

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DE Technologie

DEPARTEMENT Electronique

N° :



FILIÈRE : Electronique

OPTION : électroniques des systèmes
embarqués

Mémoire présenté pour l'obtention

Du diplôme de Master Académique

Par:

BEN AISSA CHAIMA

et

CHERGUI AMINA

THÈME

Reconnaissance des chiffres en utilisant un CNN

Soutenu devant le jury composé de :

Dr. BRIK Youcef	Université M ^{ed} Boudiaf –M'sila	Président
Dr. DERDOUR KHedidja	Université M ^{ed} Boudiaf –M'sila	Rapporteur
Dr. ATTALLAH Bilal	Université M ^{ed} Boudiaf –M'sila	Examineur

Année universitaire : 2021 /2022

Table des matières

Introduction générale	1
Chapitre I: Reconnaissance des formes	1
I-1 Introduction	1
I-2 Définition	1
I-3 Domaines d'application	1
I-4 Schéma générale d'un système RdF	2
I-4-1 Acquisition de données	3
I-4-2 Prétraitement	4
a- Suppression du bruit	4
b- Binarisation	4
c- Squelettisation	5
d- Redressement de l'écriture.....	5
e- la normalisation	5
I-4-3 Extraction de caractéristiques	6
I-4-3-1 Techniques d'extraction des caractéristiques	7
a- Techniques structurelles et topologiques	7
b- Techniques statistiques	7
c- Techniques basées sur des transformations globales	7
I-4-4 La classification (Modélisation)	8
I-4-4-1 L'apprentissage.....	8
a- Apprentissage supervisé	8
b- Apprentissage non supervisée	8
I-4-5 Post-Traitement.....	9
I-5 Conclusion	9
Chapitre II: Convolutional Neural Network (CNN)	
II-2 Introduction	11
II-3 Définition	11
II-3 Historique	12
II-4 Les couches de réseaux de neurones convolutionnels :	13
II-4-1 Couches d'entrée.....	13
II-4-2 Couches convolutionnelles	14
II-4-3 Pooling :	16
II-4-4 Couch fully-connected:	17

II-5 Architecture d'un CNN :.....	17
II-6Le paramétrage des couches :.....	19
II-7Avantages de CNN :.....	20
II-7 Conclusion :.....	21
Chapitre III : Implémentation et résultats expérimentaux	22
III-1 Introduction :	22
III-2 l'environnement de développement :	22
III-2-1Langage de programmation et bibliothèques utilisées :	22
III -2-1-1 Python :.....	22
III -2-1-2 PyCharm	23
III -2-2 Bibliothèques utilisées	23
a- TensorFlow	23
b- Keras.....	23
c- Numpy	24
d-Pilow.....	24
III-2-2La base de données MNIST	24
III-3 Implémentation	26
III-3-1 Importation des bibliothèques nécessaires.....	26
III-3-2 Traitement de la base donnée.....	26
III-3-3 Construction du modèle CNN	27
III-3-4 Compilation et ajustement Évaluation du modèle	28
III-4 Résultats expérimentaux	29
III-5Comparaison aux autres travaux.....	33
III-6 conclusion	33
Conclusion générale.....	35
Bibliographie	37

Liste des figures

Figure I.1 : Schéma général d'un système de RdF	3
Figure I.2. Squelettisation.....	5
Figure II. 1 RNC et la vision par ordinateur	12
Figure II. 2 Les couches de CNN.....	13
Figure II.2. Volume 3D de la couche d'entrée.....	14
Figure II-4: Exemple simpliste des valeurs.....	15
Figure II-5: Exemple de valeurs d'une des pixels d'une image 5x5 matrices utilisées comme filtre.....	15
Figure II-6: les valeurs devraient varier entre 0 et 255.....	15
Figure II-7: Le filtre	16
Figure II-8: Pooling	16
Figure II-9 :Exemple d'un véhicule en utilisant le CNN	18
Figure II-10 :un exemple d'une couche convolution	19
Figure III.1 : Quelques échantillons extraits de la base MNIST	25
Figure III.2 : exemple de numéro 8 dans la base MNIST	26
Figure III.3 : Importation des bibliothèques	27
Figure III.4 : le modèle CNN utilisée.....	28
Figure III.5: téléchargent de la base MNIST	29
Figure III.6:Architecture du notre modèle CNN	29
Figure III.7:Entrainement du notre modèle	30
Figure III.8 Représentation de l'erreur en fonction des epochs	31
Figure III.9: des autres exemples de test	32

Liste des tableaux

Tableau 1 : Tableau individu/caractère.....	6
Tableau 2 : la répartition de la base MNIST pour les chiffres	25
Tableau3 : Tableau de comparaison	32

REMERCIEMENT

Nous remercions avant tout Dieu tout puissant de nous avoir donné la force et la volonté pour Accomplir ce modeste travail.

Nous tenons à remercier le Docteur Brik Youcef président du jury et monsieur l'examineur le Docteur Attallah Bilal, membres de jury pour avoir accepté de juger et d'évaluer ce travail.

Nous tenons à remercier vivement nos encadreur, Dr Kh. Derdour, d'avoir accepté de nous guider tous le long de ce travail. Et cela, avec leur grande disponibilité, leur rigueur scientifique et leurs précieux conseils qui nous ont permis de travailler dans les meilleures conditions, et ce, depuis le début de la préparation de notre projet.

Nos remerciements vont aussi au corps académique de l'Université de M'sila en général et en particulier les enseignants du département de Génie électronique qui ont contribué à notre formation.

Nous voudrions exprimer notre gratitude et notre remerciements particuliers à l'étudiant Mr. BarakatLahcen, ; Et BouguerraYaacoub, professeur à CFPA de M'Sila 03 pour leur générosité et leur soutien scientifique.

Enfin, nous voudrions adresser nos vifs remerciements à tous à tous les amis et collègues de la promotion.

DEDICACE

A ma très chère mère

Quoi que je fasse ou que je dise, je ne saurai point te remercier comme il se doit. Ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force pour affronter les différents obstacles.

A mon très cher père

Tu as toujours été à mes côtés pour me soutenir et m'encourager. Que ce travail traduit ma gratitude et mon affection.

A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès.

A tous ceux que j'aime.

Chaima

DEDICACE

*Toutes les lettres ne sauraient trouver les mots qu'il faut...
Tous les mots ne sauraient exprimer la gratitude,
L'amour, le respect, la reconnaissance...
Aussi, c'est tout simplement que
Je dédie ce mémoire ...*

À mon père

Propriétaire d'un visage bienveillant et de bonnes actions, il ne m'a pas épargné toute sa vie (mon cher père).

A ma mère

A qui je le préfère à moi-même et pourquoi pas, car elle s'est sacrifiée pour moi, et n'a ménagé aucun effort pour me rendre toujours heureux (ma mère bien-aimée).

A mon très cher frère Hassan, sa femme et leur fille (Djouri)

*Mon cher frère qui est mon père et ma mère, les mots me suffisent à peine
Exprimez l'attachement, l'amour et l'affection que j'ai pour vous.*

*Mon ange gardien et compagnon fidèle dans les moments les plus sensibles
Cette vie mystérieuse. Je vous dédie ce travail avec mes meilleurs vœux
Santé et réussite.*

A mon très cher frère Salim,

Je te souhaite tout le bonheur dans le monde et que tu réalises tes rêves.

A mes compagnes de chemin et mon soutien tout au long de ma carrière scientifique, mes sœurs, Samira, Farida, Asma, Bouchra

À mes amis et à tous ceux qui m'ont soutenu et m'ont aidé avec tout ce qu'ils avaient et de bien des façons.

AMINA

Résumé

La reconnaissance de chiffres manuscrits présente un défi très grand et joue un rôle très important dans le monde actuel pour rendre les machines capables de connaître comme un homme et capable de résoudre des problèmes complexes tel que la reconnaissance de chiffres du (chèque bancaire, relevé des compteurs d'électricité, gaz et eau, etc....).

Les travaux de ce mémoire s'inscrivent dans le cadre de la reconnaissance automatique des chiffres manuscrits. L'approche proposée consiste à utiliser un CNN pour reconnaître les chiffres en utilisant la base MNIST. Notre modèle a été entraîné avec succès avec un taux de reconnaissance qui vaut 97.03%.

Mots clés : Reconnaissance de chiffres, CNN, Python, Apprentissage profond.

Abstract

Recognition of handwritten digits presents a very big challenge and plays a very important role in today's world to make machines able to know like a man and able to solve complex problems such as the recognition of digits from (bank check, statement of electricity, gas and water ...).

The work of this thesis is part of the automatic recognition of handwritten digits. The proposed approach consists of using the CNN to recognize digits using the MNIST database. our model has been successfully trained with an accuracy rate of 97.03%.

Keywords: digit recognition, CNN, Python, deep learning

ملخص

يمثل التعرف على الأرقام المكتوبة بخط اليد تحديًا كبيرًا ويلعب دورًا مهمًا للغاية في حاليا لجعل الآلات قادرة على التعرف كالإنسان وقادرة على حل المشكلات المعقدة مثل التعرف على الأرقام من (شيك مصرفي، كشف الكهرباء، الغاز وعدادات المياه، إلخ).

يعد عمل هذه الأطروحة جزءًا من التعرف التلقائي على الأرقام المكتوبة بخط اليد. يتمثل النهج المقترح في استخدام CNN للتعرف على الأرقام باستخدام قاعدة بيانات MNIST. تم تدريب (تعليم) نموذجنا بنجاح بنسبة 97.03% الكلمات المفتاحية: التعرف على الأرقام، CNN، بايثون، التعلم العميق.

Introduction Générale

Introduction générale

Depuis la fin des années soixante, des travaux intensifs ont été accomplis dans le domaine de la reconnaissance de l'écriture. En effet, des systèmes commerciaux sont aujourd'hui indisponibles, particulièrement pour la poste (lecture des chiffres des chèques, tri postale) et dans les banques (traitement des chèques, des factures). À l'heure actuelle, les recherches s'orientent vers l'interprétation de l'écriture manuscrite.

Dans ce mémoire, nous nous intéressons à la reconnaissance des chiffres manuscrits avec le CNN qui reste aujourd'hui un thème de recherche ouvert.

En effet, bien que le nombre de classes naturelles soit très réduit (chiffres '0' à '9'), on trouve à l'intérieur de chacune d'entre elles, une très grande variabilité de l'écriture, de plus, les conditions souvent relativement précaires dans lesquelles sont écrits les chiffres (chèques écrits rapidement sur un coin de table) et la variabilité du matériel utilisé (utilisation de divers stylos, de différentes qualités de papier) tendent à compliquer la reconnaissance.

Un réseau de neurones convolutifs (ou réseau de neurones à convolution, ou CNN ou ConvNet) 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. Les neurones de cette région du cerveau sont arrangés de sorte à ce qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage multi-couche de perceptrons, dont le but est de prétraiter³ de petites quantités d'informations. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel.

Dans un réseau de neurones convolutif (CNN), les couches sont disposées dans une matrice 3D (coordonnée sur l'axe des X, coordonnée sur l'axe des Y et couleur). En conséquence, un nœud de la couche cachée ne serait connecté qu'à une petite région située au voisinage de la couche d'entrée correspondante, ce qui rendrait le processus beaucoup plus efficace qu'un réseau de neurones traditionnel.

Nous allons utiliser la base de données MNIST de chiffres manuscrits. Nous allons former les données sur le kit d'entraînement (données d'apprentissage) et valider les résultats en fonction des données de test.

L'objet de ce mémoire est l'implémentation de CNN pour la reconnaissance des chiffres manuscrits en utilisant le langage de programmation python.

Ce mémoire est organisée en 3 chapitres :

- Le premier chapitre présentera les différentes étapes du processus de reconnaissance de formes.
- Le Deuxième chapitre présente le réseau de neurones convolutif (CNN)
- Le dernier chapitre, contient les résultats obtenus par notre application, avec la discussion et nous allons définir notre architecture de CNN mise en point pour la reconnaissance des chiffres manuscrits sur la base MNIST, via le langage de programmation python. Nous commençons tout d'abord par la présentation des ressources, du langage et de l'environnement de développement que nous avons utilisé. Puis les étapes de la réalisation du modèle et on termine par les tests effectués.

Chapitre01



Reconnaissance des formes



I-1 Introduction :

La reconnaissance de formes est une branche de l'intelligence artificielle qui fait appel aux techniques d'apprentissage automatique et à diverses techniques mathématiques. Elle peut être définie comme un ensemble de méthodes visant à identifier des objets à partir de données brutes caractérisées ou des paramètres. Dans ce chapitre nous présenterons les différentes étapes du processus de reconnaissance de formes.

I-2 Définition :

La reconnaissance des formes (RdF) a pour objet de concevoir et d'analyser des algorithmes de classification applicables aux représentations d'images, de sons ou de signaux de toute nature. Un exemple typique est la reconnaissance de caractères qui permet de transformer l'image d'une page imprimée en une suite de caractères, faisant d'ailleurs passer sa « taille » mémoire de quelques millions d'octets à quelques milliers. **(S.Watanabe, 1985)** On considère que c'est une branche de l'intelligence artificielle qui fait largement appel aux techniques d'apprentissage automatique et aux statistiques. **(D.DJEZZAR, M.Rafik, & F.BENKADA, 2022)**

Le problème que cherche à résoudre la RdF est d'associer une classe à une forme inconnue (qui n'a pas encore de classe associée). On considère souvent la RdF comme un problème de classification : trouver la fonction qui affecte à toute forme inconnue sa classe la plus pertinente. Elle est partie intégrante de tout système intelligent destiné à la prise de décision **(D.DJEZZAR, M.Rafik, & F.BENKADA, 2022)**. Une forme est un ensemble de valeurs, appelés attributs, auxquels est associé un nom (ou étiquette), qui est leur classe. Plusieurs formes peuvent avoir la même classe, on dit alors que ce sont les exemples ou réalisations de la classe. **(S. Theodoridis, K. Koutroumbas, 2003)**

I-3 Domaines d'application :

RdF est utilisée dans plusieurs domaines d'activité. Parmi ces domaines on peut citer :

- Vision artificielle : En donnant à la machine la possibilité de voir, l'homme a certainement franchi un pas important dans l'automatisation de ses tâches quotidiennes. Les progrès techniques, tant au niveau des caméras que des systèmes informatiques de traitement des images numériques, ont permis un élargissement considérable du champ d'application de ce qu'il est convenu d'appeler la vision artificielle. La vision comprend le traitement,

l'analyse et l'interprétation des images. Les applications sont très nombreuses. On peut citer la sûreté avec les empreintes digitales, la médecine avec l'analyse d'images de radiographie, d'échographie ou IRM, la vision sous-marine, le secteur industriel où des machines de vision sont couramment utilisées pour l'inspection et le contrôle de qualité, le tri automatique, la lecture automatique

- Reconnaissance automatique de la parole : Les systèmes de reconnaissance automatique de la parole (SRAP) ont pour objectif de transcrire un message oral en texte (**M.Bahashwan, Y.Tindoras, & Z.figo, Efficient segmentation of Arabic handwritten characters using structural caractéristiques, 2015**). Les principales applications sont la transcription automatique, l'indexation de documents multimédias, le codage acoustico-phonétique et le dialogue homme machine. Parmi les applications les plus courantes, la dictée automatique, la commande vocale, la traduction en temps réel de langues étrangères (**M.E & P, 2002**)
- Biométrie : est une discipline adoptée pour la reconnaissance et la vérification de l'identité sur un dispositif portable.
- Finances : pour la détection de transactions bancaires frauduleuses ainsi que la prédiction des banqueroutes. (**Kaastra & Boyd, 1996**)
- Usinage : la qualité des produits dépend souvent de la paramétrisation correcte, et les relations exactes entre la qualité et les valeurs des paramètres n'est pas claire. Les systèmes de reconnaissance des formes sont utilisés pour classer les paramètres selon la qualité des produits qu'ils sont susceptibles de générer. Ils permettent ainsi de réduire le nombre d'essais ce qui fait gagner du temps et de l'argent. (**S, K, W, & D.S, 1995**)
- Energie : les systèmes de reconnaissance des formes sont utilisés pour prévoir la consommation électrique (réduite, normale, élevée), permettant ainsi aux clients de réduire si nécessaire leur consommation, et aux producteurs de mieux gérer leurs unités de production.

I-4 Schéma générale d'un système RdF :

Une chaîne de traitement dans un système RdF comprend plusieurs modules (Figure 1.1) et plusieurs espaces de travail. L'objectif de la reconnaissance des formes va être de définir une suite d'opérations permettant de passer de l'espace des données ou formes, à l'espace des classes ou la catégorie de la forme est estimée. Ces opérations sont en pratique des procédures informatisées.

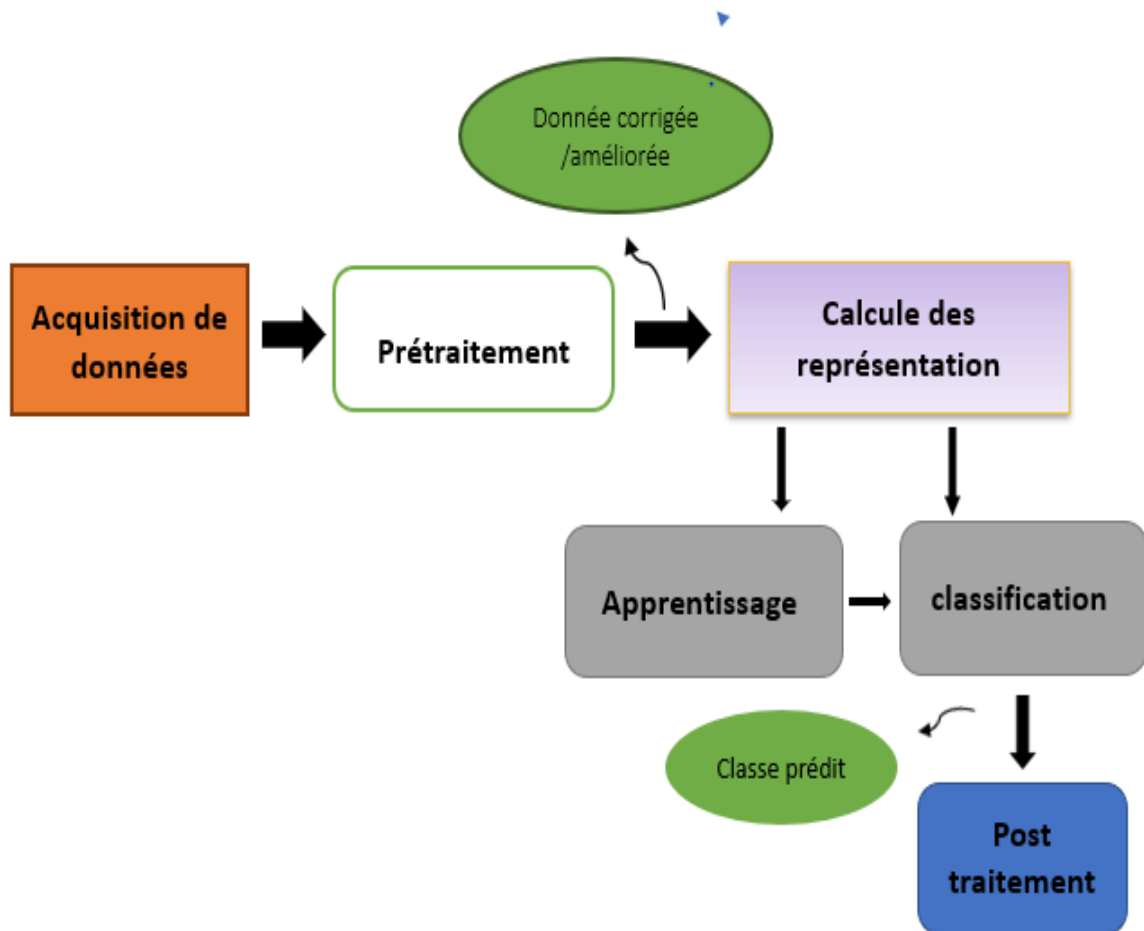


Figure I.1 : Schéma général d'un système de RdF

I-4-1 Acquisition de données :

Elle consiste à convertir une vue du monde réel en une image numérique. Dans une application, les images acquises peuvent être de type et de nature différentes et présentées en niveaux de gris, en couleur ou en texte. L'acquisition se fait à l'aide de différents moyens (microphone, caméra, instruments de mesure).

L'acquisition des données peut être faite par une tablette en cas de reconnaissance en ligne (temps réel) où la numérisation des scripts dépend de la résolution de la tablette et de la vitesse d'échantillonnage. Dans le cas de la reconnaissance hors ligne, l'acquisition est effectuée par un scanner où la qualité des données acquises dépend de la résolution de ce dernier. (J.Pierre, 1999)

I-4-2 Prétraitement :

Le prétraitement a pour but de préparer l'image du tracé à la phase suivante d'analyse [9]. Il s'agit essentiellement de réduire le bruit superposé aux données et ne garder, autant que possible, que l'information significative de la forme présentée. Le bruit peut être dû au dispositif d'acquisition, aux conditions d'acquisition (éclairage, mise incorrecte du document...), ou encore à la qualité du document d'origine (**G.Abdeldjalil, 2011**). Parmi les opérations de prétraitements généralement utilisées, citons :

a- Suppression du bruit :

Plusieurs traitements classiques sont mis en évidence, tels que la modification d'histogramme pour rehausser les contrastes ; corriger la luminosité et les techniques de filtrage pour atténuer le bruit. (**Y.Zhang, W.U, & A, 2011**)

Les opérateurs morphologiques en tant que techniques de lissage utilisent les opérateurs de base de dilatation et d'érosion ainsi que leurs compositions (ouverture et fermeture) pour corriger les défauts liés soit à l'absence de points (trous) ou à une surcharge de points au niveau du tracé d'écriture manuscrite (**R & L.O, 1995**)

Toutefois, il est important de mentionner que ces traitements destinés à corriger certains défauts peuvent avoir des effets négatifs sur d'autres éléments de l'image. Par exemple, un filtre passe-bas permet d'éliminer certains bruits mais rend les tracés d'écriture flous. D'où la nécessité de manier ces traitements avec prudence. (**J.Pierre, 1999**)

b- Binarisation :

C'est le passage d'une image en couleur ou définie par plusieurs niveaux de gris en image bitonale (composée de deux valeurs 0 et 1) qui permet une classification entre le fond (image du support papier en blanc) et la forme (traits des gravures et des caractères en noir). (**R.Firdousi & S.Parveen, 2014**) (**O.D & T, 1995**)

On distingue en général deux approches : le seuillage global et le seuillage adaptatif et le seuillage Local. Les techniques de seuillage local reportées dans la littérature, permettent de traiter des images extrêmement bruitées et de régler le problème de contraste de luminosité Une synthèse des méthodes de binarisation est proposée dans (**T.K, 2014**) (**K, Y, T, & j, 2002**).

c- Squelettisation :

Ce traitement aboutit à la représentation de l'information indépendamment de l'épaisseur initiale de l'écriture. Elle permet le passage d'une image à sa représentation en squelette. (S. Theodoridis, K. Koutroumbas, 2003) Le squelette a un pixel d'épaisseur et permet d'extraire les caractéristiques essentielles telles que les intersections, le nombre de tracés et leurs positions relatives (Figure 1.2).



Figure I.2. Squelettisation

d- Redressement de l'écriture :

L'un des problèmes rencontrés en OCR est l'inclinaison des lignes du texte, qui introduit des difficultés pour la segmentation. L'inclinaison peut provenir de la saisie, si le document a été placé en biais, ou être intrinsèque au texte. Il convient alors de le redresser horizontalement les lignes d'écriture obliques à l'aide d'une transformation géométrique de type rotation isométrique des points de l'image afin de retrouver la structure de lignes horizontales d'une image texte.

Le redressement de l'écriture se fait généralement en deux étapes (M. Cheriet, T. Z, & I, 2007). La première permet l'estimation de l'angle θ d'inclinaison globale de la ligne de base et la deuxième étape sert à corriger l'inclinaison par l'application d'une rotation de l'image d'un angle θ

e- Normalisation :

L'étape de normalisation a pour objectif de faire en sorte que l'extraction de caractéristiques soit la moins perturbée possible par la variabilité des styles d'écriture et de rendre l'écriture la plus indépendante possible du scripteur et de la taille.

Nous distinguons Trois techniques de normalisation :

- Correction de l'inclinaison des lignes
- Correction de l'inclinaison des lettres (chiffres)
- Lissage du contour

I-4-3 Extraction de caractéristiques :

L'étape de l'extraction des paramètres représente le cœur du système de reconnaissance, son rôle consiste en l'extraction des informations nécessaires qui reviennent à établir un modèle (S.F.BettaharAbdessettar, 2014). Mais elle pose un problème que plusieurs techniques d'extraction s'accompagnent d'une perte d'information. De ce fait, il faut effectuer un compromis entre la quantité et la qualité de l'information (A.Lemieux, 2003).

La réduction du nombre de caractéristiques a de nombreux avantages : elle permet d'améliorer la visualisation et la compréhension des données, de réduire les temps d'apprentissage et de classification des systèmes, d'améliorer les performances en classification, et permet de réduire la taille des bases d'apprentissage. (R.Firdousi & S.Parveen, 2014)

Ces caractéristiques discriminantes doivent aussi être invariantes à certaines transformations. Ainsi, chaque objet O_i , $i=1, 2,3,\dots, n$, est représenté par m paramètres qui le caractérisent l'on obtient un tableau de données de dimension $m.n$ (tableau 1).

Tableau 1 : Tableau individu/caractère

Caractère Individu	X_1	X_2	X_i	X_m
X_1	X_{11}	X_{12}	X_{1i}	X_{1m}
X_2	X_{21}	X_{22}	X_{2i}	X_{2m}
.
.
.
X_i	X_{i1}	X_{i2}	X_{ij}	X_{im}
.
.
X_n	X_{n1}	X_{n2}	X_{nj}	X_{nm}

Les paramètres qui peuvent décrire la forme d'un objet sont très nombreux. Ces paramètres peuvent aller de la simple information telle que la surface de l'objet ou son périmètre jusqu'au calcul d'autres paramètres plus robustes aux transformations géométriques comme la translation, la rotation et le changement d'échelle.

On les classe selon trois paramètres essentiels : Paramètres de forme, de couleur et de texture.

I-4-3-1 Techniques d'extraction des caractéristiques :

Généralement les techniques d'extraction de caractéristiques pour la reconnaissance de caractères peuvent être classées en trois catégories : structurelles et topologiques, statistiques et celles basées sur des transformations globales.

a- Techniques structurelles et topologiques :

Le principe consiste à décomposer l'image du caractère en extrayant des formes élémentaires, appelées aussi primitives. Ces primitives peuvent être extraites des données brutes ou à partir d'une autre représentation du caractère (squelette ou contour). Elles sont généralement très robustes vis à vis de la rotation, la translation, l'homothétie. **(R.Gaur, T.Mino, & D, 2017)**

Il s'agit principalement de segments de droite (stroke), des arcs, des boucles, des concavités et les convexités dans les quatre directions principales, les points d'intersections et croisement et les points terminaux, la hauteur et la largeur du caractère, le nombre de points diacritiques et leur position par rapport à la ligne de base dans le cas de l'écriture arabe et la mesure de surfaces et périmètres. **(M.Bahashwan, Y.Tindoras, & Z.figo, Efficient segmentation of Arabic handwritten characters using structural caractéristiques, 2015)**

b- Techniques statistiques :

Les méthodes statistiques cherchent à extraire à partir de la distribution des pixels noirs de la forme de chaque caractère (ou du mot) des caractéristiques qui la décrivent localement. Ces caractéristiques permettent ainsi de représenter chaque caractère sous forme d'un vecteur de caractéristiques. L'ensemble de ces vecteurs définit ainsi un nouvel espace de représentation des données initiales et qui sont ensuite utilisés par un classifieur pour distinguer les différentes classes à l'intérieur de cet espace. Différents types ont été suggérées. Parmi elles, on cite celles basées sur les histogrammes de projection, les profils, le zonage, les moments géométriques invariants et les moments de Zernike. **(C.D.Aravinda, T.Nos, & R.david, 2014)**

c- Techniques basées sur des transformations globales :

Elles sont extraites à la suite d'une transformation de l'image. Dans cette famille, nous distinguons les caractéristiques basées sur la transformée de Fourier, les filtres de Gabor et la transformée en ondelettes. **(R.Gaur, T.Mino, & D, 2017)**

I-4-4 Classification (Modélisation) :

Cette étape consiste à modéliser les paramètres extraits. Un modèle est un ensemble d'informations utiles, discriminantes et non redondantes qui caractérise un ou plusieurs individus ayant des similarités. Les classes obtenues doivent être compactes et séparables ou éloignées les unes des autres

Parmi les nombreuses méthodes de classification automatique, on peut citer : La décision bayésienne, Les méthodes hiérarchiques, La classification par le k plus proches voisins, La méthode des k-means et Les réseaux de neurones artificiels.

I-4-4-1 Apprentissage :

C'est l'étape où on fait apprendre les individus au système, elle consiste à mémoriser les paramètres, après extraction, dans une base de données bien ordonnées (A.Ghali, 2015) pour faciliter la phase de reconnaissance et la prise d'une décision, elle est en quelque sorte la mémoire du système.

Lorsque les données d'entraînement sont préalablement classées, l'apprentissage est dit supervisé, sinon il est non supervisé. (S. Theodoridis, K. Koutroumbas, 2003)

a- Apprentissage supervisé :

Lorsque les classes des formes sont connues initialement, la RdF est supervisée. Les méthodes supervisées utilisent les formes connues, c'est-à-dire l'ensemble d'apprentissage, pour construire un classifieur qui sépare au mieux les différentes classes connues, dans le but de minimiser l'erreur de classification. Le modèle de chaque classe est alors représenté par une fonction d'appartenance qui détermine la valeur d'appartenance d'une forme à une classe.

b- Apprentissage non supervisée :

Lorsqu'aucune information n'est disponible sur les classes d'un système, la RdF est non supervisée. Les méthodes non supervisées sont basées sur des fonctions de similarité. Quand des formes aux caractéristiques similaires apparaissent, elles sont classifiées dans la même classe et à l'inverse quand leurs caractéristiques sont différentes une nouvelle classe est créée par le classifieur. Une fois que le classifieur a estimé les fonctions d'appartenance des classes, les nouvelles formes peuvent être assignées à la classe pour laquelle elles ont la valeur d'appartenance maximale. (N.Benahmed, 2002)

I-4-5 Post-Traitement :

Cette étape aide à réduire considérablement des erreurs et de corriger les résultats de la classification en utilisant des outils spécifiques au domaine d'application. Elle permet la validation des décisions de l'analyse sur la base de connaissances. Cependant, ce n'est pas une étape complètement séparée des autres étapes. **(A.Boucher & N.A.Tuan, Reconnaissance d'écriture manuscrite, 2005)**

Par exemple pour un système de reconnaissance de textes manuscrits, le classificateur se charge de classer chaque caractère séparément, alors que le post traitement applique un correcteur orthographique sur tout le texte pour valider et éventuellement corriger le résultat de la classification. Bien que facultative, cette phase permet d'améliorer considérablement la qualité de la reconnaissance. **(H.Nemmour & R.Diboune, 2007)**

I-5 Conclusion :

Dans ce chapitre, nous avons donné un aperçu sur la RdF, nous avons introduit la notion et le fonctionnement de la reconnaissance des formes et son processus qui est basé généralement sur : l'acquisition des données, prétraitement, extraction des caractéristiques, classification.

Dans le chapitre suivant nous allons nous intéresser à la méthode de classification basée sur les réseaux de neurones convolutionnels, voir le taux de précision qu'ils ont déjà obtenu auparavant, alors cette méthode sera l'objet principal de ce mémoire.

Chapitre02



Convolutional Neural Network

(CNN)



II-2 Introduction :

Dans ce chapitre, nous présentons Le CNN (Convolutional Neural Network), Les réseaux de neurones convolutifs désignent une sous-catégorie de réseaux de neurones. Cependant, les CNN sont spécialement conçus pour traiter des images en entrée. Leur architecture est alors plus spécifique elle est composée de deux blocs principaux.

Le premier bloc : fait la particularité de ce type de réseaux de neurones, puisqu'il fonctionne comme un extracteur de caractéristiques (caractéristiques). Pour cela, en appliquant des opérations de filtrage par convolution. La première couche filtre l'image avec plusieurs noyaux de convolution, et renvoie des "caractéristiques maps", qui sont ensuite normalisées (avec une fonction d'activation) et/ou redimensionnées. Ce procédé peut être réitéré plusieurs fois

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

II-3 Définition :

Un réseau de neurones convolutif (RNC) est un algorithme d'apprentissage profond. Ce réseau a été largement exploité en vision par ordinateur pour la classification et la détection des objets grâce à ses fonctionnalités inspirées du cortex visuel. (N.Malki, 2018/2019)

Les RNCs sont un type de réseau neuronal artificiel acyclique (Prédictifs), dans lequel le schéma de connexion entre les neurones est inspiré par le cortex visuel de l'animal. Les neurones de cette région du cerveau sont disposés de manière à correspondre aux régions qui se chevauchent lors du pavage du champ visuel.

Le but d'un CNN est d'apprendre des caractéristiques d'ordre supérieur dans les données via des convolutions. Ils sont bien adaptés à la reconnaissance des objets avec des images et à des concours de classement des images toujours supérieurs. Ils peuvent identifier les visages, les individus, les panneaux de rue, les ornithorynques et de nombreux autres aspects des données visuelles. Les CNN se chevauchent avec l'analyse de texte par reconnaissance optique des caractères, mais ils sont également utiles pour analyser les mots en tant qu'unités textuelles discrètes. Ils sont aussi bons pour analyser le son. La Figure.1 illustre que les CNN sont bons

pour établir la position et (quelque peu) les caractéristiques invariantes de rotation à partir des données d'image brutes. (N.Malki, 2018/2019)

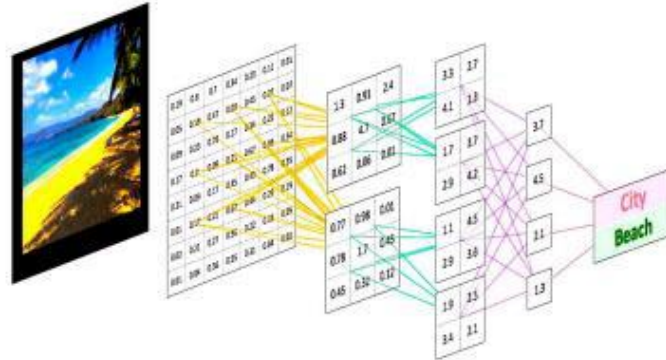


Figure II. 1 RNC et la vision par ordinateur

II-3 Historique :

Le premier réseau de neurones convolutif a été en réalité développé en 1998 par le chercheur français Yann Le Cun. (D.Kadous, 2012) Ce réseau appelé Le Net a permis d'atteindre de très bonne performance en reconnaissance des caractères, Bien que cette approche donne des résultats, ses progrès et son évolution ont été limités par les progrès technologiques en matière de micro-processeurs, de puissance de calculs, et du manque d'accessibilités à des données afin de pouvoir entraîner les neurones. (D.Kadous, 2012)

Pendant certains chercheurs ont continué à travailler sur ce modèle pendant environ deux décennies. . (D.Kadous, 2012) Et, avec l'aide des évolutions en matière de technologies et avec la disponibilité toujours plus grande de données, ont pu améliorer cette technique. (N.Malki, 2018/2019)

Ce n'est qu'en 2012 que le Deep Learning est remis au goût du jour en remportant avec succès le concours de reconnaissance d'image fondé par l'université de Stamford (Large Squale Visual Recognition Challenge : ILSVRC) grâce à l'évolution en matière de technologies et à la disponibilité toujours plus grande de données. [30] Un nouvel algorithme de Deep Learning expose les records Il s'agit d'un réseau de neurones convolutif appelé Alex Net, largement inspiré du réseau LeNet. Image Net regroupant 15 000 000 d'images naturelles comportant différents objets et diverses scènes (véhicules, animaux...). . (D.Kadous, 2012)

Aujourd'hui, les réseaux de neurones convolutifs, sont toujours les modèles les plus performants pour la classification d'images. [30] Google, Microsoft, Facebook, Baidu (le

moteur de recherche chinois), Alibaba (site marchand chinois), Nvidia (géant du processeur graphique) utilisent les CNN dans leurs applications. [30]

II-4 Les couches de réseaux de neurones convolutionnels :

Il y a plusieurs couches différentes dans CNN comme le montre la figure II.1 :

- Couche d'entrée (Input layer).
- Couche de convolution (Convo layer : Convolution + ReLU).
- Couche de Pooling.
- Couche entièrement connectée (Couche Fullyconnected).
- Couche Softmax/logistique.
- Couche de sortie (Output layer).

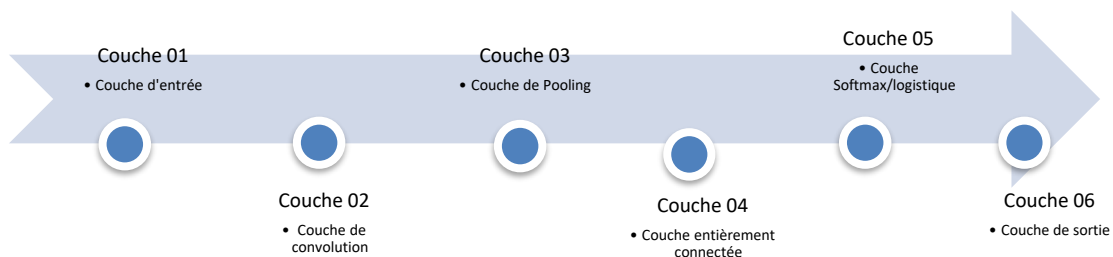


Figure II. 2 Les couches de CNN [31]

II-4-1 Couches d'entrée

Les couches d'entrée sont là où nous chargeons et stockons les données d'entrée brutes de l'image pour le traitement dans le réseau. Ces données d'entrée spécifient la largeur, la hauteur et le nombre de canaux. Typiquement, le nombre de canaux est de trois, pour les valeurs RVB pour chaque pixel.

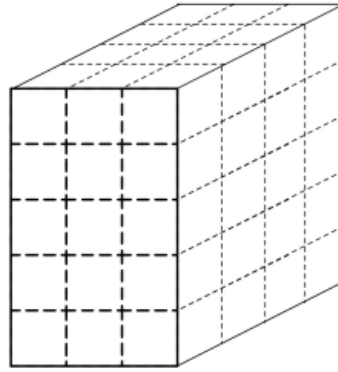


Figure II-3 : Volume 3D de la couche d'entrée [32]

II-4-2 Couches convolutionnelles :

Les couches convolutionnelles sont considérées comme les éléments de base des architectures RNC. Comme la montre la figure II.4. Les couches convolutionnelles transforment les données d'entrée en utilisant un patch de neurones connectés localement de la couche précédente. La couche calculera un produit de point entre la région des neurones dans la couche d'entrée et les poids auxquels ils sont connectés localement dans la couche de sortie.

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. Prenons la Figure II-4 ci-dessous à gauche comme exemple d'image et la Figure à droite comme exemple

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Figure II-4 : Exemple simpliste des valeurs

1	0	1
0	1	0
1	0	1

Figure II-5: Exemple de valeurs d'une des pixels d'une image 5x5 matrices utilisées comme filtre

Dans le cas de la Figure II-4, les valeurs sont binaires. Dans un cas réel, les valeurs devraient varier entre 0 et 255. Dans la Figure II-5, 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. (M.H.Yang & N.Ahuja, 1998)

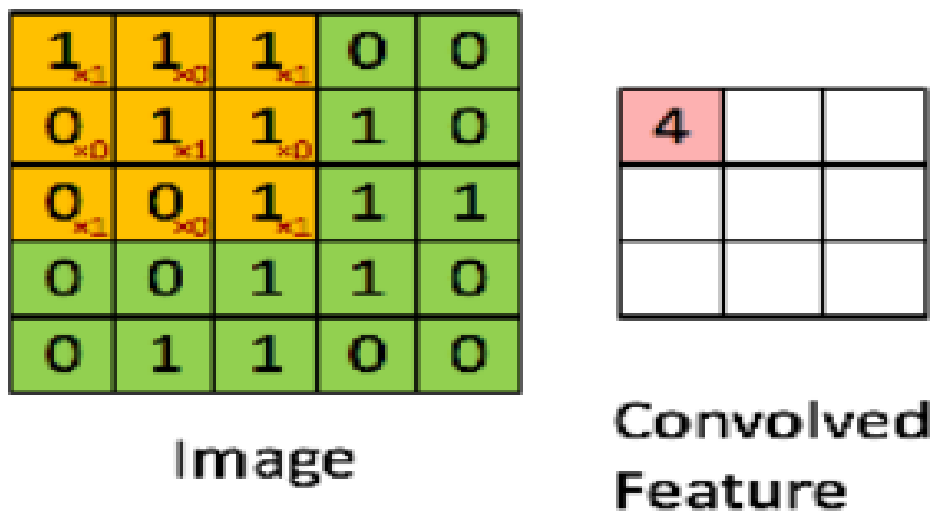


Figure II-6 : Les valeurs devraient varier entre 0 et 255

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. Voir l'animation suivante :

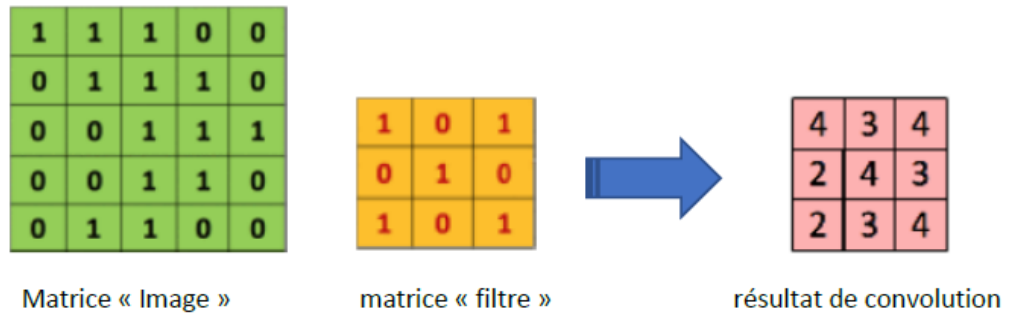


Figure II-7 : Le filtre

II-4-3 Pooling :

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs caractéristiques 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 régulières, puis on garde au sein de chaque cellule la valeur maximale. (**researchgate, 2022**)

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. (**L.AbdelhakBilal, 2018**)

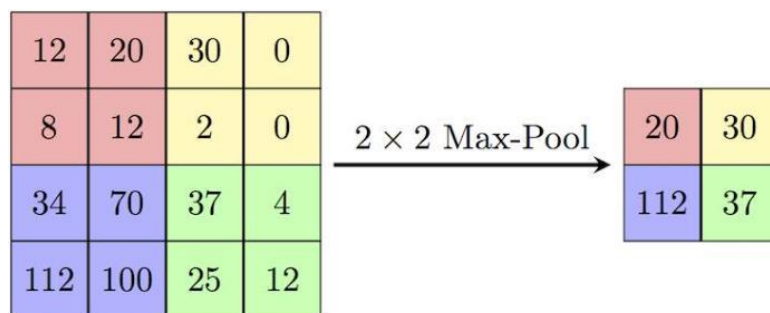


Figure II-8: Pooling

II-4-4 Couch 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 (respectivement, le deuxième) donne la probabilité d'appartenir à la classe "chat" (respectivement "chien"). Ainsi, le vecteur $[0.9 \ 0.1]$ signifie que l'image a 90% de chances de représenter un chat.

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 Fully-connected. (**penseeartificielle, 2022**)

II-5 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.

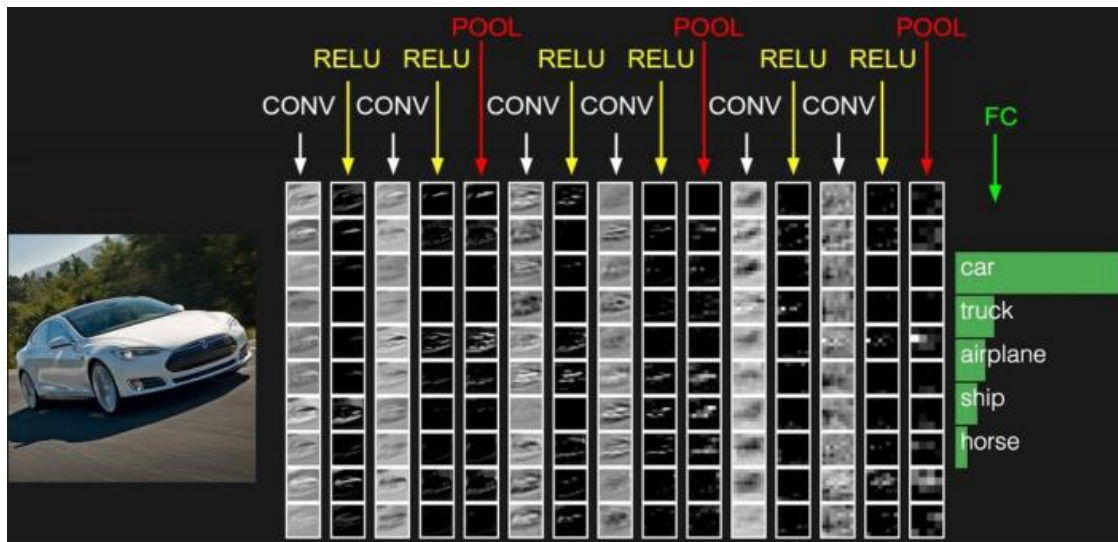


Figure II-9 : Exemple d'un véhicule en utilisant le CNN

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, 8classification, 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 DeepLearning.

La première couche de convolution apprend des futures 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 caractéristiques apprises sont complexes : celles-ci se composent des caractéristiques plus simples des couches précédentes. Un carré est un exemple de caractéristique complexe, formée de contours et de coins.

Les couches de convolution les plus hautes apprennent donc des caractéristiques 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. (Dr.Merzougue, 2020)

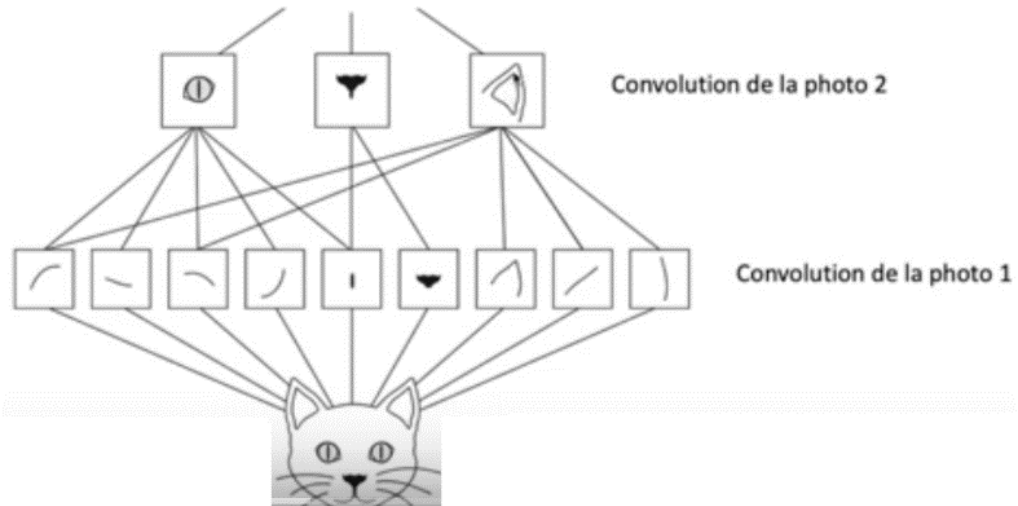


Figure II-10 : un exemple d'une couche convolution

II-6Le 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 hyperparamètres, c'est-à-dire des paramètres dont vous devez préalablement définir la valeur.

Remarque : Les caractéristiques de la couche de convolution et les poids de la couche Fully-connected ne sont pas des hyper paramètres, puisqu'ils sont appris par le réseau de neurones lors de la phase d'entraînement. La taille des caractéristiques maps en sortie des couches de convolution et de Pooling dépend des hyperparamètres.

Chaque image (ou caractéristiques maps) est de dimensions

$$W \times H \times D$$

Où

- W est sa largeur en pixels,
- H sa hauteur en pixels
- 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 :

- ⇒ Le nombre de filtres K
- ⇒ La taille F des filtres : chaque filtre est de dimensions $F \times F \times D$ pixels.
- ⇒ 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.
- ⇒ 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 caractéristiques difficile (**Dr.Merzougue, 2020**)

II-7 Avantages de CNN :

Les réseaux de neurones convolutifs ont une méthodologie similaire à celle des méthodes traditionnelles d'apprentissage supervisé : ils reçoivent des images en entrée, détectent les caractéristiques automatiquement de chacune d'entre elles, puis entraînent un classifieur dessus. Donc les CNN réalisent eux-mêmes tout le boulot fastidieux d'extraction et description de caractéristiques.

Lors de la phase d'entraînement, l'erreur de classification est minimisée afin d'optimiser les paramètres du classifieur et les caractéristiques. De plus, l'architecture spécifique du réseau permet d'extraire des caractéristiques de différentes complexités, des plus simples au plus sophistiquées. L'extraction et la hiérarchisation automatiques des caractéristiques, qui s'adaptent au problème donné, constituent une des forces des réseaux de neurones convolutés. (**researchgate, 2022**)

II-7 Conclusion :

Aujourd'hui, les réseaux de neurones convolutionnels, sont toujours les modèles les plus performants pour la classification d'images. Cette partie leur est donc naturellement consacrée. Dans ce chapitre nous avons décrit les démarches à suivre pour réaliser un CNN. Les différents types de couches d'un CNN sont les suivants :

- ⇒ Couches convolutives : elles passent en entrée à travers certains filtres, qui identifient les caractéristiques dans l'image
- ⇒ Couches de regroupement : elles combinent des caractéristiques convolutives, contribuant à la réduction des caractéristiques
- ⇒ Couches aplaties : elles convertissent une couche N-dimensionnelle en une couche de classification 1D
- ⇒ Couche finale : qui indique le résultat final

Le chapitre suivant sera consacré essentiellement à l'implémentation de ces méthodes et le test de la performance de notre application, pour la reconnaissance des chiffres.

Chapitre03



**Implémentation et résultats
expérimentaux**



III-1 Introduction :

Dans ce chapitre nous allons définir notre architecture de CNN mise en point pour la reconnaissance des chiffres manuscrits sur la base MNIST, via le langage de programmation python. Nous commençons tout d'abord par la présentation des ressources, du langage et de l'environnement de développement que nous avons utilisé. Puis les étapes d'implémentation du système modèle et on termine par les tests effectués.

III-2 l'environnement de développement :

Pour réaliser notre travail, nous avons utilisé un PC portable HP avec les caractéristiques suivantes :

- Processeur : Intel(R) Core (TM) i3-310M CPU.
- Vitesse d'horloge : 2.40 GHz. }
- Capacité Mémoire (RAM) :4.00 Go. }
- Capacité disque dur : 500 Go. }

Pour la réalisation de notre système nous avons choisi le langage de programmation Python, ainsi que les différents bibliothèques (Tensorflow, Keras, Numpy...).

III-2-1 Langage de programmation et bibliothèques utilisées :

III -2-1-1 Python :

Python est un langage de programmation de haut niveau utilisé pour la programmation générale. Créé par Guido van Rossum et sorti en 1991, Python a une philosophie de conception qui met l'accent sur la lisibilité du code, notamment en utilisant des espaces importants. Il combine un pouvoir remarquable avec une syntaxe très claire. Il comporte des modules, des classes, des exceptions, des types de données dynamiques de très haut niveau et le typage dynamique. Il existe des interfaces vers de nombreux appels systèmes et bibliothèques, ainsi que vers différents systèmes de fenêtrage.

Les nouveaux modules intégrés sont faciles à écrire en C ou C ++ (ou dans d'autres langages, selon l'implémentation choisie). Python est également utilisable comme langage d'extension pour les applications écrites dans d'autres langages nécessitant des interfaces de script ou d'automatisation facile à utiliser.

III -2-1-2 PyCharm :

Est un environnement de développement intégré utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django.

C'est un logiciel multi-plateforme qui fonctionne sous Windows, Mac OS X et GNU/Linux.

III -2-2 Bibliothèques utilisées :

Python dispose de plusieurs bibliothèques et chaque bibliothèque a un travail spécifique, ce qui lui rend un langage très riche et très utilisé. Les bibliothèques utilisées dans ce travail sont : Tensorflow, Keras. Numpy

a-Tensorflow :

Est une bibliothèque logicielle open source pour le calcul numérique de haute performance. Son architecture flexible permet un déploiement facile du calcul sur diverses plates-formes (CPUs, GPUs, TPUs), et des ordinateurs de bureau aux clusters de serveurs, aux périphériques mobiles. Initialement développé par des chercheurs et des ingénieurs de l'équipe de Google Brain au sein de l'organisation de l'IA de Google, il s'appuie sur l'apprentissage automatique et l'apprentissage en profondeur. (**tensorflow, 2022**)

Pour l'installation de Tensorflow, nous allons dans le menu démarrer de notre machine Windows, nous recherchons "cmd", et à l'aide du clic droit nous choisissons "exécuter en tant qu'administrateur", une commande est exécutée pour installer Tensorflow.

Voici la commande :

```
C:\Users\micro>pip install Tensorflow
```

b-Keras :

C'est l'outil le plus utilisé en Python dans le monde pour l'apprentissage profond (Deep Learning). Cette librairie open-source, créée par François Chollet, Permet de créer facilement et rapidement des réseaux de neurones, en se basant sur les principaux frameworks (Tensorflow, Pytorch). Keras permet de diminuer de 30% le temps de développement d'un prototype de réseaux de neurones). (**keras, 2022**)

c- Numpy:

Est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion), on peut aussi l'intégrer le code C / C ++ et Fortran.

d-Pilow :

Est une bibliothèque de traitement d'images pour le langage de programmation Python. Elle permet d'ouvrir, de manipuler, et de sauvegarder différents formats de fichiers graphiques.

Une classe du même nom qui est utilisée pour représenter une image PIL. Elle fournit à l'interpréteur python des capacités d'édition d'image. L'image module fournit également un certain nombre de fonctions, y compris des fonctions pour charger des images à partir de fichiers et pour créer de nouvelles images.

III-2-2La base de données MNIST :

La base de données MNIST est une base standard, publique, très utilisée dans le domaine de la reconnaissance des chiffres. Elle est composée de 70000 images en niveaux de gris, extraites d'une base plus large (NIST). 60000 images sont utilisées pour former la base d'apprentissage et les 10000 autres images constituent la base de test. Le nombre d'images dans chaque classe est indiqué dans le tableau 3.1. La taille de chaque image est de 28x28. Le taux de reconnaissance obtenu par un être humain est évalué à 99.8%. La figure 3.2 montre quelques échantillons de cette base.

Le tableau suivant représente la répartition de la base MNIST pour les chiffres :

Tableau 2 : la répartition de la base MNIST pour les chiffres

Classes	Apprentissage	Test	Totale
0	5923	980	6903
1	6742	1135	7877
2	5958	1032	6990
3	6131	1010	7141
4	5842	982	6824
5	5421	892	6313
6	5918	958	6876
7	6265	1028	7293
8	5851	974	6825
9	5949	1009	6958
Totale	60000	10000	70000



Figure III.1 : Quelques échantillons extraits de la base MNIST

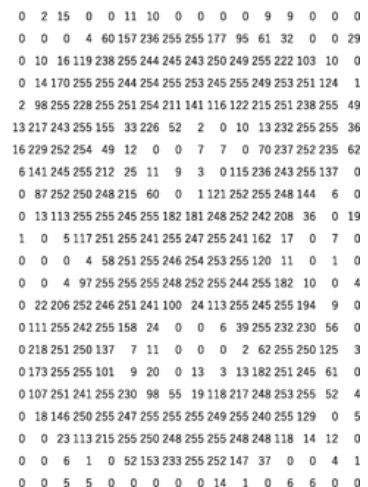
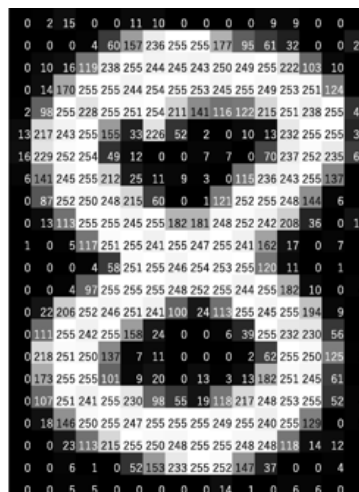


Figure III.2 : exemple de numéro 8 dans la base MNIST

III-3 Implémentation :

III-3-1 Importation des bibliothèques nécessaires :

- La première des choses à faire consiste tout d'abord à configurer l'environnement et installer toutes les bibliothèques puis Importer les bibliothèques installées.

- Ensuite, nous allons importer le type de modèle séquentiel de Keras. Il s'agit simplement d'un empilement linéaire de couches de réseaux de neurones. Nous importerons aussi les couches "principales" de Keras :

- Dense : régulateur de couche, chaque couche reçoit une entrée de chaque neurone de la couche précédente.
- Flatten : ça permet de rendre un modèle sous forme d'une colonne.
- Dropout : elle permet d'éviter le sur apprentissage « overfitting » en désactivant une partie de neurones.

Pour un entraînement plus efficace sur les données d'image nous allons importer les couches CNN de Keras. Ce sont les couches convolutives qui nous aideront à former notre modèle :

- Convolution2D : agit comme un filtrage.
- MaxPooling2D : est utilisé pour réduire les dimensions spatiales du volume de sortie.

```
1 import keras
2 from keras.datasets import mnist
3 from keras.models import Sequential
4 from keras.layers import Dense, Dropout, Flatten
5 from keras.layers import Conv2D, MaxPooling2D
```

Figure III.3 :Importation des bibliothèques

III-3-2 Traitement de la base de données :

La fonction Load_data télécharge les données et les divise en deux groupes, un pour la formation (60 000 images) et un pour les tests (10 000 images).

Lors de la lecture des données, nous utiliserons One Hot Encoding pour représenter les classifications des images. One-Hot Encoding utilise un vecteur de valeurs binaires pour représenter des valeurs numériques ou de classe. Étant donné que nos classes sont destinées à représenter des nombres de 0 à 9, le vecteur contiendra 10 valeurs, une pour chaque nombre possible. L'une de ces valeurs est attribuée en plaçant la valeur 1, afin de représenter le nombre dans cet indicateur du vecteur, et les valeurs restantes se verront attribuer la valeur 0. Par exemple, le nombre 3 sera représenté par le vecteur comme ceci :

[0, 0, 0, 1, 0, 0, 0, 0, 0, 0] et vous remarquerez que la valeur 1 est à l'indice 3, donc le vecteur représentera le nombre 3.

Afin de représenter les images réelles qui sont de 28 par 28 pixels, nous devons les aplatir en vecteur 10 avec une taille de 784 pixels, qui est le produit de 28 par 28. Nous stockerons ces pixels, qui formeront l'image plus tard, dans des valeurs comprises entre 0 et 255, car ces valeurs détermineront le niveau de gris du pixel, et nos images seront affichées en noir et blanc uniquement. Il représentera donc un pixel noir avec une valeur de 255 et un pixel blanc avec une valeur de 0, avec différentes nuances de gris entre les deux.

III-3-3 Construction du modèle CNN :

La figure 3.4 illustre une partie du code qui représente notre modèle de réseau de neurones convolutif pour reconnaître les chiffres manuscrits.

- Batch size : 128
- Nombre de classes = 10 (0-9)
- Nombre d'itérations : 300itérations.

```
34 model = Sequential()
35 model.add(Conv2D(32, kernel_size=(5, 5), activation='relu', input_shape=input_shape))
36 model.add(MaxPooling2D(pool_size=(2, 2)))
37 model.add(Conv2D(64, (3, 3), activation='relu'))
38 model.add(MaxPooling2D(pool_size=(2, 2)))
39 model.add(Flatten())
40 model.add(Dense(128, activation='relu'))
41 model.add(Dropout(0.3))
42 model.add(Dense(64, activation='relu'))
43 model.add(Dropout(0.5))
44 model.add(Dense(num_classes, activation='softmax'))
```

Figure III.4 :le modèle CNN utilisée

III-3-4 Compilation et ajustement Évaluation du modèle :

Avec le code ci-dessus, nous avons créé un CNN vide non optimisé. Il est maintenant temps de définir un optimiseur et une fonction de perte « Loss » pour ajuster le modèle. Puis il faut l'évaluer avec `x_test` et `y_test` en utilisant les instructions suivantes :

- Optimiser : permet de réduire le poids des erreurs
- Loss : désigne le taux d'erreur.
- Accuracy : désigne le taux de précision.

III-4 : Résultats expérimentaux :

Dans cette partie nous nous intéressons aux résultats d'exécution de l'approche développée.

- 1- On voit que la base MNIST est déjà téléchargée à partir de Keras via ce folder 'https://storage.googleapis.com/Tensorflow/tf-keras-datasets/'

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [=====] - 45s 4us/step
(60000, 28, 28) (60000,)
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
```

Figure III.5: téléchargement de la base MNIST

- 2- Pour la visualisation de notre modèle CNN on utilise l'instruction suivante :

Model.summary()

Donc on obtient :

```
Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
conv2d (Conv2D)             (None, 24, 24, 32)       832
max_pooling2d (MaxPooling2D) (None, 12, 12, 32)       0
conv2d_1 (Conv2D)           (None, 10, 10, 64)       18496
max_pooling2d_1 (MaxPooling2D) (None, 5, 5, 64)        0
flatten (Flatten)           (None, 1600)              0
dense (Dense)                (None, 128)               204928
dropout (Dropout)           (None, 128)               0
dense_1 (Dense)              (None, 64)                8256
dropout_1 (Dropout)         (None, 64)                0
dense_2 (Dense)              (None, 10)                650
-----
Total params: 233,162
Trainable params: 233,162
Non-trainable params: 0
-----
```

Figure III.6 : Architecture du notre modèle CNN

La figure ci-dessus nous présente les différentes couches de notre architecture de CNN ainsi que leurs paramètres, nous avons un total de paramètres qui vaut 233162. Le calcul des paramètres s'effectue comme suit :

Pour les couches de convolutions : le nombre de paramètres est égal à la taille du filtre multiplié par le nombre de filtres plus le biais.

Pour les couches denses : le nombre de paramètres est égal aux entrées multiplié par le nombre de neurones dans cette couche plus les biais.

3- Après l'exécution de notre code on obtient :

```
Epoch 1/300
469/469 [=====] - 54s 114ms/step - loss: 2.2984 - accuracy: 0.1107 - val_loss: 2.2849 - val_accuracy: 0.1802
Epoch 2/300
469/469 [=====] - 53s 112ms/step - loss: 2.2854 - accuracy: 0.1337 - val_loss: 2.2697 - val_accuracy: 0.2691
Epoch 3/300
469/469 [=====] - 51s 110ms/step - loss: 2.2720 - accuracy: 0.1573 - val_loss: 2.2535 - val_accuracy: 0.3618
Epoch 4/300
469/469 [=====] - 52s 110ms/step - loss: 2.2583 - accuracy: 0.1810 - val_loss: 2.2360 - val_accuracy: 0.4311
Epoch 5/300
469/469 [=====] - 52s 112ms/step - loss: 2.2416 - accuracy: 0.2095 - val_loss: 2.2170 - val_accuracy: 0.4801
Epoch 292/300
469/469 [=====] - 52s 111ms/step - loss: 0.2420 - accuracy: 0.9314 - val_loss: 0.1025 - val_accuracy: 0.9698
Epoch 293/300
469/469 [=====] - 52s 111ms/step - loss: 0.2436 - accuracy: 0.9309 - val_loss: 0.1023 - val_accuracy: 0.9700
Epoch 294/300
469/469 [=====] - 52s 111ms/step - loss: 0.2431 - accuracy: 0.9309 - val_loss: 0.1020 - val_accuracy: 0.9701
Epoch 295/300
469/469 [=====] - 52s 111ms/step - loss: 0.2387 - accuracy: 0.9321 - val_loss: 0.1014 - val_accuracy: 0.9700
Epoch 296/300
469/469 [=====] - 52s 111ms/step - loss: 0.2419 - accuracy: 0.9309 - val_loss: 0.1013 - val_accuracy: 0.9700
Epoch 297/300
469/469 [=====] - 52s 111ms/step - loss: 0.2373 - accuracy: 0.9326 - val_loss: 0.1012 - val_accuracy: 0.9699
Epoch 298/300
469/469 [=====] - 52s 110ms/step - loss: 0.2371 - accuracy: 0.9324 - val_loss: 0.1008 - val_accuracy: 0.9701
Epoch 299/300
469/469 [=====] - 53s 113ms/step - loss: 0.2420 - accuracy: 0.9320 - val_loss: 0.1007 - val_accuracy: 0.9704
Epoch 300/300
469/469 [=====] - 53s 113ms/step - loss: 0.2367 - accuracy: 0.9333 - val_loss: 0.1003 - val_accuracy: 0.9703
The model has successfully trained
```

Figure III.7: Entraînement du notre modèle

On voit que le Test accuracy :0.9703

Alors notre modèle a été entraîné avec succès avec un taux de reconnaissance qui vaut 97.03%

4- On illustre dans ce qui suit les résultats en termes de taux de reconnaissance :

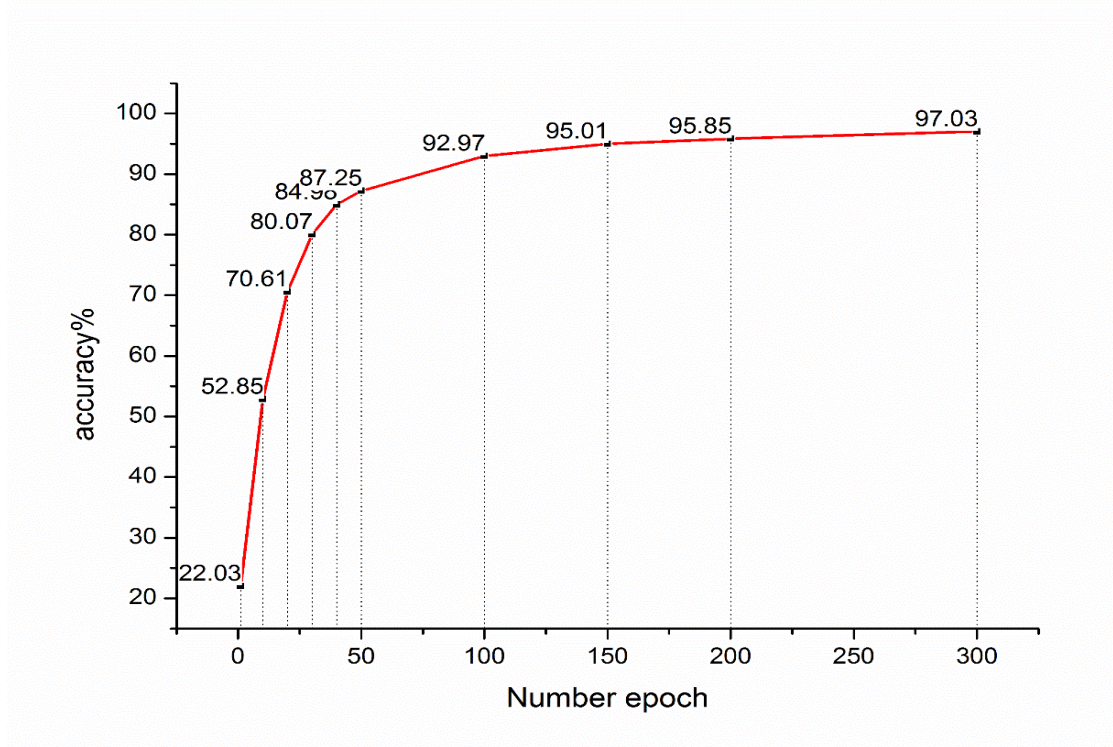


Figure III.8 Représentation de la précision en fonction des itérations

Après l'analyse des résultats obtenus, Nous avons constaté les remarques suivantes :

- D'après les figures précédents La précision de l'apprentissage et de test augmente avec le nombre d'itération, ceci reflète que à chaque itération le modèle apprend plus d'information
- 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'itération.
- De même, l'erreur d'apprentissage et de la validation diminue avec le nombre d'itération.

Tableau3 : Tableau de comparaison

Itération	Taux de reconnaissance
50	87.55%
100	92.96%
300	97.03%

Si nous faisons une comparaison entre les itérations 50 ,100 et 300 nous constatons que la précision augmente cela signifie que notre modèle a été bien entraîné et réponds d'ailleurs à la définition du réseau de neurones qui disait que plus le réseau de neurones est profond meilleur sont ses performances.

5- Pour vérifier l'efficacité du modèle développé dans ce projet, nous allons utiliser des chiffres écrits manuellement et nous voyons s'il peut les identifier (figure III.10).

Le programme arrive à reconnaître correctement les chiffres cerclés (2, 8, 0) avec différentes précisions (0.61, 0.62 0.70) respectivement.

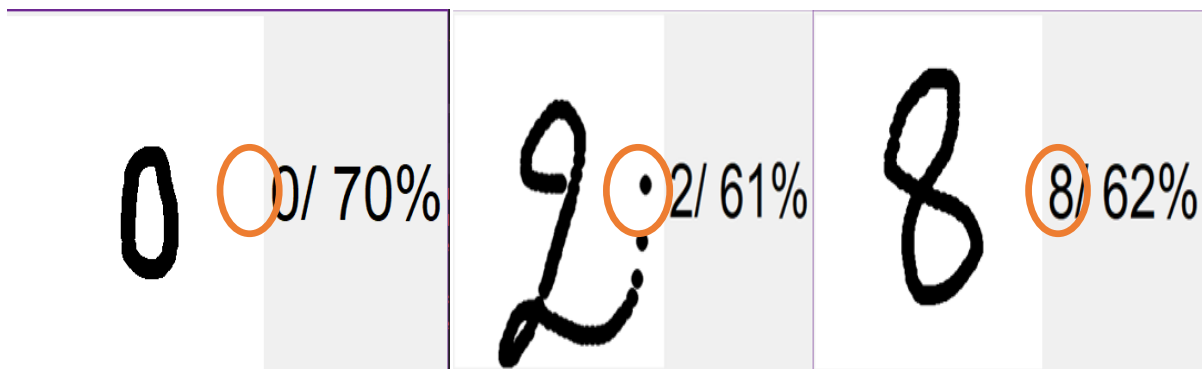


Figure III.10:Exemple de chiffres écrites sur écran.

III-5 Comparaison aux autres travaux :

Après la comparaison avec d'autres travaux tel que le mémoire intitulé "reconnaissance automatique des chiffres avec le Deep Learning" par Mr SLIMANI et Mr KGALED ADEL en 2020. (M.Slimani & K.ADEL, 2020) Nous remarquons que leur modèle a donné un taux de reconnaissance de 99.09 cela est dû à la puissance du matériel utilisée.

III-6 conclusion :

Nous avons présenté dans ce chapitre l'implémentation de l'approche de reconnaissance des chiffres basée sur les réseaux de neurones convolutifs, nous avons détaillé les étapes d'implémentation du système modèle ainsi que tous les tests effectués.

La comparaison des résultats trouvés a montré que le nombre d'itérations, la taille de la base et la profondeur de réseaux et aussi la configuration du matériel utilisé, sont des facteurs importants pour l'obtention de meilleurs résultats.

Conclusion Générale

Conclusion générale

Jusqu' au jour hui, La reconnaissance de l'écriture et la reconnaissance de chiffres manuscrits en particulier présente un défi très grand, malgré les efforts et les travaux intensifs réalisés dans ce domaine, aucun système de reconnaissance de chiffres manuscrits n'est jugé fiable à 100%, Mais au fur et à mesure les autres essayent d'améliorer les scores pour de meilleurs résultats. Et elle joue un rôle très important dans le monde actuel. Elle est capable de résoudre des problèmes complexes et rendre les activités de l'homme plus simple. L'objectif de ce travail est la reconnaissance des chiffres manuscrits avec classifieur CNN

Notre modèle a été entraîné avec succès avec un taux de précision qui vaut 97.03% en constatant que le nombre d'itérations, la taille de la base et la profondeur de réseaux et aussi la configuration du matériel utilisé, sont des facteurs importants pour l'obtention de meilleurs résultats.

Nous pouvons envisager plusieurs perspectives possibles pour en durer le travail :

1. L'évaluation sur d'autres bases de donnée ses ainsi que sur d'autres types de noyaux.
2. Les méthodes d'extraction des caractéristiques peuvent être envisagées (par exemple en utilisant les caractéristiques qui contiennent les paramètres structurels extraits du squelette de la forme).

Bibliographie

Bibliographie

1. A.Boucher, & N.A.Tuan. *Reconnaissance d'écriture manuscrite*.
2. A.Boucher, & N.A.Tuan. (2005). *Reconnaissance d'écriture manuscrite*.
3. A.Ghali. (2015). *Amélioration de la reconnaissance par le visage*.
4. A.Lemieux. (2003). *Systèmes d'identification de personnes par vision numérique*. (U. Lava, Éd.) Québec.
5. Benahmed, N. (2002). *Optimisation des réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés : Sélection et pondération des primitives par algorithmes génétiques*. MONTRÉAL, École de technologie supérieur: Thèse de Master.
6. C.Aravinda, D.Cors, R.Tibo, R.Gime, & T.Fato. (2014). Kannada handwritten character recognition using multi caractéristique extraction techniques. *IJSR* .
7. C.D.Aravinda, T.Nos, & R.david. (2014). Kannada handwritten character recognition using multi caractéristique extraction techniques. *International Journal of Science and Research (IJSR)* .
8. Chois, W., James, R., T.Rini, Timed, Z., & Fari, Z. (2003). Extraction of the euclidean skeleton based on a connectivity criterion. *Pattern Recognition* , 36, 721-729.
9. D.Abdelhakim. (2011). *La reconnaissance des chiffres manuscrits par les machines à vecteurs de support(SVMs)*. Université de Tébessa , algerie: Thèse de Master.
10. D.DJEZZAR, M.Rafik, & F.BENKADA. (2022). *Mise au Point d'une Application de Reconnaissance de Formes*. algerin: Mémoire Master.
11. D.Kadous. (2012). *utilisation des réseaux de neurones comme outil du datamining : génération de modèle comportemental d'un processus physique à partir de données automatique de leur plaque d'immatriculation*. (u. M. khider, Éd.) Biskra: master académique.
12. Dr.Merzougue. (2020). *deeplearning*. (c. m. sds, Éd.) universite batna 02.
13. F.Grandidier. (24 janvier 2003). *un nouvel algorithme de sélection de caractéristiques – application à la lecture automatique de l'écriture manuscrite*. (d. e. ph.d, Éd.) montréal.
14. G.Abdeldjalil. (2011). *Segmentation automatique pour la reconnaissance numérique des chèques bancaires Algériens* . Khanchela, centre Universitaire de Khanchela, algerié: Thèse de MAGISTER.
15. H.Nemmour, & R.Diboune. (2007). *Utilisation des machines à vecteur de support (SVM) pour la reconnaissance des chiffres manuscrits*. Université des Sciences et de la Technologie Houari Boumediene: Mémoire d'ingénieur d'état.
16. J.Pierre. (1999). *reconnaissance de l'écriture manuscrite*. Département Images, ENSTParis et Guy LORETTE RISA: université de Rennes 1.
17. K, K., Y, r., T, t., & j, j. (2002). Document image binarization based on topographic analysis using a water flow model. *Pattern Recognition* , 35, 265-277.
18. Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing* , 10, 215-236.
19. keras. (2022, 05 01). Récupéré sur <https://keras.com>

20. L.AbdelhakBilal. (2018). *deeplearning basé sur les méthodes de réduction pour la reconnaissance de visage*. Biskra, université Mohamed khider: master académique.
21. M.Bahashwan, T.Ahmed, & Y.Karime. (2015). Efficient segmentation of Arabic handwritten characters using structural caractéristiques. *IAJIT*.
22. M.Bahashwan, Y.Tindoras, & Z.figo. (2015). Efficient segmentation of Arabic handwritten characters using structural caractéristiques. *IAJIT*.
23. M.Cheriet, T, R., Z, F., & I, R. (2007). *Character recognition systems. A guide for students and practioners*. (J. Wiley, Éd.) Inc.
24. M.E, M., & P, P. (2002). Visual Input for Pen-Based Computers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 3.
25. M.H.Yang, & N.Ahuja. (1998). Detecting Human Faces in Color Images». Proc. Int'lConf. Image Processing. *IEEE*, 01, 127-130.
26. M.Slimani, & K.ADEL. (2020). *Reconnaissance automatique des chiffres avec le Deep Learning*.
27. N, A., D.J, K., H, Y., & M, Y. (2002). Detecting Faces in Images: A Survey. *IEEE Transactions On Pattern Analysis And Machine Intelligence PAMI*, 24, 34-58.
28. N.Benahmed. (2002). *Optimisation des réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés : Sélection et pondération des primitives par algorithmes génétiques*. MONTRÉAL, École de technologie supérieure: Thèse de Master.
29. N.Malki. (2018/2019). *classification automatique des textes par les réseaux de neurones à convolution*. (u. L. m'hidio.e.b, Éd.) mathématique et informatique: thèse doctora.
30. Nemmour, H., & R.Diboune. (2007). *Utilisation des machines à vecteur de support (SVM) pour la reconnaissance des chiffres manuscrits*. (U. d. Boumediene, Éd.) Mémoire d'ingénieur d'état.
31. O.D, T., & T, T. (1995). Evaluation of binarization methods for document images On Pattern Analysis and Machine Intelligence. *IEEE*, 11, 312-314.
32. *penseeartificielle*. (2022, 05 02). Récupéré sur https://penseeartificielle.fr/focus-reseau-neuronesconvolutifs/#4_le_flattening_ou_mise_a_plat
33. R, K., & L.O, G. (1995). Document image analysis. *IEEE Computer Society Press*, 05 (01), 12.
34. R.Firdousi, & S.Parveen. (2014). Local thresholding techniques in image binarization. *International Journal of Engineering and Computer Science*, 03, 4062- 4065.
35. R.Gaur, T.Mino, & D, T. (2017). A survey on caractéristique extraction techniques for handwrittencharacter recognition. *International Journal on Recent and Innovation Trends in Computing and Communication*, 05.
36. *researchgate*. (2022, 05 12). Récupéré sur https://www.researchgate.net/figure/typical-architecture-of-cnn-patterson-gibson-2017_fig3_356159364
37. S, G., K, G., W, R., & D.S, N. (1995). *Manufacturing Caractéristique Instances: Which Ones to Recognize*. Symposium on Solid Modeling and Applications.
38. S. Theodoridis, K. Koutroumbas. (2003). *Pattern Recognition, Second Edition*. Elsevier: Academic Press.

39. S.F.BettaharAbdessettar. (2014). *Extraction des caractéristiques pour l'analyse biométrique d'un visage*.
40. S.Watanabe. (1985). *Pattern recognition: Human and Mechanical*. New York Wiley.
41. T.K, G. (2014). Document image binarization techniques: A review. *International Journal of Computer Applications* , 98 (12), 0975-8887.
42. *tensorflow*. (2022, 03 01). Récupéré sur <https://www.tensorflow.org/>
43. Y.Zhang, W.U, L., & A, M. (2011). fast document image denoising method based on packed binary format and source word accumulation. *Convergence Information Technology* , 06 (02), 131-137.