergus

(Zeiler et Fergus 2013) ont donné le concept de visualisation par couche de CNN pour améliorer la compréhension des étapes d'extraction de caractéristiques, ce qui a déplacé la tendance vers l'extraction de caractéristiques à faible résolution spatiale dans une architecture profonde telle qu'elle est effectuée dans VGG (Simonyan et Zissermann 2015). De nos jours, la plupart des nouvelles architectures sont construites sur le principe de la topologie simple et homogène, tel qu'introduit dans VGG. Le groupe d'apprentissage en profondeur de Google a introduit une idée innovante de division, de transformation et de fusion, avec le bloc correspondant appelé bloc de démarrage. Le bloc de démarrage a donné pour la toute première fois le concept de ramification au sein d'une couche, ce qui permet l'abstraction d'entités à différentes échelles spatiales (Szegedy et al. 2015). Dans cette section, nous présenterons l'évolution des architectures convolutives commençant avec l'architecture LeNet.

2.4.2.1. LeNet

L'architecture LeNet a été initialement proposée en 1995 par Yann LeCun et ses collaborateurs (LeCun et al., 1995). Elle est célèbre en raison de son importance historique car elle s'agissait du premier CNN, qui a montré des performances de pointe sur les tâches de reconnaissance des chiffres de la main. Elle a la capacité de classer les chiffres sans être affecté par de petites distorsions, rotations et variations de position et d'échelle. L'architecture LeNet est constituée de cinq couches alternées de convolution et de Pooling, suivies de deux couches Fully-Connected (Khan et al., 2020). La stratégie d'alternance de couches convolutives et de couches de sous-échantillonnage a été initialement proposée par Fukushima en 1975, et est adaptée au problème de reconnaissance de chiffres manuscrits isolés (Sueiras, 2021). Au début des années 2000, le GPU n'était pas couramment utilisé pour accélérer l'entraînement, et même les processeurs étaient lents. La principale limitation du NN multicouche traditionnel entièrement connecté était qu'il considère chaque pixel comme une entrée séparée et lui applique une transformation, ce qui représentait une charge de calcul importante, en particulier à l'époque. LeNet a exploité la base sous-jacente de l'image selon laquelle les pixels voisins sont corrélés les uns aux autres et les motifs caractéristiques sont répartis sur toute l'image. Par conséquent, la convolution avec des paramètres apprenables est un moyen efficace d'extraire des caractéristiques similaires à plusieurs endroits avec peu de paramètres. L'apprentissage avec des paramètres partageables a changé la vision conventionnelle de la formation où chaque pixel était considéré comme une caractéristique d'entrée distincte de son voisinage et ignorait la corrélation entre eux (Khan et al., 2020). LeNet était la première architecture CNN, qui non seulement réduisait le nombre de paramètres, mais était capable d'apprendre automatiquement des fonctionnalités à partir de pixels bruts.

2.4.2.2. AlexNet

L'architecture LeNet a commencé l'histoire des CNN profonds, mais à cette époque, CNN était limité aux tâches de reconnaissance manuelle des chiffres et ne fonctionnait pas bien pour toutes les classes d'images. AlexNet est considéré comme la première architecture CNN profonde, qui a montré des résultats révolutionnaires pour les tâches de classification et de reconnaissance d'images. AlexNet a été proposé par Krizhevsky et al., qui ont amélioré la

capacité d'apprentissage du CNN en l'approfondissant et en appliquant plusieurs stratégies d'optimisation des paramètres (Krizhevsky et al., 2012). Dans AlexNet, la profondeur a été étendue de 5 (LeNet) à 8 couches pour rendre CNN applicable à diverses catégories d'images. Malgré le fait que généralement, la profondeur améliore la généralisation pour différentes résolutions d'images mais, le principal inconvénient associé à une augmentation de la profondeur est le sur-apprentissage. Pour relever ce défi, Krizhevsky et al ont exploité l'idée de Hinton, selon laquelle leur algorithme saute au hasard certaines unités de transformation pendant la formation pour forcer le modèle à apprendre des fonctionnalités plus robustes. En plus de cela, ReLU a été utilisé comme fonction d'activation non saturante pour améliorer le taux de convergence en atténuant dans une certaine mesure le problème du gradient de disparition. Un sous-échantillonnage qui se chevauche et une normalisation de la réponse locale ont également été appliqués pour améliorer la généralisation en réduisant le sur-apprentissage. D'autres ajustements ont été l'utilisation de filtres de grande taille (11×11 et 5×5) aux couches initiales, par rapport aux réseaux précédemment proposés (Khan et al., 2020). En raison de l'approche d'apprentissage efficace d'AlexNet, il a une importance significative dans la nouvelle génération de CNN et a lancé une nouvelle ère de recherche dans les avancées architecturales des CNN.

2.4.2.3. ZfNet

Le mécanisme d'apprentissage de CNN, avant 2013, reposait principalement sur un jeu de hasard, sans connaître la raison exacte de l'amélioration. Ce manque de compréhension a limité les performances des CNN profonds sur des images complexes. En 2013, Zeiler et Fergus ont proposé un NN déconvolutif multicouche intéressant (DeconvNet), qui est devenu célèbre sous le nom de ZfNet (Zeiler et Fergus 2013). ZfNet a été développé pour visualiser quantitativement les performances du réseau. L'idée de visualisation des caractéristiques proposée par ZfNet a été validée expérimentalement sur AlexNet à l'aide de DeconvNet, qui a montré que seuls quelques neurones étaient actifs. En revanche, d'autres neurones étaient morts (inactifs) dans les première et deuxième couches du réseau. Sur la base de ces résultats, Zeiler et Fergus ont ajusté la topologie CNN et effectué une optimisation des paramètres (Khan et al., 2020). Zeiler et Fergus ont maximisé l'apprentissage de CNN en réduisant à la fois la taille du filtre et la foulée pour conserver le nombre maximum de caractéristiques dans les deux premières couches convolutionnelles. Ce réajustement de la topologie CNN a entraîné une amélioration des performances, ce qui suggère que la visualisation des caractéristiques peut être utilisée pour l'identification des lacunes de conception et pour l'ajustement rapide des paramètres.

2.4.2.4. VGG

L'utilisation réussie des CNN dans les tâches de reconnaissance d'images a accéléré la recherche en conception architecturale. À cet égard, Simonyan et Zisserman ont proposé un principe de conception simple et efficace pour les architectures CNN. Leur architecture, nommée VGG, était modulaire en couches (Simonyan et Zisserman, 2015). VGG a été créé à 19 couches de profondeur par rapport à AlexNet (Krizhevsky et al., 2012) et ZfNet (Zeiler et Fergus, 2013) pour simuler la relation de profondeur avec la capacité de représentation du

réseau. ZfNet, qui était un réseau de première ligne du concours 2013-ILSVRC a suggéré que les filtres de petite taille peuvent améliorer les performances des CNN. Sur la base de ces résultats, VGG a remplacé les filtres (11×11) et (5×5) par une pile de couches de filtres (3×3) et a démontré expérimentalement que le placement simultané de filtres de petite taille (3×3) pouvait induire l'effet du filtre de grande taille (5×5 et 7×7). L'utilisation de filtres de petite taille offre un avantage supplémentaire de faible complexité de calcul en réduisant le nombre de paramètres. Ces résultats ont défini une nouvelle tendance dans la recherche pour travailler avec des filtres de plus petite taille dans CNN. VGG a montré de bons résultats à la fois pour les problèmes de classification et de localisation des images. VGG était à la 2e place du concours 2014-ILSVRC mais s'est fait connaître en raison de sa simplicité, de sa topologie homogène et de sa profondeur accrue.

2.4.2.5. GoogleNet

GoogleNet a remporté le concours 2014-ILSVRC et est également connu sous le nom d'Inception-V1. L'objectif principal de l'architecture GoogleNet était d'atteindre une grande précision avec un coût de calcul réduit (Szegedy et al., 2015). Il a introduit le nouveau concept de bloc de démarrage dans CNN, dans lequel il intègre des transformations convolutives à plusieurs échelles en utilisant l'idée de division, de transformation et de fusion. Dans GoogleNet, les couches conventionnelles sont remplacées dans de petits blocs similaires à l'idée de remplacer chaque couche par des micros NN comme proposé dans l'architecture Network in Network (NIN) (Lin et al. 2013). Ce bloc encapsule des filtres de différentes tailles (1×1 , 3×3 et 5×5) pour capturer des informations spatiales à différentes échelles. L'exploitation de l'idée de diviser, transformer et fusionner par GoogleNet a permis de résoudre un problème lié à l'apprentissage de divers types de variations présentes dans la même catégorie d'images ayant des résolutions différentes. Cependant, le principal inconvénient de GoogleNet était sa topologie hétérogène qui doit être personnalisée d'un module à l'autre. Une autre limitation de GoogleNet était la limitation de représentation qui réduit considérablement l'espace des fonctionnalités dans la couche suivante, et peut donc parfois entraîner une perte d'informations utiles.

Le tableau 2.1 Résume l'état de l'art des architectures convolutives présentées ci-dessus, leurs paramètres et leurs performances sur les ensemble de données ImageNet.

Tableau 2.1 : Comparaison des performances des architectures convolutives (Khan et al., 2020), (OpenGenus, 2019)

Architecture	Auteurs	Caractéristique	Profondeur	Paramètres (Millions)	Taux d'erreur
LeNet 1998	LeCun et al	Première architecture CNN populaire.	5	0.060 M	/
AlexNet 2012	Krizhevsky et al	Plus profond et plus large que le LeNet Utilise Relu, dropout and overlap Pooling.	8	60 M	16.4
ZfNet 2014	Zeiler et Fergus	Visualisation des couches intermédiaires.	8	60 M	11.7
VGG 2014	Simonyan et Zisserman	Topologie homogène Utilise des noyaux de petite taille.	19	138 M	7.3
GoogleNet 2015	Szegedy et al	Introduction de l'idée de diviser, transformer et fusionner.	22	4 M	6.7

Au-delà de 2015, la plupart des nouveaux modèles sont des améliorations par rapport aux modèles révolutionnaires précédents et certains sont un ensemble des modèles précédents. Ces nouveaux modèles fonctionnent mieux que les modèles précédents, mais aucune idée significative ou de changement de voie n'a été formée au cours des dernières années.

2.5. Quelques systèmes existants de reconnaissance de caractères manuscrits basé CNN

Dans le domaine de reconnaissance de l'écriture manuscrite, les réseaux de neurones convolutifs (CNN) ont été initialement utilisés pour la reconnaissance des chiffres de l'écriture manuscrite. Ces dernières années, de nombreux auteurs ont exploré les performances des réseaux de neurones convolutifs (CNN) et d'autres techniques d'apprentissage en profondeur sur l'ensemble de données MNIST. De nos jours, un nombre important de travaux ont atteint un taux d'erreur de test inférieur à 1 % sur ce jeu de données ; qui devient sans défi. À la mi-2017, un nouvel ensemble de données a été introduit : EMNIST, qui implique à la fois des chiffres et des lettres, avec une plus grande quantité de données acquises à partir d'une base de données différente de celle du MNIST. Dans cette section, nous présenterons quelques travaux de reconnaissance de l'écriture manuscrite à base des réseaux de neurones convolutifs (CNN) pour la base de données EMNIST. Bien que le nombre de travaux traitant d'EMNIST soit nettement inférieur à ceux testant sur MNIST, du fait que la base de données a

été introduite en 2017, certains travaux obtenant des résultats très compétitifs commencent à émerger.

Le tableau 2.2 présente un récapitulatif précisant les caractéristiques et les performances de certain systèmes de reconnaissance de l'écriture manuscrite à base des réseaux de neurones convolutifs (CNN).

Tableau 2.2 : Résumé des résultats de quelques systèmes de reconnaissance de l'écriture manuscrite basé CNN.

Système	Classificateur	Base de données	Performance (%)
(Singh et al., 2017)	CNN	EMNIST Digits	99.62
(Botalb et al., 2018)	CNN	EMNIST	99.2
(Shawon et al., 2018)	CNN	EMNIST Balanced	90.59
		EMNIST Digits	99.79
(Baldominos et al., 2019)	CNN	EMNIST Letters	95.35
		EMNIST Digits	99.77
(dos Santos et al., 2019)	CNN	EMNIST Digits	99.775
(Jayasundara, et al., 2019)	CNN	EMNIST Letters	95.36
		EMNIST Digits	99.79
(Mor et al., 2019)	CNN	EMNIST	87.1
(Erdoğan et Tümer, 2021)	CNN	EMNIST	87.81
(Mustafa et al., 2021)	CNN	EMNIST Balanced	99
		EMNIST Letters	95
		EMNIST MNIST	89

(Pashine et al., 2021)	CNN	MNIST	99.31
---------------------------	-----	-------	-------

2.6. Conclusion

L'objectif de ce chapitre est de donner un aperçu général sur l'approche de reconnaissance adoptée dans ce travail. Nous avons présenté dans un premier temps les différentes méthodes et techniques de l'apprentissage automatique et l'apprentissage profond. Par la suite, nous avons introduit quelques propriétés sur les réseaux de neurones et les architectures classiques rencontrées dans la littérature. L'accent est surtout mis sur les réseaux de neurones convolutifs (CNN) qui sont l'approche adoptée pour la réalisation de notre système de reconnaissance de caractères manuscrit hors ligne. En fin, nous avons décrit quelques travaux récents dans le domaine de la reconnaissance automatique de l'écriture manuscrite à base des réseaux de neurones convolutifs (CNN).

Dans le chapitre suivant, nous présenterons la structure générale de notre système de reconnaissance des caractères hors ligne, ainsi que le protocole expérimental suivi et les résultats obtenus.

UN SYSTEME DE RECONNAISSANCE DES CARACTERES MANUSCRITS : CONCEPTION, EXPERIMENTATION ET RESULTATS

Sommaire

3.1.	Introduction	48
3.2.	Motivation et objectif	48
3.3.	Architecture et description du système de reconnaissance.....	49
3.3.1.	Présentation détaillée du système.....	49
3.4.	Expérimentations et résultats	53
3.4.1.	Bases de données utilisées.....	53
3.4.2.	Détermination de l'architecture du modèle CNN	57
3.4.3.	Evaluation des performances	62
3.4.4.	Comparaison avec les algorithmes d'apprentissage automatique	68
3.4.5.	Comparaison avec des travaux de la littérature.....	71
3.5.	Conclusion	76

3.1. Introduction

La reconnaissance de caractères manuscrits isolés est au cœur de la chaîne de reconnaissance de l'écriture manuscrite. Elle a fait l'objet de recherches intensives au cours des dernières décennies, en raison de son utilité dans un large éventail de problèmes du monde réel. En conséquence, de nombreux systèmes ont été réalisés dans ce domaine. En effet, la majorité de ces systèmes adoptent des approches basées sur des modélisations statistiques telles que les machines à vecteurs de support, les forêts aléatoires, les modèles de Markov cachés, etc.

Ces dernières années, les techniques basées sur l'apprentissage profond ont suscité un intérêt considérable dans la communauté des chercheurs pour résoudre une variété de problèmes de classification. L'une des techniques les plus connues et les plus largement utilisées sont les réseaux de neurones convolutifs (en anglais : Convolution Neural Networks CNN), une sorte de réseaux de neurones capables d'extraire automatiquement les caractéristiques pertinentes des données d'entrée (Baldominos et al., 2019). Les propriétés des réseaux de neurones convolutifs, notamment le fait qu'ils sont capables de récupérer des caractéristiques à partir d'entrées multidimensionnelles, en font une alternative très intéressante pour résoudre des problèmes de classification d'images en général, et de la reconnaissance de l'écriture en particulier.

Ce chapitre est consacré à la mise en place d'un système de reconnaissance de caractères manuscrits isolés hors ligne. Dans la première partie de ce chapitre, nous présentons, les motivations et les objectifs que nous nous sommes fixés. Ensuite, nous abordons les différentes étapes pour la mise en œuvre de notre modèle réalisant la classification des caractères manuscrits. Dans la troisième section, nous présentons les expérimentations et l'évaluation des différents modèles construits.

3.2. Motivation et objectif

L'objectif principal de notre travail est de construire un modèle adapté pour la classification des caractères manuscrits ayant une faible marge d'erreur. Pour ce faire, nous allons opter pour un modèle basé sur l'apprentissage profond, plus précisément les CNN du fait qu'un CNN est une architecture spécifiquement destinée au traitement de données ayant une topologie en forme de grille. Mais, prenons en considération le fait que EMNIST, base de données que nous avons utilisée, est une base récente qui représente jusqu'à présent une tâche de classification difficile, même pour les réseaux de neurones (Cohen et al., 2017). A ce fait, nous devons tout d'abord effectuer une série d'expérimentations afin de déterminer les composants et les paramètres du modèle CNN qui fournissent les meilleurs résultats en termes de capacité de reconnaissance. Enfin, nous comparons les résultats obtenus avec certains algorithmes d'apprentissage automatique et quelques travaux récents de la littérature afin d'évaluer notre système.

3.3. Architecture et description du système de reconnaissance

L'architecture d'un système de reconnaissance de caractères manuscrits isolés à partir d'une image suit quatre étapes essentielles : Acquisition d'images, prétraitement, extraction de caractéristiques et enfin classification des caractères. Le schéma bloc illustré dans la figure 3.1 montre l'architecture du processus global de reconnaissance de caractères.

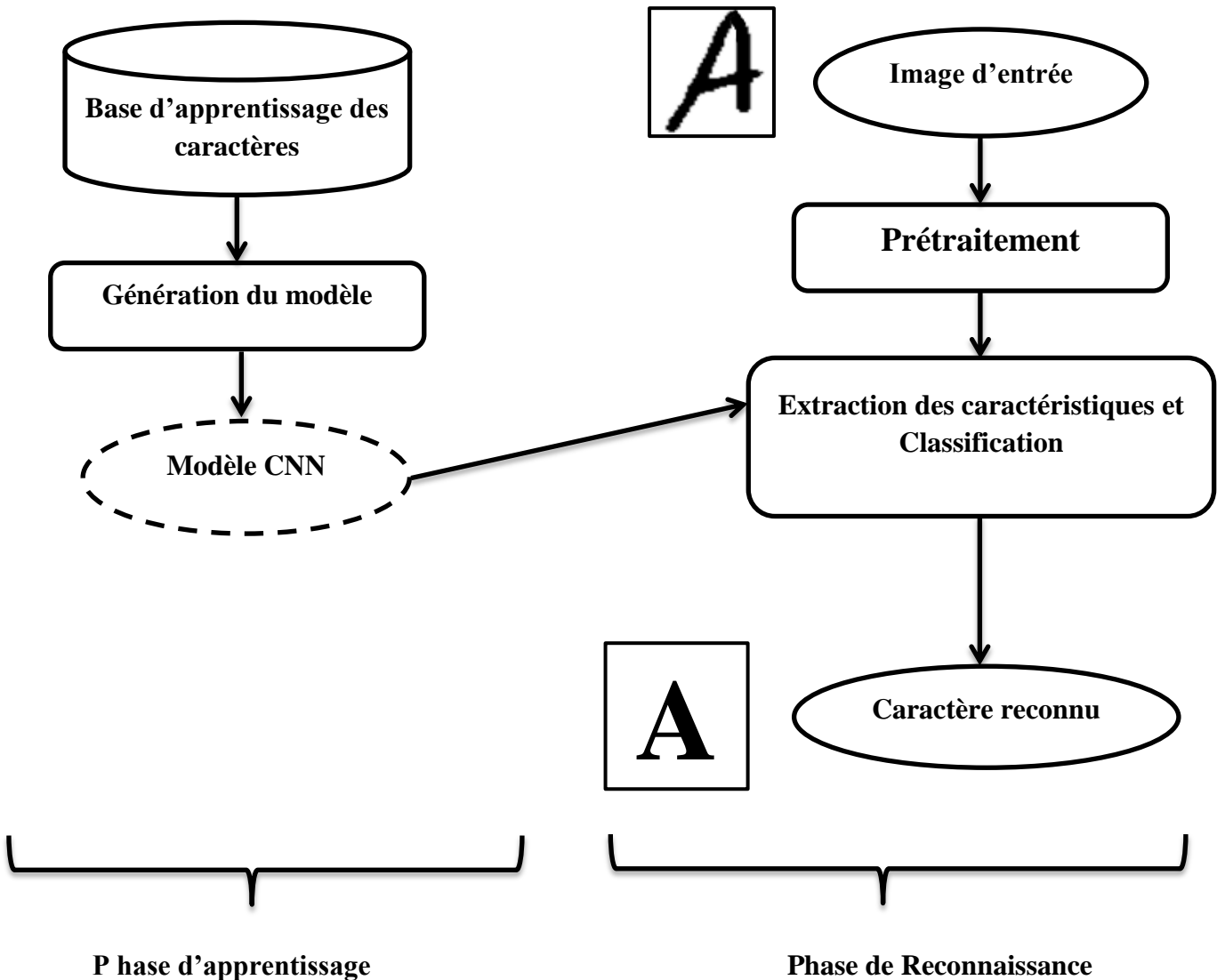


Figure 3.1 : Architecture globale de notre système

Après avoir présentée l'architecture globale de notre système de reconnaissance des caractères manuscrits. Nous abordons en détail, dans la sous-section suivante, les différentes étapes nécessaires pour la mise en place de ce système.

3.3.1. Présentation détaillée du système

Dans cette section, nous décrivons en détail les différentes phases constituant notre système.

3.3.1.1. Phase d'apprentissage

L'apprentissage consiste à générer le modèle CNN, qui sera par la suite utilisé dans la phase de reconnaissance.

a. Préparation de données

Afin de réaliser notre modèle, nous devons disposer de bases de données représentant les objets à étudier. Nous avons utilisé la base de données EMNIST (Extended Modified National Institute of Standards and Technology), qui est un ensemble de caractères manuscrits isolés dérivés de la base de données spéciale NIST 19 et convertis en un format d'image 28 x 28 pixels et une structure d'ensemble de données qui correspond directement à la célèbre ensemble de données MNIST. L'ensemble de données EMNIST est divisé en six sous-ensembles de données. Dans le cadre de cette étude seulement 3 sous-ensembles de données sont utilisées à savoir : EMNIST Letters, EMNIST Balanced et EMNIST MNIST. Ainsi, trois modèles aux caractéristiques différentes seront construits pour être mis en œuvre selon le type des entrées.

La base de données EMNIST, ainsi que les sous-ensembles utilisés seront présentés en détail plus loin dans la section 3.4.1.

b. Génération du modèle

Le modèle de reconnaissance de l'écriture manuscrite construit dans cette étude utilise CNN. L'architecture CNN prend en charge les processus d'extraction et de classification des caractéristiques : CNN extrait des cartes de caractéristiques à partir d'images bidimensionnelles, puis les utilise dans le processus de classification.

Comme nous l'avons mentionné précédemment dans le chapitre 2, les architectures CNN ont les mêmes principes de conception généraux, appliquant une couche de convolution aux données d'entrée, effectuant un sous-échantillonnage (Max Pooling) sur les dimensions spatiales tout en augmentant régulièrement le nombre de cartes des caractéristiques. Il existe également des couches entièrement connectées (Fully Connected), des fonctions d'activation et des fonctions de Dropout. Cependant, de toutes les opérations du CNN, le cœur du CNN est constitué de couches de Convolution, de couches de Pooling et de couches Fully Connected. Mais des couches supplémentaires peuvent également être introduites qui auront un effet, par exemple une couche Softmax qui améliore la précision de la détection d'image (Rosyda et Purboyo, 2018). La figure 3.2 illustre un exemple d'une architecture CNN. Le nombre de couches des CNN n'est pas toujours le même. Ce dernier est défini en fonction de l'application, de ses besoins et même de la langue d'écriture considérée.

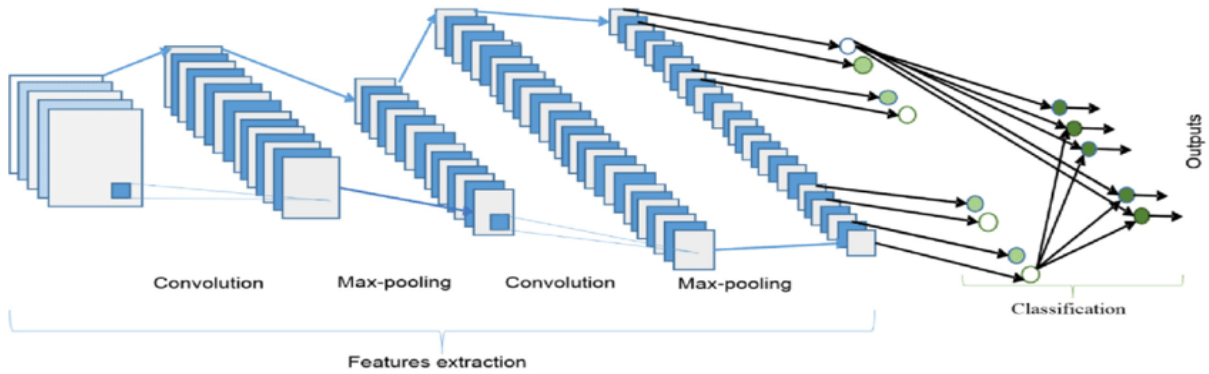


Figure 3.2 : Architecture CNN à deux couches de convolution (Alom et al., 2019)

Bien qu'ils existent de nombreuses architectures de modèles CNN pré-entraînés (voir section 2.4.2), dans notre système, nous avons conçu notre propre architecture (Figure 3.3). Cette architecture est optimisée via plusieurs tests expérimentaux en modifiant à chaque fois les valeurs des différents hyper-paramètres à savoir, le nombre de paires de couches Convolution–Max Pooling, la taille des cartes des caractéristiques, des couches denses et enfin les Dropout.

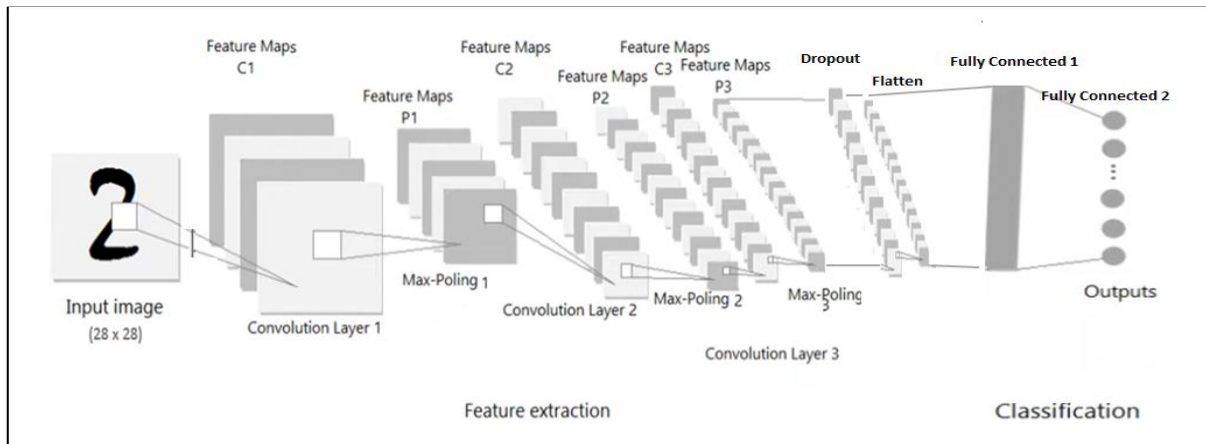


Figure 3.3 : L'architecture de notre modèle CNN (Modèle Lettre)

Notre modèle CNN, est construit comme suit :

- **Entrée** : Les données d'entrée du CNN sont représentées sous la forme d'un vecteur 4D (16, 28, 28, 1), où 16 représente la taille du batch, (28 x 28 x 1) le format de l'image, (hauteur, largeur et profondeur), dont la profondeur de l'image représente le nombre de canaux de couleur (image en niveaux de gris).
- **Couche de Convolution** : Cette couche applique un filtre a une entrée pour créer une carte de caractéristiques, indiquant les emplacements et le poids d'une caractéristique détectée dans une entrée.

- **Couche de Pooling** : Cette couche est utilisée pour réduire la dimension spatiale des cartes de caractéristiques d'image et le nombre des caractéristiques tout en préservant les informations discriminantes, ce qui à son tour entraîne une réduction de la complexité de calcul globale ainsi que, le risque de sur-apprentissage.
- **Dropout** : Le Dropout est une technique de régularisation dans laquelle certains neurones seront sélectionnés au hasard et ne seront pas utilisés pendant l'apprentissage, afin d'éviter le sur-apprentissage et accélérer le processus d'apprentissage.
- **Couche Flatten** : Cette couche transforme les données en un tableau unidimensionnel pour être directement exploitable par la couche Fully Connected.
- **Couche Fully Connected avec la fonction d'activation 'ReLU'** : Cette couche permet de réduire le vecteur en entrées à un vecteur de taille 256. Elle remplace toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation.
- **Couche Fully Connected avec une activation 'softmax'** : C'est la couche de classification. Elle réduit les entrées en un vecteur de taille égale aux nombre de classes à prédire (Par exemple : Un vecteur de 26 éléments représente chacun une classe de lettres à prédire (A, B,..., Z)). Chaque élément correspond donc à la probabilité pour l'image en entrée d'appartenir à une classe de caractère.

3.3.1.2. Phase de reconnaissance

Le processus de reconnaissance des caractères manuscrits, comme montré précédemment dans la figure 3.1, passe par les étapes suivantes : Acquisition, prétraitement, extraction des caractéristiques et classification.

a. Acquisition

Dans cette étape, le system reçoit une image en entrée. Cette image sera utilisée pour alimenter le modèle de reconnaissance de l'écriture manuscrite qui a été construit lors de la phase d'apprentissage.

b. Prétraitement

Le prétraitement est la deuxième étape d'un système de reconnaissance qui consiste en plusieurs opérations effectuées pour améliorer la qualité de l'image en entrée. Avant de l'utiliser, nous devons d'abord effectuer un prétraitement afin qu'elle puisse être exploitée par le modèle CNN. Dans notre cas, nous n'avons utilisé que l'opération de binarisation. Cela est du fait que les CNN nécessitent peu de prétraitement en raison que ces derniers sont responsables de faire évoluer tous seuls leurs propres filtres.

La binarisation est le processus de conversion d'une image couleur (souvent RGB) ou en niveaux de gris en une image à deux niveaux (noir et blanc), c'est-à-dire la conversion d'un spectre 0-255 en un spectre 0-1. La méthode de seuillage adaptatif est la méthode que nous avons utilisée dans ce processus de binarisation. Cette méthode calcule la valeur seuil pour une petite partie de l'image afin qu'il soit possible d'obtenir des valeurs différentes pour chaque partie de la même image.

c. Extraction des caractéristiques et classification

La reconnaissance de formes ou d'objets se fait généralement par extraction et classification de caractéristiques. Le processus d'extraction de caractéristiques utilise généralement diverses méthodes pour obtenir une représentation des données. Après cela, le classifieur est utilisé pour classifier les données. Ces processus sont effectués manuellement et séparément (Mustafa et al., 2021). Dans le cadre de ce travail, l'approche utilisée est basée sur l'apprentissage profond, c'est-à-dire, l'extraction et la classification des caractéristiques sont intégrées automatiquement dans une seule étape.

L'idée principale est de déterminer l'appartenance d'un caractère non connu présenté dans une image en entrée à une classe prédéfinie à partir des vecteurs de caractéristiques extraits de cette image. Les couches de Convolution et Pooling du CNN permettent l'extraction des caractéristiques de l'image dont on souhaite reconnaître le caractère. Alors que les couches Fully Connected sont utilisées pour la classification. Elles prennent en entrée les sorties des couches précédentes, et applique des pondérations pour prédire l'étiquette correcte.

3.4. Expérimentations et résultats

Cette section décrit l'architecture du modèle CNN pour la classification des caractères manuscrits et les résultats expérimentaux. La première sous-section présente l'ensemble de données EMNIST, qui est utilisé pour l'apprentissage et l'évaluation du CNN. Dans la deuxième section, des expérimentations sont réalisées pour déterminer la combinaison optimale d'hyperparamètres. Enfin, les résultats expérimentaux obtenus sont comparés aux résultats d'autres algorithmes d'apprentissage automatique ainsi qu'aux ceux de l'état de l'art du problème de la reconnaissance de caractères manuscrits.

3.4.1. Bases de données utilisées

L'ensemble de données EMNIST est un ensemble de caractères manuscrits dérivés de la base de données spéciale NIST 19 et convertis en un format d'image 28 x 28 pixels et une structure d'ensemble de données qui correspond directement à l'ensemble de données MNIST. La figure 3.4 illustre un échantillon des lettres et des chiffres de l'ensemble de données EMNIST.



Figure 3.4 : Échantillons de toutes les lettres et chiffres de l'ensemble de données EMNIST (Cohen et al., 2017)

La source pour la construction de la base de données MNIST était la base de données spéciale 19 du NIST (NIST SD 19), contenant l'ensemble du corpus NIST (National Institute of Standards and Technology of the US) de matériel de formation pour la reconnaissance des documents et des caractères manuscrits, y compris plus de 800 000 exemples vérifiés manuellement. Et des caractères étiquetés de près de 3700 scripteurs qui ont rempli un formulaire. Même si cette base de données était disponible à partir du milieu des années 1990, elle est restée majoritairement inutilisée en raison de la difficulté d'y accéder et de l'utiliser dans les ordinateurs modernes, en raison de la manière dont elle était stockée. Cela a été corrigé en 2016 lorsqu'une deuxième version de la base de données a été publiée avec un format plus simple. Les auteurs d'EMNIST ont intentionnellement rendu la base de données conforme à MNIST en termes de structure, effectuant donc un traitement similaire, qui implique les étapes suivantes :

- a) Stockage des images originales dans NIST SD 19 sous forme d'images en noir et blanc de 128×128 pixels.
- b) Application d'un filtre de flou gaussien pour adoucir les bords.
- c) Suppression du rembourrage vide pour réduire l'image à la région d'intérêt.
- d) Centrage du caractère dans une case carrée en préservant le rapport d'aspect.
- e) Ajout d'un rembourrage vierge de 2 pixels par côté et sous-échantillonnage de l'image 28×28 pixels à l'aide d'une interpolation bi-cubique. La plage de valeurs d'intensité est ensuite mise à l'échelle $[0, 255]$, ce qui donne des images en niveaux de gris de 28×28 pixels.

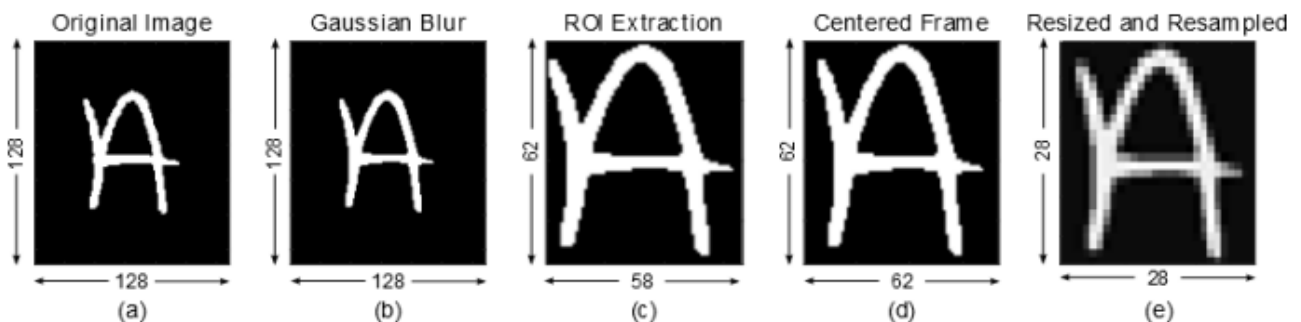


Figure 3.5 : Processus de conversion utilisé pour convertir l'ensemble de données NIST (Cohen et al., 2017).

L'ensemble de données EMNIST est divisé en six sous-ensembles de données : EMNIST Balanced, EMNIST by Merge, EMNIST by Class, EMNIST Letters, EMNIST Digits et EMNIST MNIST. Le tableau 3.1 contient un résumé des sous-ensembles de données EMNIST et indique la répartition des ensembles d'apprentissage et de test d'origine spécifiés dans les sous-ensembles de la base de données EMNIST.

Tableau 3.1 : Résumé des sous-ensembles de données EMNIST (Cohen et al., 2017)

Sous-ensemble	Nombre de Classes	Données d'apprentissage	Données de test	Total
By_Class	62	697 932	116 323	814 255
By_Merge	47	697 932	116 323	814 255
Balanced	47	112 800	18 800	131 600
Digits	10	240 000	40 000	280 000
Letters	26	124 800	28 000	145 600
MNIST	10	60 000	10 000	70 000

Dans le cadre de ce travail, seulement trois sous-ensembles sont utilisés à savoir : EMNIST Letters, EMNIST Balanced et EMNIST MNIST.

3.4.1.1. EMNIST Lettres (Cohen et al., 2017)

L'ensemble de données EMNIST Letters combine les versions majuscules et minuscules de chaque lettre en une seule classe. Cet ensemble de données fournit donc une tâche de classification différente des autres ensembles de données. Comme certaines lettres ont des représentations majuscules et minuscules nettement différentes, les classifieurs sont tenus d'associer deux représentations différentes d'une lettre à une seule étiquette de classe.

Les classes de l'ensemble de données EMNIST Letters sont des lettres minuscules et des lettres majuscules combinées en une classe. EMNIST Letters contient 26 classes correspondant aux 26 lettres (de A à Z). Le nombre d'échantillons de chaque classe dans les ensembles de donnée sont équivalents. La figure 3.6 montre la distribution des échantillons dans chaque classe.

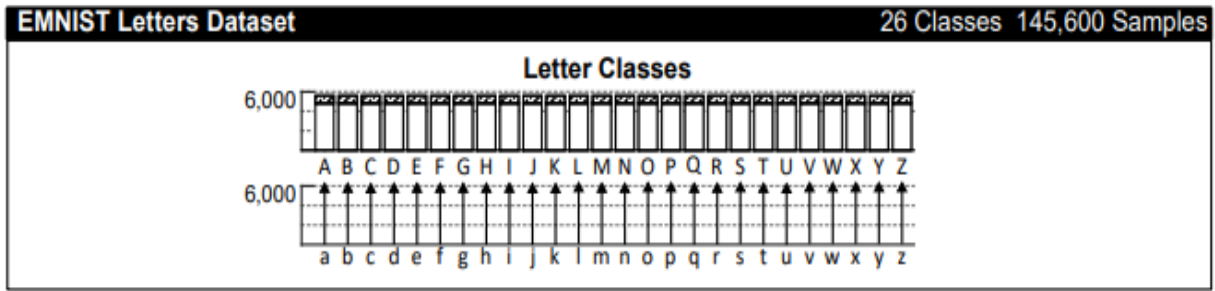


Figure 3.6 : La distribution des échantillons dans chaque classe pour l'ensemble de données EMNIST Lettres (Cohen et al., 2017).

3.4.1.2. EMNIST Blanced (Cohen et al., 2017)

L'ensemble de données EMNIST Balanced est destiné à fournir la tâche de classification la plus juste et la plus cohérente dérivée de la base de données spéciale 19 du NIST. Il contient un sous-ensemble de l'ensemble de données By_Merge contenant un nombre égal d'échantillons pour chaque classe. L'intention de cet ensemble de données est de fournir une tâche de classification juste, équilibrée et suffisamment difficile.

Les classes de l'ensemble de données EMNIST Balanced sont des chiffres, des lettres minuscules et des lettres majuscules. EMNIST Balanced contient 47 classes au lieu de 62 classes, en raison que cet ensemble tient compte du fait que certaines lettres sont assez similaires dans leurs variantes minuscules et majuscules, ainsi les deux classes peuvent être fusionnées. En particulier, ces lettres sont 'c', 'i', 'j', 'k', 'l', 'm', 'o', 'p', 's', 'u', 'v', 'w', 'x', 'y' et 'z'. La figure 3.7 montre les classes de l'ensemble de données EMNIST Balanced.

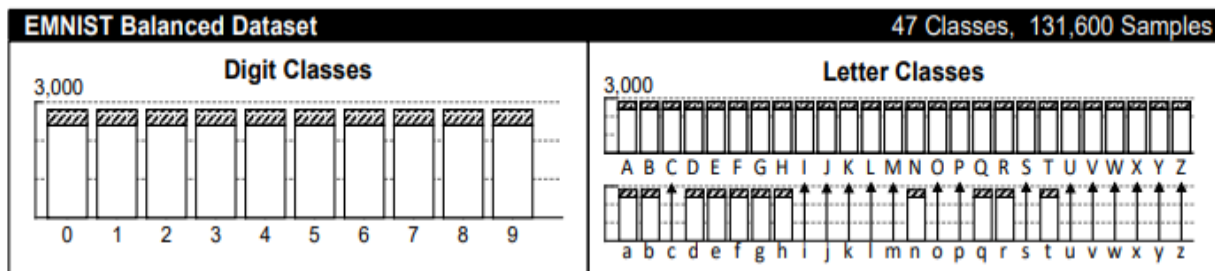


Figure 3.7 : La distribution des échantillons dans chaque classe pour l'ensemble de données EMNIST Balanced (Cohen et al., 2017).

3.4.1.3. EMNIST MNIST (Cohen et al., 2017)

L'ensemble de données EMNIST MNIST est destiné à correspondre exactement à la taille et aux spécifications de l'ensemble de données MNIST d'origine. Il est destiné à remplacer directement l'ensemble de données MNIST original contenant des chiffres créés par le processus de conversion décrit dans la section 3.4.1. Il s'agit d'un ensemble de données de 60 000 échantillons en niveaux de gris 28 x 28 des 10 chiffres, ainsi qu'un ensemble de test de

10 000 échantillons. La figure 3.8 montre les classes de l'ensemble de données EMNIST MNIST.

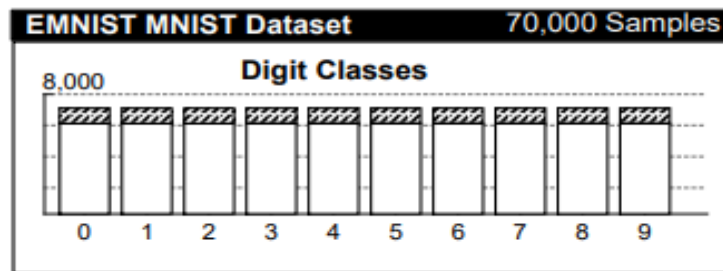


Figure 3.8 : La distribution des échantillons dans chaque classe pour l'ensemble de données EMNIST MNIST (Cohen et al., 2017).

3.4.2. Détermination de l'architecture du modèle CNN

Les réseaux de neurones convolutifs (CNN) sont des réseaux complexes et comprennent de nombreux hyper paramètres. Les hyper paramètres sont essentiellement des paramètres sur les paramètres, décrivant les valeurs des paramètres du modèle. Le bon choix d'hyper paramètres a un effet important sur le résultat d'un réseau (Nazir et al., 2018). On considère, généralement, qu'il est difficile de concevoir un modèle optimal du premier coup, et qu'il faut des essais et des expérimentations pour trouver les valeurs optimales.

Cette section détaille les résultats expérimentaux obtenus avec l'architecture proposée pour le problème de reconnaissance des caractères manuscrits. Les expérimentations visent à explorer les capacités de l'architecture et à identifier les composants et les hyper paramètres qui fournissent les meilleurs résultats en termes de taux de reconnaissance. La première expérimentation consiste à déterminer la profondeur du modèle via le nombre de paires de Convolution – Max Pooling qui seront utilisées. Avec une dimension d'image de 28 x 28 pixels, il sera seulement testé si une, deux, trois ou quatre paires montrent des résultats optimaux. De plus, il n'est pas possible d'appliquer plus de 4 paires car les dimensions de l'image seront trop petites. Ensuite, une deuxième expérimentation est adoptée pour déterminer la taille des cartes des caractéristiques (Feature Maps), des couches denses et enfin les Dropout. Les expérimentations permettant de déterminer l'architecture du modèle utiliseront EMNIST Letters pour construire le modèle Lettres. Le choix s'est porté sur cet ensemble de données car il représente une tâche de classification complexe et en même temps, nous avons pris en compte la taille et le nombre de classes afin que la durée d'apprentissage du modèle soit relativement plus rapide. Ensuite, l'architecture sélectionnée sera appliquée au modèle chiffre en utilisant EMNIST MNIST et au modèle mixte en utilisant EMNIST Balanced. L'ensemble des expérimentations permettant le choix de ces paramètres sont décrites dans ce qui suit.

3.4.2.1. La profondeur du réseau CNN

La recherche de l'architecture du modèle CNN à utiliser commence par déterminer le nombre de paires de couches Convolution - Max Pooling. Le tableau 3.2 présente les taux de

reconnaissance obtenus sur des modèles en cours d'exécution de 1 paire, 2 paires, 3 paires et 4 paires

Tableau 3.2 : Résultats expérimentaux pour déterminer le nombre de paires de couches Convolution – MaxPooling

Paire Convolution – MaxPooling	Précision sur la base d'apprentissage (%)	Précision sur la base de test (%)
1	92.67	92.49
2	93.58	93.49
3	94.26	94.23
4	94.38	94.29

D'après le tableau 3.2, il semble clairement que le taux de reconnaissance s'augmente plus que la profondeur du réseau CNN augmente, c'est-à-dire lorsque on ajoute des paires de couches convolution - Max Pooling. Ainsi, il est remarquable que le meilleur taux de reconnaissance qui est 94.38 % sur la base d'apprentissage, 94.29 % sur la base de test est obtenu avec 4 paires de couches convolution - Max Pooling. Ce taux de reconnaissance est légèrement meilleur que celui obtenu avec 3 paires (94.26 % sur la base d'apprentissage, 94.23 % sur la base de test). Cependant, afin de réduire les coûts de calcul et de stockage supplémentaires (le temps d'exécution et l'espace mémoire), 3 paires de couches convolution - Max Pooling sont choisies comme premier paramètre architectural.

3.4.2.2. La taille des cartes des caractéristiques

Après avoir déterminé la profondeur du CNN, une deuxième expérimentation a été menée pour déterminer la taille de la carte de caractéristiques (Feature Maps) à inclure dans les 3 paires de couches Convolution - MaxPooling. Les résultats obtenus en termes de précision pour des tailles de cartes différentes sont reportés dans le tableau 3.3.

Tableau 3.3 : Résultats expérimentaux pour déterminer la taille des cartes de caractéristiques

Feature Maps (cartes des caractéristiques)			Précision sur la base d'apprentissage (%)	Précision sur la base de test (%)
1^{ère} Couche	2^{ème} Couche	3^{ème} Couche		
8	16	32	94.06	93.65
16	32	64	94.30	94.21
24	48	96	93.87	93.81
32	64	128	94.33	94.24

D'après le tableau 3.3, il est observable que les résultats obtenus pour toutes les combinaisons des cartes des caractéristiques sont assez similaires. Cependant, le meilleur taux de reconnaissance est obtenu avec une combinaison de 32 cartes de caractéristiques dans la première couche de convolution, 64 dans la deuxième couche de convolution et 128 cartes de caractéristiques dans la troisième couche de convolution, mais n'a pas été sélectionnée en raison de la grande taille qui ne vaut pas le coût de calcul supplémentaire. Par contre, la combinaison de 16 cartes de caractéristiques dans la première couche, 32 dans la deuxième et 64 dans la troisième avec une taille moitié donne une précision assez proche avec 94.30 % contre 94.33 % sur les données d'apprentissage et 94.21 % contre 94.24 % sur les données de test. Donc, sur la base de ces résultats la combinaison de 16 cartes de caractéristiques dans la première couche de convolution, 32 dans la deuxième et 64 dans la troisième couche a été sélectionnée pour être incluse dans les trois paires de couches Convolution – Max Pooling.

3.4.2.3. La taille de la couche dense

La taille de la couche dense (ou bien, les unités) est l'un des paramètres les plus fondamentaux et nécessaires de la couche dense qui définit la taille de la sortie de la couche. Le tableau 3.4 montre les résultats obtenus des expérimentations réalisées pour analyser l'effet de la taille de la couche sur le taux de reconnaissance en utilisant différentes tailles de couche dense.

Tableau 3.4 : Résultats expérimentaux pour différentes tailles de la couche dense

Taille de la couche dense	Précision sur la base d'apprentissage (%)	Précision sur la base de test (%)
0	92.88	92.76
32	93.71	93.41
64	93.74	93.66
128	94.09	93.92
256	94.48	94.28
512	94.34	94.16
1024	93.99	93.80
2048	93.75	93.42

La remarque qui peut être faite à partir du tableau 3.4, est que le taux de reconnaissance augmente avec l'augmentation de la taille de la couche Dense jusqu'à atteindre la meilleure précision (94.48 % sur la base d'apprentissage et 94.28 % sur la base de test) avec 256 unités de couche. À partir d'une taille de couche Dense supérieure à 256 unités, le taux de reconnaissance commence à se diminuer. Face à cette variabilité, nous avons choisi 256 unités de couche dense, ce qui est un bon compromis entre la précision et le coût de calcul.

3.4.2.4. Taux du Dropout

Les réseaux de neurones profonds avec un grand nombre de paramètres sont susceptibles de sur-apprendre (Overfit) rapidement un ensemble de données d'apprentissage. Cela a pour effet que le modèle fonctionnera bien sur l'ensemble de données d'apprentissage et mal sur les nouvelles données, ce qui signifie que le modèle ne peut pas généraliser. Les grands réseaux sont également lents à utiliser, ce qui rend difficile la gestion du sur-apprentissage (Overfitting) en combinant les prédictions de nombreux grands réseaux de neurones différents au moment du test. Le Dropout est une technique pour résoudre ce problème. L'idée clé est de supprimer au hasard des nœuds du réseau de neurones pendant l'apprentissage. Cette technique offre une méthode de régularisation efficace pour réduire

l'Overfitting et améliorer le pouvoir de généralisation dans les réseaux de neurones profonds de toutes sortes.

Dans le cadre de cette étude, l'objectif principal est d'avoir un modèle final qui fonctionne bien à la fois sur les données que nous avons utilisées pour l'entraîner (la base d'apprentissage) et sur les nouvelles données sur lesquelles le modèle sera utilisé pour faire des prédictions (la base de test), autrement dit, nous exigeons que le modèle CNN adopté apprenne à partir des exemples connus et généralise à partir de ces exemples vers de nouveaux exemples inconnus.

La dernière expérimentation consiste à trouver le taux de la régularisation du Dropout à utiliser pour réduire l'overfitting et améliorer la généralisation de notre modèle CNN. Les résultats de cette expérimentation pour différents taux de Dropout sont présentés dans le tableau 3.5.

Tableau 3.5 : Résultats expérimentaux pour déterminer le taux de Dropout

Dropout	Précision sur la base d'apprentissage (%)	Précision sur la base de test (%)
0	93.39	93.15
0.1	93.49	93.41
0.2	94.48	94.28
0.3	94.17	93.96
0.4	93.99	93.82
0.5	93.84	93.80
0.6	93.67	93.51
0.7	93.49	93.35

Comme le montrent les résultats présentés dans le tableau 3.5, il est clairement observable que le Dropout de 0.2 est la meilleure mesure du résultat expérimental. Cette mesure signifie que seulement 20 % de neurones sont éliminés à chacune des époques. Il est également remarquable que l'écart entre la précision sur les données d'apprentissage (données connues) et la précision sur les données de test (nouvelles données) est de 0.2 % (inférieur à 1 %), c'est-à-dire notre modèle CNN possède une capacité de généralisation excellente.

Pour résumer, à travers les expérimentations effectuées, nous avons pu déterminer ce qui suit :

L'architecture du modèle choisie pour être implémentée dans le développement du modèle CNN dans cette étude se présente sous la forme de trois paires de couches convolution – Max Pooling au total pour l'extraction des caractéristiques suivies de deux couches denses entièrement connectées pour la classification. L'image d'entrée est de 28×28 . Il y a 16 filtres convolutifs dans la première couche de convolution, 32 filtres dans la deuxième et 64 filtres dans la troisième. Comme fonction d'activation pour toutes les couches de convolution, l'unité linéaire rectifiée (ReLU) est utilisée. Après chaque couche de convolution, une couche Max Pooling est utilisée. Ici, la taille du Pool de la couche Max Pooling est de 2×2 . Pour l'insérer dans la couche suivante, le Flatten transforme les données en un tableau unidimensionnel. Ici, nous avons utilisé 20 % de Dropout, afin d'éviter le sur-apprentissage, et par conséquent, améliorer le pouvoir de généralisation du modèle. La première couche Fully Connected est connectée à 256 filtres et la dernière couche Fully Connected est connectée à un nombre de filtres égal aux nombres de classes. Dans notre réseau de neurones, Softmax transforme la sortie de la dernière couche en ce qui est essentiellement une distribution de probabilité permettant de sélectionner la classe la plus probable.

Dans ce qui suit, nous passons en revue les résultats obtenus sur les bases de données que nous avons utilisées.

3.4.3. Evaluation des performances

Pour évaluer les performances du modèle proposé, des expérimentations ont été réalisées sur les trois bases de données EMNIST décrites précédemment. Nous avons opté pour la distribution initiale des données pour l'ensemble d'apprentissage et l'ensemble de test. Nous avons également utilisé des techniques de validation croisée pour l'évaluation de la qualité de l'apprentissage. La validation croisée est une méthode d'estimation de la fiabilité des résultats, fondée sur une technique d'échantillonnage. Dans un système de classification, cette technique permet de mesurer les taux d'erreur du système en utilisant toutes les données disponibles, à la fois en apprentissage et en validation. Pour effectuer la validation croisée pendant l'apprentissage, nous avons utilisé 15% de l'ensemble de données d'apprentissage.

Parmi les différentes métriques d'évaluation de performances qui existent, nous avons opté pour le taux de reconnaissance et la matrice de confusion afin d'évaluer notre système. Cette dernière est la base de plusieurs évaluations statistiques des performances, elle nous renseigne sur la distribution de l'erreur de classification interclasses. Elle se présente sous forme d'un tableau qui met en relation le nombre d'échantillons bien classés et mal classés.

• Résultats sur la base de données EMNIST Lettres

Le Tableau 3.6 présente les résultats du modèle proposé sur la base de données EMNIST Lettres.

Tableau 3.6 : Résultats de reconnaissance sur la base EMNIST Lettres

EMNIST Letters	Apprentissage	Test
Taille de la base	124 800	28 000
Précision (%)	94.48	94.28

La Figure 3.9 montre les résultats de notre modèle sur la base de test. La matrice de confusion illustre les classes prédites par rapport aux vraies classes.

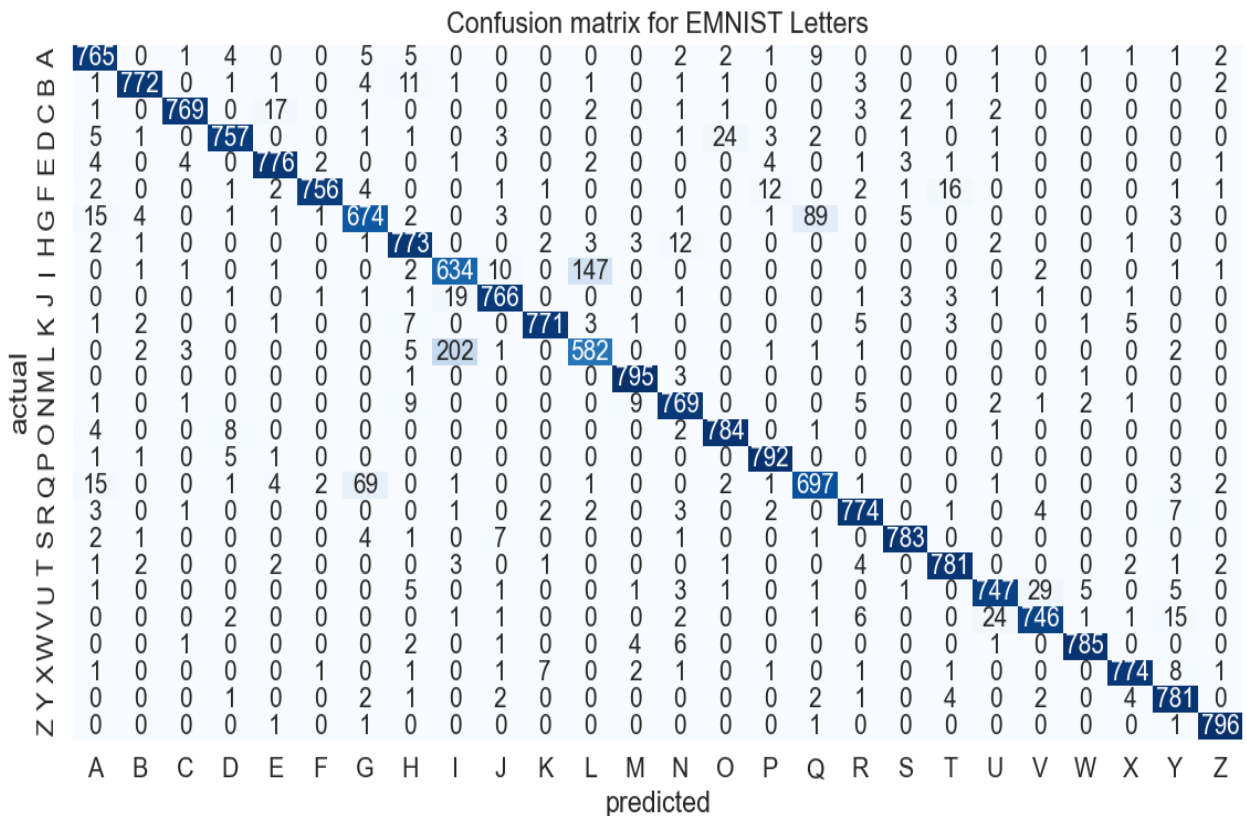


Figure 3.9 : Matrice de confusion pour la base EMNIST Letters

Le taux de reconnaissance du modèle CNN est de 94.28 %, ce qui peut être qualifié de bon. Cependant, les erreurs de reconnaissance de caractères de lettre sont constatées lorsque la mise en œuvre du modèle de lettre est effectuée sur des caractères de lettre qui ont une grande

similarité morphologique. Comme le montre la Figure 3.9, l'erreur la plus fréquente est la confusion des lettres 'I' et 'L', 'G' et 'Q'.

- **Résultats sur la base de données EMNIST MNIST**

Le Tableau 3.7 présente les résultats du modèle proposé sur la base de données EMNIST MNIST.

Tableau 3.7 : Résultats de reconnaissance sur la base EMNIST MNIST.

EMNIST MNIST	Apprentissage	Test
Taille de la base	60 000	10 000
Précision (%)	99.43	99.32

La Figure 3.10 montre les résultats du modèle proposé sur la base de test. La matrice de confusion illustre les classes prédites par rapport aux vraies classes.

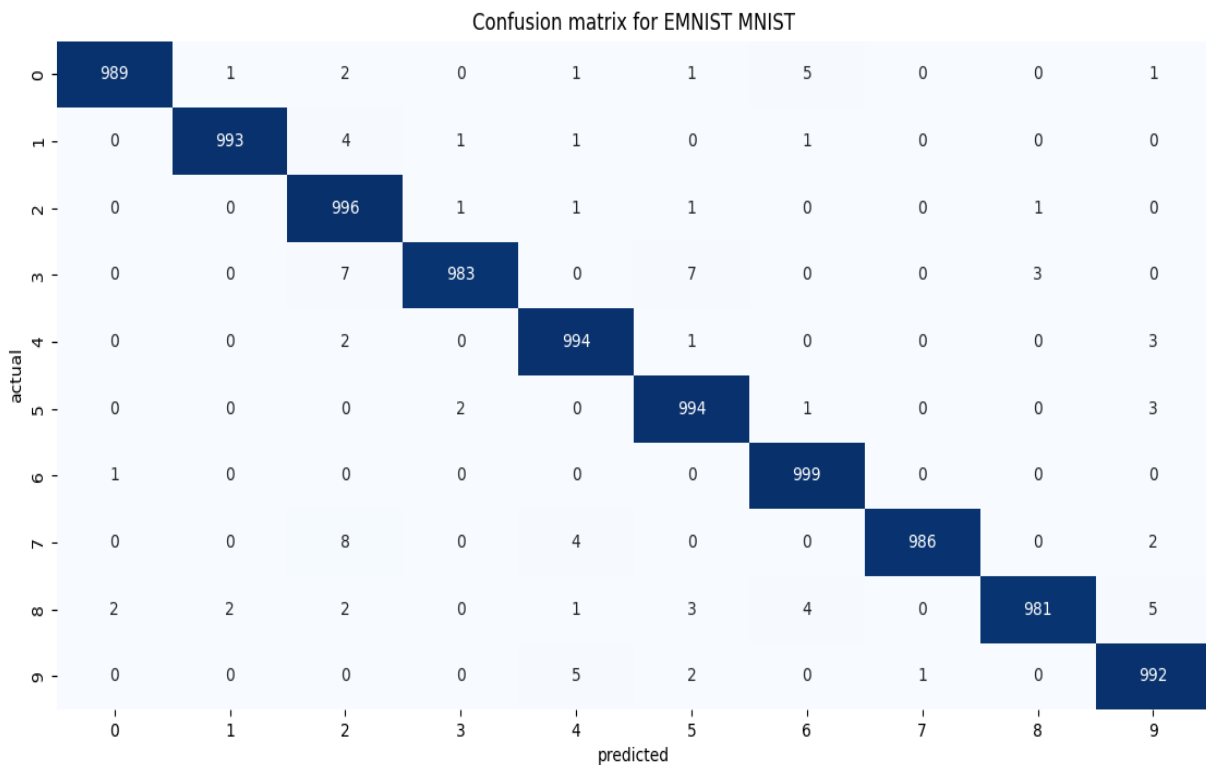


Figure 3.10 : Matrice de confusion pour la base EMNIST MNIST

Le taux de reconnaissance du modèle CNN proposé est de 99.32%, ce qui est un résultat prometteur. C'est-à-dire, presque tous les chiffres ont été correctement attribués à la bonne

classe et cela indique que le système a traité et reconnu la plupart des chiffres de manière correcte.

- **Résultats sur la base de données EMNIST Balanced**

Le Tableau 3.8 présente les résultats du modèle proposé sur la base de données EMNIST Balanced.

Tableau 3.8 : Résultats de reconnaissance sur la base EMNIST Balanced

EMNIST Balanced	Apprentissage	Test
Taille de la base	112 800	18 800
Précision (%)	89.12	88.93

La Figure 3.11 montre les résultats du modèle proposé sur la base de test. La matrice de confusion illustre les classes prédites par rapport aux vraies classes.

Le taux de reconnaissance du modèle CNN proposé est de 88.93 %. En effet, l'étude de la matrice de confusion du système obtenue sur la base de données EMNIST Balanced, présentée sur la figure 3.11, a mis en évidence que la majorité des erreurs étaient faites sur les caractères ('O' et '0'), ('I' et '1' et 'L'), ('F' et 'f') et ('9' et 'g' et 'q'). C'est l'un des facteurs qui entraînent une diminution de la précision du modèle CNN.

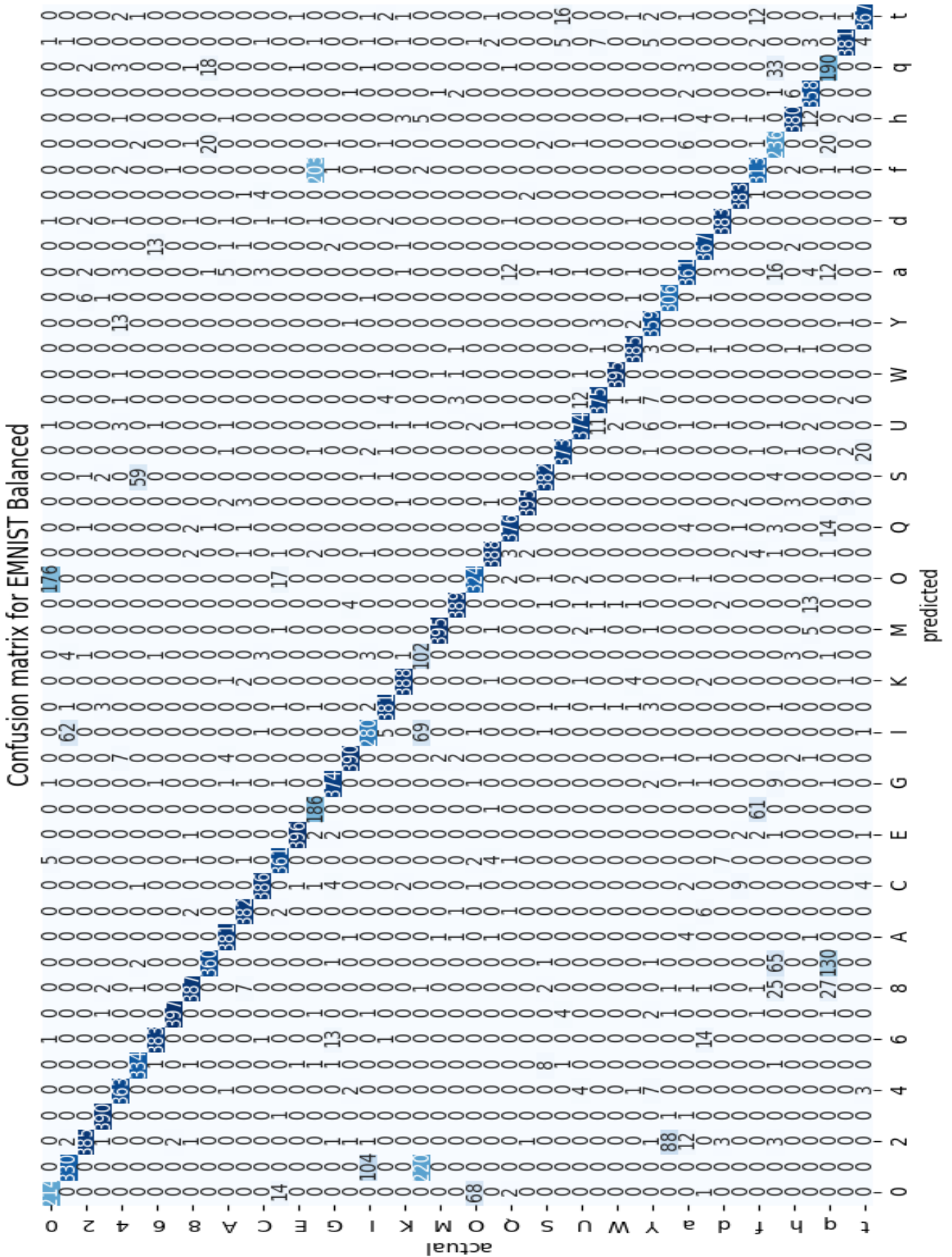


Figure 3. 11: Matrice de confusion pour la base EMNIST Balanced

D'après les matrices de confusions illustrées dans les figures 3.9, 3.10, 3.11, on suppose que la confusion entre les caractères est due à la grande similarité morphologique des caractères. Les figures 3.12 et 3.13 Montre quelques exemples de caractères qui partagent presque les mêmes segments et arcs. On peut donc expliquer cette confusion par la similitude des formes.

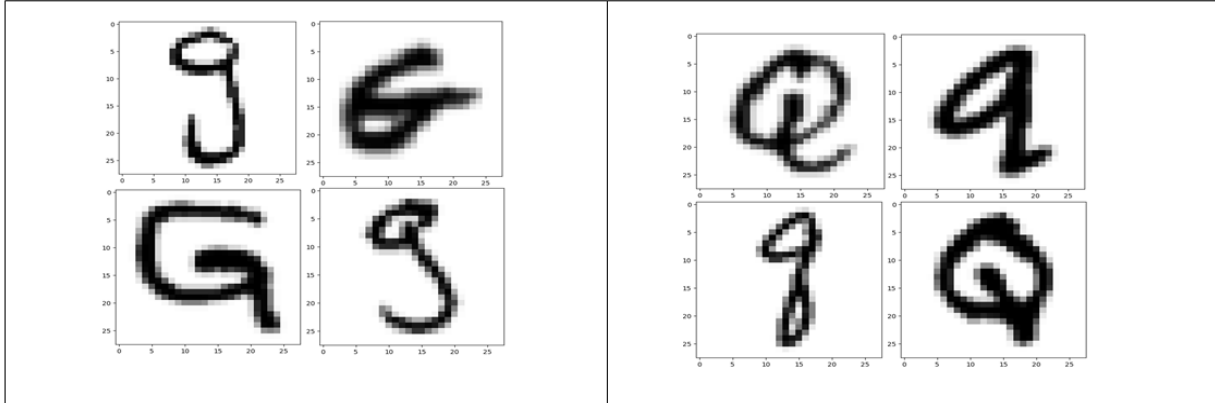


Figure 3.12 : Similarité de la lettre G et la lettre Q

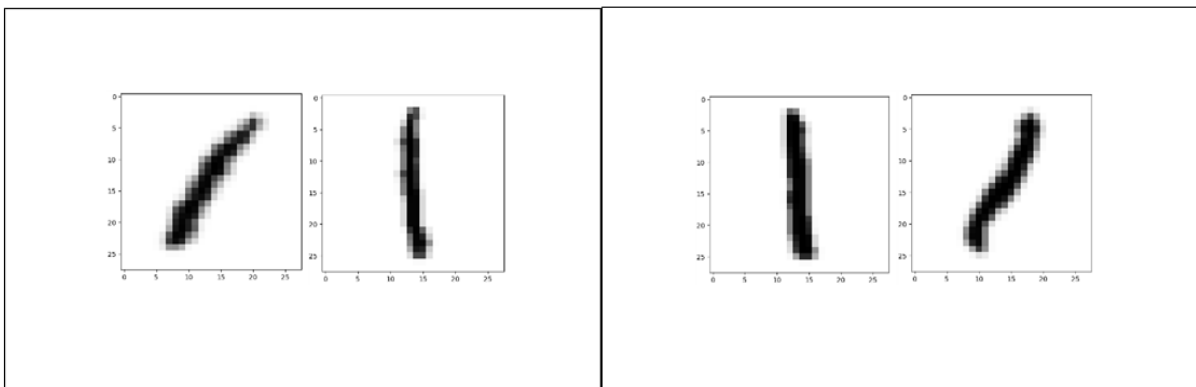


Figure 3.13: Similarité de la lettre I majuscule et la lettre L minuscule

Pour surmonter les erreurs comme ci-dessus, des ajustements peuvent être apportés du côté de l'ensemble de données d'apprentissage du modèle pour les caractéristiques d'écriture ou du côté de l'écriture manuscrite des scripteurs des caractères. En effet, le modèle CNN apprend les caractéristiques d'écriture manuscrite en fonction des caractéristiques des données d'apprentissage utilisées.

Avec l'architecture adoptée, trois modèles de reconnaissance de l'écriture manuscrite utilisant l'architecture CNN proposée ont été construits avec succès pour reconnaître les entrées. Le modèle Lettre est utilisé pour reconnaître les entrées de lettres, le modèle Chiffre est utilisé pour identifier les entrées numériques et le modèle Mixte est utilisé pour identifier les entrées qui contiennent à la fois des lettres et des chiffres. La précision de chaque modèle dans la reconnaissance de l'écriture manuscrite est présentée dans la figure 3.14.

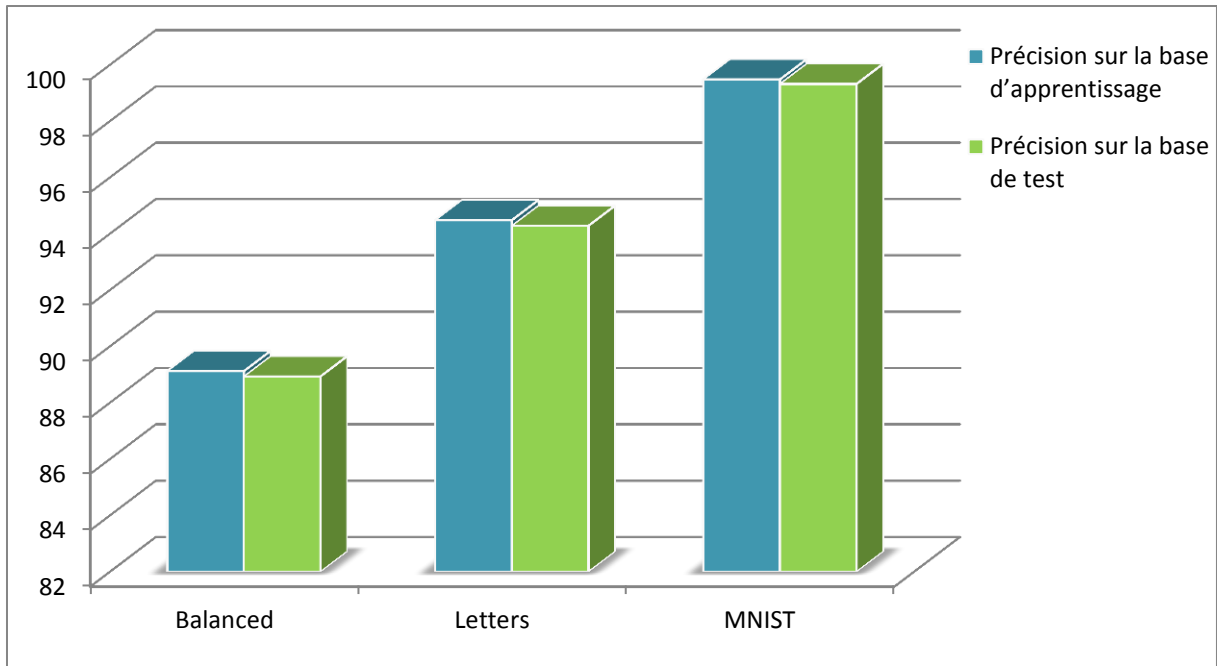


Figure 3.14 : Précision des modèles CNNs conçus

D'après les résultats obtenus, on peut constater la bonne capacité des trois modèles à généraliser, c'est-à-dire la capacité des modèles à faire des prédictions non seulement sur les données que nous avons utilisées pour la construction (données d'apprentissage), mais surtout sur de nouvelles données (données de test), ce qui confirme l'efficacité de l'utilisation de la technique de régularisation Dropout. Cela nous permet de dire que nous avons évité le problème de sur apprentissage et que les trois CNNs sont de bons modèles.

3.4.4. Comparaison avec les algorithmes d'apprentissage automatique

Dans le cadre de ce travail, les modèles CNN proposés sont comparés avec quelques méthodes d'apprentissage automatique en termes de précision. Pour ce faire, la reconnaissance des caractères de l'écriture manuscrite a été réalisée avec six algorithmes d'apprentissage automatique différents. Les ensembles de données de caractères manuscrits EMNIST Letters, EMNIST Balanced et EMNIST MNIST ont été utilisées dans tous les tests.

Après avoir implémenté tous les algorithmes qui sont K-plus proches voisins (KNN), Bayes Naïf, Arbre de décision, Régression logistique, Machine à vecteurs de support, Forêt Aléatoire et les modèles CNNs proposés, nous avons comparé leurs précisions à l'aide de graphiques expérimentaux pour une compréhension claire. Les systèmes sont triés de gauche à droite du moins performant au plus performant.

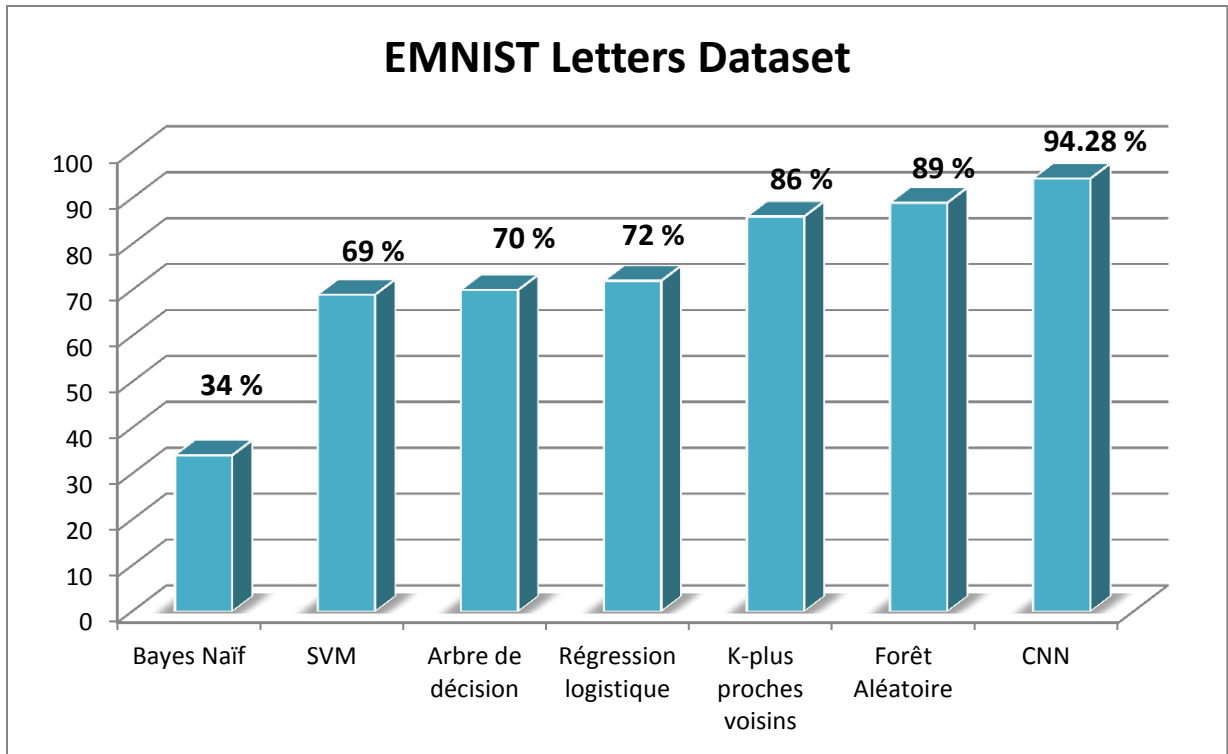


Figure 3.15 : Graphique à barres illustrant la comparaison de précision sur la base EMNIST Letters

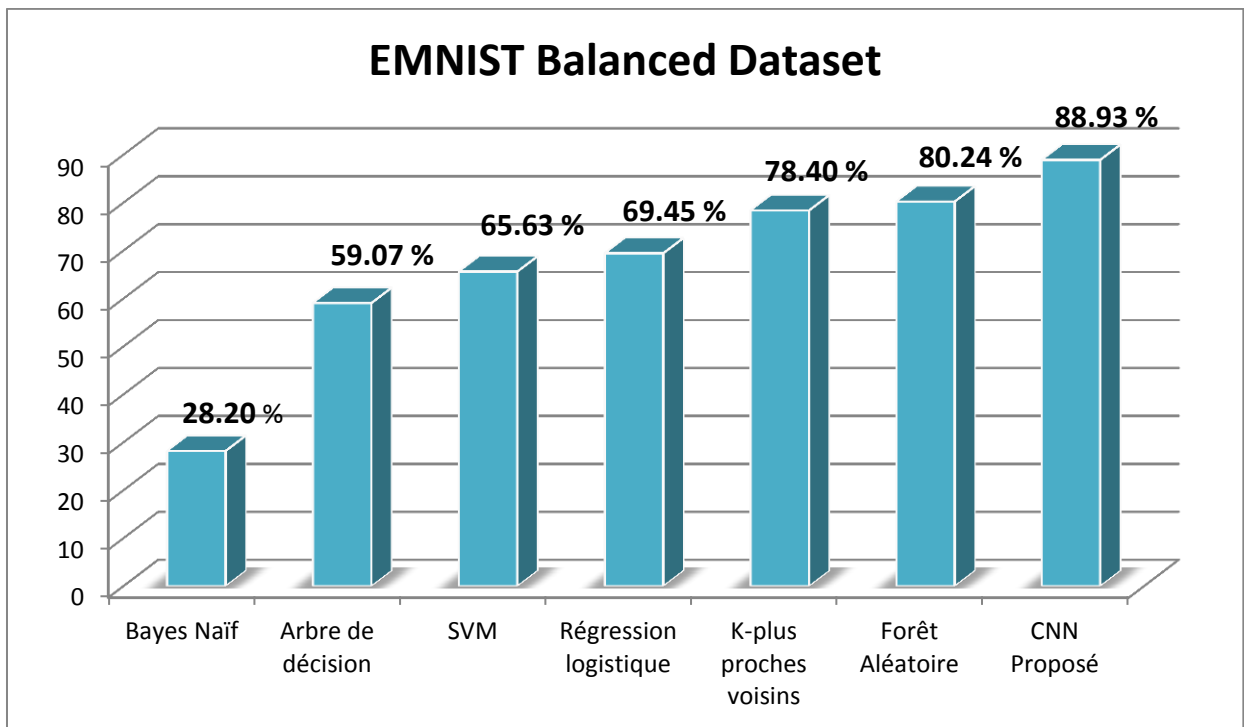


Figure 3.16 : Graphique à barres illustrant la comparaison de précision sur la base EMNIST Balanced

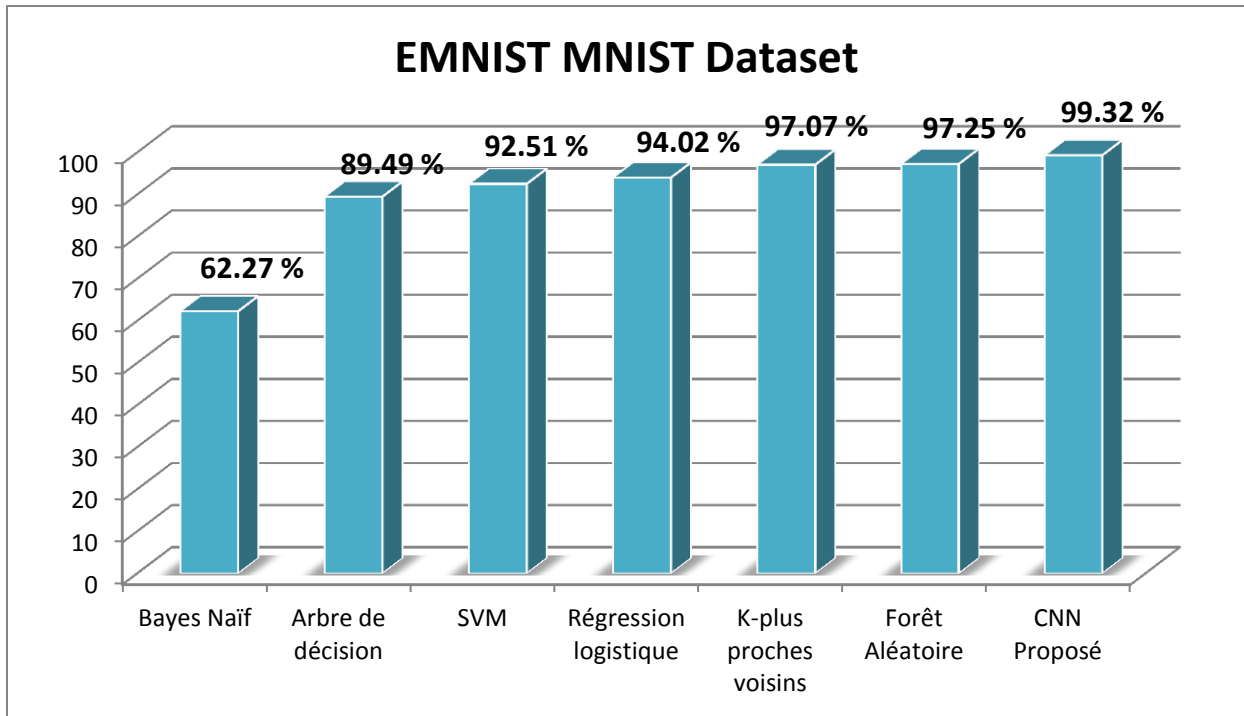


Figure 3.17 : Graphique à barres illustrant la comparaison de précision sur la base EMNIST MNIST

Pour commencer, nous observons très nettement que l'on a obtenu de meilleurs résultats avec le système CNN proposé avec un taux de reconnaissance égal à 94.28 %, 88.93 % et 99.32 % pour EMNIST Letters, EMNIST Balanced et EMNIST MNIST respectivement.

En comparant le taux de reconnaissance obtenu par le système proposé avec celui obtenu par les algorithmes d'apprentissage automatique testés, nous constatons que :

- Malgré que Naïve Bayes, l'un des algorithmes de classification les plus rapides et les plus simples, est utilisé généralement comme référence pour les problèmes de classification, il ne semble pas bien fonctionner pour les ensembles des données EMNIST car il a produit une faible précision globale de 28.20 %, 35 % et 62.27 % pour EMNIST Balanced, EMNIST Letters et EMNIST MNIST respectivement.
- La régression logistique est un algorithme de classification utilisé lorsque la sortie est une catégorie. La régression logistique est considérée plus en moins plus efficace que l'algorithme Naïve Bayes. La précision de la régression logistique est moyenne et peut être acceptable mais elle n'arrive toujours pas à classer des images complexes et ambiguës.
- L'algorithme KNN est l'un des algorithmes de classification les plus appréciés. Bien qu'il ait une structure simple, il a donné des résultats de haute précision.

- Les machines à vecteurs de support sont l'un des classifieurs de base. Dans le cadre de cette étude, il a été observé que le SVM convient à une utilisation dans le domaine de reconnaissance des caractères manuscrits, comme en témoignent ses valeurs de précision.
- Les arbres de décision et les algorithmes de forêts aléatoires sont en fait construits sur la même base. Les algorithmes de forêt aléatoire consistent en des algorithmes d'arbres de décision entraînés. Comme on peut le voir, les forêts aléatoires offrent des précisions plus élevées, par rapport à l'arbre de décision et par rapport aux autres algorithmes d'apprentissage automatique et cela est dû à la prédiction et l'aléatoire qui les prédominent.

Sur la base de ces résultats, nous pouvons conclure que les réseaux de neurones convolutifs (CNN) sont les méthodes les plus adaptées pour le problème de la reconnaissance des caractères. Cela pourrait, à notre avis, être justifié par le fait que :

- Les CNN est une technique d'apprentissage profond, c'est-à-dire, les caractéristiques sont extraites automatiquement sans la nécessité de l'intervention humaine, contrairement aux algorithmes d'apprentissage automatique.
- Les CNN sont des méthodes destinées pour la classification des données bidimensionnelles (2D). Les CNN couvrent les caractéristiques locales et globales à partir des images par des filtres convolutifs qui indiquent les emplacements et la force d'une caractéristique détectée dans une image d'entrée.

3.4.5. Comparaison avec des travaux de la littérature

Bien que le nombre de travaux traitant l'ensemble de données EMNIST soit très faible, en raison de la récence de la base de données, des travaux commencent à émerger avec des résultats très compétitifs. Dans cette section, nous allons effectuer une comparaison des résultats pour l'ensemble de données EMNIST pour certains travaux rapportés dans la littérature et notre système.

Les précisions des travaux sont présentées dans le tableau 3.9. Le but de cette comparaison est d'évaluer notre système par rapport aux travaux existant dans la littérature.

Tableau 3.9 : Comparaison des performances avec l'état de l'art

Base de données	Système	Technique	Précision
EMNIST Letters	(Cohen et al., 2017)	OPIUM	85.15%
	(Ghadekar et al., 2018)	SVM	89.51%
	(Cavalin et al., 2019)	CNN	92.42%
	(Notre système, 2022)	CNN	94.28%
	(Jayasundara, et al., 2019)	CNN	95.36%
	(Jeevan, et al., 2022)	CNN	95.96%
EMNIST Balanced	(Cohen et al., 2017)	Linear Classifier	78.02%
	(Mor et al., 2019)	CNN	87.1%
	(Cavalin et al., 2019)	CNN	87.18%
	(Erdoğan et Tümer, 2021)	CNN	87.81%
	(Notre système, 2022)	CNN	88.93%
	(Mustafa et al., 2021)	CNN	89%

	(Jeevan et al., 2022)	CNN	91.06%
EMNIST MNIST	(Cohen et al., 2017)	Linear classifier	84.70%
	(Ghadekar et al., 2018)	SVM	97.74%
	(Botalb et al., 2018)	CNN	99.2%
	(Dufourq et al., 2017)	HMM	99.30%
	(Notre système, 2022)	CNN	99.32%
	(dos Santos et al., 2019)	CNN	99.77%
	(Shawon et al., 2018)	CNN	99.79%

Et pour une comparaison objective, nous avons présenté les résultats obtenus séparément selon la base de données utilisée dans les graphiques présentés dans les figures 3.18, 3.19 et 3.20. Les systèmes sont triés de gauche à droite du moins performant au plus performant.

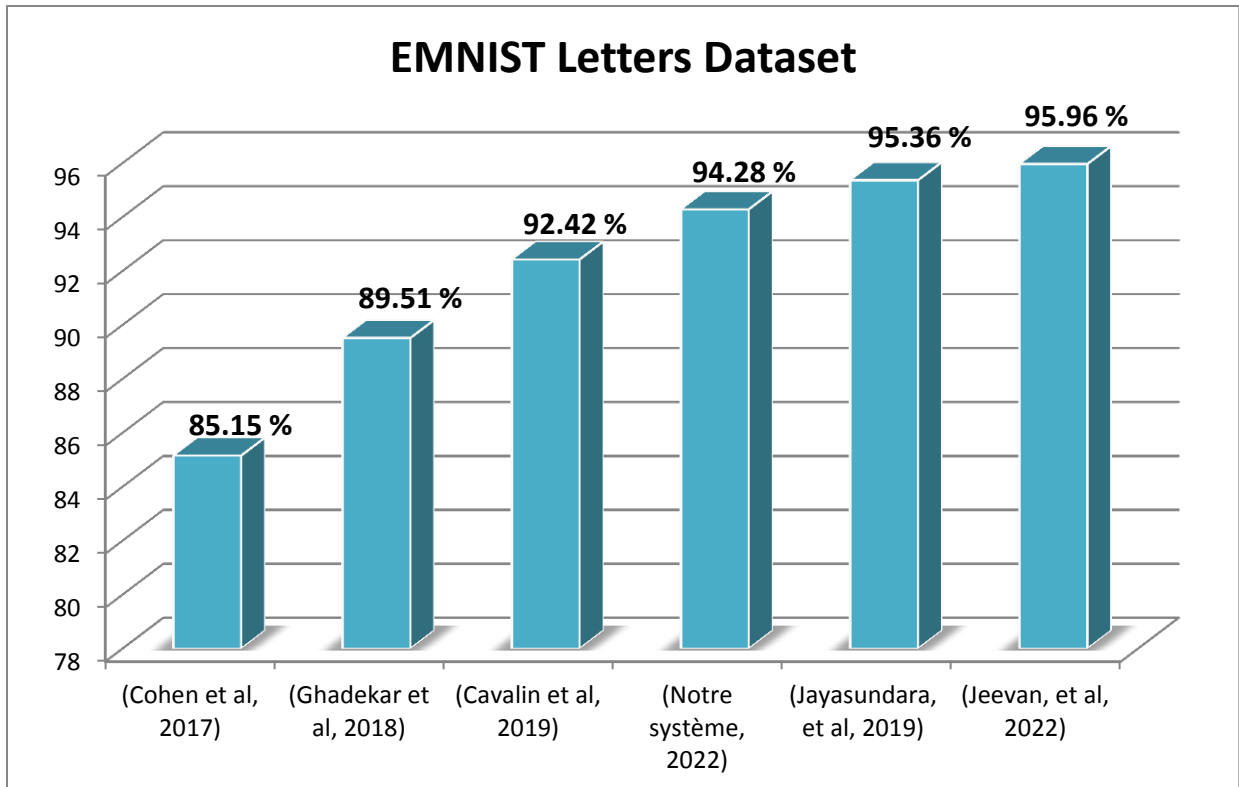


Figure 3.18 : Graphique à barres illustrant la comparaison de précision sur la base EMNIST Letters

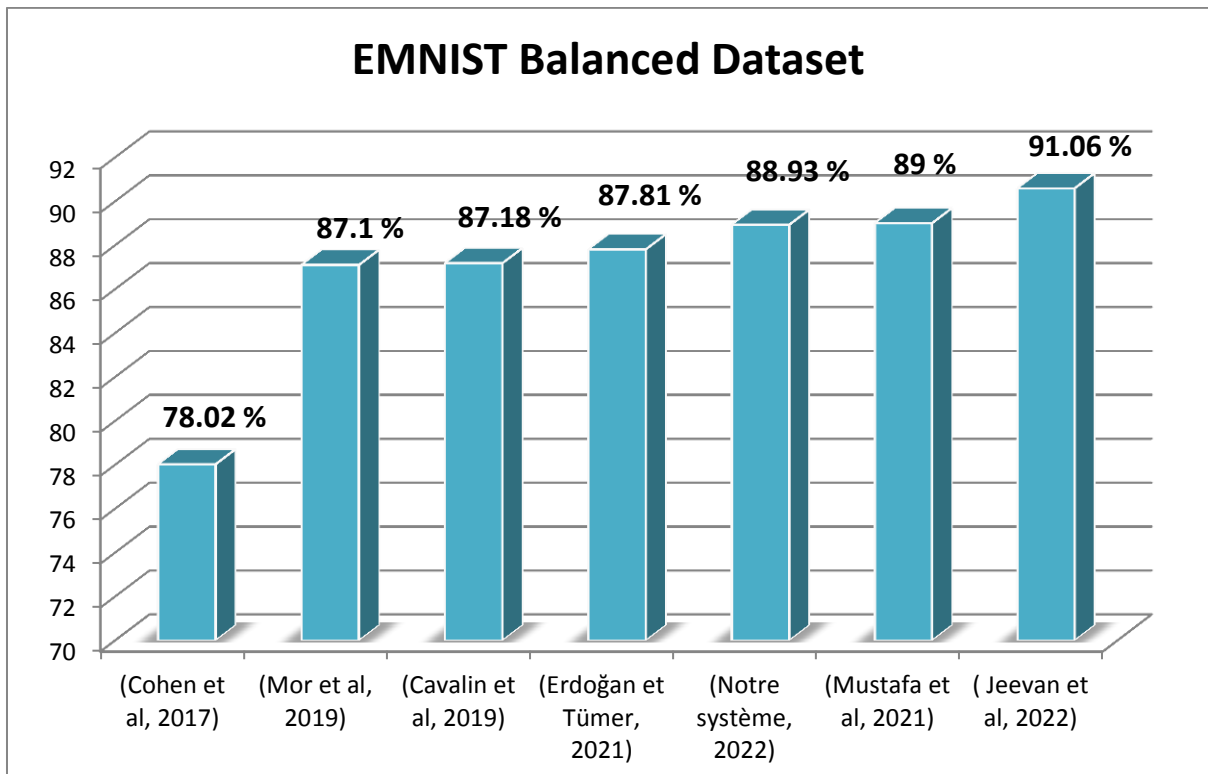


Figure 3.19 : Graphique à barres illustrant la comparaison de précision sur la base EMNIST Balanced

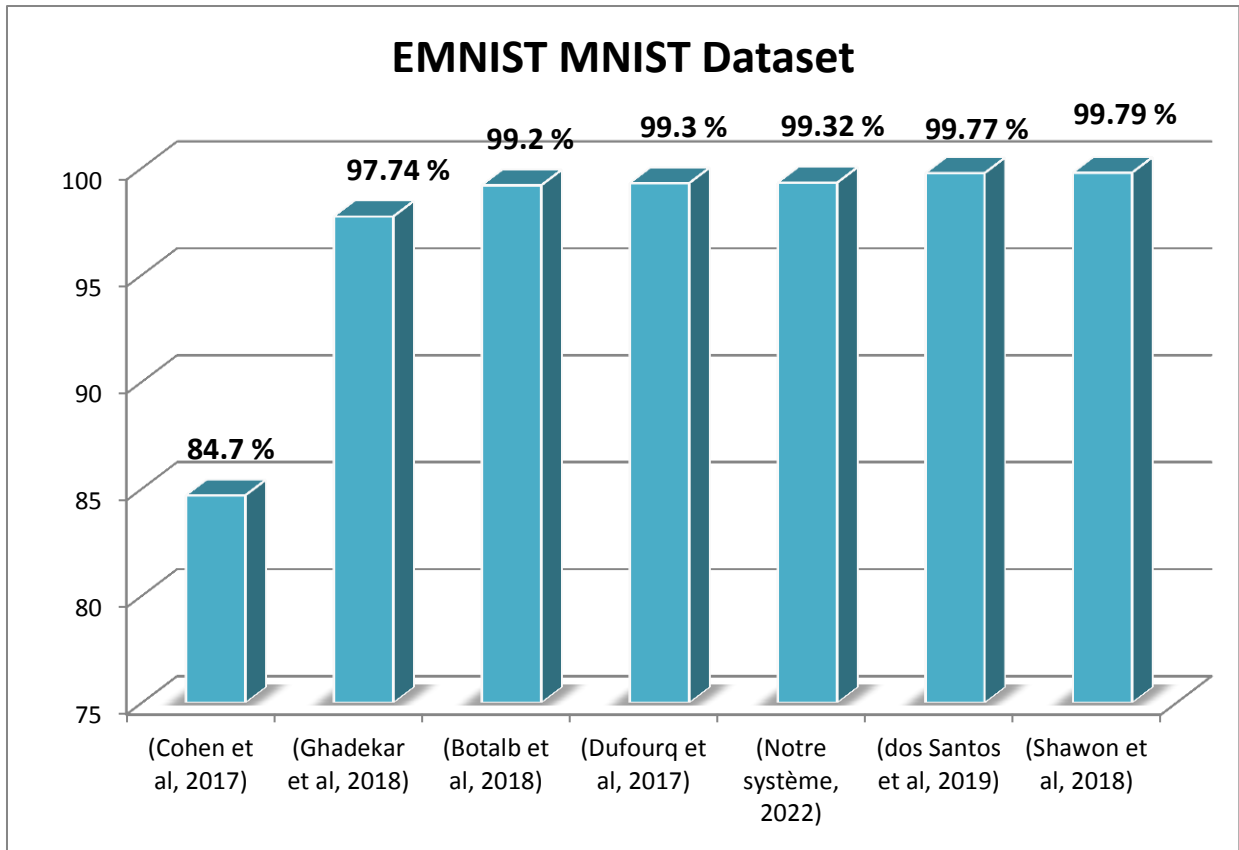


Figure 3.20 : Graphique à barres illustrant la comparaison de précision sur la base EMNIST MNIST

D'après les graphiques, il est remarquable que les meilleurs résultats sont obtenus avec des réseaux de neurones convolutifs. À ce jour (Juin 2022) selon [Web 07], le meilleur résultat obtenu pour cet ensemble de données, en termes de taux de reconnaissance, est 95.96% (Jeevan, et al., 2022) pour l'ensemble des lettres (MNIST Letters), 91.06% (Jeevan et al., 2022) pour la base de données mixte (EMNIST Balanced) et 99.79% (Shawon et al., 2018) pour l'ensemble des chiffres (EMNIST MNIST). Il est, également, intéressant de remarquer que notre système donne des résultats très encourageants (compétitives) par rapport aux meilleurs résultats rapportés avec 94.28% pour l'ensemble EMNIST Lettres, 89.93% pour l'ensemble de données EMNIST Balanced et 99.32% pour l'ensemble de données EMNIST MNIST.

De plus, à partir des résultats, nous pouvons constater que les techniques basées sur le Deep Learning donnent toujours des performances meilleures que celles des algorithmes d'apprentissage classique. Ces techniques sont le plus souvent utilisées avec des réseaux de neurones convolutifs, elles sont capables de se classer en tête du classement absolu dans l'ensemble de données EMNIST et pour la classification des images en général.

3.5. Conclusion

Dans ce chapitre nous avons présenté l'architecture du modèle CNN basé sur l'apprentissage profond pour la reconnaissance des caractères manuscrits. De plus, nous avons analysé l'effet de différentes paramétrisations dans l'architecture proposée sur la précision du modèle. Pour cette tâche, une série d'expérimentations a été réalisée, ce qui nous a permis de fixer les valeurs optimales des hyper paramètres. Par la suite, nous avons étudié l'effet des différents hyper paramètres de modèle sur le problème de la classification des caractères manuscrits pour les trois ensembles de données EMNIST Letters, EMNIST Balanced et EMNIST MNIST. Les résultats obtenus sur ces ensembles étaient très prometteurs : Le modèle de reconnaissance des lettres a eu une précision de 94.28 %, le modèle de reconnaissance des chiffres a eu une précision de 99.32 % et le modèle de reconnaissance des lettres et des chiffres a eu une précision de 88.93 %.

CONCLUSION ET PERSPECTIVES

Sommaire

I.	Bilan.....	78
II.	Perspectives	78

Conclusion et perspectives

I. Bilan

La reconnaissance des caractères et des chiffres est essentielle dans le monde numérisé d'aujourd'hui, en particulier dans les organisations qui traitent des documents manuscrits qu'elles doivent analyser à l'aide de systèmes informatiques. Les systèmes utilisés pour la classification et la reconnaissance de l'écriture manuscrite aident les organisations et les individus à résoudre des tâches complexes.

Notre travail s'inscrit dans le cadre de la reconnaissance des caractères manuscrits. Nous nous intéressons en particulier, à améliorer les performances des systèmes de reconnaissance automatique des caractères manuscrits pour la base de données EMNIST. La difficulté de ce sujet a conduit de nombreux chercheurs à entreprendre de nombreux travaux pour aborder le problème de la reconnaissance. Cependant, la recherche demeure loin d'atteindre le niveau de la capacité humaine dans ce domaine.

Dans ce contexte, nous nous sommes concentrés sur la reconnaissance des lettres, des chiffres et des caractères manuscrits en utilisant un apprentissage profond (Deep Learning). Pour cela, nous avons proposé une architecture CNN. Dans un premier temps, une étude expérimentale est effectuée, dont le but est d'explorer les capacités de l'architecture et identifier les composants et les paramètres qui fournissent les meilleurs résultats en termes de précision. À partir de cette étude nous avons pu définir un système très performant, basé sur des réseaux convolutifs (CNN). Trois modèles de reconnaissance de l'écriture manuscrite ont été construits avec succès pour reconnaître les entrées. Le modèle Lettre est utilisé pour reconnaître les entrées de lettres, le modèle Chiffre pour identifier les chiffres et le modèle Mixte pour identifier les entrées qui contiennent à la fois des lettres et des chiffres. Le taux de reconnaissance obtenu est 94.28% pour le modèle Lettre, 99.32% pour le modèle Chiffre et 88.93 % pour le modèle Mixte. Dans un second temps, Une étude comparative a été menée. Cette comparaison a été divisée en deux, d'abord nous avons comparé le système CNN proposé avec différents algorithmes d'apprentissage automatique en termes de précision, ce qui a montré l'efficacité des réseaux de neurones convolutifs (CNN) pour le problème de reconnaissance de l'écriture manuscrite. Ensuite afin d'évaluer les performances de notre système par rapport à l'état de l'art, les résultats obtenus ont été comparés avec ceux de certains travaux récents utilisant la même base de données. À partir de cette comparaison, nous avons pu conclure que notre système est bien positionné.

II. Perspectives

En ce qui concerne nos perspectives pour les travaux futurs, de nombreuses perspectives sont envisageables pour l'amélioration et la poursuite de nos travaux. Dont les plus importantes sont les suivantes :

Nous envisageons d'améliorer les performances du modèle de reconnaissance des caractères manuscrits en utilisant quelques techniques d'optimisation combinatoire afin de trouver les

meilleurs hyper paramètres car, même si le résultat obtenu est jugé encourageant, de nombreuses améliorations restent à envisager.

Comme travaux futurs, nous souhaitons utiliser les réseaux de neurones convolutifs (CNN) en combinaison avec d'autres classifieurs afin de diminuer le taux d'erreur de notre système.

Nous souhaitons également étendre le modèle pour identifier non seulement les caractères isolés mais aussi les mots et les textes.

BIBLIOGRAPHIE

- Adankon, M. M., & Cheriet, M. (2009). Model selection for the LS-SVM. Application to handwriting recognition. *Pattern Recognition*, 42(12), 3264–3270. doi:10.1016/j.patcog.2008.10.023
- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET). IEEE.
- Ali, A. A. A., & Mallaiah, S. (2021). Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout. *Journal of King Saud University - Computer and Information Sciences*. doi:10.1016/j.jksuci.2021.01.012
- Ali, A. A. A., & Suresha. (2019). Arabic handwritten character recognition using machine learning approaches. 2019 Fifth International Conference on Image Information Processing (ICIIP). IEEE.
- AlKhateeb, J. H., Pauplin, O., Ren, J., & Jiang, J. (2011). Performance of hidden Markov model and dynamic Bayesian network classifiers on handwritten Arabic word recognition. *Knowledge-Based Systems*, 24(5), 680–688. doi:10.1016/j.knosys.2011.02.008
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... Asari, V. K. (2019). A state-of-the-art survey on Deep Learning theory and architectures. *Electronics*, 8(3), 292. doi:10.3390/electronics8030292
- Altwayjry, N., & Al-Turaiki, I. (2021). Arabic handwriting recognition system using convolutional neural network. *Neural Computing & Applications*, 33(7), 2249–2261. doi:10.1007/s00521-020-05070-8
- Amalou, K., & Ben Belkacem, F. (2011). Reconnaissance automatique des noms arabes manuscrits. Mémoire de Master. Université de Tizi-Ouzou. Retrieved May 20, 2022, from Ummto.dz website: <https://www.ummtto.dz/dspace/bitstream/handle/ummtto/12944/BenBelkacemFarida.pdf?sequence=1>
- Amrouch, M., Rabi, M., & Es-Saady, Y. (2018). Convolutional feature learning and CNN based HMM for Arabic handwriting recognition. In *Lecture Notes in Computer Science* (pp. 265–274). Cham: Springer International Publishing.

- Anil, M. C. N. (2022, March 9). Handwriting recognition with ML (an in-depth guide). Retrieved May 20, 2022, from AI & Machine Learning Blog website: <https://nanonets.com/blog/handwritten-character-recognition/>
- Arbain, N. A., Azmi, M. S., Muda, A. K., Muda, N. A., & Radzid, A. R. (Eds.). (2018). Offline handwritten digit recognition using triangle geometry properties". *International Journal of Computer Information Systems and Industrial Management Applications*, 10, 87–097.
- Aries, A. (2009). Proposition d'une architecture multi-agent pour la reconnaissance de l'écriture manuscrite arabe. Mémoire de Master. Université de Jijel. Retrieved May 20, 2022, from Slideshare.net website: <https://fr.slideshare.net/kariminf/pfe-58224378>
- Assegie, T. A., & Nair, P. S. (2019). Handwritten digits recognition with decision tree classification: a machine learning approach". *International Journal of Electrical and Computer Engineering*, 9(5).
- Azeem, S. A., & Ahmed, H. (2013). Effective technique for the recognition of offline Arabic handwritten words using hidden Markov models. *International Journal on Document Analysis and Recognition*, 16(4), 399–412. doi:10.1007/s10032-013-0201-8
- Baldominos, A., Saez, Y., & Isasi, P. (2019). A survey of handwritten character recognition with MNIST and EMNIST. *Applied Sciences (Basel, Switzerland)*, 9(15), 3169. doi:10.3390/app9153169
- Battini, D. (2018). Implementing drop out regularization in neural networks. Retrieved May 20, 2022, from Tech-Quantum website: <https://www.tech-quantum.com/implementing-drop-out-regularization-in-neural-networks/>
- Bendjeddou, T. (2008). Le choix de parametres pour la reconnaissance des chiffres manuscrits. Mémoire de Magistère. Université Badji Mokhtar Annaba. Retrieved May 20, 2022, from Univ-annaba.dz website: <https://biblio.univ-annaba.dz/wp-content/uploads/2014/06/Bendjeddou101.pdf>
- Boesch, G. (2021a). Deep Neural Network: The 3 popular types (MLP, CNN and RNN). Retrieved May 20, 2022, from viso.ai website: <https://viso.ai/deep-learning/deep-neural-network-three-popular-types/>
- Boesch, G. (2021b). What's the difference between Machine Learning and Deep Learning? Retrieved May 20, 2022, from viso.ai website: <https://viso.ai/deep-learning/deep-learning-vs-machine-learning/>
- Botalb, A., Moinuddin, M., Al-Saggaf, U. M., & Ali, S. S. A. (2018). Contrasting convolutional neural network (CNN) with multi-layer perceptron (MLP) for big data analysis. 2018 International Conference on Intelligent and Advanced System (ICIAS). IEEE.

- Brownlee, J. (2017). A gentle introduction to long Short-Term Memory networks by the experts. Retrieved May 20, 2022, from Machine Learning Mastery website: <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/>
- Carrio, A., Sampedro, C., Rodriguez-Ramos, A., & Campoy, P. (2017). A review of deep learning methods and applications for Unmanned Aerial Vehicles. *Journal of Sensors*, 2017, 1–13. doi:10.1155/2017/3296874
- Cavalin, P., & Oliveira, L. (2019). Confusion matrix-based building of hierarchical classification. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (pp. 271–278). Cham: Springer International Publishing.
- Chauhan, N. K., & Singh, K. (2018). A review on conventional machine learning vs deep learning. 2018 International Conference on Computing, Power and Communication Technologies (GUCON). IEEE.
- Chen, S., Almamlook, R., Gu, Y., & Wells, L. (2018). Offline handwritten digits recognition using machine learning”. In *Proceedings of the International Conference on Industrial Engineering and Operations Management*. Washington DC, USA.
- Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017). A. EMNIST: An extension of MNIST to handwritten letters. Retrieved from <http://arxiv.org/abs/1702.05373>
- CRETTEZ, J. P., & LORETTE, G. (1998). Reconnaissance de l'écriture manuscrite. Ed. Techniques Ingénieur.
- Darwish, A., Hassanien, A. E., & Das, S. (2020). A survey of swarm and evolutionary computing approaches for deep learning. *Artificial Intelligence Review*, 53(3), 1767–1812. doi:10.1007/s10462-019-09719-2
- Dhaka, V. S. (2015). Character Recognition of Offline Handwritten English Scripts: A Review.
- Djerouni, H. (2021). Développement d'un système de reconnaissance de chiffres manuscrits. Mémoire de Master. Université Echahid Hamma Lakhdar - D'el Oued. Retrieved May 20, 2022, from Univ-eloued.dz website: <http://dspace.univ-eloued.dz/bitstream/123456789/9757/1/d%C3%A9veloppement%20d%27un%20syst%C3%A9me%20de%20reconnaissance%20de%20chiffres%20manuscrits.pdf>
- Dos Santos, M. M., Da Silva Filho, A. G., & Santos, W. P. (2019). Deep convolutional extreme learning machines: Filters combination and error model validation. *Neurocomputing*, 329, 359–369.
- Dufourq, E., & Bassett, B. A. (2017). EDEN: Evolutionary DEep Networks for efficient machine learning. Retrieved from <http://arxiv.org/abs/1709.09161>

- Dutta, K., Krishnan, P., Mathew, M., & Jawahar, C. V. (2018). Improving CNN-RNN hybrid networks for handwriting recognition. 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE.
- El Abed, H., Kherallah, M., Märgner, V., & Alimi, A. M. (2011). On-line Arabic handwriting recognition competition: ADAB database and participating systems. *International Journal on Document Analysis and Recognition*, 14(1), 15–23. doi:10.1007/s10032-010-0124-6
- Erdoğan, A. A., & Tümer, A. E. (n.d.). Deep Learning Method for Handwriting Recognition. *MANAS Journal of Engineering*, 9(1), 85–92.
- Es Saady, Y. (2012). Contribution au développement d'approches de reconnaissance automatique de caractères imprimés et manuscrits, de textes et de documents Amazighes.
- Fawcett, A. (2021). Data science in 5 minutes: What is one hot encoding? Retrieved May 20, 2022, from Educative: Interactive Courses for Software Developers website: <https://www.educative.io/blog/one-hot-encoding>
- Gary, B. (2020). Convolutional neural network. Retrieved May 20, 2022, from Formation Data Science | DataScientest.com website: <https://datascientest.com/convolutional-neural-network>
- Ghadekar, P., Ingole, S., & Sonone, D. (2018). Handwritten digit and letter recognition using hybrid DWT-DCT with KNN and SVM classifier. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE.
- Hossam, M. (2020). Deep Learning ou Apprentissage Profond: qu'est-ce que c'est? Retrieved May 20, 2022, from Formation Data Science | DataScientest.com website: <https://datascientest.com/deep-learning-definition>
- Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1), 215–243. doi:10.1113/jphysiol.1968.sp008455
- Jayasundara, V., Jayasekara, S., Jayasekara, H., Rajasegaran, J., Seneviratne, S., & Rodrigo, R. (2019). TextCaps: Handwritten character recognition with very small datasets. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE.
- Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing: Official Publication of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, 173, 24–49. doi:10.1016/j.isprsjprs.2020.12.010
- Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), 5455–5516. doi:10.1007/s10462-020-09825-6

- Khémiri, A., Echi, A. K., & Elloumi, M. (2019). Bayesian versus convolutional networks for Arabichandwriting recognition. *Arab J Sci Eng*, 1–19.
- Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst*. [https://doi.org/10.1061/\(ASCE\)GT.1943-5606.0001284](https://doi.org/10.1061/(ASCE)GT.1943-5606.0001284)
- Kumar, N. (2021). Introduction to Support Vector Machines (SVMs). Retrieved May 20, 2022, from MarkTechPost website: <https://www.marktechpost.com/2021/03/25/introduction-to-support-vector-machines-svms/>
- Lecun, Y., Jackel, L. D., & Bottou, L. (1995). Learning algorithms for classification: a comparison on handwritten digit recognition. *Neural Netw Stat Mech Perspect*, 261.
- Lin M, Chen Q, Yan S (2013) Network in network, pp 1–10. <https://doi.org/10.1109/asru.2015.7404828> Lin T-Y, Maire M, Belongie S et al (2014) Microsoft coco: common objects in context. In: *European conference on computer vision*. Springer, pp 740–755
- Mioulet, L. (2015). Reconnaissance de l'écriture manuscrite avec des réseaux récurrents (Doctoral dissertation, Université de rouen).
- Mishra, H., Pathak, S. K., & Srivastava, A. (2020). Handwritten digits recognition using machine learning algorithms". *International Research Journal of Engineering and Technology (IRJET)*, 7(05), 8073–8079.
- Mor, S. S., Solanki, S., Gupta, S., Dhingra, S., Jain, M., & Saxena, R. (2019). Handwritten text recognition: With deep learning and Android. *Int. J. Eng. Adv. Technol*, 8, 172–178.
- Mustafa, Y. F., Ridho, F., & Mariyah, S. (2021a). Study of Handwriting Recognition Implementation in Data Entry of Survei Angkatan Kerja Nasional (SAKERNAS) using CNN. In *The 1st international conference on data science and official statistics*.
- Mustafa, Y. F., Ridho, F., & Mariyah, S. (2021b). Study of Handwriting Recognition Implementation in Data Entry of Survei Angkatan Kerja Nasional (SAKERNAS) using CNN. *Proceedings of The International Conference on Data Science and Official Statistics, 2021*, 53–65.
- Nazir, S., Patel, S., & Patel, D. (2018). Hyper parameters selection for image classification in convolutional neural networks. *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*. IEEE.
- OpenGenus. (2019). Evolution of CNN architectures: LeNet, AlexNet, ZFNet, GoogleNet, VGG and ResNet. Retrieved May 20, 2022, from OpenGenus IQ: Computing Expertise & Legacy website: <https://iq.opengenus.org/evolution-of-cnn-architectures/>

- Pashine, S., Dixit, R., & Kushwah, R. (2021). Handwritten digit recognition using machine and deep learning algorithms. Retrieved from <http://arxiv.org/abs/2106.12614>
- Peng, Y., Yin, H., Yin, H., Gao, Y., Chen, S., Wen, Y., ... Du, J. (2017). Markov Random Field Based Convolutional Neural Networks for Image Classification. In A. Tallón-Ballesteros & M. Zhang (Eds.), *IDEAL 2017: Intelligent Data Engineering and Automated Learning* (Vol. 10585, pp. 387–396). Guilin, China: Springer.
- Pluchot, P. (2021). Musée de l'école à Montceau-les-Mines. Retrieved May 20, 2022, from Blogspot.com website: <https://musee-ecole-montceau-71.blogspot.com/2021/01/lecriture-lecole.html>
- Poisson, E. (2005). Architecture et apprentissage d'un système hybride neuro-markovien pour la reconnaissance de l'écriture manuscrite en-ligne. Ecole polytechnique de l'université de Nantes.
- Prasad, B. K., & Sanyal, G. (2018). Novel features and a cascaded classifier based Arabic numerals recognition system". *Multidim Syst Sign Process*, 29, 321–333.
- Priya, A., Mishra, S., Raj, S., Mandal, S., & Datta, S. (2016). Online and offline character recognition: A survey. 2016 International Conference on Communication and Signal Processing (ICCSP). IEEE.
- Ramesh, G., Tejas, M., Thakur, R., & Champa, H. N. (2021). Handwritten Kannada digit recognition system using CNN with random forest. In *Communications in Computer and Information Science* (pp. 92–104). Cham: Springer International Publishing.
- Rashmi, J. (2017). Simple tutorial on SVM and parameter tuning in python and R. Retrieved May 20, 2022, from HackerEarth Blog website: <https://www.hackerearth.com/blog/developers/simple-tutorial-svm-parameter-tuning-python-r/>
- Rezgui, Z. (2019). Rapport de stage: Détection et classification de visages pour la description de l'égalité femme-homme dans les archives télévisuelles. University of Carthage.
- Rosyda, S. S., & Purboyo, T. W. (2018). A review of various handwriting recognition methods. *International Journal of Applied Engineering Research*, 13(2), 1155–1164.
- Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. Retrieved from <http://arxiv.org/abs/1710.09829>
- Saidi, A. (2021). Etude et réalisation d'un système de reconnaissance d'écriture arabes. Thèse de doctorat. Université Tahiri Mohamed de Bechar.
- Shawon, A., Jamil-Ur Rahman, M., Mahmud, F., & Arefin Zaman, M. M. (2018). Bangla handwritten digit recognition using deep CNN for large and unbiased dataset. 2018 International Conference on Bangla Speech and Language Processing (ICBSLP). IEEE.

- Shetty, A. B., Ail, N., Sahana, M., & Bhat, V. P. (2021). Recognition of Handwritten Digits and English Texts using MNIST and EMNIST Datasets”. International Journal of Research in Engineering.
- Siddhu, M. K., Parvez, M. T., & Yaakob, S. N. (2019). Combining statistical and structural approaches for Arabic handwriting recognition. 2019 International Conference on Computer and Information Sciences (ICCIS). IEEE.
- Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. ICLR 75:398–406. <https://doi.org/10.2146/ajhp170251>
- Singh, S., Paul, A., & Arun, M. (2017). Parallelization of digit recognition system using Deep Convolutional Neural Network on CUDA. 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS). IEEE.
- Sueiras, J. (2021). Continuous offline handwriting recognition using deep learning models. Retrieved from <http://arxiv.org/abs/2112.13328>
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
- Tappert, C. C., Suen, C. Y., & Wakahara, T. (1990). The state of the art in online handwriting recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(8), 787–808. doi:10.1109/34.57669
- Vidhya. (2021). K-Nearest Neighbors in Machine learning (k-NN) - analytics Vidhya - medium. Retrieved May 20, 2022, from Analytics Vidhya website: <https://medium.com/analytics-vidhya/k-nearest-neighbors-in-machine-learning-k-nn-e84a70b79b3c>
- Wolfewicz, A. (2021). Deep learning vs. Machine learning – what’s the difference? Retrieved May 20, 2022, from Levity.ai website: <https://levity.ai/blog/difference-machine-learning-deep-learning> [Original source: <https://studycrumb.com/alphabetizer>]

WEBOGRAPHIE

Web 01 Binarisation d'images - Binarisation. (n.d.). Retrieved May 5, 2022, from Google.com website: <https://sites.google.com/site/lizantchristopher/services>

Web 02 Reconnaissance de l'écriture manuscrite. Retrieved May 5, 2022, from Wikipedia.org website: https://fr.wikipedia.org/wiki/Reconnaissance_de_l%27écriture_manuscrite

Web 03 Python (programming language). Retrieved May 5, 2022, from Wikizero.com website: [https://www.wikizero.com/en/Python_\(language\)](https://www.wikizero.com/en/Python_(language))

Web 04 K-Nearest Neighbor (KNN) Algorithm for Machine Learning. (n.d.). Retrieved May 7, 2022, from www.javatpoint.com website: <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>

Web 05 12 types of neural networks activation functions: How to choose? . Retrieved May 5, 2022, from V7labs.com website: <https://www.v7labs.com/blog/neural-networks-activation-functions>

Web 06 Convolutional Neural Network - [javatpoint](http://javatpoint.com). (n.d.). Retrieved May 7, 2022, from www.javatpoint.com website: <https://www.javatpoint.com/pytorch-convolutional-neural-network>

Web 07 Papers with code - EMNIST-Letters benchmark (image classification). (n.d.). Retrieved June 17, 2022, from Paperswithcode.com website: <https://paperswithcode.com/sota/image-classification-on-ernist-letters>

OUTILS D'IMPLEMENTATION ET APPLICATION

Sommaire

1.	Introduction	89
2.	Outils logiciels	89
3.	Etapes d'implémentation	92
4.	Présentation de l'application	95
5.	Conclusion	98

1. Introduction

Dans cette annexe, nous présentons d'abord les outils de développements, puis nous décrivons les différentes étapes pour la mise en œuvre du système de reconnaissance des caractères manuscrits, ainsi que les interfaces et le fonctionnement de système.

2. Outils logiciels

Dans cette section, nous présenterons les outils logiciels nécessaires pour le développement de notre système.

2.1. Python

Python est un puissant langage de programmation interprété, interactif, orienté objet et de haut niveau à usage général créé par Guido van Rossum à la fin des années 1980, et publié pour la première fois en 1991. Python est basé sur une philosophie de conception qui met l'accent sur la lisibilité du code. En fait, l'une de ses forces est la simplicité de sa syntaxe. La langue se lit presque comme l'anglais ordinaire, ce qui facilite l'écriture de programmes complexes et le développement d'une grande variété d'applications.

Python est un langage de programmation multi-paradigme, il prend en charge la programmation structurée (en particulier procédurale), orientée objet et fonctionnelle. Il est souvent décrit comme un langage "piles incluses" en raison de sa bibliothèque standard complète. Étant open source, il prend également en charge une large gamme de systèmes d'exploitation [Web 03]. Python est largement utilisé pour le développement Web, l'analyse de données, l'intelligence artificielle, les réseaux de neurones, le calcul scientifique et d'autres domaines de l'informatique avancée.

2.1.1. Importance du langage de programmation Python

Python fait partie de ces rares langages qui peuvent être à la fois simples et puissants et leurs fonctionnalités sont les suivantes (Lakshminarayanan et Prabhakaran, 2020) :

- **Simple et facile à apprendre** : Python est extrêmement simple et facile à lire et à apprendre, coder en Python, c'est comme écrire des phrases simples en anglais strict. En fait, l'une de ses forces souvent vantées est la façon dont le code Python apparaît comme un pseudo-code. Cela permet aux programmeurs de se concentrer sur la solution au problème plutôt que sur la langue elle-même.
- **Libre et Open Source** : Python est un exemple de FLOSS (Free/Libre and Open Source Software). Ce FLOSS est basé sur le concept d'une communauté qui partage des connaissances et il est constamment amélioré par une communauté qui conduit à voir un meilleur Python complètement gratuit même pour un usage commercial.
- **Typé dynamiquement** : les types de variables, d'objets, etc. en Python sont généralement déduits lors de l'exécution et non déclarés statiquement.
- **Portable/ indépendant de la plate-forme/multiplateforme** : Python peut fonctionner sur une grande variété de plates-formes et possède la même interface sur toutes les plates-formes en raison de sa nature open source. Il fonctionne nativement sur Windows, Mac

OS, Linux et d'autres. En d'autre terme tous les programmes Python peuvent fonctionner sur n'importe laquelle de ces plates-formes sans nécessiter aucune modification.

- **Extensible** : Python a une intégration de première classe avec C ou C++, c'est-à-dire le code Python peut être intégré dans un programme C/C++ pour lui donner des capacités de script et vice versa.
- **Multi-paradigme** : Python prend en charge divers paradigmes de programmation et d'implémentation, tels que la programmation orientée objet, fonctionnelle ou procédurale.
- **Bibliothèques étendues** : La bibliothèque standard Python est énorme et offre un large éventail de fonctionnalités. Il contient des modules intégrés écrits en Python qui fournissent des solutions standardisées pour de nombreux problèmes mathématiques, statistiques, informatiques et d'autres domaines scientifiques.
- **Garbage Collection** : Python s'occupe lui-même de l'allocation et de la désallocation de la mémoire. En d'autres termes, un programmeur n'a pas à gérer l'allocation de mémoire et n'a pas besoin de réallouer et de désallouer de la mémoire avant de construire des variables et des objets.

2.1.2. Bibliothèques utilisées

Les bibliothèques Python (packages) simplifient efficacement de nombreux processus importants tels que l'analyse et la visualisation de données, la récupération de données non structurées sur le Web, le traitement d'images, la création de modèles d'apprentissage automatique Etc. Différents bibliothèques ont été utilisées pour développer les expérimentations et effectuer les différentes étapes du processus de reconnaissance, dans cette section nous allons présenter les bibliothèques utilisées et leurs utilités. (Gholizadeh, 2022), (Bellahmer, 2020).

a. NumPy

NumPy est l'abréviation de « Numerical Python » et c'est un ensemble fondamental pour le calcul scientifique en Python. NumPy fournit à Python une vaste bibliothèque mathématique capable d'effectuer des calculs numériques de manière efficace et efficiente afin de pouvoir travailler avec des tableaux multidimensionnels et des structures de données matricielles, très courante dans les domaines de la science des données et de l'apprentissage automatique. NumPy est très utile et plus efficace que les listes de Python normales en raison de sa vitesse et de sa mémoire.

b. Scikit-Learn

Scikit-Learn est une bibliothèque d'apprentissage automatique. Après avoir nettoyé et manipulé les données, Scikit-Learn est utilisé pour construire des modèles d'apprentissage automatique, car elle dispose de milliers d'outils utilisés pour la modélisation et l'analyse prédictive. Il existe plusieurs types de modèles d'apprentissage automatique qui peuvent être construits à l'aide de Scikit-Learn, y compris l'apprentissage supervisé et non supervisé. Scikit-Learn dispose de divers algorithmes de classification, de régression, de Clustering et réduction de la dimensionnalité.

c. Matplotlib

Matplotlib est la bibliothèque la plus populaire pour explorer et visualiser des données. Elle est utilisée pour créer des graphiques de base tels que des graphiques linéaires, des nuages de points, des histogrammes, des graphiques à barres et des graphiques à secteurs. Matplotlib est la base de toute autre bibliothèque visuelle. Il s'agit d'une bibliothèque de traçage pour le langage de programmation Python et l'extension numérique NumPy. Il permet aux décideurs de voir des modèles, des tendances et des corrélations qui pourraient ne pas être détectés dans des données textuelles.

d. Seaborn

Seaborn est construit sur Matplotlib. Utilisé pour dessiner des graphiques statistiques attrayants et informatifs. Seaborn est aussi un support spécialisé pour les variables de catégorie pour montrer les observations. Seaborn dispose d'outils pour sélectionner des palettes de couleurs qui révèlent des motifs cachés dans les données. Principales caractéristiques de la raison pour laquelle nous utilisons Seaborn :

- **Fonctionnalité** : Utilise moins de syntaxe et propose des thèmes par défaut simples et intéressants.
- **Flexibilité** : Fournit les thèmes par défaut les plus utilisés
- **Gestion de plusieurs figures** : automatise la création de plusieurs figures pouvant entraîner des problèmes de mémoire insuffisante.

e. TensorFlow

TensorFlow est une bibliothèque open source gratuite et multi-plateforme pour la mise en œuvre et l'exécution d'algorithmes d'apprentissage automatique. TensorFlow permet une exécution à la fois sur les CPU et les GPU. Toutefois, ses plus grandes capacités de performance peuvent être découvertes lors de l'utilisation des GPU. Elle peut être utilisée dans un large éventail de tâches, mais se concentre particulièrement sur la formation et l'inférence des réseaux de neurones profonds. L'API Python de TensorFlow est actuellement l'API la plus complète, attirant ainsi de nombreux praticiens de l'apprentissage automatique et de l'apprentissage profond. Les API TensorFlow sont organisées de manière hiérarchique et les API de haut niveau sont construites sur des API de bas niveau, par exemple tf.keras est une variante TensorFlow de l'API open source Keras. Les chercheurs en apprentissage automatique utilisent des API de bas niveau pour créer et découvrir de nouveaux algorithmes d'apprentissage automatique.

f. Keras

Keras est une API d'apprentissage profond de haut niveau et s'exécute au-dessus de la plate-forme d'apprentissage automatique TensorFlow. Elle a été développée en mettant l'accent sur la possibilité d'expériences rapides. Les Keras sont principalement utilisés pour créer des modèles d'apprentissage en profondeur, en particulier des réseaux de neurones. L'API de Keras est facile à utiliser et permet de construire des modèles en ajoutant simplement des couches les unes sur les autres par de simples appels. Keras peut également

être utilisé pour fournir des solutions d'apprentissage automatique appliquées fiables et performantes, ainsi que dans le traitement du langage naturel (NLP) et la vision par ordinateur (CV).

2.2. Pycharm IDE

PyCharm est un environnement de développement intégré (IDE) qui fournit une large gamme d'outils essentiels pour programmer en Python. Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui permet l'analyse de code et contient un débogueur graphique. Il permet également le développement du Web et de la science des données.

Pour le développement de notre application, nous avons choisi PyCharm Community Edition 2018.3.7.

3. Etapes d'implémentation

Voici les étapes pour mettre en œuvre le système de reconnaissance des caractères manuscrits. Ci-dessous nous prenons l'exemple de reconnaissance des lettres manuscrites :

3.1. Importation des bibliothèques et chargement de l'ensemble de données

Tout d'abord, nous allons importer tous les modules dont nous aurons besoin pour entraîner notre modèle. La bibliothèque Keras contient déjà des ensembles de données et EMNIST en fait partie. Nous pouvons donc facilement importer l'ensemble de données et commencer à travailler avec.

La figure 1 montre un morceau de code permettant d'importer les bibliothèques nécessaires.

```
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout, Dense
import numpy as np
import cv2
```

Figure 1 : Quelques bibliothèques importées

3.2. Prétraitement des données

Les données d'image ne peuvent pas être introduites directement dans le modèle, nous devons donc effectuer certaines opérations et traiter les données pour les préparer pour notre réseau de neurones.

3.3. L'architecture du CNN utilisé

Avec la bibliothèque Keras, on crée un modèle CNN séquentiel. Ce modèle est constitué de trois paires de couches convolution – Max Pooling permettant l'extraction automatique de caractéristiques.

Après avoir extrait toutes les caractéristiques, nous allons utiliser :

- Flatten : L'aplatissement transforme les données en un tableau unidimensionnel.
- Deux couches cachées (Dense) : La première prend en entrée le résultat de l'image aplati de la couche précédente. La deuxième désigne le nombre de classes.
- Softmax : est utilisée comme fonction d'activation dans la couche de sortie, qui convertit un vecteur de nombres en un vecteur de probabilités, où la somme des probabilités est égale à 1.

```
def model() :  
  
    model = Sequential()  
  
    model.add(Conv2D(16, (5,5), input_shape= (28,28,1), activation= "relu"))  
    model.add(MaxPooling2D(2,2))  
  
    model.add(Conv2D(32, (5,5), activation="relu"))  
    model.add(MaxPooling2D(2,2))  
  
    model.add(Conv2D(64, (3,3), activation="relu"))  
    model.add(MaxPooling2D(2,2))  
  
    model.add(Dropout(0.2))  
  
    model.add(Flatten())  
  
    model.add(Dense(256, activation = "relu"))  
    model.add(Dense(26, activation = "softmax"))  
  
    model.compile(loss= "categorical_crossentropy",  
                  optimizer= "adam",  
                  metrics=["accuracy"])  
  
    return model
```

Figure 2 : Code source permettant la définition de l'architecture du modèle CNN (Modèle Lettres)

3.4. Apprentissage

La fonction `model.fit()` de Keras lancera l'apprentissage du modèle. Il prend, comme entrées, les données d'apprentissage, les données de validation, le nombre d'époques et la taille du batch.

```
hist = model.fit(x_train, y_train,  
                batch_size = batch_size,  
                epochs = epochs,  
                validation_data=(x_val., y_val))
```

Figure 3 : Fonction d'apprentissage

Après l'apprentissage, nous enregistrons le modèle dans le fichier "modele_lettre.h5".

```
model.save('modele_lettre.h5')
```

Figure 4 : Sauvegarder le modèle

3.5. Reconnaissance

Nous avons utilisé la fonction `model.predict()` qui prend l'image en entrée et utilise ensuite le modèle formé pour prédire le caractère.

```
y_pred = model.predict(x_train)
```

Figure 5 : Fonction de reconnaissance

3.6. Évaluation du modèle

Nous avons 28 000 images dans notre ensemble de données qui seront utilisées pour évaluer la qualité de notre modèle. Les données de test n'ont pas été impliquées dans l'apprentissage, il s'agit donc de nouvelles données pour notre modèle. La précision obtenue est environ 94.28%.

```
Accuracy(np.argmax(y_test, axis=1), np.argmax(y_pred, axis=1))
```

Figure 6 : Fonction de précision

La reconnaissance des chiffres et la reconnaissance des caractères (Lettres et chiffres) se passent de la même manière que celle expliquée dans la section 3, c'est-à-dire, en suivant les mêmes étapes, uniquement en changeant la base de données selon le type de reconnaissance.

4. Présentation de l'application

Au démarrage de l'application, une fenêtre d'accueil apparaît avec trois boutons « Apprentissage », « Reconnaissance » et « Fermer ».

- Bouton Apprentissage : Permet d'afficher l'interface d'apprentissage.

- Bouton Reconnaissance : Permet d'afficher l'interface de reconnaissance.
- Bouton Fermer : Permet de quitter l'application.

La figure 7, représente l'interface d'accueil de notre système.

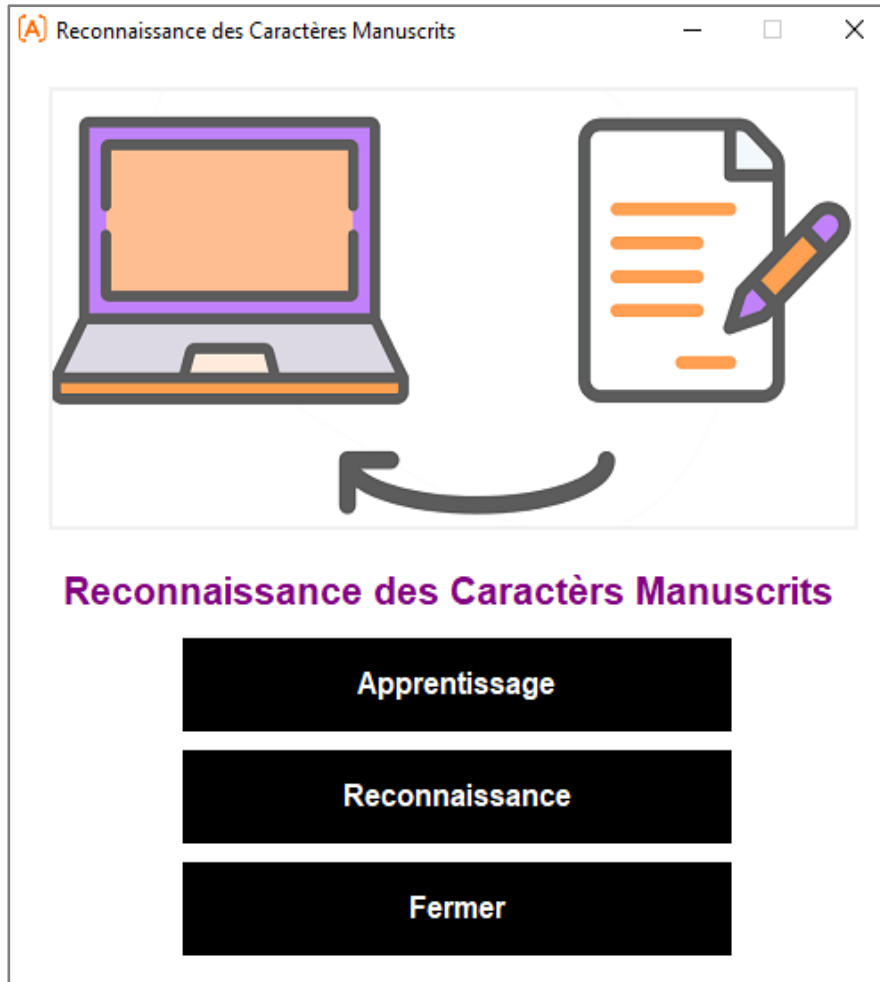


Figure 7 :L'interface d'accueil de notre application

En cliquant sur le bouton « Apprentissage », une nouvelle fenêtre apparait. Cette fenêtre permet de former le modèle. Elle contient trois boutons. La figure 8, montre l'interface d'apprentissage.

Apprentissage



Choix des hyper paramètres

Paire Couche Con_MaxPool

Feature Maps

1er Couche 2eme Couche 3eme Couche 4eme Couche

Dense

Dropout

Sauvegarder les paramètres

Train Fermer

Figure 8 : L'interface d'apprentissage.

- Bouton Sauvegarder les paramètres : Permet de sauvegarder les hyper paramètres choisis.
- Bouton Train: Permet de déclencher l'opération d'apprentissage.
- Bouton Fermer: Permet de revenir à la fenêtre d'accueil.

Le bouton « Reconnaissance », nous amène à une autre fenêtre. La figure 9, présente l'interface de la reconnaissance.

- Bouton Lettres : Pour reconnaître les entrées lettres.
- Bouton Chiffres : Pour reconnaître les entrées numériques.
- Bouton Mixtes : Pour reconnaître les entrées qui contiennent à la fois des lettres et des chiffres.

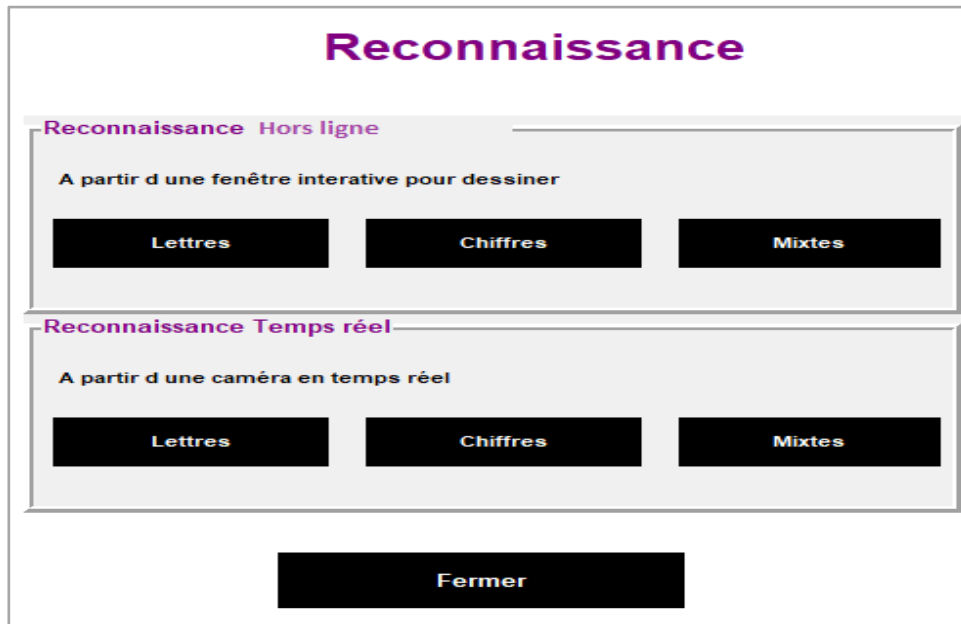


Figure 9 : L'interface de reconnaissance.

Les boutons présents dans le frame «Reconnaissance Hors-ligne», permettent d'afficher une fenêtre interactive pour dessiner des caractères et avec un bouton, nous pouvons reconnaître le caractère. Cette fenêtre contient trois boutons :

- Prédire : Permet de déclencher l'opération de reconnaissance du caractère et afficher le résultat.
- Effacer : Bouton pour la réinitialisation (supprimer le graphique).
- Fermer : Permet de revenir à la fenêtre de reconnaissance.

Les boutons présents dans le frame «Reconnaissance Temps réel», permettent la reconnaissance en temps réel à travers la caméra de l'ordinateur portable.

La figure 10, montre quelques exemples de résultats de reconnaissance Hors-ligne.

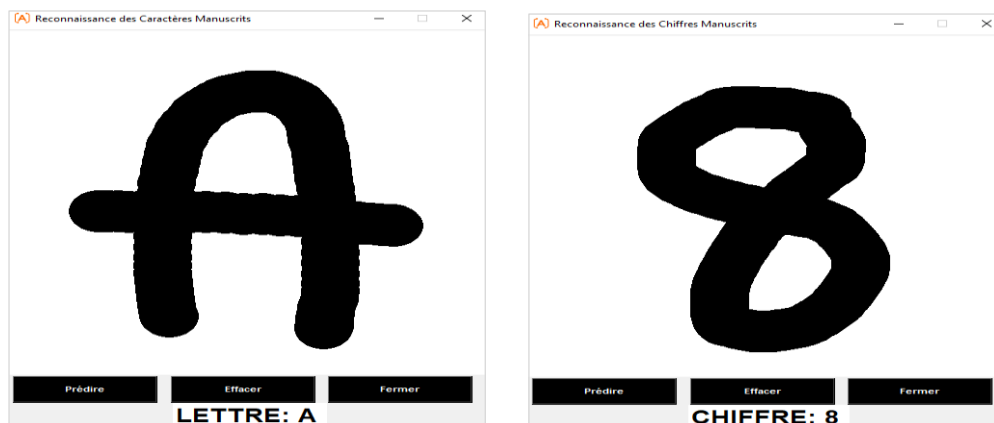




Figure 10: Quelques résultats de reconnaissance Hors-ligne

En ce qui concerne la reconnaissance en temps réel, le système n'arrive pas à identifier correctement toutes les entrées. Cela pourrait être justifié, à notre avis, par la mauvaise qualité du matériel de saisie (capteur) qui est dans notre cas la caméra du PC. La mauvaise qualité du capteur trouble la réception des signaux ce qui entraîne du bruit. Pour surmonter ce problème, une série de prétraitements est nécessaire afin de se débarrasser du bruit de l'acquisition et de ne garder que l'information significative de la forme.

5. Conclusion

Nous avons présenté dans cette annexe les différents aspects du développement de notre système. Nous avons décrit précisément l'implémentation de notre système et avons montré quelques captures d'écrans de notre application de reconnaissance des caractères manuscrits.