



Université mohammed V
Ecole nationale supérieur d'informatique et d'analyse des systèmes

Handwritten Digit Recognition

Réalisé par :
TOUJI Oumayma
ZRHARI Zineb

Encadré par :
Mme BENBRAHIM Houda

e-Management And Business Intelligence Students

Année universitaire:
2020-2021

Remerciements

Nous tenons tout d'abord à remercier Dieu Le Tout Puissant et Miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail. Au terme de ce travail, il est agréable d'adresser quelques expressions de remerciements à toute personne, dont l'intervention au cours de ce projet a favorisé son aboutissement. Nos remerciements vont en particulier à notre encadrant Mme. BENBRAHIM Houda ainsi que Mme. BENHIBA Lamia qui nous ont fait profiter de leurs savoir ainsi que leurs orientations pertinentes par leurs accompagnement très sérieux. Finalement, nous exprimons notre profonde gratitude et nos chaleureux remerciements à l'ensemble du corps professoral et administratif de l'école pour l'effort qu'ils fournissent afin de nous garantir une bonne formation.

Sincèrement, Oumayma Touji et Zghari Zineb.

Abstract

This report summarizes the work done as part of our second-year Artificial Intelligence Project throughout our studies period of this year. The main objective of this work is to develop a handwritten digit recognizer using Feed forward neural network as a mean. This work consists of several phases including the analysis and conception phase, the development phase while diving deep into each one of them especially the analysis phase. In the first and second chapter of this report, we have briefly described the AI world more specifically the world of computer vision and image processing as well as neural networks. From there the report will go into how the project was built from the aspect of the model it self to an integration to a GUI application.

Contents

1	Introduction	6
2	Chapitre 1 : Concepts de l'intelligence artificielle	1
2.1	Définition de l'intelligence Artificielle	1
2.2	Historique de l'intelligence artificielle	2
2.3	Apprentissage Automatique	3
2.4	Apprentissage non supervisé	4
2.5	Apprentissage Supervisé	4
2.6	Les réseaux de neurones	5
3	Chapitre 2 : le traitement de l'image	1
3.1	Optical Character Recognition (O.C.R)	1
3.2	Reconnaissance des chiffres	2
3.2.1	Document manuscrit	2
3.2.2	La nature de l'écriture	3
3.2.3	Réduction de la complexité d'écriture	4
3.3	Méthodes de reconnaissance	4
3.3.1	Méthode structurelle	4
3.3.2	Méthode de texte	4
3.3.3	Méthode basée sur le prototype	5
4	Chapitre 3 : Analyse et Conception	1
4.1	Spécification du projet	1
4.2	Elaboration des concepts du Projet	1
4.3	Modèle Conceptuel Global	3
4.4	Les étapes de modèle conçu	3
4.4.1	Préprocessing de l'image	5
4.4.2	Feature Extraction	7
4.4.3	Architecture de notre NN	9
4.4.4	Résultats des expériences	10
5	Chapitre 4 : Réalisation de l'application	1
5.1	Les outils et technologies utilisés	1
5.2	Les différents résultats de l'interface	3
6	Conclusion générale	7
7	Bibliographie	8

List of Figures

1	Some definitions of AI organised into four categories	1
2	Apprentissage non supervisé	4
3	Apprentissage supervisé	5
4	shéma de question réponse en terme de raisonnement	6
5	Shéma d'input et output au niveau d'un process	6
6	Exemple d'une opération simple de calcul	6
7	Structure d'un neurone biologique	7
8	Structure d'un neurone artificiel	8
9	Réseau de neurones feed forward	9
10	Différence entre Réseau de neurones feed forward et RNN . . .	10
11	Les différents types de fonctions d'activation	11
12	le scanner de document/OCR	1
13	La dichotomie de la reconnaissance des caractères	2
14	La base de données de MNIST	3
15	The nine screen regions	4
16	Echantillon de Mnist Dataset des écritures humaines	1
17	Logo de Visual Paradigm	2
18	Shéma montrant le modèle conçu	2
19	Processus conçu de modèle global	3
20	Manières différentes d'écriture humaine	4
21	Confusion de la machine des écritures différentes humaines . . .	4
22	le thinning line de l'écriture	4
23	visualisation des données de Mnist	5
24	Before thinning the line	5
25	After thinning the line	6
26	Découpage des bords de l'image	6
27	Découpage de l'image en 16 blocs	7
28	Fit the curve line to a straight one using linear regression	7
29	les 3 features extraites	8
30	processus suivi de la construction du modèle	8
31	la précision de modèle en fonction de sa classification technique .	9
32	SGD Optimizer	10
33	Adam Optimizer	10
34	Adam Optimizer sans dropout	11
35	Adam Optimizer with 1 hidden Layer	11
36	Adam Optimizer with 2 hidden layers	12
37	Adam Optimizer with 3 hidden Layers	12

38	précision finale du modèle	13
39	architecture finale du modèle	13
40	Logo de OpenCv	1
41	Logo de PIL	1
42	Logo de Tkinter	2
43	Logo de Pytorch	2
44	Logo de Sklearn	3
45	Le chiffre 1	3
46	Le chiffre 2	4
47	Le chiffre 3	4
48	Le chiffre 4	4
49	Le chiffre 5	5
50	Le chiffre 6	5
51	Le chiffre 7	5
52	Le chiffre 8	6
53	Le chiffre 9	6
54	Tous les chiffres	6

1 Introduction

La reconnaissance de l'écriture manuscrite est un domaine scientifique dans lequel il existe de nombreux systèmes ayant déjà fait leurs preuves. De manière générale ces systèmes transforment, à l'aide d'algorithmes de traitement, une image en une séquence de vecteurs de données. Cette séquence de vecteurs de données peut ensuite être traitée par une méthode de classification. Dans ce sillage, on pourra se demander à quel point notre réseau de neuronne qu'on va construire va reconnaître les images écrites à la main qu'on lui donne. Afin de répondre à notre problématique, ce projet peut être synthétisé en évoquant la situation en début et en fin de rapport présent comme ce qui suit : Le premier chapitre est dédié à une présentation générale de l'intelligence artificielle, ses définitions, ses différents méthodes pour la résolution des problèmes, etc. Par la suite, une partie de traitement de l'image dans le domaine de l'intelligence artificielle sera traité en décrivant bien évidemment la reconnaissance des caractères. Ensuite, on va mettre en évidence la conception de notre projet ainsi que les différentes étapes suivies pour la construction du modèle de reconnaissance final. Finalement, le dernier chapitre sera dédié à l'implémentation finale de l'application.

Chapitre 1

Concepts de l'intelligence artificielle

L'Homme cherche toujours à comprendre et à reproduire les mécanismes naturels qui l'entourent. Un des domaines les plus passionnants reste celui de l'étude du cerveau humain. Qu'il soit humain ou animal, nous restons fascinés par sa capacité à analyser, à comprendre et à généraliser les problèmes que posent ou l'environnement. Dans le but de se rapprocher du fonctionnement du cerveau, on parlera dans ce chapitre sur généralités sur l'intelligence artificielle.

2 Chapitre 1 : Concepts de l'intelligence artificielle

L'Homme a toujours cherché à comprendre et à reproduire les mécanismes naturels qui l'entourent. Un des domaines les plus passionnants reste celui de l'étude du cerveau. Qu'il soit humain ou animal, nous restons fascinés par sa capacité à analyser, à comprendre et à généraliser les problèmes que posent ou *proposent* l'environnement. Dans le but de se rapprocher du fonctionnement du cerveau, on parlera sur généralités sur l'intelligence artificielle.

2.1 Définition de l'intelligence Artificielle

L'intelligence Artificielle (IA) est de former des systèmes capables de reproduire le comportement de l'humain dans ses activités de raisonnement. J'ai également trouvé des définitions pour ce terme mais ces définitions s'accordent sur le fait que l'objectif de l'IA est de créer des systèmes intelligents, mais elles diffèrent significativement dans leur façon de définir l'intelligence. On peut donc voir ces définitions de l'intelligence artificielle :

Systems that think like humans	Systems that think rationally
"The exciting new effort to make computers think ... <i>machines with minds</i> , in the full and literal sense." (Haugeland, 1985)	"The study of mental faculties through the use of computational models." (Chamiak and McDermott, 1985)
"[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ... " (Bellman, 1978)	"The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Systems that act like humans	Systems that act rationally
"The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990)	"Computational Intelligence is the study of the design of intelligent agents." (Poole <i>et al.</i> , 1998)
"The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	"AI ... is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

Figure 1: Some definitions of AI organised into four categories

- **Créer des systèmes qui se comportent comme les êtres humains :** Cette définition opérationnelle de l'IA fut promue par Alan Turing, qui introduisit "test de Turing" selon lequel une machine est considérée comme intelligente si elle peut converser de telle manière que les interrogateurs (humains) ne peuvent la distinguer d'un être humain.
- **Créer des systèmes qui pensent comme des êtres humains :** Cela implique que l'IA est une science expérimentale, car il faut comprendre au préalable la façon dont pensent les humains, et ensuite évaluer les systèmes par rapport à leurs Similarités avec le raisonnement humain.
- **Créer des systèmes qui pensent rationnellement :** Selon cette définition, les systèmes doivent raisonner d'une manière rationnelle, c'est-à-dire en suivant les lois de la logique.
- **Créer des systèmes qui possèdent des comportements rationnels :** Cette dernière définition de l'IA concerne le développement des agents qui agissent pour mieux satisfaire leurs objectifs .

2.2 *Historique de l'intelligence artificielle*

Les premiers jalons historiques de l'intelligence artificielle datent de la Protohistoire, où mythes, légendes et rumeurs dotent des êtres artificiels, réalisés par des maîtres-artisans, d'une intelligence ou d'une conscience.

La recherche en intelligence artificielle a vraiment commencé après une conférence tenue sur le campus de Dartmouth College pendant l'été 1956. À la suite de cette réunion, certains participants se sont investis dans une recherche sur l'intelligence artificielle .

Certains idéalistes avaient prédit qu'une machine aussi intelligente qu'un être humain existerait en moins d'une génération et des millions de dollars ont alors été investis pour l'organisme cette prédiction. Avec le temps, il est apparu que les difficultés inhérentes à cette annonce avaient été grossièrement sousestimées.

Au début de 1973 jusqu'à la fin des années 1980 : Nous pouvons dire que ce cycle intermittent, les périodes de gel et de dégel en alternance, il dispose

d'un support pour l'intelligence artificielle. Mais il y a encore idéal pour faire des prédictions audacieuses.

1980 jusqu'à présent : L'intelligence artificielle est devenue au fil du temps une matière scientifique de plus en plus rigoureuse et formelle. La plupart des approches étudiées aujourd'hui sont basées sur des théories mathématiques ou des études expérimentales plutôt que sur l'intuition, et sont appliquées plus souvent aux problèmes issus du monde réel .

2.3 Apprentissage Automatique

D'une façon générale, l'apprentissage humain est un processus adaptatif grâce auquel l'individu fournit des réponses adéquates à certaines situations. En Psychologie ou en Science Cognitives, le terme « Apprentissage » désigne le processus d'augmentation de l'efficacité de l'activité mentale ou comportementale sous l'effet de l'expérience.

L'apprentissage automatique (Mitchell, 1997) est un Domain de l'intelligence Artificielle (IA) dont l'objectif est d'étudier les moyens permettant à une machine d'apprendre .Apprendre, dans ce contexte, signifie être capable d'adapter son comportement en présence de situation inconnues (n'ayant pas été prévues par les concepteurs de la machine) et pouvoir extraire des lois à partir de bases de données d'exemples. L'apprentissage se fait donc par des outils qui permettent d'acquérir, élargir et améliorer les connaissances disponibles au système. Il consiste à utiliser des ordinateurs pour optimiser un modèle de traitement de l'information selon certains critères de performance à partir d'observations. Les techniques d'apprentissage automatique sont ainsi utilisées par exemple pour la reconnaissance de forme (écriture, parole, vision), la fouille de données (extraction de connaissance), la mise en place d'outils d'aide à la décision, etc. Il existe 02 familles d'apprentissage : Apprentissage supervisé et Apprentissage non supervisé.

2.4 Apprentissage non supervisé

Dans le cadre de l'apprentissage non supervisé, la tâche consistera en la découverte de similarités entre les observations dans une collection d'exemples, dans le but de regrouper celles-ci en sous-ensembles, appelés clusters ou classes.

Une technique employée consiste à implémenter des algorithmes pour rapprocher les exemples les plus similaires et éloigner ceux qui ont le moins de caractéristiques communes. Parmi les algorithmes d'apprentissage non supervisé, les techniques utilisées classiquement dans le monde des statistiques :

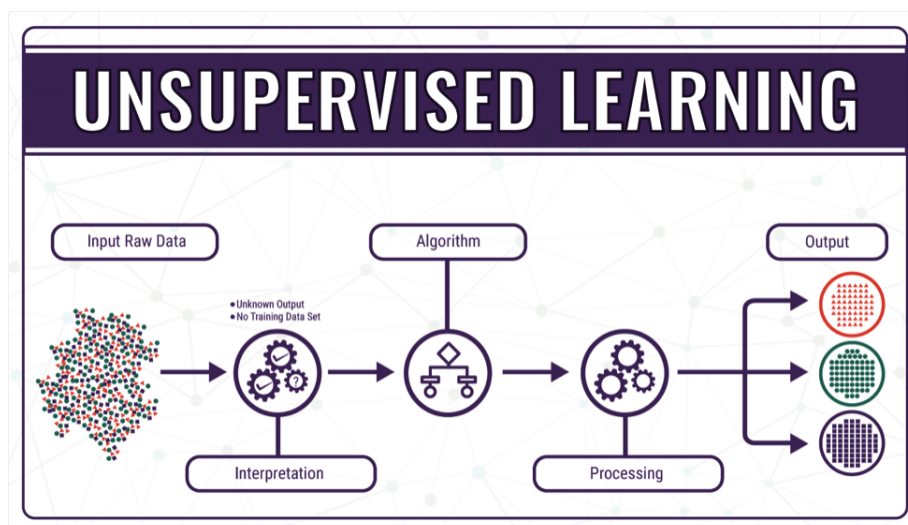


Figure 2: Apprentissage non supervisé

2.5 Apprentissage Supervisé

Dans le cadre de l'apprentissage supervisé, les observations sont accompagnées d'une information complémentaire relative à leur appartenance ou non au concept. Le but d'un algorithme d'apprentissage supervisé est de correctement classer les nouveaux exemples dans les classes définies dans la phase d'apprentissage. La classification supervisée est généralement réalisée en deux phases :

- **Phase d'apprentissage** : cette phase est consacrée à l'apprentissage des règles sur la base d'un échantillon dit d'apprentissage, défini au départ et dont le classement est connu
- **Phase de test** : Dans cette deuxième phase, un deuxième échantillon indépendant, dit de validation ou de test, est formé afin d'étudier la fiabilité des règles pour les comparer, les appliquer et évaluer la complexité du modèle par la vérification des cas de sous apprentissage ou de sur-apprentissage.

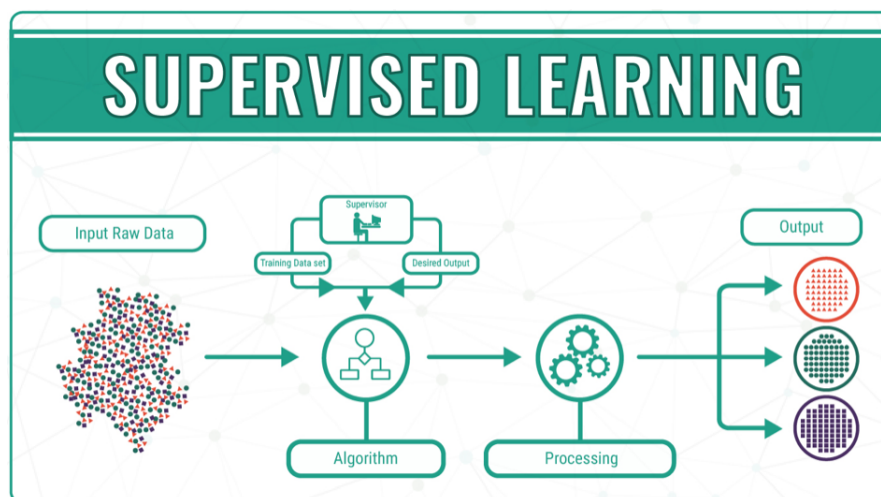


Figure 3: Apprentissage supervisé

D'une part, on parle d'un cas de sous apprentissage lorsque le modèle construit, est trop simple c.-à-d. non suffisamment représentatif des cas d'apprentissage. D'autre part, on parle de cas de sur-apprentissage, assimilé à un apprentissage « par cœur », lorsque le modèle est trop précis et risque de ne pas être généralisable.

2.6 Les réseaux de neurones

Les machines ne sont que des calculatrices au fond, ils sont très rapide en terme de calcul et d'arithmétique. On leurs pose donc une question, ils font un traitement et donc il donne un résultat calculé par la suite.

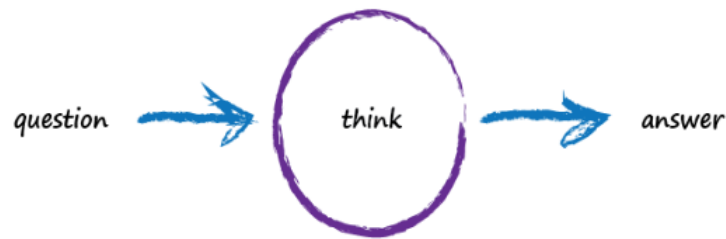


Figure 4: schéma de question réponse en terme de raisonnement

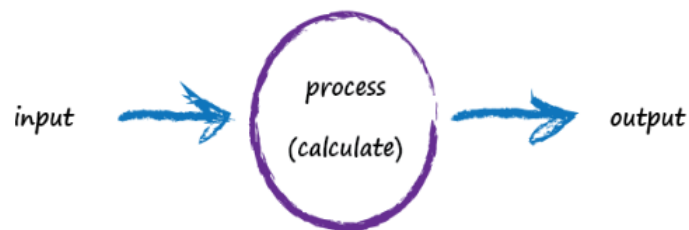


Figure 5: Schéma d'input et output au niveau d'un process

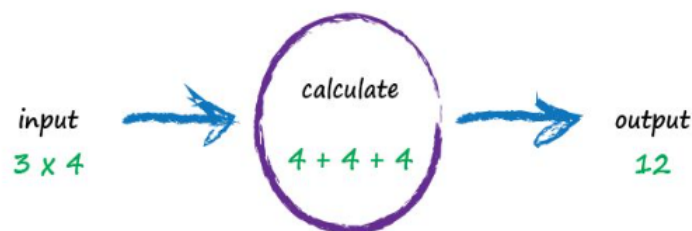


Figure 6: Exemple d'une opération simple de calcul

En réalité, les réseaux de neurones sont une modélisation mathématique du fonctionnement du cerveau humain. Le principe consiste à la construction d'un modèle simplifié du neurone biologique communément appelé neurone formel. Les réseaux de neurones, étant une connexion de plusieurs neurones formels, peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes.

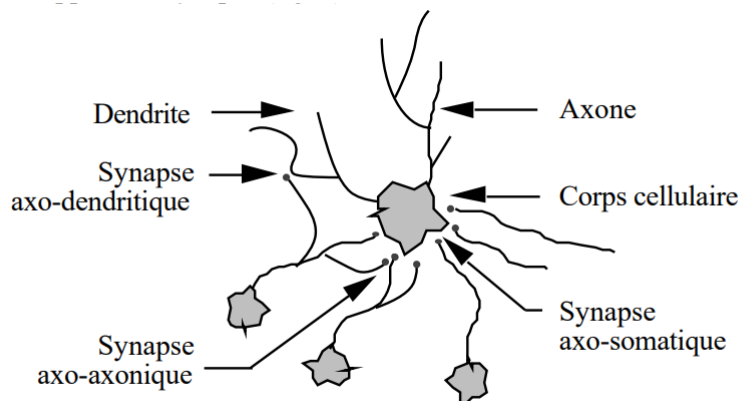


Figure 7: Structure d'un neurone biologique

Neurone Artificiel

Le neurone artificiel est une unité de traitement de l'information. Ce « mini processeur » calcule la somme pondérée des signaux en entrée et renvoie en sortie.

Les entrées x_i représentent en général les attributs de l'objet à classer (par exemple : les descripteurs d'images dans le cas d'une classification d'images) et les poids w_i (ou coefficients synaptiques). Ces derniers, associés aux entrées, forment les variables de la fonction d'activation.

Cette dernière dépendra de l'objectif désiré concernant le réseau de neurones (la fonction d'activation la plus utilisée est la somme pondérée des valeurs d'entrée : si $w_i x_i > 0$, $f=1$, $f=0$ autrement). Ensuite, la fonction de sortie prend la valeur d'activation comme argument et détermine la valeur de sortie S' . Toutefois, le neurone se sert de l'entrée supplémentaire S qui représente la valeur de sortie attendue pour corriger ses coefficients synaptiques.

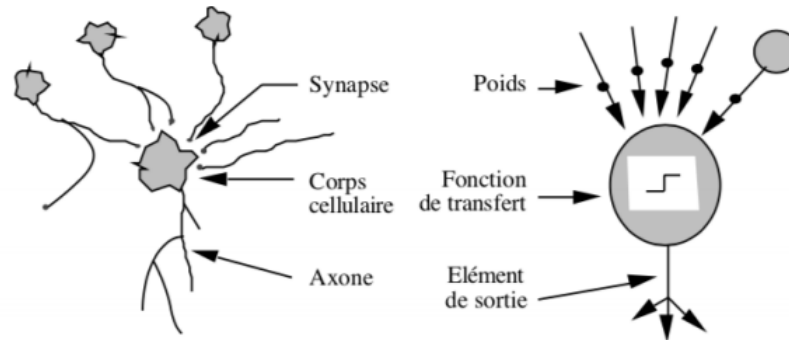


Figure 8: Structure d'un neurone artificiel

Réseaux de Neurones

Un réseau de neurones est constitué d'unités (les neurones) et de liens dirigés et pondérés (les connexions) pour lier les neurones entre eux. Un réseau de neurones peut prendre des formes différentes selon l'objet de la donnée qu'il traite et selon sa complexité et la méthode de traitement de la donnée. Les architectures ont leurs forces et faiblesses et peuvent être combinées pour optimiser les résultats. Le choix de l'architecture s'avère ainsi crucial et il est déterminé principalement par l'objectif. Les architectures de réseaux neuronaux peuvent être divisées en plusieurs types, on cite les 5 types suivants :

- Réseaux de neurones Feed forwarded
- Réseaux de neurones récurrent (RNN)
- Réseaux de neurones à résonance
- Réseaux de neurones auto-organisés
- Réseau de neurones de convolution (CNN)

Dans notre projet, on va se parler des réseaux de neurones Feed Forward, vu que c'est le type de réseaux de neurones qu'on va travailler avec pour réaliser notre projet.

Réseaux de neurones Feed forwarded

Feed-forwarded fait tout simplement référence à la procédure du traitement de la donnée par le réseau neuronal. En effet, feed-fowarded (propagation avant) signifie tout simplement que la donnée traverse le réseau d'entrée à la sortie sans retour en arrière de l'information. Typiquement, dans la famille des réseaux à propagation avant, on distingue les réseaux monocouches (perceptron simple) et les réseaux multicouches (perceptron multicouche).

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

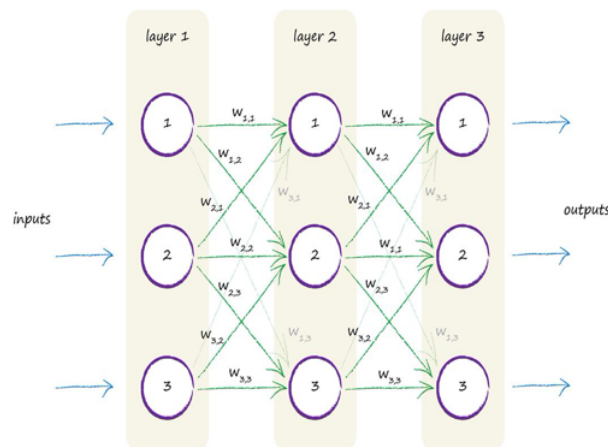


Figure 9: Réseau de neurones feed forward

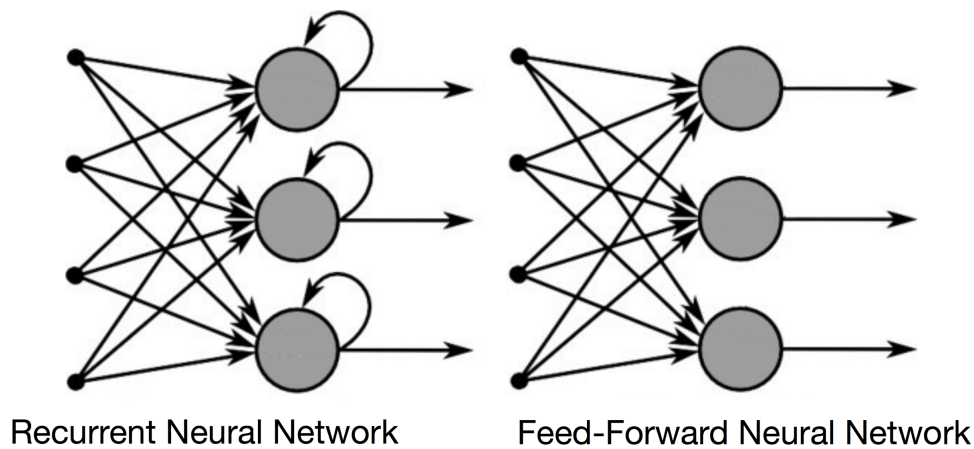


Figure 10: Différence entre Réseau de neurones feed forward et RNN

Types de fonction d'activation

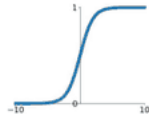
Il existe un grand nombre de fonctions d'activation, telles que:

- **Sigmoïde:** produit une courbe en forme de S. Bien que de nature non linéaire, il ne tient toutefois pas compte des légères variations des entrées, ce qui entraîne des résultats similaires.
- **Fonctions de tangente hyperbolique (tanh):** Il s'agit d'une fonction supérieure comparée à Sigmoid. Cependant, elle rend moins bien compte des relations **et elle est plus lente à converger.**
- **Unité linéaire rectifiée (ReLU):** Cette fonction converge plus rapidement, optimise et produit la valeur souhaitée plus rapidement. C'est de loin la fonction d'activation la plus populaire utilisée dans les couches cachées
- **Softmax:** utilisé dans la couche de sortie car il réduit les dimensions et peut représenter une distribution catégorique.

Activation Functions

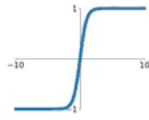
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



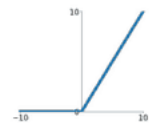
tanh

$$\tanh(x)$$



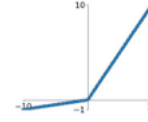
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$



Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

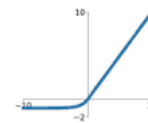


Figure 11: Les différents types de fonctions d'activation

Conclusion

Nous avons essayé à travers ce chapitre de donner une vue générale sur l'intelligence artificielle, et les méthodes d'apprentissage automatique qui sont incluses dans un processus d'extraction de connaissances. Et sont l'origine d'un nombre important d'algorithmes, nous avons écrit en détail quelques méthodes très connues dans le domaine de la classification et notamment celles que nous avons utilisés dans nos approches proposées, et aussi celles que nous avons exploité dans le cadre de reconnaissance.

Chapitre 2

Traitement de l'image

Contrairement aux êtres humains, les machines ne détectent pas facilement les choses. Une machine a besoin de certains outils sophistiqués de l'intelligence artificielle qui lui permet de traiter et d'apprendre les objets qu'elle voit devant elle. En général, une machine voit une image comme étant un ensemble de nombres avec une certaine structure.

3 Chapitre 2 : le traitement de l'image

3.1 Optical Character Recognition (O.C.R)

La reconnaissance de l'écriture est mieux connue sous le nom d'O.C.R (Optical Character Recognition). Les premières tentatives remontent aux années 1900 par TYURIN, au cours desquelles on inventa le scanner à balayage pour la télévision ; pendant les années soixante , et soixante dix, les premiers systèmes de l'écriture automatique du texte imprimé ont vu le jour. Aussi, en 1975 les japonais utilisaient couramment les lecteurs qui déchiffrent le code postal inscrit à la main ou tapé à la machine. Dans la même période, le Français CONTER construisait un système de lecture automatique de texte imprimé destiné aux non voyants, par la suite la compagnie américaine KURZWELL a amélioré le système précédent en proposant les machines à lire pour aveugle formulant le texte à haute voix par synthèse vocale. Durant cette phase les chercheurs se sont heurtés à de nombreuses difficultés outre la complexité du problème de la reconnaissance due à la grande variabilité de l'écriture manuscrite, la non disponibilité de mémoire et de puissance de calcul pour la réalisation des systèmes concrets opérant en temps réel, par contre depuis 1980 où les récents progrès électroniques et plus particulièrement l'avènement de calculateurs puissants à faible coût ont permis de résoudre ce type de problème et les recherches en reconnaissance manuscrite se sont multipliées de manière spectaculaire, et de nombreuses nouvelles techniques ont vu le jour.

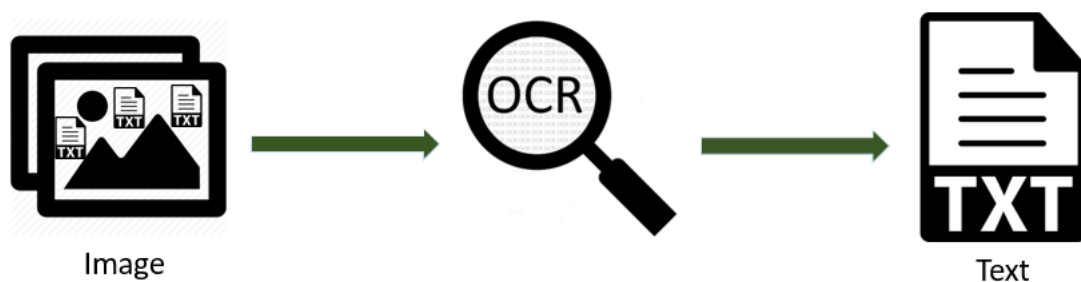


Figure 12: le scanner de document/OCR

3.2 Reconnaissance des chiffres

La reconnaissance de caractères signifie le décodage de n'importe quel texte imprimé ou manuscrit ou information symbolique, par exemple les formules mathématiques etc. On présente dans la figure (1.2) une dichotomie des types d'écriture faite selon le média, le support saisi et l'application considérée. La reconnaissance des caractères manuscrits est considérée plus difficile que la reconnaissance de l'imprimé pour lequel des systèmes opérationnels commerciaux existent.

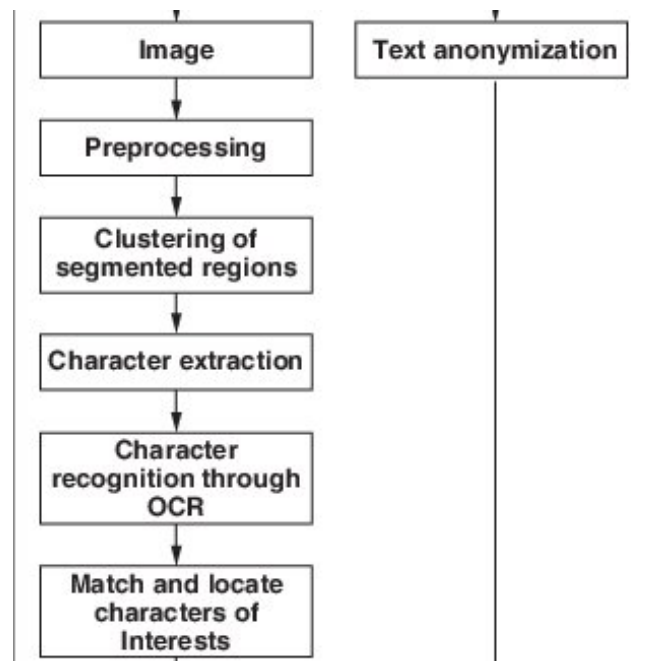


Figure 13: La dichotomie de la reconnaissance des caractères

3.2.1 Document manuscrit

L'écriture manuscrite comme la parole fait partie de nos modes d'expression les plus évolués en tous les cas les deux sont très complexes, c'est un moyen particulier de codage de l'information qui concerne nos idées, pensées, nos sentiments, d'une autre façon la reconnaissance de l'écriture en général possède deux caractéristiques : Un liée au scripteur par exemple vérifier une signature d'un auteur. L'autre liée au sens de ce qui est écrit (contextuelle).

3.2.2 La nature de l'écriture

- **Variabilité propre au scripteur:** cette variabilité traduit que l'écriture subit une légère (ou grande) modification avec le temps par rapport à ce que nous avons appris à l'école. Cette variabilité est bien attachée à la santé physique et la situation mentale du scripteur.
- **Variabilité intentionnelle :** Elle traduit la tâche à effectuer pour rédiger une note c'est pas la même chose pour écrire un brouillon, il faut tenir compte de la différence entre une écriture rapide et une autre normale .
- **Variabilité émotionnelle :** Cette variabilité se traduit par une écriture compacte, rectrice, bien arrangée dans le texte, jolie ou non. etc.
- **Variabilité circonstancielle:** Comme on le sait, l'écriture subit une déformation provoquée par les conditions matérielles et physiques dans les lesquelles le texte à été écrit (plume de stylo, l'avant bras de scripteur, position de feuille ...etc.
- **Variabilité contextuelle:** Cette variabilité est provoquée par le contexte lexical de chaque tracé, la forme d'une lettre et sa position dans un mot et sa place par rapport à la lettre voisine. Il existe ainsi une différence entre les écritures de scripteurs ce qui on appelle inter scripteur et une différence dans différentes écritures c'est-à-dire même scripteur intra scripteur

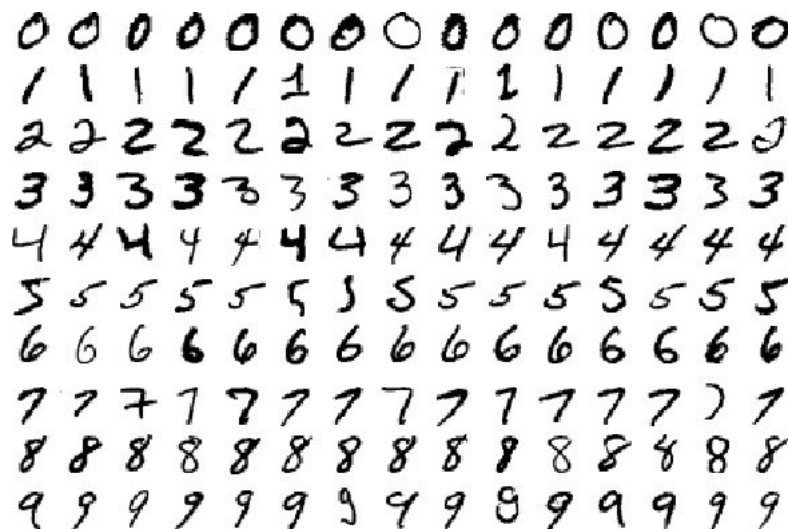


Figure 14: La base de données de MNIST

3.2.3 Réduction de la complexité d'écriture

- **Zonée** : le scripteur écrit dans des zones bien limitées.
- **Guidée** : dans ce cas l'écriture est guidée par l'existence d'une ligne support.
- **omni scripteur ou multi scripteur**
- **écriture libre**

Caractéristique	image de gris	image binaire	contour	squelette
Appariement	*	*		*
motifs déformable	*			*
transformation unitaire	*	*		
transformation log binaire	*	*		
moments géométriques	*	*		
moments de zernik	*	*		
ondelettes	*	*		
algébrique	*	*		
histogramme de projection		*		
masques	*	*		
profils de contour			*	
code de freeman			*	
spline			*	
descripteur de fourier			*	*
descripteur graphique				*
zonage	*	*	*	*

Figure 15: The nine screen regions

3.3 Méthodes de reconnaissance

3.3.1 Méthode structurelle

Elle est appelée aussi approche syntaxique où le caractère doit être indépendant le plus possible de la variation du style et de taille ; on attribue une forme élémentaire appelée dans ce cas primitive. Lors de la reconnaissance on déduit le type de caractère à partir de ces primitives

3.3.2 Méthode de texte

Elle est basée sur l'intersection du caractère avec les droites horizontales, verticales, diagonales, ou mixte, mais cette méthode devient difficile pour la reconnaissance lorsque les classes deviennent nombreuses.

3.3.3 Méthode basée sur le prototype

La méthode est basée sur la décomposition du caractère en un groupe de primitives appelé prototype, avec un ensemble de règles utilisées pour spécifier les interrelations entre les primitives qui sont nécessaires pour reconnaître les caractères.

Conclusion

Dans ce chapitre nous avons fait l'état de l'art sur le domaine de l'écriture manuscrite en général ,et les chiffres manuscrits isolés en particulier. Ainsi nous avons démontré les problèmes liés à la complexité de la reconnaissance

Chapitre 3

Analyse et Conception

Pour bien maîtriser le système, l'analyse s'impose comme une étape indispensable. Idem pour la conception qui a pour objectif de proposer de meilleures solutions qui satisfassent complètement nos besoins. Ce chapitre englobe toutes ces étapes et sert comme introduction au chapitre suivant qui traitera la phase de réalisation. Le but de ce chapitre est de bien concevoir le contexte dans lequel le développement de l'application de la reconnaissance des chiffres manuscrits s'inscrit moyennant un réseau de neurones *Feed Forwarded*.

4 Chapitre 3 : Analyse et Conception

Le but de ce chapitre est de bien concevoir le contexte dans lequel la construction de notre modèle de la reconnaissance des chiffres manuscrites s'inscrit moyennant ainsi un réseau de neurones Feed Forwarded.

4.1 Spécification du projet

Qu'est ce qui va faire exactement notre système de reconnaissance des chiffres manuscrites ? c'est la question dont la réponse représente la spécification de notre projet. En effet, le modèle doit apprendre à la base des données de Dataset de MNIST, et par la suite il doit être capable de reconnaître de nouveaux chiffres écrits à la main qu'on lui donne moyennant une interface graphique qui va permettre au utilisateur de reconnaître chaque chiffre il lui entre avec de différentes façons d'écriture humaine. Avant d'alimenter notre réseaux de neurones, il faut mentionner qu'on va tout d'abord faire une phase de preprocessing de l'image ainsi que son feature extraction. Notre NN donc va avoir plus spécifiquement non pas les images de MNIST, mais les caractéristiques extraites de chaque image. On va voir par la suite les détails de chaque phase préalablement mentionnée. La figure suivante montre un échantillon de Mnist dataset des écriture humaines :



Figure 16: Echantillon de Mnist Dataset des écritures humaines

4.2 Elaboration des concepts du Projet

La conception du projet est une phase très importante dans le projet, elle consiste à mettre à plat de façon aussi claire que possible le fonctionnement du processus de la reconnaissance faciale et poser le contexte dans lequel celui-là se déroule.

Cette étude permet de faciliter la reconnaissance des besoins et bien assimiler chaque étape. On présente par la suite l'ensemble des étapes qui nous permettent de construire le modèle conceptuel . Moyennant Visual Pradigm , on a pu concevoir le déroulement qui va être suivi avant d'alimenter notre réseau de neurones :



Figure 17: Logo de Visual Paradigm

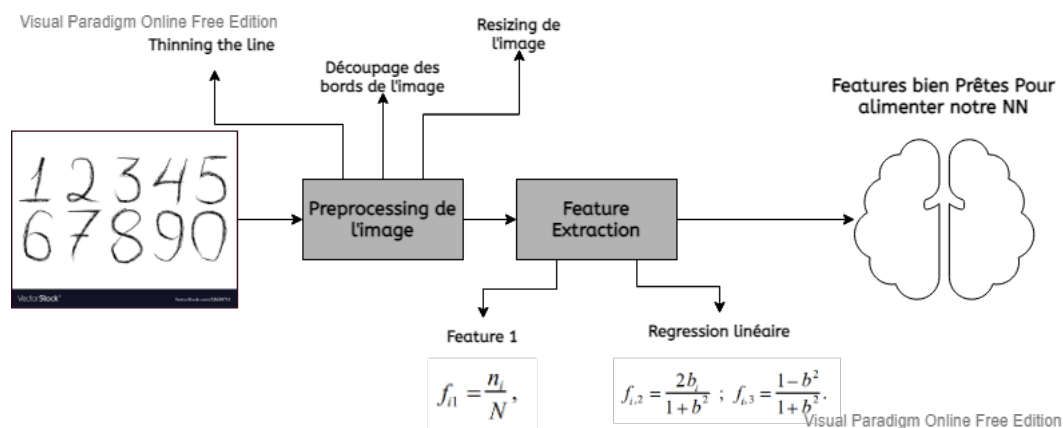


Figure 18: Schéma montrant le modèle conçu

Le processus conçu prendra comme input, les chiffres écrits à la main de l'utilisateur et par la suite on aura une phase de *Preprocessing de l'image* qui consiste en diminuer l'épaisseur de ce chiffre ainsi que découper les bords de l'image pour qu'on puisse avoir une region qui contient que les bords de chiffre seulement, sans omettre finalement de faire un resizing de l'image en 28*28 pixels pour la rendre comme celle de mnist. Pour le *feature Extraction*, on va extraire 3 caractéristiques qui vont être détaillées par la suite dans le même chapitre.

4.3 *Modèle Conceptuel Global*

Cette phase consiste à décrire comment notre modèle fonctionne d'un point de vue globale, on essayera par la suite de présenter les différentes étapes qui, en général, nous amènent au résultat escompté et ainsi de répondre aux spécifications du système préalablement abordées.

Après la phase de **Feature extraction** qui va être appliquée bien évidemment sur chaque **bloc** de l'image, les entrées de notre réseau de neurones vont être $3 \times k$ avec k : le nombre de bloc par image.

La figure suivante présente le processus qu'on a pu concevoir pour notre système d'un point de vue globale :

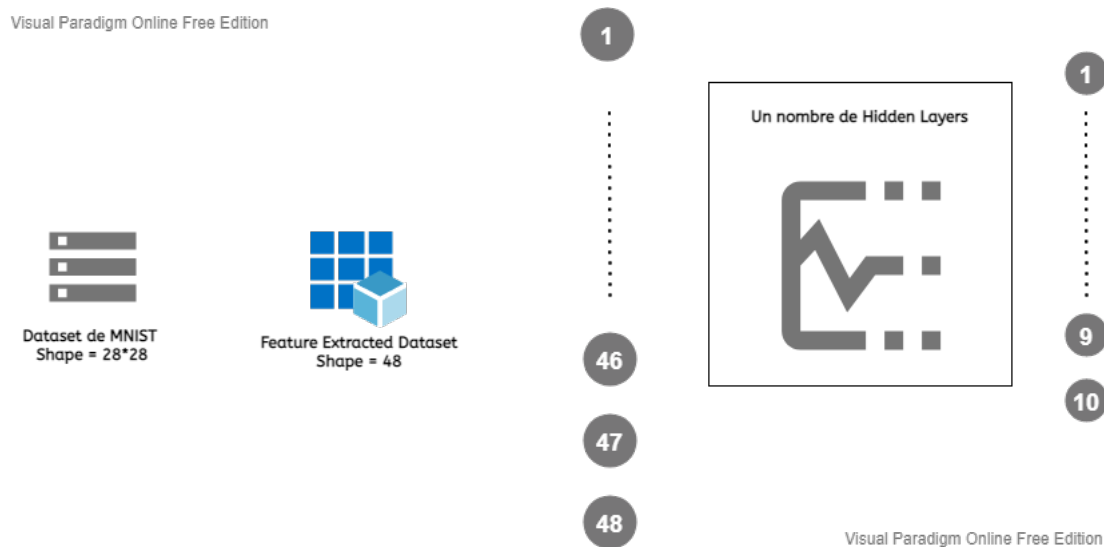


Figure 19: Processus conçu de modèle global

4.4 *Les étapes de modèle conçu*

Le preprocessing de l'image ainsi que le feature extraction de chaque chiffre représente une phase très importante dans l'élaboration de notre modèle. Ceci revient en fait, des différentes difficultés que trouve la machine afin de distinguer et de reconnaître l'écriture humaine manuscrite vu que chaque personne a sa propre façon d'écrire manuscritement, parmi ces difficultés on présentera les problèmes suivants :

$1 \rightarrow 1$ $2 \rightarrow 2$ $3 \rightarrow 3$
 $4 \rightarrow 4$ $5 \rightarrow 5$ $6 \rightarrow 6$
 $7 \rightarrow 7$ $8 \rightarrow 8$ $9 \rightarrow 9$
 $0 \rightarrow 0$

Figure 20: Manières différentes d'écriture humaine

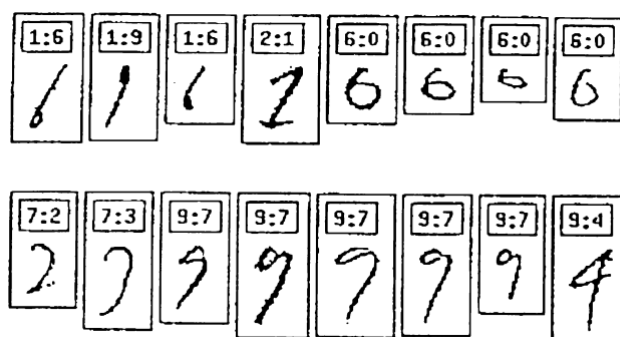


Figure 21: Confusion de la machine des écritures différentes humaines

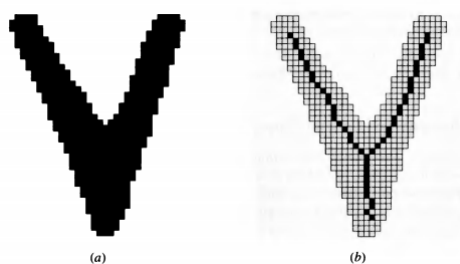


Figure 22: le thinning line de l'écriture

Pa la suite on présentera donc les détails des étapes qu'on a suivi en ce qui concerne les deux phases: **Preprocessing de l'image** et **Feature extraction**. Pour notre cas on va tester ceci sur des images de mnist pour visualiser ainsi le résultat de chaque étape.

4.4.1 Préprocessing de l'image

Pour notre cas on va tester ceci sur des images de mnist pour visualiser ainsi le résultat de chaque étape. La figure suivante représente un échantillon de données de Mnist :

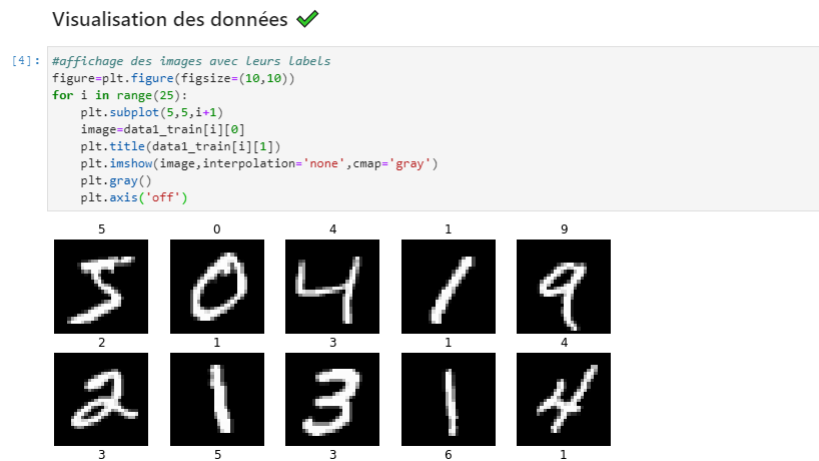


Figure 23: visualisation des données de Mnist



Figure 24: Before thinning the line

```

: #testing
img=data1_train.data[7]
img=thin(img)
plt.imshow(img, cmap="gray")
img.shape
: (28, 28)

```

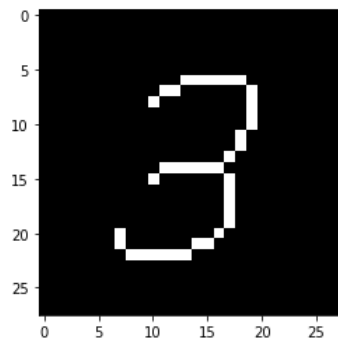


Figure 25: After thinning the line

La diminution de l'épaisseur de l'écriture de chiffre est bien réussie. Par la suite on va découper les bords notre image de test le chiffre 3 : Après la phase

```

[201]: #testing
b=crop(data1_train.data[7])
print(b.shape)
plt.imshow(b, cmap="gray")
(28, 28)
[201]: <matplotlib.image.AxesImage at 0x1d3e650f0a0>

```

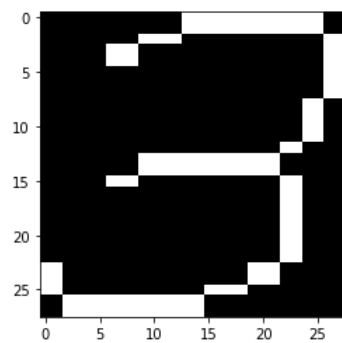


Figure 26: Découpage des bords de l'image

de preprocessing de l'image vient la phase de **Feature Extraction** .

4.4.2 Feature Extraction

On a mentionné que l'extraction des caractéristiques se fait par chaque bloc de l'image. Pour notre cas, on choisi de découper l'image en 16 bloc tout en se basant sur le paper research intitulé : "SIMPLE AND EFFECTIVE FEATURE EXTRACTION FOR OPTICAL CHARACTER RECOGNITION" et qui a été donnée comme support dans ce projet. La figure suivante montre le résultat de découpage en 16 blocs : le premier feature de notre bloc est représenté par le

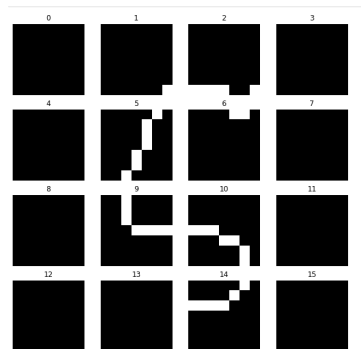


Figure 27: Découpage de l'image en 16 blocs

nombre total de pixels noir dans le bloc sur le nombre de pixels noirs dans la totalité de l'image. En ce qui concerne les 2 features restantes, on va les extraire moyennant une regression linéaire de chaque curve line comme il est illustré dans la figure suivante :

From the Paper : "SIMPLE AND EFFECTIVE FEATURE EXTRACTION FOR OPTICAL CHARACTER RECOGNITION"

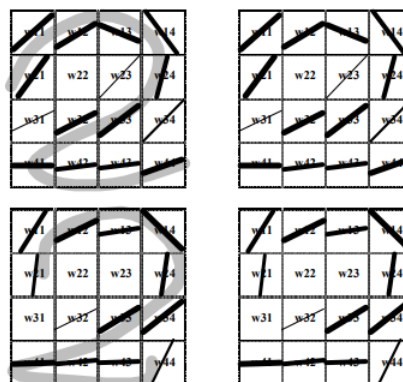


Figure 28: Fit the curve line to a straight one using linear regression

Après la récupération du slope b , on va extraire par la suite les deux features restantes sans omettre le premier feature qui est le nombre total de pixels noir de chaque bloc sur le nombre total des pixels noirs dans toute l'image :

From the Paper : "SIMPLE AND EFFECTIVE FEATURE EXTRACTION FOR OPTICAL CHARACTER RECOGNITION"

$$f_{i1} = \frac{n_i}{N}, \quad (1)$$

where n_i is the number of black pixels in the cell (or the sum of the grey values, if applicable) and N is the same magnitude measured for the whole character.

An adequate (unique and continuous) representation of the straight-line is, however, more difficult to achieve. A straight-line $y=a_i+b_ix$ is uniquely defined by two parameters: the slope b_i and the intercept a_i . In the proposed approach, the intercept is not used. The reason for this is related to a concept already mentioned: the invariance or tolerance to certain alterations of the patterns. In this case, invariance to position or size is not necessary, as a previous normalization can provide centered and fixed-size characters. Invariance to rotation is also unnecessary, as the orientation of the characters

perfectly continuous representations, but then the requirement of uniqueness fails: very different straight lines produce identical representations (e.g. $\cos(\pi/2) = \cos(-\pi/2)$). The fact seems to be that two parameters are needed to achieve the required continuity and uniqueness of the representation. Two fairly simple expressions that meet these requirements are

$$f_{i,2} = \frac{2b_i}{1+b_i^2}; \quad f_{i,3} = \frac{1-b_i^2}{1+b_i^2}. \quad (2)$$

Both are continuous (and give the same value for $b_i \nearrow \bullet$ and $b_i \searrow \bullet$) and, taken together, provide a unique representation for every value of b_i (they are the sine and cosine, respectively, of two times the angle $\arctan(b_i)$). In Figure 2, the behaviour of these

Figure 29: les 3 features extraites

Après l'extraction de features de chaque image on va préparer donc notre nouvelle dataset qui consiste en séparer les données en des données pour l'apprentissage, la visualisation et le test, et par la suite on décide sur l'architecture de notre réseau de notre réseau de neurones plus spécifiquement le choix de la fonction d'activation par chaque couche ainsi que le nombre de hidden layers finale de notre modèle :

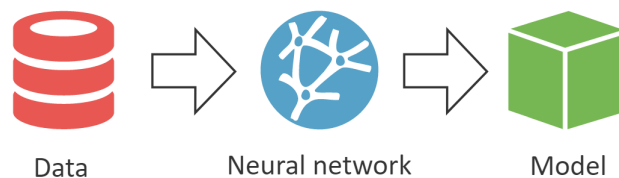


Figure 30: processus suivi de la construction du modèle

4.4.3 Architecture de notre NN

- **Nombres de couches** : 2 couches y compris avec un nombre de Hidden layers qui va être déterminé par la suite en se basant sur des expériences menées avec de différentes paramètres.
- **Nombres de neurones par couche** : Chaque image représente 16 blocs, chaque bloc a trois features, chaque feature est représenté par un seul neurone donc input = 48 neurones en entrée et 10 en sortie
- **Fonction d'activation** : la fonction selu et relu vu que leurs résultats sont les plus performants pour l'apprentissage de ce type de réseaux de neurones.
- **Loi d'apprentissage** : basé sur l'algorithme de rétropropagation qui permet ainsi de calculer le gradient de l'erreur pour chaque neurone, de la dernière couche vers la première.

Table 1: Recognition accuracy for handwritten digits at 0% rejection rate.

Feature Extraction Method	Classification Technique	Recognition rate
4x4 divisions LLF	48-20-20-10 MLP trained with Back-Propagation	91.33 %
4x4 divisions LLF	48-50-50-10 MLP trained with Back-Propagation	95.40 %
5x5 divisions LLF	75-20-20-10 MLP trained with Back-Propagation	94.75 %
5x5 divisions LLF	75-50-50-10 MLP trained with Back-Propagation	94.59 %
5x5 divisions LLF	11 prot. Incremental DSM (5% tr.set err.)	94.05 %
5x5 divisions LLF	15 prot. Incremental DSM (3% tr.set err.)	94.05 %
5x5 divisions LLF	26 prot. Incremental DSM (1% tr.set err.)	93.93 %
3x3 divisions LLF	Nearest Neighbour (2660 prototypes)	90.40 %
4x4 divisions LLF	Nearest Neighbour (2660 prototypes)	95.43 %
5x5 divisions LLF	Nearest Neighbour (2660 prototypes)	96.29 %
6x6 divisions LLF	Nearest Neighbour (2660 prototypes)	96.25 %
5x5 divisions LLF	Edited Nearest Neighbour (2558 prot.)	95.91 %
5x5 divisions LLF	Edited & Condensed Nearest Neighbour (30 prot.)	93.25 %
Re-rasterized to 12x12 pixels	Nearest Neighbour (2660 prototypes)	93.36 %
Re-rasterized to 16x16 pixels	Nearest Neighbour (2660 prototypes)	90.75 %
Re-rasterized to 22x16 pixels	Nearest Neighbour (2660 prototypes)	86.08 %

Figure 31: la précision de modèle en fonction de sa classification technique

Pour notre cas, on va travailler au début avec le 48-20-20-10 MLP vu que statistiquement il a la plus grande précision, mais ceci n'empêche pas de modifier cette structure en se basant bien évidemment sur les observations à chaque fois on fait l'apprentissage pour notre réseau de neurones.

4.4.4 Résultats des expériences

Pour les expériences menées, on a décidé de visualiser chaque paramètre en variant le nombre d'époques pour observer la variation de précision de modèle : Pour les deux premières expériences, on a choisit de travailler avec SGD optimizer et Adam optimizer et une architecture pour les deux de 3 hidden layers et avec un dropout. Les deux figures montrent une augmentation de la précision de modèle en ajoutant à chaque fois un nombre d'époques. Pour SGD optimizer la précision atteint environ le 96 %, alors que pour Adam elle a dépassé 96 %.

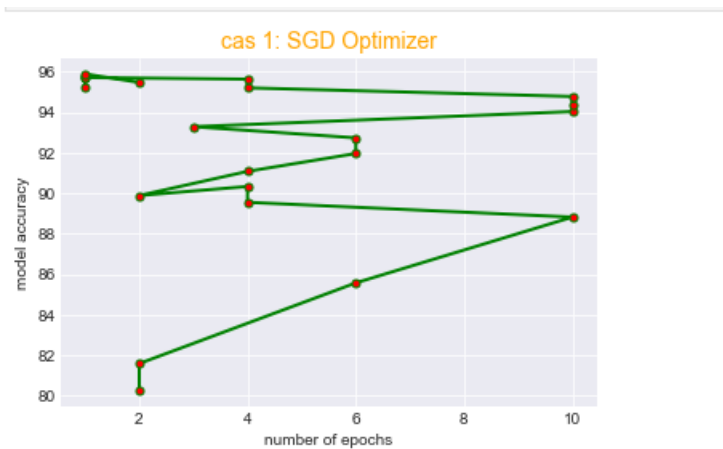


Figure 32: SGD Optimizer

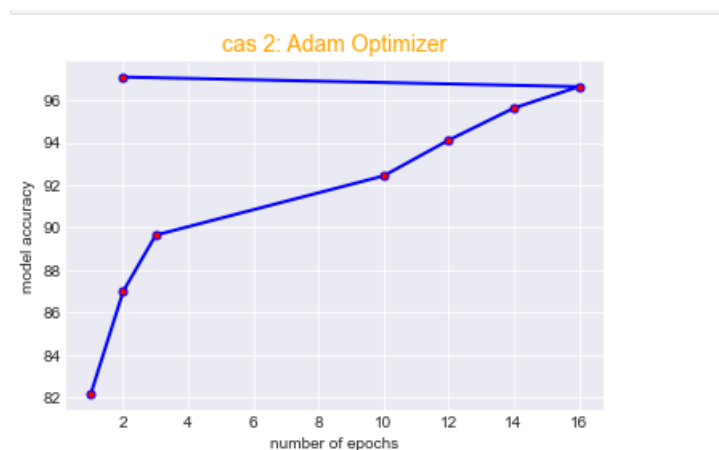


Figure 33: Adam Optimizer

On garde la même architecture avec Adam optimizer mais on enlève le dropout, la précision atteint le 96 %. Donc la 2^{ème} expérience reste la plus performante.

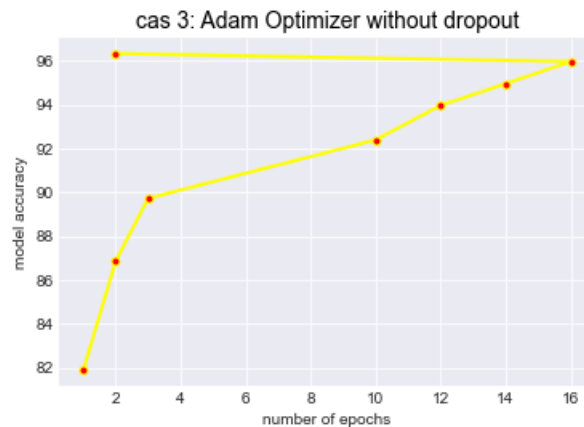


Figure 34: Adam Optimizer sans dropout

On garde la même architecture de la deuxième expérience mais avec seulement un seul Hidden layer. La précision n'atteint pas le 90 %. Donc la 2^{ème} expérience reste la plus performante.

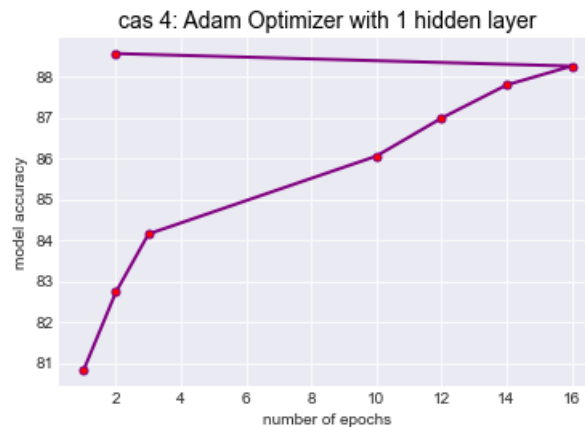


Figure 35: Adam Optimizer with 1 hidden Layer

On garde la même architecture de la deuxième expérience mais avec seulement 2 Hidden layers. La précision n'atteint pas le 94 %. Donc la 2^{ème} expérience reste la plus performante.



Figure 36: Adam Optimizer with 2 hidden layers

On garde la même architecture de la deuxième expérience mais cette fois-ci avec 3 Hidden layers. La précision dépasse donc le 96 %. Donc la 2^{ème} expérience et la dernière expérience restent les plus performantes et donnent une précision qui dépasse 96 %.

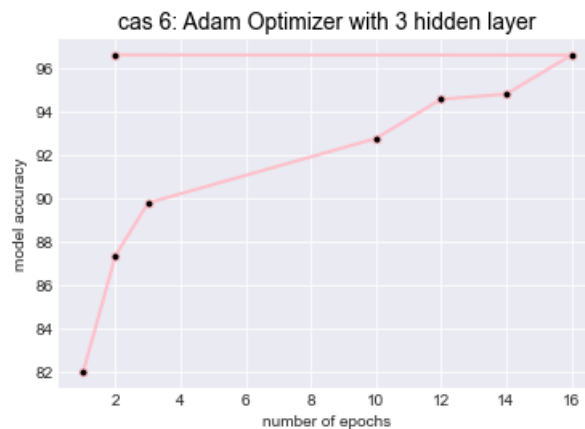


Figure 37: Adam Optimizer with 3 hidden Layers

Après les 6 expériences menées, le résultat de précision de 96 % est la grande précision que notre NN a pu atteindre en modifiant ainsi des paramètres. Ainsi, après beaucoup des modifications et de changements qu'on a fait au niveau de différentes paramètres comme le batch size, le nb d'époque, le type de fonction d'activation dans chaque couche, ainsi qu'on ajouter des régularisations et de normalisations améliorant ainsi l'apprentissage de modèle comme le dropout et aussi le batchnormalization, on a arrivé au résultat suivant :

```
epoch: 0, train loss: 0.1238, validation loss: 0.1063, correct predictions: 96.39%  
epoch: 1, train loss: 0.1215, validation loss: 0.1002, correct predictions: 96.59%
```

Figure 38: précision finale du modèle

```
class Net (nn.Module):  
    def __init__ (self):  
  
        super(Net, self).__init__()  
        self.fc1 = nn.Linear(48, 200)  
        nn.BatchNorm1d(200)  
        nn.Dropout(0.5)  
        self.fc2 = nn.Linear(200, 100)  
        nn.BatchNorm1d(100)  
        nn.Dropout(0.5)  
        self.fc4= nn.Linear(100, 20)  
        nn.BatchNorm1d(20)  
        nn.Dropout(0.5)  
        self.fc5= nn.Linear(20, 20)  
        nn.BatchNorm1d(20)  
        nn.Dropout(0.5)  
        self.fc3= nn.Linear(20, 10)  
  
    def forward(self, x):  
        x = F.selu(self.fc1(x))  
        x = F.relu(self.fc2(x))  
        x = F.selu(self.fc4(x))  
        x = F.relu(self.fc5(x))  
        x = self.fc3(x)  
        return x
```

Figure 39: architecture finale du modèle

Dans ce chapitre, on a vu la conception de notre modèle ainsi que sa construction, de l'image processing des données avec lesquels on veut alimenter notre réseau de neurones jusqu'à la précision finale de modèle . Par la suite, on va voir dans le chapitre suivant le déploiement de notre modèle au sein d'une application pour visualiser les résultats de son apprentissage.

Chapitre 4

Réalisation de l'application

La dernière étape du processus de développement concerne l'implémentation de l'interface en fonction des technologies et frameworks choisies. Cette étape correspond à la phase de construction du service principale. Nous commencerons tout d'abord, par la présentation de l'environnement de travail utilisé pour développer notre projet d'intelligence artificielle. Puis nous donnerons quelques captures de notre interface démontrant ainsi son niveau de performance et les différents résultats.

5 Chapitre 4 : Réalisation de l'application

5.1 Les outils et technologies utilisés

- **CV2:** OpenCV-Python est une bibliothèque de liaisons Python conçue pour résoudre les problèmes de vision par ordinateur. La méthode `cv2.imread()` charge une image à partir du fichier spécifié. Si l'image ne peut pas être lue (à cause d'un fichier manquant, de permissions incorrectes, d'un format non pris en charge ou non valide), cette méthode renvoie une matrice vide.

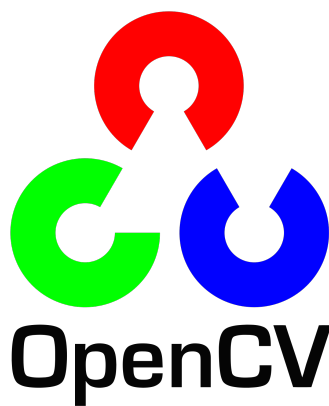


Figure 40: Logo de OpenCv

- **PIL:** Python Imaging Library est une bibliothèque supplémentaire gratuite et open-source pour le langage de programmation Python qui ajoute le support pour l'ouverture, la manipulation et la sauvegarde de nombreux formats de fichiers d'images différents. Elle est disponible pour Windows, Mac OS X et Linux.



Figure 41: Logo de PIL

- **Tkinter:** Tkinter est une liaison Python avec l'interface graphique de Tk. Il s'agit de l'interface Python standard de la boîte à outils Tk GUI, et est de facto l'interface graphique standard de Python. Tkinter est inclus dans les installations standard de Python sous Linux, Microsoft Windows et Mac OS X. Le nom Tkinter vient de Tk interface.

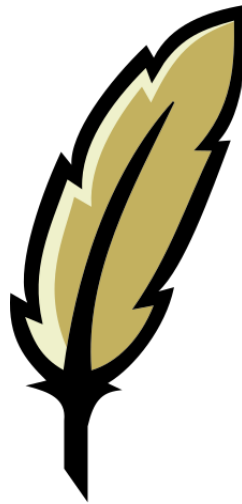


Figure 42: Logo de Tkinter

- **Pytorch:** PyTorch est une bibliothèque d'apprentissage de machines open source basée sur la bibliothèque Torch, utilisée pour des applications telles que la vision par ordinateur et le traitement du langage naturel, principalement développée par le laboratoire AI Research de Facebook. Il s'agit d'un logiciel libre et open-source publié sous la licence BSD modifiée.



Figure 43: Logo de Pytorch

- **Sklearn:** Scikit-learn (Sklearn) est la bibliothèque qui fournit une sélection d'outils efficaces pour l'apprentissage machine et la modélisation statistique, y compris la classification, la régression, le regroupement et la réduction de la dimension via une interface de cohérence en Python. Cette bibliothèque, qui est en grande partie écrite en Python, est construite sur NumPy, SciPy et Matplotlib.



Figure 44: Logo de Sklearn

5.2 Les différents résultats de l'interface

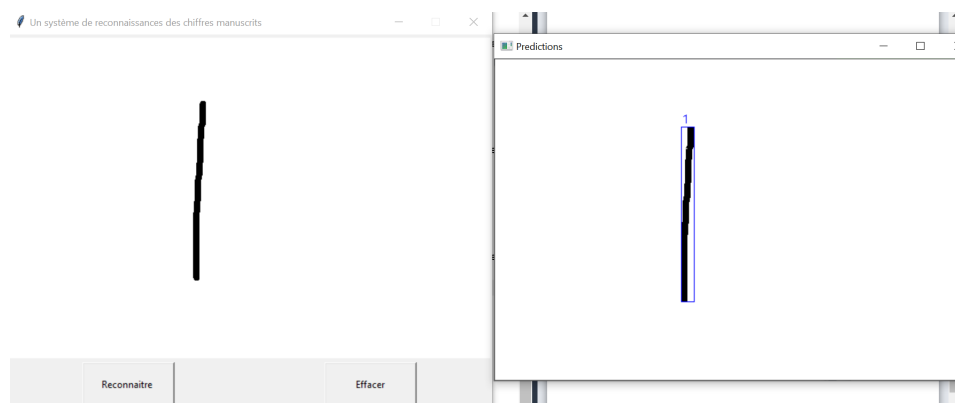


Figure 45: Le chiffre 1

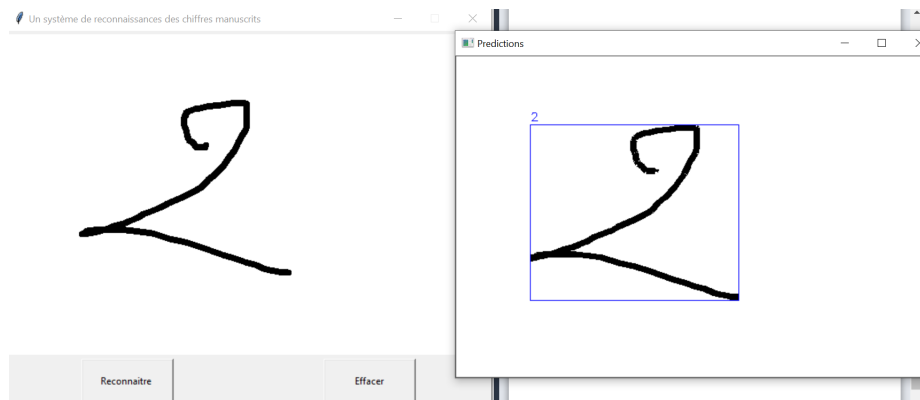


Figure 46: Le chiffre 2

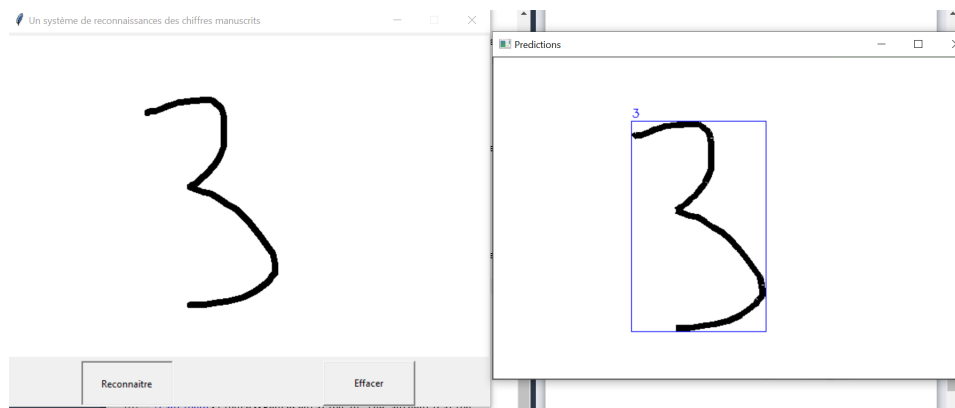


Figure 47: Le chiffre 3

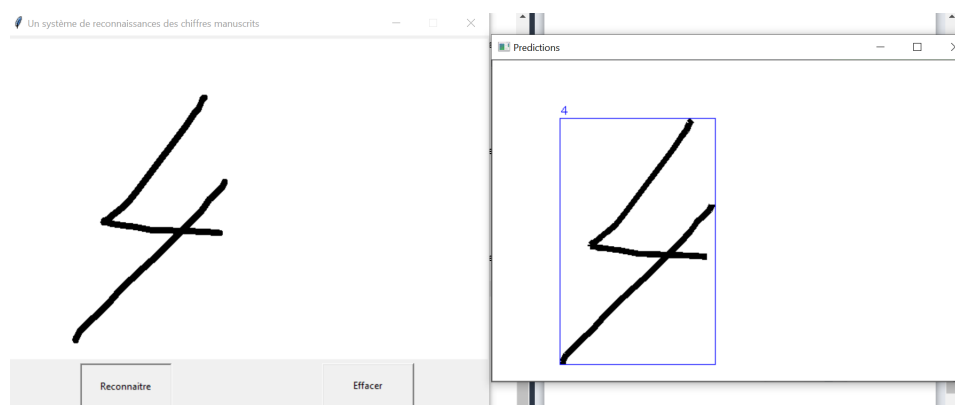


Figure 48: Le chiffre 4

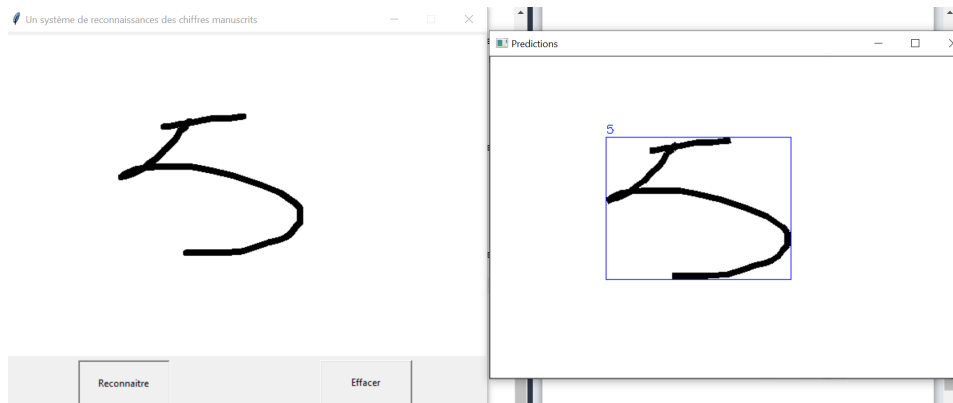


Figure 49: Le chiffre 5

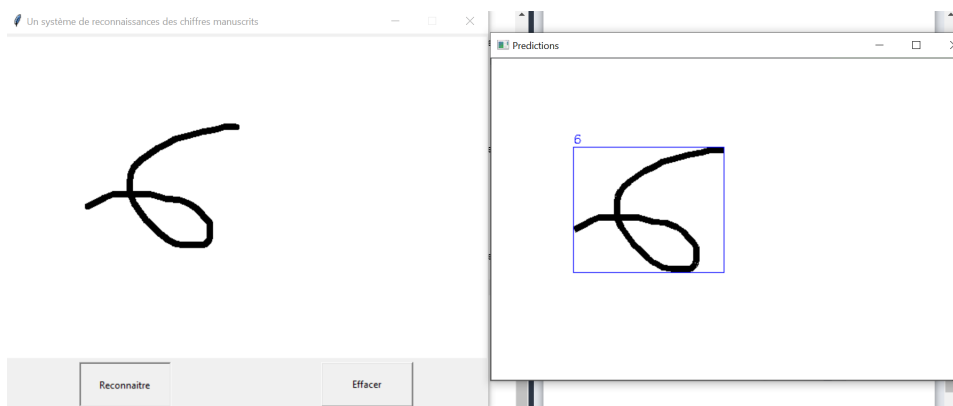


Figure 50: Le chiffre 6

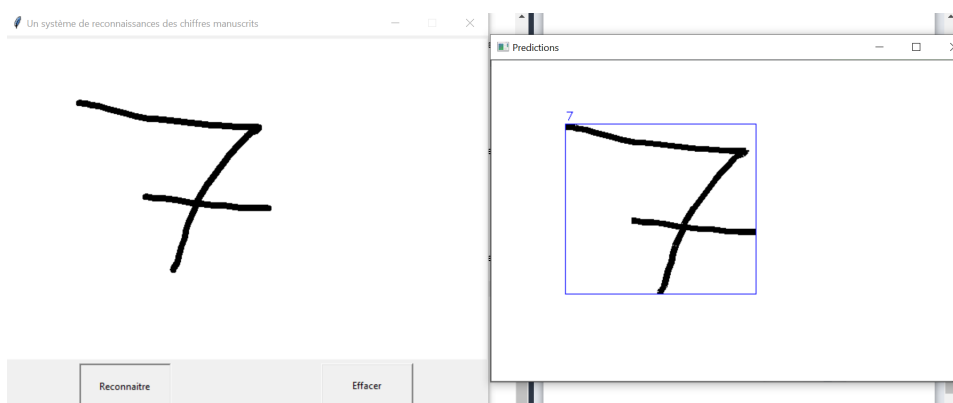


Figure 51: Le chiffre 7

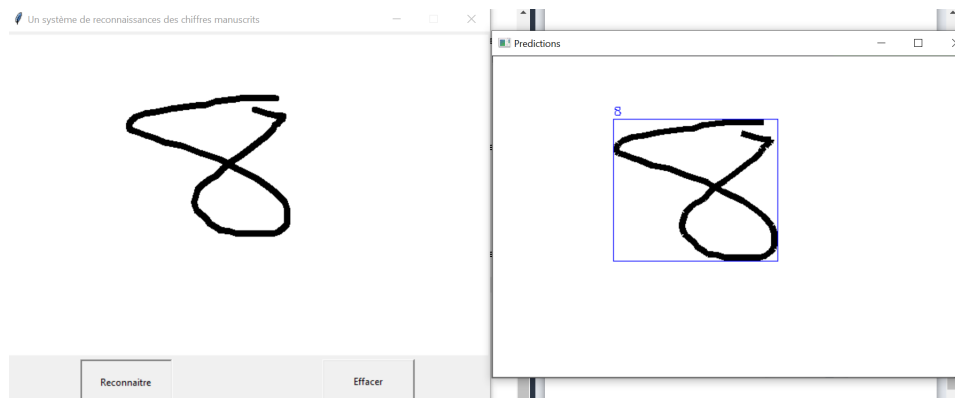


Figure 52: Le chiffre 8

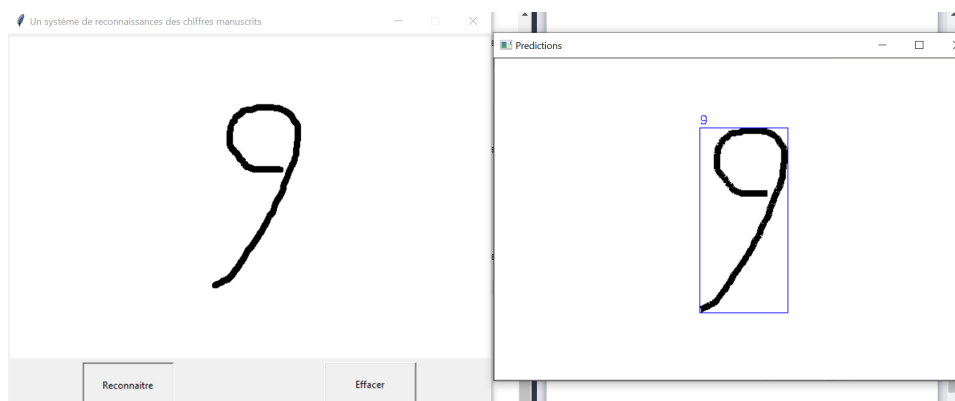


Figure 53: Le chiffre 9

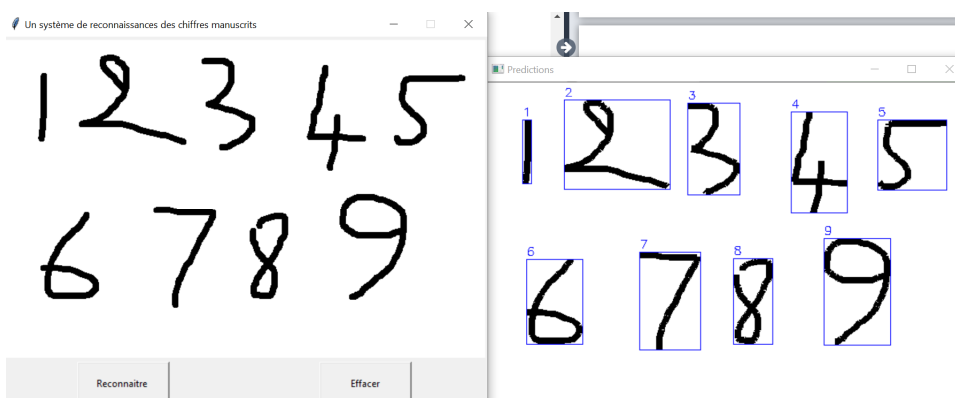


Figure 54: Tous les chiffres

6 Conclusion générale

Le projet a été couronné de succès, car après les étapes de pré-traitement et d'extraction d'attributs, les résultats obtenus sont, dans un certain sens, prometteurs. Cependant, le travail doit être développé davantage pour atteindre le résultat final d'un modèle complet de reconnaissance des chiffres manuscrits. Les éléments de base ont déjà été réalisés et nous espérons pouvoir développer ce projet à une autre occasion.

Nous avons fini par toucher à de nombreux sujets et technologies pour atteindre l'objectif final, du pytorch, sklearn, jusqu'aux extractions de fonctionnalités. Ces concepts nous ont aidés à accroître notre expérience et nos connaissances.

7 Bibliographie

- **Les Labs de l'intelligence artificielle**, de Mme BENHIBA Lamia, 2019-2020
- **Les cours de l'intelligence artificielle**, de Mme BENBRAHIM Houda, 2019-2020
- SHUNJI MORI, **Optical character Recognition**, *university of Aisu, Fukushima City, Japan copyright, ©1999 by John Wiley and Sons*
- **Documentation de Pytorch** : <https://pytorch.org/docs/stable/index.html>
- Alexander J. Faaborg, **Using Neural Networks to Create an Adaptive Character Recognition System**, ©2002 Cornell University, Ithaca NY