REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Khider – BISKRA

Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : SIOD31/M2/2022

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **System d'informations, Optimisation et Décision (SIOD)**

La détection et la reconnaissance de l'écriture arabe manuscrite en utilisant les RNC

Par :

GHETTAS KHALED.

Soutenu le 28\06\2022, devant le jury composé de :

Zerarka Moahammed Faouzi		Président
Chami Djazia		Examinateur
Slatnia Sihem	M.C.A	Rapporteur

I. Dédicacions

Je dédie ce modeste travail : À mes chers parents pour leurs soutient durant toute Ma vie d'études

A l'encadreur mme.slatnia siham

Et Mes sœurs et Mon frère Yasmine, Ikram, Islam Et aussi Mes amis

À tous les professeurs et les enseignants que j'ai eu durant tout mon cursus scolaire et qui m'ont permis de réussir dans mes études. À toute personne ayant contribué à ce travail de près ou de loin.

II. *Remerciement*

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire " All-Hamdoulillah ".

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu voir le jour sans l'aide et l'encadrement de mme.SLATNIA Siham, je les remercie pour la qualité de son encadrement exceptionnel, pour leur patience et leur disponibilité durant la préparation de ce mémoire.je remercie mes parents pour leur soutien pendant toute ma carrière scolaire et universitaire. Je remercie les membres de jury d'avoir porté intérêt à notre étude et d'avoir accepté d'examiner notre travail. J'exprime aussi mes gratitudes à tous les enseignants du département d'informatique sans exception, qui nous ont fournis les outils nécessaires à la réussite dans nos études universitaires.

III. Résumé

La reconnaissance de l'écriture arabe manuscrite est une opération parfaite pour trouver son application dans plusieurs domaines.

Cependant, il existe certains domaines où la reconnaissance de l'écriture manuscrite a été utilisée avec un certain succès. Le Tri automatique des emails, traitement automatique des fichiers de gestion, formulaires d'enquête, encaissement des chèques bancaires, etc. La reconnaissance de l'écriture manuscrite est beaucoup plus compliquée que la reconnaissance de l'écriture imprimée en raison de son extrême variabilité (variabilité de la forme, espacement des mots à la lettre, variabilité des lignes, similarité d'écriture manuscrite des mots).

Ce mémoire traite en détail de la reconnaissance hors ligne de l'écriture arabe manuscrite. Nous avons proposé un modèle d'apprentissage profond qui repose essentiellement sur un mécanisme efficace appelé réseau de neurones convolutifs.

Des études expérimentales montrent que les RNC ont considérablement amélioré la reconnaissance de l'écriture arabe manuscrite, ainsi que sa fiabilité.

Les performances de ce modèle ont été testées dans la base de données IFN \ ENIT. Les résultats obtenus sont très encourageons, avec un taux de précision de 92% pour la classification, malgré la complexité des mots et les similitudes entre l'écriture des différents mots.

Table des Matières

LISTES DES FIGURES 6

LISTE DES TABLEAUX 8

LISTE DES EQUATIONS 9

INTRODUCTION GENERALE 10

CHAPITRE 1 : L'APRENTISSAGE AUTOMATIQUE ET PROFOND 12

1. INTRODUCTION 13

2. QU'EST-CE QUE L'APPRENTISSAGE AUTOMATIQUE? 13

2.1 APPRENTISSAGE SUPERVISE 13

2.1.1 La Classification et la Régression *14*

2.2 APPRENTISSAGE AUTOMATIQUE NON SUPERVISE 15

2.2.1 Regroupement ou Clustering *16*

2.2.2 Association *16*

2.3 APPRENTISSAGE AUTOMATIQUE SEMI-SUPERVISE 17

3. APPRENTISSAGE A PROFOND (DEEP LEARNING) 18

3.1 FONCTIONNEMENT DU DEEP LEARNING 18

3.2 DOMAINES D'APPLICATION DU DL 19

3.3 LES POINTS FORTS ET LES POINTS FAIBLES DU DL LES POINTS FORTS 20

3.3.1 Les points fort *20*

3.3.2 Les points faibles *21*

4. RESEAUX DE NEURONES 22

4.1 RNN LES RESEAUX DE NEURONES NATURELS 22

4.2 RESEAUX DE NEURONES ARTIFICIELS 23

4.3 LES RESEAUX DE NEURONES PROFONDS (CNN) 24

4.3.1 Définition *24*

4.3.2 Architecture *24*

4.3.3 Couche convolution *26*

4.3.4 Pooling *28*

4.3.4.1 Définition 28

4.3.4.2 Principe 29
4.3.5 stirde *30*
4.3.6 padding *31*
4.3.7 Couche fully-connected *31*
4.3.8 Architecture d'un CNN *32*
5. HYPER PARAMETRES IN DEEPLEARNING 33
5.1 NOMBRE D'EPOQUES 34
5.2 TAILLE DU LOT 34
6. LES FONCTIONS DE DEEPLEARNING 34
6.1 FONCTIONS DE PERTE 34
6.2 FONCTION DE COUT 35
7. LA DESCENTE DE GRADIENT (GD) 36
8. MATRICE DE CONFUSION 36
8.1 DEFINITION 36
8.2 METRIQUES DE LA MATRICE DE CONFUSION 37
8.2.1 Précision *38*
8.2.2 Précision (accuracy) *38*
8.2.3 Rappeler (recall) *38*
8.2.4 F1 score *39*
9. TRANSFER LEARNING 39
10. COMMENT FONCTIONNE L'APPRENTISSAGE PAR TRANSFERT 40
11. LE PARAMETRAGE DES COUCHES 40
12. CONCLUSION 41
CHAPITRE 2 : LA RECONNAISSANCE D'ECRITURE ARABE MANUSCRITE 42
1. INTRODUCTION 43
2. RECONNAISSANCE EN LIGNE ET HORS LIGNE 43
2.1 SYSTEMES EN LIGNE 43
2.2 SYSTEMES HORS LIGNE : 43
2.3 CRITERES D'INFLUENCES 44

2.4 COMPARAISON DE LA RECONNAISSANCE DES CARACTERES EN LIGNE ET HORS-LIGNE 45
3. LA RECONNAISSANCE DE L'ECRITURE ARABE MANUSCRITE 45
3.1 PRESENTATION DE LA LANGUE ARABE 45
3.2 DIFFICULTES INHERENTES A LA RECONNAISSANCE DE L'ECRITURE ARABE 45
3.2.1 Alphabet 45
3.2.2 Signes diacritiques 46
3.2.3 Des points nécessaires pour différencier les lettres 46
4. DOMAINES D'APPLICATION DE LA RECONNAISSANCE MANUSCRITE 47
4.1 DOMAINES D'APPLICATIONS 47
4.2 QUELQUES EXEMPLES D'APPLICATIONS 48
5. TRAVAUX CONNEXES 49
6. DETECTION DE TEXTE ARABE DANS LES VIDEOS 50
7. CLASSIFICATION DES REGIONS DE TEXTE 50
8. DETECTION BASEE SUR LE RESEAU DE NEURONES A CONVOLUTION 51
9. UN APERÇU DES RESEAUX DE NEURONES A CONVOLUTION 51
10. LES DIFFERENTS TYPES DE FORMAT D'IMAGE 54
10.1 DEFINITION DE L'IMAGE NUMERIQUE 54
11. CARACTERISTIQUES DE L'IMAGE 55
11.1 LES PIXELS 55
11.2 LA RÉSOLUTION D'UNE IMAGE 56
11.3 IMAGE AU NIVEAU DE GRIS 57
11.4 CONTRASTE 57
11.5 LUMINANCE 57
11.6 BRUIT 57
11.7 CONTOUR 58
12. CONCLUSION 58
CHAPITRE 3 : CONCEPTION ET IMPLEMENTATION 59
1. INTRODUCTION 60

2. RECONNAISSANCE DE L'ECRITURE ARABE PAR LE DEEP LEARNING 60
2.1 EXPLICATION GENERALE 60
2.2 CONCEPTION GLOBALE 60
2.2.1 Architecture 60
2.2.2 Quelques étapes spécifiques à notre programme 61
2.2.3 Acquisition 61
2.2.4 Prétraitement 61
2.2.4.1 Suppression de bruit 62
2.2.4.2 Normalisation 62
2.2.4.3 Seuillage 63
2.2.4.4 FILTRAGE 64
2.2.4.5 Squelettisation : 65
2.2.5 Extraction des caractéristiques 65
2.2.5.1 Type de caractéristiques 65
2.2.5.2 Caractéristiques structurelles 65
2.2.5.3 Caractéristiques statistiques 66
2.2.5.4 Caractéristiques globales 66
2.2.5.5 Caractéristiques morphologiques 66
2.2.5.6 Caractéristiques métriques 66
2.2.5.7 Caractéristiques adaptatives 67
3. ARCHITECTURE D'UN CNN 67
3.1 TYPES DE COUCHE 67
3.1.1 Couche convolutionnelle (CONV) 67
3.1.2 Pooling (POOL) 67
3.1.3 Fully Connected (FC) 68
4. COMPRENDRE LA COMPLEXITE DU MODELE 68
4.1 FONCTIONS D'ACTIVATION COMMUNEMENT UTILISEES 69
4.1.1 Relu : 69
4.1.2 Softmax : 70
5. CLASSIFICATION 71
6. DESCRIPTION DE LA BASE DE DONNEES 72

7. METRIQUE D'EVALUATION DE PERFORMANCE 73
8. CONCLUSION 74
CHAPITRE 4 : RESULTATS EXPERIMENTAUX 75
1. INTRODUCTION 76
2. CONCEPTION DE L'ARCHITECTURE CNN 76
3. LOGICIELS ET LIBRAIRIES UTILISES DANS L'IMPLEMENTATION 76
4. CONFIGURATION UTILISE DANS L'IMPLEMENTATION : 79
5. LE DATASET UTILISE 80
6. L'ARCHITECTURE D'UN SYSTEM DE RECONNAISSANCE D'ECRITURE ARABE MANUSCRITE 80
6.1 CODE SOURCE DE NOTRE MODELE PROPOSE 82
7. RESULTATS EXPERIMENTAUX 86
7.1 L'INTERFACE DE NOTRE SYSTEM 86
7.2 L'ETAPE DE PRETRAITEMENTS 86
7.2.1 Les filtre que nous avons utilisé sur les images 86
7.3 TAUX DE PRECISION DU MODELE 89
8. QUELQUES RESULTATS EXPERIMENTAUX 90
9. MATRICE DE CONFUSION DE NOTRE TEST 92
10. DISCUSSION 93
11. CONCLUSION : 93
CONCLUSION GENERALE 94
REFERENCES 95

Listes des Figures

Figure 1 : Classification et de la Régression 14
Figure 2 : apprentissage Automatique supervisé vs d'apprentissage Automatique non-supervisé 16
Figure 3: La relation entre l'intelligence artificielle, le ML et le deep Learning..... 18
Figure 4 : exemple d'un réseau de neurones artificiel 24
Figure 5 : Architecture standard d'un réseau de neurone convolutionnel 25
Figure 6 : Ensemble de neurones (cercles) créant la profondeur d'une couche de convolution (bleu). Ils sont liés à un même champ récepteur (rouge) 26
Figure 7 : exemples des matrices qui représentent des pixels d'une image 27
Figure 8 : L'APPLICATION DU FILTRE SUR L'IMAGE 27
Figure 9 : image qui montre le résultat du filtre appliquée 28
Figure 10 : figure qui montre le max pooling 29
Figure 11 : Inter correlation avec des foulées de 3 et 2 pour la hauteur et la largeur, respectivement. 30
Figure 12 : Intercorrélation bidimensionnelle avec rembourrage. 31
Figure 13 : Les activations d'un exemple d'architecture ConvNet. 32
Figure 14 : Examples des couches de convolution 33
Figure 15 : fonction de la perte..... 35
Figure 16 : figure qui montre le transfer learning..... 40
Figure 17 : Exemple de sortie de notre système de bout en bout. Le cadre de délimitation rouge localise le texte contenant une partie de l'image. 50
Figure 18 : Convolution Neural Network architecture. 54
Figure 19 : ensemble de pixels qui représente une image 56
Figure 20 ; représente la résolution d'une image..... 56
Figure 21 : Diagramme de référence du système de reconnaissance. 61
Figure 22 : représente la suppression de bruit..... 62
Figure 23 : Exemples de mots manuscrits avec des tailles différentes 63
Figure 24 : Normalisation de certains chiffres manuscrits. 63
Figure 25 ; Exemple de binarisation d'une image en niveaux de gris..... 64
Figure 26 : l'application de filtre sur limage 64
Figure 27 : Exemples de squelettisation 65

Figure 28 : architecture d'un CNN 67
Figure 29 : l'application du couche fully connected 68
Figure 30 : fonction RELU 70
Figure 31 : fonction SOFTMAX 71
Figure 32 : CNN Feature Extraction pour l'image du chiffre manuscrit 72
Figure 33: L'architecture utilisée sur notre system 80
Figure 34 : CONFIGURATION DU NOTRE MODELE RNC 81
Figure 35 : l'interface d'un system de reconnaissance d'écritures manuscrites 86
Figure 36 : résultat en appliquant le filtre max 87
Figure 37 : résultat en appliquant le filtre gaussian 87
Figure 38 : résultat en appliquant la segmentation 88
Figure 39 : résultat en appliquant la squelettisation 88
Figure 40 : TAUX DE PRECISION DU MODELE 89
Figure 41 : image représente la précision le recall et f1-score pour chaque classe 90
Figure 42 : résultat de test 1 91
Figure 43 : résultat de test 2 91
Figure 44 : résultat de test 3 92
Figure 45 : matrice de confusion 92

Liste des Tableaux

Tableau 1 : tableau qui montre la matrice de confusion 37
Tableau 2: Comparaison de la reconnaissance des caractères en ligne et hors- ligne 45
Tableau 3 : tableau des caractères arabe 47
Tableau 4 : Points en arabe : un, deux ou trois points 47
Tableau 5 : tableau qui montre la differance entre le max pooling et l'average pooling.. 68
Tableau 6 : tableau qui montre la difference entre le pooling et la couche de convolution et la couche fully connected 69

Liste des équations

Équation 4 : equation de precision 73
Équation 5 : equation de recall 73
Équation 6 : equation de l'accuracy 73
Équation 7 : equation de F1-score 74

Introduction Générale

La reconnaissance de l'écriture manuscrite ou imprimée reste encore un sujet de recherche et d'expérimentation, le problème n'est pas encore entièrement résolu bien que l'on sache atteindre des taux assez élevés dans certaines applications pour lesquelles soit le vocabulaire est limité, soit la fonte est unique ou en nombre restreint.

Il existe cependant plusieurs domaines pour lesquels la reconnaissance de l'écriture est appliquée avec un certain succès : le tri automatique du courrier, le traitement automatique de dossiers administratifs, des formulaires d'enquêtes, ou encore l'enregistrement des chèques bancaires. La reconnaissance de l'écriture manuscrite est beaucoup plus complexe que celle de l'écriture imprimée due à son extrême variabilité: variabilité des formes, des espacements entre mots et caractères, fluctuation des lignes.

La langue arabe est considérée comme l'une des langues les plus utilisée au monde avec plus 400millions de locuteurs (internet).

Pour résoudre ce problème on a choisi les réseaux de neurones convolutifs. Dans la fin des années 80 Yan le Cun a développé un type de réseau particulier qui s'appelle le réseau de neurones convolutionnel, ces réseaux sont une forme particulière de réseau neuronal multicouche dont l'architecture des connexions est inspirée de celle du cortex visuel des mammifères. Par exemple, chaque élément n'est connecté qu'à un petit nombre d'éléments voisins dans la couche précédente [48].

Aujourd'hui, cette technique est partout. Les réseaux profonds sont utilisées quotidiennement pour reconnaître les visages de millions de citoyens qui passent devant les caméras de surveillance, permettant ainsi une baisse drastique de la délinquance, même si le procédé soulève des polémiques quant à l'utilisation de l'IA par des régimes répressifs .Dans un contexte plus positif, les prochaines années verront la sortie officielle de la première voiture autonome grâce aux algorithmes de deep learning capables de reconnaitre les piétons, les panneaux de signalisation et d'anticiper les trajectoires des objets sur la route [48].

Pour résoudre le problème de la reconnaissance des mots, on a utilisé les réseaux de neurones convolutifs en utilisant aussi la base de données IFN\ENIT pour faire l'entrainement et le test de ce modèle. Les résultats obtenus sont encourageants, le taux d'exactitudes de classification est de 92% malgré les difficultés et le manque de ressources

Dans notre modeste projet, nous avons conçu notre mémoire sur quatre 4 chapitres :

- Chapitre1 : s'informer et se documenter sur l'apprentissage automatiques et l'apprentissage profond, on discuter aussi sur l'architecture d'un CNN ainsi que les étapes principale de réaliser ce dernier
- Chapitre 2 : discuter et se documenter sur la reconnaissance de l'écriture arabe manuscrite, ces difficultés et ces domaines d'application et aussi s'informer la détection basée sur le réseau de neurones à convolution, ainsi que le différent type de l'image
- Chapitre 3 : on a discuté sur la conception globale d'un system de reconnaissance de l'écriture manuscrite arabe, l'architecture utilisé et les étapes pour réaliser ce dernier
- Chapitre 4 : dans ce chapitre on a bien expliqué comment notre system fonctionne, on a montré aussi nos résultats de test

Nous clôturons ce mémoire en indiquant les quatre chapitres de ce dernier qui nous explique comment réaliser un system de reconnaissance d'écriture arabe manuscrits.

CHAPITRE 1

1. Introduction

L'apprentissage automatique est un domaine dans lequel des algorithmes sont appliqués à un ensemble de données pour extraire des modèles de l'ensemble de données. Ceux-ci peuvent à leur tour être appliqués sur des données similaires à des fins de prédiction. Avec suffisamment de données, il est possible de formuler une approximation de la relation entre toutes les variables d'entrée et les valeurs particulières dites « cible ». Cette approche diffère de la programmation conventionnelle dans laquelle une application est développée sur la base de règles préalablement définies.

2. Qu'est-ce que l'apprentissage automatique?

L'apprentissage automatique, également appelé apprentissage machine ou apprentissage artificiel et en anglais machine learning, est une forme d'intelligence artificielle (IA) qui permet à un système d'apprendre à partir des données et non à l'aide d'une programmation explicite. Cependant, l'apprentissage automatique n'est pas un processus simple. Au fur et à mesure que les algorithmes ingèrent les données de formation, il devient possible de créer des modèles plus précis basés sur ces données. Un modèle de machine learning est le résultat généré lorsque vous entraînez votre algorithme d'apprentissage automatique avec des données. Après la formation, lorsque vous fournissez des données en entrée à un modèle, vous recevez un résultat en sortie. Par exemple, un algorithme prédictif crée un modèle prédictif. Ensuite, lorsque vous fournissez des données au modèle prédictif, vous recevez une prévision qui est déterminée par les données qui ont servi à former le modèle. [1]

2.1 Apprentissage supervisé

La majorité des apprentissages automatiques utilisent un apprentissage supervisé (supervised learning).

L'apprentissage supervisé commence généralement par un ensemble de données bien défini et une certaine compréhension de la façon dont ces données sont classifiées. L'apprentissage supervisé a pour but de déceler des modèles au sein des données et de les appliquer à un processus analytique. Ces données comportent des caractéristiques associées à des libellés qui définissent leur signification. Vous pouvez, par exemple, créer une

application d'apprentissage automatique capable de faire la distinction entre plusieurs millions d'animaux, en se basant sur des images et des descriptions écrites. [1]

2.1.1 La Classification et la Régression

L'apprentissage supervisé est généralement effectué dans le contexte de la classification et de la régression.

- **Classification :** Un problème de classification survient lorsque la variable de sortie est une catégorie, telle que «rouge», «bleu» ou «maladie» et «pas de maladie». *Exemples* :
    - En finance et dans le secteur bancaire pour la détection de la fraude par carte de crédit (fraude, pas fraude).
    - Détection de courrier électronique indésirable (spam, pas spam).
    - Dans le domaine du marketing utilisé pour l'analyse du sentiment de texte (heureux, pas heureux).
    - En médecine, pour prédire si un patient a une maladie particulière ou non [11].
- **Régression**: Un problème de régression se pose lorsque la variable de sortie est une valeur réelle, telle que «dollars» ou «poids». *Exemples :*
    - Prédire le prix de l'immobilier
    - Prédire le cours de bourse

Certains types courants de problèmes fondés sur la classification et la régression incluent la prévision et la prévision de séries temporelles, respectivement [11].

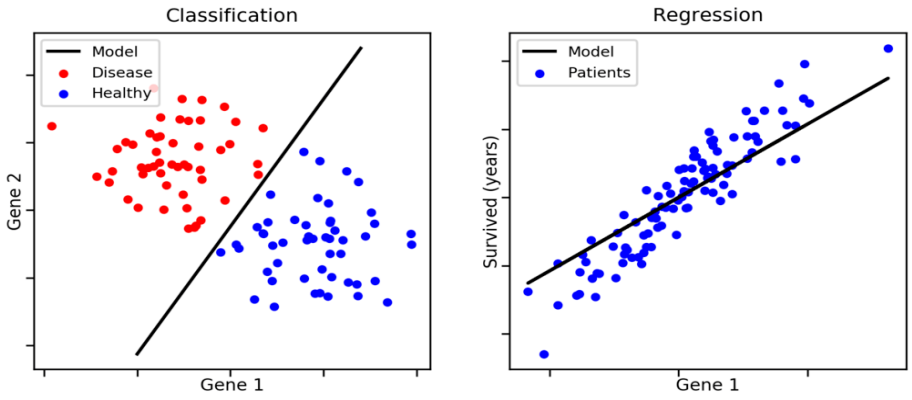


Figure 1 : Classification et de la Régression [11]

Voici quelques exemples populaires d'algorithmes d'apprentissage automatique supervisé:

- Arbres de décision
- K Nearest Neighbours
- SVC linéaire (classificateur de vecteur de support)
- Régression logistique
- Naive Bayes
- Les réseaux de neurones
- Régression linéaire
- Régression vectorielle de support (SVR)
- Arbres de régression

2.2 Apprentissage automatique non supervisé

Quand le système ou l'opérateur ne dispose que d'exemples, mais non d'étiquettes, et que le nombre de classes et leur nature n'ont pas été prédéterminés, on parle d'apprentissage non supervisé (ou clustering). Aucun expert n'est disponible ni requis. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données. Le système doit ici dans l'espace de description (la somme des données) cibler les données selon leurs attributs disponibles, pour les classer en groupe homogènes d'exemples. La similarité est généralement calculée selon la fonction de distance entre paires d'exemples. C'est ensuite à l'opérateur d'associer ou déduire du sens pour chaque groupe. Divers outils mathématiques et logiciels peuvent l'aider. On parle aussi d'analyse des données en régression. Si l'approche est probabiliste (c'est à dire que chaque exemple au lieu d'être classé dans une seule classe est associé aux probabilités d'appartenir à chacune des classes), on parle alors de « soft clustering » (par opposition au « hard clustering ») [19].

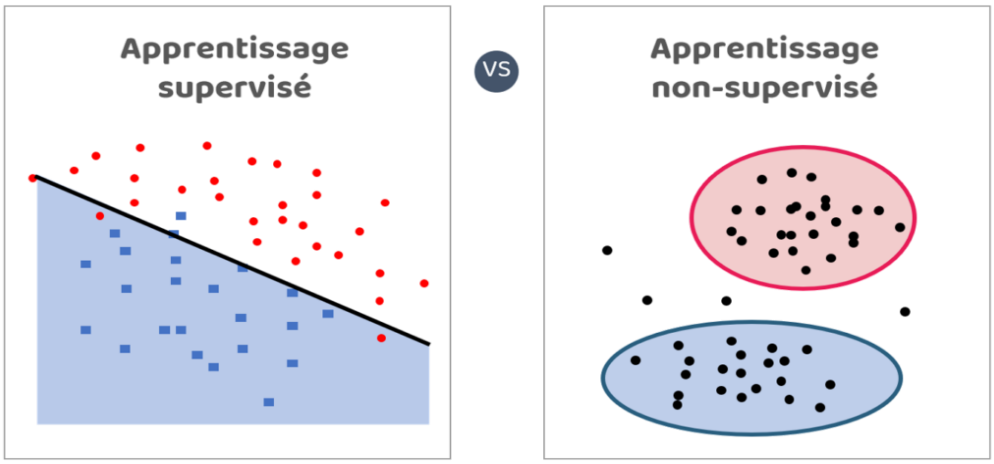


Figure 2 : apprentissage Automatique supervisé vs d'apprentissage Automatique non-supervisé [20]

2.2.1 Regroupement ou Clustering

Le regroupement ou Clustering est la technique la plus utilisée pour résoudre les problèmes d'apprentissage non supervisé. La mise en cluster consiste à séparer ou à diviser un ensemble de données en un certain nombre de groupes, de sorte que les ensembles de données appartenant aux mêmes groupes se ressemblent davantage que ceux d'autres groupes. En termes simples, l'objectif est de séparer les groupes ayant des traits similaires et de les assigner en grappes [69].

2.2.2 Association

L'association consiste à découvrir des relations intéressantes entre des variables dans de grandes bases de données [11].

Le clustering consiste à grouper des points de données en fonction de leurs similitudes, tandis que l'association consiste à découvrir des relations entre les attributs de ces points de données [11].

Voici une liste de certains algorithmes d'apprentissage automatique non supervisés:

- K-means clustering
- Dimensionality Reduction (Réduction de la dimensionnalité)
- Neural networks / Deep Learning
- Principal Component Analysis (Analyse des composants principaux)

- Singular Value Decomposition (Décomposition en valeur singulière)
- Independent Component Analysis (Analyse en composantes indépendantes)
- Distribution models (Modèles de distribution)
- Hierarchical clustering (Classification hiérarchique) [11].

2.3 Apprentissage automatique semi-supervisé

Les problèmes pour lesquels une grande quantité de données d'entrée (X) et que seules certaines données sont étiquetées (Y) sont appelés problèmes d'apprentissage semi-supervisés. Par conséquent, ces problèmes se situent entre l'apprentissage supervisé et l'apprentissage non supervisé. [11]

De nombreux problèmes de machine learning du monde réel tombent dans ce domaine. En effet, il peut être coûteux en temps ou en argent d'étiqueter des données car cela peut nécessiter un accès à des experts de domaine.

Considérant que les données sans étiquette sont peu coûteuses et faciles à collecter et à stocker.

Vous pouvez utiliser des techniques d'apprentissage non supervisées pour découvrir et apprendre la structure dans les variables d'entrée.

Vous pouvez également utiliser des techniques d'apprentissage supervisé pour établir des prévisions optimales pour les données non étiquetées, les transférer dans l'algorithme d'apprentissage supervisé en tant que données d'apprentissage et utiliser le modèle pour effectuer des prédictions sur de nouvelles données invisibles.[11]

Nous savons maintenant que:

- Supervisé: toutes les données sont étiquetées et les algorithmes apprennent à prédire le résultat des données d'entrée [11].
- Non supervisé: toutes les données ne sont pas étiquetées et les algorithmes apprennent la structure inhérente à partir des données en entrée [11].
- Semi-supervisé: Certaines données sont étiquetées mais la plupart d'entre elles ne sont pas étiquetées et un mélange de techniques supervisées et non supervisées peut être utilisé [11].

3. Apprentissage a profond (Deep Learning)

L'apprentissage a profond (Deep learning : DL) est une branche de l'apprentissage automatique qui enseigne aux ordinateurs à faire ce qui vient naturellement aux humains : apprendre de l'expérience. Les algorithmes d'apprentissage automatique utilisent des méthodes de calcul pour "apprendre" des informations directement à partir de données sans s'appuyer sur une équation prédéterminée comme modèle. DL est une classe de techniques d'apprentissage automatique qui exploitent de nombreuses couches de traitement d'informations non linéaires pour l'extraction et la transformation de fonctions supervisées ou non supervisées, et pour l'analyse et la classification de modèles. Les caractéristiques et les concepts de niveau supérieur sont donc définis en termes de niveaux inférieurs, et une telle hiérarchie de caractéristiques est appelée architecture profonde [3]. La plupart de ces modèles sont basés sur un apprentissage non supervisé des représentations

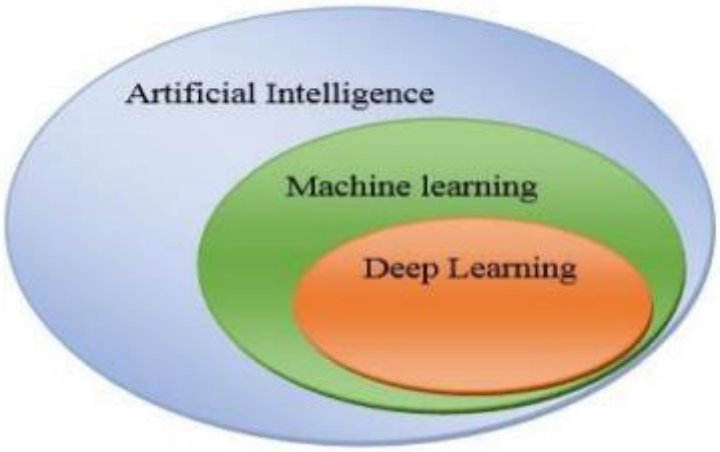


Figure 3: La relation entre l'intelligence artificielle, le ML et le deep Learning

3.1 Fonctionnement du deep learning

Le Deep Learning est un domaine d'apprentissage automatique basé sur un type particulier d'apprentissage des données. Il est caractérisé par l'effort de créer un modèle d'apprentissage automatique à plus d'un niveau, dans lequel les niveaux les plus profonds prennent des contributions des niveaux précédents, les transformant et les abstrayant de plus en plus. Cet aperçu des niveaux d'apprentissage donne le nom à l'ensemble du domaine (apprentissage en profondeur) et s'inspire de la façon dont le cerveau des mammifères traite

l'information et apprend en réagissant aux stimuli externes. Chaque niveau d'apprentissage correspond dans ce parallèle hypothétique, à l'une des différentes zones qui composent le cortex cérébral. Par exemple, le cortex visuel, responsable de la reconnaissance des images, montre une séquence de secteurs hiérarchisés. Chacun de ces secteurs reçoit une représentation d'entrée, au moyen des signaux de flux qui le relient aux autres secteurs. Chaque niveau de cette hiérarchie représente un niveau d'abstraction différent, Les caractéristiques les plus abstraites étant définies par rapport à celles du niveau inférieur. Lorsque le cerveau reçoit des images, il les traite à travers différentes phases, par exemple la détection des contours, la perception des formes (des primitives aux plus complexes graduellement). C'est pourquoi nous parlons de représentation hiérarchique de l'image au niveau de l'abstraction croissante. Au fur et à mesure que le cerveau apprend par essais et erreurs et active de nouveaux neurones apprenant de l'expérience, même dans les architectures responsables de l'apprentissage profond, les étapes d'extraction peuvent être modifiées en fonction des informations reçues à l'entrée. Les réseaux neurones sont l'un des principaux outils qui bénéficient du Deep Learning.

Les algorithmes de machine learning classiques fonctionnent très bien dans certains cas. Cependant, ils semblent être inefficaces pour les problèmes majeurs du deep learning telles que la reconnaissance vocale et la reconnaissance d'objets. La principale différence entre les l algorithmes de machine learning traditionnel et le deep learning est l'étape de l'extraction de caractéristiques. Pour le machine learning traditionnel les caractéristiques à identifier dépendent de l'expertise humaine. Cette pratique est très difficile et requiert un spécialiste en la matière, parfois les caractéristiques distinctives ne sont même pas identifiables par l'homme. Le deep learning permet de pallier à ce problème en utilisant plusieurs couches de réseau de neurones. Les premières couches extrairont des caractéristiques simples (présence de contours) que les couches suivantes combineront pour former des concepts de plus en plus complexes et abstraits : assemblages de contours en motifs, de motifs en parties d'objets, de parties d'objets en objets, etc. Un autre avantage du deep learning c'est qu'il s'adapte bien, plus la quantité de données fournie est grande plus les performances sont meilleurs. [3]

3.2 Domaines d'application du DL

Les nouvelle techniques de Deep Learning se développent dans plusieurs domaine dans la vie quotidienne on peut citer quelque domaine et comment on a utilisé Le DL [3].

- **Conduite automatisée** : Les chercheurs du secteur automobile ont recours au Deep Learning pour détecter automatiquement des objets tels que les panneaux stop et les feux de circulation. Le Deep Learning est également utilisé pour détecter les piétons, évitant ainsi nombre d'accidents [3].
- **Aérospatiale et défense** : Le Deep Learning sert à identifier des objets à partir de satellites utilisés pour localiser des zones d'intérêt et identifier quels secteurs sont sûrs ou dangereux pour les troupes au sol [3].
- **Recherche médicale** : À l'aide du Deep Learning, les chercheurs en cancérologie peuvent dépister automatiquement les cellules cancéreuses. Des équipes de l'Université de Californie à Los Angeles (UCLA) ont conçu un microscope qui génère un ensemble de données de grande dimension afin d'entraîner une application de Deep Learning à identifier avec précision des cellules cancéreuses [3].
- Automatisation industrielle : Le Deep Learning participe à l'amélioration de la sécurité des employés travaillant à proximité d'équipements lourds, en détectant automatiquement les situations dans lesquelles la distance de sécurité qui sépare le personnel ou les objets des machines est insuffisante [3].
- **Électronique** : Le Deep Learning est utilisé pour la reconnaissance audio et vocale. Par exemple, les appareils d'assistance à domicile qui répondent à votre voix et connaissent vos préférences fonctionnent grâce à des applications de Deep Learning [3].

3.3 Les points forts et les points faibles du DL Les points forts

3.3.1 Les points fort

- **De meilleurs résultats qu'avec d'autres méthodes d'apprentissage machine :**

Le plus grand point fort du Deep learning reste la qualité des résultats obtenus. Dans des secteurs tels que le traitement d'images ou la reconnaissance d'images.

- Une exécution efficace des tâches de routine, sans écarts de qualité : Parce que basé sur un apprentissage routinier, ne montrant jamais aucun signe de fatigue et avec une qualité constante, celle-ci est beaucoup plus efficace et rapide que n'importe quelle autre méthode. Puisque le système se forme de façon autonome (après une

phase d'instruction initiale), il permet d'économiser beaucoup de temps et d'argent tout en garantissant un développement de ses fonctionnalités [3].

- Le traitement des données non structurées De plus, et contrairement à d'autres moteurs d'intelligence artificielle, l'apprentissage profond est capable d'analyser des données stockées sous un format non structuré (documents, photos, mails, etc.). De ce fait, il a une force de frappe différente et potentiellement plus intéressante que les technologies limitées à l'analyse des données structurées (numéros de téléphone, carte de crédit, adresses, etc.) [3].

3.3.2 Les points faibles

Si le Deep Learning a beaucoup d'avantages, il a aussi ses limites, parmi lesquelles

- **Le Deep Learning nécessite une grande puissance de calcul**

Un énorme besoin en puissance de calcul. D'une part pour assurer la maintenance de Réseaux de neurones artificiels, mais aussi pour traiter la très grande quantité de données nécessaire

- **Une technologie couteuse à mettre en place**

Cette puissance de calcul est relative à la complexité et à la difficulté de la tâche à résoudre et de la masse de données utilisée. De ce fait, le Deep learning se révèle être un système artificiel coûteux, donc plutôt réservé à la recherche et aux géants du Big Data [3].

- **Des décisions difficilement ou pas du tout compréhensibles**

Sur un autre point, l'une des problématiques que pose cette intelligence artificielle est la complexité et le volume de données que requièrent son fonctionnement et le fait qu'il est impossible de comprendre dans le détail la motivation des décisions qu'elle prend. Ce problème entraîne avec lui l'incapacité (pour le moment) de l'intégrer à des applications où la traçabilité est essentielle [3].

- **Il nécessite une vaste base de données**

Enfin, pour être efficace, le Deep Learning doit s'appuyer sur une grande quantité de données. Sans cela, aucune machine n'est en mesure de donner de bons résultats avec cette méthode. Bien qu'il existe des bibliothèques de réseaux neuronaux artificiels mis à disposition de tous pour simplifier son utilisation, il existe bel et bien des limites à la mise en

place d'une intelligence profonde, notamment le temps nécessaire à l'élaboration des algorithmes d'apprentissage [10].

4. Réseaux de neurones

Les réseaux de neurones (Neural Networks) est l'un des algorithmes d'apprentissage automatique les plus populaires à l'heure actuelle.

Au fil du temps, il a été prouvé de manière décisive que les réseaux de neurones surpassent les autres algorithmes en termes de précision et de rapidité [5].

Avec diverses variantes telles que CNN (Réseaux de Neurones Convolutionnels (abrégé en CNN)), RNN (Réseaux de Neurones Récurrents), Auto-Encodeurs, etc…, les réseaux de neurones deviennent peu à peu pour les scientifiques ou les praticiens de l'apprentissage automatique, ce que la régression linéaire était pour les statisticiens.

Les réseaux de neurones profonds (CNN) ont connu un succès considérable dans la reconnaissance et la localisation d'objets dans des images. L'approche fondamentale qui a conduit aux CNN consiste à construire des systèmes artificiels basés sur le cerveau et la vision humaine. Pourtant, sous de nombreux aspects importants, les capacités des CNN sont inférieures à celles de la vision humaine. Un axe de recherche prometteur consiste à étudier les similitudes et les différences en comblant les lacunes, pour améliorer les CNN [5]. Il existe plusieurs algorithmes de Deep Learning par exemple :

- Les réseaux neuronaux convolutifs (CNN)
- Les réseaux de mémoire à long terme et à court terme (LSTM)
- Les réseaux antagonistes génératifs (GAN)
- Les réseaux à fonctions de base radiale (RBFN)
- Machines de Boltzmann restreintes (RBM)
- Les réseaux de croyance profonds (DBN)
- Les perceptrons multicouches (MLP)
- Les cartes auto-organisées (SOM)
- Les auto-encodeurs

4.1 RNN les réseaux de neurones naturels

L'idée derrière les RNN est d'utiliser des informations séquentielles. Dans un réseau neuronal traditionnel, nous supposons que toutes les entrées (et les sorties) sont indépendantes les unes des autres. Mais pour de nombreuses tâches, Si on veut prédire le prochain mot dans une phrase, il faut connaître les mots qui sont venus avant. Les RNN sont appelés récurrents car ils exécutent la même tâche pour chaque élément d'une séquence, la sortie étant dépendante des calculs précédents. Une autre façon de penser les RNN est qu'ils ont une « mémoire » qui capture l'information sur ce qui a été calculé jusqu'ici. En théorie, les RNN peuvent utiliser des informations dans des séquences arbitrairement longues, mais dans la pratique, on les limite à regarder seulement quelques étapes en arrière [9].

4.2 Réseaux de neurones artificiels

Les réseaux de neurones artificiels sont simplement des systèmes inspirés du Fonctionnement des neurones biologiques.

Le plus célèbre d'entre eux est le perceptron multicouche, un système artificiel capable d'apprendre par l'expérience ! Introduit en 1957 parFranck Rosenblatt, il n'est véritablement utilisé que depuis 1982 après son perfectionnement.

Grâce à la puissance de calcul des années 2000, le perceptron s'est largement démocratisé et estde plus en plus utilisé.

La première description formelle des réseaux des neurones artificiels a été proposée par McCulloch & Pitts [21].

Ces systèmes sont des algorithmes prenant plusieurs valeurs en entrée, ces valeurs sont traités par plusieurs fonctions et en sortit retourne une valeur. Ces fonctions passent d'abord par une phase d'apprentissage dans le but de calibrer les résultats en sortie. On donne à un réseau des valeurs d'entrée, on connaît le résultat de sortie et on vérifie que le réseau retourne le résultat attendu, tant que ce n'est pas le cas on continu de faire des tests jusqu'à ce que le réseau soit correctement configuré et qu'il soit en mesure de répondre systématiquement le résultat attendu. Ensuite il agit comme une boîte noire, on lui donnera des valeurs d'entrée dont on ne connaît pas encore le résultat et il nous donnera une valeur en sortie sans qu'on ne sache ce qu'il a fait. Ainsi un réseau ne donne pas forcément de règle exploitable par un humain.

L'apprentissage par l'expérience permet ensuite d'utiliser les réseaux neuronaux pour de

la reconnaissance d'image, de la prédiction boursière, la résolution de problèmes mathématiques,etc [12].

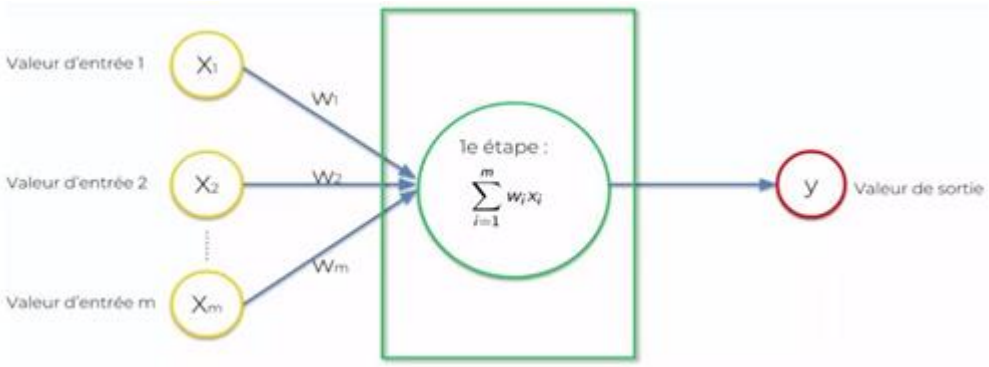


Figure 4 : exemple d'un réseau de neurones artificiel [12]

4.3 Les réseaux de neurones profonds (CNN)

4.3.1 Définition

Est un type de réseau de neurones artificiels acycliques dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux. Ils sont très utilisés dans le domaine de la vision par ordinateur. Et ont aussi beaucoup de succès dans les reconnaissances faciales, la détection d'objets très utilisée dans les robots et les voitures automatiques. En gros, tout ce qui concerne la vision par ordinateur et les images. De plus on peut utiliser les convNet dans tous les problèmes ayant en entrée une matrice. Par exemple, Gehring a utilisé une matrice de texte dans une tache de traduction automatique de langue. [22] On note que le terme "convolutional" vient de l'opération de convolution de matrices utilisée dans le traitement de signal. Deux nouveaux types de couche ont été ajoutés dans le réseau: la couche convolution (convolutional layer) et la couche de mise en commun (pool layer). [23]

4.3.2 Architecture

Les réseaux de neurones à La conception des réseaux de neurones convolutifs suit la découverte de mécanismes visuels dans les organismes vivants. Début 1968, des travaux ont montré chez l'animal que le cortex visuel contient des arrangements complexes de cellules, responsables de la détection de la lumière dans les sous-régions du champ visuel qui se chevauchent, appelés champs réceptifs. Le document a identifié deux types de cellules de base : les cellules simples, qui répondent à des pics caractéristiques (grand contraste, forte intensité...) à l'intérieur de leur champ récepteur ; et les cellules complexes, qui ont des

champs récepteurs plus grands et sont localement invariantes à la position exacte du motif. [25] ils sont basés sur le perceptron multicouche(MLP), et inspirés du comportement du cortex visuel des vertébrés. Bien qu'efficaces pour le traitement d'images, les MLP ont beaucoup de mal à gérer des images de grande taille, ce qui est dû à la croissance exponentielle du nombre de connexions avec la taille de l'image.

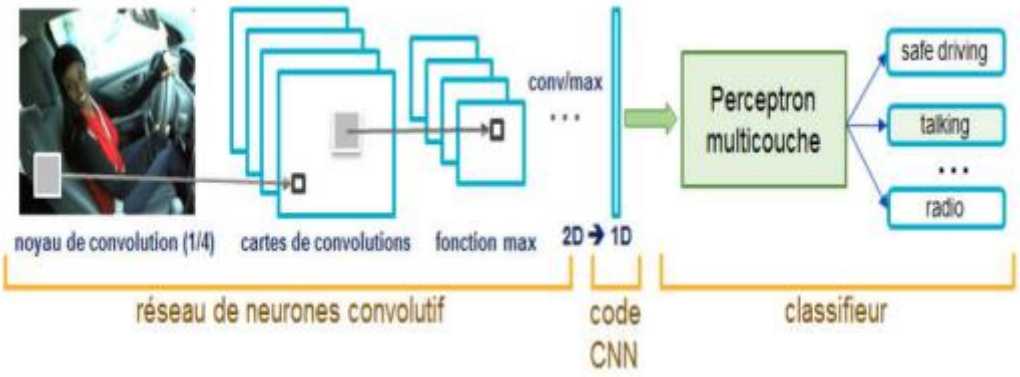


Figure 5 : Architecture standard d'un réseau de neurone convolutionnel [25]

1- Le premier bloc

Le premier bloc fait la particularité de ce type de réseaux de neurones, puisqu'il fonctionne comme un extracteur de features. Pour cela, en appliquant des opérations de filtrage par convolution. La première couche filtre l'image avec plusieurs noyaux de convolution, et renvoie des "featuremaps", qui sont ensuite normalisées (avec une fonction d'activation) et/ou redimensionnées. Ce procédé peut être réitéré plusieurs fois : on filtre les featuresmaps obtenues avec de nouveaux noyaux, ce qui nous donne de nouvelles featuresmaps à normaliser et redimensionner, et qu'on peut filtrer à nouveau, et ainsi de suite. Finalement, les valeurs des dernières featuremaps sont concaténées dans un vecteur. Ce vecteur définit la sortie du premier bloc, et l'entrée du second [4].

2- Le second bloc

Les valeurs du vecteur en entrée sont transformées (avec plusieurs combinaisons linéaires et fonctions d'activation) pour renvoyer un nouveau vecteur en sortie. Ce dernier vecteur contient autant d'éléments qu'il y a de classes : l'élément i représente la probabilité que l'image appartienne à la classe i Comme pour les réseaux de neurones ordinaires, les paramètres des couches sont déterminés par rétropropagation du gradient : l'entropie croisée est minimisée lors de la phase d'entraînement. Mais dans le cas des CNN, ces paramètres désignent en particulier les features des images [4].

Maintenant les différents types de couches d'un CNN. Il existe quatre types de couches pour un réseau de neurones convolutif : la couche de convolution, la couche de pooling, la couche de correction ReLU et la couche fully-connected [4].

4.3.3 Couche convolution

La couche de convolution est le bloc de construction de base d'un CNN.

Trois paramètres permettent de dimensionner le volume de la couche de convolution la profondeur le pas et la marge [26].

1. 'Profondeur' de la couche: nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).

2. 'Le pas' contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand [2].

3. 'La marge (à 0)' ou 'zero padding ' : Cette marge permet de contrôler la dimension spatiale du volume de sortie. En particulier, il est parfois souhaitable de conserver la même surface que celle du volume d'entrée

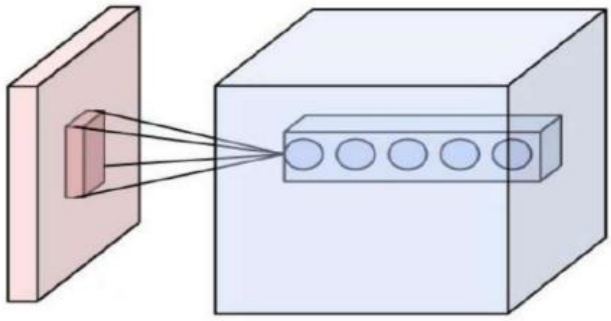


Figure 6 : Ensemble de neurones (cercles) créant la profondeur d'une couche de convolution (bleu). Ils sont liés à un même champ récepteur (rouge) [26]

La convolution, d'un point de vue simpliste, est le fait d'appliquer un filtre mathématique à une image. D'un point de vue plus technique, il s'agit de faire glisser une matrice par-dessus une image, et pour chaque pixel, utiliser la somme de la multiplication de ce pixel par la valeur de la matrice. Cette technique nous permet de trouver des parties de l'image qui pourraient nous être intéressantes [4].

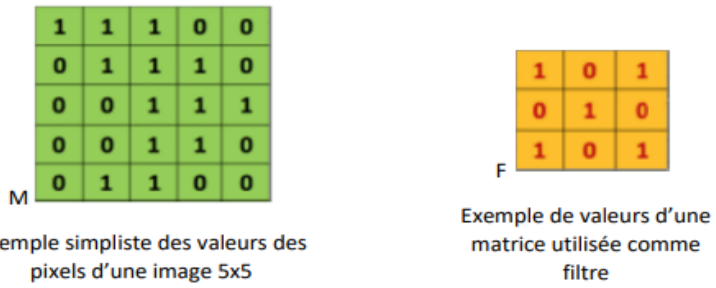


Figure 7 : exemples des matrices qui représentent des pixels d'une image [4]

Dans le cas de la Figure 7, les valeurs sont binaires. Dans un cas réel, les valeurs devraient varier entre 0 et 255. Dans la Figure 8, les valeurs sont représentées par des 1 et 0. Dans un cas réel, ces valeurs sont continuées et peuvent être positives ou négatives.

Appliquer le filtre sur l'image : dans la matrice image M, nous pouvons voir que chaque valeur des pixels de l'image tuile (les cases orange) est multipliée par chaque valeur correspondante du filtre (1x1, 1x0, 1x1 ….). Puis additionner tous ces valeurs pour obtenir une seule valeur '4' qui fera partie d'une nouvelle image convoluée [4].

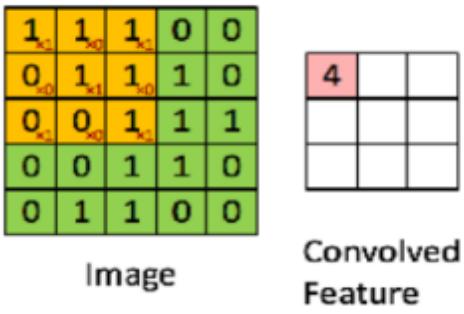


Figure 8 : L'APPLICATION DU FILTRE SUR L'IMAGE [4].

Le filtre doit se déplacer d'une case à chaque itération jusqu'à ce que la première ligne soit finie. Lorsque nous avons fini la première ligne, le filtre « descend » d'une case et la même procédure se répète pour chaque ligne et colonne [4].

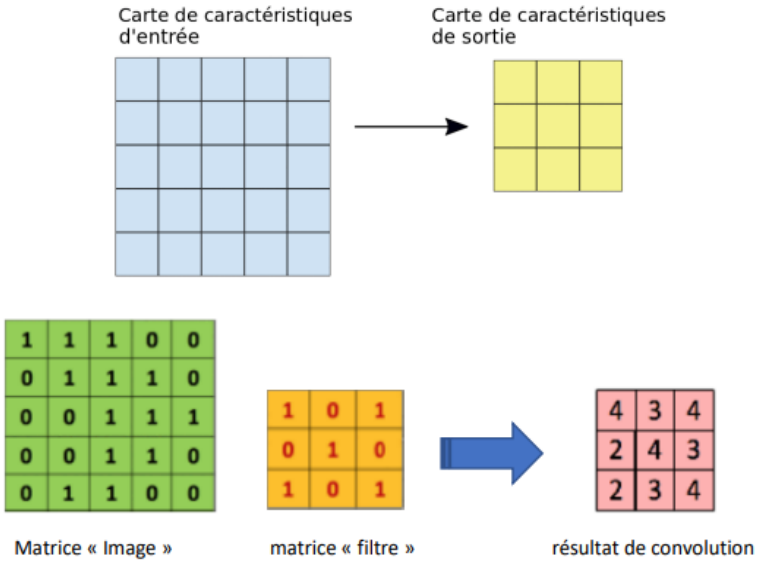


Figure 9 : image qui montre le résultat du filtre appliquée [4].

Noté qu'une convolution 3x3 de profondeur 1 effectuée sur une carte de caractéristiques d'entrée 5x5, également de profondeur 1. Comme il y a neuf emplacements 3x3 possibles pour extraire les tuiles de la carte de caractéristiques 5x5, cette convolution génère une carte de caractéristiques de sortie 3x3. Un réseau de neurones à convolution contient de multiples filtres et ces filtres sont appliqués sur l'image d'origine. Après la première étape nous avons donc autant de nouvelles images que de filtres.

La phase de convolution peut aussi être vue comme des couches de neurones cachées où chaque neurone n'est connecté qu'à quelques neurones de la couche suivante [4].

4.3.4 Pooling

4.3.4.1 Définition

Un autre concept important des CNNs est le pooling, ce qui est une forme de sous-échantillonnage de l'image. L'image d'entrée est découpée en une série de rectangles de n pixels de côté ne se chevauchant pas (pooling). Chaque rectangle peut être vu comme une tuile. Le signal en sortie de tuile est défini en fonction des valeurs près par les différents pixels de la tuile [4].

Le pooling réduit la taille spatiale d'une image intermédiaire, réduisant ainsi la quantité de paramètres et de calcul dans le réseau. Il est donc fréquent d'insérer périodiquement une

couche de pooling entre deux couches convolutées successives d'une architecture CNN pour contrôler l'overfitting (sur apprentissage) [4].

4.3.4.2 Principe

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling. L'opération de pooling consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes. Pour cela, on découpe l'image en cellules régulière, puis on garde au sein de chaque cellule la valeur maximale. La méthode utilisée consiste à imaginer une fenêtre de 2 ou 3 pixels qui glisse au-dessus d'une image, comme pour la convolution. Mais, cette fois-ci, nous faisons des pas de 2 pour une fenêtre de taille 2, et des pas de 3 pour 3 pixels. La taille de la fenêtre est appelée « kernel size » et les pas s'appellent «strides » Pour chaque étape, nous prenons la valeur la plus haute parmi celles présentes dans la fenêtre et cette valeur constitue un nouveau pixel dans une nouvelle image. Ceci s'appelle Max Pooling [4].

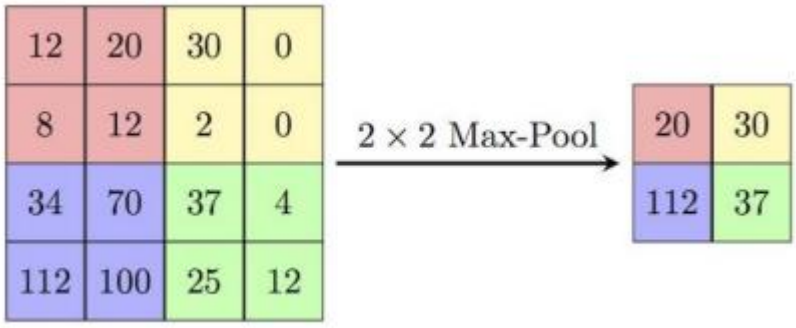


Figure 10 : figure qui montre le max pooling [4].

La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage. Les valeurs maximales sont repérées de manière moins exacte dans les feature maps obtenues après pooling que dans celles reçues en entrée – c'est en fait un grand avantage ! En effet, lorsqu'on veut reconnaître un chien par exemple, ses oreilles n'ont pas besoin d'être localisées le plus précisément possible : savoir qu'elles se situent à peu près à côté de la tête suffit ! Ainsi, la couche de pooling rend le réseau moins sensible à la position des features : le fait qu'une feature se situe un peu plus en haut ou en bas, ou même qu'elle ait une

orientation légèrement différente ne devrait pas provoquer un changement radical dans la classification de l'image [4].

4.3.5 stirde

Lors du calcul de la corrélation croisée, nous commençons par la fenêtre de convolution dans le coin supérieur gauche du tenseur d'entrée, puis la glissons sur tous les emplacements à la fois vers le bas et vers la droite. Dans les exemples précédents, nous glissons par défaut un élément à la fois. Cependant, parfois, soit pour l'efficacité des calculs, soit parce que nous souhaitons sous-échantillonner, nous déplaçons notre fenêtre de plus d'un élément à la fois, en sautant les emplacements intermédiaires.

Nous appelons le nombre de lignes et de colonnes parcourues par diapositive la foulée. Jusqu'à présent, nous avons utilisé des foulées de 1, à la fois pour la hauteur et la largeur. Parfois, nous pouvons vouloir utiliser une foulée plus large. La figure 11 montre une opération de corrélation croisée bidimensionnelle avec une foulée de 3 verticalement et de 2 horizontalement. Les parties ombrées sont les éléments de sortie ainsi que les éléments de tenseur d'entrée et du noyau utilisés pour le calcul de sortie.

Nous pouvons voir que lorsque le deuxième élément de la première colonne est sorti, la fenêtre de convolution glisse sur trois lignes. La fenêtre de convolution glisse de deux colonnes vers la droite lorsque le deuxième élément de la première ligne est sorti. Lorsque la fenêtre de convolution continue de glisser deux colonnes vers la droite sur l'entrée, il n'y a pas de sortie car l'élément d'entrée ne peut pas remplir la fenêtre (sauf si nous ajoutons une autre colonne de remplissage). [27]

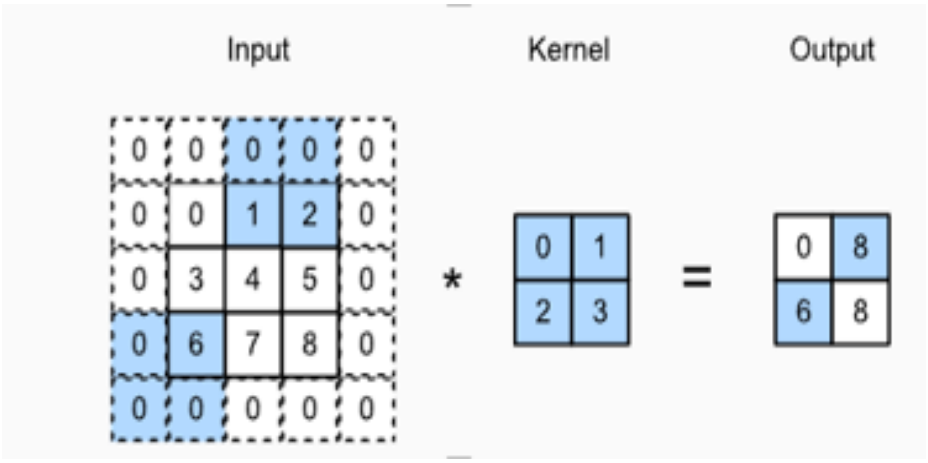


Figure 11 : Inter correlation avec des foulées de 3 et 2 pour la hauteur et la largeur, respectivement.

4.3.6 padding

Comme décrit ci-dessus, un problème délicat lors de l'application de couches convolutives est que nous avons tendance à perdre des pixels sur le périmètre de notre image. Comme nous utilisons généralement de petits noyaux, pour une convolution donnée, nous ne perdons peut-être que quelques pixels, mais cela peut s'additionner lorsque nous appliquons de nombreuses couches de convolution successives. Une solution simple à ce problème consiste à ajouter des pixels de remplissage supplémentaires autour de la limite de notre image d'entrée, augmentant ainsi la taille effective de l'image. En règle générale, nous définissons les valeurs des pixels supplémentaires sur zéro. Les parties ombrées sont le premier élément de sortie ainsi que les éléments de tenseur d'entrée et de noyau utilisés pour le calcul de sortie : 0 x 0 + 0 x 1 + 0 x 2+ 0 x 3 = 0 [27].

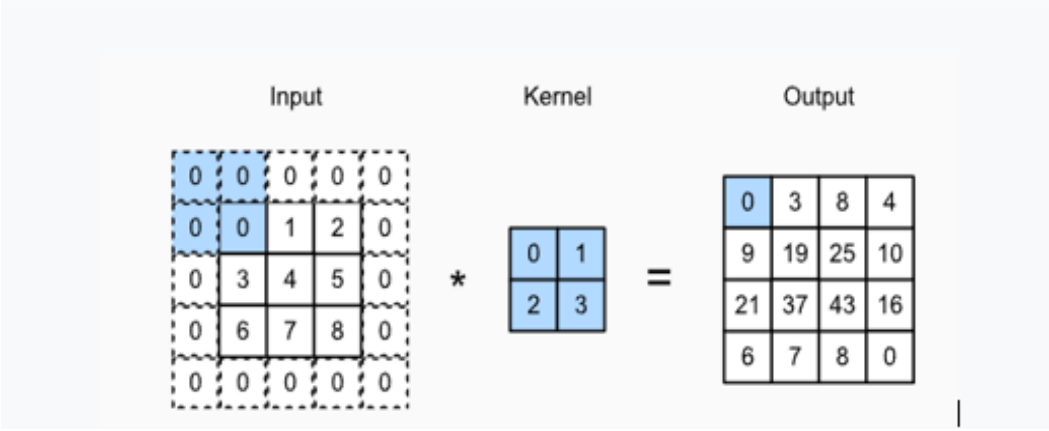


Figure 12 : Intercorrélation bidimensionnelle avec rembourrage.

4.3.7 Couche fully-connected

La couche fully-connected constitue toujours la dernière couche d'un réseau de neurones. Ce type de couche reçoit un vecteur en entrée et produit un nouveau vecteur en sortie. Pour cela, elle applique une combinaison linéaire puis éventuellement une fonction d'activation aux valeurs reçues en entrée. La dernière couche fully-connected permet de classifier l'image en entrée du réseau : elle renvoie un vecteur de taille N, où N est le nombre de classes dans notre problème de classification d'images. Chaque élément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe. Par exemple, si le problème consiste à distinguer les chats des chiens, le vecteur final sera de taille 2 : le premier élément donne la probabilité d'appartenir au classe "chat" (respectivement "chien"). Ainsi, le vecteur [0.9 0.1] signifie que l'image à 90% de chances de représenter un chat [4].

Pour calculer les probabilités, la couche fully-connected multiplie donc chaque élément en entrée par un poids, fait la somme, puis applique une fonction d'activation (logistique si N=2, softmax si N>2) : Ce traitement revient à multiplier le vecteur en entrée par la matrice contenant les poids. Le fait que chaque valeur en entrée soit connectée avec toutes les valeurs en sortie explique le terme fullyconnected [4].

4.3.8 Architecture d'un CNN

Un réseau de neurones à convolution peut avoir plusieurs étapes de convolution, ReLu et Pooling. Une règle à respecter est que la fonction de ReLu doit obligatoirement être appliquée après une étape de convolution afin d'avoir une réponse non-linéaire, mais le Pooling n'est pas obligatoire. Après être passé par toutes les étapes de convolution, Relu et Pooling, nous pouvons passer à la classification des images. La dernière phase consiste à envoyer tous les pixels dans un réseau de neurones multicouches. Étant donné que nous avons pu récupérer les parties les plus importantes d'une image que nous avons condensée, la phase de classification sera beaucoup plus performante qu'en utilisant un réseau de neurones artificiels sans convolution [4].

Figure 13 : Les activations d'un exemple d'architecture ConvNet[23].

Un CNN est simplement un empilement de plusieurs couches de convolution, pooling, correction ReLU et fully-connected. Chaque image reçue en entrée va donc être filtrée, réduite et corrigée plusieurs fois, pour finalement former un vecteur. Dans le problème de classification, ce vecteur contient les probabilités d'appartenance aux classes. Tous les réseaux de neurones convolutifs doivent commencer par une couche de convolution et finir

par une couche fully-connected. Les couches intermédiaires peuvent s'empiler de différentes manières, à condition que la sortie d'une couche ait la même structure que l'entrée de la suivante. Par exemple, une couche fully-connected, qui renvoie toujours un vecteur, ne peut pas être placée avant une couche de pooling, puisque cette dernière doit recevoir une matrice 3D. En général, un réseau de neurones empile plusieurs couches de convolution et de correction ReLU, ajoute ensuite une couche de pooling (facultative), et répète ce motif plusieurs fois ; puis, il empile des couches fully-connected. Plus il y a de couches, plus le réseau de neurones est "profond" : on est en plein dans le Deep Learning!

La première couche de convolution apprend des features simples, qui représentent des éléments de structure rudimentaires de l'image (contours, coins...) Plus les couches de convolution sont "hautes", c'est-à-dire loin de l'entrée du réseau, plus les features apprises sont complexes : celles-ci se composent des features plus simples des couches précédentes. Un carré est un exemple de feature complexe, formée de contours et de coins. Les couches de convolution les plus hautes apprennent donc des features sophistiquées : par exemple couche convolution 2, dans le cas de la reconnaissance de chat ci-dessous, elles peuvent correspondre aux oreilles, nez ou l'œil [4].

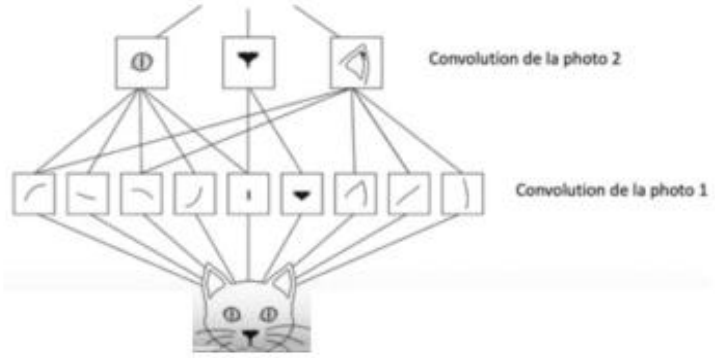


Figure 14 : Examples des couches de convolution [4].

5. Hyper paramètres in deeplearning

Que sont les hyper paramètres ?

Les hyper paramètres sont les variables qui déterminent la structure du réseau (par exemple : nombre d'unités cachées) et les variables qui déterminent la manière dont le réseau est entraîné (par exemple : taux d'apprentissage). Les hyper paramètres sont définis avant l'entraînement (avant d'optimiser les poids et les biais). Hyper paramètres liés à la structure du réseau Nombre de couches cachées et d'unités.

Les couches masquées sont les couches entre la couche d'entrée et la couche de sortie.

De nombreuses unités cachées dans une couche avec des techniques de régularisation peuvent augmenter la précision. Un plus petit nombre d'unités peut entraîner un sous-ajustement. [28]

5.1 Nombre d'époques

Le nombre d'époques correspond au nombre de fois où toutes les données d'entraînement sont affichées sur le réseau pendant l'entraînement.

Augmentez le nombre d'époques jusqu'à ce que la précision de la validation commence à diminuer même lorsque la précision de l'entraînement augmente (sur ajustement). [28]

5.2 Taille du lot

La taille du mini lot est le nombre de sous-échantillons donnés au réseau après quoi la mise à jour des paramètres se produit. Une bonne valeur par défaut pour la taille du lot pourrait être 32 [28]

6. Les fonctions de deeplearning

6.1 Fonctions de perte

Une tâche de prédiction peut être considérée comme une simple tâche d'optimisation. Le modèle tente d'optimiser ses performances en prédisant la valeur correcte. La valeur réelle que nous optimisons, s'appelle la «perte» (que nous essayons de minimiser, bien sûr). Pour ce faire, le modèle doit pouvoir mesurer ce que l'on appelle la «perte», qui dépend du problème.

Généralement, la fonction de perte, L, est une fonction sur l'entrée, un ensemble de paramètres (appelés "poids") et le vrai label (Dans la famille multi-classe, Li est défini comme la perte sur la classe i).

R est une fonction de régularisation. Cette fonction est utilisée pour pénaliser W "complexe" (par exemple, elle "préférera" w plus petit). Cela force le modèle à préférer les modèles plus simples aux plus complexes.

Lambda, le coefficient de R, est un autre paramètre que ce processus optimisé. [29]

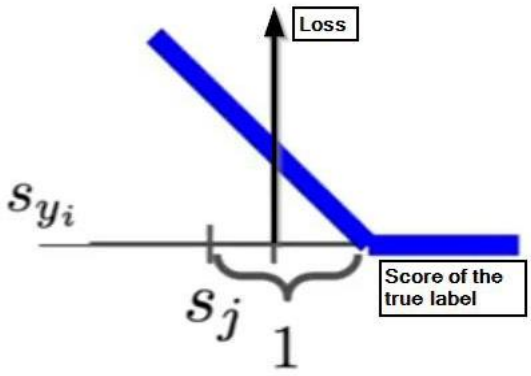


Figure 15 : fonction de la perte

6.2 Fonction de coût

C'est une fonction qui détermine les performances d'un modèle d'apprentissage automatique pour un ensemble de données donné. La fonction de coût calcule la différence entre les valeurs anticipées et attendues et l'affiche sous la forme d'un nombre réel unique. Les fonctions de coût peuvent être créées selon diverses méthodes en fonction de la situation.

Pour estimer les performances médiocres des modèles, des fonctions de coût sont utilisées. En termes simples, une fonction de coût est une mesure de l'imprécision du modèle dans l'estimation du lien entre X et y. Ceci est généralement indiqué comme une différence ou une séparation entre les valeurs attendues et réelles.

Le terme « perte » dans l'apprentissage automatique fait référence à la différence entre la valeur anticipée et la valeur réelle. La "Loss Function" est une fonction qui permet de quantifier cette perte sous la forme d'un nombre réel unique lors de la phase d'apprentissage. Ceux-ci sont utilisés dans des algorithmes qui appliquent des approches d'optimisation dans l'apprentissage supervisé.

La régression, la régression logistique et d'autres algorithmes sont des instances de ce type. Les expressions « fonction de coût » et « fonction de perte » sont interchangeables. Le but de la fonction de coût est d'être soit :

Minimum - Lorsqu'une valeur est réduite à sa forme la plus simple, on parle de coût, de perte ou d'erreur. L'objectif est d'identifier les réglages des paramètres du modèle pour lesquels la fonction de coût donne le plus petit nombre possible.

Maximum - Lorsque quelque chose est maximisé, la valeur qu'il produit est appelée récompense. L'objectif est de découvrir les valeurs des paramètres du modèle avec un nombre renvoyé aussi grand que possible. [30]

7. La descente de gradient (GD)

Est un algorithme d'optimisation itératif du premier ordre utilisé pour trouver un minimum/maximum local d'une fonction donnée. Cette méthode est couramment utilisée dans l'apprentissage automatique (ML) et l'apprentissage profond (DL) pour minimiser une fonction de coût/perte (par exemple dans une régression linéaire). En raison de son importance et de sa facilité de mise en œuvre, cet algorithme est généralement enseigné au début de presque tous les cours d'apprentissage automatique. [31]

Cependant, son utilisation ne se limite pas uniquement au ML/DL, il est également largement utilisé dans des domaines tels que :

Ingénierie de contrôle (robotique, chimique, etc.)

- Jeux d'ordinateur
- Génie mécanique

C'est pourquoi aujourd'hui, nous allons nous plonger dans les mathématiques, la mise en œuvre et le comportement de l'algorithme de descente de gradient de premier ordre. Nous naviguerons directement dans la fonction personnalisée (coût) pour trouver son minimum, il n'y aura donc pas de données sous-jacentes comme dans les didacticiels ML typiques - nous serons plus flexibles en termes de forme de fonction.

Cette méthode a été proposée avant l'ère des ordinateurs modernes et il y a eu un développement intensif entre-temps qui a conduit à de nombreuses versions améliorées de celle-ci mais dans cet article, nous allons utiliser une descente de gradient basique/vanille implémentée en Python. [31]

8. Matrice de confusion

8.1 Définition

Une matrice de confusion est une technique permettant de résumer les performances d'un algorithme de classification. [32]

Une matrice de confusion est un résumé des résultats de prédiction sur un problème de classification.

Le nombre de prédictions correctes et incorrectes est résumé avec des valeurs de comptage et ventilé par classe. C'est la clé de la matrice de confusion.

La matrice de confusion montre les façons dont votre modèle de classification

Est confus quand il fait des prédictions.

Il vous donne un aperçu non seulement des erreurs commises par votre classificateur, mais surtout des types d'erreurs commises.

C'est cette ventilation qui surmonte la limitation de l'utilisation de la précision de la classification seule. [32]

8.2 Métriques de la matrice de confusion

Lors de l'exécution d'un modèle de classification, le résultat obtenu est généralement un résultat binaire 0 ou 1, 0 signifiant Faux et 1 signifiant Vrai. Nous pouvons comparer nos résultats de classification résultants avec nos valeurs réelles de l'observation donnée pour juger de la performance du modèle de classification. La matrice utilisée pour refléter ces résultats est connue sous le nom de matrice de confusion et peut être vue ci-dessous :

Tableau 1 : tableau qui montre la matrice de confusion [70]

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Il y a quatre résultats potentiels ici : Vrai positif (TP) indique que le modèle a prédit un résultat vrai et que l'observation réelle était vraie. Un faux positif (FP) indique que le modèle a prédit un vrai résultat, mais que l'observation réelle était fausse. Faux négatif (FN) indique que le modèle a prédit un faux résultat, alors que l'observation réelle était vraie. Enfin, nous avons le vrai négatif (TN), qui indique que le modèle a prédit un résultat faux, alors que le résultat réel était également faux.

Les matrices de confusion peuvent être utilisées pour calculer les mesures de performance des modèles de classification. Parmi les nombreuses mesures de performance utilisées, les plus courantes sont l'exactitude, la précision, le rappel et le score F1 [70].

La formule de calcul de la précision, basée sur le tableau ci-dessus, est (TP+TN)/ (TP+FP+FN+TN) ou tous les cas vrais positifs et vrais négatifs divisés par le nombre de tous les cas. La précision est couramment utilisée pour juger des performances du modèle, cependant, il y a quelques inconvénients qui doivent être pris en compte avant d'utiliser la précision de manière généreuse. L'un de ces inconvénients concerne les ensembles de données déséquilibrés où une classe, vraie ou fausse, est plus courante que l'autre, ce qui oblige le modèle à classer les observations en fonction de ce déséquilibre. Par exemple, si 90 % des cas sont faux et seulement 10 % sont vrais, il y a une très forte possibilité que notre modèle ait un score de précision d'environ 90 %. Naïvement, il peut sembler que nous ayons un taux de précision élevé, mais en réalité, nous sommes à seulement 90 % susceptibles de prédire la classe "faux", donc nous n'avons pas vraiment une bonne métrique. Normalement, je n'utiliserais pas la précision comme mesure de performance, j'utiliserais plutôt la précision, le rappel ou le score F1 [70].

8.2.1 Précision (accuracy)

La précision est la mesure des vrais positifs par rapport au nombre total de positifs prédits par votre modèle. La formule de précision peut s'écrire : TP/ (TP+FP). Ce que cette métrique vous permet de calculer, c'est le taux auquel vos prédictions positives sont réellement positives [70].

8.2.2 Rappeler (recall)

Le rappel (c'est-à-dire la sensibilité) est la mesure de votre vrai positif par rapport au nombre de résultats positifs réels. La formule de rappel peut être exprimée comme suit : TP/(TP+FN).

En utilisant cette formule, nous pouvons évaluer dans quelle mesure notre modèle est capable d'identifier le vrai résultat réel [70].

8.2.3 F1 score

Le score F1 est la moyenne harmonique entre la précision et le rappel. La formule du score F1 peut être exprimée comme suit : 2(p*r)/ (p+r) où "p" est la précision et "r" le rappel. Ce score peut être utilisé comme une mesure globale qui intègre à la fois la précision et le rappel. La raison pour laquelle nous utilisons la moyenne harmonique par opposition à la moyenne régulière est que la moyenne harmonique punit les valeurs les plus éloignées.

Un exemple de ceci pourrait être vu où p = 0,4 et r = 0,8. En utilisant notre formule, nous voyons que 2(0.4*0.8)/ (0.4+0.8), qui se simplifie en 0.64/1.20=0.533 ; alors que la moyenne normale serait juste (0.4+0.8)/2=0.6

Il existe de nombreuses mesures que l'on pourrait utiliser pour déterminer les performances de leur modèle de classification [70].

9. transfer learning

La réutilisation d'un modèle préformé sur un nouveau problème est connue sous le nom d'apprentissage par transfert dans l'apprentissage automatique. Une machine utilise les connaissances acquises lors d'une tâche précédente pour augmenter la prédiction d'une nouvelle tâche dans l'apprentissage par transfert. Vous pouvez, par exemple, utiliser les informations obtenues lors de la formation pour distinguer les boissons lors de la formation d'un classificateur pour prédire si une image contient de la cuisine.

La connaissance d'un modèle d'apprentissage automatique déjà formé est transférée à un problème différent mais étroitement lié tout au long de l'apprentissage par transfert. Par exemple, si vous avez entraîné un classificateur simple pour prédire si une image contient un sac à dos, vous pouvez utiliser les connaissances d'entraînement du modèle pour identifier d'autres objets tels que des lunettes de soleil. [33]

Les pondérations sont automatiquement transférées vers un réseau exécutant la « tâche A » à partir d'un réseau qui a effectué une nouvelle « tâche B ».

En raison de l'énorme quantité de puissance CPU requise, l'apprentissage par transfert est généralement appliqué dans la vision par ordinateur et les tâches de traitement du langage naturel telles que l'analyse des sentiments. [33]

10. Comment fonctionne l'apprentissage par transfert

En vision par ordinateur, les réseaux de neurones visent généralement à détecter les contours dans la première couche, les formes dans la couche intermédiaire et les caractéristiques spécifiques à une tâche dans les dernières couches. Les couches précoce et centrale sont employées dans l'apprentissage par transfert, et les dernières couches ne sont que recyclées. Il utilise les données étiquetées de la tâche sur laquelle il a été formé. [33]

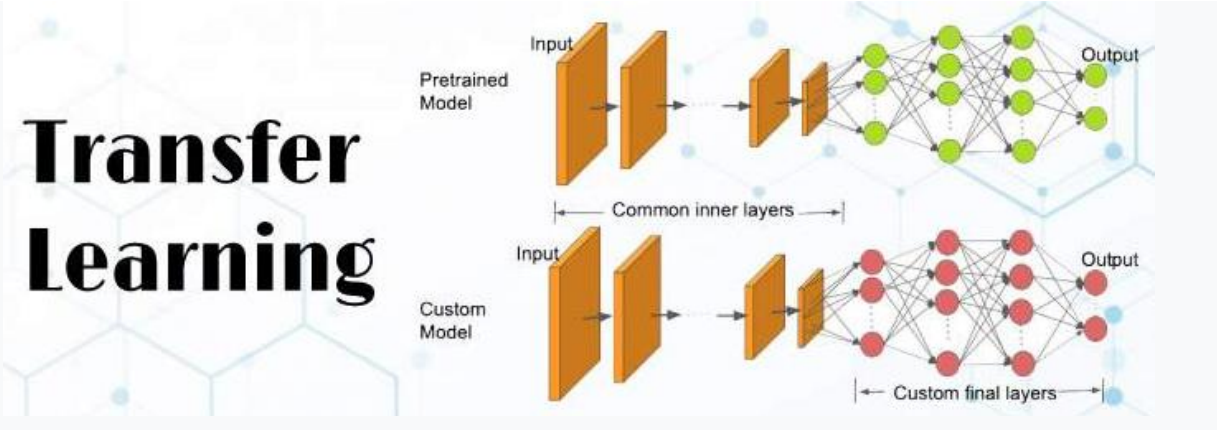


Figure 16 : figure qui montre le transfer learning

11. Le paramétrage des couches

Un réseau de neurones convolutif se distingue d'un autre par la façon dont les couches sont empilées, mais également paramétrées. Les couches de convolution et de pooling possèdent en effet des hyper paramètres, c'est-à-dire des paramètres dont vous devez préalablement définir la valeur [4].

Les features de la couche de convolution et les poids de la couche fullyconnected ne sont pas des hyperparamètres, puisqu'ils sont appris par le réseau de neurones lors de la phase d'entraînement. La taille des feature maps en sortie des couches de convolution et de pooling dépend des hyperparamètres [4].

Chaque image (ou feature map) est de dimensions W × H × D où W est sa largeur en pixels, H sa hauteur en pixels et D le nombre de canaux (1 pour une image en noir et blanc, 3 pour une image en couleurs).

La couche de convolution possède quatre hyperparamètres :

1. Le nombre de filtres K

2. La taille F des filtres : chaque filtre est de dimensions F×F×D pixels.

3. Le pas S avec lequel on fait glisser la fenêtre correspondant au filtre sur l'image. Par exemple, un pas de 1 signifie qu'on déplace la fenêtre d'un pixel à la fois.

4. Le zero-padding P : on ajoute à l'image en entrée de la couche un contour noir d'épaisseur P pixels. Sans ce contour, les dimensions en sortie sont plus petites. Ainsi, plus on empile de couches de convolution avec P=0, plus l'image en entrée du réseau rétrécit. On perd donc beaucoup d'informations rapidement, ce qui rend la tâche d'extraction de features difficile [4].

La couche de pooling présente seulement deux hyperparamètres :

• La taille F des cellules : l'image est découpée en cellules carrées de taille F×F pixels

• Le pas S : les cellules sont séparées les unes des autres de S pixels

Tout comme l'empilement, le choix des hyperparamètres se fait selon un schéma classique :

• Pour la couche de convolution, les filtres sont de petite taille et glissés sur l'image d'un pixel à la fois. La valeur du zero-padding est choisie de sorte que la largeur et la hauteur du volume en entrée ne soient pas modifiées en sortie. En général, on choisit alors F=3, P=1, S=1 ou F=5, P=2, S=1

• Pour la couche de pooling, F=2 et S=2 est un choix judicieux. Cela permet d'éliminer 75% des pixels en entrée. On peut également trouver F=3 et S=2 : dans ce cas, les cellules se chevauchent [4].

12. Conclusion

Nous avons présenté au cours de ce chapitre les notions importantes qui sont en relation avec l'apprentissage automatique et l'apprentissage profond. On a également cité les différents modes d'apprentissage, à savoir, supervisé, non- supervisé, En plus, nous avons détaillé les réseaux de neurones convolutifs

Chapitre 2

1. Introduction

La reconnaissance de l'écriture manuscrite (HTR) est un processus informatique visant à convertir un texte écrit en texte codé numériquement.

Vous devez faire la distinction entre deux perceptions différentes qui ont des problèmes et des solutions différents.

La reconnaissance de l'écriture manuscrite utilise la reconnaissance des formes, mais elle utilise également le traitement automatique du langage naturel. Cela signifie que, comme le cerveau humain, le système reconnaît des mots et des phrases qui existent dans des langues connues, plutôt que des séquences de lettres.

2. Reconnaissance en ligne et hors ligne

Ce sont deux modes différents d'OCR, ayant chacun ses outils propres d'acquisition et ses algorithmes correspondants de reconnaissance [12].

2.1 Systèmes en ligne

Ce mode de reconnaissance s'opère en temps réel (pendant l'écriture). Les symboles sont reconnus au fur et à mesure qu'ils sont écrits à la main. Ce mode est réservé généralement à l'écriture manuscrite. C'est une approche où la reconnaissance est effectuée sur des données à une dimension. L'écriture prend l'aspect d'un couple de signaux temporels numérisés [24]. La reconnaissance en ligne présente un avantage majeur c'est la possibilité de correction et de modification de l'écriture de manière interactive vu la réponse en continu du système [13]. L'acquisition de l'écrit est généralement assurée par une tablette graphique munie d'un styloélectronique.

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

2.2 Systèmes hors ligne :

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

l'informationest indépendante de la source de génération [16].

Il s'agit de données bien structurées dont la lecture nécessite la connaissance de la typographie et de la mise en page du document. Ici la démarche n'est plus un simple prétraitement, mais une démarche experte d'analyse de document il y'a localisation des régions, séparation des régions graphiques et photographique, étiquetage sémantique des zones textuelles à partir de modèles, détermination de l'ordre de lecture et de la structure du document [17].

Reconnaissance de l'imprimé ou du manuscrit :

Les approches diffèrent selon qu'il s'agisse de reconnaissance de caractères imprimés ou manuscrits. Les caractères imprimés sont dans le cas général alignés horizontalement et séparés verticalement, ce qui simplifie la phase de lecture [12].

La forme des caractères est définie par un style calligraphique (fonte) qui constitue un modèle pour l'identification. Dans le cas du manuscrit, les caractères sont souvent ligaturés et leur graphisme est inégalement proportionné provenant de la variabilité intra et inter scripteurs. Cela nécessite généralement l'emploi de techniques de délimitation spécifiques et souvent des connaissances contextuelles pour guider la lecture [12].

2.3 Critères d'influences

On classe souvent les méthodes de reconnaissance en fonction du mode d'acquisition de l'écriture.

- **L'écriture en ligne (ou dynamique)** est obtenue par une saisie en continue et se présente sous la forme d'une séquence de points ordonnée dans le temps. Dans ce cas, la donnée est de type signal et l'approche doit tirer profit de la représentation temporelle. L'analogie avec la reconnaissance de la parole est très fréquente et il n'est pas rare de voir des chercheurs appliquer des techniques issues de ce domaine.

- **L'écriture hors-ligne (ou en différé, ou encore statique)** est obtenue par la saisie d'un texte déjà existant, obtenue par un scanner ou une caméra. Dans ce cas, on dispose d'une image binaire ou en niveaux de gris, ayant perdu toute information temporelle sur l'ordre des points. De plus, ce mode introduit une difficulté supplémentaire relative à la variabilité du tracé en épaisseur et en connectivité, nécessitant l'application de techniques de prétraitement. [37]

2.4 Comparaison de la reconnaissance des caractères en ligne et hors-ligne

Tableau 2: Comparaison de la reconnaissance des caractères en ligne et hors- ligne [12]

Caractères en ligne	Caractères hors-ligne
Utilisation de stylos numériques	Utilisation de papier
Exigence d'échantillons	Exigence de points
Disponibilité de coup de stylo	Non disponible
Taux de reconnaissance élevé	Taux de reconnaissance bas
Grande précision	Faible précision

3. La reconnaissance de l'écriture arabe manuscrite

3.1 Présentation de la langue arabe

L'arabe littéral, arabe moderne unifié ou encore classique est le nom que l'on donne à une variante de la langue arabe, utilisée comme langue officielle dans tous les pays arabes, et comme langue commune entre pays arabes. Elle est également employée dans la plupart des écrits et, à l'oral, dans les situations officielles ou formelles (discours religieux, politiques, journaux télévisés).

L'arabe littéral se distingue ainsi de l'arabe dialectal, qui est la langue vernaculaire parlée au quotidien et ce depuis l'expansion de l'islam. Cette variété de la langue recouvre plusieurs dialectes locaux pouvant varier assez fortement d'un pays à l'autre. Dans tous les pays arabes, un dialecte national composé par plusieurs dialectes locaux est parlé. Aucun d'entre ces dialectes n'est identique complètement à l'arabe classique ou littéraire. Notons également que le farsi (persan), utilisé principalement en Iran et en Afghanistan, partage un grand nombre de points communs avec l'écriture arabe. [34]

3.2 Difficultés inhérentes à la reconnaissance de l'écriture arabe

3.2.1 Alphabet

L'alphabet arabe comporte 28 lettres (voir Tableau I.1) La forme des lettres dépend de leur position dans le mot. Certaines lettres prennent jusqu'a 4 formes différentes : par. (ﻫ ﻫ ﻫ ﻫ) ou (ع ﻋ ﻌ ﻉ) exemple Mais pour la plupart des lettres, les formes début/milieu et fin/isolé sont identiques à la ligature près. [35]

La présence d'une ligature avec la lettre précédente ou avec la lettre suivante ne modifie pas la forme de la lettre de manière significative (pas plus que dans l'écriture manuscrite cursive latine).[35]

En arabe, les ligatures se situent toujours au niveau de la ligne d'écriture, c'est-à-dire qu'il n'existe pas de lettre à liaison haute comme le 'o' ou le 'v' en alphabet latin, il existe toutefois des ligatures verticales. [35]

3.2.2 Signes diacritiques

Le terme ≪ signe diacritique ≫ peut porter à confusion : dans certains travaux, seules les voyelles arabes sont appelées diacritiques. Dans d'autres travaux, en revanche, tous les signes secondaires sont appelés diacritiques, qu'il s'agisse des voyelles, des points ou des autres signes (chaddah, hamzah, ...). [35]

C'est cette deuxième terminologie que nous employons ici : un signe diacritique est une composante secondaire d'une lettre, qui vient la compléter ou en modifier le sens.[35]

Dans la suite de cette thèse, les ≪ signes diacritiques ≫ désigneront à la fois points, voyelles et autres signes secondaires. [35]

3.2.3 Des points nécessaires pour différencier les lettres

Dans l'alphabet arabe, 15 lettres parmi les 28 possèdent un ou plusieurs points. Ces signes diacritiques sont situés soit au-dessus, soit en dessous de la forme à laquelle ils sont associés, mais jamais les deux à la fois.

Tableau 3 : tableau des caractères arabe

N°	A la fin	Au milieu	Au début	Isolée
1	ـى	ـا	ا	ا
2	ـب	ـبـ	بـ	ب
3	ـت	ـتـ	تـ	ت
4	ـث	ـثـ	ثـ	ث
5	ـج	ـجـ	جـ	ج
6	ـح	ـحـ	حـ	ح
7	ـخ	ـخـ	خـ	خ
8	ـد	ـد	د	د
9	ـذ	ـذ	ذ	ذ
10	ـر	ـر	ر	ر
11	ـز	ـز	ز	ز
12	ـس	ـسـ	سـ	س
13	ـش	ـشـ	شـ	ش
14	ـص	ـصـ	صـ	ص
15	ـض	ـضـ	ضـ	ض
16	ـط	ـطـ	طـ	ط
17	ـظ	ـظـ	ظـ	ظ
18	ـع	ـعـ	عـ	ع
19	ـغ	ـغـ	غـ	غ
20	ـف	ـفـ	فـ	ف
21	ـق	ـقـ	قـ	ق
22	ـك	ـكـ	كـ	ك
23	ـل	ـلـ	لـ	ل
24	ـم	ـمـ	مـ	م
25	ـن	ـنـ	نـ	ن
26	ـه	ـهـ	هـ	ه
27	ـو	ـو	و	و
28	ـي	ـيـ	يـ	ي

Tableau 4 : Points en arabe : un, deux ou trois points

La Figure Tableau 2 : illustre la variabilité des styles d'écriture des points ou groupes de points en écriture arabe manuscrite. Un groupe de deux points peut ainsi s'écrire sous forme d'une seule, ou de deux composantes connexes. [35]

4. Domaines d'application de la reconnaissance manuscrite

4.1 DOMAINES D'APPLICATIONS

Aucun système n'est actuellement capable de reconnaître l'écriture manuscrite de façon universelle comme peut le faire tout être humain lettré. Les systèmes existants se

répartissent en deux grandes classes de méthodes de reconnaissance : la reconnaissance **hors ligne** ou « statique » et la reconnaissance **en ligne** ou « dynamique » [36]

- Dans le cas **hors ligne**, il s'agit de reconnaître des textes manuscrits à partir de documents écrits au préalable. L'image du texte écrit est numérisée à l'aide d'un scanner, les informations recueillies se présentent sous la forme d'une image discrète constituée d'un ensemble de pixels. L'écriture prend l'aspect d'un signal spatial bidimensionnel numérisé. [36]
- Dans le cas **en ligne**, il s'agit de reconnaître l'écriture au fur et à mesure de son tracé. Le texte est saisi avec un stylo et une tablette à numériser, les informations recueillies sont constituées par une suite ordonnée de points (définis par leurs coordonnées) échantillonnés à cadence fixe. L'écriture prend l'aspect d'un couple de signaux temporels numérisés [36]

La reconnaissance **hors ligne** ne peut pas a priori s'appuyer sur l'information temporelle du tracé qui est perdue, mais elle peut tenir compte de l'épaisseur du tracé (les pleins et les déliés). La reconnaissance **en ligne** peut disposer de l'information temporelle (vitesse, accélération, levés de stylo, retours en arrière, barres de *t*, points diacritiques), mais d'aucune information sur l'épaisseur du tracé si on ne dispose pas d'un signal de pression de la pointe du stylet sur le support. [36]

4.2 QUELQUES EXEMPLES D'APPLICATIONS

Voici quelques exemples d'applications informatiques utilisant des systèmes de reconnaissance d'écriture:

1- prendre des notes avec un stylo électronique directement sur l'écran de son PDA ou Tablet PC et pouvoir les éditer avec un logiciel de traitement de texte.

2- rédiger des messages ou des mails à l'aide d'un stylo sur l'écran de son Smartphone.

3- saisir des formules mathématiques sur l'écran de son Tablet PC.

4- rétro convertir des documents papiers : par exemple pour numériser à l'aide d'un scanner un document manuscrit et le transformer en document éditable avec un logiciel de traitement de texte.

5- indexer des documents d'archives : ce sont souvent des documents abîmés et peu accessibles au public, les numériser et reconnaître certaines parties du document (par exemple le nom sur une fiche d'état civil) permet de faciliter les recherches.

6- trier automatiquement le courrier en analysant automatiquement des adresses postales sur des enveloppes.

7- traité des formulaires remplis à la main : comme par exemple un sondage, un bon de commande ou un chèque.

Ces différents exemples correspondent à l'analyse de documents plus ou moins complexes contenant de l'écriture manuscrite. Suivant les cas, la reconnaissance consiste à reconnaître des documents complets, des mots ou seulement des caractères dans une partie des documents.

5. Travaux connexes

Jusqu'à récemment, le problème de la reconnaissance de texte de bout en bout n'avait suscité qu'un intérêt modeste de la part de la communauté de la vision. Dans [6], les auteurs proposent une reconnaissance pipeline comprenant plusieurs boucles de rétroaction pour hypothèse validation. En bref, ils trouvent Maximalement Stable Extremal Régions (MSER), puis classez ces régions en tant que régions à caractère ou non à caractère. Ils forment des lignes possibles contenant du texte et utilisent la reconnaissance de caractères, suivie d'une modélisation typographique et linguistique pour affiner la sortie de reconnaissance. Dans [7], les auteurs utilisent Random Ferns pour la détection de caractères et le descripteur HOG [1] pour décrire les patchs d'images. Ils évaluent la détection des mots et performance de reconnaissance d'une approche en deux étapes comprenant un détecteur de texte et un moteur OCR. Ils montrent que leur système basé sur HOG surpasse celui utilisant un système conventionnel

Figure 17 : Exemple de sortie de notre système de bout en bout. Le cadre de délimitation rouge localise le texte contenant une partie de l'image.

Moteur OCR. Dans [8], les auteurs utilisent le score de sortie d'un réseau de neurones convolutifs (CNN) pour la détection de caractères et un autre CNN pour reconnaître les caractères détectés. Ils utilisent une approche de fenêtre glissante à plusieurs échelles et considérer les fenêtres dans différentes rangées indépendamment. Ils supposer que le texte de l'image apparaît horizontalement. Pour chaque ligne, les scores de détection des différents glissements les fenêtres sont calculées et NMS (suppression non max) est utilisé pour supprimer les pics parasites. De cette façon, les cadres de délimitation sont construits pour les lignes avec un nombre de pics non nul. Une autre NMS est effectuée pour élaguer les cadres de délimitation avec 50% de chevauchement ou plus. La reconnaissance de caractères CNN est puis utilisée sur chacun d'eux et une recherche de faisceau est effectuée pour mots de sortie. Ils supposent la connaissance d'un lexique de environ 20-30 mots (qui sont différents pour différentes images, et potentiellement achetés à l'aide de la géolocalisation et d'autres méta data), et ils ne sortent que des mots de ce lexique. Le La méthode utilisée dans [7] suppose également la connaissance d'un tel lexique.

6. Détection de texte arabe dans les vidéos

Contrairement au texte de la scène, les légendes ou le texte assimilé dans les vidéos ne se limitent pas obligatoirement à quelques mots. Il peut composer en une ou certains échelons de longueurs contradictoires. Par conséquent, la préparation des échelons de textes intégrés dans les vidéos est un refuge obligatoire avant la reconnaissance. La préparation consiste à percevoir des régions de texte de l'arrière-plan et à les border précisément dans l'image vidéo à l'aide de rectangles englobant par exemple. La tâche est simple avec un fond uniforme où un simple seuillage peut fixer le problème. Cependant, dans le cas de contenus

vidéo avec un arrière-plan fréquenté et un environnement complexe (couleurs variables, contraste, luminosité, etc.), la détection de texte devient une nécessiter difficile. Ceci est lié, dans un premier temps, à la complication de la discrimination entre régions textuelles et non textuelles et à la vigilance d'atteindre un positionnement précise. En particulier, pour le texte arabe, de multiples problèmes de texte supplémentaires sont rencontrés. Le texte arabe a des caractéristiques de texture divergentes par rapport aux textes latins ou chinois : plus de traits pour d'autres directions, différentes polices et proportions de caractères, plus de signes diacritiques au-dessus et en dessous des caractères, etc.

7. Classification des régions de texte

La première étape d'une procédure de détection de texte consiste à créer un classificateur puissant qui peut faire la distinction entre les motifs textuels et non textuels. Ce n'est pas une tâche anodine compte tenu de son niveau élevé niveau d'asymétrie. Elle consiste à trouver un modèle discriminant entre les motifs « texte » et tous les autres motifs pouvant exister dans une image ou une trame vidéo qui sont sans contrainte, correspondant aux modèles du reste du monde. [38][39][40][41]

ConvNets et Boosting Les approches se sont avérées être parmi les outils les plus puissants utilisés pour la classification des modèles dont visages, objets et textes.

8. Détection basée sur le réseau de neurones à convolution

Les réseaux de neurones à convolution (ConvNets) sont une classe spéciale de réseaux de neurones artificiels feedforward qui permettent d'apprendre l'extraction et la classification des caractéristiques visuelles. Directement à partir des données en utilisant une architecture neuronale holistique.

9. Un aperçu des réseaux de neurones à convolution

De manière générale, l'un des principaux avantages des réseaux de neurones, à savoir les Perceptrons Multicouches (MLPs) [42], est leur capacité à apprendre des modalités de données complexes. Grâce à ses couches cachées, il peut projeter les données d'entrée dans un espace où elles deviennent linéairement séparables. Compte tenu d'un ensemble d'exemples de formation et d'un algorithme d'apprentissage supervisé, il peut donc approximer des fonctions non linéaires. Généralement, les algorithmes utilisés sont basés sur la descente de gradient, comme l'algorithme de rétro propagation. [43]

Théoriquement, pour la classification des modèles dans le domaine de la vision par ordinateur, les MLP peuvent apprendre extraction et classification de caractéristiques si elles sont appliquées directement sur des pixels bruts d'images de données.

Cependant, le problème est que pour ce faire, ils ont besoin d'une énorme quantité de données pour converger. Pour une petite image d'entrée de 24 par 24 pour des exemples, la mission du réseau est de classer les images de données de dimension 576 ce qui implique une architecture neuronale avec un grand nombre de paramètres.

L'apprentissage de ces dernières nécessite donc un grand nombre d'images d'apprentissage sinon cela sur-adapte.

De plus, pour les images de données complexes, ce nombre requis devient beaucoup plus grand afin pour explorer et apprendre différentes modalités de données. C'est pourquoi, les MLP sont généralement appliqués sur un ensemble de caractéristiques empiriques ou artisanales préalablement extraites des données. Ce réduit la dimensionnalité d'entrée évitant ainsi le sur-ajustement et permet au MLP de se concentrer uniquement sur la tâche de classification. Cependant, dans ce cas, leur efficacité dépend encore toujours sur la mesure dans laquelle les traits choisis sont discriminants.

Réseaux de neurones à convolution [44] [45], sont des modèles neuronaux bio-inspirés proposé de résoudre ce problème en utilisant une architecture spéciale qui prend en compte informations spatiales et combine trois concepts principaux :

- des champs récepteurs locaux qui permettent de détecter des caractéristiques locales.

- le partage des poids qui permet de répliquer la recherche de ces caractéristiques et ainsi, implique une réduction des paramètres du réseau et évite le sur-ajustement.

- sous-échantillonnage spatial qui réduit la sensibilité à quelques petites translations, variations de rotation et d'échelle et à de faibles distorsions. [46]

Les premières formes de ConvNets ont été proposées par Fukushima qui est inspiré des travaux de Hubel et Wiesel sur le cortex visuel du chat. Le modèle proposé, appelé Neocognitron, est une séquence de couches de neurones, dans laquelle les neurones sont organisés sous forme de «feature maps ». Chaque neurone d'une carte de caractéristiques donnée est connecté à un seul voisinage correspondant à un nombre fixe de neurones voisins dans une ou plusieurs cartes d'entités de la couche précédente. Ce quartier définit le

local le champ récepteur et ses poids de connexion associés sont appris de manière à extraire formes élémentaires pertinentes comme les coins et les arêtes. Différent des cartes d'entités peuvent être définies pour une image d'entrée donnée, chacune ayant sa propre connexion poids. Cependant, les pondérations sont partagées par tous les champs récepteurs locaux dans une entité donnée carte appliquée à différents endroits des cartes d'entrée correspondantes dans le précédent couche. Par conséquent, chaque carte d'entités dans une couche donnée est le résultat d'une convolution de ses carte(s) de caractéristiques d'entrée par un masque défini par les poids de connexion correspondants. Ce mécanisme définit les couches de convolution. [46]

Les cartes de ces couches sont ensuite sous-échantillonnées afin de réduire la sensibilité des réseaux à de petites variations d'échelles, de formes ou de centrage dans l'image d'entrée. De même, les poids de connexion entre les deux couches sont automatiquement appris. La succession de couches alternées de convolution et de sous-échantillonnage définit l'architecture du Néocognitron. [46] Pour une tâche de classification, la couche de sortie correspond à une classification couche des caractéristiques hiérarchiques extraites. Il est défini par un ensemble de neurones, chacun correspondant à une classe. Une bonne classification d'une image d'entrée correspond à l'activation du neurone de la classe correspondante.

La particularité d'un tel modèle de classification est que tous les paramètres (poids) relatifs à l'extraction, la combinaison et la classification des caractéristiques sont appris. D'où le la sophistication du réseau dépend fortement de la procédure de formation utilisée. Première les algorithmes d'entraînement étaient par couche. Les couches sont formées une par une de manière supervisée. Les cartes d'entités sont formées pour détecter les entités fixées a priori. Plus tard, un réseau de convolution a été proposé par (Lecun et al) Sur la base des travaux de (Fu Kushima) [46]. La particularité la plus importante qui fait que le réseau atteint une apogée de succès et d'efficacité est liée à la procédure de formation. Au lieu d'une formation par couche et au lieu de fixer certaines fonctionnalités a priori à apprendre à chaque carte de chaque couche, les auteurs ont proposé un algorithme d'apprentissage de bout en bout. Le réseau reçoit des images d'entraînement et leurs classes en entrée et apprend à minimiser une erreur globale fonction utilisant l'algorithme de rétro propagation des gradients. Ce processus de formation permet le réseau pour déduire automatiquement les caractéristiques pertinentes et apprendre les paramètres de leur combinaisons et

classification qui correspondent à la classe cible. Par ailleurs, le réseau proposé a une architecture plus légère que le Neocognitron initial. [46]

De nombreuses architectures dérivées de ce ConvNet ou CNN proposé ont été proposées pour s'adapter à différents problèmes comme la lecture de documents, l'OCR et la reconnaissance de l'écriture manuscrite reconnaissance faciale, détection de visages, textes, piétons et corps humains dans des images naturelles reconnaissance vocale avec Time-Delay Neural Networks et reconnaissance des panneaux de signalisation. Récemment, une revue sur les ConvNets, le deep learning architectures en général, leurs différentes applications et les travaux connexes a été donnée par Lecun, Bengio et Hinton dans [46]

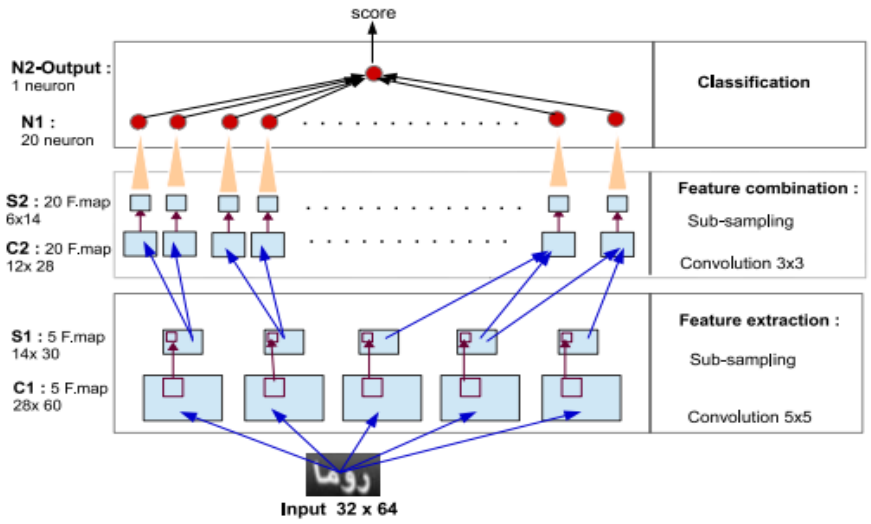


Figure 18 : Convolution Neural Network architecture.

10. Les différents types de format d'image

10.1 Définition de l'image numérique

Le terme d'image numérique désigne, dans son sens le plus général, toute image qui a été acquise, traitée et sauvegardée sous une forme codée représentable par des nombres (valeurs numériques).

La numérisation est le processus qui permet de passer de l'état d'image physique (image optique par exemple) qui est caractérisée par l'aspect continu du signal qu'elle représente (une infinité de valeur dans l'intensité lumineuse par exemple), à l'état d'image numérique qui est caractérisée par l'aspect discret

C'est cette forme numérique qui permet une exploitation ultérieure par des outils logiciels sur ordinateur. [49]

- **Image couleur RVB :** L'œil humain analyse la couleur à l'aide de trois types de cellules photo 'les cônes'. Ces cellules sont sensibles aux basses, moyennes, ou hautes fréquences (rouge, vert, bleu). Pour représenter la couleur d'un pixel, il faut donc donner trois nombres, qui correspondent au dosage de trois couleurs de base : Rouge, Vert, Bleu. On peut ainsi représenter une image couleur par trois matrices chacune correspondant à une couleur de base. [48]

- **Image d'intensités :** C'est une matrice dans laquelle chaque élément est un réel compris entre 0 (noir) et 1 (blanc). On parle aussi d'image en niveaux de gris, car les valeurs comprises entre 0 et 1 représentent les différents niveaux de gris. [48]

- **Image binaire :** Une image binaire est une matrice rectangulaire dans l'élément valent 0 ou 1. Lorsque l'on visualise une telle image, les 0 sont représentés par du noir et les 1 par du blanc. [47]

11. Caractéristiques de l'image

11.1 LES PIXELS

Une image numérique est constituée d'un ensemble de points appelés pixels (abréviation de PICture Element) pour former une image.
Le pixel représente ainsi le plus petit élément constitutif d'une image numérique [50].

L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image:

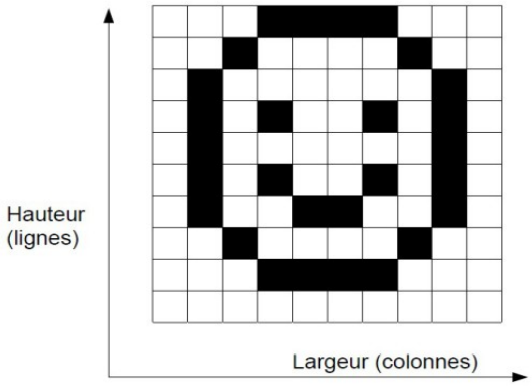


Figure 19 : ensemble de pixels qui représente une image

11.2 LA RÉSOLUTION D'UNE IMAGE

La dimension est la taille de l'image. Elle se présente sous forme d'une matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image. Par contre, la résolution est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateur, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels horizontaux et verticaux sur un moniteur. Plus ce nombre est grand, plus la résolution est meilleure. [48]

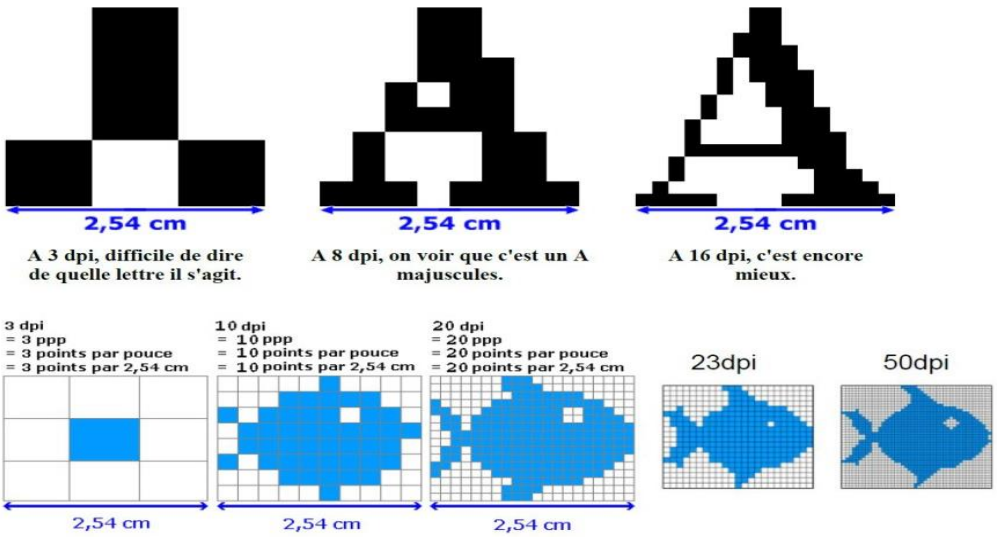


Figure 20 ; représente la résolution d'une image

11.3 Image au Niveau de gris

C'est la valeur d'intensité lumineuse d'un pixel. Cette valeur peut aller du noir (0) jusqu'au blanc (255) en passant par les nuances qui sont contenues dans l'intervalle [0, 255]. Elle correspond en fait à la quantité de la lumière réfléchie. Pour 8 bits, on dispose de 256 niveaux de gris dont 40 sont reconnus à l'œil nue. Plus le nombre de bit est grand plus les niveaux sont nombreux et plus la représentation est fidèle [51].

11.4 Contraste

C'est l'opposition marquée entre deux régions d'une image. Une image contrastée présente une bonne dynamique de la distribution des valeurs de gris sur tout l'intervalle des valeurs possibles, avec des blancs bien clairs et des noirs profonds. Au contraire une image peu contrastée a une faible dynamique, la plupart des pixels ayant des valeurs de gris très proches. Si L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image, le contraste est défini par le rapport : C=L1-L2/L1+L2 [48].

11.5 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes)
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites. [48]

11.6 Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur. C'est un parasite qui représente certains défauts (poussière, petits nuages, baisse momentanée de l'intensité électrique sur les

capteurs, ...etc.). Il se traduit par des taches de faible dimension et dont la distribution sur l'image est aléatoire. [52]

11.7 Contour

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentant une différence significative. Dans une image numérique, les contours se situent entre les pixels appartenant à des régions ayant des intensités moyennes différentes; il s'agit de contours de type « saut d'amplitude ». Un contour peut également correspondre à une variation locale d'intensité présentant un maximum ou un minimum; il s'agit alors de contour « en toit » [48]

12. Conclusion

Dans ce chapitre nous avons traité la problématique de la reconnaissance des caractères arabes et les méthodes utilisées à cet effet. Le principe de reconnaissance est basé sur l'obtention d'une base de données contenant des images des mots, on a discuté aussi sur les domaines d'application d'un system de reconnaissance d'écriture arabe manuscrits, ainsi que les difficultés inhérentes à la reconnaissance de l'écriture arabe. On a parlé aussi sur les caractéristiques d'image et les différents composants de l'image.

Chapitre 3

1. Introduction

Les systèmes de reconnaissance d'écriture arabes manuscrits font face à plusieurs défis, y compris la variation illimitée de l'écriture humaine et des grandes bases des données publique.

L'objectif de ce chapitre est de présenter les étapes de l'implémentation de l'approche proposée et les différentes étapes de réalisation, Nous nous sommes intéressés à l'utilisation des CNN

D'abord, Nous commençons par présenter la conception globale, la base de données qui a servi à l'entrainement et l'évaluation de notre modèle et les étapes de la réalisation de ce dernier.

2. Reconnaissance de l'écriture Arabe par le deep learning

2.1 Explication générale

Compte tenu de l'avancement de l'apprentissage à profond, le CNN (réseau de neurones convolutifs) a été longuement examiné. Il présente les avantages d'être insensible à la déformation. Le coût de calcul impliqué dans l'extraction d'informations immédiatement à partir d'une image est minime. Le CNN a des paramètres tellement développés que le la fonction de détection et de reconnaissance du texte est massivement améliorée dans la scène naturelle. Courant les techniques fondées sur l'utilisation du CNN peuvent être généralement classées en un certain nombre de sous-catégories comprenant les techniques basées sur la segmentation, les techniques basées sur les propositions régionales et des techniques hybrides qui utilisent l'apprentissage multitâches. Les sections suivantes comprennent une discussion sur détection et reconnaissance de texte par apprentissage profond (cnn).

2.2 Conception globale

2.2.1 Architecture

La tâche essentielle de système de reconnaissance d'écriture hors-ligne est basée sur les RNC Pour se faire, il nécessite un certain nombre d'étapes à mettre en œuvre. L'ensemble de ces phases forment généralement la structure de système de reconnaissance d'écriture, qui peut se résumer par le schéma (figure 21).

2.2.2 Quelques étapes spécifiques à notre programme

Voici les étapes d'un system de reconnaissance d'un mot manuscrit :

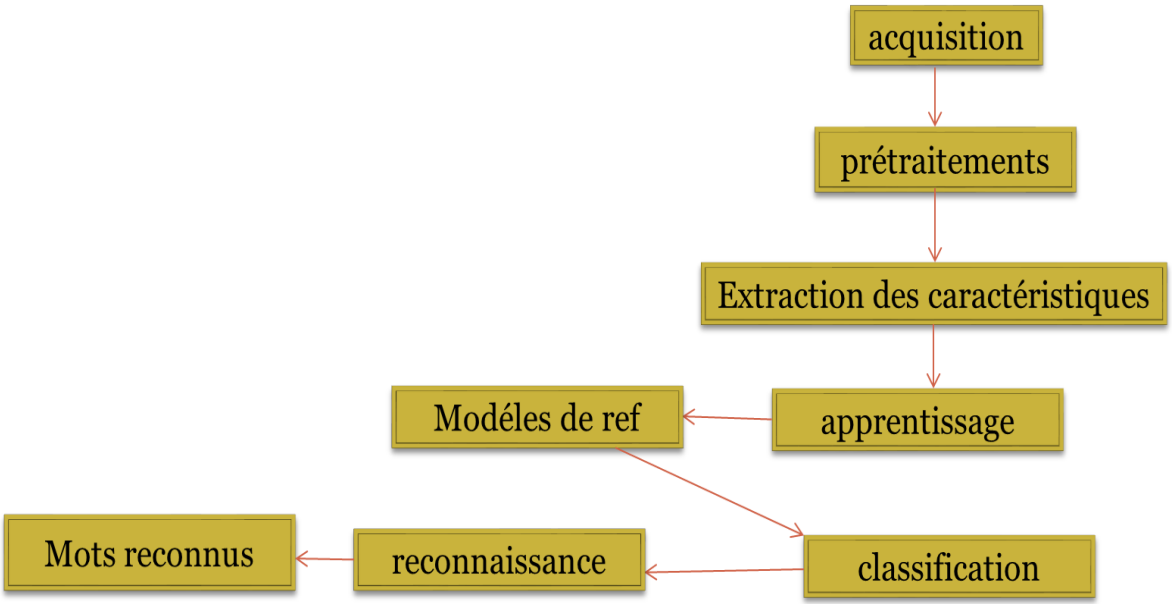


Figure 21 : Diagramme de référence du système de reconnaissance.

2.2.3 Acquisition

L'acquisition permettant la conversion du document papier sous la forme d'une image numérique (bitmap). Cette étape est importante car elle se préoccupe de la préparation des documents à saisir, du choix et du paramétrage de scanner, ainsi que du format de stockage des images [56].

2.2.4 Prétraitement

Le prétraitement est l'étape cruciale dans tout système de reconnaissance [49]. Le but de cette étape dans la reconnaissance de texte manuscrit est d'améliorer la lisibilité de l'image de texte et de supprimer les détails qui n'ont pas de pouvoir discriminatif dans le processus de reconnaissance [55].

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

Le prétraitement de l'image d'entrée est effectué en convertissant l'image donnée en image en niveaux de gris.

Habituellement, une image colorée normale se compose de trois canaux - canal rouge, canal vert, canal bleu communément appelé RVB.

Ensuite, l'image colorée est convertie en image en niveaux de gris qui consiste en un seul canal monochrome afin d'éviter bruit indésirable dans l'image. L'image d'entrée donnée serait de taille variée, ce qui pourrait entraîner une perte de prédiction précise lorsque l'image est comparée à celle du réseau neuronal convolutif formé.

2.2.4.1 Suppression de bruit

Le bruit est introduit lors de l'acquisition d'images d'entrée via des scanners optiques ou des dispositifs d'écriture et provoque une distorsion dans l'image d'entrée. Avant la reconnaissance du texte, il est essentiel de supprimer ces variantes. Bien qu'il existe des nombreuses techniques utilisées pour éliminer le bruit, deux approches principales sont largement utilisées [50]

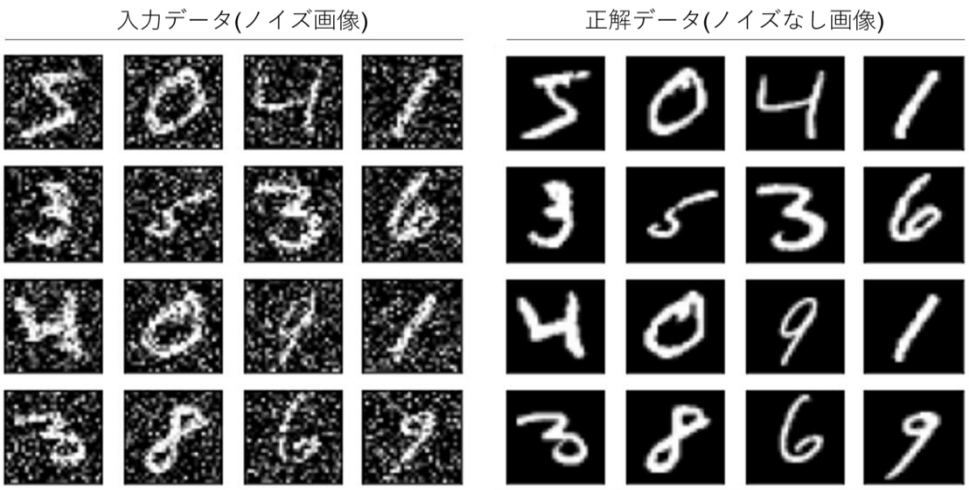


Figure 22 : représente la suppression de bruit

2.2.4.2 Normalisation

La méthode de normalisation est généralement utilisée dans la reconnaissance des mots pour réduire tous les types de variations, et pour obtenir des données normalisées. Cependant, il engendre également à une distorsion de la forme excessive et élimine quelques informations utiles. Les méthodes usuelles pour normaliser un caractère sont les suivantes :

- Normalisation de la taille ;
- Correction de l'inclinaison des lignes (Skew correction) ;
- Correction de l'inclinaison des caractères (Slant correction) ;
- Estimation de la ligne de base [57].

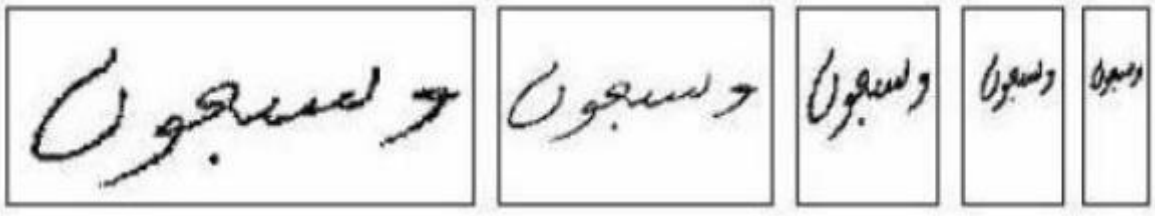


Figure 23 : Exemples de mots manuscrits avec des tailles différentes [57].

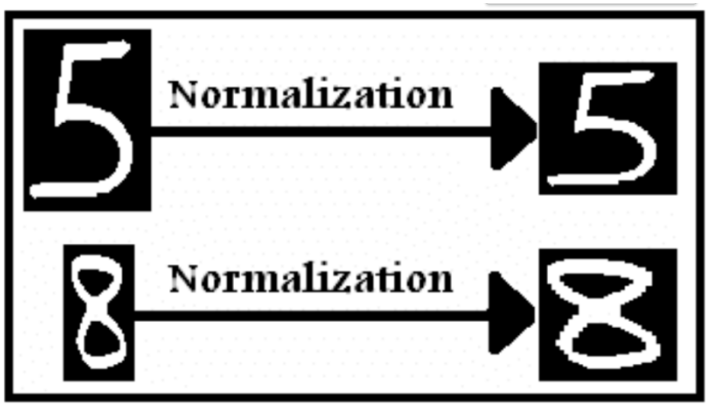
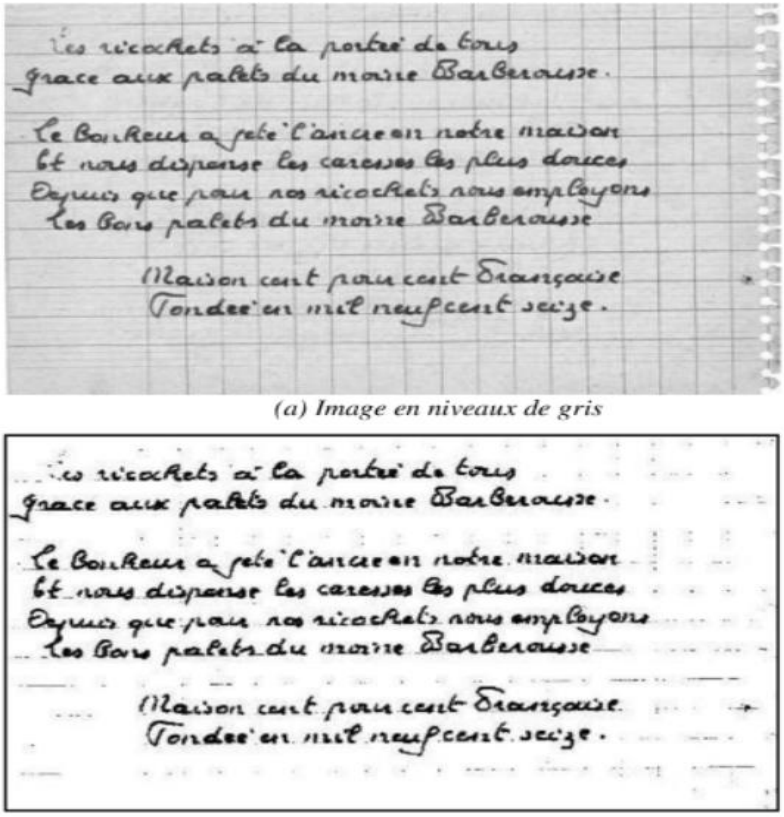


Figure 24 : Normalisation de certains chiffres manuscrits.

2.2.4.3 Seuillage

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



(a) Image en niveaux de gris

Figure 25 ; Exemple de binarisation d'une image en niveaux de gris

2.2.4.4 FILTRAGE

Il existe plusieurs filtres de domaine spatial et de fréquence qui peuvent être conçus pour supprimer le bruit et réduire le nombre des points parasites [73]. Les filtres peuvent être conçus pour le seuillage, l'affûtage, le lissage et pour le réglage du contraste. La mise en œuvre d'un filtre spatial linéaire implique la conception d'un masque et l'exécution d'une convolution de l'image et du masque

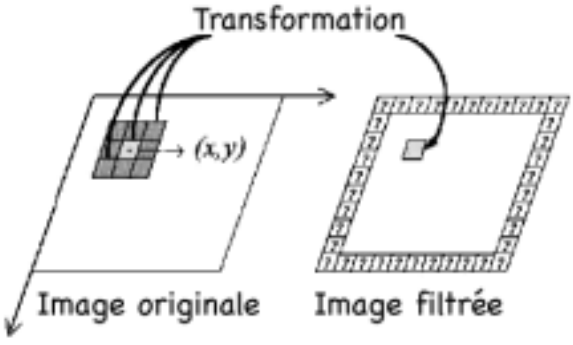


Figure 26 : l'application de filtre sur limage

2.2.4.5 Squelettisation :

La squelettisation sert à obtenir une épaisseur égale à 1 du trait d'écriture et de se ramener ainsi à une écriture linéaire. Le squelette doit préserver la forme, connexité, topologie et extrémités du tracé, et ne doit pas introduire d'éléments parasites [58].

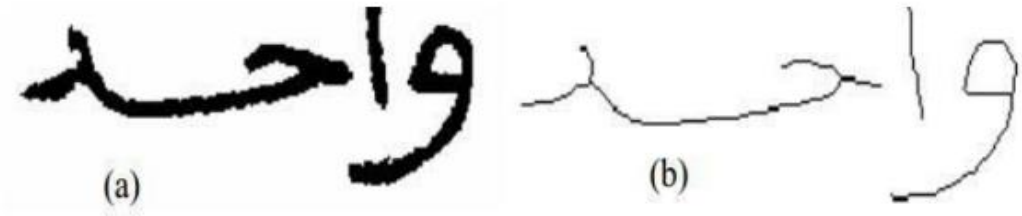


Figure 27 : Exemples de squelettisation [59].

2.2.5 Extraction des caractéristiques

Dans les textes imprimés et manuscrits, les caractéristiques saisissent les informations extraites de l'image textuelle. Ces informations doivent avoir les caractéristiques essentielles du caractère ou du mot qui le différencient; En d'autres termes, filtrer tous les attributs et préserver les propriétés qui rendent un caractère ou un mot différent d'un autre [53]. Ces informations sont transmises au classificateur pour faciliter le processus de classification. Les techniques d'extraction des caractéristiques diffèrent d'une application à l'autre. Les techniques qui réussissent dans une application, peuvent ne pas être couronnées de succès pour d'autres applications [54]. Par conséquent, la sélection de la méthode d'extraction des caractéristiques reste l'étape la plus importante pour obtenir une précision de reconnaissance élevée.

2.2.5.1 Type de caractéristiques

Les différentes techniques d'extraction des caractéristiques sont classées en fonction des types des primitives [60]:

2.2.5.2 Caractéristiques structurelles

Les primitives structurelles sont généralement extraites non pas de l'image brute, mais à partir d'une représentation de la forme par le squelette ou par le contour. Ainsi, on ne parle plus de trous, mais de boucles. Cependant, pour le reste, les primitives structurelles correspondent à peu près aux primitives topologiques, il s'agit

principalement :

¬ Des segments de droite,

¬ Des arcs, boucles et concavités, des pentes,

¬ Des angularités, points extremum et points terminaux, jonctions et croisements [61].

2.2.5.3 Caractéristiques statistiques

On cherche ici à représenter le mot par des mesures statistiques de l'image. On peut par exemple utiliser la distribution des pixels dans différentes régions de l'image, ou bien des histogrammes (nombre de points noirs par colonne, par ligne, ou dans d'autres directions) [61].

2.2.5.4 Caractéristiques globales

On parle de caractéristiques globales lorsque le codage ne fait pas intervenir la position spécifique d'éléments particuliers de l'image. L'image est considérée globalement sans chercher à distinguer les différentes zones [61].

2.2.5.5 Caractéristiques morphologiques

L'extraction des caractéristiques morphologiques s'appuie sur une étude des positions relatives des différentes composantes noires et blanches de l'image. On décrit alors le mot en termes de composantes blanches et noires, de cavités (parties blanches partiellement entourées de noir) et de boucles (parties blanches entièrement entourées de noir). La détection des caractéristiques morphologiques peut être effectuée par des opérateurs morphologiques de dilatation selon les quatre directions, et d'intersection d'images [61].

2.2.5.6 Caractéristiques métriques

Cette catégorie comprend des caractéristiques basées sur des mesures physiques de l'image. Outre des caractéristiques assez simples, comme la hauteur, la largeur et le rapport de ces deux grandeurs, on peut utiliser des caractéristiques plus complexes comme le codage des profils. On peut définir les profils par rapport aux quatre axes naturels (gauche, droit, haut et bas), mais aussi par rapport à d'autres directions (des profils à 0°, 45°, 90° et 135°) [61].

2.2.5.7 Caractéristiques adaptatives

Les caractéristiques adaptatives sont obtenues directement de l'image et requièrent une phase d'apprentissage. Autrement dit, le système opère sur une représentation proche de l'image d'origine et doit lui-même construire et optimiser l'extracteur de caractéristiques [61].

3. Architecture d'un CNN

Les réseaux de neurones convolutionnels (en anglais *Convolutional neural networks*), aussi connus sous le nom de CNNs, sont un type spécifique de réseaux de neurones qui sont généralement composés des couches suivantes :

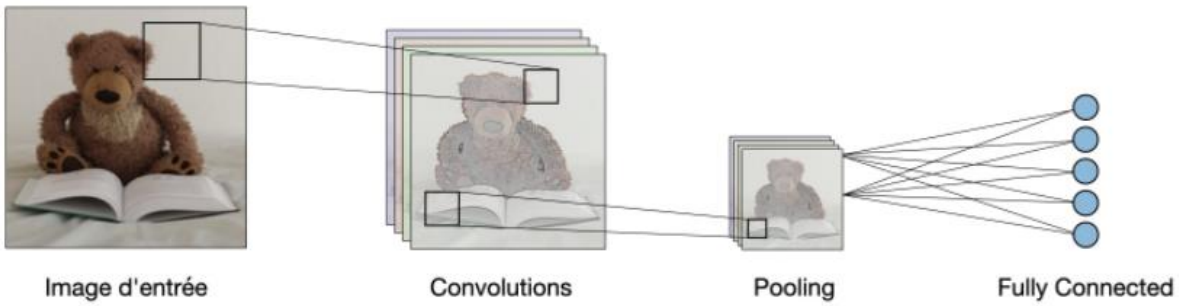


Figure 28 : architecture d'un CNN [15]

La couche convolutionnelle et la couche de pooling peuvent être ajustées [15].

3.1 Types de couche

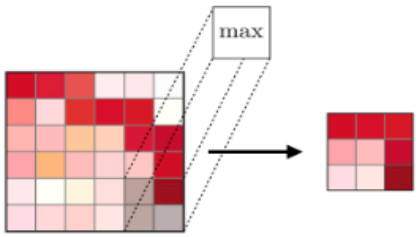
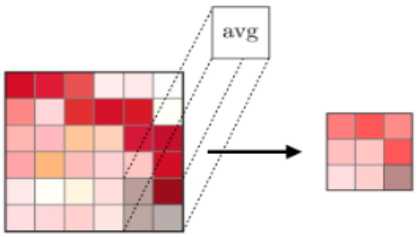
3.1.1 Couche convolutionnelle (CONV)

La couche convolutionnelle (en anglais *convolution layer*) (CONV) utilise des filtres qui scannent l'entrée I*I* suivant ses dimensions en effectuant des opérations de convolution. Elle peut être réglée en ajustant la taille du filtre F*F* et le stride S*S*. La sortie O*O* de cette opération est appelée *feature map* ou aussi *activation map* [15].

3.1.2 Pooling (POOL)

La couche de pooling (en anglais *pooling layer*) (POOL) est une opération de sous-échantillonnage typiquement appliquée après une couche convolutionnelle. En particulier, les types de pooling les plus populaires sont le max et l'average pooling, où les valeurs maximales et moyennes sont prises, respectivement [15].

Tableau 5 : tableau qui montre la differance entre le max pooling et l'average pooling [15].

Type	Max pooling	Average pooling
But	Chaque opération de pooling sélectionne la valeur maximale de la surface	Chaque opération de pooling sélectionne la valeur moyenne de la surface
Illustration		
Commentaires	• Garde les caractéristiques détectées • Plus communément utilisé	• Sous-échantillonne la *feature map* • Utilisé dans LeNet

3.1.3 Fully Connected (FC)

La couche de fully connected (en anglais *fully connected layer*) (FC) s'applique sur une entrée préalablement aplatie où chaque entrée est connectée à tous les neurones. Les couches de fully connected sont typiquement présentes à la fin des architectures de CNN et peuvent être utilisées pour optimiser des objectifs tels que les scores de classe [15].

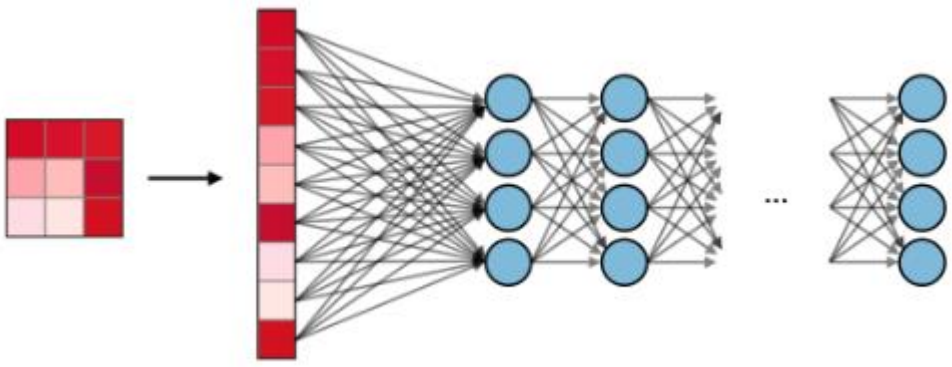


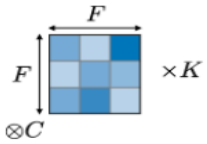
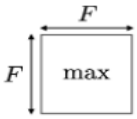
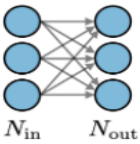
Figure 29 : l'application du couche fully connected [15].

4. Comprendre la complexité du modèle

Pour évaluer la complexité d'un modèle, il est souvent utile de déterminer le nombre de paramètres que l'architecture va avoir. Dans une couche donnée d'un réseau de

neurones convolutionnels, on a :

Tableau 6 : tableau qui montre la difference entre le pooling et la couche de convolution et la couche fully connected [15].

	CONV	POOL	FC
Illustration			
Taille d'entrée	$I \times I \times C$	$I \times I \times C$	N_{in}
Taille de sortie	$O \times O \times K$	$O \times O \times C$	N_{out}
Nombre de paramètres	$(F \times F \times C + 1) \cdot K$	0	$(N_{\text{in}} + 1) \times N_{\text{out}}$
Remarques	• Un paramètre de biais par filtre • Dans la plupart des cas, $S < F$ • $2C$ est un choix commun pour K	• L'opération de pooling est effectuée pour chaque canal • Dans la plupart des cas, $S = F$	• L'entrée est aplatie • Un paramètre de biais par neurone • Le choix du nombre de neurones de FC est libre

4.1 Fonctions d'activation communément utilisées

4.1.1 Relu :

La fonction Rectified Linear Unit **(**Relu**)** est la fonction d'activation la plus simple et la plus utilisée**.**

Elle donne x si x est supérieur à 0**,** 0 sinon**.** Autrement dit, c'est le maximum entre x et 0 **:**

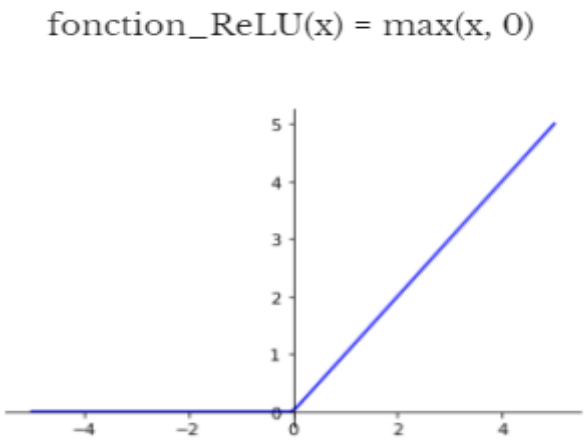


Figure 30 : fonction RELU

Cette fonction permet d'effectuer un filtre sur nos données. Elle laisse passer les valeurs positives (x > 0) dans les couches suivantes du réseau de neurones. Elle est utilisée presque partout mais surtout pas dans la couche finale, elle est utilisée dans les couches intermédiaires [71].

```
tf.keras.activations.relu(x, alpha=0.0, max_value=None, threshold=0)
```

- x : donnée d'entrée, tenseur
- alpha : Un nombre réel qui régit la pente pour les valeurs inférieures au seuil.
- max_value : Un nombre réel qui définit le seuil de saturation (la plus grande valeur que la fonction retournera).
- threshold : Un nombre réel qui donne la valeur seuil de la fonction d'activation en dessous de laquelle les valeurs seront amorties ou mises à zéro [71].

4.1.2 Softmax :

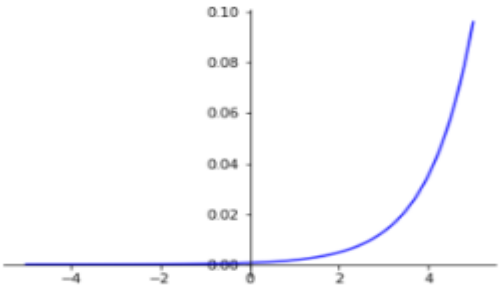
La fonction softmax permet-elle de transformer un vecteur réel en vecteur de probabilité.

On l'utilise souvent dans la couche finale d'un modèle de classification, notamment pour les problèmes multi classe.

Dans la fonction Softmax, chaque vecteur est traité indépendamment. L'argument axis définit l'axe d'entrée sur lequel la fonction est appliquée [71].

$$fonction\_Softmax(x) = exp(x) / tf.reduce\_sum(exp(x))$$

$$fonction\_Softmax(x) = exp(x) / sum(exp(x_i))$$



Fonction Softmax

Figure 31 : fonction SOFTMAX [71].

```
tf.keras.activations.softmax(x, axis=-1)
```

- axis : Nombre entier, axe le long duquel la normalisation softmax est appliquée [71].

5. CLASSIFICATION

Le réseau de neurones convolutifs est utilisé comme classificateur pour classer l'écriture manuscrite de l'image d'entrée. Un CNN se compose d'une couche d'entrée et d'une couche de sortie, ainsi que de plusieurs couches cachées. Les couches cachées d'un CNN se composent généralement de couches convolutives, de couches de regroupement, de couches entièrement connectées et de couches de normalisation. Un CNN se compose de trois les principaux composants qui sont la couche convolutive, la couche de regroupement et la couche de sortie.

La fonction d'activation couramment utilisée avec CNN est ReLU qui signifie Rectified Linear Unit.

La couche de convolution calculera la sortie des neurones qui sont connectés aux régions locales dans l'entrée, chacun calculant un point produit entre leurs poids et une petite région à laquelle ils sont connectés dans le volume d'entrée. La couche de regroupement est une forme de sous-échantillonnage non linéaire. La mise en commun maximale est la plus courante qui partitionne l'image d'entrée en un ensemble de rectangles qui ne se chevauchent pas et, pour chacune de ces sous-régions, produit le maximum. Relu applique la

fonction d'activation non saturante. Il augmente les propriétés non linéaires de la fonction de décision et du réseau global sans affecter les champs récepteurs de la convolution couche.

Une unité linéaire redressée a une sortie 0 si l'entrée est inférieure à 0, et une sortie brute sinon. Sa valeur est obtenue en fonction de la formule qui est la suivante :

F(x)= max(x, 0)

La fonction softmax est souvent utilisée dans la couche finale d'un classificateur basé sur un réseau de neurones [72]. La fonction softmax écrase sorties de chaque unité entre 0 et 1, tout comme une fonction sigmoïde. Mais il divise également chaque sortie de sorte que la somme totale de la sortie est égale à 1. La sortie de la fonction softmax est équivalente à une distribution de probabilité catégorielle. Ainsi, softmax La fonction calcule la distribution des probabilités de l'événement sur "n" événements différents.

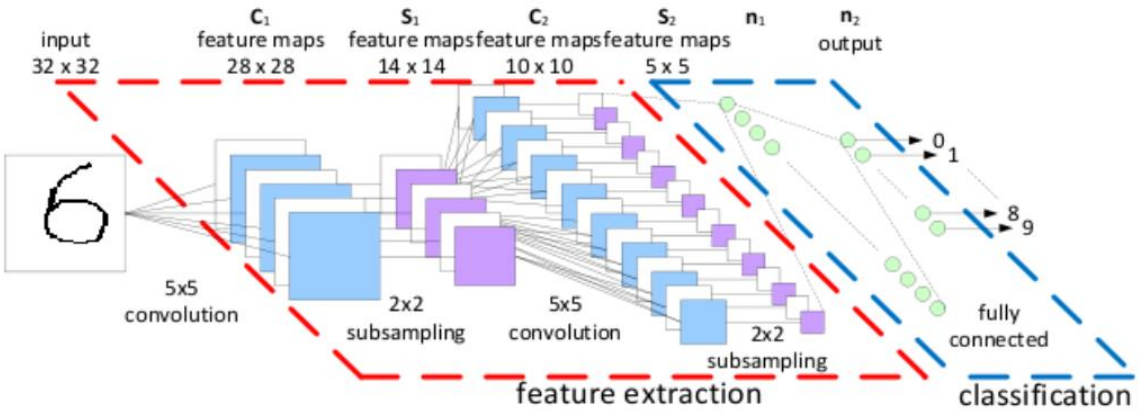


Figure 32 : CNN Feature Extraction pour l'image du chiffre manuscrit

6. Description de la base de données

La base de donnes IFN\ENIT c'est une base de donnes qui contient des noms manuscrits de villes/villages arabes. Pour chaque nom les informations de base, par exemple l'ordre des formes de caractère, les informations sur le style de l'écriture et la ligne de base, sont codées. 411 auteurs ont rempli des formulaires avec plus de 26400 noms contenant plus de 210 000 caractères. La base de données est décrite en détail, et elle est conçue pour la formation et l'essai des systèmes d'identification pour les mots arabes manuscrits [63].

7. Métrique d'évaluation de performance

Certaines mesures de performance nous aident à améliorer nos modèles qui sont :

Précision : Il s'agit de la mesure des cas positifs correctement identifiés parmi tous les cas positifs prédits. Ainsi, il est utile lorsque le coût des faux positifs est élevé [55].

Équation 1 : equation de precision

$$\text{Precision} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Positive})}$$

Recall : C'est la mesure des cas positifs correctement identifiés parmi tous les cas positifs réels. C'est important lorsque le coût des faux négatifs est élevé [55].

Équation 2 : equation de recall

$$\text{Recall} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Negative})}$$

Accuracy : L'une des métriques les plus évidentes, c'est la mesure de tous les cas correctement identifiés. Il est surtout utilisé lorsque toutes les classes sont d'égale importance. [55]

Équation 3 : equation de l'accuracy

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{(\text{True Positive} + \text{False Positive} + \text{True Negative} + \text{False Negative})}$$

F1-score : C'est la moyenne pondérée de l'accuracy et du rappel. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. F1 est généralement plus utile que l'accuracy, surtout si vous avez une distribution de classe inégale. L'accuracy fonctionne mieux si les faux positifs et les faux négatifs ont un coût similaire. [55]

Équation 4 : equation de F1-score

$$\text{F1-score} = \left(\frac{\text{Recall}^{-1} + \text{Precision}^{-1}}{2}\right)^{-1} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

8. CONCLUSION

Dans ce chapitre, on a essayé d'expliquer l'architecture de notre programme ainsi que les étapes nécessaires pour réaliser ce projet, on a discuté aussi d'architecture CNN et de types de couches ainsi que Histogramme des gradients orientés et les fonctions d'activation communément utilisées, nous avons aussi décrit la base de donnes utiliser Le but de ce chapitre est d'expliquer le fonctionnement du programme.

CHAPITRE 4

1. Introduction

Les systèmes de reconnaissance de mots arabes manuscrits sont confrontés à plusieurs défis, notamment des variations illimitées de l'écriture humaine et de grandes bases de données publiques. Ce chapitre utilise un réseau de neurones convolutifs qui peut être appliqué efficacement à la reconnaissance manuscrite de l'arabe. Ainsi que ce chapitre décrit l'utilisation des CNN dans les systèmes de reconnaissance des mots manuscrits.

2. Conception de l'architecture CNN

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

3. Logiciels et librairies Utilisés dans l'implémentation

Il existe un très grand nombre de langages de programmation, chacun avec ses avantages et ses inconvénients.

Dans notre cas la reconnaissance des mots arabes manuscrits, le langage de programmation approprié est le python.

Python : Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion

d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages [64].

Visual Studio Code : est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS [67].

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Le code source de Visual Studio Code provient du projet logiciel libre et open source VSCode de Microsoft publié sous la licence MIT permissive, mais les binaires compilés constituent un freeware, c'est-à-dire un logiciel gratuit pour toute utilisation mais privateur [68].

tensorflow : est un outil open source d'apprentissage automatique développé par Google. Le code source a été ouvert le 9 novembre 2015 par Google et publié sous licence Apache.

Il est fondé sur l'infrastructure DistBelief, initiée par Google en 2011, et doté d'une interface pour Python.

TensorFlow est l'un des outils les plus utilisés en IA dans le domaine de l'apprentissage machine [3].

Kesras : est une API de réseaux neuronaux de haut niveau, écrite en Python et capable de s'exécuter sur TensorFlow, CNTK ou Theano. Il a été développé dans le but de permettre une expérimentation rapide. Être en mesure de passer de l'idée au résultat le plus rapidement possible, est la clé pour faire de la recherche :

- Permet un prototypage facile et rapide (grâce à la convivialité, à la modularité et à l'extensibilité) [3].
- Prend en charge les réseaux Convolutionnels et les réseaux récurrents ainsi que les combinaisons des deux.
- Fonctionne de manière transparente sur le processeur et le processeur graphique [3].

OpenCv : ou Open Source Computer Vision est une bibliothèque contenant plus de 2500 algorithmes de vision par ordinateur. Il est accessible au travers d'API pour plusieurs langages dont Python qui est le principal langage de programmation utilisé dans ce projet. Initialement écrite en C par des chercheurs de la société Intel, OpenCv est aujourd'hui la bibliothèque de référence pour la vision par ordinateur, aussi bien dans le monde de la recherche que celui de l'industrie. En comparaison avec d'autre langage comme C ou C++, Python est plus lent. Mais Python peut être étendu avec C ou C++. Cette fonctionnalité aide à écrire des codes gourmands en termes de calculs en C ou C++ et à créer un wrapper (une fonction qui est destinée à appeler une ou plusieurs autres fonctions) Python pour pouvoir les utiliser en tant que modules [3].

Numpy : est une bibliothèque pour le langage de programmation Python, destiné à manipuler des matrices ou des tableaux multidimensionnels ainsi que

des fonctions mathématiques opérant sur ces tableaux [3].

Pandas : La bibliothèque logicielle open-source Pandas est spécifiquement conçue pour la manipulation et l'analyse de données en langage Python. Elle est à la fois performante, flexible et simple d'utilisation.

Grâce à Pandas, le langage Python permet enfin de charger, d'aligner, de manipuler ou encore de fusionner des données. Les performances sont particulièrement impressionnantes quand le code source back-end est écrit en C ou en Python [65].

Matplotlib : est une bibliothèque Python open source, initialement développée par le neurobiologiste John Hunter en 2002. L'objectif était de visualiser les signaux électriques du cerveau de personnes épileptiques. Pour y parvenir, il souhaitait répliquer les fonctionnalités de création graphique de MATLAB avec Python [66].

4. Configuration Utilisé dans l'implémentation :

La configuration du matériel utilisé dans notre implémentation est :

- ✓ PC portable DELL i5 CPU 2.40 GHZ
- ✓ RAM de taille 8 GO
- ✓ Disque dur de taille 500 GO SSD
- ✓ Système d'exploitation windows 10

5. Le dataset utilisé

Pour faire l'entrainement et le test de notre system de reconnaissance d'écriture arabe manuscrite on a pris une partie de la base de données qui s'appelle IFN\ENIT (7560 IMAGES 1512 images de test parmi ces 7560 images) on à bien défini la base de donnes ifn\enit dans le chapitre 3

6. L'architecture d'un system de reconnaissance d'écriture arabe manuscrite

L'architecture de notre réseau pour la reconnaissance des mots arabes manuscrit est basée sur le réseau de neurones convolutionnel dont la structure est donné ci-dessous Figure 33.

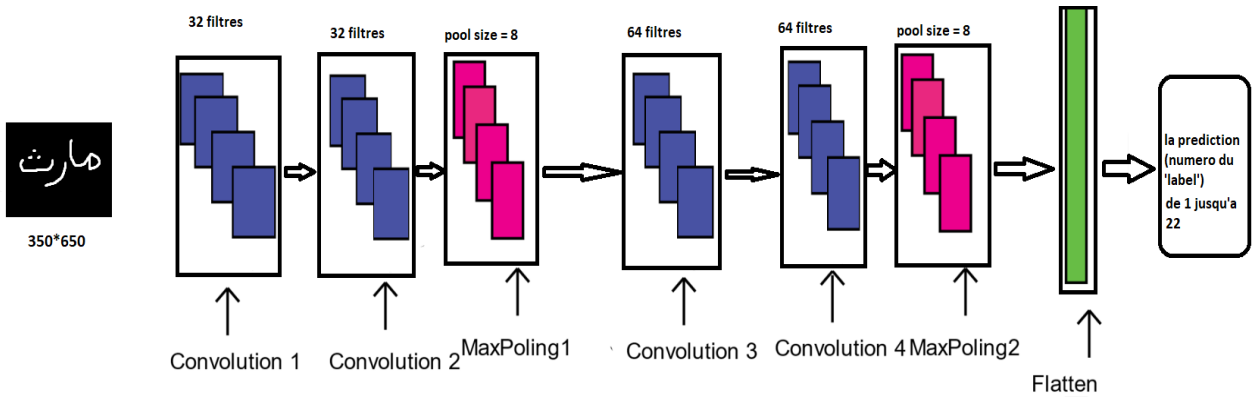


Figure 33: L'architecture utilisée sur notre system

Nous présentons dans la figure 33 l'architecture de notre RNC qui est composé de quatre couches de convolution et deux couches de maxpooling et 6 couches de fully connected.

L'image en entrée est de taille 350*650, l'image passe d'abord à la première couche de convolution. Cette couche est composée de 32 filtres, Chacune de nos couches de convolution est suivie d'une fonction d'activation ReLU cette fonction force les neurones à retourner des valeurs positives, après cette convolution 32 features maps de taille 350*650 seront créés.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 350, 32)	83232
conv1d_1 (Conv1D)	(None, 350, 32)	4128
max_pooling1d (MaxPooling1D)	(None, 86, 32)	0
dropout (Dropout)	(None, 86, 32)	0
conv1d_2 (Conv1D)	(None, 86, 64)	8256
conv1d_3 (Conv1D)	(None, 86, 64)	16448
max_pooling1d_1 (MaxPooling1D)	(None, 20, 64)	0
dropout_1 (Dropout)	(None, 20, 64)	0
flatten (Flatten)	(None, 1280)	0
dense (Dense)	(None, 512)	655872
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 700)	179900
dense_3 (Dense)	(None, 500)	350500
dense_4 (Dense)	(None, 150)	75150
dense_5 (Dense)	(None, 22)	3322

Total params: 1,508,136
Trainable params: 1,508,136
Non-trainable params: 0

Figure 34 : CONFIGURATION DU NOTRE MODELE RNC

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

On répète la même chose avec les couches de convolutions trois, quatre, ces couches sont composées de 64 filtres, la fonction d'activation ReLU est appliquée toujours sur chaque convolution. Une couche de Maxpooling est appliquée après la couche de convolution quatre. Àla sortie de cette couche, nous aurons 64 feature maps de taille 20*20. Le vecteur de caractéristiques issu des convolutions a une dimension de 1280.

Après ces quatre couches de convolution, nous utilisons un réseau de neurones composé de 6 couches fully connected. Les deux premières couches ont chacune 512, 256 neurones où la fonction d'activation utilisée est le ReLU, et la troisième, 4eme et 5eme couche chacune 700,500, 150 neurones respectivement et la dernière couche est un softmax qui permet de

calculer la distribution de probabilité des 22 classes (nombre de classe dans la base d'image).

6.1 Code Source de notre modèle proposé

Strides=4 #le pas de convolution

convolution1=32 # nombre de filtre (32 features maps)

convolution2=32 # nombre de filtre (32 features maps)

convolution3 = 64 # nombre de filtre (64 features maps)

convolution4 = 64 # nombre de filtre (64 features maps)

Pool_size=8 # taille de pool

dropout1 = 0.25 #appliquer un dropout avec probabilité de 25 %

dropout2=0.50 #appliquer un dropout avec probabilité de 50 %

couchecache1= 512 #créer une première couche de 512 neurones

couchecache2= 256 #créer une première couche de 256 neurones

couchecache3= 700 #créer une première couche de 700 neurones

couchecache4= 500 #créer une première couche de 500 neurones

couchecache5= 150 #créer une première couche de 150 neurones

```
df = pd.read_csv('C:\\Users\\pc\\Desktop\\Nouveau dossier\\data_2021\\data.csv')
df.head()
```

Commençant par mettre les labels des images dans un data frame (df)

```
  haut.append(h)
  larg.append(w)

df['haut'] = haut
df['larg'] = larg

df.head()

df.haut.hist()

df.larg.hist()
```

En affichant histogramme qui nous montre la taille majoritaire des images, comme ça on peut fixer les tailles des images

```
fixed_haut = 350
fixed_larg = 650
images = []
```

#ici on a fixé la taille des images (350*650)

```
image = cv2.imread('C:\\Users\\pc\\Desktop\\Nouveau dossier\\data_2021\\bdh\\'+image_path,cv2.IMREAD_GRAYSCALE)
image = cv2.resize(image,(fixed_larg,fixed_haut))
```

#charger la base de donnes et fixer la taille des images

#7650 images d'apprentissage

#1512 images de test

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=tf.keras.metrics.categorical_accuracy )
```

Compiler le model.

```
train_images,test_images , train_labels , test_labels = train_test_split(images,labels,test_size=0.2,stratify = labels,random_state = 123)

history = model.fit(train_images,train_labels,validation_split= 0.1,epochs = 25,batch_size =40)
```

#test_size=0.2 #20% de la base de donnes est réserver pour le test

#On a utilisé (Stratify) pour qu'on soit sure que toutes les classes sont disponibles dans les images de test

#Lancement de l'apprentissage du modèle sur 25 epochs

#nombre de classes = 22 classes

#batch_size = 40 images

#Image d'entrée de taille 350*650

```
model.add(Conv1D(filters=32,padding='same', kernel_size=4, input_shape=(350, 650),activation='relu'))
```

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

```
model.add(Conv1D(filters=32, padding='same',kernel_size=4,activation='relu'))
```

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

```
model.add(MaxPooling1D(pool_size=8,strides=4))
```

3 : Cette ligne de commande permet de réduire la taille de l'image, La méthode Max Pooling est utilisée et la taille de l'image sera divisée sur 8.

```
model.add(Conv1D(filters=64, padding='same',kernel_size=4,activation='relu'))
```

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

```
model.add(Conv1D(filters=64, padding='same',kernel_size=4,activation='relu'))
```

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

```
model.add(MaxPooling1D(pool_size=8,strides=4))
```

6 : Cette ligne de commande permet de réduire la taille de l'image, La méthode Max

Pooling est utilisée et la taille de l'image sera divisée sur 8.

```
model.add(Flatten())
```

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

```
model.add(Dense(units = 512, input_dim = 1280, activation='relu'))
model.add(Dense(256,activation='relu'))
model.add(Dense(700,activation='relu'))
model.add(Dense(500,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(22,activation='softmax'))
```

8 : Cette commande permet de créer une couche cachée avec une taille de 512, 256, 700, 500,150neurones, lafonction RELU est utilisé comme fonction d'activation.
Cette commande permet aussi de créer une couche cachée avec une taille de 1280neurones, la fonction RELU est utilisée comme fonction d'activation.

#Softmax : Cette commande permet de créer une couche de sortie composée de 22 neurones(nombre de classes) la fonction softmax est utilisée pour calculer la probabilité de chaque classe.

```
model.add(Dropout(0.25))
```

```
model.add(Dropout(0.5))
```

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

- Le paramètre type avec pour valeur 'dropout'

- *Le paramètre rate avec pour valeur 0.25 et 0.5.*

7. Résultats expérimentaux

7.1 L'interface de notre system

Voici l'interface utilisé sur notre system de reconnaissance d'écriture arabe manuscrite

Figure 35 : l'interface d'un system de reconnaissance d'écritures manuscrites

7.2 L'étape de prétraitements

7.2.1 Les filtre que nous avons utilisé sur les images

```
### filtre median
median_image = ndimage.median_filter( image , size = 5)
plt.imshow( median_image )
plt.show()
```

1 : le filtre median : Le filtre médian est un filtre numérique non linéaire, souvent utilisé pour la réduction de bruit. La réduction de bruit est une étape de prétraitement

```
### filtre maximum
maximum_image = ndimage.maximum_filter( image , size = 5)
plt.imshow( maximum_image )
plt.show()
```

2 : le filtre maximum : Les filtres minimum et maximum ordonnent l'ensemble des pixels du voisinage, et sélectionnent soit la plus petite ou la plus élevée. Cette famille de filtre permet de supprimer des petits détails très lumineux ou très sombres, mais affecte fortement la taille des objets.

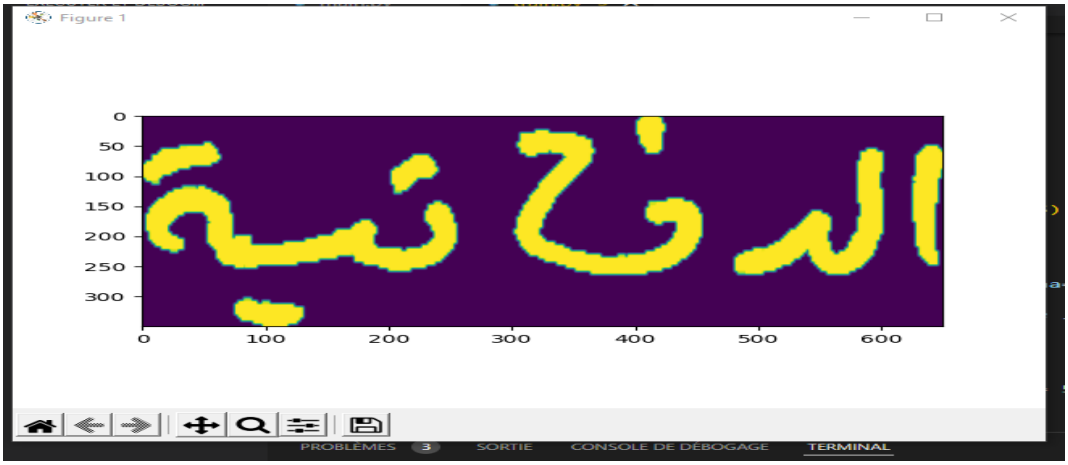


Figure 36 : résultat en appliquant le filtre max

```
# filtre gaussien pour réduir le bruit
grey_image_gaussian = ndimage.gaussian_filter( image*255 , 3)
```

3 : le filtre gaussian : Un filtre gaussien est un filtre passe-bas utilisé pour réduire le bruit

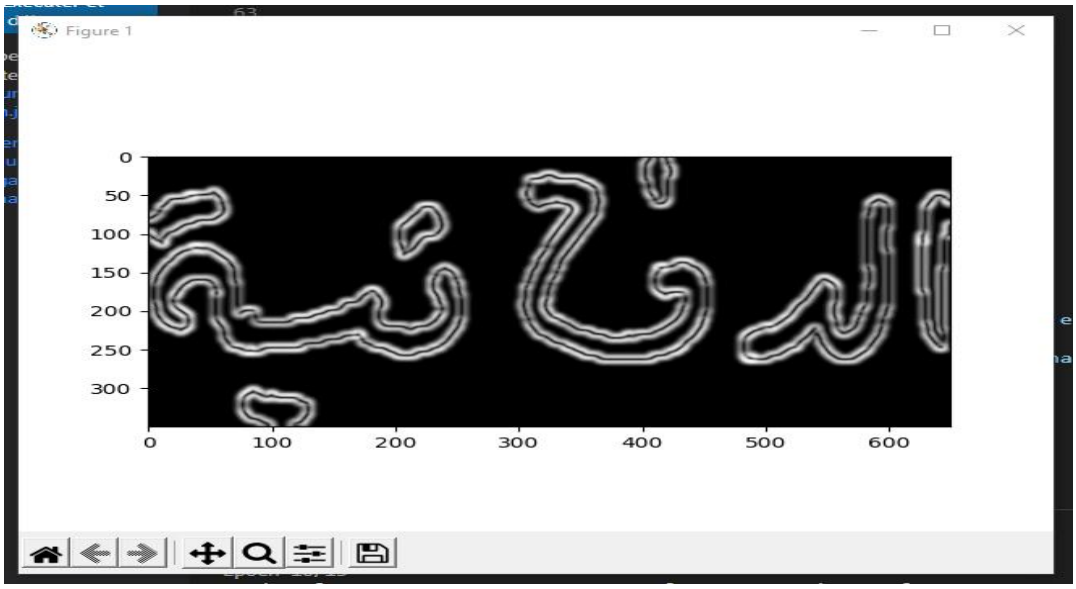


Figure 37 : résultat en appliquant le filtre gaussian

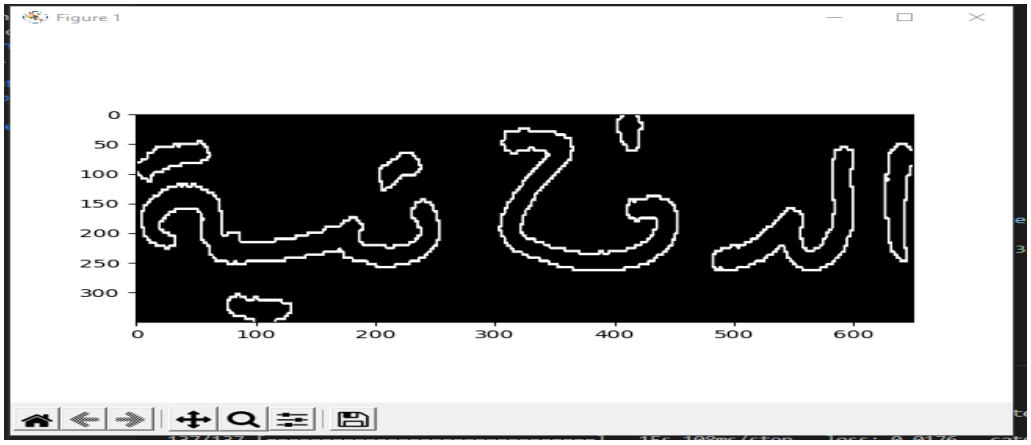


Figure 38 : résultat en appliquant la segmentation

```
#la squelittisation
img = cv2.imread('C:\\Users\\pc\\Desktop\\Nouveau dossier\\data_2021\\bdh\\'+image_path)

img = img.max(2)
T_otsu = mahotas.otsu(img)
img = img > T_otsu
print("Image threshold using Otsu Mehtod")
imshow(img)
show()
new_img = mahotas.thin(img)
print("Skeletonised Image")
imshow(new_img)
show()
```

3 : dans cette partie en a utiliser la squelettisation pour squelettiser les mots dans les images

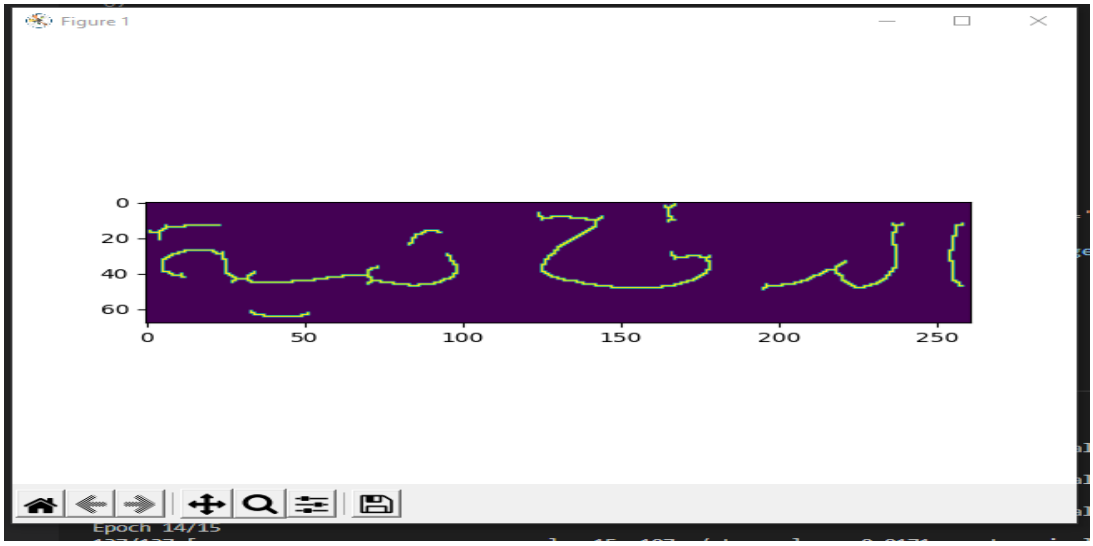


Figure 39 : résultat en appliquant la squelettisation

```
# Tkinter UI
window = Tk()

window.title("Handwritten Arabic text Recognition")
window.geometry('750x250')


# The browse function
def browsefunc():
    browsefunc.file_name = filedialog.askopenfilename()
    predictionfunc()

    # Display the predicted letter
    letter = dic[int(predictionfunc.prediction_index)]
    path_label.config(text=letter, font=("Courier", 30))

    # Display the selected image
    img = Image.open(Path(browsefunc.file_name))
    img = img.resize((350, 150), Image.ANTIALIAS)
    img = ImageTk.PhotoImage(img)
    img_label.config(image=img)
    img_label.image = img

browse_button = Button(window, text="Browse", command=browsefunc)
browse_button.grid(column=2, row=3)
browse_button.config(font=("Courier", 20))

path_label = Label(window)
path_label.grid(column=10, row=10)

img_label = Label(window)
```

4 : en utilisant (Tkinter) on a créé une simple interface pour notre application

7.3 Taux de précision du modelé

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

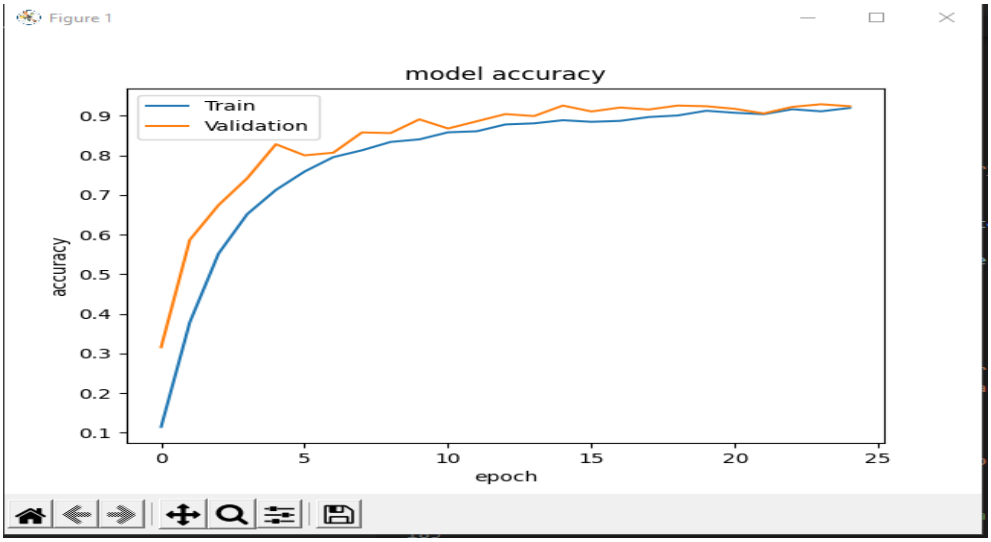


Figure 40 : TAUX DE PRECISION DU MODELE.

D'après la Figure 40 La précision de l'apprentissage et de test augmente avec le nombre d'époque, ceci reflète qu'à chaque époque le modèle apprenne plus d'informations. Si la précision est diminuée alors on aura besoin de plus d'information pour faire apprendre notre modèle et par conséquent on doit augmenter le nombre d'époque.

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

Dans cette section, la performance de CNN a été étudiée pour la formation et la reconnaissance des mots arabes.

Les expériences sont réalisées dans un environnement de programmation python avec la bibliothèque TensorFlow et API keras.

Au début, pour évaluer les performances de CNN sur les mots arabes, une approche de formation progressive a été utilisée pour l'approche proposée. Le taux de classification des données d'apprentissage a atteint 92% pour les époques de 20 à 25. Notre approche est conçue pour 25 époques, mais à partir de 20, le CNN affiche un taux d'erreur declassification faible.

8. Quelques résultats expérimentaux

#cette image représente un test sur un échantillon de 90 images

#chaque classe et sa précision, recall et f1-score

	precision	recall	f1-score	support
class 1	0.6667	1.0000	0.8000	4
class 2	1.0000	1.0000	1.0000	5
class 3	1.0000	0.6667	0.8000	6
class 4	1.0000	0.6000	0.7500	5
class 5	1.0000	1.0000	1.0000	3
class 6	0.6000	0.6000	0.6000	5
class 7	1.0000	1.0000	1.0000	3
class 8	1.0000	0.6667	0.8000	3
class 9	0.8000	1.0000	0.8889	4
class 10	1.0000	1.0000	1.0000	3
class 11	1.0000	0.8000	0.8889	5
class 12	1.0000	1.0000	1.0000	5
class 13	1.0000	0.7500	0.8571	4
class 14	0.8000	1.0000	0.8889	4
class 15	1.0000	1.0000	1.0000	4
class 16	1.0000	1.0000	1.0000	4
class 17	1.0000	1.0000	1.0000	4
class 18	1.0000	1.0000	1.0000	5
class 19	1.0000	1.0000	1.0000	5
class 20	0.5000	1.0000	0.6667	3
class 21	1.0000	1.0000	1.0000	3
class 22	1.0000	1.0000	1.0000	3
accuracy			0.9000	90

Figure 41 : image représente la précision le recall et f1-score pour chaque classe

Voici quelques figures qui montrent les résultats de test

Figure 42 : résultat de test 1

Pour ce premier test on a utilisé une image de la base donnes utilisé dans l'entrainement du modelé

Cette image représente la classe 21 d'après la figure 41 sa précision sur le test des 90 images est : 100%

Le system a bien reconnu le mot

Figure 43 : résultat de test 2

Pour ce 2eme test on a utilisé une image de la partie du test de la base de donnes, le system na vu cette image dans la phase d'entrainement

Cette image représente la classe 3 d'après la figure 41 sa précision sur le test des 90 images est : 100%

Le system a bien reconnu la phrase

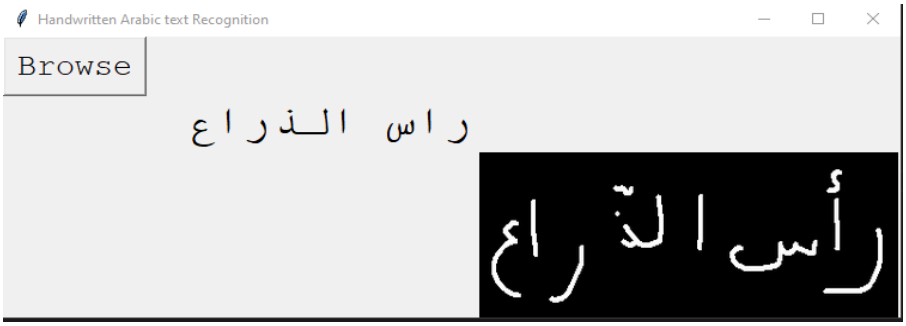


Figure 44 : résultat de test 3

Pour le 3eme test on utiliser une nouvelle image qu'on a créé en utilisant paint (on a écrit cette phrase montre dans la figure en utilisant paint)

Cette image représente la classe 6 d'après la figure 41 sa précision sur le test des 90 images est : 60%

Le system a bien reconnu cette écriture

A cause de petite taille de dataset le system ne donne pas des bons résultats avec toutes les nouvelles images

9. Matrice de confusion de notre test

Voici la matrice de confusion qui nous représente le résultat de test sur l'échantillon de 90 images

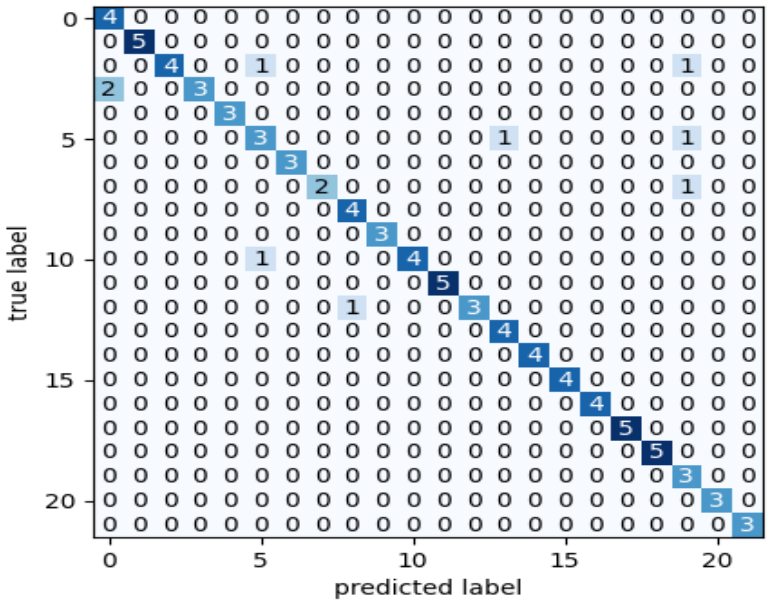


Figure 45 : matrice de confusion

On remarque que notre system a reconnu la majorité des mots dans notre test, avec une précision de 90% (dans ce test). Il Ya des prédictions fausse dans les classes 3, 4, 5, 7, 11,13

10. Discussion

Nous avons montré dans ce projet l'utilisation de deep learning et les RNC pour réaliser un modèle de classification de l'image pour faire la reconnaissance des mots arabes manuscrits. Malgré les difficultés trouvées et les manques de références on a réalisé ce system de reconnaissance des mots arabe manuscrits avec un taux de précision égale à 92%

11. Conclusion :

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

Les résultats obtenus sont très encourageants avec un taux d'exactitudes de classification de 92% et ce malgré la complexité des mots et la similitude entre eux.

Conclusion Générale

La capacité des ordinateurs à reconnaître le texte arabe manuscrit pose de nouveaux défis pour la recherche scientifique moderne.

La communication entre les humains et les appareils électroniques a considérablement augmenté alors que les chercheurs cherchaient à développer des programmes intelligents capables de reconnaître les lettres arabes manuscrites et de reconnaître les mots en un court laps de temps.

Tout au long de ce mémoire, on a présenté une étude théorique de la reconnaissance de texte manuscrit arabe et de l'apprentissage en profondeur à l'aide de RNC.

Notre modèle a été développé et testé dans la base de données IFN \ ENIT. Des résultats encourageront en termes de performances ont prouvé l'utilité de notre approche.

En résumé, ce projet nous a donné de nouvelles perspectives. Au cours de ce travail, on a découvert de nouveaux concepts tels que la reconnaissance manuscrite de l'arabe, le deep learning et le RNC, et on a pu ressentir spécifiquement les différentes difficultés liées à la mise en place d'un modèle de deep learning. On a trouvé des difficultés à cause de manque de références dans la partie de programmation ainsi que c'était une nouvelle expérience d'utiliser le deep learning et le CNN. Et au final nous pouvons dire que nous avons beaucoup appris de ce projet, et nous pouvons développer ce dernier et élargir la portée de la reconnaissance des phrases du system.

REFERENCES

[1] https://www.ibm.com/fr-fr/analytics/machine-learning.

[2] https://datasciencetoday.net/index.php/en-us/deep-learning/173-les-reseaux-de-neurones-convolutifs/.

[3] UNIVERSITE LARBI BEN M'HIDI -OUM EL BOUAGHI-MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLOME DE MASTER EN INFORMATIQUE-Thème : « La reconnaissance des caractères Arabes manuscrits par les Transformateurs »-PRESENTE PAR : Benmedjber kahina ANNEE UNIVERSITAIRE 2020/2021.

[4] http://staff.univ-batna2.dz/sites/default/files/merzougui_ghalia/files/support_de_cours_-deep_learning-chapitre3-cnn.pdf

[5] Université Mohamed Khaider de Biskra -louam_abdelhak_bilal.pdf - MÉMOIRE DE MASTER-thème : <Deep Learning basé sur les méthodes de réduction pour la reconnaissance de visage>-Année universitaire : 2018 – 2019.

[6] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In ACCV, 2010.

[7] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[8] T. Wang, D. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE.

[9] P. J.Werbos, "Generalization of backpropagation with application to a recurrent gas market model," Neural networks, vol. 1, no. 4, 1988.

[10] Deep Learning : définition, applications, avantages et inconvénients (retengr.com)

[11] https://analyticsinsights.io/apprentissage-supervise-vs-non-supervise/

[12] Université Blida 1 - MEMOIRE DE FIN D'ETUDE theme : La reconnaissance des caractères arabes manuscrits par les réseaux des neurones convolutionnels Présenté et soutenu publiquement par : Daoud Fouad et Louali Farouk

[13] P.M. Lallican, C. Viarp-Gaudin, S. Knerr : « From off-line to on-line handwriting recognition ».Proc. 7th workshop on frontiers in handwriting recognition, pp. 303-312, Amsterdam 2000.

[14] B. Al-Badr, S.A. Mahmoud: « Survey and bibliography of Arabic optical text recognition ».Signal processing, vol. 41, pp. 49-77, 1995.

[15] Depuis https://stanford.edu/~shervine/l/fr/teaching/cs-230/ écrite Par Afshine Amidi et Shervine Amidi (Ecole Centrale Paris, Stanford University)

[16] I.R. Tsang: «Pattern recognition and complex systems». Thèse de doctorat, universitéd'Anterwerpen, 2000.

[17] J. Trenkle, A. Gillies, S.Schlosser: « An off-line Arabic recognition system for machine printed documents ». Proc. Of the symposium on document image understanding technology (SDIUT'97), pp. 155-161 1997.

[18] N. Benamara « Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée ». Thèse de doctorat, spécialité Génie Electrique, Université des sciences, des Techniques et de médecine de Tunis II, 1999.

[19] A. Cornuéjols, L. Miclet, Y.Kodratoff, « Apprentissage Artificiel, Concepts et algorithmes » ISBN 2-212-11020-0, 2002.

[20] 10 concepts to understand machine learning by: “marc sanselme”

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

[22] Master thesis Ignace 2018 - IRIT d'images médicales par apprentissage profond. Présenté le 09 mars 2018 par : RANDRIANARIVONY Mamitiana Ignace.

[23] http://cs231n.github.io/convolutional-networks/ 14. « Convolutional Neural Networks (LeNet) – DeepLearning 0.1

[24] B. Al-Badr , S.A. Mahmoud : « Survey and bibliography of Arabic optical text recognition ».Signal processing , vol. 41, pp. 49-77,1995.

[25] « Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation » [archive], sur DeepLearning 0.1, LISA Lab (consulté le 31 août 2013).

[26] Université Abou Bakr Belkaid Tlemcen Faculté Des Sciences Département D’informatique - Mémoire de fin d’études Pour l’obtention du diplôme de Master en informatique OPTION : Modèle Intelligent et Décision -Thème Classification des images avec les réseaux de neurones convolutionnels -Réalisé par : Mr Mokri Mohammed Zakaria Présenté le 03/07/2017.

[27] depuis : « https://bjpcjp.github.io/pdfs/math/cnns-dive.pdf»

[28] Depuis:<https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>-by Pranoy Radhakrishnan Aug 9, 2017 (machine learning developer).

[29] auteur: HOURRANE Oumaima PhD. sous la direction de Pr. Ben Lahmar à la faculté des sciences Ben M'Sik. 26 juin 2018.

[30] *Source:* www.kaggle.com

[31] Robert Kwiatkowski May 22, 2021 sur https://towardsdatascience.com/

[32] by Jason Brownlee on November 18, 2016 in Code Algorithms From Scratch

[33] Pranshu Sharma — October 30, 2021 in https://www.analyticsvidhya.com/

[34] M. Pechwitz, S. Snoussi Maddouri, V. Maergner, N. Ellouze, and H. Amiri, "IFN/ENIT database of handwritten arabic words," CIFED, 2002.

[35] MEMOIRE de fin d'étude Présenté pour l'obtention du diplôme de MASTER Par: Baza Halima Saadia 2015 /2016 UNIVERSITE MOHAMED BOUDIAF - M'SILA

[36] Auteur(s) : Jean-Pierre CRETTEZ, Guy LORETTE Date de publication : 10 févr. 1998

[37] Abdel Belaïd LORIA-CNRS Campus scientifique B.P. 239 54506 Vandoeuvre-Lès-nancy

[38] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In International Conference on Computer Vision and Pattern Recognition (CVPR), pages I–511–I–518, 2001.

[39] Viola and M.J. Jones. Robust real-time face detection. International Journal of Computer Vision, 57(2):137–154, 2004.

[40] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10):255–258, 1995.

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[42] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. Nature, 323(6088):533–536, 1986.

[43] J. Werbos. Backpropagation: Past and future. In IEEE International Conference on Neural Networks, pages 343–353, 1988.

[44] K. Fukushima and S. Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. Pattern recognition, 15(6):455–469, 1982.

[45] LeCun et al. Generalization and network design strategies. Connections in Perspective, pages 143–55, 1989.

[46] Sonia Yousfi. Embedded Arabic text detection and recognition in videos. Document and Text Processing. Université de Lyon, 2016. English. ffN

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

[48] Université Mohamed Sadik Benyahia de Jijel-Mémoire de fin d'étude pour obtention du diplôme Master de Recherche en Informatique Thème : « Reconnaissance d'images par les réseaux de neurones convolutifs » Préparé par Hani Zerzaihi et Fouad Zarour Année Universitaire 2019/2020.

[49] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.

[50] T. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. International Conference on Computer Vision Theory and Applications, 2009.

[51] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE.

[52] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Computer Science Department, University of Toronto, and Technical Report 1.4 (2009).

[53] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competition. ICDAR, 2003.

[54] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In ACCV, 2010.

[55] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[56] T. Wang, D. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE.

[57] Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. Systems, Man and Cybernetics, IEEE Transactions on, (6), 610-621.

[58] Hu, Y., Zhao, C. X., & Wang, H. N. (2008, December). Directional analysis of texture images using gray level co-occurrence matrix. In Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on (Vol. 2, pp. 277-281). IEEE.

[59] "Local Binary Patterns: New Variants and Applications."

[60] O. Dniz, G. Bueno, J. Salido, and F. De la Torre, "Face recognition using Histograms of Oriented Gradients," vol. 32, no. 12, pp. 1598– 1603.

[61] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893.

[62] B. Al-Badr, S.A. Mahmoud: « Survey and bibliography of Arabic optical text recognition ».Signal processing, vol. 41, pp. 49-77, 1995.

[63] https://www.researchgate.net/publication/228904501_IFNENITdatabase_of_handwritten_Arabic_words

[64] Depuis:https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/

[65] Depuis: https://datascientest.com/pandas-python-data-science

[66] Depuis: https://datascientest.com/matplotlib-tout-savoir

[67] Frederic Lardinois, « Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux And Windows », *TechCrunch*, 29 avril 2015 (lire en ligne [archive]).

[68] « Stack Overflow Developer Survey 2021 » [archive], sur *Stack Overflow* (consulté le 25 janvier 2022)

[69] 28 janvier 2019/ Machine Learning/ Par Zakariyaa ISMAILI sur :< https://archive.wikiwix.com/cache/index2.php?url=https%3A%2F%2Fle-datascientist.fr%2Fapprentissage-supervise-vs-non supervise#federation=archive.wikiwix.com>

[70] Depuis:<https://ichi.pro/fr/guide-des-matrices-de-confusion-et-des-mesures-de-performance-de-classification-176935406157966>

[71] https://www.inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simpl

[72] Depuis:<https://wikilivre.org/wiki/une-hyperbole-est-elle-un-sophisme/>

[73] UEUNIVERSITE KASDI MERBAH OUARGLA MEMOIRE DE FIN D'ETUDE Présenté et soutenu publiquement par : LAKHDARIAhmedToufiket ABBACI Salih le 23\05\2017.