

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique  
Université Ahmed Draia - Adrar  
Faculté des Sciences et de la Technologie  
Département des Mathématiques et Informatique



Mémoire de fin d'étude, en vue de l'obtention du diplôme de master en  
informatique

**Option : Systèmes Intelligents**

**Reconnaissance de caractères arabe manuscrits par  
réseau de neurones convolutifs**

**Préparé par :**

- Yousfi Aissa
- Boussaid Abdelkarim

**Encadré par :**

- **Dr. MAMOUNI Elmamoun**

**Année Universitaire 2021/2022**

الجمهورية الجزائرية الديمقراطية الشعبية  
People's Democratic Republic of Algeria

Ministry of Higher Education and  
Scientific Research  
University Ahmed Draia of Adrar  
The central library



وزارة التعليم العالي و البحث العلمي  
جامعة أحمد دراية- أدرار  
المكتبة المركزية  
مصلحة البحث البليوغرافي

## شهادة الترخيص بالإيداع

انا الأستاذ : ماموني المأمون  
المشرف على مذكرة الماستر الموسومة بـ :

Reconnaissance de caractères arabe manuscrits par réseau de neurones convolutifs

من إنجاز الطالب : يوسف عيسى

و الطالب : بوسعيد عبد الكريم

كلية : العلوم والتكنولوجيا

القسم : الرياضيات والاعلام الآلي

التخصص : إعلام آلي

تاريخ تقييم / مناقشة: 2022/06/18

أشهد ان الطلبة قد قاموا بالتعديلات والتصحيحات المطلوبة من طرف لجنة التقييم / المناقشة، وان المطابقة بين  
النسخة الورقية والإلكترونية استوفت جميع شروطها.  
ويامكانهم إيداع النسخ الورقية (02) والايكترونية (PDF).

- امضاء المشرف:

ادرار في : 2022/07/11

مساعد رئيس القسم:



## *Remerciement*

- ❖ Louange à Dieu et cela suffit, et que la prière et la paix soient sur le Prophète, l'Élu, quant à ce qui suit
- ❖ Nous remercions Dieu Tout-Puissant qui, avec le succès de la prévention et sa grâce, nous a permis d'accomplir cette note
- ❖ Nous adressons nos remerciements particuliers à notre professeur :  
**Mr. MAMOUNI Elmamoun,**
  - ❖ Pour les instructions, les directions et les notes qu'il nous a données, ainsi que pour sa patience tout au long de sa supervision de ce mémorandum.
- ❖ Nous adressons également nos sincères remerciements à tous les professeurs et collègues qui nous ont apporté leur aide, quelle qu'en soit la nature, et à tous ceux qui nous ont apporté leurs encouragements, quel que soit leur diplôme.
- ❖ Nous adressons également nos sincères remerciements à tous nos honorables professeurs du Département de mathématiques et d'information informatique, spécialisé dans les systèmes intelligents, université Ahmed Deraya, Adrar, pour tout ce qu'ils nous ont fourni pendant la période de notre formation.

## *Dédicaces*

*Je dédie ce mémoire à*

*A mon père, que Dieu ait pitié de lui, à ma  
chère mère, à mes frères et sœurs à qui je dois ce  
que je suis maintenant. Que Dieu vous protège*

*A tous ceux qui m'ont accompagné et soutenu  
durant ce travail*

**Aissa**

**[yosfiaissa107@gmail.com](mailto:yosfiaissa107@gmail.com)**

# Dédicaces

*Je le dédie à*

*Mes chers parents, que Dieu les préserve et les  
perpétue comme une lumière sur mon chemin*

*La généreuse famille qui m'a soutenu et qui est  
toujours frères et sœurs*

*Compagnons de voyage et tous ceux qui ont  
contribué avec nous à cette remise des diplômes.*

**Abdelkarim**

**[boussaidkarim50@gmail.com](mailto:boussaidkarim50@gmail.com)**

### Résumé

Ce présent travail s'inscrit dans le domaine de la reconnaissance de l'écriture manuscrite arabe, ce domaine reste un grand défi et une concurrence entre les chercheurs. L'apprentissage profond est actuellement la technique la plus célèbre dans le domaine de reconnaissance de formes en général.

Dans ce travail nous avons construit un système de reconnaissance de caractères Arabe manuscrite basé sur les réseaux de neurones convolutifs. Nous avons fait varier les différents paramètres du réseau pour obtenir des résultats compétitifs avec ceux de la littérature. Les bases de données MNIST et AHADB ont été utilisé pour évaluer notre système.

Mots clés : reconnaissance, écriture, caractères manuscrite, apprentissage, réseaux de neurones convolutifs.

### Abstract

This present work is in the field of Arabic handwriting recognition, this field remains a great challenge and competition between researchers. Deep learning is currently the most famous technique in the field of pattern recognition in general.

In this work we have built a handwritten Arabic character recognition system based on convolutional neural networks. We varied the different parameters of the network to obtain competitive results with those of the literature. The MNIST and AHADB databases were used to evaluate our system.

Keywords: recognition, writing, handwritten characters, learning, convolutional neural networks

### ملخص

هذا العمل الحالي في مجال التعرف على خط اليد العربية ، ويبقى هذا المجال تحديًا كبيرًا ومنافسة بين الباحثين. يعد التعلم العميق حاليًا أشهر تقنية في مجال التعرف على الأنماط بشكل عام.

في هذا العمل قمنا ببناء نظام التعرف على الحروف العربية المكتوبة بخط اليد على أساس الشبكات العصبية التلافيفية. قمنا بتغيير المعايير المختلفة للشبكة للحصول على نتائج تنافسية مع تلك الموجودة في الأدبيات. تم استخدام قواعد بيانات MNIST و AHADB لتقييم نظامنا.

الكلمات المفتاحية: التعرف ، الكتابة ، الحروف المكتوبة بخط اليد ، التعلم ، الشبكات العصبية التلافيفية

## Sommaire

Remerciement.....	I
Dédicace.....	II
Résumé.....	V
Liste des tableaux.....	VIII
Liste des Figures.....	X
Introduction générale.....	01

### **CHAPITRE 1 Reconnaissance de caractères arabe manuscrits**

1.1 Introduction.....	03
1.2 Caractéristiques de l'écriture arab.....	03
1.3 Types de reconnaissance d'écriture.....	09
1.3.1 Reconnaissance en-ligne.....	09
1.3.2 Reconnaissance hors-ligne.....	09
1.4 Structure d'un système de reconnaissance d'écriture manuscrite.....	11
1.4.1 Acquisition.....	12
1.4.2 Prétraitement.....	13
1.4.3 Segmentation.....	16
1.4.4 Extraction de caractéristiques.....	17
1.4.5 Reconnaissance.....	19
1.4.6 Post-traitement.....	21
1.5 Conclusion.....	22

### **CHAPITRE 2 Réseau de neurones convolutifs**

2.1 Introduction.....	24
2.2 Concepts de réseau de neurones convolutifs.....	24
2.2.1 Couches réseau.....	25
2.2.1.1 Couche convolutive.....	25
2.2.1.2 Couche de regroupement [Pooling Layer].....	29
2.2.1.3 Fonctions d'activation (non linéaires.....	30
2.2.1.4 Couche entièrement connectée (Fully Connected (FC)).....	31
2.2.2 Fonctions de perte.....	31
2.3 Fondation du réseau de neurones convolutifs.....	32
2.4 Processus de formation du réseau de neurones convolutifs.....	33
2.4.1 Prétraitement des données et augmentation des données.....	34

## Sommaire

---

2.4.2	Initialisation des paramètres.....	34
2.4.2.1	Initialisation aléatoire.....	34
2.4.2.2	Initialisation Xavier.....	35
2.4.2.3	Initialisation préformation non supervisée.....	35
2.4.3	Régularisation à CNN.....	36
2.4.4	Sélection de l'optimiseur.....	37
2.5	Progrès récents dans les architectures CNN.....	38
2.5.1	Classification des images.....	38
2.5.2	Détection d'objets.....	40
2.5.3	Segmentation des images.....	40
2.5.3.1	Segmentation sémantique.....	40
2.5.3.2	Segmentation des instances.....	40
2.6	domaines d'application des CNN.....	40
2.6.1	Classification des images.....	41
2.6.2	Reconnaissance de texte.....	41
2.6.3	Reconnaissance des actions.....	41
2.6.4	Génération de légendes d'image.....	41
2.6.5	Analyse des images médicales.....	41
2.6.6	Sécurité et surveillance.....	42
2.6.7	Colorisation automatique de l'image et transfert de style.....	42
2.6.8	Imagerie satellitaire.....	42
2.7	Conclusion.....	42
<b>CHAPITRE 3 résultats expérimentaux et discussions</b>		
3.1	Introduction.....	44
3.2	environnement de développement .....	44
3.3	Aperçu de l'environnement de programmation.....	46
3.3.1	Python.....	44
A.	TensorFlow .....	45
B.	Keras.....	45
3.4	Les Outils de travail.....	46
3.4.1	Navigateur Anaconda.....	46
3.5	Base des bas de données.....	47
3.5.1.	(MNIST) .....	47

## Sommaire

---

3.5.2. (AHCD).....	48
3.6 Résultats et discussion.....	49
3.7 Conclusion générale.....	54
Bibliographies.....	55

## Liste des tableaux

---

### Liste des tableaux

<b>Tableau 1.1</b>	L'alphabet arabe dans ses différentes formes.....	04
<b>Tableau 1.2</b>	Positions des Hamza et Madd avec Alif, Waw et Ya.....	04
<b>Tableau 1.3</b>	Les caractères additionnels.....	05
<b>Tableau 1.4</b>	Positions des Hamza et Madda avec Lam-Alif.....	05
<b>Tableau 1.5</b>	Lettres arabes ayant des points diacritiques.....	08
<b>Tableau 1.6</b>	Systèmes de reconnaissances d'écriture arabe.....	22

### Liste de figures

#### CHAPITRE 1

<b>Figure 1.1</b>	Sens d'écriture de la langue arabe.....	05
<b>Figure 1.2</b>	Ligne de base dans une phrase arabe.....	06
<b>Figure 1.3</b>	Exemple de mot arabe avec trois composantes connexes.....	06
<b>Figure 1.4</b>	Formes des boucles dans les caractères arabes.....	06
<b>Figure 1.5</b>	Différents styles calligraphiques.....	07
<b>Figure 1.6</b>	Chevauchements et ligature dans une phrase arabe.....	07
<b>Figure 1.7</b>	Forme de lettres ne pouvant être rattachées aux lettres qui les suivent..	08
<b>Figure 1.8</b>	Etendu des espaces intra et inter-mots.....	08
<b>Figure 1.9</b>	Outils d'acquisition en-ligne.....	09
<b>Figure 1.10</b>	Schéma général d'un système de reconnaissance d'écriture manuscrite.....	11
<b>Figure 1.11</b>	Communication écriture Homme-machine.....	12
<b>Figure 1.12</b>	Application de certaines opérations de prétraitement sur un mot arabe.	13
<b>Figure 1.13</b>	Exemple de binarisation d'un chèque algérien.....	14
<b>Figure 1.14</b>	Exemple de lissage d'un chèque algérien.....	15
<b>Figure 1.15</b>	Normalisation de deux mots arabes " " قسنطينة et "[ 38ام البواقي].....	15
<b>Figure 1.16</b>	Exemples de Segmentations.....	16
<b>Figure 1.17</b>	Caractéristiques structurelles dans un mot arabe.....	17

#### CHAPITRE 2

<b>Figure 2.1</b>	Modèle conceptuel de CNN.....	25
<b>Figure 2.2</b>	Exemple d'un noyau (kernel) 2 _ 2.....	26
<b>Figure 2.3</b>	exemple d'une image RVB.....	26

## Liste de figures

---

<b>Figure 2.4</b>	Une image en niveaux de gris 4 _ 4.....	26
<b>Figure 2.5</b>	Un noyau de taille 2 _ 2.....	26
<b>Figure 2.6</b>	Illustrant les 5 premières étapes de l'opération de convolution.....	27
<b>Figure 2.7</b>	La carte des caractéristiques final après l'opération de convolution complète.....	27
<b>Figure 2.8</b>	Les calculs sont effectués à chaque étape.....	28
<b>Figure 2.9</b>	Quelques étapes initiales ainsi que le produit final du processus d'assemblage max.....	29
<b>Figure 2.10</b>	L'architecture des couches entièrement connectées.....	30
<b>Figure 2.11</b>	Clarifier l'interconnexion entre les couches d'un neutron, Kunihiko Fukushima.....	32
<b>Figure 2.12</b>	Le bloc de base du réseau résiduel.....	35
<b>Figure 2.13</b>	Ajuster les hypothèses du modèle avec une classification binaire par rapport à elle-même.....	36
<b>Figure 2.14</b>	L'effet de la valeur différente du taux d'apprentissage (LR) sur le processus de formation.....	37
<b>Figure 2.15</b>	Structure de LeNet-5[25].....	38
<b>Figure 2.16</b>	Structure de AlexNet[26].....	38
<b>Figure 2.17</b>	Structure de VGGNet[28].....	38
<b>Figure 2.18</b>	Structure de ResNet[29].....	39

## CHAPITRE 3

<b>Figure 3.1</b>	Langages de programmation les plus populaires.....	45
<b>Figure 3.2</b>	La structure de base de Keras.....	46
<b>Figure 3.3</b>	Interface du navigateur Navigator.....	47
<b>Figure 3.4</b>	interface Jupyter Note.....	47

## Liste de figures

---

<b>Figure 3.5</b>	image de la base de données (AHCD).....	48
<b>Figure 3.6</b>	Architecture Define Model .....	49
<b>Figure 3.7</b>	fonction Define Model ().....	49
<b>Figure 3.8</b>	Résultat de l'application MNIST .....	50
<b>Figure 3.9</b>	Architecture du modèle 1.....	51
<b>Figure 3.10</b>	fonction Define Model () 1.....	51
<b>Figure 3.11</b>	Résultat Model () 1.....	51
<b>Figure 3.12</b>	Architecture du modèle 2.....	52
<b>Figure 3.13</b>	fonction Define Model () 2.....	52
<b>Figure 3.14</b>	Résultat Model () 2.....	52

### Introduction générale

La reconnaissance de l'écriture manuscrite est l'un des domaines clés pour quiconque a besoin de saisir des données dans un ordinateur. Son histoire remonte à plus de trente ans. L'objectif des chercheurs est de trouver des algorithmes capables de résoudre des problèmes informatiques. Aujourd'hui, il existe de nombreux domaines où la reconnaissance de l'écriture manuscrite est très attendue, par exemple dans le tri automatique du courrier, le traitement automatique des dossiers administratifs, les formulaires d'enquête ou l'enregistrement des chèques bancaires.

Les réseaux de neurones sont actuellement l'un des outils les plus efficaces pour traiter les problèmes de classification. Dans notre cas, il s'agit de la classification des lettres arabes manuscrites. La reconnaissance de l'écriture manuscrite comprend deux types de systèmes qui se distinguent par la nature du support d'écriture auquel ils sont confrontés :

Les systèmes de reconnaissance de l'écriture manuscrite les plus couramment utilisés dans l'étude sont appelés systèmes hors ligne. Après numérisation, le support informatique des données écrites est une image. Pour que les documents classiques soient traités automatiquement.

Le deuxième mode d'acquisition concerne les systèmes de reconnaissance d'écriture manuscrite qualifiés de systèmes en ligne. L'écriture pour ces systèmes est généralement reconnue au moment où l'écrivain écrit ; L'acquisition se fait alors à l'aide d'un stylet et d'une tablette électronique, pouvant être absorbée directement sur un écran (ex : e-paper).

Par conséquent, notre travail vise à reconnaître les caractères arabes manuscrites.

- Le premier chapitre de ce manuscrit contient une description des caractéristiques des lettres arabes avec une introduction aux deux types de reconnaissance, en ligne et hors ligne

- Le deuxième chapitre est consacré à l'étude du réseau de neurones convolutionnels (CNN). Dans ce chapitre, nous fournissons une meilleure compréhension de CNN et facilitons les activités de recherche futures et les développements appliqués dans le domaine de CNN.

Dans le troisième chapitre, nous présenterons l'analyse des résultats expérimentaux et des discussions utilisées en Python sur l'environnement Anaconda Navigator , en utilisant keras et tensorflow.

## Chapitre 1

# Reconnaissance de caractères arabe manuscrits

## 1.1 Introduction

La reconnaissance automatique de l'écriture manuscrite présente un grand intérêt dans l'accomplissement des tâches fastidieuses comme celles que l'on rencontre dans certains domaines tels que le tri postal, la lecture de chèques bancaires, la lecture des bordereaux, des bons de commande et des feuilles de déclaration. Ce chapitre sera consacré à la description du domaine de la reconnaissance d'écriture en mettant le point sur l'écriture arabe. Premièrement, on va commencer par présenter les différentes caractéristiques de l'écriture arabe qui la distingue des autres scripts. Ensuite, on va donner les différents types de reconnaissance d'écriture existants. Et finalement on va introduire la structure générale d'un système de reconnaissance d'écriture manuscrite hors-ligne en détaillant chacune des étapes les constituants à savoir le prétraitement, la segmentation, l'extraction de caractéristiques, la reconnaissance et le post-traitement. Ce chapitre sera clôturé par une conclusion portant sur les difficultés rencontrées lors de la reconnaissance de l'écriture arabe manuscrite.

## 1.2 Caractéristiques de l'écriture arabe

L'arabe est écrit par plus de cent millions de gens, dans plus de vingt pays différents. L'écriture arabe a été développée à partir d'un type d'Araméen. La langue araméenne comporte moins de consonants que l'arabe, alors des nouvelles lettres ont été créé en ajoutant des points aux lettres déjà existantes. D'autres petites marques appelées diacritiques sont utilisées pour indiquer des courtes voyelles, mais elles ne sont généralement pas utilisées.[16]

L'écriture arabe se caractérise par:

- ❖ L'arabe est une écriture consonantique qui utilise un alphabet de 28 lettres. Selon sa position dans le mot (en tête, à l'intérieur, à la fin ou isolé), un même caractère arabe peut avoir jusqu'à quatre formes différentes, ce qui multiplie le nombre de modèles (tableau 1.1) auquel il faut ajouter la Hamza « ء » qui est le plus souvent considérée comme signe complémentaire [3]. Elle peut s'écrire seule ou sur le support de trois voyelles (alif, waw et ya) (tableau 1.2) de plus l'alphabet arabe comprend d'autres caractères additionnels tels que « ؤ » et « ڤ » de ce fait, certains auteurs considèrent que l'alphabet arabe comprend plutôt 31 lettres que 29 (tableau 1.3)
- La considération du symbole « ~ » qui s'écrit uniquement sur le support du caractère « ل »; fait apparaître d'autres graphismes (tableaux 1.2 et 1.4).

Au début	Au milieu	Fin	Isolé	Au début	Au milieu	Fin	Isolé
أ		آ	أ	ظ	ظ	ض	ض
ب	ب	ب	ب	ط	ط	ظ	ظ
ت	ت	ت	ت	ظ	ظ	ظ	ظ
ث	ث	ث	ث	ع	ع	ع	ع
ج	ج	ج	ج	غ	غ	غ	غ
ح	ح	ح	ح	ف	ف	ف	ف
خ	خ	خ	خ	ق	ق	ق	ق
د		د	د	ك	ك	ك	ك
ذ		ذ	ذ	ل	ل	ل	ل
ر		ر	ر	م	م	م	م
ز		ز	ز	ن	ن	ن	ن
س	س	س	س	ه	ه	ه	ه
ش	ش	ش	ش			و	و
ص	ص	ص	ص	ي	ي	ي	ي

Tableau 1.1 L'alphabet arabe dans ses différentes formes.

caractère	initiale	médiane	finale	isolé
Alif+~			آ	آ
Alif+ء			أ	أ
			ب	ب
Waw+ء			و	و
Ya+ء	ي	ي	ي	ي

Tableau 1.2 Positions des Hamza et Madd avec Alif, Waw et Ya.

caractère	initiale	médiane	finale	isolé
<b>Ta marbouta</b>			ة	ة
<b>Lamalif</b>			لا	لا

Tableau 1.3 Les caractères additionnels.

caractère	initiale	médiane	finale	isolé
<b>Lamalif +<sup>-</sup></b>			لا	لا
<b>Lamalif +ء</b>			لا	لا
			لا	لا

Tableau 1.4 Positions des Hamza et Madda avec Lam-Alif.

- ❖ L'écriture arabe est cursive, c'est-à-dire que les lettres sont liées généralement entre elles qu'elles soient imprimées ou manuscrites. Et le sens d'écriture va de droite à gauche comme montre le mot d'une wilaya algérienne présenté dans la figure 1.1



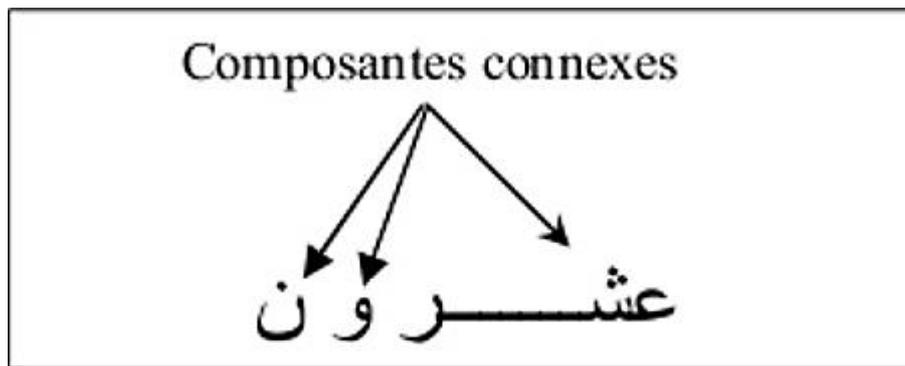
Figure 1.1 Sens d'écriture de la langue arabe.

- ❖ Un trait caractéristique de l'écriture arabe est la présence d'une ligne de base horizontale dite encore ligne de référence ou d'écriture [4]. C'est la partie la plus dense (figure 1.2).



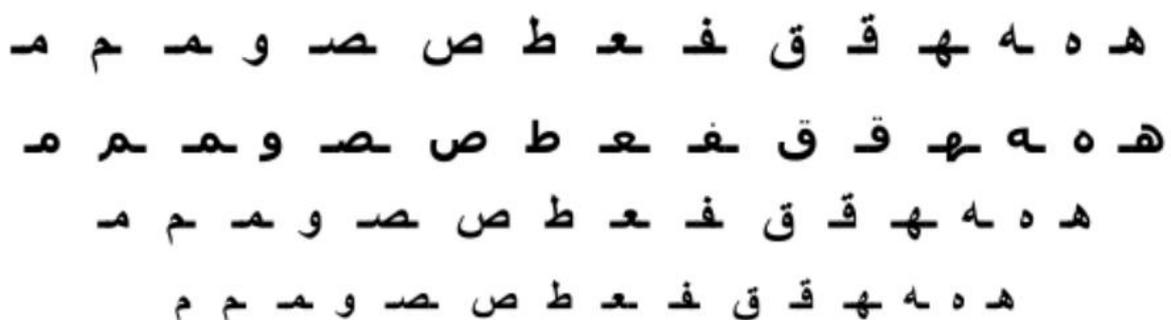
**Figure 1.2** Ligne de base dans une phrase arabe.

- ❖ Un caractère arabe peut contenir un trait vertical (TAA ( ط )) un trait oblique (KAF ( ك )) ou un zigzag (ALIF ( أ )).
- ❖ Un mot arabe se compose généralement d'une ou plusieurs composantes connexes appelées également sous-mots, chacune contient un ou plusieurs caractères, la figure 1.3 présente un mot composé de 3 composantes connexes [11]



**Figure 1.3** Exemple de mot arabe avec trois composantes connexes.

- ❖ Certains caractères arabes incluent une boucle qui peut avoir différentes formes (figure 1.4).



**Figure 1.4** Formes des boucles dans les caractères arabes.

- ❖ Comme l'écriture arabe est une écriture calligraphique. La calligraphie est l'art de la belle écriture; c'est le fait de former d'une façon élégante et ornée les caractères d'une écriture.

Pour l'arabe, six styles graphiques différents sont souvent utilisés : Tholoth, Naska, Requeh, Dewani, Farci, et Koufique. La figure 1.5 montre trois styles différents pour écrire la même phrase

<< بسم الله الرحمن الرحيم >> .



Figure 1.5 Différents styles calligraphiques.

- ❖ Les dimensions des caractères (chasse et hauteur) sont variables, même s'il s'agit des différentes formes d'un caractère.
- ❖ Plusieurs caractères peuvent se combiner verticalement pour former une ligature [10]. Les chevauchements quand à eux représentent des recouvrements des caractères par d'autres caractères voisins. La figure 1.6 présente un exemple de phrase comportant une ligature et trois chevauchements.

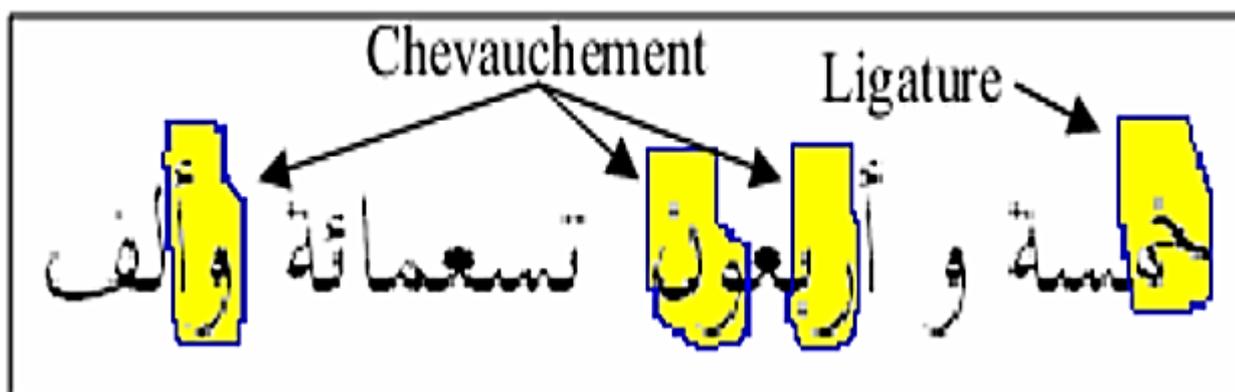


Figure 1.6 Chevauchements et ligature dans une phrase arabe [10].

- ❖ Certains caractères arabes incluent dans leur forme des points qui peuvent être au nombre de 1, 2 ou 3.

- ❖ Ces points peuvent se situer au dessus ou en dessous du corps du caractère, mais jamais en haut et en bas simultanément [5] (tableau 1.5).

Position \ Nombre de pointes	Au-dessus	Au-dessous
Un point diacritique	ذ خ ز ض ف غ ن ظ	ب ج
Deux points diacritiques	ت ق	ي
Trois points diacritiques	ث ش	

Tableau 1.5 Lettres arabes ayant des points diacritiques.

- ❖ Certaines formes de lettres que nous avons regroupées dans la figure 1.7, ne peuvent dans aucun cas être rattachées à la lettre suivante, ce qui fait qu'un mot unique peut être entrecoupé d'un ou plusieurs espaces, lesquels sont aussi utilisés pour séparer les mots. La longueur de cet espace inter-mot est généralement supérieure à l'espace intra-mot entre caractères nonrattachés comme montré à la figure 1.8

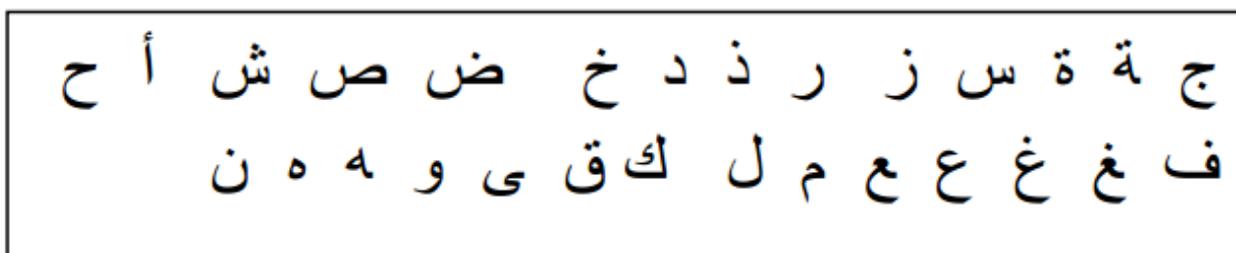


Figure 1.7 Forme de lettres ne pouvant être rattachées aux lettres qui les suivent.

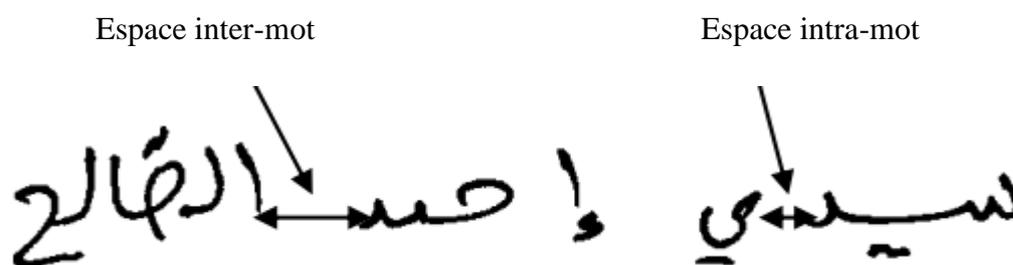


Figure 1.8 Etendu des espaces intra et inter-mots.

### 1.3 Types de reconnaissance d'écriture

La reconnaissance de l'écriture manuscrite est un traitement informatique qui a pour but de traduire un texte écrit en un texte codé numériquement. Selon la manière d'acquisition du tracé on peut distinguer l'écriture hors-ligne et en-ligne

#### 1.3.1 Reconnaissance en-ligne

Dans les applications de reconnaissance d'écriture manuscrite en-ligne, l'utilisateur écrit sur une table spéciale, le système va reconnaître l'écriture et envoyer le résultat à l'ordinateur. Ces systèmes sont utilisés dans plusieurs équipements électroniques comme PDA, Pocket PC ou Tablet-PC. Il y a pas mal d'avantages de reconnaissance d'écriture manuscrite en-ligne, par rapport à celle hors-ligne. Par exemple, car l'utilisateur écrit sur une table spéciale, il y a moins de bruit. De plus, on peut déterminer comment un caractère est écrit, c'est à dire, l'ordre de traits constituant ce caractère. D'ailleurs, la contrainte du temps de reconnaissance n'est pas stricte, on peut utiliser des algorithmes complexes. C'est pour quoi le taux de reconnaissance de ces systèmes est assez élevé (figure 1.9).



**Figure 1.9** Outils d'acquisition en-ligne.

#### 1.3.2 Reconnaissance hors-ligne

La reconnaissance hors-ligne ou en différé démarre après l'acquisition du document en entier. Ce type de reconnaissance convient pour les documents imprimés et les manuscrits déjà rédigés. Ce mode permet l'acquisition directe d'un nombre important de caractères, mais impose d'effectuer des prétraitements coûteux pour retrouver l'ordre de la lecture. L'approche n'est pas la même selon qu'il s'agisse de reconnaître un imprimé ou un manuscrit

- ❖ Dans le cas de l'imprimé, les caractères sont bien classés et souvent bien séparés verticalement, ce qui simplifie la phase de lecture, bien que certaines fontes présentent parfois des accolements qu'il faut défaire. De plus, le graphisme des caractères est conforme à un style calligraphique (fonte) qui constitue un modèle pour l'identification. Dans ce type, on trouve plusieurs types, un système est dit mono fonte s'il ne traite qu'une fonte à la fois, c'est-à-dire que le système ne connaît l'alphabet que dans une seule fonte. L'apprentissage y est simple puisque l'alphabet représenté est réduit. Un système est dit multi fonte s'il est capable de reconnaître un mélange de quelques fontes parmi un ensemble de fontes préalablement apprises. Dans ce cas, le prétraitement doit réduire les écarts entre les caractères (taille, épaisseur et inclinaison), l'apprentissage doit gérer les ambiguïtés dues aux éventuelles ressemblance des caractères des différentes fontes, et la reconnaissance doit identifier les subtiles différences entre ces caractères. Enfin, un système est dit omni fonte s'il est capable de reconnaître toute fonte sans l'avoir absolument apprise, ce qui relève actuellement du domaine de la recherche.
- ❖ Dans le cas du manuscrit, les caractères sont souvent ligaturés et leur graphisme est inégalement proportionné. Cela nécessite l'emploi de techniques de délimitation très spécifiques et souvent des connaissances contextuelles pour guider la lecture

✓ **La reconnaissance de caractères**

C'est le fait de considérer le caractère comme entité à reconnaître, c'est-à-dire on va utiliser une phase de segmentation. Touj et al utilisent une transformée de Hough généralisée (THG) [36], Un tableau de référence nommé R-table définit la correspondance entre l'espace de définition de la forme recherchée (l'espace image), et l'espace paramétrique. La THG peut être utilisée non seulement pour reconnaître les caractères, mais aussi pour déterminer leurs positions dans l'image grâce à la localisation de leurs points de référence. L'idée est de réaliser une reconnaissance de l'écriture arabe imprimée sans avoir recours à une étape de segmentation (en caractères ou en graphèmes).

✓ **La reconnaissance de mots**

Ici, c'est le mot qui va être reconnu comme un tout sans aucune segmentation. Parmi les systèmes basés sur une reconnaissance on cite ceux de : Menasri [45], Al-Hajj et al [46], et Benouareth et al [47].

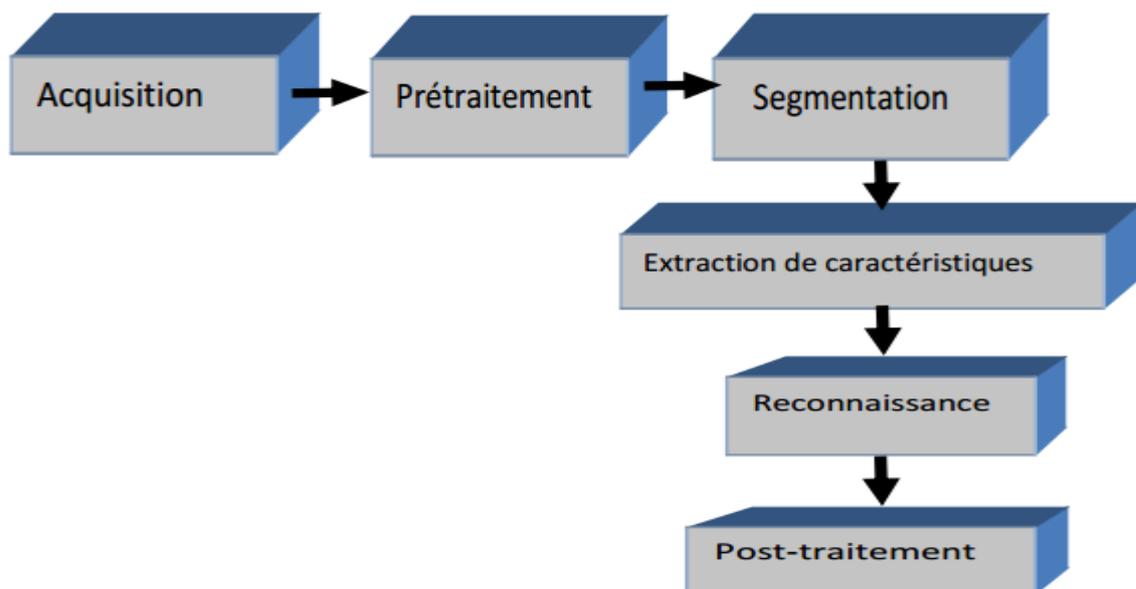
✓ **La reconnaissance de sous-mots**

Ce type de reconnaissance est basé sur l'unité sous-mot qui est composée à son tour d'un ensemble de caractères on cite ceux de Miled et al [12]. Encore à l'aide de Réseau de neurones, AbdulKader décompose le problème en deux parties [1]

- Un premier problème, qui consiste à reconnaître un pseudo-mot à partir des lettres qui le constituent.
- Un deuxième problème, qui consiste à reconnaître un mot à partir des pseudo-mots qui le constituent.

#### 1.4 Structure d'un système de reconnaissance d'écriture manuscrite

Un système de reconnaissance fait appel généralement aux étapes d'acquisition, prétraitements, segmentation, extraction des caractéristiques, reconnaissance, suivis éventuellement d'une phase de post-traitement (figure 1.10).



**Figure 1.10** Schéma général d'un système de reconnaissance d'écriture manuscrite.

### 1.4.1 Acquisition

Dans les systèmes hors lignes, l'acquisition est une phase où l'écriture est digitalisée à l'aide d'un scanner. Le résultat de cette phase représente l'entrée de système. Il dépend de deux paramètres importants Résolution et Niveau d'éclairage. L'acquisition est assez simple mais très importante car elle influence sérieusement les étapes suivantes [4].

- **Résolution** : la résolution normale est 300 dpi. Pourtant, quand la taille de l'écriture est petite, il faut augmenter la résolution.
- **Niveau d'éclairage** : si on ajuste le scanner pour que l'image soit plus claire, le bruit est réduit mais des traits minces disparaissent aussi [11].

Selon le type d'écriture en-ligne ou hors-ligne ; l'acquisition se diffère comme ceci :

#### ❖ Acquisition de l'écriture hors ligne

Dans le contexte de l'écriture hors ligne, les systèmes d'acquisition les plus courants sont essentiellement des scanners ou des caméras linéaires. La résolution de l'image numérisée influence les étapes ultérieures d'un système de lecture automatique. Il est communément admis que la résolution optimale d'une image est fonction de l'épaisseur du trait d'écriture

#### ❖ Acquisition de l'écriture en ligne

Dans le cas de l'acquisition en ligne, les dispositifs d'acquisition les plus répandus sont des tablettes à numériser ou des « papiers électroniques ». L'échantillonnage du tracé délivre une série de coordonnées décrivant la trajectoire du stylet au cours du temps. Les tablettes les plus perfectionnées permettent également d'avoir accès aux informations de pression et d'inclinaison du stylo [11].

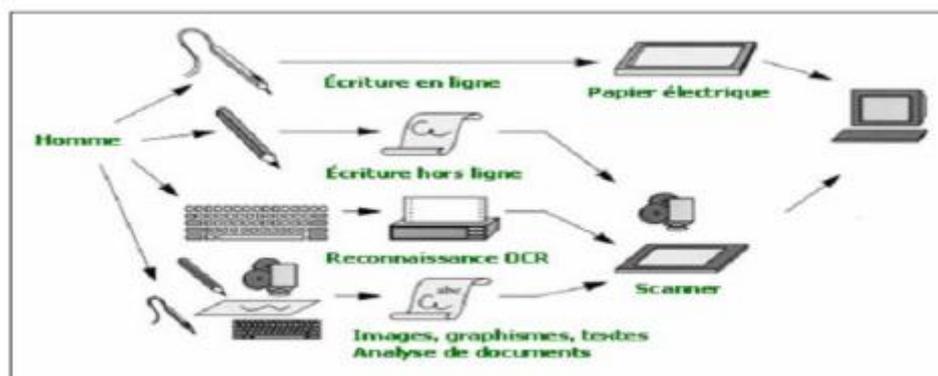


Figure 1.11 Communication écriture Homme-machine.

### 1.4.2 Prétraitement

Le but des prétraitements est la réduction du bruit, de la distorsion, et de la variation des styles pour faciliter les traitements ultérieurs tels que la segmentation et l'extraction de primitives et pour n'en garder que les informations utiles. Ils comprennent : la binarisation, la suppression du bruit (le lissage), la squelettisation et les normalisations (de la taille, de l'inclinaison des lignes, et de l'inclinaison des caractères).



**Figure 1.12** Application de certaines opérations de prétraitement sur un mot arabe [39].

Il est à noter que selon la qualité du document à traiter, le type de l'écriture et la méthode d'analyse adoptée, une ou plusieurs techniques de prétraitement sont utilisées. Mais pas forcément toutes. Dans ce qui suit, on va expliquer le principe des opérations de prétraitement les plus appliquées dans le domaine de la reconnaissance d'écriture arabe manuscrite

- **Binarisation** : les systèmes de reconnaissance nécessitent une étape de binarisation qui vise à séparer les pixels de texte des pixels de l'arrière-plan de l'image traitée. En fait, la plupart des systèmes ne fonctionnent que sur des images binaires. La plus simple façon pour obtenir une image binaire est de choisir une valeur seuil, puis de classer tous les pixels dont les valeurs sont au-dessus de ce seuil comme étant des pixels d'arrière plan, et tous les autres pixels comme étant des pixels de texte. Soit l'image  $I(M \times N)$ , supposons que  $f(x,y)$  représente le niveau de gris du pixel aux coordonnées  $(x,y)$ ,  $0 \leq x \leq M$ ,  $0 \leq y \leq N$  et  $S$  est le seuil choisi, les pixels de l'objet sont ceux ayant le niveau de gris inférieur à  $S$  et les autres ayant le niveau de gris supérieur à  $S$  sont des pixels du fond. Alors, l'image binarisée  $G$  est déterminée par les pixels  $(x,y)$  dont la valeur est donnée par l'équation (1.1) [4] :

$$G(x,y) = \begin{cases} 1 & \text{si } f(x,y) > s \\ 0 & \text{si } f(x,y) \leq s \end{cases}$$

(1.1)

Le problème alors est de savoir comment sélectionner le bon seuil. La plupart des de binarisation des images de texte se basent ces deux types [10] :

- **Seuillage globale** : consiste à prendre un seuil ajustable mais identique pour toute l'image, cette méthode convient pour les documents de contraste fort et de bonne qualité.
- **Seuillage adaptatif** : est plus approprié dans le cas contraire, le calcul du seuil se fait localement en fonction du voisinage du pixel traité. et le seuillage adaptatif

Le but de la binarisation est de faire surgir l'information utile par rapport à l'arrière-plan

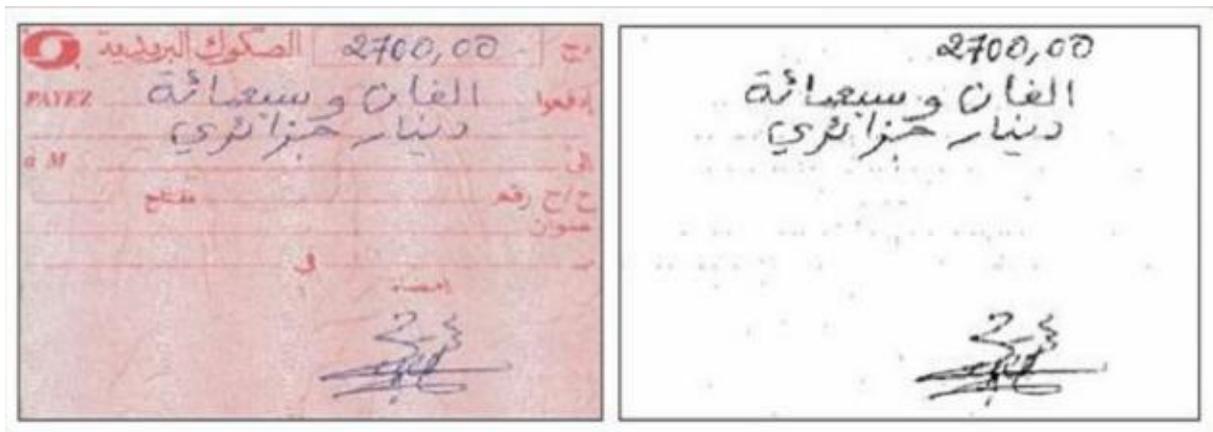
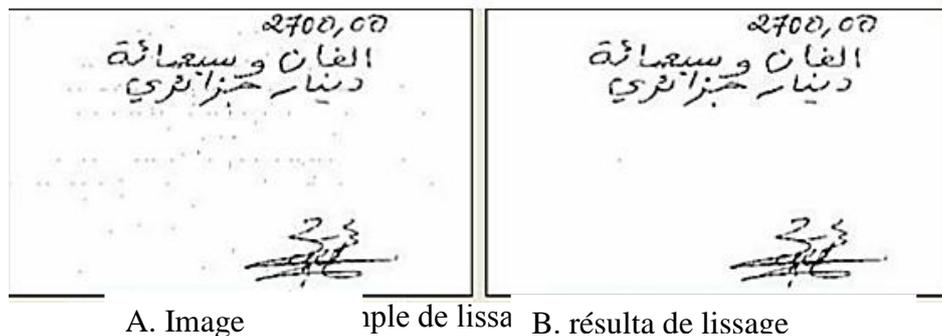


Figure 1.13 Exemple de binarisation d'un chèque algérien.

- **Lissage** : l'image des caractères peut être entachée de bruits dus aux artefacts de l'acquisition et à la qualité du document, conduisant soit à une absence de points ou à une surcharge de points. Les techniques de lissage permettent de résoudre ces problèmes par des opérations locales qu'on appelle opérations de bouchage et de nettoyage [16]. L'opération de nettoyage permet de supprimer les petites tâches et les excroissances de la forme. Pour le bouchage il s'agit d'égaliser les contours et de boucher les trous internes à la forme du caractère en lui ajoutant des points noirs. Plusieurs autres techniques similaires sont utilisées dont la méthode statistique, une méthode basée sur la morphologie mathématique, ...etc, (pour plus de détail sur ces technique le lecteur peut se référer à [11]).



A. Image

exemple de lissage

B. résultat de lissage

- **Normalisation de la taille** : la taille d'un caractère peut varier d'une écriture à l'autre, ce qui peut causer une instabilité des paramètres. Une technique naturelle de prétraitement consiste à ramener les caractères à la même taille [4]. Après la normalisation de la taille, les images de tous les caractères se retrouvent définies dans une matrice de même taille, Pour faciliter les traitements ultérieurs. Cette opération introduit généralement de légères déformations sur les images. Cependant certains traits caractéristiques tels que la hampe dans les caractères (الظ) (par exemple) peuvent être éliminées à la suite de la normalisation, ce qui peut entraîner à des confusions entre certains caractères [40].

Des techniques de normalisation tels que la correction de l'inclinaison des lignes, la correction de l'inclinaison des lettres et la normalisation des caractères tendent à réduire cette variation de styles, tailles et orientations d'écriture pour arriver à une forme plus ou moins standard de données [10].

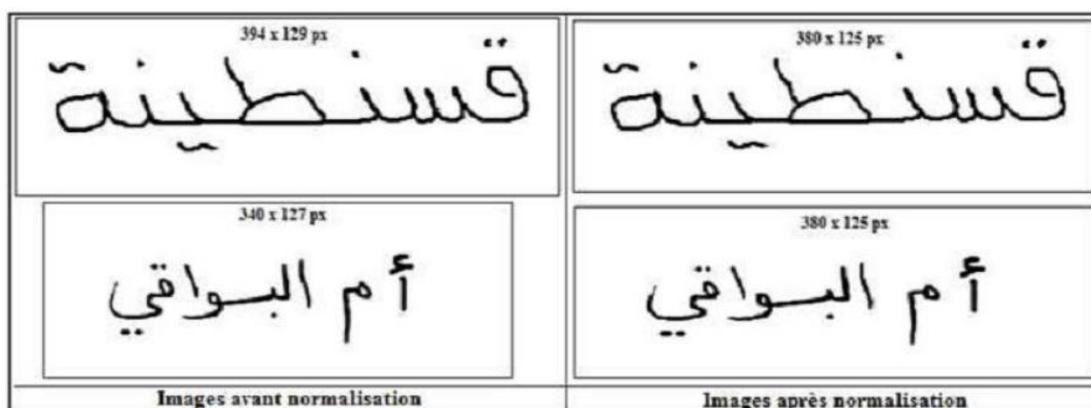
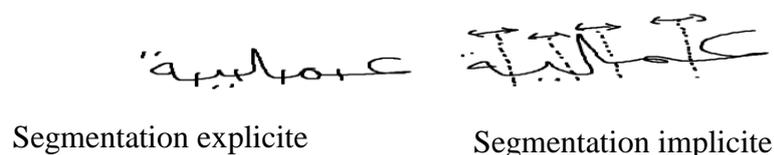


Figure 1.15 Normalisation de deux mots arabes "قسنطينة" et "أم البواقي" [38].

### 1.4.3 Segmentation

Dans cette phase les différentes parties logiques d'une image sont extraites. Dans le cas d'un texte arabe, à partir de l'image acquise il y'a d'abord une séparation des blocs de texte et des blocs graphiques, puis à partir d'un bloc de texte il y'a extraction des lignes, ensuite à partir de ces lignes sont extraits les mots puis les caractères (ou parties du caractère) [5]. Il existe deux techniques permettant la mise en œuvre de la segmentation:

- ❖ **Segmentation implicite** : l'avantage de cette segmentation c'est que l'information est localisée par les modèles des lettres et la validation se fait par ses modèles. Il n'y aura pas d'erreur de segmentation et enfin on contourne le dilemme de Sayre car en connaissant les lettres, on n'engendre pas d'erreur de segmentation. Les défauts de cette segmentation viennent du fait que l'espace de recherche des limites se trouve très augmenté et le problème est ramené à un problème de recherche de zones où se trouvent ces limites, reposant le problème de Sayre sur les zones. Cette segmentation présente des avantages et des inconvénients ce qui fait que cette approche est insuffisante pour une modélisation optimale de l'écriture .segmentation d'après les modèles de lettres (figure 2b).
- ❖ **Segmentation explicite** : l'avantage de cette segmentation c'est que l'information est localisée explicitement et ça va dans le sens de Sayre puisque on sépare les lettres non pas d'après leur reconnaissance, mais d'après des critères topologiques ou morphologiques. Le défaut majeur de cette segmentation vient en premier lieu de choix des limites indépendant des critères des modèles : les limites sous-optimales pour les modèles et modélisation sousoptimale. En deuxième lieu il n'existe pas de méthode de segmentation fiable à 100% et pour toute erreur de segmentation pénalise le système dès la base. La segmentation explicite n'est pas parfaite pour un système de reconnaissance des mots manuscrite .segmentation sur des critères topologiques (figure 2a).



**Figure 1.16** Exemples de Segmentations.

### 1.4.4 Extraction de caractéristiques

C'est l'une des étapes les plus délicates et les plus importantes dans OCRS (Optical Character Recognition System). La reconnaissance de caractère passe d'abord par l'analyse de sa forme et l'extraction de ses traits caractéristiques (primitives) qui seront exploités pour son identification [32]. Les types de caractéristiques peuvent être classés en quatre groupes principaux : caractéristiques structurelles, caractéristiques statistiques, transformations globales, et superposition des modèles et corrélation [5].

❖ **Caractéristiques structurelles** : les caractéristiques structurelles décrivent une forme en termes de sa topologie et sa géométrie en donnant ses propriétés globales et locales [32]. Parmi ces caractéristiques on peut citer :

- Les traits et les anses dans les différentes directions ainsi que leurs tailles.
- Les points terminaux.
- Les points d'intersections.
- Les boucles.
- Le nombre de points diacritiques et leur position par rapport à la ligne de base.
- Les voyellations et les zigzags (hamza).
- La hauteur et la largeur du caractère.
- La catégorie de la forme (partie primaire ou point diacritique, ..., etc).
- Plusieurs autres caractéristiques peuvent être tirées, suivant qu'ils soient extraits d'une courbe, un trait ou un segment de contour.
- 

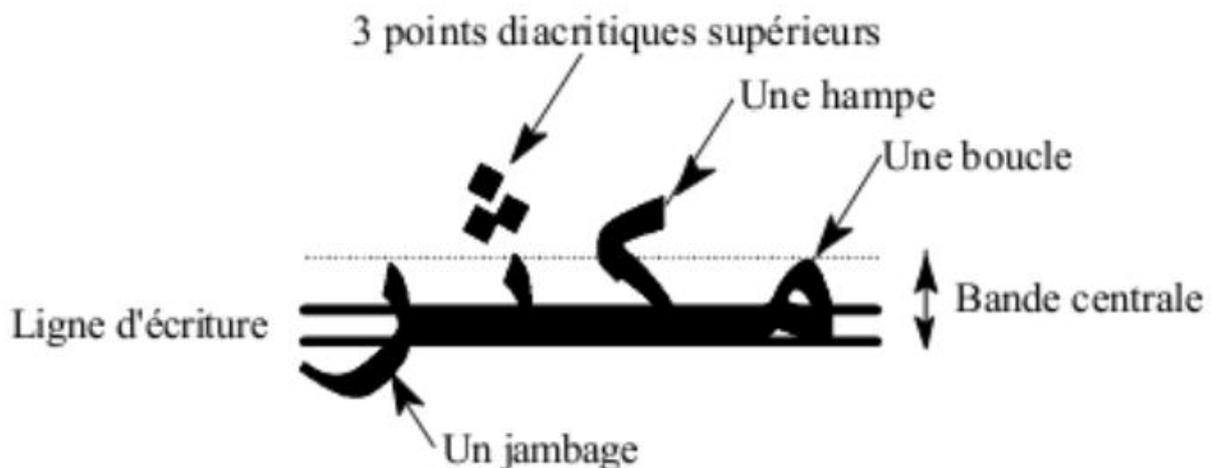


Figure 1.17 Caractéristiques structurelles dans un mot arabe.

- ❖ **Caractéristiques statistiques** : ce type de caractéristiques décrivent une forme en termes d'un ensemble de mesures extraites à partir de cette forme. Les caractéristiques utilisées pour la reconnaissance de textes arabes sont : le zonage (zoning), les caractéristiques de lieu géométrique (Loci : pluriel de mot locus. En phonétique acoustique, zone des fréquences du spectre d'une consonne vers laquelle se dirigent les transitions des formants des voyelles adjacentes) et les moments [14].
- Le zonage consiste à superposer une grille  $n \times m$  sur l'image du caractère et pour chacune des régions résultantes, calculer la moyenne ou le pourcentage de points en niveaux de gris, donnant ainsi un vecteur de taille  $n \times m$  de caractéristiques.
  - La méthode Loci (Venue de l'antiquité, cette méthode très facile fut seulement supplantée au XVIIIème siècle par la méthode du fichier numérique) est basée sur le calcul du nombre de segments blancs et de segments noirs le long d'une ligne verticale traversant la forme, ainsi que leurs longueurs [5].
  - La méthode des moments : les moments d'une forme par rapport à son centre de gravité sont invariants par rapport à la translation et peuvent être invariants par rapport à la rotation [8]. Ils sont aussi indépendants de l'échelle. Ces caractéristiques peuvent être facilement et rapidement extraites d'une image de texte, ils peuvent tolérer modérément les bruits et les variations. Les moments les plus utilisés pour l'écriture arabe sont les moments de Hu et les moments de Zernike.
- ❖ **Caractéristiques globales** : elles sont naturellement basées sur une transformation globale de l'image. La caractéristique d'une primitive globale est de dépendre de la totalité des pixels d'une image [20]. Parmi ces primitives on décrit ici les transformées de Hough et de Fourier.
- **La transformée de Hough** : c'est une méthode d'extraction de primitive largement utilisée dans le domaine de la reconnaissance de forme. Elle est capable d'extraire des lignes droites, des courbes, et n'importe quelle autre forme exprimée sous forme d'une équation de paramètres [4]. Les deux essentielles étapes de cette méthode sont : le mapping des pixels de la forme de l'espace de l'image vers un espace de paramètres (dépend de la caractéristique à extraire), et l'extraction de primitives sur cet espace.
  - **La transformée de Fourier (TF)**: la transformée de Fourier est fréquemment utilisée en traitement d'image lorsque l'on désire effectuer un filtrage un peu délicat qui ne pourrait être réalisé dans le domaine spatial à l'aide d'une simple convolution par un masque [20]. Etant donné  $(x_k, y_k)$ ,  $k = 0, 1, \dots, N-1$  les coordonnées des pixels du contour de la forme. Pour chaque couple  $(x_k, y_k)$  on définit une variable complexe  $U_k$  :
 
$$\mathbf{u}_k = \mathbf{x}_k + \mathbf{j}y_k \quad (1.3)$$

Pour les  $N$   $U_k$  pixels du contour en calcul les descripteurs de Fourier FI comme suit :

$$f_1 = \sum_{k=0}^{N-1} U_k e^{(-j\frac{2\pi}{N})lk}, l = 0, 1, \dots, N-1 \quad (1.4)$$

Le type de descripteur de Fourier le plus utilisé est celui introduit par F. P. Kuhl et C. R. Giardina [39], l'approche est dite descripteur de Fourier elliptique, pour les formes des contours fermés, qui sont approximées par les équations suivantes :

$$\dot{x}(t) = A_0 + \sum_{n=1}^N \left[ a_n \cos \frac{2n\pi t}{T} + b_n \sin \frac{2n\pi t}{T} \right] \quad (1.5)$$

$$\dot{y}(t) = C_0 + \sum_{n=1}^N \left[ c_n \cos \frac{2n\pi t}{T} + d_n \sin \frac{2n\pi t}{T} \right] \quad (1.6)$$

Avec  $T$  est la longueur de contour,  $(t) \equiv x(t)$  et  $(t) \equiv y(t)$  pour  $N \rightarrow \infty$  et :

$$A_0 = \frac{1}{T} \int_0^T x(t) dt \quad (1.7)$$

$$C_0 = \frac{1}{T} \int_0^T y(t) dt \quad (1.8)$$

$$a_n = \frac{2}{T} \int_0^T x(t) \cos \frac{2n\pi t}{T} dt \quad (1.9)$$

$$b_n = \frac{2}{T} \int_0^T x(t) \sin \frac{2n\pi t}{T} dt \quad (1.10)$$

$$c_n = \frac{2}{T} \int_0^T y(t) \cos \frac{2n\pi t}{T} dt \quad (1.11)$$

$$d_n = \frac{2}{T} \int_0^T y(t) \sin \frac{2n\pi t}{T} dt \quad (1.12)$$

Les coefficients  $a_n$ ,  $b_n$ ,  $c_n$  et  $d_n$  peuvent être utilisés comme primitives.

### 1.4.5 Reconnaissance

Cette étape consiste à déterminer la classe d'appartenance de la forme considérée en utilisant un classifieur. Les classifieurs les plus utilisés dans la reconnaissance d'écriture sont les MMCs (Modèles de Markov Cachés) [8], les SVMs (Machine à Vecteur de Support) [6], les K-ppv (K-plus proche voisins) [1], les réseaux bayésiens [13], et les RNA (Réseau de Neurone Artificiel). Ce dernier type fera l'objet de notre deuxième chapitre.

L'étape de la reconnaissance englobe deux phases : l'apprentissage et la décision ou aussi la classification.

**Apprentissage** : dans le cas d'apprentissage il s'agit en fait de fournir au système un ensemble de formes qui sont déjà connues (on connaît la classe de chacune d'elles). C'est cet ensemble d'apprentissage qui va permettre de « régler » le système de reconnaissance de façon à ce qu'il soit capable de reconnaître ultérieurement des formes de classes inconnues [30]. Il existe deux types d'apprentissage :

- **Supervisé** : l'apprentissage est dit supervisé si les différentes familles des formes sont connues a priori et si la tâche d'apprentissage est guidée par un superviseur ou professeur, c'est-à-dire le concepteur, qui indique, pour chaque forme de l'échantillon rentrée, le nom de la famille qui la contient.
- **Non-supervisé** : l'apprentissage non-supervisé ou sans professeur, consiste à faire une classification automatique. Il s'agit, à partir d'échantillon de référence et de règles de regroupement, de construire automatiquement les classes ou les modèles sans intervention de l'opérateur. Ce mode d'apprentissage nécessite un nombre élevé d'échantillons et des règles de construction précises et non-contradictoires pour bien assurer la formation des classes. Il évite l'assistance d'un opérateur mais n'assure pas toujours une classification correspondante à la réalité (celle de l'utilisateur) [39].
- **Classification** : à partir du vecteur descriptif d'une entrée, le module de reconnaissance cherche parmi les classes d'apprentissage connues, celles qui sont les plus proches. La classification peut conduire à un succès si la réponse est unique (un seul modèle répond à la description de la forme), et à une confusion si la réponse est multiple (plusieurs modèles correspondent à la description). Elle peut aussi conduire à un rejet de la forme s'il n'existe aucun modèle correspond à sa description. Dans le cas où la classification est un succès ou une confusion, la décision peut être accompagnée d'une mesure de vraisemblance, appelée aussi score ou taux de reconnaissance. Les équations suivantes montrent respectivement comment calculer les trois mesures de performance d'un OCR ; le taux de reconnaissance (TL), le taux d'erreur (TE) et le taux de rejet (TR).

(1.13)

$$TL = \frac{\text{Nombre d'entités bien reconnues}}{\text{Nombre total d'entités présentées au classifieur}}$$

$$TE = \frac{\text{Nombre d'entités mal-reconnues}}{\text{Nombre total d'entités présentées au classifieur}}$$

(1.14)

$$TR = \frac{\text{Nombre d'entités rejetées}}{\text{Nombre total d'entités présentées au classifieur}}$$

(1.15)

Le tableau 1.6 présente quelques systèmes pertinents de la littérature. Pour chaque système on donne le classifieur et la base de test utilisée ainsi que le taux de reconnaissance.

Système	Classifieur	Base utilisée	Taux de reconnaissance
Saeed et al [41]	Classification par distance euclidienne	IFN / ENIT	92%
Sabri et al [40]	KNN	Modified Fourier Spectrum (MFS)	95.9%
Lawgali et al [34]	Réseaux de neurones	Leur propre base	-
Elglaly et Quek [23]	K-plus proches voisins	Leur propre	-

	et Réseaux de neurones	base	
Abdulrahman et Muhamad [2]	Algorithme génétique	IFN/ENIT	-
Sahloul et Suen [42]	Les réseaux de neurones	CENPARMI	100% pour la plupart des lettres à 88% en moyenne

**Tableau 1.6** Systèmes de reconnaissances d'écriture arabe.

**1.4.6 Post-traitement** Le post-traitement est effectué quand le processus de reconnaissance aboutit à la génération d'une liste de mots possibles, éventuellement classés par ordre décroissant de vraisemblance, afin de prendre compte des nombreuses relations d'ordre : logique, lexical et syntaxique entre les différents composants d'un document [2]. On peut dire que le rôle de cette étape consiste à corriger les erreurs de l'étape de reconnaissance, en tenant compte de tous les types d'informations propres au document :

- **L'information lexicale** : le contrôle lexical vérifie la présence d'un mot dans un lexique de référence. Dans notre cas la reconnaissance sera effectuée au niveau du mot, donc le mot présent dans le lexique sera attribué par une valeur de confiance qui dépend de celle attribuée par l'étape de classification.
- **L'information syntaxique** : l'utilisation d'un ensemble de règles grammaticales permet de prendre en compte le contexte syntaxique d'un document pour confirmer ou infirmer la

séquence de mots proposés. C'est le fait de mettre en corrélation des résultats obtenus sur des champs différents de l'image, mais codant la même information ; c'est le cas des chèques où on trouve une zone pour les montants numériques et une autre zone pour les montants littérales.

## **1.5 Conclusion**

La reconnaissance de l'écriture arabe manuscrite est une tâche délicate à cause de différentes difficultés rencontrées liées à plusieurs facteurs. Le premier facteur réside dans les caractéristiques de l'écriture arabe qui compliquent un peu le processus de la segmentation tels que la nature cursive de l'écriture arabe. Un autre problème lié à la variabilité inter-scripteur et intra-scripteur désignant le fait qu'un même mot arabe peut s'écrire de différentes manières par le même scripteur ou par des scripteurs différents. Nous avons présenté dans ce chapitre, les principales caractéristiques de l'écriture arabe, les types de reconnaissance d'écriture ainsi que les différentes étapes constituant un système de reconnaissance d'écriture manuscrite hors-ligne

## Chapitre 2

# Réseau de neurones convolutifs

## 2.1 Introduction

Parmi les différentes architectures d'apprentissage en profondeur [17], un type particulier de réseau neuronal multicouche pour les données spatiales est un réseau neuronal convolutif (ou CNN ou ConvNet.).

L'architecture du CNN s'inspire de la perception visuelle des êtres vivants.

Bien qu'il soit devenu populaire après les performances de référence d'AlexNet [18] en 2012, il a en fait commencé en 1980.

Après 2012, CNN a pris de la vitesse pour prendre en charge divers domaines de la vision par ordinateur et du langage naturel, traitement et bien d'autres.

Après le succès d'AlexNet, CNN a été très populaire dans trois domaines principaux de la classification d'images [19], de la détection d'objets [20] et de la segmentation [21], et de nombreux modèles CNN avancés ont été proposés dans ces domaines au cours des années successives. on fonctionne très bien avec de nombreuses applications, sauf qu'ils ne savent pas encore pourquoi et comment cela fonctionne fondamentalement.

Par conséquent, des efforts supplémentaires sont nécessaires pour étudier les principes de base des CNN.

La classe fournira une meilleure compréhension de CNN et facilitera les futures activités de recherche et les développements appliqués dans le domaine de CNN.

## 2.2 Concepts de réseau de neurones convolutifs

Le réseau de neurones convolutifs (CNN), également appelé ConvNet, est un type de réseau de neurones artificiels (ANN), qui possède une architecture approfondie et une capacité étonnante à généraliser par rapport à d'autres réseaux avec des couches FC, et peut apprendre des caractéristiques très abstraites d'objets, en particulier les données spatiales, Il peut être identifié plus efficacement.

Un modèle CNN profond consiste en des couches de traitement qui peuvent apprendre différentes propriétés des données d'entrée (par exemple, une image) avec plusieurs niveaux d'abstraction. les couches initiales apprennent et extraient des fonctionnalités de haut niveau (avec moins d'abstraction), et les couches plus profondes apprennent et extraient des fonctionnalités de bas niveau (avec une abstraction plus élevée). Le modèle conceptuel de base de CNN est présenté à la **figure 2.1**, les différents types de couches illustrés dans les sections suivantes :

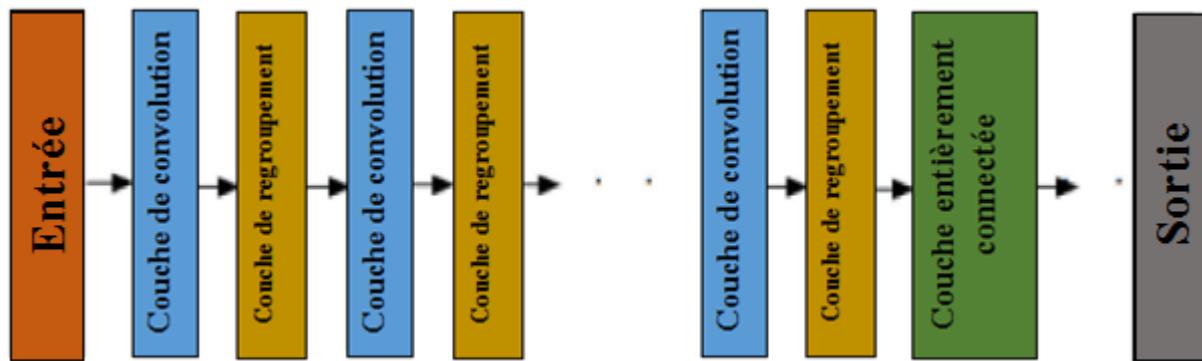


Figure 2.1 Modèle conceptuel de CNN[19].

### - L'importance des réseaux de neurones convolutifs

- L'une des principales raisons d'envisager un CNN dans un tel cas est la fonction de partage de poids du CNN, qui réduit le nombre de paramètres pouvant être formés dans le réseau, ce qui aide le modèle à éviter la surestimation ainsi qu'à mieux généraliser.

- Dans CNN, la couche de classification et les couches d'extraction de caractéristiques apprennent ensemble, ce qui rend la sortie du modèle plus ordonnée et rend la sortie plus dépendante des caractéristiques extraites.

- La création d'un grand réseau est plus difficile en utilisant d'autres types de neurones qu'en utilisant des réseaux de neurones convolutifs de nos jours, CNN est devenu un mécanisme permettant d'obtenir des résultats prometteurs dans diverses applications basées sur la vision par ordinateur telles que la classification d'images, la détection d'objets, la détection de visages, la reconnaissance vocale, la reconnaissance de véhicules, la reconnaissance d'expressions faciales, la reconnaissance de texte et bien d'autres.

Maintenant, décrivez brièvement les différents composants ou éléments de base d'un CNN comme suit :

#### 2.2.1 Couches réseau

Comme mentionné précédemment, un CNN se compose de plusieurs blocs de construction (appelés couches d'architecture), dans cette sous-section, nous décrivons en détail certains de ces blocs de construction avec leur rôle dans l'architecture.

##### 2.2.1.1 Couche convolutive

La couche convolutive 1 est le composant le plus important de toute architecture CNN.

Il contient un ensemble de noyaux convolutionnels, qui sont transformés avec l'image d'entrée pour créer une carte de caractéristiques de sortie. un noyau peut être décrit comme un réseau de valeurs ou de nombres discrets, où chaque valeur est connue comme le poids de ce noyau. lorsque le processus de formation du modèle CNN est lancé, tous les poids du noyau se voient

attribuer des nombres aléatoires. ensuite, à chaque étape d'apprentissage, les poids sont modifiés et le noyau apprend à extraire des caractéristiques significatives. Dans la **figure 2.2**, nous avons démontré un filtre bidimensionnel.

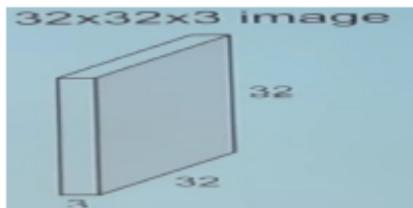
0	1
-1	2

**Figure 2.2** Exemple d'un noyau (kernel) 2 \_ 2.

### - Opération de convolution

Avant d'aller plus loin, comprenons d'abord le format d'entrée CNN.

Contrairement à d'autres réseaux de neurones classiques (où l'entrée est au format vectoriel), l'entrée d'un CNN est une image multicanal (par exemple, pour une image RVB comme sur la **figure 2.3**, il s'agit de 3 canaux et pour une image en niveaux de gris, il est 1 canal). maintenant, pour comprendre le processus de convolution, si nous prenons une image en niveaux de gris de 4\_4 diamètre, comme le montre la **figure 2.4** et une image de 2\_2 noyaux avec des poids initialisés de manière aléatoire, comme le montre la **figure 2.4** **Figure 2.5**.



**Figure 2.3** exemple d'une image RVB.

1	0	-2	1
-1	0	1	2
0	2	1	0
1	0	0	1

**Figure 2.4** Une image en niveaux de gris 4 \_ 4.

0	1
-1	2

**Figure 2.5** Un noyau de taille 2 \_ 2.

Maintenant, dans le processus de convolution, nous prenons le noyau 2\_2 et le déplaçons sur toute l'image 4\_4 horizontalement, verticalement et en cours de route, nous prenons le produit ponctuel entre le noyau et l'image d'entrée en multipliant leurs valeurs correspondantes et en additionnant toutes les valeurs pour générer une valeur d'échelle dans la carte d'entités en sortie.

Ce processus se poursuit jusqu'à ce que le cœur ne puisse plus glisser. pour mieux comprendre cela, effectuons quelques calculs préliminaires qui sont effectués graphiquement à chaque étape, comme indiqué sur la **figure 2.6**, où le noyau 2\_2 (indiqué en bleu clair) est multiplié

par la même région de volume (indiquée en jaune) dans l'image d'entrée 4\_4 et les valeurs résultantes sont additionnées pour obtenir l'entrée correspondante (indiquée en bleu foncé) dans la sortie de la carte à chaque étape de convolution.

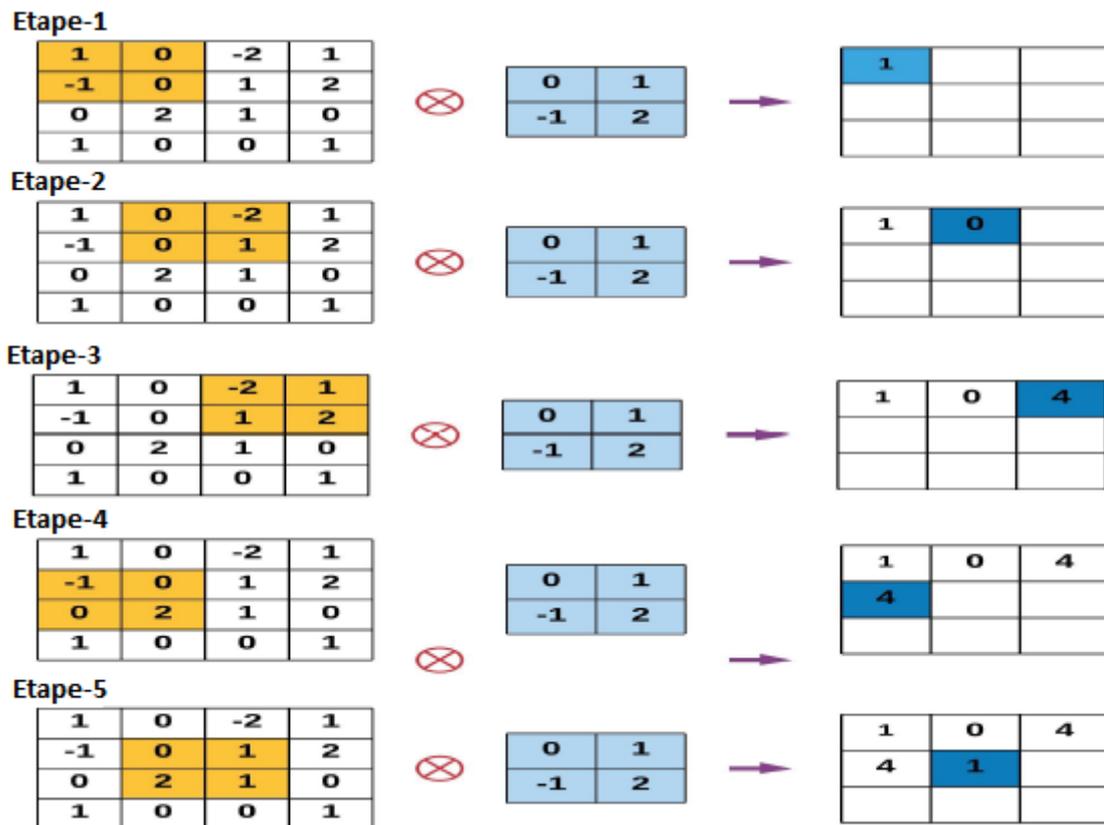


Figure 2.6 Illustrant les 5 premières étapes de l'opération de convolution.

Après avoir effectué l'opération de convolution complète, la carte de caractéristiques de sortie final est affichée dans la Figure 2.7 comme suit.

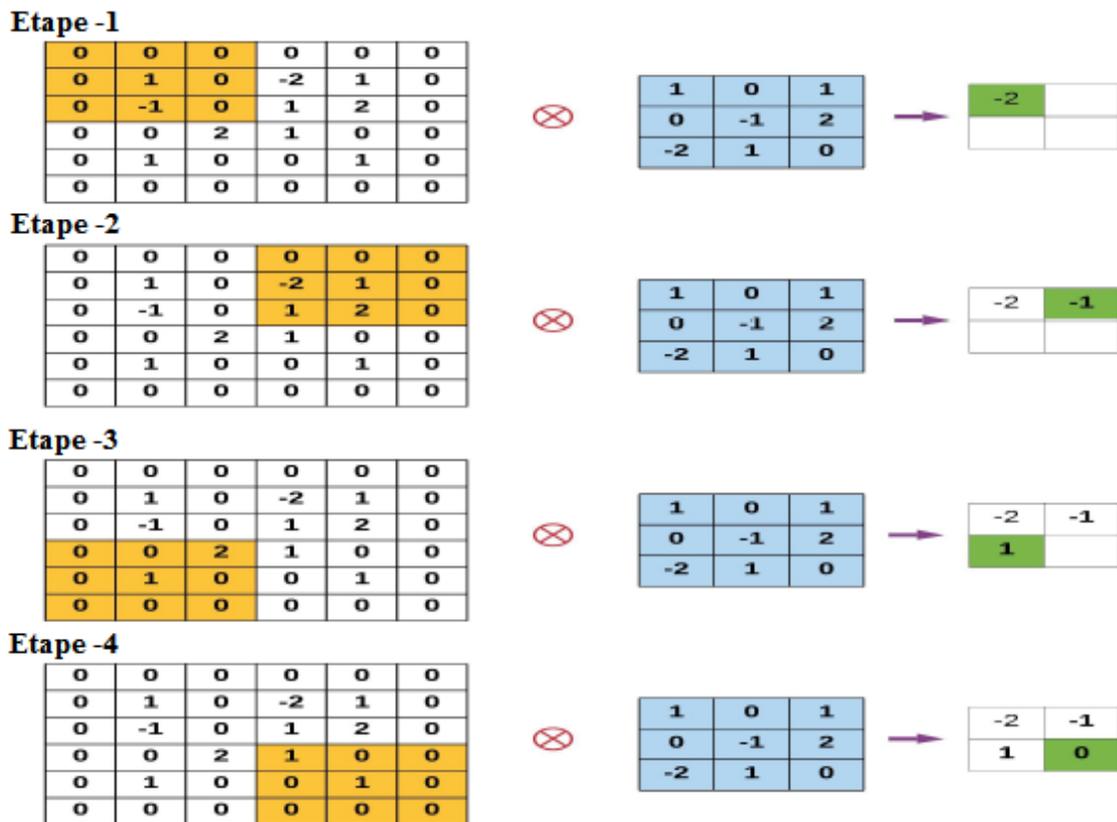
1	0	4
4	1	1
1	1	2

Figure 2.7 La carte des caractéristiques final après l'opération de convolution complète.

Dans l'exemple ci-dessus, nous appliquons l'opération de convolution sans remplissage à l'image d'entrée et avec un pas (c'est-à-dire la taille du pas prise le long de la position horizontale ou verticale) de 1 au centre.

Mais nous pouvons utiliser une autre valeur de pas (au lieu de 1) dans le processus de convolution. La chose notable est que si nous augmentons les étapes du processus de convolution, cela se traduira par une carte de caractéristiques avec des dimensions inférieures.

le remplissage est important pour donner plus d'importance aux informations de taille de bordure de l'image d'entrée, sinon sans remplissage, les fonctionnalités côté bordure sont supprimées très rapidement. le rembourrage est également utilisé pour augmenter la taille de l'image d'entrée, de sorte que la taille de la carte d'entités en sortie est également augmentée. La **figure 2.8** donne un exemple montrant le processus de convolution avec une valeur de remplissage nulle et 3 étapes.



**Figure 2.8** Les calculs sont effectués à chaque étape.

### 2.2.1.2 Couche de regroupement (Pooling Layer)

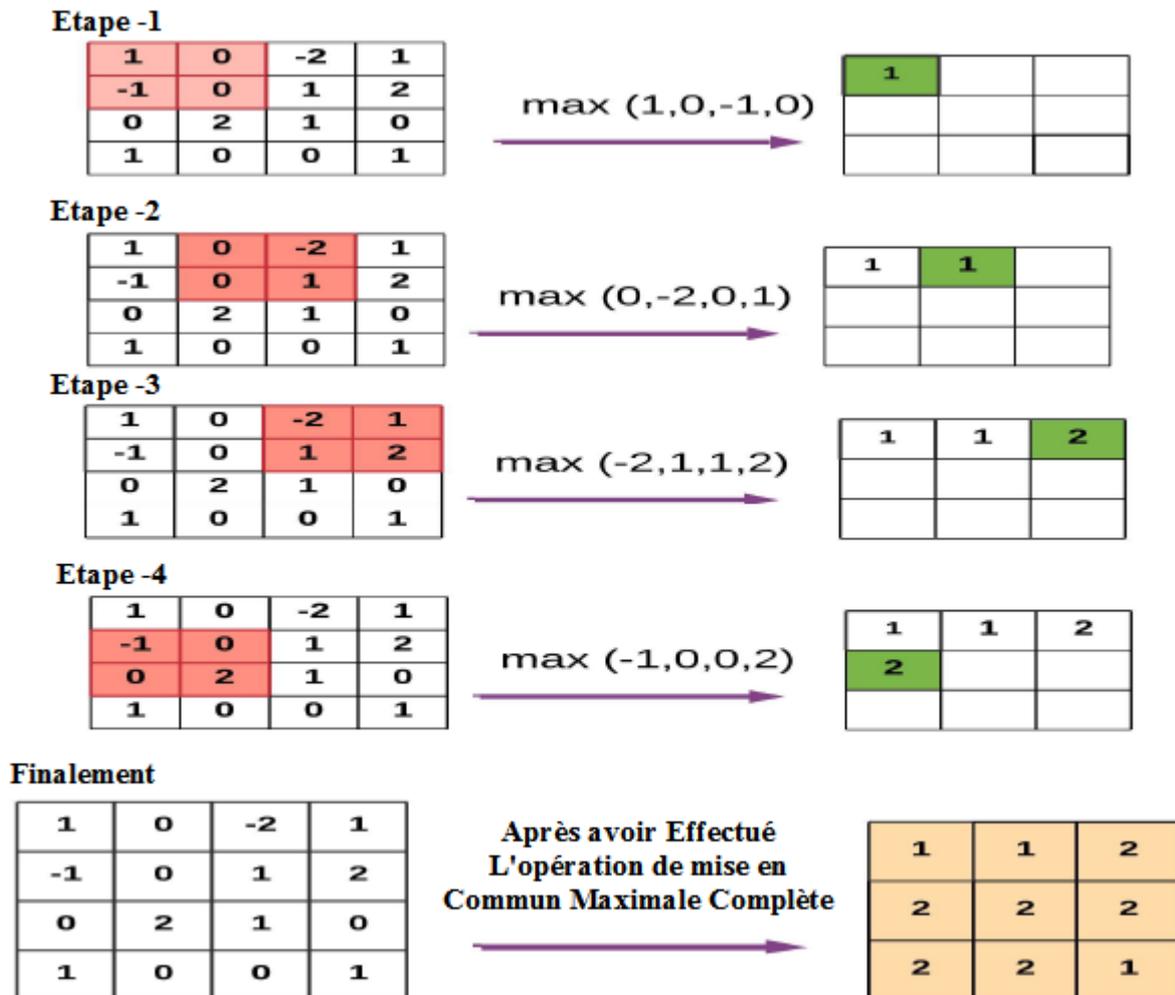
Les couches Pooling sont utilisées pour réduire les cartes d'entités (qui sont produites après les opérations de convolution), c'est-à-dire qu'elles prennent des cartes d'entités plus grandes et les réduisent à des entités plus petites.

Tout en réduisant les cartes de fonctionnalités, il conserve les fonctionnalités (ou informations) les plus courantes à chaque étape du pool.

Le processus d'assemblage est effectué en définissant la taille de la zone de bloc et l'étape du processus, similaire au processus de convolution. Il existe différents types de techniques de mise en commun utilisées dans différentes couches de mise en commun, telles que la mise en commun maximale, la mise en commun minimale, la mise en commun moyenne, la mise en commun fermée, les d-trees, etc. le regroupement maximal est la méthode de regroupement la

plus courante et la plus largement utilisée. Le principal inconvénient de la couche de mutualisation est qu'elle réduit parfois les performances globales du CNN.

La raison derrière cela est que la couche de regroupement aide CNN à déterminer si une caractéristique particulière est présente ou non dans l'image d'entrée donnée sans se soucier de la position correcte de cette caractéristique.



**Figure 2.9** Quelques étapes initiales ainsi que le produit final du processus d'assemblage max.

### 2.2.1.3 Fonctions d'activation (non linéaires)

La tâche principale de toute fonction d'activation dans tout modèle basé sur un réseau de neurones est de mapper les entrées aux sorties, la valeur d'entrée étant obtenue en calculant la somme pondérée des entrées des neurones et en y ajoutant un biais (s'il y a un biais). En d'autres termes, la fonction d'activation détermine si un neurone se déclenchera ou non pour une entrée donnée en produisant la sortie correspondante.

Dans une architecture CNN, des couches d'activation non linéaires sont utilisées après chaque couche apprenable (couches avec poids, c'est-à-dire couches convolutionnelles et couches FC).

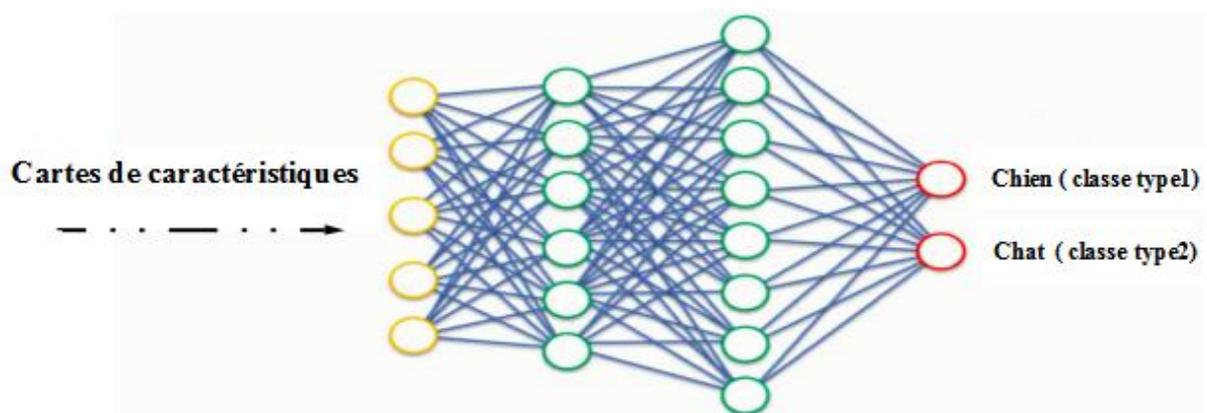
Le comportement non linéaire de ces couches permet au modèle CNN d'apprendre des choses plus complexes et de mapper avec succès les entrées aux sorties de manière non linéaire.

Une caractéristique importante de la fonction d'activation est qu'elle doit être différentiable pour permettre la rétropropagation des erreurs pour former le modèle.

Les fonctions d'activation les plus fréquemment utilisées dans les réseaux de neurones profonds sont décrites.

#### 2.2.1.4 Couche entièrement connectée (Fully Connected (FC))

La dernière partie (ou couches) de chaque architecture CNN (utilisée pour la classification) se compose généralement de couches entièrement connectées, où chaque neurone d'une couche est connecté à chaque neurone de sa couche précédente. la dernière des couches entièrement connectées est utilisée comme couche de sortie (classificateur) de l'architecture CNN. Les couches entièrement connectées sont un type de réseau de neurones artificiels à action directe (ANN) qui suit le principe d'un réseau de neurones multicouche traditionnel (MLP). les couches FC prennent l'entrée de la couche d'agrégation convolutive ou finale, qui se présente sous la forme d'un ensemble de métriques (cartes de caractéristiques) et ces métriques sont prises en compte pour créer un vecteur, puis ce vecteur est introduit dans la couche FC pour générer la sortie CNN finale, comme illustré à la **Figure 2.10**.



**Figure 2.10** L'architecture des couches entièrement connectées.

#### 2.2.2 Fonctions de perte

Nous décrivons différents types de couches utilisées dans une architecture CNN. nous savons maintenant que la dernière couche de chaque architecture CNN (basée sur la classification) est la couche de sortie, où la classification finale a lieu. dans cette couche de sortie, nous calculons l'erreur de prédiction générée par le modèle CNN dans les échantillons d'apprentissage à l'aide de la fonction de perte.

Cette erreur de prédiction indique au réseau comment prédire à partir de la sortie réelle, puis cette erreur sera optimisée pendant le processus de formation du modèle CNN.

La fonction de perte utilise deux paramètres pour calculer l'erreur, le premier paramètre est la sortie estimée du modèle CNN (également appelée prédiction) et le second est la sortie réelle (également appelée étiquette). Il existe différents types de fonctions de perte utilisées dans différents types de problèmes, notamment les suivants :

- ✓ **Fonction Cross-Entropy ou Soft-Max Loss.**
- ✓ **Fonction de perte euclidienne.**
- ✓ **Fonction de perte de charnière.**

### 2.3 Fondation du réseau de neurones convolutifs

En 1959, deux neurophysiologistes David Hubel et Torsten Wiesel ont expérimenté et publié plus tard leur article, intitulé "Champs récepteurs de neurones uniques dans le cortex strié du chat"[22], décrivant que les neurones à l'intérieur du cerveau d'un chat sont organisés en couches.

Ces couches apprennent à reconnaître les modèles visuels en extrayant d'abord les caractéristiques locales, puis en combinant les caractéristiques extraites pour une représentation de niveau supérieur. Plus tard, ce concept est essentiellement devenu l'un des principes fondamentaux du Deep Learning. Inspiré par les travaux de Hubel et Wiesel, en 1980, Kunihiko Fukushima a proposé Neocognitron[23], qui est un réseau de neurones auto-organisé, contenant plusieurs couches, capable de reconnaître les modèles visuels de manière hiérarchique par l'apprentissage et cette architecture est devenue le premier modèle théorique de CNN. comme dans la **figure 2.11**. une amélioration majeure par rapport à l'architecture de Neocognitron a été réalisée par LeCun et. en 1989 en développant un cadre moderne de CNN, appelé LeNet-5, qui a reconnu avec succès l'ensemble de données de chiffres manuscrits MNIST.

LeNet-5 a été formé à l'aide d'un algorithme de rétro propagation d'erreurs et il peut reconnaître des modèles visuels directement à partir d'images d'entrée brutes, sans utiliser de mécanisme d'ingénierie de caractéristiques séparées.

Après avoir découvert LeNet-5, en raison de plusieurs limitations telles que le manque de données de formation volumineuses, le manque d'innovation dans l'algorithme et une puissance de calcul insuffisante, CNN n'a pas bien performé dans divers problèmes complexes.

Mais de nos jours, à l'ère du Big Data, nous avons de grands ensembles de données étiquetés, des algorithmes plus innovants et des machines GPU particulièrement puissantes.

Avec ce type de mise à niveau, en 2012, Krizhevsky et al. a conçu AlexNet, qui a atteint une précision fantastique lors du défi de reconnaissance visuelle à grande échelle ImageNet (ILSVRC[24]).

La victoire d'AlexNet a ouvert la voie à l'invention de plusieurs modèles CNN[19] ainsi qu'à l'application de ces modèles dans divers domaines de la vision par ordinateur et du traitement du langage naturel.

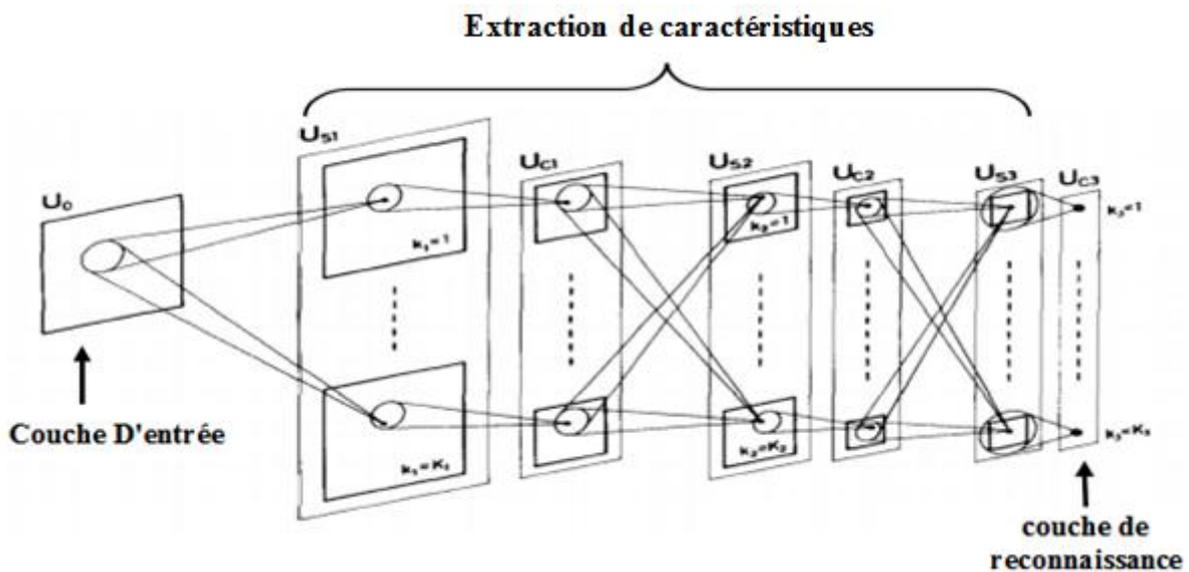


Figure 2.11 Clarifier l'interconnexion entre les couches d'un neurone, Kunihiko Fukushima.

## 2.4 Processus de formation du réseau de neurones convolutifs

Dans la section 3 précédente, nous avons décrit les concepts de base du réseau de neurones convolutifs (CNN) ainsi que les différents composants clés de l'architecture CNN. ici, dans cette section, nous essayons discuter du processus de formation ou d'apprentissage d'un modèle CNN avec certaines directives afin de réduire le temps de formation requis et améliorer la précision du modèle. le processus de formation principalement comprend les étapes suivantes:

- Prétraitement des données et augmentation des données.
- Initialisation des paramètres.
- Régularisation de CNN.
- Sélection de l'optimiseur.

Ces étapes sont décrites dans les sous-sections suivantes.

### 2.4.1 Prétraitement des données et augmentation des données

Le prétraitement des données fait référence à certaines transformations artificielles de l'ensemble de données brutes (y compris ensembles de données de formation, de validation et de test) afin de rendre l'ensemble de données plus propre, plus fonctionnel complète, plus facile à apprendre et dans un format uniforme. le prétraitement des données est effectué avant l'alimentation les données au modèle CNN. Dans un réseau neuronal convolutif, il est un fait que la performance de CNN est directement proportionnel à la quantité de données utilisées pour l'entraîner, c'est-à-dire un bon prétraitement, augmente toujours la précision du modèle. mais d'un autre côté, un mauvais prétraitement peut aussi diminue les performances du modèle. Les techniques générales de prétraitement les plus utilisées sont données dans la sous-section suivante.

- ✓ **Moyenne-soustraction (Centrage zéro)**

- ✓ **Normalisation**

### 2.4.2 Initialisation des paramètres

Un CNN profond se compose de millions ou de milliards de paramètres. donc, il doit être bien initialisé au début du processus d'entraînement, car l'initialisation du poids détermine directement la vitesse le modèle CNN convergerait et avec quelle précision il pourrait finir. Ici, dans cette section, nous discuter de certaines techniques d'initialisation de paramètres les plus utilisées dans CNN comme suit:

La façon la plus simple de le faire est d'initialiser tous les poids avec zéro. cependant, ceci s'avère être une erreur, car si nous initialisons les poids de toutes les couches à zéro, la sortie comme ainsi que les gradients (lors de la rétro propagation) calculés par chaque neurone du réseau être le même. par conséquent, la mise à jour de tous les poids serait également la même. en conséquence, il y a aucune disparité entre les neurones et le réseau n'apprendra aucune fonctionnalité utile. pour casser ça disparité entre les neurones, nous n'initialisons pas tous les poids avec la même valeur, plutôt que, nous utilisons différentes techniques pour initialiser aléatoirement les poids comme suit:

#### 2.4.2.1 Initialisation aléatoire

Comme son nom l'indique, nous initialisons ici les poids (appartenant à la fois aux poids convolutifs et couches FC) de manière aléatoire à l'aide de matrices aléatoires, où les éléments de ces matrices sont échantillonnés à partir d'une distribution avec un petit écart type (par exemple, 0,1 et 0,01) et avec une moyenne nulle. mais le problème clé de l'initialisation aléatoire est qu'elle peut potentiellement conduire à la disparition problèmes de gradients ou de gradients explosifs. Quelques méthodes d'initialisation aléatoire populaires sont:

- Initialisation aléatoire gaussienne : Ici, les poids sont initialisés de manière aléatoire à l'aide de matrices aléatoires, où les éléments de ces matrices sont échantillonnés à partir d'une distribution gaussienne.
- Initialisation aléatoire uniforme : Ici, les poids sont initialisés de manière aléatoire à l'aide de matrices uniformes aléatoires, où les éléments de ces matrices sont échantillonnés à partir d'une distribution uniforme.
- Initialisation aléatoire orthogonale : Ici, les poids sont initialisés de manière aléatoire à l'aide de matrices orthogonales aléatoires, où les éléments de ces matrices sont échantillonnés à partir d'une distribution orthogonale.

#### 2.4.2.2 Initialisation Xavier

Cette technique est proposée par Xavier Glorot et Yoshua Bengio en 2010, elle tente de rendre la variance des connexions de sortie et des connexions d'entrée doit être égale pour chaque couche du réseau. l'idée principale est d'équilibrer la variance des fonctions d'activation. cette technique est tournée s'est avéré très utile à cette époque et puisque Xavier Glorot et Yoshua Bengio ont conçu ce technique pour la fonction d'activation sigmoïde logistique, elle ne fonctionne donc pas bien avec ReLU (principalement utilisé dans l'architecture CNN de nos jours) fonction d'activation. Plus tard, l'initialisation de He technique proposée par Kaiming He et al est utilisée pour travailler avec l'activation ReLU sur le même idée.

#### 2.4.2.3 Initialisation préformation non supervisée

Ici, dans cette technique, nous initialisons un réseau de neurones convolutifs avec un autre réseau convolutif réseau de neurones (qui a été formé à l'aide d'une technique non supervisée), par exemple, un auto-encodeur ou un réseau de croyances profondes. Cette méthode peut parfois très bien fonctionner en aidant à gérer à la fois les problèmes d'optimisation et d'overfitting.

#### - Problème de gradient de disparition

Au cours de la rétro propagation sur un réseau de neurones à convolution profonde (CNN avec de nombreux Lay-ers par exemple, 1000 ), nous devons calculer les gradients de perte (Erreur) par rapport à poids dans les neurones de chaque couche pour mettre à jour ces poids. donc, ici, nous avons utilisé la dérivée opération pour effectuer cette tâche et cela conduit les gradients à devenir de plus en plus petits à mesure que nous continuons à reculer dans le réseau. de ce fait, les neurones de la couche antérieure recevoir de très petits gradients (parfois les gradients peuvent devenir presque nuls), par conséquent, les poids de la couche précédente

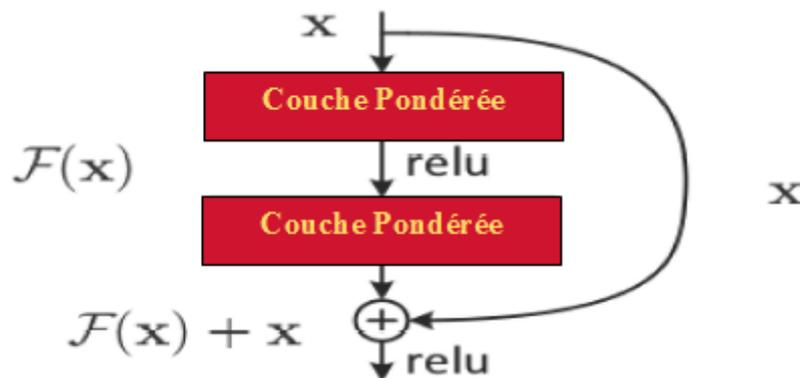
sont mis à jour très légèrement et ces couches apprennent très lentement et inefficent. Ce problème est connu sous le nom de problème de gradient de fuite.

#### - Quelques solutions

1. Nous pouvons utiliser ReLU/fuite ReLU au lieu d'autres (tanh, sigmoïde) comme activation fonction dans l'architecture CNN, qui aide à éviter ce problème.
2. Les réseaux résiduels offrent une autre solution à ce type de problème.

La **figure 2.12** montre bloc de base du réseau Résiduel.

3. Les couches de normalisation par lots peuvent également résoudre le problème.



**Figure 2.12** Le bloc de base du réseau résiduel.

#### - Problème de gradient explosif

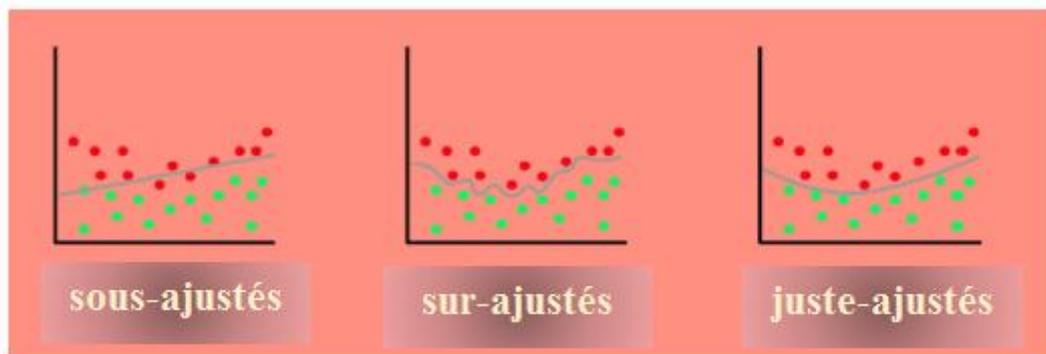
C'est l'exact opposé du problème de gradient de fuite, ici de grands gradients d'erreur accumuler pendant la rétro-propagation et cela a entraîné de très grandes mises à jour des poids du réseau et rendre le modèle instable et ensuite ce modèle instable ne peut pas apprendre efficacement. L'explosion de la mise à jour des poids se produit par la croissance exponentielle du gradient par ré-tourbé les gradients à travers les couches du réseau lors de la rétro propagation lorsque les gradients reculent dans le réseau. à l'extrême, les valeurs des poids (après mise à jour avec de grands gradients) peut devenir si grand qu'il déborde et entraîne Valeurs NaN. quelques solutions:

- Nous pouvons utiliser différentes techniques de régularisation des poids pour éviter ce problème.
- Nous pouvons reconcevoir l'architecture du modèle de réseau pour résoudre le problème.

### 2.4.3 Régularisation de CNN

Le principal défi des algorithmes d'apprentissage en profondeur est de s'adapter correctement aux nouveaux ou inédits. entrée, tirée de la même distribution que les données d'entraînement, la possibilité de le faire est appelée généralisation. Le principal problème pour qu'un modèle

CNN obtienne une bonne généralisation est le sur-ajustement lorsqu'un modèle fonctionne exceptionnellement bien sur les données d'apprentissage mais qu'il échoue sur les données de test (invisible données), alors ce type de modèle est appelé sur-ajusté. le contraire est un modèle sous-équipé, qui se produit lorsque le modèle n'a pas suffisamment appris des données d'entraînement et lorsque le modèle fonctionne bien sur les données d'entraînement et de test, alors ces types de modèles sont appelés juste ajustés maquette. La **Figure 2.13** essaie de montrer les exemples de modèles sur-ajustés[Over-fitting], sous-ajustés[Under-fitting], et juste-ajustés[Just-fitting].



**Figure 2.13** Ajuster les hypothèses du modèle avec une classification binaire par rapport à elle-même.

La régularisation aide à éviter le sur-ajustement en utilisant plusieurs idées intuitives, dont certaines sont discuté dans les prochaines sous-sections.

#### - Les avantages de l'utilisation de la normalisation par lots sont indiqués ci-dessous

- ☒ Il évite également le problème du gradient de fuite.
- ☒ Il peut gérer très efficacement une mauvaise initialisation du poids.
- ☒ Il améliore considérablement le temps de convergence du réseau (il devient très utile en cas de ensemble de données à grande échelle).
- ☒ Il essaie de réduire la dépendance de la formation sur les hyper-paramètres.
- ☒ Il réduit les risques de sur ajustement car il a un léger effet de régularisation.

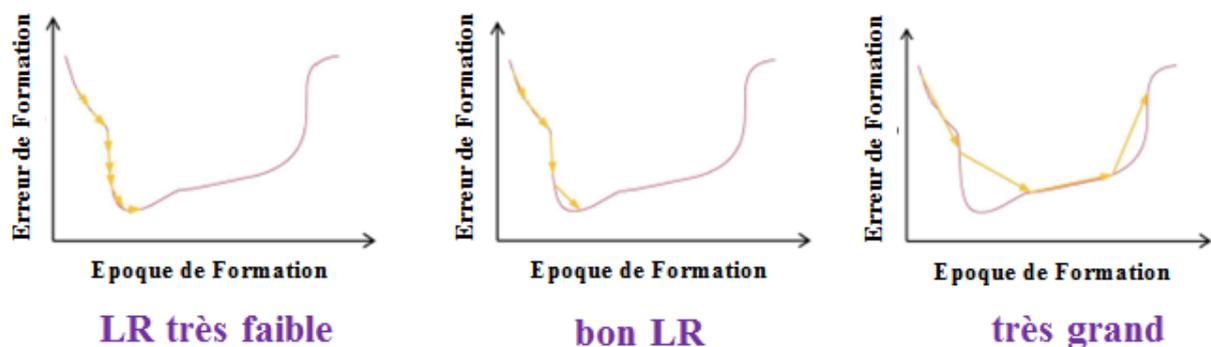
#### 2.4.4 Sélection de l'optimiseur

Après avoir discuté avec succès des étapes de prétraitement avec les échantillons de données et appliqué techniques de régulation au modèle CNN, nous allons ici discuter du processus d'apprentissage de le modèle CNN. le processus d'apprentissage comprend deux choses principales, la première est la sélection de l'algorithme d'apprentissage (Optimizer) et le suivant consiste à utiliser plusieurs améliorations (telles que momentum, Adagrad, AdaDelta) à cet algorithme d'apprentissage afin d'améliorer le résultat. l'objectif principal de tout algorithme d'apprentissage supervisé est de minimiser l'erreur (différence entre la sortie

prévue et la sortie réelle) ou nous pouvons dire les fonctions de perte, basées sur plusieurs paramètres apprenables comme les poids, les biais, etc.

En cas d'apprentissage d'un modèle CNN les méthodes d'apprentissage basées sur les gradients sont un choix naturel. Pour réduire l'erreur, les paramètres du modèle sont continuellement mis à jour pendant chaque période d'entraînement et le modèle recherche de manière itérative la solution localement optimale à chaque époque d'entraînement.

La taille des étapes de mise à jour des paramètres est appelée "taux d'apprentissage" et une itération complète de la mise à jour des paramètres qui inclut l'ensemble des données d'entraînement une fois est appelée une "époque d'entraînement." bien que le taux d'apprentissage soit un hyper-paramètre, nous devons le choisir avec tant de soin qu'il n'affecte pas trop le processus d'apprentissage, comme le montre la **Figure 2.14**.



**Figure 2.14** L'effet de la valeur différente du taux d'apprentissage (LR) sur le processus de formation.

## 2.5 Progrès récents dans les architectures CNN

Nous avons présenté les concepts de base de CNN. En plus des divers composants de base ou blocs de construction d'un CNN, nous discutons du processus d'apprentissage d'un CNN avec plusieurs algorithmes différents.

Formation avec des directives pour améliorer l'efficacité (y compris le prétraitement, l'initialisation des paramètres et la régulation de CNN).

Dans cette section, nous tentons d'expliquer un exemple d'architecture CNN réussie qui montre les avancées majeures récentes de l'architecture CNN dans le domaine de la vision par ordinateur. La vision par ordinateur a trois sous-domaines principaux où de nombreuses structures CNN (modèles) jouent un rôle essentiel pour obtenir d'excellents résultats, et nous discuterons de ces sous-domaines avec les modèles CNN associés comme suit.

### 2.5.1 Classification des images

Dans la classification d'images, nous supposons que l'image d'entrée contient un seul objet, puis nous devons classer l'image dans l'une des catégories cibles prédéfinies à l'aide de

modèles CNN. il existe plusieurs architectures (modèles) CNN principales conçues pour la classification des images, notamment :

▪ **LeNet-5**

LeNet-5 est l'une des plus anciennes architectures CNN, conçue pour classer les nombres manuscrits. il a été introduit par LeCun et al en 1998. Il comporte 5 couches pondérées (entraînables). comme le montre la figure suivante :

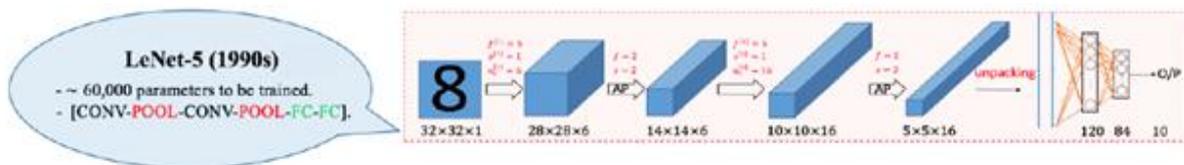


Figure 2.15 Structure de LeNet-5[25].

▪ **AlexNet**

Il a conçu le premier modèle CNN à grande échelle, appelé AlexNet en 2012, qui a été conçu pour classer les données ImageNet. Sa structure se présente comme suit :

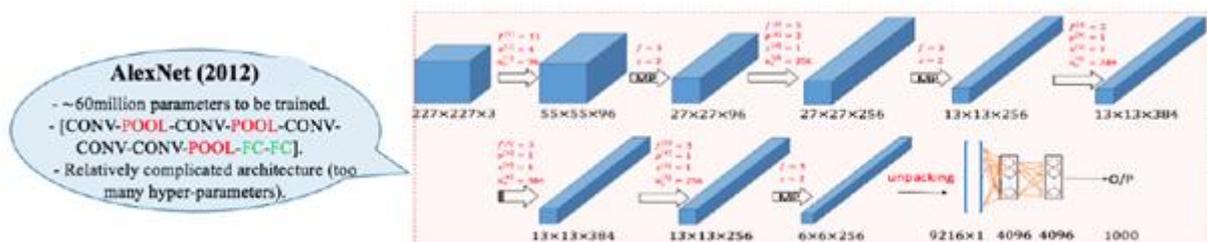


Figure 2.16 Structure de AlexNet[26].

▪ **VGGNet**

VGGNet est l'une des architectures CNN les plus célèbres, qui a été introduite par Simonyan et Zisserman en 2014. Les auteurs ont présenté un total de 6 conférences CNN différentes, parmi lesquelles VGGNet-16 (configuration D) et VGGNet-19 (con guration E) sont les plus réussis. Sa structure se présente comme suit :

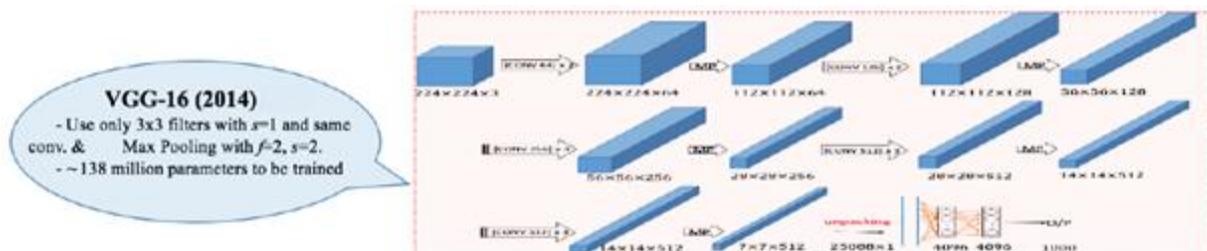


Figure 2.17 Structure de VGGNet[28].

▪ **ResNet**

Il et al ont introduit l'idée de corrélation de saut d'identité pour résoudre le problème du gradient de fuite en proposant un modèle ResNet. L'architecture ResNet complète se compose

de plusieurs poids résiduels avec des couches d'enveloppement. Comme dans la figure suivante :

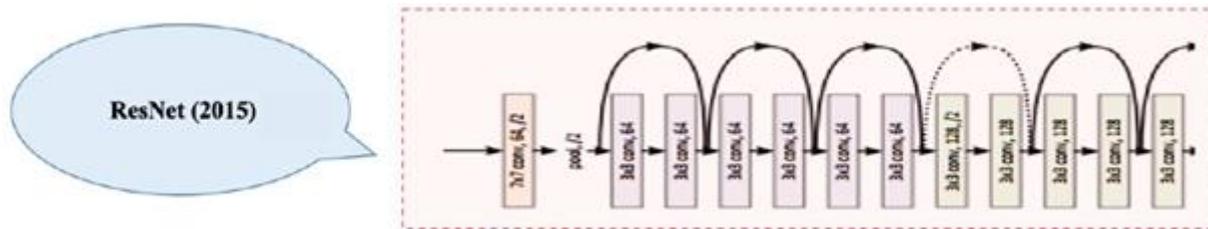


Figure 2.18 Structure de ResNet[29].

### 2.5.2 Détection d'objets :

Contrairement à la classification d'image, l'image d'entrée peut ici contenir plus d'un élément. nous essayons de détecter ces objets dans l'image d'entrée avec une identification correcte de chaque objet ainsi que leur emplacement correct dans cette image à l'aide de modèles CNN. il existe plusieurs architectures CNN majeures (modèles) conçues pour détecter des objets tels que :

- ❖ R-CNN.
- ❖ SPP-Net.
- ❖ Fast R-CNN.
- ❖ Faster R-CNN.
- ❖ Mask R-CNN.
- ❖ YOLO.

### 2.5.3 Segmentation des images

CNN a également fait preuve d'excellence dans la tâche de segmentation.

Après la performance record d'AlexNet en 2012, nous avons de nombreux modèles de pointe de segmentation sémantique et segmentation des instances.

Certains d'entre eux sont mis en évidence ci-dessous.

#### 2.5.3.1 Segmentation sémantique

Contrairement à la Classification et à la détection d'objets, la segmentation sémantique est une tâche de vision de bas niveau.

Ce est le processus d'association de chaque pixel d'une image avec une étiquette de classe i. e. il détecte tous les objets présents dans une image.

#### 2.5.3.2 Segmentation des instances

La segmentation d'instance fait avancer la segmentation sémantique. il détecte ainsi distingue toutes les instances d'un objet présent dans une image.

## 2.6 Domaines d'application des CNN

Dans cette section, nous discutons de certains des principaux domaines d'application qui appliquent CNN pour atteindre l'état-performances de pointe, y compris la classification d'images, la reconnaissance de texte, la reconnaissance d'action, détection d'objet, estimation de pose humaine, légende d'image, etc.

### **2.6.1 Classification des images**

En raison de plusieurs fonctionnalités telles que le partage de poids, différents niveaux d'extraction de fonctionnalités, tels que classificateurs, etc., le CNN a atteint une meilleure précision de classification[19] par rapport à d'autres méthodes, en particulier dans le cas d'ensembles de données à grande échelle.

La première percée dans l'image la classification vient avec le développement d'AlexNet en 2012, qui a remporté l'ILSVRC[24] défi cette même année. après cela, plusieurs améliorations du modèle CNN ont été apportées par les chercheurs au fil du temps, ce qui fait de CNN le premier choix pour la classification des images problème.

### **2.6.2 Reconnaissance de texte**

La détection de texte et la reconnaissance de texte à l'intérieur d'une image ont été largement étudiées depuis longtemps.

La première contribution révolutionnaire de CNN dans ce domaine commence avec LeNet-5, qui a reconnu les données du jeu de données MNIST[32] avec une bonne précision. après cela ces dernières années, avec plusieurs améliorations, CNN contribue un rôle essentiel[33] pour reconnaître le texte (chiffres, alphabet et symboles appartenant à plusieurs langues) à l'intérieur de l'image.

### **2.6.3 Reconnaissance des actions**

Basé sur l'apparence visuelle et la dynamique de mouvement de tout corps humain, divers Les méthodes de base CNN sont désormais capables de prédire l'action ou le comportement de sujets humains avec une précision notable.

Cela conduit le CNN au niveau supérieur dans le contexte de l'IA. il comprend reconnaissance d'action à partir d'une séquence vidéo ou à partir des images fixes.

### **2.6.4 Génération de légendes d'image**

Cela signifie obtenir une description de l'image cible, qui comprend la détection et reconnaissance de différents objets à l'intérieur de cette image avec leur description d'état.

Ici, nous avons utilisé CNN pour effectuer la première tâche[27] et nous avons utilisé plusieurs traitements de langage naturel (NLP) techniques pour une description textuelle du statut.

### **2.6.5 Analyse des images médicales**

Avec les progrès de l'analyse d'images basée sur CNN, CNN s'est rapidement avéré être un état-fondation de pointe, en obtenant des performances accrues dans le diagnostic des maladies par pro-Cesser les images médicales[34] comme l'IRM, les rayons X, etc.

De nos jours, les modèles basés sur CNN peuvent réussir diagnostiquer les divers problèmes de santé comme le cancer du sein, la pneumonie, la tumeur au cerveau, le diabète, maladies de Parkinson et bien d'autres.

### **2.6.6 Sécurité et surveillance**

De nos jours, le système de sécurité avec des capacités de vision par ordinateur fournit une surveillance constante à maisons, stations de métro, routes, écoles, hôpitaux et bien d'autres endroits, ce qui donne la possibilité pour trouver ou identifier les criminels même dans les zones surpeuplées[35].

### **2.6.7 Colorisation automatique de l'image et transfert de style**

Au cours des dernières années, avec la révolution de l'apprentissage en profondeur, certains modèles CNN populaires donnent un moyen d'automatisation pour convertir des images en noir et blanc ou des images grises en RVB coloré équivalent image.

En conséquence, nous pouvons maintenant voir les vieux films en noir et blanc au format couleur. Sur d'autre part, le transfert de style d'image est un concept de représentation d'une image dans le style d'un autre image, pour cela une nouvelle image artificielle pourrait être générée.

Ce transfert de style pourrait être efficacement fait à l'aide de réseaux de neurones convolutifs[30].

### **2.6.8 Imagerie satellitaire**

De nos jours, CNN joue un rôle vital dans la détection de déférents aléas naturels[31] comme les tsunamis, les ricanes, inondations et glissements de terrain. par analyse d'images satellites, nous pouvons faire un plan de ville intelligent, une chaussée et extraction fluviale, classification des terres, classification des schémas de culture, prévention de la déforestation et beaucoup plus.

## **2.7 Conclusion**

Les réseaux de neurones convolutifs (CNN) sont devenus un algorithme des dernières technologies pour les problèmes de vision par ordinateur, de traitement du langage naturel et de reconnaissance de formes.

CNN a été utilisé pour créer une variété de modèles de cas d'utilisation allant de la reconnaissance des chiffres et des lettres à l'analyse complexe d'images médicales.

Ce chapitre a tenté d'expliquer tous les composants d'un CNN, leur fonctionnement dans l'analyse d'images et d'autres éléments connexes.

Ce chapitre fournit également un examen de CNN depuis sa création jusqu'aux derniers modèles et mentionne certains de ses domaines d'application.

## Chapitre 3

# Résultats expérimentaux et discussions

### 3.1 Introduction

Dans ce chapitre, nous traiterons de la dernière partie qui représente la réalisation de notre projet, basé sur les mécanismes évoqués plus haut dans le chapitre Réseaux de neurones.

Nous avons divisé ce chapitre en deux parties : la première présente l'environnement de notre programme, et la seconde les résultats des tests effectués.

### 3.2 Environnement matériel

- Type de PC Acer
- Processeur: Intel Core i3-5005U (2.0GHz, 3MB L3 Cache)
- Mémoire: 4GB DDR L.
- Intel HD Graphics 5500
- Disque Dure: 1000 GB HDD
- Ecran : 1000Mbps
- Système d'exploitation: Windows 10 64bits (Professional).

### 3.3 Aperçu de l'environnement de programmation

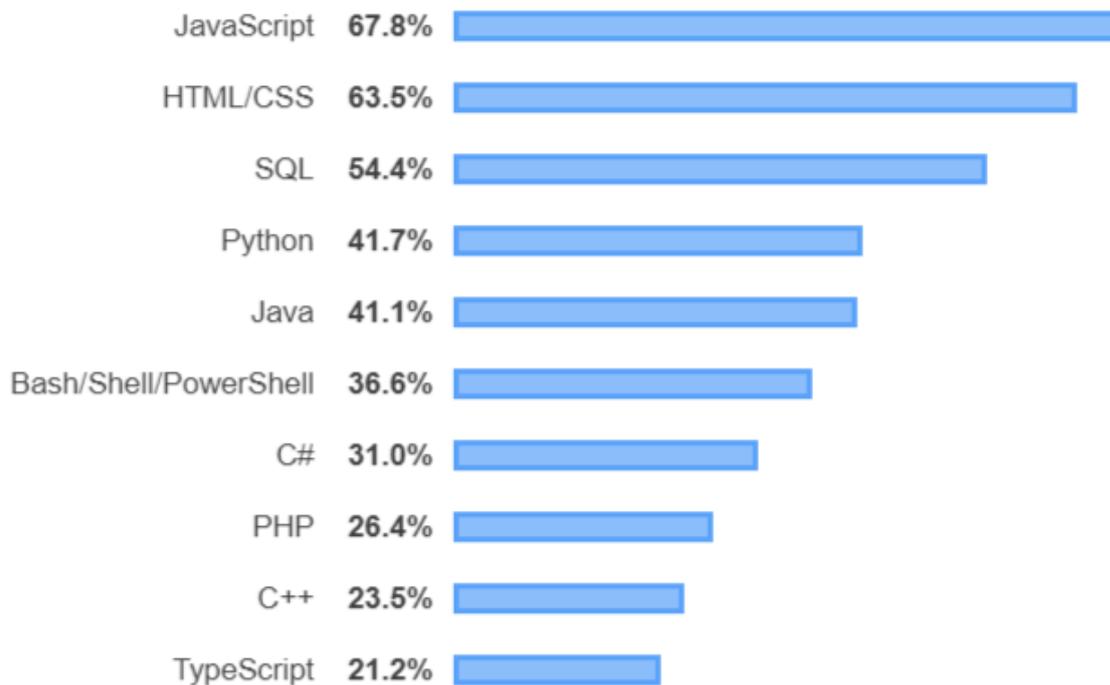
#### 3.3.1 Python

C'est l'un des langages de programmation les plus populaires et les plus utilisés, idéal pour les nouveaux programmeurs et facile à lire. Ils peuvent être utilisés dans tout, des jeux vidéo et du traitement du langage à l'analyse de données et à l'apprentissage automatique. Il comporte des mots anglais simples contrairement aux autres langues.

Python est utilisé dans tous les domaines, y compris :

- Analyse de données, robotique, applications REST.
- Développement de sites Web et de jeux, graphismes 3D.
- Utilisé dans de nombreux sites Web et entreprises géantes.

- Les graphiques suivants montrent la popularité du langage Python parmi les programmeurs



**Figure 3.1** Langages de programmation les plus populaires.

Nous avons utilisé de deux types de bibliothèque Python :

#### **A. TensorFlow**

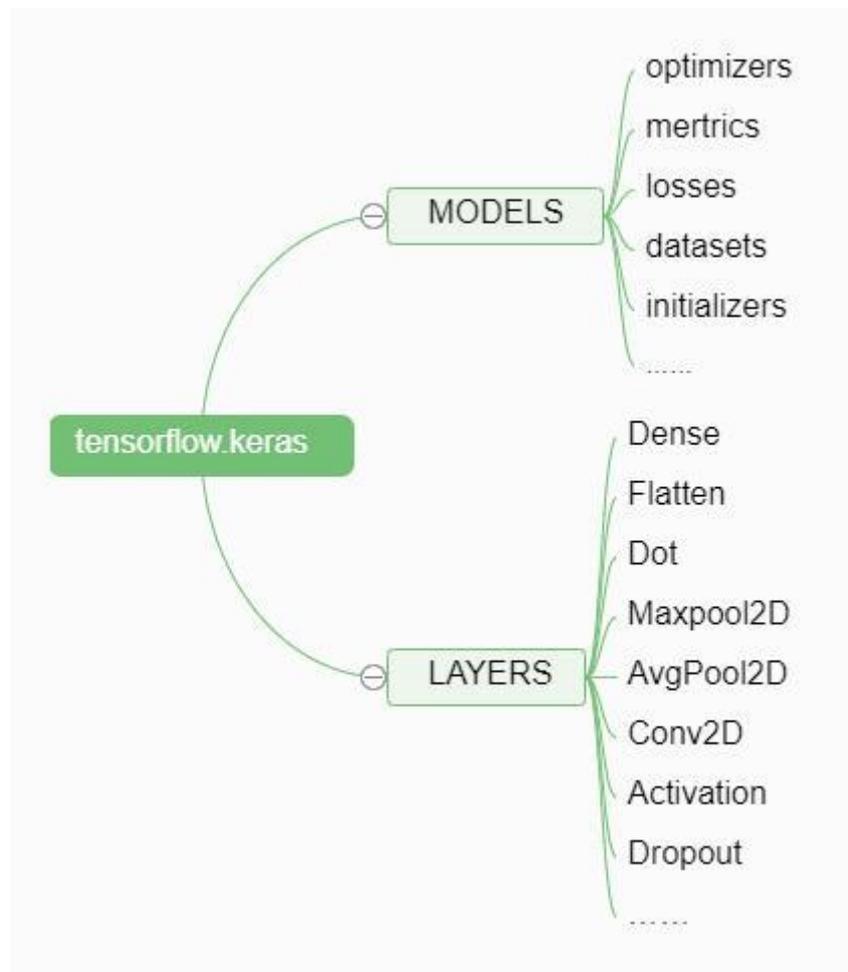
Il s'agit d'une bibliothèque open source qui a initialement adopté python comme langage de programmation pour créer de nombreux modèles, principalement basés sur des algorithmes d'apprentissage automatique.

Il est actuellement utilisé dans de nombreux domaines de recherche différents et tensorflow permet aux développeurs de concevoir, construire et former facilement des modèles d'apprentissage en profondeur. il a une caractéristique majeure qui est l'abstraction.

#### **B. Keras**

Il s'agit d'une API de réseau neuronal de haut niveau écrite en Python, facile à utiliser, extensible et standard.

La structure de base de Keras est la suivante :



**Figure 3.2** La structure de base de Keras.

Keras se compose de deux parties importantes : le modèle et la classe.

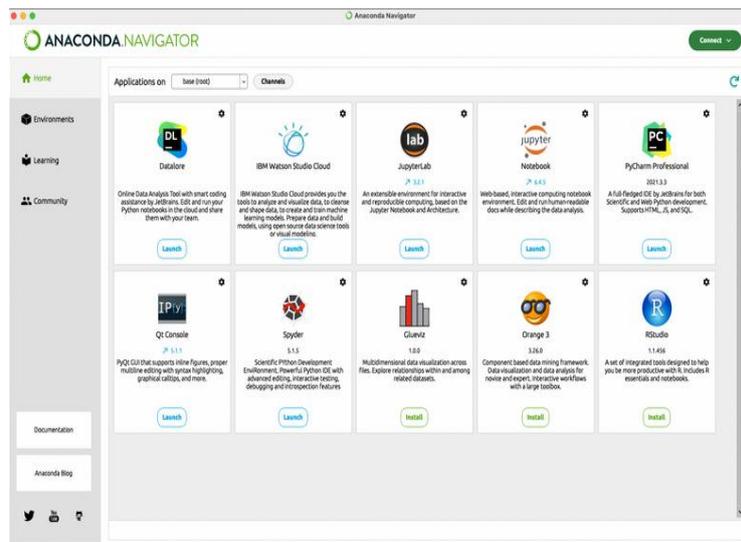
## 3.4 Les Outils de travail

### 3.4.1 Navigateur Anaconda

Il s'agit d'une interface utilisateur graphique (GUI) de bureau, qui vous permet d'exécuter facilement des applications et de gérer des packages, des environnements et des canaux conda sans utiliser de commandes de ligne de commande [45]

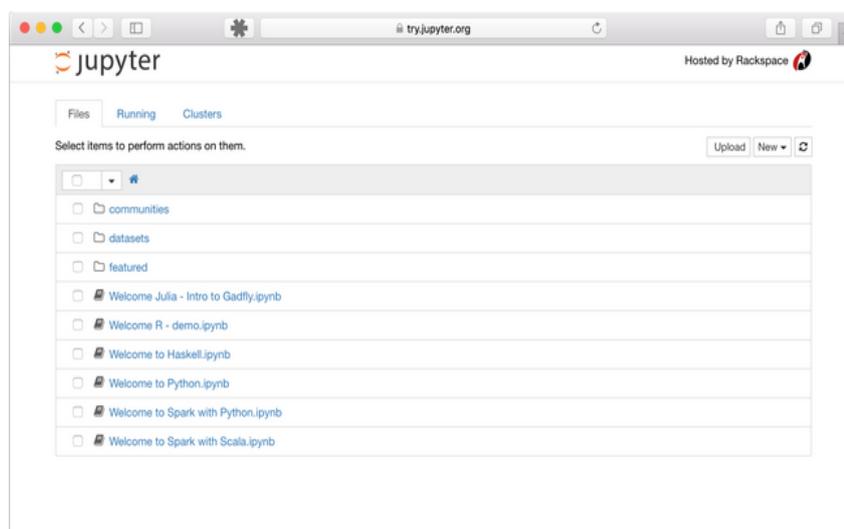
- **Applications auxquelles j'ai accès avec Navigator**

Ils sont disponibles par défaut dans Navigator et nous expliquerons par la suite l'interface Classic Notebook qui nous avons utilisé.



**Figure 3.3** Interface du navigateur.

L'interface Classic Notebook est une interface orientée document qui vous permet de créer, d'afficher et d'exécuter du code dans Jupyter Note [45]



**Figure 3.4** interface Jupyter Note.

## 3.5 Base des données

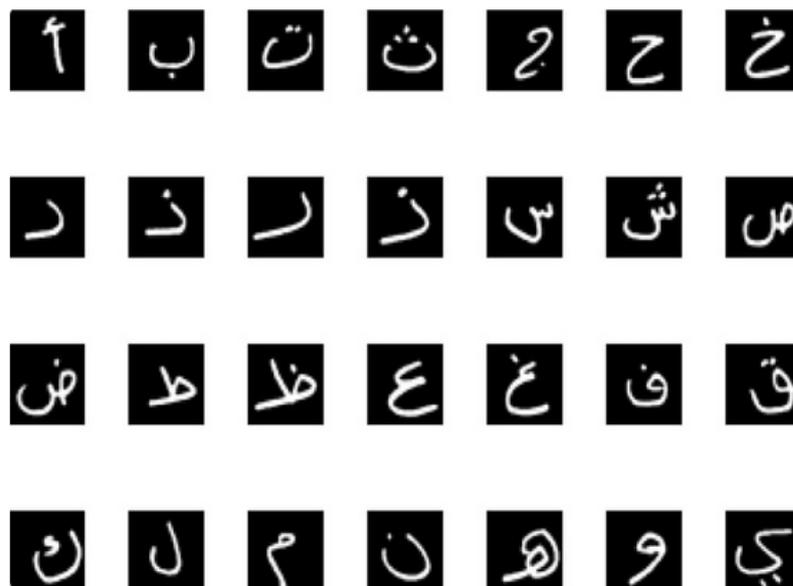
### 3.5.1 (MNIST)

MNIST est une base de données de chiffres manuscrits qui est utilisée pour la formation et le test de nombreux systèmes de traitement d'images et la recherche sur l'apprentissage automatique [41]. Il contient 60 000 images d'apprentissage et 10 000 images de test.

L'ensemble de données est considéré comme la norme pour l'apprentissage automatique dans le monde entier. La base de données MNSIT est utile pour les personnes qui souhaitent expérimenter des techniques d'apprentissage automatique et des méthodes de reconnaissance de formes sur des données du monde réel avec peu d'efforts de prétraitement et de coordination [41].

### 3.5.2 (AHCD)

La base de données AHCD a été créée par le Dr Mohamed Loy, Maître de Conférences au Département d'Informatique, Faculté d'Informatique, Université de Benha, Egypte. Cette base de données contient 16800 caractères arabes écrits par 60. Les participants ont entre 19 et 40 ans et 90% des participants sont Ayman. Chaque participant a écrit chaque lettre (de "a" à "oui") dix fois. Les images ont été numérisées à 300 dpi (nombre par pouce : pixels par pouce de pouce). Le jeu de données est divisé en deux groupes : un jeu d'apprentissage de 13 440 caractères avec 480 images par classe et un jeu de test de 3 360 caractères avec une moyenne de 120 images par classe, Les auteurs du groupe d'apprentissage et de test sont exclusifs. L'ordre dans lequel les éditeurs sont inclus L'ensemble de test est sélectionné au hasard pour s'assurer que les rédacteurs de l'ensemble de test n'ont pas été obtenus d'une seule institution pour assurer la variance des tests croisés. La base de données est disponible gratuitement sur [43] pour les chercheurs, et sa création pose d'autres défis car elle traite de nombreux problèmes tels que le style d'écriture, l'épaisseur, le nombre et la position. Il s'agit d'une base de données moderne créée en 2017, qui est disponible en ligne gratuitement



**Figure 3.5** La base de données des caractères arabes manuscrits AHCD [44]

### 3.6 Résultats et discussion

Les résultats obtenus sont présentés en deux parties consistent en un modèle pour la base de données MNIST et un modèle pour la base de données AHCD. Chacun comporte deux phases : une phase d'apprentissage (Training) et une phase d'évaluation des performances du modèle (test).

Dans cette section, on va présenter ces étapes en détail.

#### 3.6.1 Architecture et résultats du modèle MNIST

Ce modèle de la figure 3.9 se compose d'une couche et de deux classes d'assemblage maximum. L'image saisie en taille  $32 * 32$  passe en premier au calque. Cette couche était composée de 32 filtres de taille  $3 * 3$  suivis d'une couche maxpooling ( $2 \times 2$ ), puis nous avons implémenté deux couches entièrement connectées de 100 et 10 neurones respectivement.

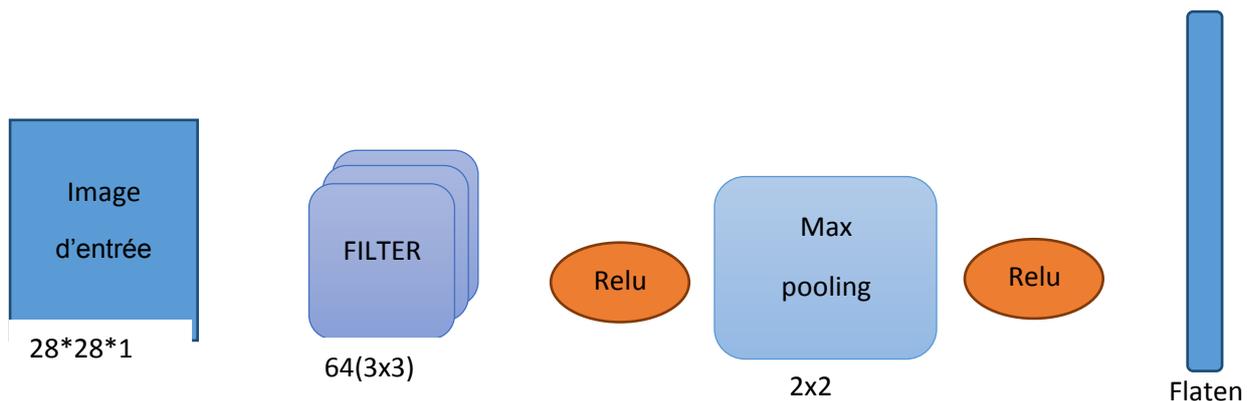


Figure 3.6 Architecture Define Model ()

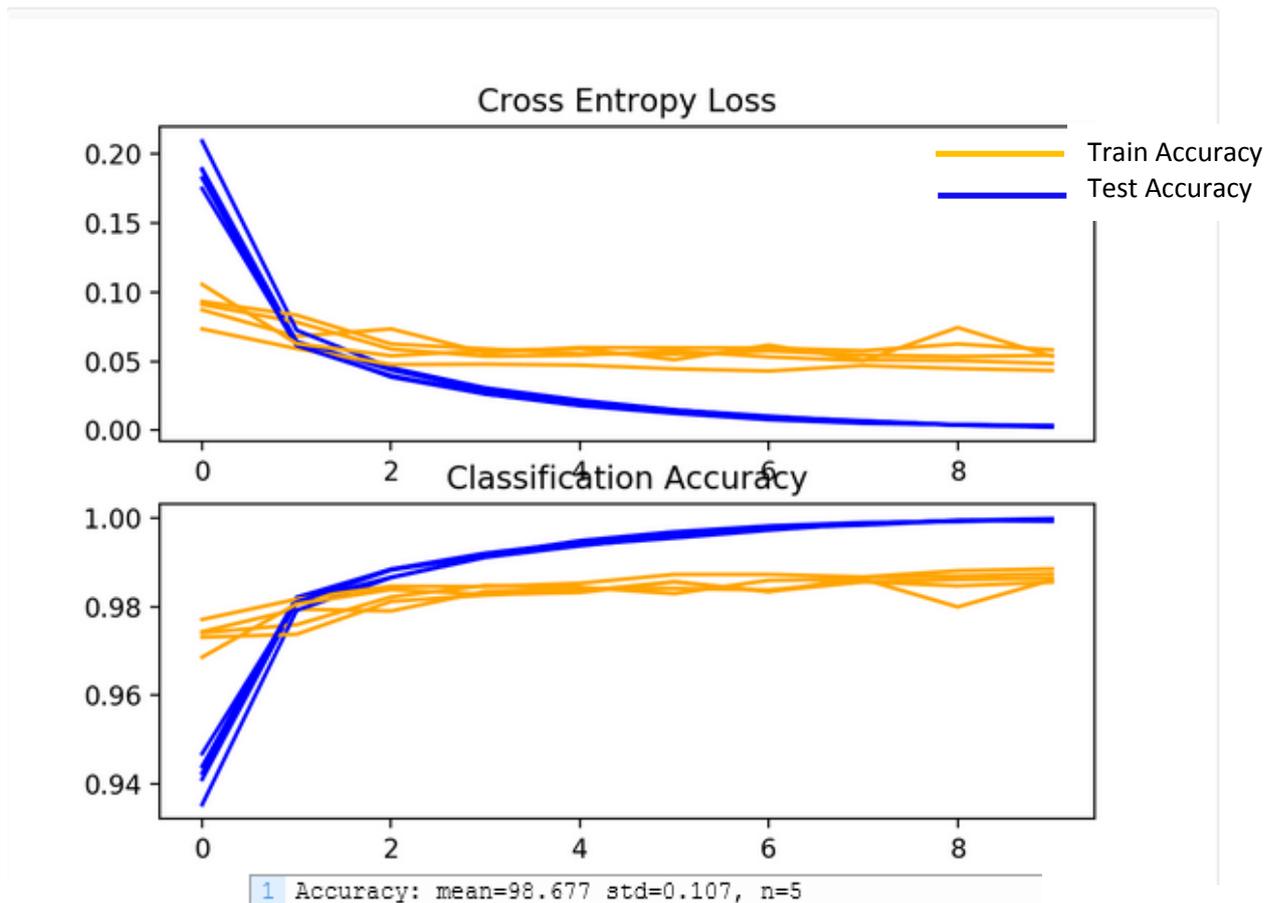
```

2 def define_model():
3     model = Sequential()
4     model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform',
5     model.add(MaxPooling2D((2, 2)))
6     model.add(Flatten())
7     model.add(Dense(100, activation='relu', kernel_initializer='he_uniform'))
8     model.add(Dense(10, activation='softmax'))
9     # compile model
10    opt = SGD(lr=0.01, momentum=0.9)
11    model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])
12    return model

```

Figure 3.7 function Define Model ()

Nous obtenons le résultat suivant après implémentation:



**Figure 3.8** Résultat de l'application MNIST

Parce qu'un tableau de diagnostic est affiché, il nous donne un aperçu du comportement d'apprentissage du modèle à travers chaque pli.

Dans ce cas, nous pouvons voir que le modèle atteint généralement un bon ajustement, les courbes d'apprentissage, d'entraînement et de test convergeant. Il n'y a aucun signe évident de sur- ou de sous-ajustement.

### 3.6.2 Architecture et résultats des modèles AHCD

Au cours de nos expériences, nous avons créé et testé plusieurs modèles dans ce qui suit, nous offrons la structure et les résultats de deux modèles.

#### - Architecture du modèle 1

Le premier modèle que nous proposons dans la figure 3.9 se compose de deux couches et de deux catégories maxpooling

Nous avons créé deux couches similaires. L'image d'entrée est de taille 32 \* 32 passe d'abord au premier couche de convolution. Cette couche se compose de 32 filtres de taille 3 \* 3 suivis

d'une couche maxpooling (2x2) et ce processus se répète une autre fois, ensuite nous avons implémenté deux couches totalement connecté de 100 et 10 neurones respectivement.

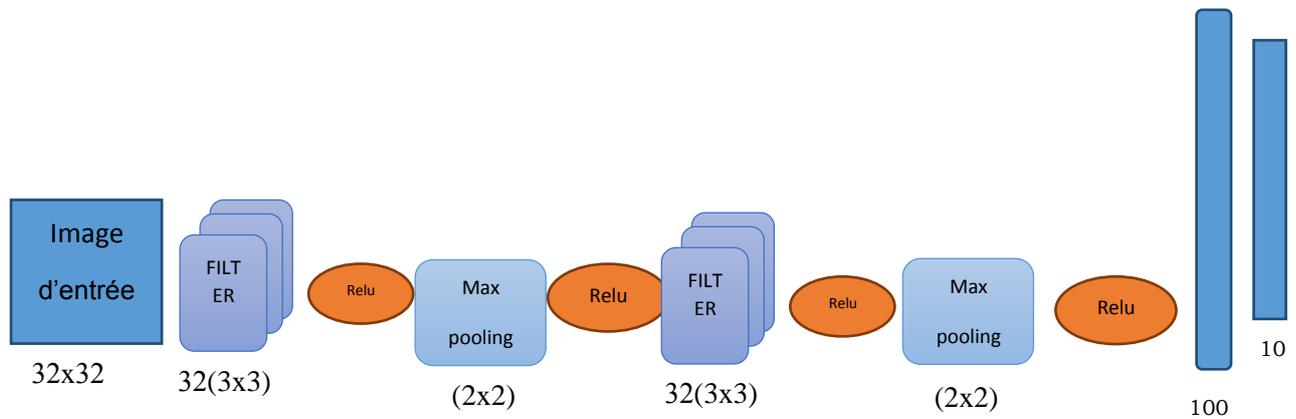


Figure 3.9 Architecture du modèle 1

```
def define_model():
    model = Sequential()
    model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', input_shape=(32, 32, 1)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', input_shape=(32, 32, 1)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Flatten())
    model.add(Dense(100, activation='relu', kernel_initializer='he_uniform'))
    model.add(Dense(28, activation='softmax'))
```

Figure 3.10 function Define Model () 1

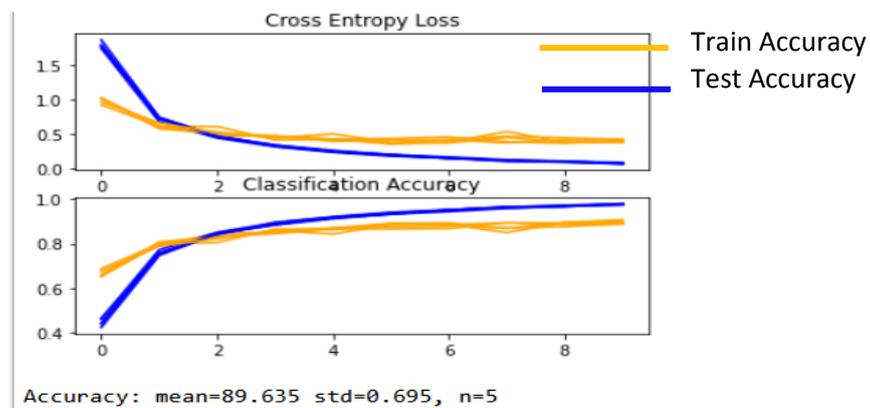


Figure 3.11 Résultat du model () 1

After calculating the performance summary of the model. We can see in this case that the model has an estimated skill of about 89.6% which is a poor result

### Architecture du modèle 2

Le deuxième modèle que nous présentons sur la figure 3.12 consiste en trois couches de convolution et trois classes de maxpooling similaires. Chaque couche de convolution se compose de 120 filtres d'une taille de 3 \* 3.

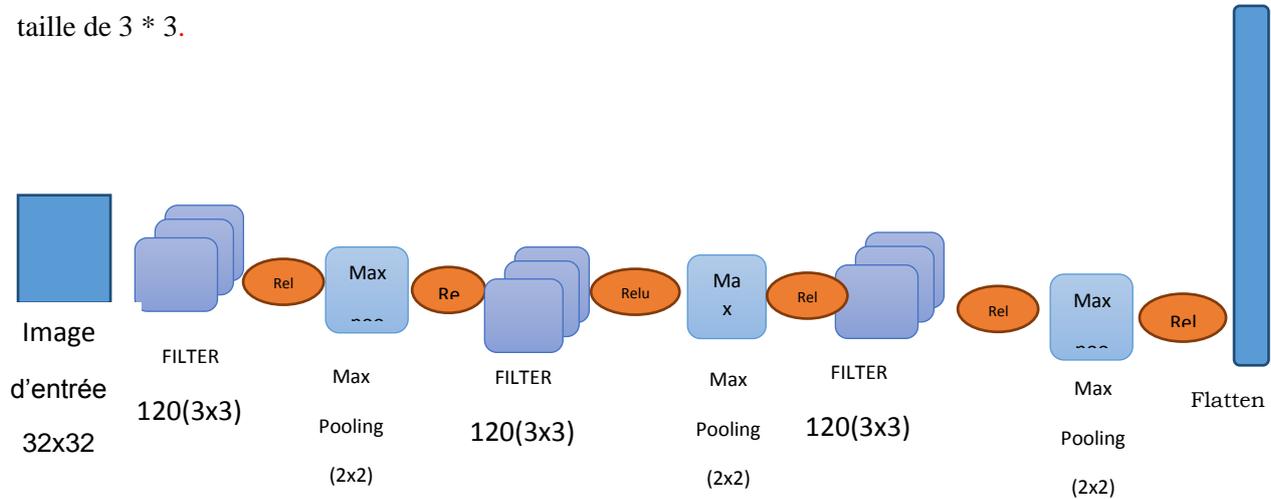


Figure 3.12 Architecture du modèle 2

```
def define_model():
    model = Sequential()
    model.add(Conv2D(120, (3, 3), activation='relu', kernel_initializer='he_uniform', input_shape=(32, 32, 1)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(120, (3, 3), activation='relu', kernel_initializer='he_uniform', input_shape=(32, 32, 1)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(120, (3, 3), activation='relu', kernel_initializer='he_uniform', input_shape=(32, 32, 1)))
    model.add(MaxPooling2D((2, 2)))
    model.add(Flatten())
    model.add(Dense(100, activation='relu', kernel_initializer='he_uniform'))
    model.add(Dense(28, activation='softmax'))
```

Figure 3.13 function Define Model () 2

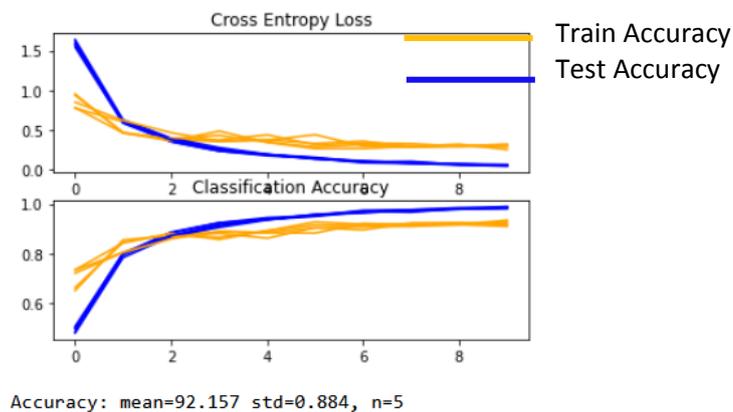


Figure 3.14 Résultat Model () 2

### 3.7 Conclusion

Dans ce chapitre, nous avons d'abord présenté l'environnement matériel et logiciel utilisé, ensuite nous avons montré l'utilisation des réseaux de neurones convolutifs pour reconnaître les lettres arabes manuscrites, deux base de données populaire MNIST et AHCD ont été utilisé les résultats obtenus sont 98.67% et 92.15% respectivement.

## Conclusion générale

---

### Conclusion générale

Enfin, Reconnaître les mots cursifs est une tâche très importante, utilisée dans de nombreuses applications telles que la lecture d'adresses postales et la reconnaissance de montants littéraux. Actuellement, malgré toutes les découvertes des chercheurs, l'identification de la calligraphie arabe manuscrite est encore aujourd'hui au niveau de la recherche et de l'expérimentation. Même parce qu'il n'existe pas de solution optimale au problème de la reconnaissance cursive, les recherches dans ce domaine ont permis le développement d'un certain nombre de systèmes pratiques pour cette raison, ce domaine de recherche offre encore un terrain fertile pour des travaux futurs manuscrits.

Dans ce mémoire nous avons réalisé un système de reconnaissance de caractères Arabe manuscrite en utilisant les réseaux de neurones convolutifs. Nous avons fait varier les différents paramètres du réseau pour tirer des bons résultats. Pour évaluer notre système, les deux bases de données populaires MNIST et AHCDB ont été utilisés. Nous avons obtenu un taux de reconnaissance de 98.67% et 92.15% respectivement.

## Bibliographies

---

### Bibliographies

- [1] T. Liu, A.W. Moore, and A. Gray, "Efficient exact k-NN and nonparametric classification in high dimensions", in proceeding of: Advances in Neural Information Processing Systems, 2002
- [2] G. Kim, V. Govindaraju, and S.N. Srihari, "An Architecture for handwritten text recognition systems", International Journal on Document Analysis and Recognition (IJ DAR), vol.2, pp. 37-44, 1999p.1377-1385, 2005
- [3] B. Al-Badr, and R.M. Haralick , "Segmentation-free word recognition with application to Arabic ", IEEE. Proceeding of the 3rd International Conference on Document Analysis and Recognition (ICDAR'95), pp. 355-359, Montreal-Canada,
- [4] T. Steinherz, E. Rivlin, N. Intrator , "Off-line cursive word recognition :a survey". International journal on document analysis and recognition, 2(2), pp. 90-110, 1999.
- [5] B. Al-Badr and S.A. Mahmoud, "Survey and bibliography of Arabic optical text recognition", Signal processing, vol.41, pp.49-77, 1995.
- [6] p E.J.R. Justino, F. Bortolozzi, and R. Sabourin, "A comparison of SVM and HMM classifiers in the off-line signature verification", Pattern Recognition Letter, vol.26, no.9,
- [7] S. HAITAMARS, Segmentation de textes en caractères pour la reconnaissance optique de l'écriture arabe, thèse de Magister, université de Batna, Algérie, 2007.
- [8] J.J. De Oliveira, J. De Carvalho, C.O. De A. Freitas, and R. Sabourin, "Evaluating NN and HMM classifiers for handwritten word recognition", Brazilian Symposium on Computer
- [9] N. Ben Amara, "Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée". These de doctorat, specialite Genie Electrique, Universite des sciences, des Techniques et de medecine de Tunis II, 1999.
- [10] R. Azizi, Une approche hybride pour la reconnaissance d'écriture arabe manuscrite, mémoire de magistère, université les frères Montouri, Constantine, 2007.
- [11] N. Benamara, Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée, thèse de doctorat, université de Tunis II, 1999.
- [12] H. Miled, C. Olivier, et M. Cheriet, "Modélisation de la notion de pseudo-mots en reconnaissance de mots manuscrits arabes", dans le proceeding de la conférence CIFED, 2000.

## Bibliographies

---

- [13] I.R. Tsang, "Pattern recognition and complex systems". Thèse de doctorat, université d'Anterwerpen, 2000.
- [14] S. Kermi, Classifieur neuronal base connaissances, application à la reconnaissance des caractères arabes isolés manuscrits, thèse de magister, université d'Annaba, Algérie, 1999.
- [15] S. Kermi, "Classifieur neuronal base connaissances, application à la reconnaissance des caractères arabes isolés manuscrits ", Thèse de magister, université Badji Mokhtar, Annaba, Algérie 1999.
- [16] P. Burrow, Arabic handwriting recognition, Master of science thesis, university of Edinburg, England, 2004.
- [17] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016.  
<http://www.deeplearningbook.org>.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097{1105. Curran Associates, Inc., 2012.
- [19] F. Sultana, A. Sufian, and P. Dutta. Advancements in image classification using convolutional neural network. In 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pages 122{129, Nov 2018.
- [20] F. Sultana, A. Sufian, and P. Dutta. A review of object detection models based on convolutional neural network. CoRR, abs/1905.01614, 2019.
- [21] N. M. Zaitoun and M. J. Aqel. Survey on image segmentation techniques. Procedia Computer Science, 65:797 { 806, 2015. International Conference on Communications, management, and Information technology (ICCMIT'2015).
- [22] D. H. Hubel and T. N. Wiesel. Receptive \_elds and functional architecture of monkey striate cortex. Journal of Physiology (London), 195:215{243, 1968.
- [23] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4):193{202, Apr 1980.
- [24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV),

## Bibliographies

---

- 115(3):211{252, 2015.
- [25] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Apprentissage basé sur les gradients appliqué à la reconnaissance de documents. Actes de l'IEEE, 86(11), 2278-2324.
- [26] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Classification Imagenet avec réseaux de neurones à convolution profonde. Dans les progrès des systèmes de traitement de l'information neuronale (pp. 1097-1105).
- [27] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.
- [28] Simonyan, K., & Zisserman, A. (2014). Réseaux convolutionnels très profonds pour la reconnaissance d'images à grande échelle. arXiv preprint arXiv:1409.1556.
- [29] He, K., Zhang, X., Ren, S. et Sun, J. (2016). Apprentissage résiduel profond pour la reconnaissance d'images. Dans Actes de la conférence IEEE sur la vision par ordinateur et la reconnaissance des formes (pp. 770-778).
- [30] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.
- [31] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Convolutional neural networks for large-scale remote-sensing image classification. IEEE Transactions on Geoscience and Remote Sensing, 55(2):645{657, Feb 2017.
- [32] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010.
- [33] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), pages 3304{3308, Nov 2012.
- [34] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan. Medical image analysis using convolutional neural networks: A review. J. Med. Syst., 42(11):1{13, Nov. 2018.
- [35] P. Rasti, T. Uiboupin, S. Escalera, and G. Anbarjafari. Convolutional neural network super resolution for face recognition in surveillance monitoring. volume 9756, pages 175{184, 07 2016.
- [36] S. Touj, N. Ben-Amara, and H. Amiri, "Reconnaissance hors ligne de caractères arabes isolés manuscrits", dans le proceeding de la conférence CIFED, 2002

## Bibliographies

---

- [37] S. Haitaamars, Segmentation de textes en caractères pour la reconnaissance optique del'écriture arabe, thèse de Magister, université de Batna, Algérie, 2007.
- [38] F. Menasri, Contributions à la reconnaissance de l'écriture arabe manuscrite, thèse de Doctorat, université Paris Descartes, France, 2008.
- [39] H. Miled, C. Olivier, et M. Cheriet, "Modélisation de la notion de pseudo-mots en reconnaissance de mots manuscrits arabes", dans le proceeding de la conférence CIFED,2000
- [40] T. Steinherz, E. Rivlin, and N. Intrator, "Off-line cursive word recognition: a survey", International Journal on Document Analysis and Recognition (IJ DAR), vol.2, no.2, pp.90-110, 1999.
- [41] Y. LeCun. "The MNIST database of handwritten digits."  
<http://yann.lecun.com/exdb/mnist>
- [42] E. Kussul and T. Baidyk, "Improved method of handwritten digit recognition tested on MNIST database," Image and Vision Computing, vol. 22, pp. 971-981, 10/01 2004, doi: 10.1016/j.imavis.2004.03.008
- [43] Ashish Vaswani , Noam Shazeer , Niki Parmar , JakobUszkoreit , Llion Jones , Aidan N. Gomez , Łukasz Kaiser , IlliaPolosukhin ,Attention is all you need , NIPS Proceedings2017.
- [44] Armin Shmilovici, Support Vector Machines,  
[https://link.springer.com/chapter/10.1007/0-387-25465-X\\_12](https://link.springer.com/chapter/10.1007/0-387-25465-X_12)
- [45] S. Mozaffari, K. Faez, F. Faradji, M. Ziaratban and S.M. Golzan, "A comprehensive isolated Farsi/Arabic character database for handwritten OCR research", proceedings of IWFHR, Paris-France, 2006.
- [46] R. Al-Hajj, C. Mokbel, and L. Likforman-Sulem, "Combination of HMM-based classifiers for the recognition of Arabic handwritten words", proceedings of International Conference on Document Analysis and Recognition (ICDAR), vol.2, pp.959-963, 2007.
- [47] A. Benouareth, A. Ennaji, and M. Sellami, "Semi-continuous HMM with explicit state duration applied to Arabic handwritten word recognition", in the tenth International Workshop on Frontiers in Handwriting Recognition (IWFHR), La Baule-France, 2006.