

N°d'ordre :

N°de série :

République Algérienne Démocratique et Populaire



Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique



UNIVERSITÉ ECHAHID HAMMA LAKHDAR - D'EL OUED

FACULTÉ DES SCIENCES EXACTES

DÉPARTEMENT D'INFORMATIQUE

Mémoire de fin d'étude

MASTER ACADÉMIQUE

Domaine: Mathématique et Informatique

Filière: science

Spécialité : **Informatique**

Thème

**Développement d'un système
de reconnaissance
de chiffres manuscrits**

Présenté par:

Hala Djerouni

Soutenu devant le jury composé de

M. Berdjouh Chafik	Université de Eloued	..	Président
M. Guia Sana Alsahar	Université de Eloued	..	Examineur
M. Zaiz Faouzi	Université de Eloued	..	Encadreur

Année universitaire 2020/2021

Dédicaces

Tout d'abord , je veux rendre grâce à Dieu , le Clément et le Très Miséricordieux pour son amour éternel .

Je dédie ce travail :

A mon chère père , que Dieu lui fasse miséricorde , qui m'a toujours soutenu ,

A ma chère mère ,

À mes chères sœurs et leurs enfants pour leurs encouragements constants et leur soutien moral ,

A mes chers frères ,

À mes chères amies , à tous ceux qui m'aiment et à tous ceux que j'aime .

À mon encadreur Zaiz Faouzi , je tenais à vous remercier de tout mon cœur pour le soutien dont , tout au long de cette année scolaire.

Merci à tous d'être toujours à mes côtés .

Que Dieu vous accorde la santé , le bonheur , le courage et , par-dessus tout , le succès .

Résumé

Jusqu'aujourd'hui, la reconnaissance des chiffres joue un rôle important dans plusieurs domaines tels que l'authentification de chèques bancaires, l'échange à distance des fichiers informatiques et la reconnaître des notes des élèves et être saisie dans fichier Excel ...etc. La reconnaissance des chiffres manuscrit consiste à convertir une image de texte imprimée ou manuscrit en un fichier numérique, puis les traiter par ordinateur. Nombreuses recherches ont été faites durant les dernières années afin d'améliorer le taux de reconnaissance. Mais, malgré ça il reste encore beaucoup d'imperfections où on trouve des lignes irrégulières et qui se chevauchent et connectées ... etc.

Pour un système de reconnaissance des notes des étudiants, la phase la plus importante reside dans la séparation des chiffres les uns des autres. Cette phase appelée "segmentation" reste cependant delicate à cause du chevauchement et/ou l'accolement de deux chiffres consecutifs. Dans ce travail, deux méthodes de segmentation ont été combinées fondées sur l'histogramme de projection et détection des contours. Nous avons utilisé la méthode de projection verticale afin de définir les colonnes de l'image pour séparer les chiffres connectés. Ensuite, nous appliquons la méthode de détection des contours pour définir les bords des objets dans l'image et segmenter chaque objet en une petite image et apprendis à la connaître. En passant par phases. Tout d'abord un prétraitement à été effectué pour transformer l'image en image binaire et changer les images à des tailles standard. Ensuite une phase de segmentation en utilisé une méthode de projection verticale et de détection de contour sur l'image, et enfin nous appliquons algorithmes d'apprentissage pour reconnaître les chiffres de l'image.

Nous avons mené un groupe de tests pour un ensemble d'algorithmes d'apprentissage automatique et d'apprentissage profondeur sur une base de données Mnist. Nous avons remarqué que la algorithmes CNN est efficace dans le processus de reconnaissance, atteignant la précision 96 % et les résultats obtenus ont été très convaincants et satisfiants. Et donc nous avons opté la méthode de réseau neuronal convolutif pour décider la classe d'appartenance pour chaque chiffre. L'approche proposée à été testée sur deux cas des modèles et toutes les expériences réalisées ont montré des résultats très encourageants. Pour le premier Modèle nous avons obtenus 95.81 % et pour le second Modèle nous obtenus 93.94 % .

Mots Clés: chiffres manuscrits, segmentation, l'apprentissage automatique, L'apprentissage profond, détection de contour, projection verticale, CNN

Abstract

Until today, digit recognition plays an important role in several areas such as authentication of bank checks, remote exchange of computer files and recognition of student notes and being entered in Excel file ...etc. Handwritten digit recognition involves converting an image of printed or handwritten text to a digital file and then processing them by computer. Much research has been done in recent years to improve the recognition rate. But, despite this there are still a lot of imperfections where we find irregular and overlapping lines and connected ...etc.

The most important phase for a student grade recognition system is to separate the digits from each other. This phase, called "segmentation", however, remains delicate because of the overlap and / or the juxtaposition of two consecutive digits. In this work, two segmentation methods were combined based on the projection histogram and edge detection. We used the vertical projection method to define the columns of the image to separate the connected digits. Then we apply the edge detection method to define the edges of the objects in the image and segment each object into a small image and get to know it. Going through phases. First a preprocessing was done to transform the image into a binary image and change the images to standard sizes. Then a segmentation phase using a method of vertical projection and edge detection on the image, and finally we apply Learning algorithms to recognize the numbers in the image.

We conducted a group of tests for a set of machine learning and deep learning algorithms on a Mnist database. We noticed that the CNN algorithm is efficient in the recognition process, achieving 96 % accuracy and the results obtained have been very convincing and satisfactory. And so we opted for the Convolutional Neural Network method to decide the membership class for each digit. The proposed approach has been tested on two cases of the models and all the experiments carried out have shown very encouraging results. For the first Model we obtained 95.81 % and for the second Model we obtained 93.94 % .

Keywords: handwritten digits, segmentation, machine learning, Deep Learning, edge detection, vertical projection, CNN

الملخص

حتى اليوم، يلعب التعرف على الأرقام دوراً مهماً في العديد من المجالات مثل مصادقة الشيكات المصرفية، وتبادل ملفات الكمبيوتر عن بعد والتعرف على علامات الطلاب بعد التصحيح وادخالها في ملف Excel... إلخ. يتضمن التعرف على الأرقام المكتوبة بخط اليد تحويل صورة نص مطبوع أو مكتوب بخط اليد إلى ملف رقمي ثم معالجتها بواسطة الكمبيوتر. تم إجراء الكثير من الأبحاث في السنوات الأخيرة لتحسين معدل التعرف. ولكن بالرغم من ذلك مازال هناك العديد من العيوب حيث نجد خطوط غير منتظمة ومتداخلة ومتصلة... إلخ.

في نظام التعرف على علامات الطلاب، فإن أهم مرحلة هي فصل الأرقام عن بعضها البعض. ومع ذلك، تظل هذه المرحلة التي تسمى التجزئة حساسة بسبب التداخل و \ أو تجاوز رقمين متتاليين. في هذا العمل، تم الجمع بين طريقتين للتجزئة بناءً على الرسم البياني للإسقاط وكشف الحواف. استخدمنا طريقة الإسقاط الرأسي من أجل تحديد أعمدة الصورة لفصل الأرقام المتصلة، ثم نطبق طريقة اكتشاف الحواف لتحديد حواف الكائنات في الصورة وتقسيم كل كائن إلى صورة صغيرة والتعرف عليها. حيث تمر بمراحل أولاً، تم إجراء معالجة مسبقة لتحويل الصورة إلى صورة ثنائية وتغيير الصورة إلى الأحجام القياسية. ثم مرحلة التجزئة باستخدام طريقة الإسقاط الرأسي واكتشاف الحواف في الصورة، وأخيراً نقوم بتطبيق خوارزميات التعلم للتعرف على الأرقام الموجودة في الصورة.

لقد أجرينا مجموعة من الاختبارات لمجموعة من خوارزميات التعلم على قاعدة بيانات Mnist. لاحظنا أن طريقة CNN فعالة في عملية التعرف، حيث بلغت دقته % 96 وكانت النتائج التي تم الحصول عليها مقنعة ومرضية للغاية. لذا اخترنا طريقة CNN لتحديد فئة العضوية لكل رقم. تم اختبار النهج المقترح على حالتين من النماذج وأظهرت جميع التجارب التي أجريت نتائج مشجعة للغاية. بالنسبة للنموذج الأول، حصلنا على دقة % 95.81. وللنموذج الثاني حصلنا على دقة % 93.94.

كلمات مفتاحية: الأرقام المكتوبة بخط اليد، التجزئة، تعلم الآلة، التعلم العميق، كشف الحواف، الإسقاط الرأسي، CNN.

Table des matières

Dédicaces	i
Résumé	ii
Table des matières	v
Liste des figures	viii
Liste des tableaux	x
Liste des algorithmes	xi
Introduction	1
1 Reconnaissance de l'écriture Manuscrite	3
Introduction	3
1.1 Différents aspects de reconnaissance	3
1.1.1 Production et reconnaissance	3
1.1.2 Reconnaissance en-ligne / hors-ligne	4
1.2 Architecture générale d'un system de reconnaissance	5
1.2.1 Phase d'acquisition	5
1.2.2 Phase de pré-traitement	6
1.2.3 Phase de Segmentation	8
1.2.4 Phase d'extraction de caractéristiques	8
1.2.5 Phase de Classification	10
1.3 Les approches de reconnaissance	11
1.3.1 Approche statistique	11

1.3.2	Approche structurale	11
1.3.3	Approche stochastique	12
1.3.4	Approche hybride	12
1.4	Domain d'application	12
	Conclusion	13
2	Segmentation et reconnaissance des chiffres	14
	Introduction	14
2.1	Problématique posée par l'écriture du cursif	14
2.2	Principe général de la segmentation	15
2.2.1	Segmentation de page	15
2.2.2	Segmentation en lignes	16
2.2.3	Segmentation en séquence de chiffres	16
2.2.4	Segmentation en chiffres isolés	16
2.3	Approches de segmentation	17
2.3.1	Segmentation explicite	17
2.3.2	Segmentation implicite	17
2.4	Méthodes de segmentation	17
2.4.1	Segmentation à partir du squelette	17
2.4.2	Segmentation basée sur les contours	18
2.4.3	Segmentation basée sur L'HPV	18
2.4.4	Segmentation basée sur les fenêtres glissantes	19
2.5	l'apprentissage automatique	19
2.5.1	Définition	19
2.5.2	Les différents types d'apprentissage automatique	20
2.5.3	Classifications	21
2.5.4	Régression	21
2.5.5	Algorithmes d'apprentissage automatique	21
2.6	L'apprentissage profond	25
2.7	Réseau neuronal convolutif	25
2.7.1	Couches CNN	26
2.7.2	Fonctions d'activation	27
2.7.3	Variables de réseau de neurones	28
2.8	Quelque travaux antérieurs	28

Conclusion	30
3 Architecture du système	31
Introduction	31
3.1 Architecture générale du système	31
3.2 Explication d'architecture	32
3.2.1 Acquisition	32
3.2.2 Prétraitement	33
3.2.3 Segmentation	34
3.2.4 Apprentissage	37
3.2.5 Prédiction	37
3.2.6 Insertion des notes dans le fichier Excel	38
Conclusion	40
4 Validation	41
Introduction	41
4.1 Langage de programmation et bibliothèques utilisées	41
4.1.1 Python	41
4.1.2 Jupyter Notebook	42
4.1.3 OpenCV	42
4.1.4 TensorFlow	42
4.1.5 Scikit-learn	42
4.1.6 Keras	42
4.1.7 Configuration utilisée	42
4.2 Le base d'image MNIST	42
4.3 Modèle CNN utilisé	43
4.4 Résultats & Discussion	44
Conclusion	47
Conclusion et perspectives	48
Bibliographie	49

Liste des figures

1	Reconnaissance de l'écriture Manuscrite	3
1.1	Processus de production et de reconnaissance de documents.[50]	4
1.2	Différence entre les deux types de reconnaissance (online , offline).[45] . .	5
1.3	Architecture général d'un système de reconnaissance des chiffres.[50] . . .	5
1.4	Exemple de la binarisation.	6
1.5	Exemple d'un chiffre manuscrit normalisé.[2]	7
1.6	Exemple de deux types de redressement.[24]	7
1.7	Exemple de squelettisation.[2]	8
1.8	Exemple de Segmentation.	8
1.9	Exemple d'extraction de caractéristiques.	9
1.10	Division d'une image en zone.	9
2	Segmentation et reconnaissance des chiffres	14
2.1	Exemple chaîne de chiffre cursif "52".	14
2.2	Illustration du processus de segmentation.	15
2.3	Détection des zones d'une page de document.	16
2.4	Segmentation du tableau en Lignes.	16
2.5	Segmentation de ligne en séquence de chiffres.	16
2.6	Segmentation de séquence de chiffres en chiffres.	17
2.7	Exemple de squelettes de chiffre.	18
2.8	Extraction des contours.	18
2.9	Segmentation des chiffres par l'histogramme de projection verticale. . . .	18
2.10	Segmentation à base de fenêtre glissante.	19
2.11	L'algorithme ML.	19
2.12	Apprentissage supervisée.	20
2.13	Apprentissage non supervisée.	20
2.14	Machines à Vecteurs de Support SVM.	22
2.15	K plus proches voisins.	22
2.16	Exemple de réseaux de neurones.	23
2.17	Exemple des arbres de décision.	23
2.18	Exemple de forêt aléatoire.	24
2.19	Bootstrap aggregating.	25
2.20	Illustration de l'opération de convolution.	26

2.21	Différence entre Max, Average pooling.	26
2.22	Fonction Relu.	27
2.23	Architecture complète d'un CNN.	27
3	Architecture du système	31
3.1	Architecture du système de reconnaissance de chiffres manuscrits.	32
3.2	Exemples des images des deux modèles.	33
3.3	Exemple d'image binaire.	33
3.4	Normalisation de taille.	34
3.5	Détecter les rectangles.	35
3.6	Numéro d'inscription (images séparés)	35
3.7	modèle 1: La note est composée des images séparés.	36
3.8	model 2: La note est une séquence de chiffres (une image).	36
3.9	modele 2 : la note séparé en deux.	36
3.10	modele 2 : Détection l'objet.	37
3.11	Fichier Excel	39
4	Validation	41
4.1	Répartition de la base MNIST pour les chiffres.	43
4.2	Exemples de la base de données MNIST.	43
4.3	Modèle CNN.	44
4.4	Évaluation de la précision du modèle.	45
4.5	Évaluation de la perte du modèle.	45

Liste des tableaux

2	Segmentation et reconnaissance des chiffres	14
2.1	Comparaison entre quelques travaux antérieurs.	29
4	Validation	41
4.1	Les résultats de training et test sur MNIST.	44
4.2	Résultats de taux de reconnaissance pour chaque modèle.	46

Liste des algorithmes

3	Architecture du système	31
3.1	Algorithme localiser les rectangles.	34
3.2	Segmentation de la séquence de chiffres.	37
3.3	Algorithme de prédiction d'image	38

Introduction

Dans le domaine de l'intelligence artificielle, la reconnaissance de l'écriture (OCR: Optic Character Recognition) recouvre un domaine très vaste, est l'un des problèmes les plus difficiles. En effet, vue la nature très complexe de la langue arabe, De tels systèmes reste un défi soulevé.

En général, deux types de systèmes de reconnaissance de l'écriture sont connus : reconnaissance hors-ligne et reconnaissance en-ligne. Le premier, fonctionne sur les images numérisées à l'aide d'un scanner ou caméra. Le second prend les données directement à partir d'un stylo électronique ou une tablette de numérisation pour les transformées en un texte digitalisé.

Le processus de reconnaissance très facile aux humains, mais il est si difficile pour la machine d'exécuter une telle tâche. 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 variabilité de l'écriture, des différents styles et formes de manuscrits, des chiffres qui se chevauchent, des chiffres connectes.) tendent à compliquer la reconnaissance.

Dans notre vie quotidienne nous utilisons beaucoup de formulaires administratifs, des chèques postaux et bancaires, l'échange à distance des fichiers informatiques pour les télécommunications, la lecture automatique de documents administratifs et la reconnaissance des notes des étudiants / élèves pour les saisir dans fichier Excel ... etc. Toutes ces opérations peuvent être réalisées de manière automatique . Économisant ainsi du temps et des efforts et réduisant les erreurs.

La phase clé de la reconnaissance de chiffre manuscrit est la segmentation. Elle fonctionne en segmentant chaque séquence de chiffres en une suite d'images des chiffres isolés. Par la suite, il est possible de reconnaître chaque chiffre séparément. Les résultats de cette phase affectent directement la performance du système.

Dans ce travail, nous avons essayé de présenter une solution concernant la reconnaissance des chiffres manuscrits et surtout les cas de nombres connectés et chevauchement en utilise méthode base sur projection verticale et detection de contour.

La reconnaissance des notes des élèves se fait principalement en cinq (05) étapes. La

premiere étape consiste à l'acquisition d'une image brute, puis procède à pretraitement. Ensuite, les images passent par une phase de segmentation afin d'extraire les segments de base. Ensuite, il est passé à l'étape d'apprentissage où les tests ont été exécutés sur la base de données MNIST qui sera utilisée par les classeurs (Réseau Neuronal Convolutif, Support Vector Machine, K-Nearest Neighbors, Bagging, Les arbres de décision, La forêt aléatoire... etc).

Nous avons remarqué que L'algorithme CNN est efficace dans le processus de reconnaissance, atteignant la précision 96 % et les résultats obtenus ont été très convaincants et satisfaisants. Et donc nous avons opté la méthode de réseau neuronal convolutif pour décider la classe d'appartenance pour chaque chiffre.

Enfin, nous avons mené une ensemble d'expériences sur deux des modèles, et toutes les expériences réalisées ont montré des résultats très encourageants. Pour le premier Modèle nous avons obtenus 95.81 % et pour le second Modèle nous obtenus 93.94 %.

Ce travail est organisé en quatre chapitres:

Le premier chapitre peut être considéré comme une introduction à la reconnaissance de l'écriture manuscrite. Nous allons donner un aperçu sur les différentes étapes constituant un système de reconnaissance d'écriture manuscrite.

Le deuxième chapitre est consacré à la présentation des différentes méthodes de segmentation et les différents algorithmes d'apprentissage automatique et d'apprentissage profondeur. En plus, nous allons essayer de présenter quelques travaux antérieurs.

Le troisième chapitre présente l'architecture proposée de notre système et son fonctionnement. De plus, nous expliquons chaque étape de cette architecture en détails.

Le quatrième chapitre présente (l'environnement de développement du système de travail) et les différents tests réalisées, ainsi que les résultats obtenus.

Dans la conclusion, nous rappelons l'objectif du travail, les résultats obtenus ainsi que nous présentons quelque perspectives pour ce travail.

Reconnaissance de l'écriture Manuscrite

Introduction

La reconnaissance de l'écriture manuscrite est un traitement informatique qui a pour but de convertir une image d'un texte imprimé ou manuscrit en texte dans un fichier au format numérique. Dans ce chapitre, nous présentons les différentes phases de reconnaissance des chiffres manuscrits en mettant l'accent sur toutes les phases du système de reconnaissance.

1.1 Différents aspects de reconnaissance

La première étape du système de reconnaissance consiste à convertir l'écriture manuscrite en quantités numériques adaptées au système de traitement avec un minimum de dégradations possibles. Tout dépend du type de dispositifs d'acquisition et de données à traiter.

Nous allons présenter quelques aspects des systèmes de reconnaissance manuscrite tels que la différence entre la reconnaissance et la production et les différents types de systèmes de reconnaissance de l'écriture manuscrite selon le mode d'acquisition (Les systèmes de reconnaissance en ligne, Les systèmes de reconnaissance hors-ligne).[\[24, 1\]](#)

1.1.1 Production et reconnaissance

La reconnaissance consiste à transformer une image d'un document en une représentation sous forme d'un fichier informatique. On peut aussi dire que la reconnaissance est le processus inverse de la production. La figure [1.1](#) illustre les deux processus. Le document est saisi à l'aide d'un scanner ou caméra de manière à obtenir une image numérique. Souvent, cette image contient des bruits. Cette image est ensuite prétraitée pour fournir une image épurée avec une représentation plus claire.[\[50\]](#)

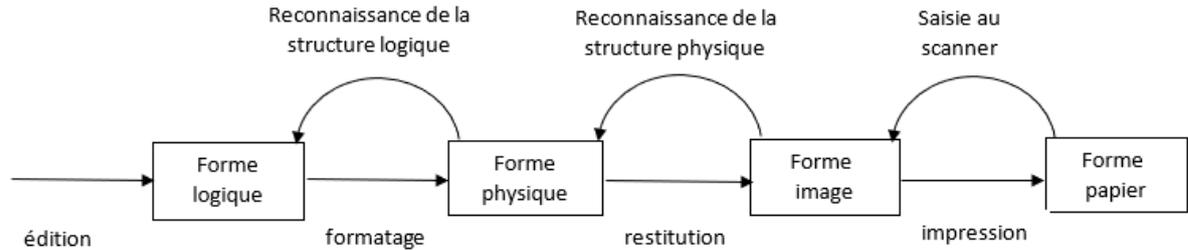


Figure 1.1: Processus de production et de reconnaissance de documents.[50]

1.1.2 Reconnaissance en-ligne / hors-ligne

Dans la reconnaissance de l'écriture manuscrite en-ligne, l'acquisition de l'écriture se fait au cours de l'écriture, alors que dans la reconnaissance de l'écriture hors-ligne, l'acquisition se fait à partir d'un document papier numérisé par un scanner ou par caméra.

1.1.2.1 Reconnaissance en-ligne

La reconnaissance en-ligne, l'acquisition en temps réel à l'aide d'un stylo ou une tablette électronique.

Les systèmes de reconnaissance de l'écriture en-ligne sont généralement plus faciles et plus efficaces que les systèmes de reconnaissance hors-ligne (Voir la figure 1.2).

la reconnaissance en-ligne est principalement employée aux domaines de l'informatique industrielle et de sécurité tels que la certification d'auteur et la vérification de signature.[24]

1.1.2.2 Reconnaissance hors-ligne

La reconnaissance hors-ligne, l'acquisition à l'aide d'un scanner ou d'un caméra pour acquérir une image binaire ou en niveaux de gris. Elle contient souvent des effets secondaires comme le bruit et la perte d'informations temporelles. Donc La reconnaissance de l'écriture hors-ligne est plus complexe (que celle qui est en-ligne) (Voir la figure 1.2).

La reconnaissance hors-ligne est consacrée au traitement des chèques bancaires, le tri des courriers postaux, la lecture des formulaires commerciaux... etc.[24]

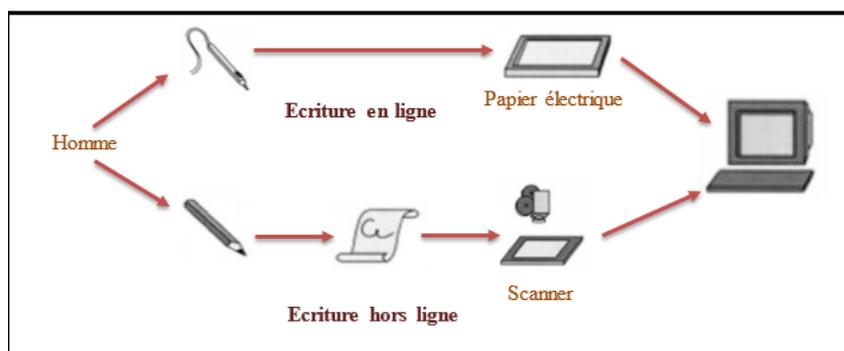


Figure 1.2: Différence entre les deux types de reconnaissance (online , offline).[45]

1.2 Architecture générale d'un system de reconnaissance

Généralement, la reconnaissance de l'écriture manuscrite est réalisée en cinq étapes : acquisition, prétraitement, segmentation, extraction des caractéristiques, classification. voir La figure 1.3 illustre le schéma général d'un système de reconnaissance des chiffres.

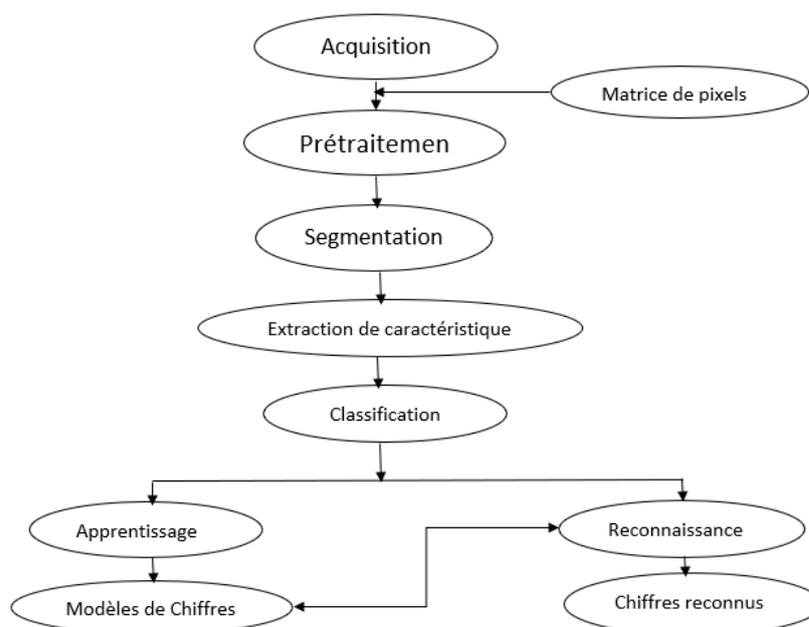


Figure 1.3: Architecture général d'un système de reconnaissance des chiffres.[50]

1.2.1 Phase d'acquisition

La phase d'acquisition consiste à capter l'image d'un chiffre au moyen des capteurs physiques telsque scanner, caméra ...etc et de la convertir en élévations numériques adaptés au système de traitement informatisés avec un minimum de dégradation possible.

Durant cette phase, malgré la bonne qualité des systèmes d'acquisition, des bruits parasites peuvent apparaitre et causent une hétérogénéité du fond ceci est dû à la nature de la texture, l'aire de travail et de son éclairage.[32]

1.2.2 Phase de pré-traitement

Lorsque l'acquisition est effectuée, la plupart des systèmes comportent une étape de prétraitement. Le prétraitement a pour but de réduire le bruit superposé aux données et ne garder, autant que possible, que les informations pertinentes. Le bruit peut être dû au dispositif d'acquisition, aux conditions d'acquisition (éclairage, mise incorrecte du document... etc).

les opérations de prétraitements généralement utilisées sont: la binarisation, le lissage (bruit), la normalisation (normalisation de taille, normalisation de Redressement), squelettisation.[24]

1.2.2.1 La binarisation

Dans un système de reconnaissance d'écriture est une opération qui permet le passage d'une image de niveaux de gris à une image binaire (0 et 1), plus simple à traiter. la figure 1.4 illustre de la binarisation.

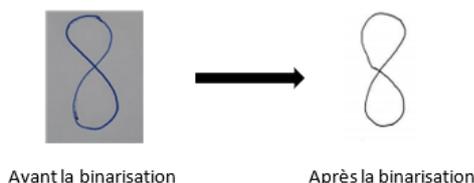


Figure 1.4: Exemple de la binarisation.

Selon la méthode de calcul du seuil de binarisation, on distingue deux types de binarisation : le seuillage globale et le seuillage adaptatif.[1]

1.2.2.2 Lissage

Le lissage est une transformation locale examiner qui consiste à le voisinage d'un pixel et de lui attribuer la valeur 1 si le nombre de pixel noir dans cette zone est supérieur à un seuil. La plupart du temps, l'image contient du bruit peut être dû au dispositif d'acquisition et souvent à la qualité du document, conduisant soit à des absences de points ou des excroissances. Les techniques de lissage permettent de résoudre ces problèmes.[24]

1.2.2.3 La normalisation

une étape de normalisation est nécessaire, car le problème majeur pour la segmentation est la taille variable des images et l'inclinaison des lignes.

Normalisation de taille :

Cette étape permet de changer les images des chiffres manuscrits à des tailles standard. la normalisation en général déforme les images mais elle reste indispensable pour certains types de systèmes et de méthodes de reconnaissance qui sont sensibles aux petites variations dans la taille et la position comme c'est le cas dans les templates matching et

les méthodes de corrélation.[9] la figure 1.5 illustre la normalisation de taille.

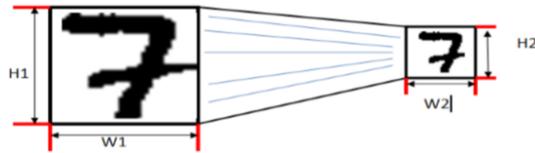


Figure 1.5: Exemple d'un chiffre manuscrit normalisé.[2]

Normalisation de l'inclinaison (Redressement) :

Le redressement, est une technique de correction de l'inclinaison qui à été développée pour corriger le défaut de positionnement du papier dans le scanner et aussi il dresser l'inclinaison interne de certaines parties du document.

Plusieurs algorithmes de reconnaissance ne donne pas de bons résultat si le document n'est pas bien ajusté. Le redressement se fait selon deux niveaux.[38]

- Le redressement de la ligne de base : L'idée est de rendre horizontaux les chiffre à l'aide d'une transformation géométrique de type rotation isométrique des points de l'image.
- Le redressement des écritures penchées : Ce redressement facilite la segmentation préalable des chiffres. L'idée est de trouver l'angle moyen d'inclinaison puis de faire une transformation géométrique de type cisaillement de l'image.

la figure 1.6 illustre de deux types de redressement.

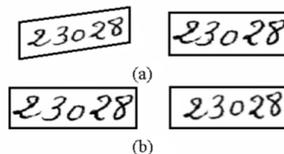


Figure 1.6: Exemple de deux types de redressement.[24]

1.2.2.4 Squelettisation (Thinning)

l'opération de squelettisation est appliquée pour simplifier l'image du caractère en une image a ligne. Le principe est de trouver un axe médian possédant une égalité de distance par rapport aux pixels de frontières qui les entourent.

Elle est définie comme étant l'équivalent de la forme, mais avec une épaisseur très réduite qui est égale à 1 pixel. La squelettisation ou amincissement est une opération essentielle dans plusieurs systèmes de reconnaissance d'écriture manuscrite.

Les algorithmes de squelettisation se basent sur des méthodes itératives. Le processus s'effectue par passes successives pour déterminer si un tel ou tel pixel est essentiel pour le garder ou non dans le tracé.[1, 4] la figure 1.7 illustre de squelettisation.

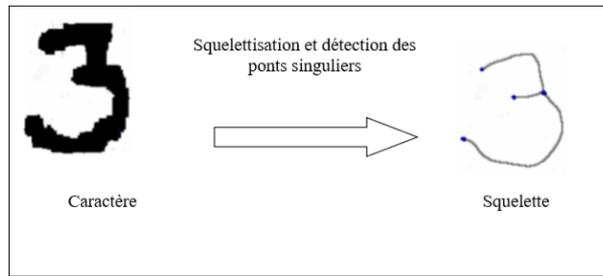


Figure 1.7: Exemple de squelettisation.[2]

1.2.3 Phase de Segmentation

la segmentation est l'étape la plus importante et la plus difficile dans le processus de reconnaissance. l'étape la segmentation est de generer un decoupage de l'image (chaîne des chiffres) en plusieurs images chacune un chiffre isolé susceptibles d'être reconnus par le classifieur.[32] la figure 1.8 illustre phase de segmentation.

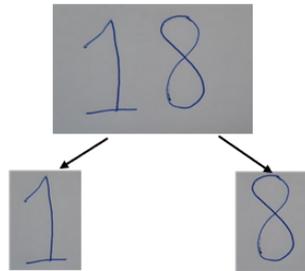


Figure 1.8: Exemple de Segmentation.

1.2.4 Phase d'extraction de caractéristiques

Dans un système de reconnaissance des chiffres manuscrits, La reconnaissance d'un chiffre passe d'abord par l'analyse de sa forme et l'extraction de ses traits caracteristiques.

L'objectif de l'extraction des caracteristiques est en générant en un vecteur de primitives de taille fixe pour un image de chiffre isolé. la figure 1.9 illustre d'extraction de caractéristiques.

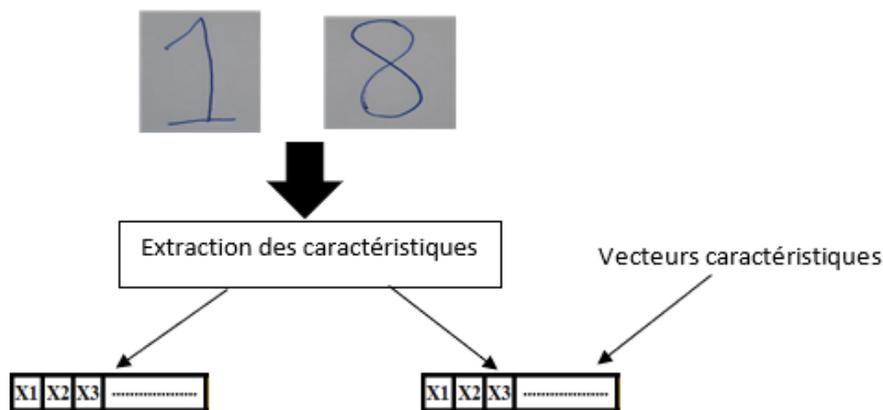


Figure 1.9: Exemple d'extraction de caractéristiques.

Les types de caractéristiques peuvent être classés en quatre catégories principales de primitives : structurelles, statistiques, globales et topologiques ou Métriques.[26, 43]

1.2.4.1 Les primitives structurelles

Les primitives structurelles basées sur une représentation linéaire du caractère cette primitive est extraite principalement à partir de représentation squelette ou contour de la forme et pas de l'image brute. Ainsi, parmi ces caractéristiques, il s'agit principalement des segments de droite, des arcs, boucles et concavités, des pentes, la hauteur et la largeur du caractère ... etc.[26, 43]

1.2.4.2 Les Primitives statistiques

Les primitives statistiques donnent des informations concernant la distribution des pixels dans l'image du caractère ou chiffres, Les primitives statistiques décrivent une forme en termes d'un ensemble de mesures extraites à partir de cette forme. Les caractéristiques utilisées pour la reconnaissance des chiffres manuscrits sont :

- Le zonage (zoning) :consiste à répartir l'image en région ou zones. L'image source est une image binaire.[24]
- L'histogramme : c'est un moyen de représentation du nombre de pixels sur chaque ligne ou colonne de l'image.[1]



Figure 1.10: Division d'une image en zone.

1.2.4.3 Les primitives globales

La caractéristique d'une primitive globale est de dépendre de la totalité des pixels d'une image et d'une transformation globale de l'image. ces primitives sont donc dérivées de la distribution des pixels, elles produisent trois types de caractéristiques telles que : moments invariants, projections et profils.

Une des transformations les plus simples est celle qui représente le squelette ou le contour d'un caractère sous forme d'une chaîne de codes de directions (code de Freeman). Il existe de nombreuses méthodes de transformation, on peut citer les transformations de Hough et Fourier et les ondelettes.[1, 26]

1.2.4.4 Primitives topologiques ou métriques

Ce type est basé sur des densités de pixels. Il s'agit d'effectuer une mesure sur l'échantillon au moyen d'une métrique. Parmi les mesures on trouve : Compter le nombre de trous, Evaluer les concavités, Mesurer des pentes, des courbures et évaluer des orientations principales, Mesurer la longueur, surfaces, les périmètres et l'épaisseur des traits, Détecter les croisements et les jonctions des traits, déterminer le rectangle délimitant l'échantillon , ou le polygone convexe. . . etc.

Toutes ces mesures peuvent être intégrées dans un seul vecteur de caractéristiques pour la reconnaissance du chiffre manuscrit.[26, 12]

1.2.5 Phase de Classification

Après la segmentation de caractères et l'extraction des attributs caractéristiques, nous passons maintenant à la phase suivante, qui a pour tâche principale de classer les différents segments selon leurs caractéristiques.[9] Cette phase passe par deux étapes :

- Apprentissage
- Décision /reconnaissance

1.2.5.1 Etape d'apprentissage

Les données d'apprentissage sont les données utilisées pour construire un classificateur capable de reconnaître des formes inconnues pour caractériser les classes.[19] Il existe deux types d'apprentissage :

- l'apprentissage supervisé : la sortie correcte doit être fournie à l'avance.
- l'apprentissage non supervisé : la sortie correcte n'est pas exigée à l'avance, elle résulte après une étape d'apprentissage.

1.2.5.2 Etape de reconnaissance et décision

Le résultat de la décision considéré comme un "avis" indiquant si la forme appartient ou non aux modèles d'apprentissage.[35]

La reconnaissance peut produire les réponses suivant:

- Un succès si la réponse est un seul modèle correspond à la description de la forme du caractère.
- Si la réponse est multiple, cela peut conduire à une confusion (plusieurs modèles correspondent à la description).
- peut conduire à un rejet de la forme si aucun modèle ne correspond à sa description.

il est possible que le classificateur soit incapable de reconnaître d'autres formes inconnues, si toutes les données sont employées pour l'apprentissage et le test en même temps. Alors est important d'avoir trois ensembles de données pour améliorer la généralisation d'un classificateur : le premier pour l'apprentissage, le second pour la validation et le dernier pour le test.

1.3 Les approches de reconnaissance

Les approches de reconnaissance peuvent être regroupées en quatre groupes principaux: approche statistique, approche structurelle, approche stochastique et l'approche hybride.

1.3.1 Approche statistique

Dans cette approche La reconnaissance est basée sur l'étude statistique des mesures que l'on effectue sur les formes à reconnaître. Mais effectue un apprentissage correct des lois de probabilité des différentes classes elle à besoin d'un nombre élevé d'exemples.

L'étude des classes de caractérisation statistique et de distribution dans un espace métrique permet de prendre une décision du type : plus forte probabilité d'appartenance à une classe.

Cette approche bénéficie des méthodes d'apprentissage automatique qui s'appuient sur des bases théoriques connues telles que :les réseaux de neurones, les chaînes de Markov ,méthode des k plus proches voisins... etc).[35, 12, 37]

1.3.2 Approche structurelle

Les méthodes structurelles reposent sur l'extraction de primitives en prenant compte de l'information structurelle (la structure physique des caractères). De manière générale, les approches syntaxiques ou structurelles permettent la description de formes complexes à partir de formes élémentaires. La différence que l'on peut distinguer entre les méthodes statistiques et ces méthodes est que ces caractéristiques sont des formes élémentaires (les primitives sont de type topologiques) et non pas des mesures.

On distingue plusieurs techniques telles que les structures de graphes, le calcul de distance d'édition entre deux chaînes et la programmation dynamique la comparaison de chaînes, les méthodes de tests, les structures syntaxiques... etc.[26, 43]

1.3.3 Approche stochastique

Contrairement aux méthodes précédemment décrites, l'approche stochastique utilise un modèle pour la reconnaissance, prenant en compte la grande variabilité de la forme. Dans ce type d'approche, les modèles sont souvent discrets et de nombreux travaux reposent sur la théorie des champs de Markov et l'estimation bayésienne. Les champs de Markov permettent de ramener des propriétés globales à des contraintes locale. Le modèle décrit ces états à l'aide de probabilités de transitions d'états et de probabilités d'observation par état. La comparaison consiste à chercher dans ce graphe d'état, le chemin de probabilité forte correspondant à une suite d'éléments observés dans la chaîne d'entrée.[19, 12]

1.3.4 Approche hybride

Pour obtenir de meilleures performances en reconnaissance, la direction actuelle est de créer un système hybride qui utilise des différents types d'entités et qui fusionne plusieurs classificateur en couches. pour surmonter les faiblesses de chaque approche et obtenir des résultats plus précis plus meilleure que les résultats qui auraient été obtenus si l'application de chaque approche séparément, comme les approches qui ont été fusionnés pour former celui intégré. Un exemple d'une approche hybride qui combine les approches statistiques et structurelles.[24]

1.4 Domain d'application

Aujourd'hui, il existe plusieurs domaines dans lesquels la reconnaissance des chiffres manuscrit par exemple :

- Domaine bancaire : Pour l'authentification des chèques par les banques.
- Lecture des adresses postales : la lecture des codes postaux manuscrits associés à la lecture des noms de villes a permis d'atteindre le développement des machines de tri automatique de courrier.
- La police et la sécurité : pour la reconnaissance des numéros minéralogiques pour le contrôle routier ... etc.

Conclusion

Dans ce chapitre, nous avons présenté les différentes phases d'un système de reconnaissance des chiffres manuscrits, Ainsi nous avons démontré les problèmes liés à la complexité de la reconnaissance.

Nous présentons dans le chapitre suivant, techniques de segmentation et d'apprentissage automatique et d'apprentissage profond.

Segmentation et reconnaissance des chiffres

Introduction

Dans ce chapitre, nous présentons les techniques ou méthodes utilisées pour segmenter une série des chiffres manuscrit, en un ensemble de segments de base. Ainsi que, nous présentons les différentes algorithmes d'apprentissage automatique et l'apprentissage profondeur et quelque travaux antérieur.

2.1 Problématique posée par l'écriture du cursif

L'image des chaîne de chiffres cursif est représentée par un signal bidimensionnel dans lequel, aucune information d'ordonnancement n'est présente.

En regardant la figure 2.1 on constate qu'il est pas possible de déterminer à priori si le point B se trouve ou non dans le chiffre antérieure au point A.

La seule façon de restituer l'ordre cohérent du tracé est de segmenter à sequence chiffres en chiffres isolés. Or, dans ce type d'écriture, la localisation précise du début et de la fin d'une chiffre s'avère très difficile à réaliser, voire impossible. C'est pourquoi, le problème de la segmentation se trouve au cours de la reconnaissance hors-ligne de chaîne des chiffres cursifs.[15]

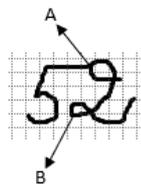


Figure 2.1: Exemple chaîne de chiffre cursif "52".

2.2 Principe général de la segmentation

L'étape de segmentation est l'une des étapes les plus importantes du système de reconnaissance des nombres manuscrit. Où nous extrayons des différentes zones ou segments d'images. parce qu'une bonne segmentation va déterminer les bons segments et par conséquent un bon taux de reconnaissance, à l'inverse, une mauvaise segmentation quant à elle va entraîner une chute du taux de reconnaissance. Là-bas plusieurs facteurs tels que la variabilité du style et de l'outil d'écriture augmente la complexité de segmentation.[50]

Généralement, nous pouvons distinguer quatre niveaux de segmentation, comme suit :

- Segmentation de page.
- Segmentation en lignes.
- Segmentation en séquence de chiffres.
- Segmentation en chiffres isolés.

La figure 2.2 illustre le passage entre ces étapes.

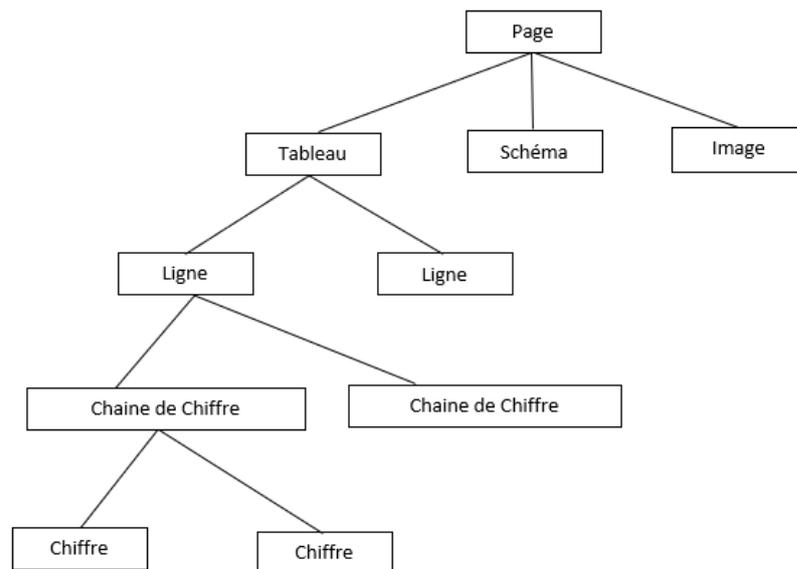


Figure 2.2: Illustration du processus de segmentation.

2.2.1 Segmentation de page

Pour reconnaître un chiffre, alors la segmentation de la page est très importante. Il est généralement lié à retrouver la structure physique du document en délimitant les différentes parties homogènes (image, tableau, séquence de chiffre ... etc).[28] La figure 2.3 illustre cette étape.

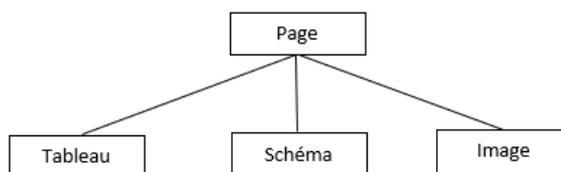


Figure 2.3: Détection des zones d'une page de document.

2.2.2 Segmentation en lignes

Cette étape consiste à séparer les différentes lignes de toute structure linéaire pouvant contenir plus d'une ligne pour extraire la séquence de chiffre puis les chiffres.

il existe certaines méthodes utilisées pour localiser les lignes, le plus utilisé c'est la projection horizontale.[28] La figure 2.4 illustre cette étape.

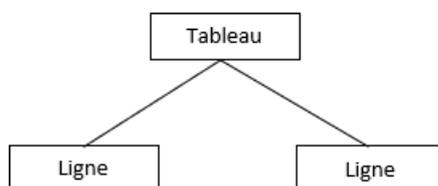


Figure 2.4: Segmentation du tableau en Lignes.

2.2.3 Segmentation en séquence de chiffres

La segmentation de ligne en séquence de chiffres est réalisée en déterminant l'histogramme des projections verticales des lignes pour détecter les espaces entre les sequence de chiffre et pouvoir les séparer.[28] La figure 2.5 illustre cette étape.

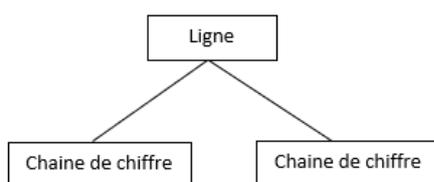


Figure 2.5: Segmentation de ligne en séquence de chiffres.

2.2.4 Segmentation en chiffres isolés

La segmentation de séquence de chiffres en chiffres isolés est l'étape le plus délicate dans tout le processus d'un système de reconnaissance d'écriture.

La séparation de chiffres est effectuée en analysant l'image des chiffres séquentiels et en la segmentant en sous-images de chiffres individuels .[28] La figure 2.6 illustre cette étape.

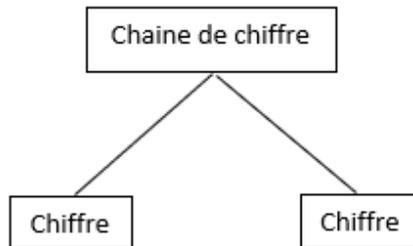


Figure 2.6: Segmentation de séquence de chiffres en chiffres.

2.3 Approches de segmentation

D'une manière générale, il existe deux approches de segmentation : l'approche implicite et l'approche explicite.[32]

2.3.1 Segmentation explicite

l'approche explicite (appelée aussi discrète) consiste à trouver le meilleur chemin de séparation de séquence de chiffres en chiffres isolés. Les chemins de segmentation sont généralement obtenus par des points caractéristiques issus d'une analyse des contours de la forme, du squelette ou d'un amincissement du fond, d'une analyse en deux dimensions du tracé, ou d'une combinaison analyse des contours / amincissement du fond.[32, 35, 12]

2.3.2 Segmentation implicite

Pour contourner le problème difficile du choix des points de segmentation, les approches à segmentation implicite (ou continues) considèrent tous les points du tracé comme points de segmentation potentiels. Dans cette approche, La segmentation et la reconnaissance sont réalisées conjointement. Il s'agit de méthodes à fenêtres glissantes qui parcourent la séquence de chiffres à l'aide d'une fenêtre de taille fixe. Contrairement à la segmentation explicite, il n'y a pas de pré-segmentation de chiffres.[35, 12]

2.4 Méthodes de segmentation

Dans cette section, nous présentons des méthodes de segmentation, où la segmentation de l'écriture est une opération très délicate et coûteuse, leur résultat influe considérablement sur le rendement des prochaines phases.

2.4.1 Segmentation à partir du squelette

Dans ce cas, le système commence par la création d'une image contenant le squelette du texte à traiter. Ensuite, à travers ce squelette, des points de connexion sont identifiés entre les chiffres. Le taux d'erreur dans cette approche est de 10 %.[51] la figure 2.7 illustre le principe du squelette.

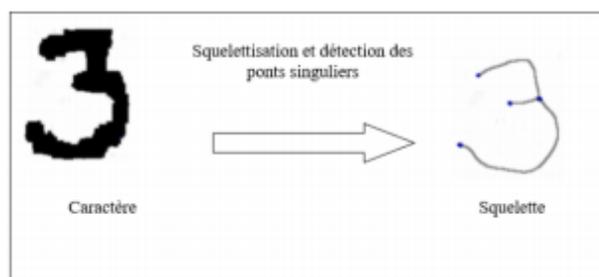


Figure 2.7: Exemple de squelettes de chiffre.

2.4.2 Segmentation basée sur les contours

Cette approche permet de séparer des chiffres en utilisant des informations sur le contour. L'extraction des contours se fait sur des images binaires en utilisant la morphologie mathématique. Chaque chiffres est isolé en utilisant une distance pour résoudre les problèmes de chevauchement entre les chiffres.[23] La Figure 2.8 illustre le principe d'extraction des contours.

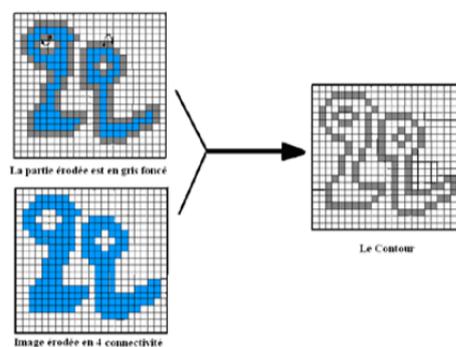


Figure 2.8: Extraction des contours.

2.4.3 Segmentation basée sur L'HPV

Cette approche est appliquée à l'image binaire, où l'emplacement des nombres est déterminé en trouvant les plages correspondantes d'un nombre de pixels noirs non nuls. La séparation des chiffres contigus consiste à détecter les passages par zéro de l'histogramme de projection. On peut alors déterminer l'emplacement de chaque chiffre de l'image.[23] la figure 2.9 illustre le principe L'HPV.

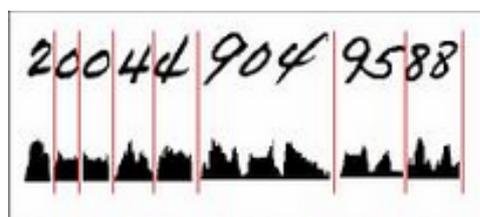


Figure 2.9: Segmentation des chiffres par l'histogramme de projection verticale.

2.4.4 Segmentation basée sur les fenêtres glissantes

Cette technique est simple, robuste au bruit et indépendante des connexités. Néanmoins, la détermination de la taille de la fenêtre appropriée n'est pas facile. Le principe est d'utiliser une fenêtre mobile de largeur variable en découpant l'image en bandes verticales. [21] la figure 2.10 illustre le principe fenêtres glissantes.



Figure 2.10: Segmentation à base de fenêtre glissante.

2.5 l'apprentissage automatique

Récemment, des méthodologies d'apprentissage automatique ont été adoptées pour améliorer la précision de la reconnaissance des chiffres manuscrits. Ci-dessous, nous expliquons les algorithmes de d'apprentissage automatique les plus importants.

2.5.1 Définition

L'apprentissage automatique (ML) est une branche de l'intelligence artificielle (IA) qui consiste à programmer des algorithmes permettant d'apprendre automatiquement à partir des données et d'expériences passées.

Le principe de base de l'apprentissage automatique est de créer des algorithmes capables de recevoir des données d'entrée et d'utiliser une analyse statistique pour prédire une sortie tout en les mettant à jour à mesure que de nouvelles données deviennent disponibles.

Il s'agit, très schématiquement, de concevoir des algorithmes est capable d'apprendre à partir d'un nombre important d'exemples et d'adapter ses résultats en fonction de nouvelles données.[48] la figure 2.11 illustre L'algorithme ML.



Figure 2.11: L'algorithme ML.

2.5.2 Les différents types d'apprentissage automatique

Il existe plusieurs façons d'apprendre automatiquement à partir des données dépendamment des problèmes à résoudre et des données disponibles.

2.5.2.1 L'apprentissage supervisé

L'apprentissage supervisé, dans le contexte de l'intelligence artificielle, est la méthode d'apprentissage la plus utilisée en Machine Learning et en Deep Learning.

L'apprentissage supervisé consiste en des variables d'entrée (X :input), une variable de sortie (Y :output), et un algorithme (F :modèle) utilisé pour prédire le résultat Y en se basant sur des données X . $Y = F(X)$

L'objectif de l'apprentissage supervisé est de développer un modèle (classifieur) capable de classer de nouvelles instances (ne faisant pas partie de X) avec un minimum d'erreur.[6] la figure 2.12 illustre apprentissage supervisé.

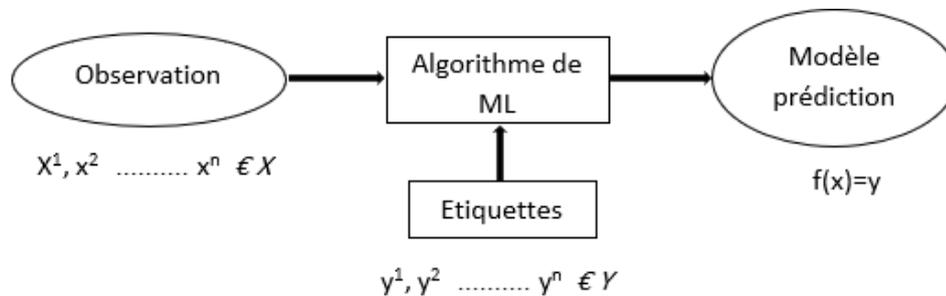


Figure 2.12: Apprentissage supervisé.

2.5.2.2 l'apprentissage non supervisé

l'apprentissage non supervisé à contrario de l'apprentissage supervisé celui-ci correspond au fait de n'utiliser que des données d'entrée X et aucune variable de sortie Y correspondante. L'apprentissage non supervisé vise à caractériser la distribution des données, et les relations entre les variables, sans discriminer entre les variables observées et les variables à prédire.[6] la figure 2.13 illustre apprentissage non supervisé.

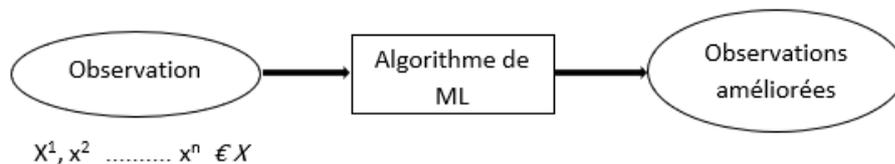


Figure 2.13: Apprentissage non supervisé.

2.5.3 Classifications

La classification est un processus de recherche d'une fonction qui aide à diviser l'ensemble de données en classes en fonction de différents paramètres. Dans la classification, un programme informatique est formé sur l'ensemble de données d'entraînement et sur la base de cet entraînement, il catégorise les données dans différentes classes.

La tâche de l'algorithme de classification est de trouver la fonction de mappage pour mapper l'entrée X à la sortie discrète Y .

Exemple : Le meilleur exemple pour comprendre le problème de classification est la détection de spam par e-mail. Le modèle est formé sur la base de millions d'e-mails sur différents paramètres, et chaque fois qu'il reçoit un nouvel e-mail, il identifie si l'e-mail est du spam ou non. Si l'e-mail est du spam, il est déplacé vers le dossier Spam.

2.5.4 Régression

La régression est un processus de recherche des corrélations entre les variables dépendantes et indépendantes. Il aide à prédire les variables continues telles que la prédiction des tendances du marché, la prédiction des prix des logements... etc.

La tâche de l'algorithme de régression est de trouver la fonction de mappage pour mapper la variable d'entrée X à la variable de sortie continue Y .

Exemple : Supposons que nous voulions faire des prévisions météorologiques, donc pour cela, nous utiliserons l'algorithme de régression. Dans la prévision météorologique, le modèle est formé sur les données passées, et une fois la formation terminée, il peut facilement prédire la météo pour les jours futurs.

2.5.5 Algorithmes d'apprentissage automatique

Plusieurs algorithmes ont été développés dans les systèmes de la reconnaissance de chiffre manuscrit afin de réaliser la tâche de classification. Dans cette section, nous expliquons les méthodes de classification les plus utilisées.

2.5.5.1 Les Machines à Vecteur Support (SVM)

Les SVM sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de classification binaire et de la régression.

Les SVM reposent sur deux idées, la notion de la marge maximale et la notion de la fonction noyau. Le classificateur SVM est un algorithme qui maximise la marge entre les classes du problème à résoudre et réduit au minimum l'erreur de classification. L'objectif de la marge maximale est de faire séparer deux classes par un hyperplan de telle sorte que la distance par rapport aux vecteurs supports soit maximale. Dans la tâche de classification, un SVM construit l'hyperplan optimal de séparation des attributs caractéristiques dans un espace de haute dimension. Le calcul de cet hyperplan est fondé

sur la maximisation de la marge entre les exemples d'apprentissages les plus proches qui appartiennent à différentes classes.[30, 27, 1]

la figure 2.14 illustre le principe SVM.

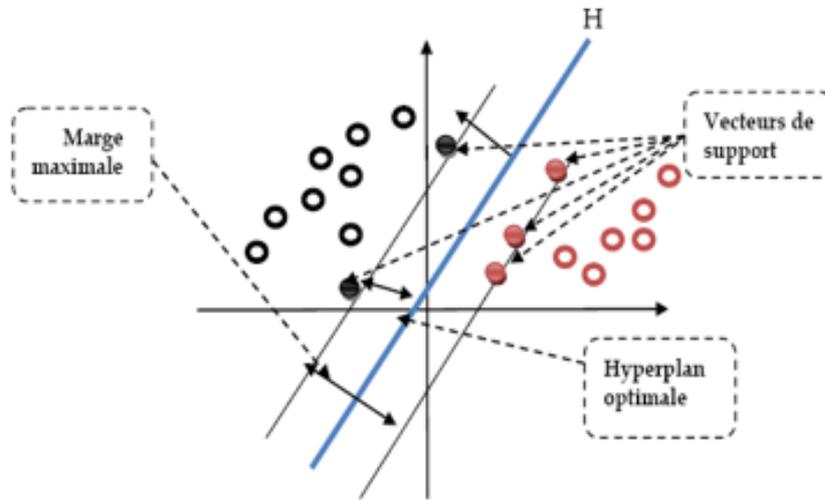


Figure 2.14: Machines à Vecteurs de Support SVM.

2.5.5.2 Le classifieur du K plus proches voisins

L'algorithme K plus proches voisins (K Nearest Neighbors) c'est un classificateur simple basé sur le calcul de distance entre les exemples d'apprentissage et les exemples de tests.

Pour trouver la classe d'un nouveau cas, cet algorithme se base sur le principe suivant : il cherche les k plus proches voisins de ce nouveau cas, ensuite, il choisit parmi les candidats trouvés le résultat le plus proche.[34, 38, 49] la figure 2.15 illustre le principe KPPV.

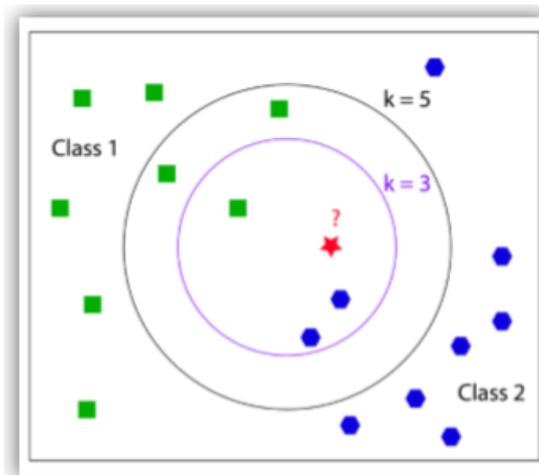


Figure 2.15: K plus proches voisins.

2.5.5.3 Réseau de neurones

Sous-ensemble de technologies de l'intelligence artificielle, un réseau de neurones peut être vu comme un assemblage de neurones connectés ensemble suivant une loi de connectivité que l'on définit. Le réseau peut être décrit par un graphe où les noeuds sont les neurones et les arêtes sont les connexions entre les neurones. La capacité de traitement de ce réseau est stockée sous forme de poids d'interconnexions obtenus par un processus d'apprentissage à partir d'un ensemble d'exemples d'apprentissage.[17] la figure 2.16 illustre Exemple de réseaux de neurones.

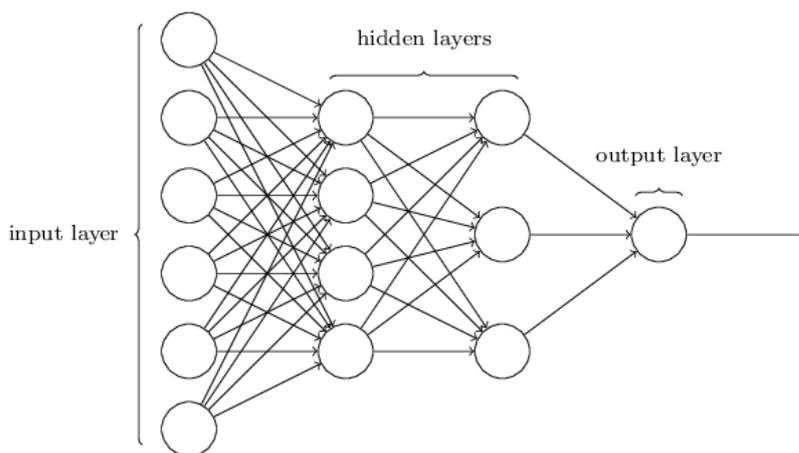


Figure 2.16: Exemple de réseaux de neurones.

2.5.5.4 Les arbres de décision

L'arbre de décision est un algorithme qui se base sur un modèle de graphe (les arbres) pour définir la décision finale. Son but est de créer un modèle qui prédit la valeur d'une variable-cible depuis la valeur de plusieurs variables d'entrée.

Un arbre de décision commence généralement par un nœud d'où découlent plusieurs résultats possibles. Chacun de ces résultats mène à d'autres nœuds, d'où émanent d'autres possibilités. L'objectif est de créer un modèle qui prédit la valeur d'une variable cible en apprenant les règles de décision.[39] la figure 2.17 illustre Exemple des arbres de décision.

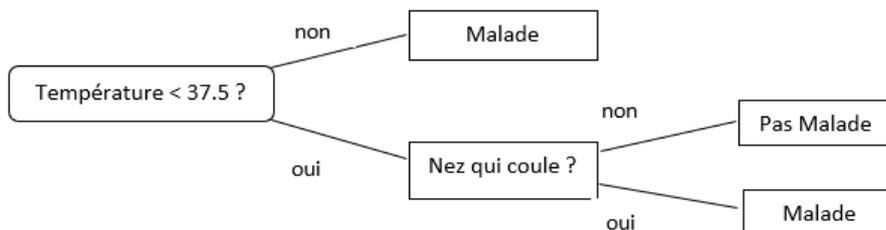


Figure 2.17: Exemple des arbres de décision.

2.5.5.5 La forêt aléatoire

Les forêts aléatoires est un ensemble d'arbres de décisions entraînés individuellement, légèrement différents les uns des autres. Pour prédire une nouvelle valeur, on effectue la classification pour chaque arbre de cette forêt. La forêt choisit la valeur ayant le plus de votes parmi tous ses arbres .[3] la figure 2.18 illustre Exemple de forêt aléatoire.

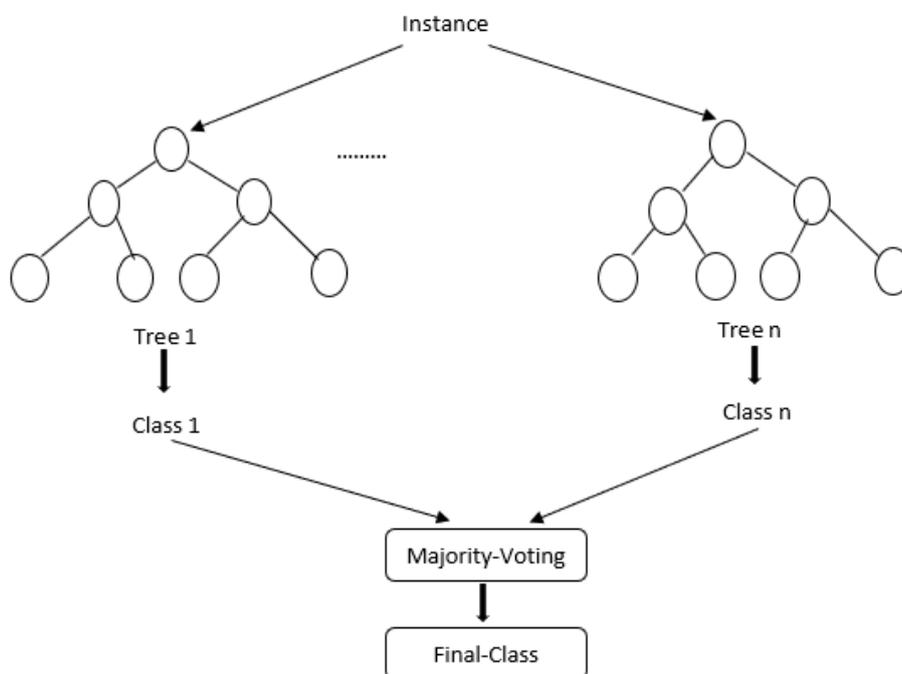


Figure 2.18: Exemple de forêt aléatoire.

2.5.5.6 Classification naïve bayésienne

Le Naïve Bayes Classifier (classifieur bayésien) cette classification appartient à la catégorie des classifieurs linéaires, qui permet de classer un ensemble d'observations selon des règles déterminées par l'algorithme lui-même.

Cet outil de classification doit dans un premier temps être entraîné sur un jeu de données d'apprentissage qui montre la classe attendue en fonction des entrées. Pendant la phase d'apprentissage, l'algorithme élabore ses règles de classification sur ce jeu de données, pour les appliquer dans un second temps à la classification d'un jeu de données de prédiction.[3]

2.5.5.7 Le bootstrap aggregating

L'agrégation bootstrap(également appelé bagging) est une technique qui entraîne de nombreux classificateurs sur différentes partitions des données d'apprentissage et utilise une combinaison des prédictions de tous ces classificateurs pour former la prédiction finale pour le vecteur d'entrée. la figure 2.19 illustre le principe bootstrap aggregating.

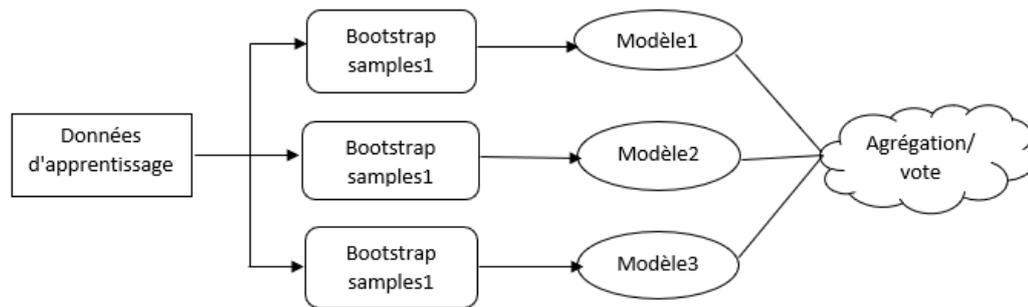


Figure 2.19: Bootstrap aggregating.

2.5.5.8 Régression linéaire

L'algorithme de régression linéaire est un algorithme d'apprentissage supervisé c'est-à-dire qu'à partir de la variable cible ou de la variable à expliquer Y , le modèle a pour but de faire une prédiction grâce à des variables dites explicatives X ou prédictives.

2.6 L'apprentissage profond

L'apprentissage profond «deep learning» est un type d'intelligence artificielle dérivé du machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même.

Le deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente.

DL a été appliquée à des domaines tels que la reconnaissance d'images, le traitement du langage naturel, la traduction automatique et de nombreux autres domaines.[\[31\]](#)

2.7 Réseau neuronal convolutif

Convolution Neural Networks (CNN) est un algorithme d'apprentissage en profondeur qui peut reconnaître et classer les caractéristiques des images pour la vision par ordinateur. Les CNN sont l'un des meilleurs algorithmes d'apprentissage pour comprendre le contenu des images et ont montré des performances exemplaires dans les tâches liées à la segmentation, la classification, la détection et l'extraction d'images. l'entrée considérée comme image (pixel). Le réseau se compose de plusieurs couches chaque couche qui ont de nombreux neurones.

La classification d'image consiste à sélectionner une catégorie d'image par exemple (cat-dog... etc).[\[42, 22\]](#)

2.7.1 Couches CNN

Le réseau de neurones à convolution se compose de plusieurs couches : couches de convolution, couches de pooling, couches relu et entièrement couches connectées.

2.7.1.1 Couche de convolution (CONV)

La convolution est la première couche à extraire des entités d'une image d'entrée. C'est une opération mathématique qui prend deux entrées telles qu'une matrice d'image et un filtre .

Le filtre est passé sur l'image, c'est une matrice qui fait généralement 3 x 3, 5 x 5. Chaque pixel est multiplié par la matrice. [14] la figure 2.20 illustre un exemple de l'opération de convolution.

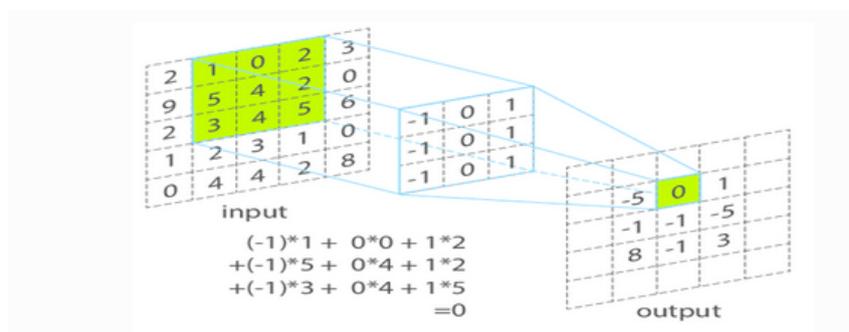


Figure 2.20: Illustration de l'opération de convolution.

2.7.1.2 Couche de pooling (POOL)

Le pooling est une opération simple qui consiste à remplacer un carré de pixels par une valeur unique. De cette manière, l'image diminue en taille et se retrouve simplifiée. cette opération est appliquée après une couche convolutionnelle.

il existe deux façons de mettre en commun: Max Pooling et Average Pooling.[10] la figure 2.21 illustre la différence entre Max, Average pooling.

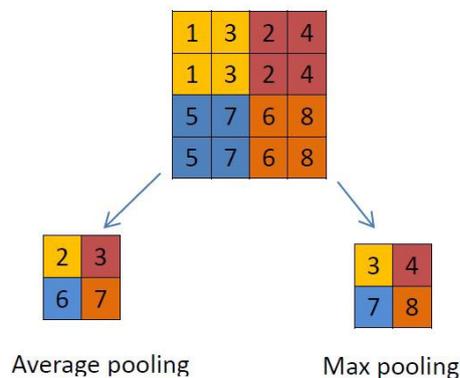


Figure 2.21: Différence entre Max, Average pooling.

2.7.1.3 Couche de correction Relu (Unités Rectifié Linéaire)

nous allons appliquer la fonction Relu ($F(x) = \max(0, x)$) pour forcer les neurones à retourner des valeurs positif car le pixel ne peut pas prendre une valeur négative ou une valeur supérieure à 255. En simple terme, la couche ReLU remplace tous les nombres négatifs par la valeur 0. Le résultat d'une couche ReLU est de la même taille que ce qui lui est passé en entrée.[11] la figure 2.22 illustre la fonction Relu.

15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23

➔

15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

Figure 2.22: Fonction Relu.

2.7.1.4 Couche entièrement connectée(FC)

Les neurones dans une couche entièrement connectée ont des connexions vers toutes les sorties de la couche précédente comme on le voit dans les réseaux réguliers de neurones.

Ce type de couche reçoit un vecteur en entrée et produit un nouveau vecteur en sortie. Pour cela, elle applique une fonction d'activation telle que softmax ou sigmoïde pour classer les sorties.[20] la figure 2.23 illustre Architecture d'un CNN.

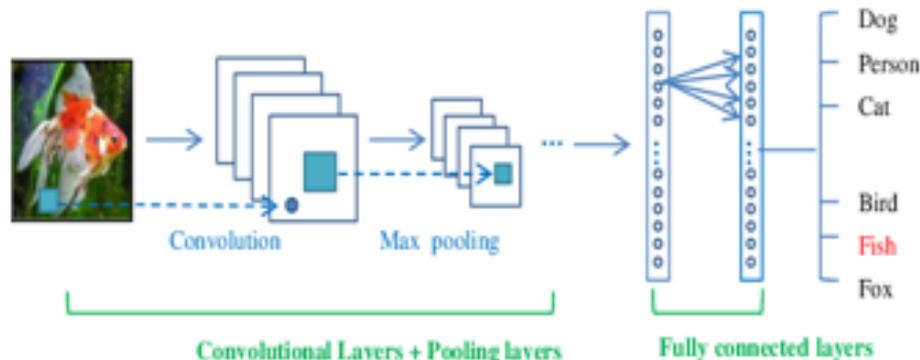


Figure 2.23: Architecture complète d'un CNN.

2.7.2 Fonctions d'activation

C'est une fonction qui s'ajoute à la fin de la sortie de tout réseau de neurones. Par exemple : Sigmoid , Tanh , SoftPlus.

- Sigmoid : $f(x) = \frac{1}{1+e^{-x}}$
- Tanh : $f(x) = \tanh(x)$
- SoftPlus : $f(x) = \log_e(1 + e^{-x})$

2.7.3 Variables de réseau de neurones

Ce qui suit est une explication des variables du réseau de neurones :

- Nombre d'époques (Epoch) : Lorsqu'un ensemble de données complet (images) traverse le réseau neuronal une fois et revient une fois, le processus est appelé une époque.
- Taille du lot (Batch_size) : Il s'agit de le nombre de image passé au modèle durant une époque.
- Nombre d'étape (Step) : Combien de tailles de lots ont été passées en une époque.
- Taux d'apprentissage (learning rate) : Taux de variation des poids à l'entraînement.

2.8 Quelques travaux antérieurs

Dans le cadre de notre projet nous avons fait l'étude de quelques travaux antérieurs (voir le tableau [2.1](#))

Tableau 2.1: Comparaison entre quelques travaux antérieurs.

Réf	Technique d'apprentissage	Dataset	Résumé	Evaluation
[46]	SVM	MNIST	Dans cet article, les auteurs présentent une technique combinée d'extraction de caractéristiques statiques et dynamiques pour reconnaître les chiffres manuscrits.	Acc=90%
[41]	CNN-SVM	MNIST	Dans cet article, les auteurs présentent un nouveau classificateur hybride CNN - SVM pour reconnaître les chiffres manuscrits.	Acc=89,81%
[33]	SVM	MNIST	Dans cet article, les auteurs présentent reconnaissance des chiffres manuscrits à l'aide de l'apprentissage automatique.	Acc=90,65%
	ANN			Acc=80.39%
[36]	Decision Tree	MNIST	Dans cet article, les auteurs présentent reconnaissance manuscrite de chiffres à l'aide d'un arbre décisionnel et de forêts aléatoires - Une approche comparative.	Acc=89.54%
	Random forest			Acc=91,40%
[7]	KNN	MNIST	Dans cet article, les auteurs présentent reconnaissance de chiffres manuscrits à l'aide de caractéristiques structurelles et statistiques et du classificateur de voisin le plus proche K.	Acc=90.42%
[25]	CNN	SVHN	Dans cet article, les auteurs présentent reconnaissance de nombres à plusieurs chiffres à partir d'images Street View à l'aide de réseaux neuronaux à convolution profonde.	Acc=90%
[13]	KNN	MNIST	Dans cet article, les auteurs présentent reconnaissance des chiffres manuscrits hors ligne à l'aide de l'apprentissage automatique.	Acc=91.7%
	NN			Acc=91.8%
[52]	CNN	MNIST	Dans cet article, les auteurs se concentrent sur Fusion de plusieurs classifieurs et extraction de caractéristiques CNN pour la reconnaissance des chiffres manuscrits.	Acc=90%
[5]	Decision Tree	kaggle	Dans cet article, les auteurs présentent reconnaissance des chiffres manuscrits avec classification par arbre de décision: une approche d'apprentissage automatique.	Acc=83.4%
[47]	SVM	MNIST	Dans cet article, les auteurs présentent, un algorithme pour la reconnaissance de chiffres manuscrits à l'aide d'histogrammes de projection et d'un classificateur SVM.	Acc=92.05%
[29]	NN	MNIST	Dans cet article, les auteurs présentent, reconnaissance des chiffres manuels hors ligne à l'aide du réseau neural.	Acc=91.2%
[18]	SVM	MNIST	Dans cet article, les auteurs présentent, une étude comparative entre les machines de vecteurs de support et les K-voisins les plus proches dans la reconnaissance des chiffres latins manuscrite.	Acc=91.13%
	KNN			Acc=83.50%
[8]	KNN	MNIST	Dans cet article, les auteurs présentent, reconnaissance de chiffres manuscrits à l'aide le classifieur du K plus proches voisins.	Acc=92.94%
[40]	Random forest	MNIST	Dans cet article, les auteurs présentent, reconnaissance des chiffres manuscrits à l'aide d'algorithmes d'apprentissage automatique.	Acc=92.84%
	KNN			Acc=92.60%
	SVM			Acc=94,75%

Conclusion

Dans ce chapitre, nous avons présenté l'étape la plus importante du système de reconnaissance de l'écriture manuscrite " la segmentation ", ainsi que les différents niveaux de segmentation et des méthodes utilisées dans ce processus.

Nous avons également expliqué les différentes techniques d'apprentissage automatique et l'apprentissage profond ainsi que expliqué quelques travaux antérieurs.

Dans le chapitre suivant, nous expliquerons la méthode proposée.

Architecture du système

Introduction

Dans les chapitres précédents, nous avons donné un aperçu sur les systèmes de reconnaissance des chiffres manuscrit, dans lesquels nous avons expliqué les différentes phases des systèmes de reconnaissance des chiffres. Après cela, nous avons présenté les différentes méthodes et techniques de segmentation des chiffres et les techniques d'apprentissage automatique et apprentissage en profondeur.

Dans ce chapitre, nous présentons notre méthode proposée, ainsi que les différentes étapes à suivre en détails.

3.1 Architecture générale du système

D'abord, le système fait l'acquisition de l'image à l'aide d'une caméra ou un téléphone portable. Une fois l'image à traiter est obtenue, elle est préparée pour un traitement futur en appliquant des processus de prétraitement : la binarisation , normalisation de taille ...etc.

Après avoir prétraité l'image, il cherche à localiser les cadres de chiffres (les rectangles existants) Ensuite, appliqué un ensemble d'opérations : inverser les couleurs , supprimer le bruit , augmenter la zone de l'objet et supprimer ou boucher les trous.

Après cela, une étape de segmentation est nécessaire, dans laquelle il applique une projection verticale et détecte le contour pour segmenter l'image (une série des chiffres) en petites images séparées.

Par la suite, en mode d'apprentissage, le modèle est entraîné à l'aide des images de chiffres de la base d'images MNIST.

Par contre, en mode de test ou de prédiction les segments d'images sont utilisées afin de reconnaître le numéro d'inscription ou la note. Dans ce cas, ces segments d'images viennent des copies d'examens. Finalement, la note est insérée dans le fichier

Excel cible en se basant sur le numéro d'inscription de l'étudiant (voir la figure 3.1).

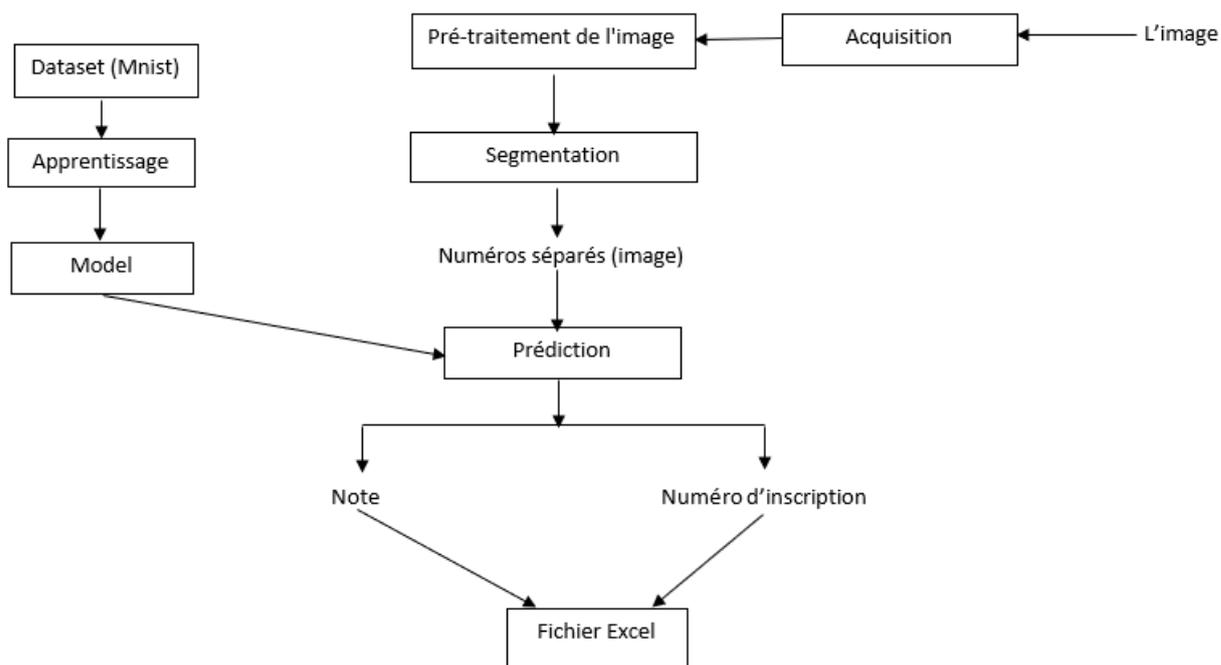


Figure 3.1: Architecture du système de reconnaissance de chiffres manuscrits.

3.2 Explication d'architecture

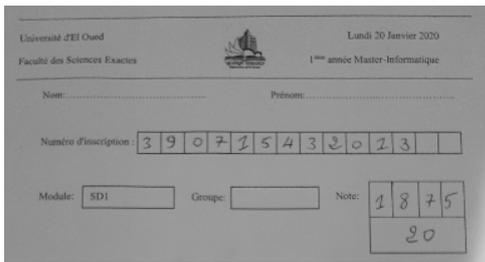
Dans cette section nous allons voir une description détaillée des phases de traitement, où nous allons voir les différents schémas et algorithmes pour bien comprendre le fonctionnement du système.

3.2.1 Acquisition

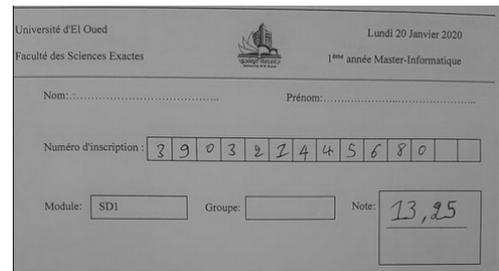
L'acquisition est l'étape dans laquelle nous obtenons l'image par une caméra ou du téléphone portable pour la traiter et l'extraire les informations voulus (voir la figure 3.2).

Nous avons deux modèles source possible :

- modèle 1: La note est séparer (chaque rectangle contient un chiffre séparé)
- modèle 2: La note n'est pas séparer (le rectangle contient une série des chiffres)



(a) modèle 1



(b) modèle 2

Figure 3.2: Exemples des images des deux modèles.

3.2.2 Prétraitement

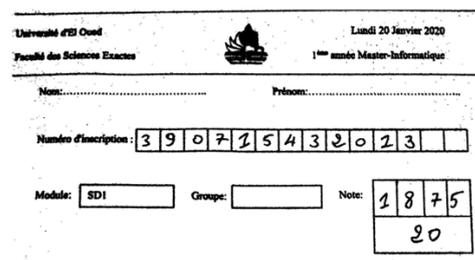
Le prétraitement est la phase suivante, elle consiste à préparer l'image aux traitements futur en appliquant des opérations dans le but de réduire le bruit superposé aux données et ne garder, autant que possible, que les informations pertinentes.

les opérations de prétraitements généralement utilisées, sont :

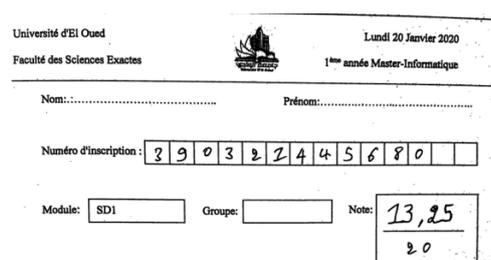
- La binarisation; et
- la normalisation de taille.

3.2.2.1 Binarisation

Elle consiste à transformer l'image brute en une image binaire (une image en noir et blanc), en utilisant un seuillage global où chaque pixel de l'image est comparé à ce seuil et prend la valeur blanc ou noir selon qu'il est supérieur ou inférieur . Finalement , nous obtenant une matrice représentant l'image et qui ne contient que des 0 et des 1 (voir la figure 3.3).



(a) modèle 1



(b) modèle 2

Figure 3.3: Exemple d'image binaire.

3.2.2.2 Normalisation de taille

Cette étape permet de changer les tailles des images à des tailles norme. l'étape de normalisation est nécessaire, parce que chaque caméra donne une image de taille différente.(voir la figure 3.4).

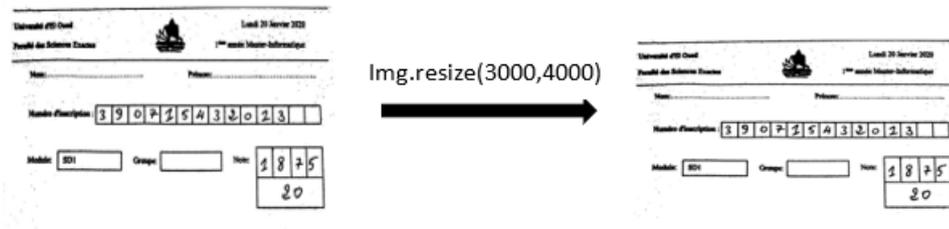


Figure 3.4: Normalisation de taille.

3.2.3 Segmentation

Après avoir prétraitée l'image source, nous arrivons maintenant à l'étape de segmentation, qui est considérée comme l'une des étapes les plus importantes du système. Elle a comme but de localiser les segments d'images de chiffres. Elle consiste à faire deux opérations de bases :

- Localiser les rectangles de chiffres et les recadrer.
- Segmenter l'image d'une série de chiffres.

3.2.3.1 Localiser les rectangles et recadrer

Cette étape consiste à trouver dans l'image les différents rectangles ou cadre de chiffres (voir la figure 3.5 et l'algorithme 3.1).

Nous appliquons la fonction `cv2.connectedComponentsWithStats` qui renvoie les informations suivantes :

- Les zones des chiffres délimitées en se basant sur les composants connectés,
- Les coordonnées des zones de délimitation (en pixels).

Algorithme 3.1: Algorithme localiser les rectangles.

Entrée:

`img`;

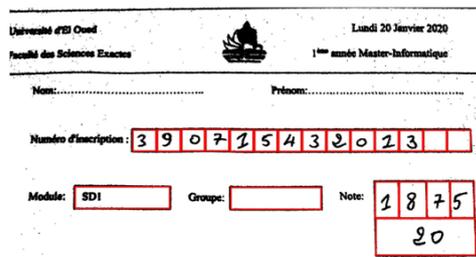
Sortie:

`rectangles`;

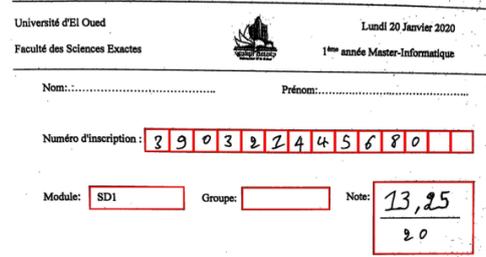
Début

1. Acquisition l'image
 2. Transforme l'image en image binaire.
 3. Applique la fonction `cv2.connectedComponentsWithStats` qui renvoie les informations Le zone de délimitation du composant connecté, y compris les coordonnées et le zone de le zone de délimitation (en pixels).
 4. localiser les cadres de chiffres (les rectangles existants).
-

Pour chaque composant connecté, nous dessinons la boîte englobante (rectangle rouge), ces composants connectés sont des série de nombres ou de nombres séparés.



(a) modèle 1



(b) modèle 2

Figure 3.5: Détecter les rectangles.

Après cela il faut recadrer les rectangles des petites images. Ensuite, le système procède aux opérations suivantes :

- Nous appliquons la fonction bitwise-not () pour séparer l'arrière-plan du premier plan, qui est une opération unaire qui effectue une négation logique de chaque bit, reflétant ainsi les couleurs de l'image. (le premier plan en blanc et l'arrière plan en noir)
- Nous appliquons la fonction del-noise () pour supprimer le bruit, elle érode les limites de l'objet au premier plan (le premier plan en blanc). Donc, ce qui se passe, c'est qu'elle supprime tous les petits pixels regroupés, où leur taille est petite par rapport à une image de chiffre. Ainsi, l'épaisseur ou la taille de l'objet au premier plan diminue ou simplement la zone blanche diminue dans l'image. Elle est utile pour supprimer les petits bruits blancs .
- Nous appliquons la fonction morphologyEx (), qui augmente la zone blanche de l'image ou la taille de l'objet au premier plan augmente. Parce que del-noise() supprime les bruits blancs, mais elle rétrécit aussi notre objet. Alors on le dilate . Puisque le bruit est éliminé, ils ne reviendront pas, mais notre zone d'objet augmente.
- Nous appliquons la fonction del-empty () , pour supprimer des espaces vides de l'image

Après ces opérations, on distingue les cas suivants:

- Dans le cas du numéro d'inscription, chaque rectangle est isolé, ce qui donne des petites images contenant des numéros séparés (voir la figure 3.6).



Figure 3.6: Numéro d'inscription (images séparés)

- Dans le cas du note, on distingue les deux cas suivants :
 - dans le cas du modèle 1, chaque rectangle est isolé, ce qui donne des petites images contenant des nombres séparés (voir la figure 3.7).



Figure 3.7: modèle 1: La note est composée des images séparés.

- dans le cas du modèle 2, on segmente le rectangle en deux et on prend la partie supérieure (voir la figure 3.8).



Figure 3.8: model 2: La note est une séquence de chiffres (une image).

3.2.3.2 Segmenter l'image d'une séquence de chiffres

Afin de réaliser cette segmentation, il faut combiné des segmentations verticales avec une segmentation de contour pour segmenter la séquence de chiffres. Le but de la première segmentation est de séparer les chiffres connectés dans l'image en segmentant l'image en tranches verticales.

Après avoir extraire la note, nous appliquons une projection verticale pour sépare les nombres connectés si le cas échéant (voir la figure 3.9 et l'algorithme 3.2).



Figure 3.9: modele 2 : la note séparé en deux.

Algorithme 3.2: Segmentation de la séquence de chiffres.

Fonction *segmentation* (*img*)

Début

Si (*Le rectangle contient un nombre*) **Alors** Il applique l'algorithme de
 déttection de contour

Sinon

Si (*Le rectangle contient une série de nombres séparé*) **Alors** Il applique
 l'algorithme de déttection de contour

Sinon

 Il applique d'abord une projection verticale pour les séparé , puis Il
 applique l'algorithme de déttection de contour

return *numéros séparé*

Après avoir séparé les chiffres, nous appliquons un algorithme de déttection des contours pour segmenter l'image en images en un ensemble de chiffres séparés. Un contour est une courbe fermée joignant tous les points continus, ils représentent les formes d'objets trouvés dans une image. La déttection des contours est une technique utile pour détecter les bordures des objets (voir la figure 3.10).

Pour cela, il faut appliquer les opérations suivantes:

- Nous appliquons la fonction `findContours` () pour trouver les contours.
- Nous appliquons la fonction `drawContours` () pour dessiner les contours .



Figure 3.10: modele 2 : Déttection l'objet.

Une fois les objets sont sélectionnés,un cadrage d'objets est réalisé afin de pouvoir faire la reconnaissance des chiffres.

3.2.4 Apprentissage

Dans cette phase, le modèle est entraîné à l'aide des images de chiffres de la base d'images MNIST. C'est une grande base de données des images de chiffres manuscrits, ce qui permet d'utiliser des algorithmes d'apprentissage automatique et l'apprentissage profondeur pour reconnaître les images numériques.

3.2.5 Prédiction

Dans cette phase, les images des chiffres manuscrits des nombres d'inscription ou des notes sont reconnus en se basant sur le modèle créé préalablement dans la phase d'apprentissage. La prédiction suit l'algorithme 3.3 suivant :

Algorithme 3.3: Algorithme de prédiction d'image

Fonction *pred_digit* (*path*, *MP*)

```
n = 0
img = cv2.imread(path,0)
dim =(80,100)
img = cv2.resize(img,dim)
img = del_noise(img)
img= del_empty(img)
(thresh,img)=threshold(img,128,255,cv2.THRESH_BINARY|cv2.THRESH_OTSU)
img.reshape(-1,size_x,size_y,1)
digit=model_CNN.predict(img)
return digit[0]
```

3.2.6 Insertion des notes dans le fichier Excel

Dans cette phase, des numéros d'inscription et des notes permettent de faire la mise à jour du fichier Excel cible. la modification est réalisée en se basant sur le numéro d'inscription comme critère de recherche dans le fichier Excel afin d'insérer la note dans la cellule correcte (voir la figure 3.11).

0	39023418957	11.50	
1	39090015492	13.75	
2	39004512305	12.25	
3	39030512397	08.15	
4	39080135672	06.75	
5	39081023562	03.50	
6	39012356420	14.00	
7	39075213610	01.50	
8	39034421891	17.00	
9	39023756910	20.00	
10	3903105067	19.75	
11	3908050703	19.00	
12	3906012309	12.25	
13	3907018805	17.50	
14	3904560210	17.25	
15	3903014903	17.00	

(a) modèle 1

	Mat	Note	
0	39043004581	14	
1	59081542035	06.75	
2	39041958349	17.00	
3	39030148923	08.15	
4	39085781031	11.2	
5	39401582231	0	
6	39091452830	20	
7	3902537108	11.75	
8	39034500341	19.50	
9	3903750735	12.25	
10	3903015723	20	
11	3907075523	13.95	
12	3903420588	17.25	
13	3902410325	06.50	
14	3903250391	14.25	
15	3301347528	10.75	

(c) modèle 2

16	3903450701	16.75	
17	3901340532	15.25	
18	3902091432	14.75	
19	3907310221	13.25	
20	3903013502	05.25	
21	3908456201	05.75	
22	3903425670	06.00	
23	3901350204	06.50	
24	3909503301	07.25	
25	3903050012	07.75	
26	3901002013	08.50	
27	3908571032	08.35	
28	3903503001	11.25	
29	3904150832	11.00	

(b) modèle 1

16	3908875032	16.50	
17	3903408410	03.25	
18	8901035801	18.50	
19	3905310254	04.50	
20	3904035981	5	
21	3904524109	19.50	
22	3911240395	3.5	
23	3902350013	00.50	
24	3902472043	18.25	
25	39084150320	09.75	
26	3920390178	00.50	
27	3901008302	12.75	
28	3903402549	11	

(d) modèle 2

Figure 3.11: Fichier Excel

Conclusion

Dans ce chapitre, nous avons commencé par illustrer l'architecture globale du système proposé. Ensuite, nous avons expliqué en détail les différents étapes du système, où nous avons vu comment les deux modèles des copies d'examens sont traités. Dans le chapitre suivant, nous présentons les résultats obtenus, ainsi, qu'une discussion sur ces résultats.

Validation

Introduction

Dans le chapitre précédent, nous avons présenté l'architecture générale du système et montré les détails de chaque phase et chaque étape du traitement. Dans ce chapitre, nous présentons les outils utilisés : Python, Jupyter Notebook ainsi que les bibliothèques utilisées Open CV, TensorFlow, Scikit-learn, Keras. Finalement, nous présentons sur les résultats obtenus.

4.1 Langage de programmation et bibliothèques utilisées

Dans cette section, nous allons donner un aperçu sur le langage de programmation Python, ainsi que les différentes bibliothèques (Jupyter Notebook, OpenCV, TensorFlow, Scikit-learn, Keras) utilisées dans l'implémentation de l'application. De plus, nous allons présenter la configuration matériel et logiciel utilisée.

4.1.1 Python

Python est un langage de programmation de haut niveau et de licence libre (open source). Il fonctionne sur de nombreux systèmes d'exploitation : Windows, Mac OS X, Linux.

Python est récemment devenu le langage de programmation le plus utilisé dans les projets de machine learning et deep learning, science de données (data science), et l'intelligence artificielle.

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 les plus utilisées en machine learning sont : Scikitlearn, Tensorflow, Keras.

4.1.2 Jupyter Notebook

Jupyter est une application web open source qui permet d'écrire du code en Python (et d'autres langages), largement utilisée pour apprendre Python et les algorithmes d'apprentissage automatique, facile à utiliser et dispose d'une interface utilisateur simple.

4.1.3 OpenCV

OpenCV est une bibliothèque open source pour l'analyse d'images et de vidéos. Aujourd'hui, la bibliothèque à > 2500 algorithmes optimisés. OpenCV prend en charge une grande variété de langages de programmation tels que C++, Python, Java, ... etc, et il est disponible sur différentes plates-formes, notamment Windows, Linux, Android et iOS.

4.1.4 TensorFlow

TensorFlow c'est un framework open source compatible avec python pour l'apprentissage automatique, il regroupe un grand nombre de modèles et d'algorithmes de Machine Learning et de Deep Learning. Cette bibliothèque permet notamment d'entraîner et d'exécuter des réseaux de neurones pour la classification de chiffres manuscrit.

4.1.5 Scikit-learn

Scikit-learn est une bibliothèque libre Python dédiée à l'apprentissage automatique. Elle comporte une diversité d'algorithmes tels que la machine à vecteurs de support, les forêts aléatoires et les k-voisins ... etc.

4.1.6 Keras

Keras est une bibliothèque Python open source gratuite, puissante et facile à utiliser pour développer et évaluer des modèles d'apprentissage en profondeur.

4.1.7 Configuration utilisée

Ce programme à été créé dans un type d'ordinateur portable HP avec les spécifications suivantes :

- Processeur : i5- 4300U CPU @ 1.90 GHz.
- Mémoire installé (RAM) : 12.0GB.
- Type de système : système d'exploitation 64 bits.
- System d'exploitation : Windows 10 pro.

4.2 Le base d'image MNIST

MNIST (Mixed National Institute of Standards and Technology) est une base de d'images de chiffres manuscrits qui est couramment utilisée par divers systèmes de traitement

d'image. La base d'images est également largement utilisée pour l'apprentissage et les tests. Elle contient 60.000 images d'apprentissage et 10.000 images de test. [16, 44]

<i>Classes</i>	<i>APPRENTISSAGE</i>	<i>TEST</i>	<i>TOTALE</i>
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
<i>Totale</i>	60000	10000	70000

Figure 4.1: Répartition de la base MNIST pour les chiffres.

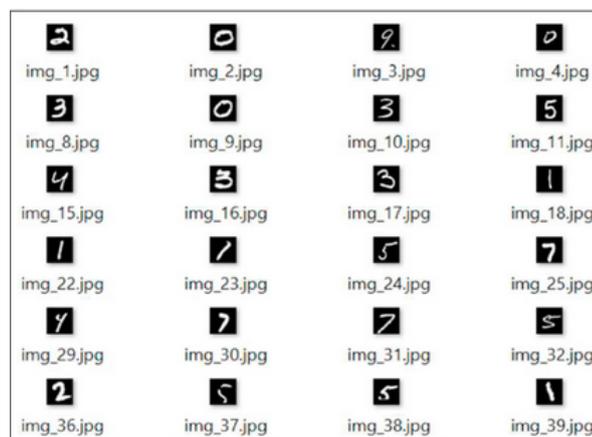


Figure 4.2: Exemples de la base de données MNIST.

4.3 Modèle CNN utilisé

La figure 4.3 illustre une partie du code qui représente notre modèle de réseau de neurones convolutif pour reconnaître les chiffres manuscrits.

- Batch_size : 32
- Nombre de layers : 7 layers
- Nombre d'époques : 50
- Nombre d'étape : 32
- Taux d'apprentissage : 0.0001
- Fonction d'activation : sigmoid

```

model = Sequential()

model.add(Conv2D(32, (5, 5), input_shape=(IMG_SIZE_X, IMG_SIZE_Y, 1), activation='relu'))
model.add(AveragePooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3), padding='same', activation='relu'))
model.add(AveragePooling2D(pool_size=(2, 2), padding='same'))

model.add(Flatten())
#model.add(Activation('relu'))
#model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dense(10))
model.add(Activation('sigmoid'))
model.summary()

```

Figure 4.3: Modèle CNN.

4.4 Résultats & Discussion

Dans cette étude, la reconnaissance des chiffres de l'écriture manuscrite à été réalisée avec 09 algorithmes différents. L'ensemble de données de chiffres manuscrits MNIST à été utilisé dans tous les tests. Le tableau 4.1 illustre les résultats du training et test sur la base d'images MNIST.

Voici quelques concepts de base :

- Précision de l'entraînement (training accuracy) : le taux de réussite et prediction corrects des images sur lesquelles le réseau est formé.
- Précision de la validation (validation accuracy) : taux de réussite et prediction corrects des images testées.
- F1 Score : c'est le résultat complet de l'efficacité et de la précision du modèle.

Algorithm	Training Accuracy	Test Accuracy	F1 Score
CNN	0.99	0.96	0.92
SVM	0.99	0.95	0.94
Random Forest	1	0.93	0.93
KNN	0.95	0.92	0.91
Bagging	1	0.84	0.83
Neural network	0.72	0.68	0.67
Arbre Descion	1	0.66	0.67
LogisticRegression	1	0.89	0.88
Naive Bayse	0.81	0.7	0.7

Tableau 4.1: Les résultats de training et test sur MNIST.

Dans cette étude, nous avons remarqué que l'algorithme CNN est le plus efficace dans le processus de reconnaissance, en atteignant la précision de 96 % et les résultats obtenus ont été très convaincants et satisfaisants. Et donc nous avons opté la méthode de Réseau neuronal convolutif pour décider la classe d'appartenance pour chaque chiffre. CNN est l'un des algorithmes les plus utilisés dans le domaine de la reconnaissance

des chiffres manuscrits, il joue deux rôles principaux dans le traitement de l'image. Le rôle d'un extracteur de caractéristiques avec un processus de convolution qui traite les données et le rôle d'un classificateur qui contient plusieurs couches pour classer les images.

De plus, les algorithmes SVM, KNN, Bagging, NN, DT,RF, LR, NB donne des résultats acceptable et ils sont très utilisés dans le domaine la reconnaissance des chiffres manuscrite.

La figure 4.4 illustre une évaluation du réseau de neurones convolutif.

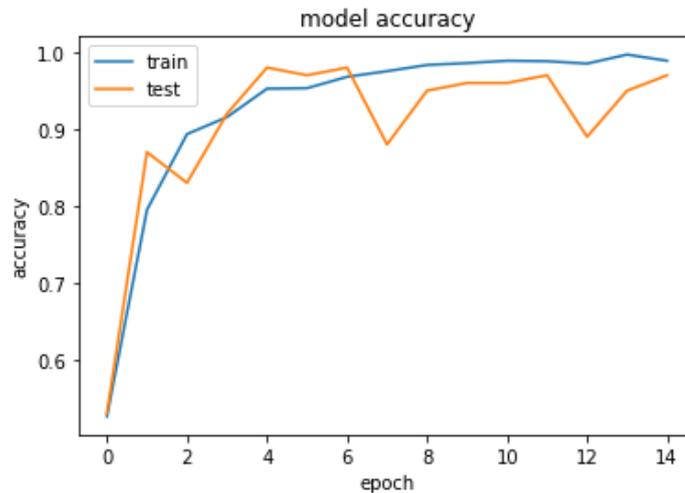


Figure 4.4: Évaluation de la précision du modèle.

La figure 4.5 illustre modèle loss.

- Modèle perte (loss): différence entre le résultat réel et le résultat attendu, lorsque vous approchez de 0 un réseau est meilleur.

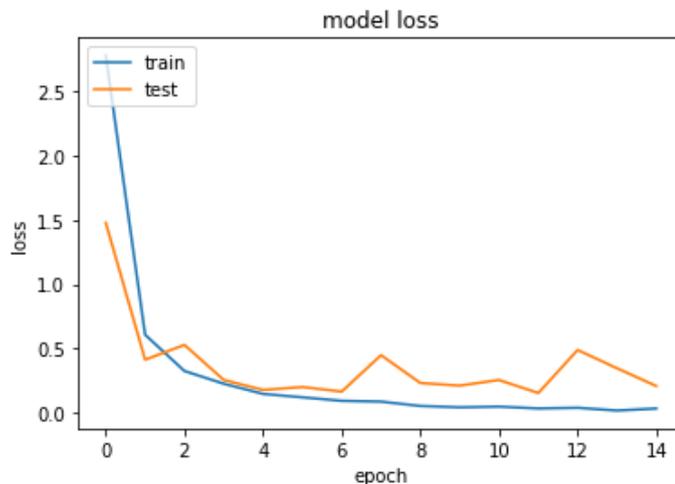


Figure 4.5: Évaluation de la perte du modèle.

Le tableau 4.2 illustre les résultats de taux de reconnaissance en utilisant 30 exemples pour chaque modèle.

Modèle	Nbr d'exemples	Taux de reconnaissance
Modèle 1	30	95.81%
Modèle 2	30	93.94%

Tableau 4.2: Résultats de taux de reconnaissance pour chaque modèle.

du tableau 4.2 de résultats présenté ci-dessus, nous pouvons remarqué que les résultats obtenus avec le Modèle 1 avec un taux de reconnaissance de 95.81 %, et plus mieux que les résultats obtenus avec le second Modèle avec un taux de reconnaissance de 93.94 % . De ces résultats nous pouvons confirmer que l'utilisation des cellules qui divise les chiffres peut aider considérablement à l'amélioration de la performance des système de reconnaissance de chiffres.

Conclusion

Dans ce chapitre, nous avons présenté les résultats obtenus à l'aide d'un ensemble d'algorithmes d'apprentissage, où nous avons obtenu des résultats très encourageants en utilisant le classificateur CNN avec un taux de reconnaissance qui atteint 96 %. Pour cela, nous avons opté la méthode de Réseau neuronal convolutif comme classificateur pour notre système.

Nous avons également présenté les résultats obtenus à l'aide de deux modèles différents des copies d'examens, et toutes les expériences réalisées ont montré des résultats très encourageants. Pour le premier Modèle nous avons obtenu 95.81 % et pour le second Modèle nous obtenus 93.94 % .

Conclusion et perspectives

Jusqu'aujourd'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. Le système de reconnaissance de chiffres manuscrits, 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. Le but du travail est de reconnaître les notes des étudiants à partir d'une copie d'examen corrigé, plus précisément. l'enseignant insère manuellement les notes des étudiants dans un fichier excel. ce travail peut être si dur et prend beaucoup de temps surtout si le nombre des copies d'examen est très grand, dans ce travail nous avons essayé de le rendre plus facile par la méthode proposée.

Dans ce projet, la segmentation et la reconnaissance des notes est l'étape la plus complexe du système, et en général, dans notre système , nous avons utilisé la méthode de projection verticale pour séparer les chiffres et la méthode de détection des contours pour détecter les objets dans l'image. Ensuite, nous obtenons des petites images avec des numéros séparés, puis nous apprenons à les connaître.

Après avoir exécuté un groupe de tests pour un ensemble d'algorithmes d'apprentissage tels que cnn, svm, Le classifieur du K plus proches voisins, bagging, Les arbres de décision. . . etc sur une base de données MNIST .

Nous avons remarqué que la méthode CNN est efficace dans le processus de reconnaissance, atteignant la précision 96 % et les résultats obtenus ont été très convaincants et satisfaisants. Et donc nous avons opté la méthode de Réseau neuronal convolutif pour décider la classe d'appartenance pour chaque chiffre.

Enfin, nous avons mené une ensemble d'expériences sur deux des modèles , et toutes les expériences réalisées ont montré des résultats très encourageants. Pour le premier Modèle nous avons obtenus 95.81 % et pour le second Modèle nous obtenus 93.94 %. Ainsi, nous avons prouvé l'efficacité et l'efficacité de notre algorithme de segmentation et classificateur de reconnaissance proposés.

Comme perspectives, il est possible d'améliorer le travail en proposant une configuration CNN plus mieux. En plus, nous espérons de développer le programme en une application mobile afin qu'il puisse être facilement utilisé par un large segment d'enseignants.



Bibliographie

- [1] D. ABDELHAKIM – “La reconnaissance des chiffres manuscrits par les machines à vecteurs de support (svms)”, Thèse, Thèse de Master, Université de Tébessa, 2011.
- [2] G. ABDELJALIL – “Segmentation-verification for handwritten digit recognition”, Thèse, MINISTRY OF HIGHER EDUCATION, 2016.
- [3] A. ABDI – “Three types of machine learning algorithms”, 2016.
- [4] S. ABDULLA, A. AL-NASSIRI et R. A. SALAM – “Off-line arabic handwritten word segmentation using rotational invariant segments features.”, *Int. Arab J. Inf. Technol.* **5** (2008), no. 2.
- [5] T. A. ASSEGIE et P. S. NAIR – “Handwritten digits recognition with decision tree classification: a machine learning approach”, *International Journal of Electrical and Computer Engineering* **9** (2019), no. 5, p. 4446.
- [6] C.-A. AZENCOTT – *Introduction au machine learning*, Dunod, 2019.
- [7] U. R. BABU, A. K. CHINTHA et Y. VENKATESWARLU – “Handwritten digit recognition using structural, statistical features and k-nearest neighbor classifier”, *International Journal Information Engineering and Electronic Business* **6** (2014), no. 1, p. 62–68.
- [8] U. R. BABU, Y. VENKATESWARLU et A. K. CHINTHA – “Handwritten digit recognition using k-nearest neighbour classifier”, *2014 World Congress on Computing and Communication Technologies*, IEEE, 2014, p. 60–65.
- [9] N. BENAHMED – *Optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés: Sélection et pondération des primitives par algorithmes génétiques*, École de technologie supérieure, 2002.
- [10] P. BUYSENS – “Fusion de différents modes de capture pour la reconnaissance du visage appliquée aux e_transactions”, Thèse, Université de Caen, 2011.
- [11] A. CHAARI – “Nouvelle approche d’identification dans les bases de données biométriques basée sur une classification non supervisée”, Thèse, Université d’Evry-Val d’Essonne, 2009.
- [12] C. CHATELAIN – “Extraction de séquences numériques dans des documents manuscrits quelconques”, Thèse, Université de Rouen, 2006.
- [13] S. CHEN, R. ALMAMLOOK, Y. GU et L. WELLS – “Offline handwritten digits recognition using machine learning”, *Proceedings of the International Conference on Industrial Engineering and Operations Management Washington DC, USA*, 2018.
- [14] A. CHOUCANE – “Analyse d’images d’expressions faciales et orientation de la tête basée sur la profondeur”, Thèse, Université Mohamed Khider-Biskra, 2016.
- [15] M. COTÉ – “Utilisation d’un modèle d’accès lexical et de concepts perceptifs pour la reconnaissance d’images de mots cursifs”, Thèse, Paris, ENST, 1997.
- [16] L. DENG – “The mnist database of handwritten digit images for machine learning research [best of the web]”, *IEEE Signal Processing Magazine* **29** (2012), no. 6, p. 141–142.
- [17] P. Q. DUNG – “Réseaux de neurones pour la reconnaissance des formes”, *Travail d’Intérêt Personnel Encadré par Prof. HO Tuong Vinh, Hanoi* **15** (2005).
- [18] B. EL KESSAB, C. DAoui, B. BOUIKHALENE et R. SALOUAN – “A comparative study between the support vectors machines and the k-nearest neighbors in the handwritten latin numerals recognition”, *International Journal of Signal Processing, Image Processing and Pattern Recognition* **8** (2015), no. 2, p. 325–336.

- [19] A. EL-YACOUBI, R. SABOURIN, M. GILLOUX et C. Y. SUEN – “Improved model architecture and training phase in an off-line hmm-based word recognition system”, *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, vol. 2, IEEE, 1998, p. 1521–1525.
- [20] T. C. FALTEMIER – “Flexible and robust 3d face recognition”, Thèse, Citeseer, 2007.
- [21] M. Z. FAOUZI – “Les supports vecteurs machines (svm) pour la reconnaissance des caractères manuscrits arabes”, Thèse, 2010.
- [22] K. FUKUSHIMA – “Neocognitron: A hierarchical neural network capable of visual pattern recognition”, *Neural networks* **1** (1988), no. 2, p. 119–130.
- [23] A. GATTAL – *Combinaison de méthodes de segmentation explicite pour la reconnaissance automatique des chiffres manuscrits*, 2010.
- [24] A. GATTAL – “Segmentation automatique pour la reconnaissance numérique des chèques bancaires algériens”, Thèse, Thèse de MAGISTER, centre Universitaire de Khanchela, 2011.
- [25] I. J. GOODFELLOW, Y. BULATOV, J. IBARZ, S. ARNOUD et V. SHET – “Multi-digit number recognition from street view imagery using deep convolutional neural networks”, *arXiv preprint arXiv:1312.6082* (2013).
- [26] F. GRANDIDIER – “Un nouvel algorithme de sélection de caractéristiques: application à la lecture automatique de l’écriture manuscrite”, Thèse, École de technologie supérieure, 2003.
- [27] I. GUYON, J. WESTON, S. BARNHILL et V. VAPNIK – “Gene selection for cancer classification using support vector machines”, *Machine learning* **46** (2002), no. 1, p. 389–422.
- [28] S. HAITAAMAR – “Segmentation de textes en caracteres pour la reconnaissance optique de l’écriture arabe”, Thèse, Université de Batna 2, 2007.
- [29] S. B. HALLALE et G. D. SALUNKE – “Offline handwritten digit recognition using neural network”, *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* **2** (2013), no. 9, p. 4373–4377.
- [30] C.-L. HUANG et C.-J. WANG – “A ga-based feature selection and parameters optimization for support vector machines”, *Expert Systems with applications* **31** (2006), no. 2, p. 231–240.
- [31] A. JAUMARD-HAKOUN – “Modélisation et synthèse de voix chantée à partir de descripteurs visuels extraits d’images échographiques et optiques des articulateurs”, Thèse, Université Pierre et Marie Curie-Paris VI, 2016.
- [32] JEAN-PIERRE – “reconnaissance de l’écriture manuscrite”, Thèse, Université de Rennes 1.
- [33] R. KARAKAYA et S. KAZAN – “Handwritten digit recognition using machine learning”, *Sakarya University Journal of Science* **25** (2021), no. 1, p. 65–71.
- [34] E. KUSSUL et T. BAIDYK – “Improved method of handwritten digit recognition tested on mnist database”, *Image and Vision Computing* **22** (2004), no. 12, p. 971–981.
- [35] P. M. LALLICAN, C. VIARD-GAUDIN et S. KNERR – “From off-line to on-line handwriting recognition”, (2000).
- [36] K. LAVANYA, S. BAJAJ, P. TANK et S. JAIN – “Handwritten digit recognition using hoeffding tree, decision tree and random forests—a comparative approach”, *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)*, IEEE, 2017, p. 1–6.
- [37] Y. LEI, C. LIU, X. DING et Q. FU – “A recognition based system for segmentation of touching handwritten numeral strings”, *Ninth International Workshop on Frontiers in Handwriting Recognition*, IEEE, 2004, p. 294–299.
- [38] F. MENASRI – “Contributions à la reconnaissance de l’écriture arabe manuscrite”, *UNIVERSITE PARIS DESCARTES, Thèse de doctorat* (2008).
- [39] R. MIFDAL – “Application des techniques d’apprentissage automatique pour la prédiction de la tendance des titres financiers”, Thèse, École de technologie supérieure, 2019.

- [40] H. MISHRA, S. K. PATHAK et A. SRIVASTAVA – “Handwritten digits recognition using machine learning algorithms”, *International Research Journal of Engineering and Technology (IRJET)* **7** (2020), no. 05, p. 8073–8079.
- [41] X.-X. NIU et C. Y. SUEN – “A novel hybrid cnn–svm classifier for recognizing handwritten digits”, *Pattern Recognition* **45** (2012), no. 4, p. 1318–1325.
- [42] K. O’ SHEA et R. NASH – “An introduction to convolutional neural networks”, *arXiv preprint arXiv:1511.08458* (2015).
- [43] D. PATRICE – “Contribution à la segmentation et à la reconnaissance de l’écriture manuscrite”, Thèse, thesis, The National Institute of Applied Sciences of Lyon, 1994.
- [44] J. C. PLATT et al. – “Using analytic qp and sparseness to speed training of support vector machines”, *Advances in neural information processing systems* (1999), p. 557–563.
- [45] N. SALIMA – “Reconnaissance de caractères manuscrits par intelligence collective”, Thèse, Université de Sétif 1-Ferhat Abbas, 2013.
- [46] A. SHARMA – “A combined static and dynamic feature extraction technique to recognize handwritten digits”, *Vietnam Journal of Computer Science* **2** (2015), no. 3, p. 133–142.
- [47] E. TUBA et N. BACANIN – “An algorithm for handwritten digit recognition using projection histograms and svm classifier”, *2015 23rd Telecommunications Forum Telfor (TELFOR)*, IEEE, 2015, p. 464–467.
- [48] P. VINCENT – “Modeles a noyaux a structure locale (french text).”, (2003).
- [49] M. WU et Z. ZHANG – “Handwritten digit classification using the mnist data set”, *Course project CSE802: Pattern Classification & Analysis* (2010).
- [50] F. ZAIZ – “Technique basée puzzle/svm pour l’amélioration de la reconnaissance du texte arabe manuscrit”, Thèse, Université Mohamed Khider-Biskra, 2017.
- [51] F. ZAIZ, M. C. BABAHENINI et A. DJEFFAL – “Puzzle based system for improving arabic handwriting recognition”, *Engineering Applications of Artificial Intelligence* **56** (2016), p. 222–229.
- [52] H.-H. ZHAO et H. LIU – “Multiple classifiers fusion and cnn feature extraction for handwritten digits recognition”, *Granular Computing* **5** (2020), no. 3, p. 411–418.

