



UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED
FACULTÉ DES SCIENCES EXACTES
Département D'Informatique



Mémoire de Fin D'étude
Présenté pour l'obtention du Diplôme de

MASTER ACADEMIQUE

Domaine : **Mathématique et Informatique**
Filière: **Informatique**
Spécialité : **Systemes Distribués et Intelligence Artificielle**

Présenté par :

- **HENKA Abdeljabbar**
- **SOUDA Fares**

Thème

**Proposition d'une architecture de
reconnaisseur de chiffres,
Application saisie de notes**

Soutenue le xx-xx- 2019 Devant le jury:

M.	BERDJOUH Chafik	MCA	Président
M.	GUIA Sana Alsahar	MAA	Rapporteur
M.	ZAIZ Faouzi	MAA	Encadreur

Année Universitaire: 2018-2019

Dédicaces

Tout d'abord, je veux rendre grâce à Dieu, le Clément et le Très Miséricordieux pour son amour éternel. C'est ainsi que je dédie cette thèse à:

*ma mère pour sa tendresse et mon père pour sa patience et encouragement
mes très chers frères et ma chère soeur pour leurs conseils,
mes cousins et cousines,
tous ceux que j'aime,
tous mes amis.*

Remerciements

Nous remercions Allah le tout puissant, qui nous a donné la force et la patience pour l'accomplissement de ce travail. Nous remercions notre encadrant : Mr. (ZAIZ Faouzi) son précieux conseil et son aide durant toute la période du travail. Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Nos remerciements s'étendent également à tous nos enseignants durant les années des études.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Résumé

Jusqu'aujourd'hui, la reconnaissance automatique de l'écriture (OCR: **O**ptic **C**haracter **R**ecognition) a un impact considérable sur notre vie quotidienne en raison de son importance et de ses exigences dans différents domaines, tels que l'identification du contenu des chèques bancaires, la reconnaissance de fichiers manuscrits, la saisie des notes des étudiants. La reconnaissance automatique de l'écriture manuscrite consiste à transformer des textes manuscrits ou imprimés en un fichier numérique, puis les traiter par ordinateur.

Diverses recherches ont été effectuées dans ce domaine au cours des dernières années pour améliorer les systèmes de reconnaissance de l'écriture. Mais, malgré ça il reste encore beaucoup d'imperfections dans les systèmes actuels (surtout des langues de nature complexe et cursive comme l'Arabe) en particulier dans la reconnaissance des chiffres et l'écriture artistique, ce qui représente un grand défi où on trouve des lignes irrégulières et qui se chevauchent, ... etc.

Dans ce travail, nous proposons une nouvelle méthode basée sur le scan de quatre côtés de l'image en trois étapes afin de segmenter les chiffres (chevauchés et connectés). La méthode est basée sur la projection verticale et la projection horizontale. Ensuite, nous recherchons des points intérêts qui vont être utilisés pour surmonter le problème des formes irrégulières et des chevauchements des lignes de l'écriture.

Mots Clés: chiffres manuscrits, OCR, segmentation, classification, chiffres chevauchés.

Abstract

Until today, automatic recognition of writing (OCR: **O**ptic **C**haracter **R**ecognition) has a considerable impact on our daily lives because of its importance and its requirements in various fields, such as the identification of the contents of bank checks, the recognition of handwritten files and typing of student notes. Automatic recognition of handwriting consists of transforming handwritten or printed texts into an digital file, then processing them by computer.

Various researches have been carried out in this field in recent years to improve the systems of recognition of writing. But, despite this, there are still many imperfections in the current systems (especially languages of complex and cursive nature like Arabic) especially in the recognition of numbers and artistic writing, which is a big challenge where there are irregular and overlapping lines . . . etc.

In this work, we propose a new method based on the scan of the four sides of the image in three steps in order to segment the numbers (overlapped and connected). The method is based on vertical projection and horizontal projection. Then we look for points of interest that will be used to overcome the problem of irregular shapes and overlapping lines of writing.

Key words: handwriting digits, OCR, segmentation, classification, overlapping digits.

الملخص

إن التعرف على الأرقام المكتوبة بخط اليد أصبح يلعب دورا هاما في العالم ، نظرا لأهميتها والحاجة إليها في العديد من المجالات منها : التعرف على محتوى الصكوك البنكية ، وصل عدادات الكهرباء والغاز ، الكشف على نقاط التلاميذ بعد التصحيح . حيث لها القدرة على تحويل صورة لنص مكتوب بخط اليد أو مطبوع إلى ملف إلكتروني يحتوي على نفس النص لكي يتمكن المستخدم من معالجته من جديد عن طريق الألة ..

بالرغم من العديد ومختلف البحوث المنجزة في هذا المجال خلال السنوات الاخيرة قصد تحسين أنظمة التعرف على الخط المكتوب يدويا ، لاتزال هناك العديد من النقائص بالأنظمة اللاتينية للأرقام ، الكتابة الفنية للأرقام تمثل تحدي كبير لأنها تكتب مع التركيز ومراعات الجانب الفني والجمالي للمظهر على حساب الأشكال الأساسية لمختلف الأرقام . لهذا، تتسم بعدم إنتظام وتداخل وتقاطع في خطوط رسم الأرقام .

في هذا العمل ، نقتراح طريقة تعتمد على المسح من الجهات الأربع وبثلاثة مراحل لتجزئة كتابة الأرقام (المتداخلة والمتصلة) . والتي تعتمد على الإسقاط العمودي والأفقي ، ثم البحث عن النقاط ذات الأهمية بعدها تستخدم هذه النقاط لتغلب على مشكل عدم الإنتظام ، والتداخل والتقاطع في خطوط الكتابة ..

كلمات مفتاحية: الرقم المكتوب بخط اليد ، التعرف ، التجزئة ، التصنيف ، الأرقام المتداخلة .

Table des matières

Dédicaces	i
Remerciements	ii
Résumé	iii
Table des matières	vi
Liste des figures	ix
Liste des tableaux	xi
Liste des algorithmes	xii
Introduction	1
1 Reconnaissance de l'écriture manuscrite	3
Introduction	3
1.1 Différents aspects de la reconnaissance	3
1.1.1 Production et reconnaissance	4
1.1.2 Reconnaissance en-ligne / hors-ligne	4
1.2 Phases d'un OCR	5
1.2.1 Phase d'acquisition	6
1.2.2 Phase de pré-traitement	6
1.2.3 phase de segmentation	8
1.2.4 Extraction des caractéristiques	9
1.2.5 Classification	10

1.3	Méthode de classification statistique	12
1.3.1	Réseaux de neurones	13
1.3.2	Machines à Vecteur Support (SVMs)	13
1.3.3	K plus proches voisins	14
	Conclusion	14
2	Technique de segmentation	15
	Introduction	15
2.1	Problématique posée par l'écriture du cursif	15
2.2	Concepts de segmentation de chiffres	16
2.3	Techniques de segmentation	16
2.3.1	Segmentation à partir du squelette	16
2.3.2	Segmentation à partir du contour	17
2.3.3	Segmentation à partir des histogrammes de projection verticales	17
2.3.4	Segmentation basée sur les fenêtres glissantes	18
2.4	Niveaux de segmentation	18
2.5	Segmentation de page	19
2.5.1	Segmentation en lignes	19
2.5.2	Segmentation en séquence de chiffres	20
2.5.3	Segmentation en chiffres isolés	20
2.6	Les travaux antérieurs	21
	Conclusion	26
3	Conception et Architecture	27
	Introduction	27
3.1	Architecture générale du système	27
3.2	Génération de l'image binaire	29
3.2.1	Calcul de la taille du stylo	29
3.3	Projection verticale	30
3.3.1	Calcul de l'histogramme vertical	31
3.3.2	Segmentation des chiffres	31
3.4	Détection des chiffres isolés / connectés	31
3.4.1	Images des chiffres isolés	32
3.4.2	Image des chiffres connectés	32

3.5	Calcul de l'histogramme de projection horizontale	32
3.6	Détermination des points d'intérêts	33
3.7	Segmentation des chiffres	35
3.8	Choisissez deux points	35
3.9	Reconnaissance	36
	Conclusion	36
4	Réalisation et validation	37
	Introduction	37
4.1	Choix du langage de programmation	37
4.2	Environnement de développement intégré (IDE)	38
4.3	Caractéristiques de l'ordinateur utilisé	38
4.4	Interface et fenêtres	38
4.5	Transformation image	41
4.6	Résultat et Test	42
	4.6.1 Segmentation	42
	4.6.2 Reconnaissance	43
	Conclusion	44
	Conclusion et perspectives	45
	Bibliographie	46

Liste des figures

1	Reconnaissance de l'écriture manuscrite	3
1.1	Processus de production et de reconnaissance de documents[32]	4
1.2	Différence entre les deux types de reconnaissance (online , offline) [24]. . .	4
1.3	Exemples d'allographes des caractères [10].	5
1.4	Schéma général d'un système de reconnaissance des chiffres manuscrits[32].	6
1.5	Exemple d'un chiffre manuscrit normalisé[1].	7
1.6	Exemple de squelettisation[1].	8
1.7	Exemple de squelettisation[23].	10
1.8	Figure 1.6. Exemple d'hybridation entre deux approches (statistiques et structurelles).	12
2	Technique de segmentation	15
2.1	Chiffre cursif "5".	15
2.2	Exemple du chevauchement de chiffre.	16
2.3	: Exemple de squelettes de chiffre.	17
2.4	: Exemple de squelettes de chiffre [30].	17
2.5	Exemple d'un histogramme de projection de chiffre.	17
2.6	Exemple d'utilisation de la fenêtre glissante sur un chaine de chiffre. . . .	18
2.7	Illustration de l'arborescence du processus de segmentation.	19
2.8	Exemple de segmentation de page.	19
2.9	Exemple de segmentation en lignes d'une structure tableau : (a) le tableau et le profil lissé de la projection horizontale en rouge,avec la position de tous les maxima, (b) exclusion des fausses détections et détermination des bords de lignes [11].	20
2.10	Exemple de segmentation en séquence de chiffres d'un document. Les rectangles bleus délimitent les séquence de chiffres extraits [11].	20
2.11	Exemple de segmentation en chiffres isolés [11].	20

3	Conception et Architecture	27
3.1	Architecture du système de reconnaissance de chiffres manuscrits.	28
3.2	Transformation image originale en une image binaire.	29
3.3	Grille de pixels d'un chiffres.	29
3.4	Illustration de la projection verticale de chiffres.	31
3.5	Illustration du calcul de la projection horizontale d'un chiffres.	33
3.6	Mettres les points sur les chiffres connectés.	35
4	Réalisation et validation	37
4.1	Interface principale de l'application.	39
4.2	Fenêtre de téléchargement de photo.	40
4.3	Fenêtre d'affichage résultat de segmentation	41
4.4	Fenêtre d'affichage résultat de reconnaissance	41
4.5	Interface d'affichage résultat de reconnaissance	42
4.6	Taux et vitesse de segmentation de chiffres.	43
4.7	Taux de reconnaissance de chiffres.	43

Liste des tableaux

2.1	Comparaison entre quelques travaux antérieurs.	21
-----	--	----

Liste des algorithmes

3 Conception et Architecture	27
3.1 Calcul de la taille du stylo	30
3.2 Calcul de l’histogramme de projection verticale.	30
3.3 Calcul de l’histogramme de projection horizontale.	32
3.4 Construction du cycle.	34
3.5 Segmentation chevauchement	36

Introduction

La reconnaissance de l'écriture (OCR: **O**ptic **C**haracter **R**ecognition) recouvre un domaine très vaste dont lequel les recherches sont ouvertes jusqu'aujourd'hui. L'une des parties de la reconnaissance automatique est DC (Digits Recognition). Le travail essentiel de DC consiste à transformer une image de texte contenant des chiffres en un format de fichier numérique contenant les mêmes chiffres, format que la machine peut facilement comprendre et utiliser. Il simule exactement le processus de lecture des êtres humains. Ce processus (lecture) semble si facile pour les humains, mais il est si difficile pour la machine d'exécuter une telle tâche. La difficulté apparaît dans différents styles et formes de manuscrits, dans les textes et dans les chiffres qui se chevauchent, ainsi que dans les tirages aléatoires de l'écriture humaine.

Jusqu'à présent, ce domaine reste un domaine très vif pour la recherche vu les lacunes existait dans les systèmes développés surtout pour les langues qui s'écrit de manière complexe telle que l'Arabe. Dans notre vie quotidienne nous utilisons beaucoup de formulaires administratifs, des chèques postaux et bancaires ... etc, qui peuvent être saisi de manière automatique. La reconnaissance optique de caractères et/ou des chiffres vient pour simplifier cette tâche. La phase clé de l'OCR est la segmentation, qui est considérée comme le processus d'extraction de segments de base de d'une séquence de chiffres. Les résultats de cette phase affectent directement la performance d'un système OCR.

Dans ce travail, nous avons essayé de présenter une solution concernant la reconnaissance des chiffres et surtout les cas de chevauchement des chiffres. Nous proposons une nouvelle méthode de segmentation, qui fonctionne en segmentant chaque séquence de chiffres en suites d'images de chiffres isolés. Par la suite, il est possible de reconnaître chaque chiffre séparément. La segmentation est divisée en deux étapes : la segmentation verticale qui vient en premier lieu et qui utilise une projection verticale, puis, une segmentation horizontale qui applique une projection horizontale afin de segmenter le reste des chiffres chevauchés. Ces deux étapes éliminent toute possibilité de chevauchement des chiffres

existants. Les tests ont été réalisés sur un ensemble d'images créées localement plus un ensemble d'images de la base MNIST. Les résultats obtenus sont très suffisants et en-courageux.

Le reste du travail est divisé comme suit:

Dans le premier chapitre, nous allons essayer de donner un aperçu du sujet en définissant la plupart des termes de base et des techniques utilisés par les systèmes OCR en général, et des systèmes de reconnaissance de chiffres en particulier.

Dans le deuxième chapitre, nous allons présenté en plus de détails la segmentation en général et la segmentation des chiffres en particulier. En plus, nous allons essayer de faire une comparaison entre quelques travaux antérieurs en présentant l'approche, et les points positifs et les points négatifs de chacun.

Le troisième chapitre traitera l'architecture proposée de notre système et son fonctionnement. De plus, nous expliquons chaque étape de cette architecture en détails en mentionnant l'entrée et la sortie de chacune des étapes.

Le quatrième chapitre, nous présentons les résultats obtenus par la méthode proposée. Également, nous parlerons des obstacles rencontrés et de la manière selon laquelle nous les avons dépassés.

Enfin, nous terminons le travail par une conclusion sur les résultats obtenus par les méthode utilisées, et des perspectives de ce travail.

Reconnaissance de l'écriture manuscrite

Introduction

Malgré les avancements durant ces dernières décennies dans les systèmes de reconnaissance du manuscrit (OCR: **O**ptic **C**haracter **R**ecognition), il reste quelques améliorations qui doivent être ajoutées à ces systèmes surtout pour des langues de nature cursive et compliquée comme la langue Arabe. Ce qui laisse un domaine de recherche ouvert en prenant en considération les applications pratiques de ces systèmes comme la lecture de formulaires, de chèques bancaires, d'adresses postales... En plus, avec l'utilisation croissante des interfaces homme-machine et orientée vers la communication manuscrite dans les tablettes, PC, les applications orientées stylet, ..., ce domaine est considéré comme un domaine très actif [24]. Le système OCR consiste à transformer une image de texte manuscrite en un format numérique utilisable par la suite.

Cette transformation passe par une étape très cruciale appelée segmentation qui permet de séparer l'image de texte en un ensemble de segments de base (des graphèmes, des mots ou des chaînes numériques) selon l'architecture du système [6, 24].

Dans ce chapitre, nous présentons un état de l'art sur les systèmes OCRs. En plus, les différentes phases nécessaires pour la reconnaissance des chiffres manuscrits avec emphase.

1.1 Différents aspects de la reconnaissance

Dans cette section, nous allons présenter quelques aspects des systèmes de reconnaissance du manuscrit tels que la différence entre la reconnaissance et la production, les différents types de systèmes OCR.

1.1.1 Production et reconnaissance

La reconnaissance de chiffre manuscrit, consiste à faire la conversion de l'image de texte manuscrit en un fichier en format numérique, avec un minimum d'erreurs possibles. On peut aussi dire que La reconnaissance est le processus inverse de la production. La différence entre les deux processus (production, reconnaissance) est illustré par la figure 1.1 [24].

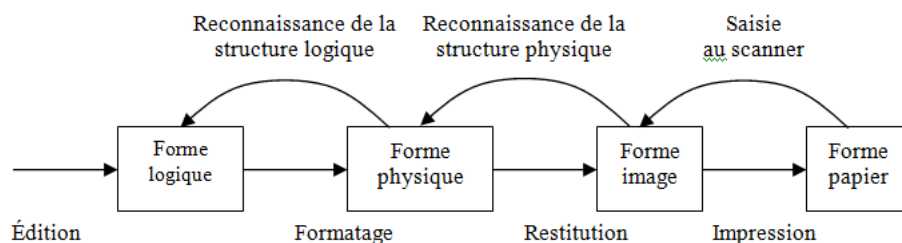


Figure 1.1: Processus de production et de reconnaissance de documents[32] .

La reconnaissance dépend considérablement des types de dispositifs d'acquisition ,Il existe plusieurs difficultés, telles que : la qualité du document , L'impression , La discrimination entre les formes. Par conséquent, il existe deux type d'acquisition : acquisition en-ligne et acquisition hors-ligne.

1.1.2 Reconnaissance en-ligne / hors-ligne

La reconnaissance en-ligne (online) et la reconnaissance hors-ligne (offline) diffèrent par les outils d'acquisition et les algorithmes de reconnaissance utilisés :

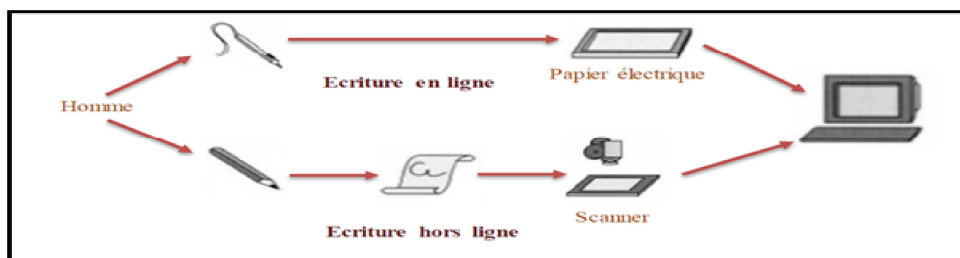


Figure 1.2: Différence entre les deux types de reconnaissance (online , offline) [24].

1.1.2.1 Reconnaissance en-ligne

La reconnaissance en-ligne diffère de la reconnaissance hors ligne lors de l'acquisition. la première faisant une acquisition en temps réel à l'aide d'un stylo ou une tablette électronique. Mais, la seconde, utilise un scanner ou camera comme l'illustre la figure 1.2. La reconnaissance en ligne est plus coûteuse mais plus efficace[10].

Nous devons aussi savoir que l'écriture prend l'apparence d'une paire de signaux temporelle numérisés. Les informations collectées consistent en une séquence ordonnée

de points (définis par leurs coordonnées) échantillonnés à un taux fixe (voir figure 1.3).

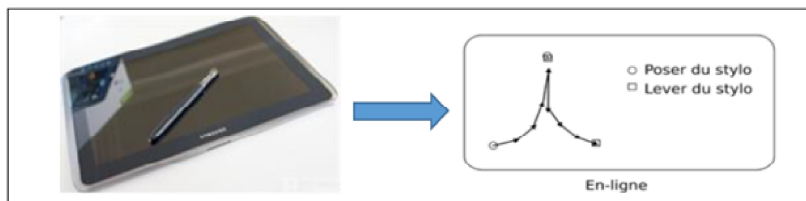


Figure 1.3: Exemples d'allographes des caractères [10].

1.1.2.2 Reconnaissance hors-ligne

Comme nous l'avons expliqué dans la section précédente, ce mode est différent de la reconnaissance en ligne et cette différence est présentée lors de la phase d'acquisition. Où, dans ce mode, l'acquisition provient d'un scanner ou d'un appareil photo. Le résultat de l'acquisition est donc une image du texte qui va être traitée comme une image binaire ou en niveaux de gris, qui contient souvent des effets secondaires comme le bruit et la perte d'informations temporelles d'ordre de points.

Donc, de tout ce qui précède, ce mode est bien plus difficile à traiter que le précédent. mais il est nécessaire dans des différents domaines tels que l'acquisition des formulaires, l'acquisition des identifiants numériques, l'acquisition des fiches de soins, le traitement automatique des adresses postales ... etc [10].

1.2 Phases d'un OCR

Généralement, la reconnaissance de l'écriture manuscrite est composée de grandes phases. Dans ce qui suit, nous allons expliquer chacune d'elle avec un schémas bien détaillé, ces phases sont : acquisition, pré-traitements, segmentation, extraction des caractéristiques, suivis éventuellement d'une phase de classification, parfois une phase supplémentaire de post-traitements (voir la figure 1.4).

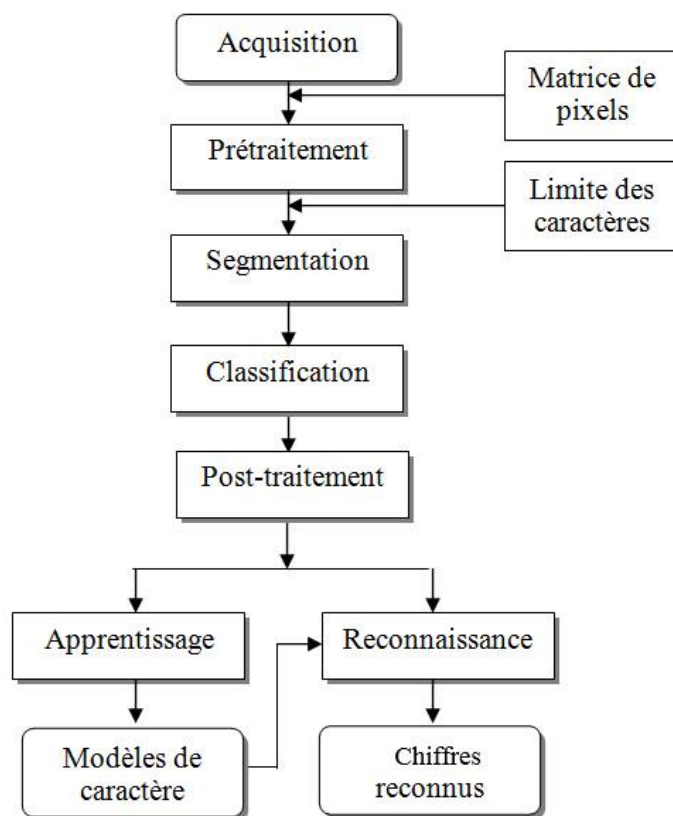


Figure 1.4: Schéma général d'un système de reconnaissance des chiffres manuscrits[32].

1.2.1 Phase d'acquisition

La phase d'acquisition consiste essentiellement à capturer l'image d'écriture manuscrite à l'aide de capteurs physiques et à la convertir en élévations numériques adaptées au système de traitement informatisé avec un minimal de dégradations possible.

Durant cette phase, malgré la bonne qualité de fonctionnement des systèmes d'acquisition, des parasites puissent apparaître et provoquent un bruit de fond hétérogène. Cela est dû à la nature de la texture, au lieu de travail et à son éclairage [3].

1.2.2 Phase de pré-traitement

Le but de la phase de pré-traitement est de réduire la quantité d'information pour ne conserver que les informations pertinentes. L'image résultante du processus de numérisation peut contenir une certaine quantité de bruit . Les chiffres peuvent être tachés ou cassés. Certains de ces défauts peuvent causer un taux de reconnaissance faible.

Parmi les opérations de pré-traitement les plus utilisées on peut citer : la binarisation, le lissage (bruit) , la normalisation (normalisation de taille , normalisation de Redressement) ,squelettisation [10].

1.2.2.1 Binarisation

La binarisation est une opération qui permet le passage d'une image de niveaux de gris à une image binaire composée de 2 valeurs 0 et 1, plus simple à traiter. L'opération de binarisation permet de séparer l'information manuscrite du fond de l'image à l'aide d'une méthode de seuillage, et de réduire la quantité d'information, tout en conservant le signal à traiter autant que possible [6].

1.2.2.2 Lissage

Le lissage est une transformation locale qui ne tient en compte que du voisinage de chaque point de l'image. La transformation la plus fréquemment utilisée pour le lissage des textes hors-ligne, consiste à parcourir l'image pixel par pixel, au moyen d'un masque de dimensions 3x3 et à modifier éventuellement la valeur du pixel examiné en fonction de ses huit pixels voisins (nord, nord-est, est, etc.). Ce masque peut être appliqué, par exemple, pour le bouchage des trous et l'élimination des pixels isolés[10].

1.2.2.3 Normalisation

L'un des principaux problèmes affrontés lors de la reconnaissance de chiffres manuscrits est la taille de l'image acquise et le redressement des lignes, ce qui pose un problème majeur pour la segmentation. Pour cela, dans n'importe quel système de reconnaissance de l'écriture manuscrite l'étape de normalisation est requise lorsque la taille varie et que le style des chiffres manuscrits se change d'un auteur à un autre.

1.2.2.4 Normalisation de taille

Cette étape permet de changer les images des chiffres manuscrits à une taille standard. Mais la normalisation a des conséquences dont l'une est de donner une légère déformation aux images. Ce qui n'est pas approprié pour certains systèmes et méthodes de reconnaissance qui est sensible aux petites variations, à la taille et à la position, comme dans le cas de "templates matching"[1, 3].

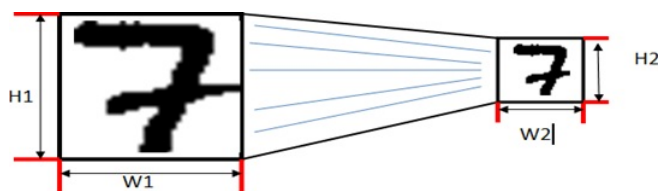


Figure 1.5: Exemple d'un chiffre manuscrit normalisé[1].

1.2.2.5 Redressement de l'image de document

Le redressement ou on peut dire l'inclinaison, est une technique de correction d'angle qui a été étendue pour corriger le dés-alignement du papier dans le scanner et également pour

incliner l'inclinaison interne de certaines parties du document qui peuvent apparaître.

Plusieurs algorithmes de reconnaissance ne donnent pas de bons résultats si le document n'est pas bien ajusté. Il existe deux types de redressement [31].

- **Le redressement de la ligne de base** : Le fonctionnement de ce redressement est de redresser les chiffres horizontaux en utilisant une transformation géométrique de la rotation isométrique des points de l'image.
- **Le redressement des écritures penchées** : Ce redressement facilite la segmentation préliminaire des chiffres. Il travaille pour trouver l'angle d'inclinaison moyen puis effectuer une transformation géométrique de type cisaillement de l'image.

1.2.2.6 Squelettisation

La squelettisation simplifie la forme du texte et réduit la quantité de données à traiter et permet d'obtenir le squelette de chiffre connecté de l'image. L'épaisseur de toute ligne dans les chiffres après la squelettisation est égale à un pixel [6]. La figure suivante illustre l'étape de la squelettisation

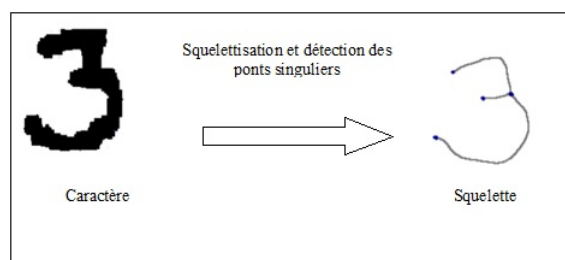


Figure 1.6: Exemple de squelettisation[1].

1.2.3 phase de segmentation

Lors de la conception d'un système de reconnaissance de chiffres, l'étape la plus importante est la segmentation de la chaîne de chiffres pour obtenir des chiffres isolés. Cette étape est un problème non trivial, en raison de plusieurs complexités. Le premier est la nature inhérente du script qui est cursif et en même temps le temps s'est chevauché. Le second est le degré élevé de la variation dans les styles d'écriture produits par les écrivains. Les systèmes de segmentation peuvent être divisés en deux approches : implicite et explicite[31].

1.2.3.1 Explicite

Dans l'approche explicite (également appelée discrète), une étape de sélection des points de segmentation les plus probables est effectuée. Certaines liaisons multiples ou à contact prolongé imposent de séparer les composantes selon un chemin. Les trajectoires de

segmentation sont généralement obtenues par des points caractéristiques dérivés d'une analyse de contour de la forme du squelette ou de l'amincissement du fond d'une analyse bidimensionnelle de la courbe, ou d'une combinaison analyse des contours / amincissement du fond [22, 18, 4].

1.2.3.2 Implicite

Pour contourner la difficulté du problème du choix des points de segmentation, les approches à segmentation implicite (ou continues) considèrent tous les points de traçabilité comme des points potentiels de segmentation. Dans cette approche, La segmentation et la reconnaissance sont effectuées conjointement, d'où le nom parfois utilisé de segmentation-reconnaissance intégrée. Ce sont des méthodes de fenêtre glissantes qui analysent la séquence de chiffres à l'aide d'une fenêtre de taille fixe, en extrayant les fonctionnalités de bas niveau. Les fenêtres sont analysées soit par un classifieur classique, soit par des modèles dynamiques. Cette fois-ci, il n'y a pas de pré-segmentation de chiffres. La segmentation s'effectue pendant la reconnaissance. Le système recherche dans l'image, des composants ou des groupements de graphèmes qui correspondent à ses classes de chiffre [18, 4].

1.2.4 Extraction des caractéristiques

Le but principal de la phase d'extraction caractéristiques est de pouvoir traiter la taille important des données qui sera fourni au système. Cette phase est critique pour construire un système de reconnaissance. Mais cette phase a un problème majeur accompagnant la méthode d'extraction, ce problème est la perte des données. Alors nécessaire de faire un compromis entre la quantité et la qualité des données [12].

La réduction du nombre de caractéristiques présente de multiples avantages[7] :

- Améliorer la visualisation d'informations.
- Améliorer la compréhension d'informations.
- Temps d'apprentissage réduit.
- Temps de classification réduit des systèmes.
- Performances en classification amélioré.
- Rétrécir la taille des bases d'apprentissage

1.2.4.1 Primitives structurelles

Les primitives structurelles sont généralement basée sur la représentation linéaire du caractère [12]. Cette primitive est extraite principalement à partir de représentation squelette ou contour de la forme [7] et pas de l'image brute. Il s'agit fondamentalement

de la décomposition du caractère en segments de droites et courbes, des pentes, la hauteur et la largeur du caractère . . . etc.

1.2.4.2 Primitives statiques

Ils contiennent des données sur la distribution des pixels qui sont utilisés dans différentes régions de l'image des chiffres et cette caractéristique est appelée zonage (voir la figure 1.7), on a aussi d'autres caractéristiques qui s'appellent histogramme (vertical ou horizontal)[6] dont lequel on cherche à trouver un moyen qui nous permet de représenter l'image en nombre de pixels sur chaque ligne ou colonne[10]. Les primitives préliminaires dérivent un ensemble de mesures du modèle pour décrire une forme :

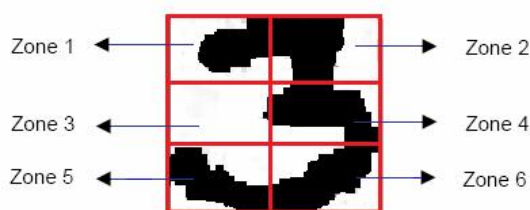


Figure 1.7: Exemple de squelettisation[23].

1.2.4.3 Primitives globales

Ces primitives sont dérivées de la distribution des pixels, ce qui signifie qu'elles dépendent d'une transformation globale et de la totalité des pixels d'une image [12], elles produisent trois types de caractéristiques telles que : moments invariants, projections et profils.

Représenter par le squelette ou le contour d'un caractère sous la forme d'une chaîne de codes de direction (code de Freeman) et elles sont considérées comme l'une des transformations les plus simples. Il existe de nombreuses méthodes de transformation, on peut citer les transformations de Hough et Fourier et les ondelettes[6].

1.2.4.4 Primitives topologiques

Cette catégorie dépend véritablement de mesures physiques telles que : la hauteur, la largeur et le rapport de ces deux quantités, ainsi que de fonctions complexes telles que le codage de profil. Pour définir les profils, vous devez respecter les quatre axes naturels (gauche, droite, haut et bas), ainsi que les angles (profils à 0 degré, 45 degré, 90 degré et 135 degré)[4, 12].

1.2.5 Classification

Après avoir terminé la phase d'extraction des caractéristiques, nous passons maintenant à la phase suivante, qui a pour tâche principale de classer les différents segments selon leurs caractéristiques [3]. Cette phase passe par deux étapes.

- l'apprentissage;
- la décision.

1.2.5.1 Etape d'apprentissage

L'apprentissage a lui-même deux familles majeures: supervisé ou non-supervisé. Nous allons commencer par l'apprentissage supervisé qui est utilisé dans le cas où les différents paramètres sont définis par l'utilisateur du système. Dans le second cas, les classes ainsi que les paramètres du systèmes sont réglés automatiquement [33].

1.2.5.2 Etape de décision

Elle consiste de choisir l'étiquette de la classe le plus proche parmi les modèles de référence. Le problème est d'attribuer une forme inconnue à l'une des classes obtenues pendant l'apprentissage. Le résultat de la décision obtenue peut être considéré comme un "avis" indiquant si la forme appartient ou non aux modèles d'apprentissage [31].

La reconnaissance peut mener au succès si la réponse est unique (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). Enfin, cela peut conduire à un rejet de la forme si aucun modèle ne correspond à sa description. Dans les deux premiers cas, la décision peut être accompagnée d'une mesure de vraisemblance, également appelée score ou taux de reconnaissance.

Le classificateur peut être incapable de reconnaître certains formes si toutes les données sont utilisées pour l'apprentissage et le test en même temps. Pour cela, nous avons besoin de trois ensembles de données pour améliorer les performances du classificateur. Ils peuvent être organisés comme suit : le premier pour l'apprentissage, le second pour la validation et le dernier pour le test. La question que l'on peut se poser est de savoir pourquoi nous avons besoin des données de validation. La réponse peut être telle que nous l'utilisions pour améliorer les performances et poursuivre le processus d'apprentissage. Nous nous arrêtons lorsque ce processus n'apporte plus d'amélioration. Parce que si nous n'arrêtons pas à ce niveau, nous avons un problème de sur-formation "sur-formation" qui nous a coûté une augmentation de la capacité de généralisation du classificateur [18, 4].

1.2.5.3 Approche structurelle

Cette approche est basée sur l'étude statistique des mesures de la forme que nous pouvons appliquer pour la reconnaître[4]. Mais pour bien comprendre les lois de probabilité des différentes classes, il faut un grand nombre d'exemples. L'étude des classes de caractérisation statistique et de distribution dans un espace métrique permet de prendre une décision du type : probabilité plus élevée d'appartenir à une classe. Cette approche

tire parti des méthodes d'apprentissage automatique qui reposent sur des bases théoriques bien connues, comme indiqué ci-dessous :

- Les méthodes paramétriques (les réseaux de neurones , la règle de Bayes ... ect)[4, 20].
- Les méthodes non paramétriques (méthode des k plus proches voisins, ... ect)[18, 4].

1.2.5.4 Approche stochastique

Les méthodes structurelles reposent sur l'extraction de primitives en tenant compte des informations structurelles (la structure physique des caractères) [12]. En général, les approches syntaxiques ou structurelles donnent la possibilité de décrire des formes complexes à partir de formes élémentaires. La différence que l'on peut distinguer entre les méthodes statistiques et ces méthodes. Est-ce que les caractéristiques sur lesquelles travaillent ces méthodes sont des formes élémentaires (c'est du type topologique) et non des mesures. On distingue plusieurs techniques telles que les méthodes de tests, la comparaison de chaînes, les structures de graphes, le calcul de distance d'édition entre deux chaînes et la programmation dynamique, les structures syntaxiques, ... etc[?].

1.2.5.5 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 classificateurs en couches. Pour obtenir des résultats plus précis et vaincre les faiblesses de chaque approche. Les résultats obtenus sont meilleurs que ceux qui auraient été obtenus si l'application de chaque approche était séparée, telles que les approches combinées pour former l'approche intégrée [10].

Nous fournissons un exemple expliquant les avantages cette approche qui fusionne les approches statistiques et structurelles.

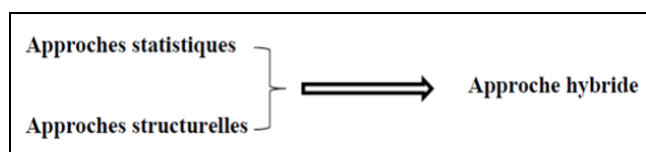


Figure 1.8: Figure 1.6. Exemple d'hybridation entre deux approches (statistiques et structurelles).

1.3 Méthode de classification statistique

Ces dernières décennies les méthodes statistiques ont gagnées un grand essor vu leurs performances. Dans ce qui suit nous allons présenter quelques classificateurs statistiques très connus.

1.3.1 Réseaux de neurones

Les réseaux de neurones sont une dénomination plutôt floue qui couvre un ensemble de mécanismes de calcul inspirés initialement par des modèles de fonctionnement du système nerveux que les techniciens considèrent comme une source d'inspiration pour la construction de systèmes automatiques[3, 26].

Ils présentent une alternative aux architectures classiques, et ils ont été développés après un grand effort. Ils sont capable de travailler sur des structures parallèles et avec une performance de classification sur des phénomènes non linéaires.

Un réseau de neurone ne fonctionne pas qu'après avoir appris suffisamment de connaissances sur les résultats souhaités des données entrées; le développement de ces réseaux repose principalement sur l'apprentissage et les lois de modification des poids de connexion [26]

1.3.2 Machines à Vecteur Support (SVMs)

L'algorithme des machines à vecteurs de support est un ensemble de techniques d'apprentissage supervisées destinées à résoudre des problèmes de la classification binaire supervisée et de la régression. Les SVMs reposent sur deux idées:

- La notion de marge maximale et la notion de fonction du noyau. La marge maximale est utilisée pour les problèmes de classification linéaire . Elle représente la distance entre la limite de séparation et les échantillons d'apprentissage les plus proches. Ces derniers sont les vecteurs supports. Les fonctions noyau sont employées dans le cas des problèmes de la classification non-linéaire pour transformer l'espace de représentation des données d'entrées en un espace de plus grand dimension dans lequel il est probable qu'il existe de séparateurs linéaires [33].
- 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 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 [6]. 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 [13]

1.3.3 K plus proches voisins

Le classificateur du K le plus proche voisin (KNN: **K**-Nearest Neighbor) est un classificateur simple basé sur le calcul de la distance entre les exemples d'apprentissage et les exemples de test, La règle euclidienne est le principe de base de la mesure de distance à chaque étape de l'apprentissage [16], l'algorithme stocke les k meilleurs exemples de l'ensemble d'apprentissage ($kppv(x)$) qui sont proches à l'exemple de test x . Cet algorithme est souvent efficace s'il ya suffisamment d'exemples d'apprentissage, mais nécessite un temps de prédiction très long pour traiter tous les exemples et trouver les K meilleures résultat [31, 16].

Conclusion

Dans ce chapitre, nous avons présenté globalement les différentes phases d'un système OCR (système de reconnaissance de chiffres), les problèmes et les difficultés qui peuvent confronté un tel système. De plus, nous avons mentionnés quelques solutions proposées par plusieurs durant les dernières années. Aussi, nous abordons dans le prochain chapitre, la description détaillée de diffèrent étape de segmentation appliquées à la reconnaissance des chiffres manuscrits.

Technique de segmentation

Introduction

Le processus de segmentation consiste à diviser les éléments du document en unités primaires (lignes, mots, caractères (chiffre), graphèmes ... etc.). Dans le cas des chiffres manuscrit, cette phase est très cruciale pour avoir un bon fonctionnement du système de reconnaissance

Dans ce chapitre, nous présenterons des différentes approches et techniques de segmentation et quelques méthodes d'extraction de lignes de chiffres et chiffres isolés.

2.1 Problématique posée par l'écriture du cursif

L'image d'un chiffre manuscrit est représentée par un signal bidimensionnel dans lequel, aucune information d'ordonnancement n'est présentée, bien que par convention, il existe dans cette chaîne de chiffre une séquence de chiffre dont l'ordre logique d'interprétation (de la droite vers la gauche) [5].

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

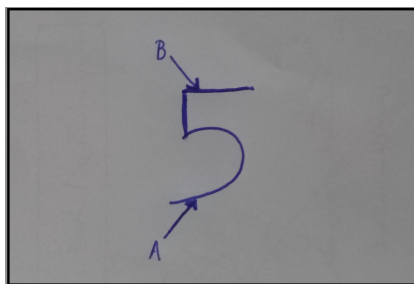


Figure 2.1: Chiffre cursif "5".

2.2 Concepts de segmentation de chiffres

Extraire les lignes et les chaînes de chiffres d'un document manuscrit est une tâche difficile. La difficulté vient des caractéristiques des documents manuscrits, surtout quand ils sont anciens, parce que ces documents contiennent des espaces irréguliers entre les lignes de chaînes de chiffres. Comme le montre la figure suivante:

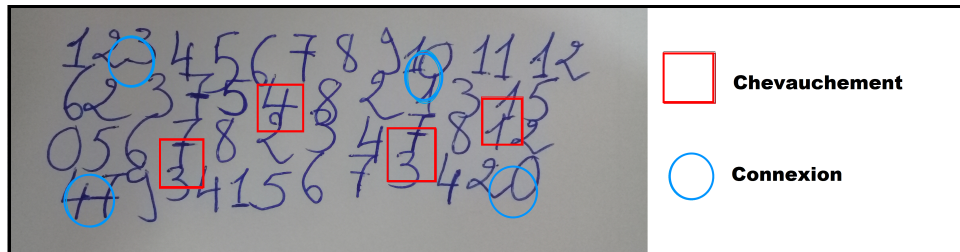


Figure 2.2: Exemple du chevauchement de chiffre.

L'une des premières étapes d'un système de reconnaissance de l'écriture manuscrite consiste à découper le document en lignes. Ce processus est relativement facile si les chiffres ne sont pas inclinés, et si les lignes ne se chevauchent pas.

Dans le cas d'un manuscrit qui n'est pas limité par le système de document général, l'écriture fluctue et peut avoir une grande tendance à l'horizontale : et deux lignes adjacentes peuvent se chevaucher, rendant la séparation difficile. Aussi ce qui rend la division difficile la variété de polices est la présence d'un grand nombre de points, ce qui complique cette tâche

La plupart des études sur la segmentation d'une page s'appuient sur une décomposition de l'image en composantes connexes.

2.3 Techniques de segmentation

Dans cette section, nous allons présenter les différentes techniques de segmentation proposées durant ces dernières années.

2.3.1 Segmentation à partir du squelette

Dans ce cas, l'image du texte doit être converti en une représentation en squelette. Ensuite, à travers ce squelette, des points de connexion sont identifiés entre les chiffres. Ce processus donne des calculs de courbes et d'angles sur un seuil spécifié. Le taux d'erreur dans cette approche est de 10. Cette méthode dépend de la qualité du squelette obtenu (voir figure 2.3) [9].

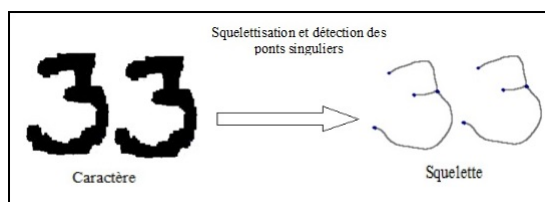


Figure 2.3: : Exemple de squelettes de chiffre.

2.3.2 Segmentation à partir du contour

La segmentation à partir du contour, consiste à sélectionner les points idéaux pour le rognage des graphèmes, basés sur des extrema locaux du contour, liés selon un critère de proximité.

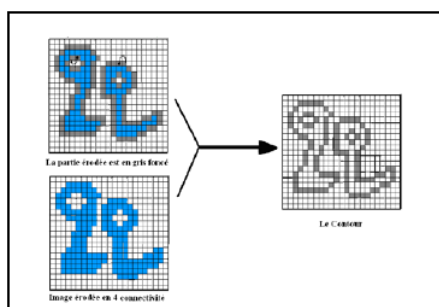


Figure 2.4: : Exemple de squelettes de chiffre [30].

Le développement par essais et erreurs est recommandé en ce qui concerne les schémas de réglage flous pour la reconnaissance de l'écriture manuscrite, et il suit également la qualité des caractères manuscrits, ce qui donne des résultats très étranges lorsque la qualité est médiocre. (voir figure 2.4) [30].

2.3.3 Segmentation à partir des histogrammes de projection verticales

Ce processus est fait en calculant un histogramme de projection horizontale pour déterminer les différentes lignes de chaîne de chiffre et l'historgramme de projection verticale afin de déterminer les différents caractères [16]. Cette approche est proposée par le Professeur B. Yanikoglu et P. Sandon [5].

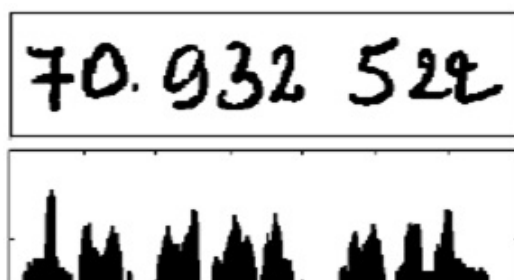


Figure 2.5: Exemple d'un histogramme de projection de chiffre.

2.3.4 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. Cette technique dépend de la division verticale de l'image de chiffre en fenêtres de taille fixe ou variable (voir la figure 2.6).



Figure 2.6: Exemple d'utilisation de la fenêtre glissante sur un chaîne de chiffre.

Enfin, la taille moyenne des lignes inférieure au seuil approuvé est choisie.

2.4 Niveaux de segmentation

L'extraction des nombres et leur détection constituent l'une des étapes les plus importantes du processus d'identification, car il est très important au moins d'identifier les mêmes numéros, tant que la bonne division permet de bien fournir les caractères au système, inversement, une mauvaise segmentation provoque une baisse du taux de reconnaissance

Généralement, nous pouvons distingués quatre niveaux de segmentation, comme suit(voir la figure 2.7):

- segmentation de la page.
- segmentation de le table de chiffres en lignes.
- segmentation de lignes en séquence de chiffres.
- segmentation de séquence de chiffres en chiffres isolés.

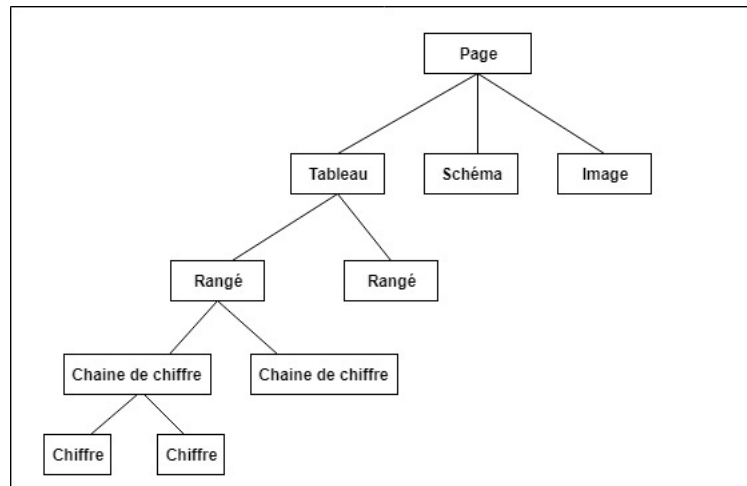


Figure 2.7: Illustration de l'arborescence du processus de segmentation.

2.5 Segmentation de page

Cette étape permet de définir les limites et les zones d'informations en fonction de leur forme physique sur chaque page. Il est généralement lié au statut des signes logiques permettant l'identification de la nature médiatique représentée dans chaque région (séquence de chiffres, image, dessin), ce qui permet également de diviser des régions de formes différentes et multiples (voir la figure 2.8) [14].

\$ 934.07	25	34177
\$ 175.08	47	34215
\$ 182.47	28	13346
\$ 54.32	4	4676
\$ 5,056.89	31	31111
\$ 3,500.10	40	31169
\$ 196.12	15	31270
\$ 4,011.65	27	5607
\$ 45.63	25	31492
\$ 1,465.29	28	12704
\$ 1,356.45	29	12419
\$ 250.75	1	12485
\$ 313.05	2	12544
\$ 107.56	7	11495
\$ 1,386.69	45	11911
\$ 6,813.93	50	11941
\$15,168.82	10	12096
\$ 262.31	15	31393
\$ 367.11	45	12289
\$ 4,408.27	47	5569

1518	143	414	3026
1619	80	515	3165
2557	107	471	2379
1410	113	526	2437
2125	102	3	418
2364	150	547	2925
2072	86	405	2359
2354	93	413	3258
1743	136	483	2446
1819	122	360	3271
2371	124	466	2437
2646	103	514	3102

Figure 2.8: Exemple de segmentation de page.

2.5.1 Segmentation en lignes

Cette segmentation consiste à examiner les structures linéaires numériques et à affiner leur segmentation en exploitant l'information sur la hauteur des composantes connexes numérique. L'idée consiste à utiliser la technique de projection pour segmenter toute structure linéaire susceptible de contenir plus qu'une ligne (voir la figure 2.9) [11].

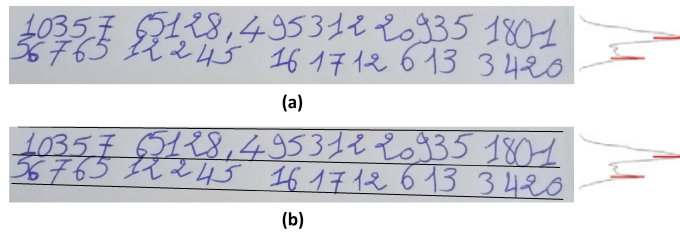


Figure 2.9: Exemple de segmentation en lignes d'une structure tableau : (a) le tableau et le profil lissé de la projection horizontale en rouge, avec la position de tous les maxima, (b) exclusion des fausses détections et détermination des bords de lignes [11].

2.5.2 Segmentation en séquence de chiffres

La méthode de regroupement est utilisée en supposant que les espaces dans la séquence de chiffres sont plus petits que les espaces entre les séquences de chiffres imbriqués dans chaque ligne. Les composants associés sont ensuite extraits et ceux qui se chevauchent horizontalement sont regroupés. Les distances horizontales entre les bords des boîtes liées sont ensuite déterminées pour les composants contigus [11].

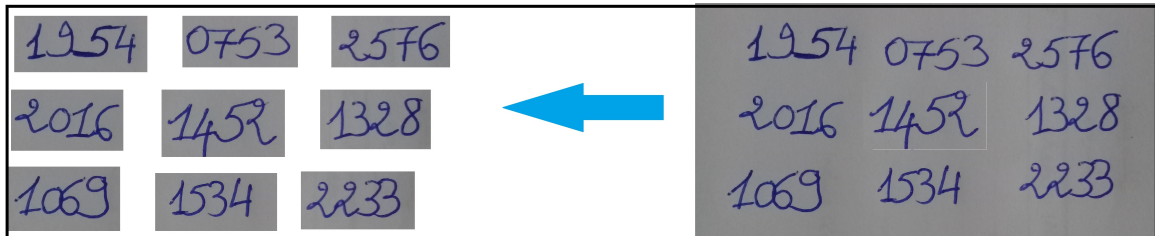


Figure 2.10: Exemple de segmentation en séquence de chiffres d'un document. Les rectangles bleus délimitent les séquence de chiffres extraits [11].

2.5.3 Segmentation en chiffres isolés

La séparation de chiffres est effectuée en analysant l'image des chiffres séquentiels et en la segmentant en sous-images de chiffres individuels. C'est l'un des processus de décision dans un système de reconnaissance optique de chiffres. Le but de ce système est de décider si l'élément isolé de l'image (chiffre, etc.) est vrai ou non.

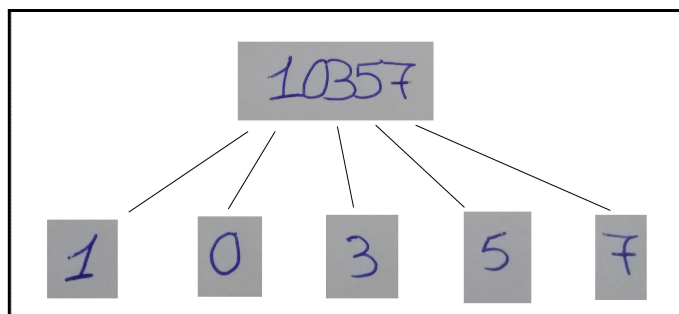


Figure 2.11: Exemple de segmentation en chiffres isolés [11].

2.6 Les travaux antérieurs

Dans cette section, nous allons voir une comparaison entre les quelques systèmes, en essayant de présenter pour chacun l'approche ou méthode proposée, les avantages et les inconvénients (voir le tableau 2.6).

Tableau 2.1: Comparaison entre quelques travaux antérieurs.

N°	Article	Approche	Méthodologie	Avantages	Inconvénient
1	Reconnaître les chiffres manuscrits à l'aide de Produits hiérarchiques des experts G. Mayraz et G. E. Hinton[21]	Produits hiérarchiques des experts.	<ul style="list-style-type: none"> - générer des modèles en évaluant des modèles d'apprentissage distincts pour chaque catégorie de chiffres. - en comparant les probabilités non normalisées des images de test selon les 10 modèles de classe spécifiques différents. - améliorer les performances discriminatoires en formant une hiérarchie de modèles distincts pour chaque classe de chiffres. - Chaque modèle de la hiérarchie apprend une couche de détecteurs de caractéristiques binaires qui distribue la probabilité des vecteurs d'activité dans la couche inférieure. 	<ul style="list-style-type: none"> - peut produire des hiérarchies efficaces de modèles génératifs de données de grande dimension - Le taux de reconnaissance atteint 98.3/100 sur la base de donnée MNIST 	<ul style="list-style-type: none"> - pour obtenir 98,3/100des résultats, il nous faut 90 - il faut un grand ensemble de données pour obtenir des résultats satisfaisants.

continue sur la prochaine page

Table 2.1 – Suite de la page précédente

N°	Article	Approche	Méthodologie	Avantages	Inconvénient
2	Improved method of handwritten digit recognition tested on MINIST database kussul.E Baidyk.[17]	classifieur neuronal Limited Réceptive Area (LIRA).	<ul style="list-style-type: none"> – contient trois couches de neurones . – entrée : connecté à la couche associative et la connexion n'est pas une connexion aléatoire modifiable. – couche associative : connecté à la couche de sortie avec une connexion qui prend en charge l'apprentissage. 	<ul style="list-style-type: none"> – Le processus d'apprentissage converge assez rapidement – taux haut de reconnaissance 99.59 	<ul style="list-style-type: none"> – Ce classificateur n'utilise pas d'opérations en virgule flottante. – Ce classificateur n'utilise pas multiplication. – opérer sur de grandes et petites bases de données.
3	les caractéristiques de direction extraites pour la réduction de la dimensionnalité. K-ième voisin le plus proche Wu et Zhang [27]	K-ième voisin le plus proche, gaussiens et SVM	<ul style="list-style-type: none"> – utilise les caractéristiques directionnelles extraites pour la réduction de la dimensionnalité.en mélangeant SVM, les modèles de Gaussien et de K-ème voisin le plus proche permettent de prouver que ce sont les meilleurs candidats pour les fonctionnalités . 	<ul style="list-style-type: none"> – utilise plusieurs methode pour une meilleur resultat – * plus haut taux 98.81/100 	<ul style="list-style-type: none"> – La methode est plus compliqué . – test sur une seul base de donnes MNIST.
4	Ebrahim et Jampour [15]	l'Histogramme de gradient Orienté (HOG), SVM,	Utilisation de HOG comme un descripteur de caractéristique et de plus SVM comme classificateur .	<ul style="list-style-type: none"> – HOG est un descripteur de fonction très utile pour les chiffres manuscrits – Taux plus haut 97,25/100de reconnaissance. 	<ul style="list-style-type: none"> – Taux de reconnaissance moins que l'autre méthode . – utilisant une seul base de donnés .

continue sur la prochaine page

Table 2.1 – Suite de la page précédente

N°	Article	Approche	Méthodologie	Avantages	Inconvénient
5	Classification k-NN des caractères manuscrits via la corrélation GAT accélérée Wakahara et Yamashita[28]	GAT Global affine transformation Et KNN	<ul style="list-style-type: none"> – il utilise la méthode de corrélation GAT. Une nouvelle mesure d'assimilation NNDEGD (Nearest Neighbors Distance of Equi-Gradient Direction) . – Le paramètre NNDEGD est un paramètre de la fenêtre de fonction gaussienne utilisée dans la méthode de corrélation GAT. Il est considéré comme la distance minimale entre deux points, le premier dans une image et le second dans une autre image, et les deux images ont le même sens de dégradé. – ils ont étendu la méthode de corrélation GAT pour gérer le changement de largeur de ligne en plus de la transformation affine. – la version appliquée de la corrélation GAT dans la classification de FC-NN est la version originale et étendue 	<ul style="list-style-type: none"> – la grande réduction de la complexité et de la mémoire dans la charge de calcul impliquée. – taux de reconnaissance atteint 99.51 	<ul style="list-style-type: none"> – dans le cas du bruit, le calcul de la distance ne peut pas être exact. – le test dans une seule base de données MNIST.

continue sur la prochaine page

Table 2.1 – Suite de la page précédente

N°	Article	Approche	Méthodologie	Avantages	Inconvénient
6	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree, NSF awards, Lee et al,[19]	Réseaux neurons	<ul style="list-style-type: none"> – réseaux de neurones profonds en utilisant des opérations de pooling : max pooling et la moyenne pooling . – il fusionne les fonctions de regroupement sous la forme d'une arborescence structurée contenant des filtres de regroupement. Chaque opération de pooling améliore les résultats obtenus . 	<ul style="list-style-type: none"> – exploration sûre des approches. – exploration rapide des approches. – Le taux de reconnaissance atteint 99.71 	<ul style="list-style-type: none"> – utilisation d'une seule base de données. – consommation de ressources de mémoire .
7	Classification k-NN des caractères manuscrits via la corrélation GAT accélérée Wakahara et Yamashita[29]	GAT Global affine transformation Et KNN global projection transformation de mesure d'apparement (GPT)	<ul style="list-style-type: none"> – le travail est d'abord sur le développement d'une méthode d'accélération GPT de corréler apparement. – Deuxièmement, pour améliorer les performances de reconnaissance, ils ont utilisé la courbure des contours au degré apparent. La courbure est généralement utilisée comme caractéristique des caractères. Cependant, ils l'avaient appliqué comme poids dans le NNDEGD. 	<ul style="list-style-type: none"> – amélioration de performance de reconnaissance . – taux de reconnaissance plus haut 99.70 	exploitation de ressources .

continue sur la prochaine page

Table 2.1 – Suite de la page précédente

N°	Article	Approche	Méthodologie	Avantages	Inconvénient
8	Dundar et al [8]	réseaux con-volutional profond + Kmeans	cette méthode utilise un filtre non supervisé et propose une version améliorée de k-means Les expériences montrent que l'algorithme suggéré est plus performant que d'autres techniques utilisant des filtres non supervisés.	– le nombre de paramètres corrélés est réduit par rapport à des filtres similaires. – la précision de la catégorisation des tests est accrue.	– taux de reconnaissance est bas comparé à d'autres méthodes. – * utiliser un seul base de données MNIST

Conclusion

Ce chapitre fournit une vue d'ensemble des méthodes et techniques de segmentation des systèmes de reconnaissance de chiffre manuscrit, y compris l'identification des méthodes les plus célèbres dans la littérature: segmentation explicite, fragmentation implicite et approches mixtes. Enfin, une étude comparative entre quelques travaux déjà réalisés dans le domaine a été présentée en mettant l'accent sur les avantages et les inconvénients de chaque travail.

Conception et Architecture

Introduction

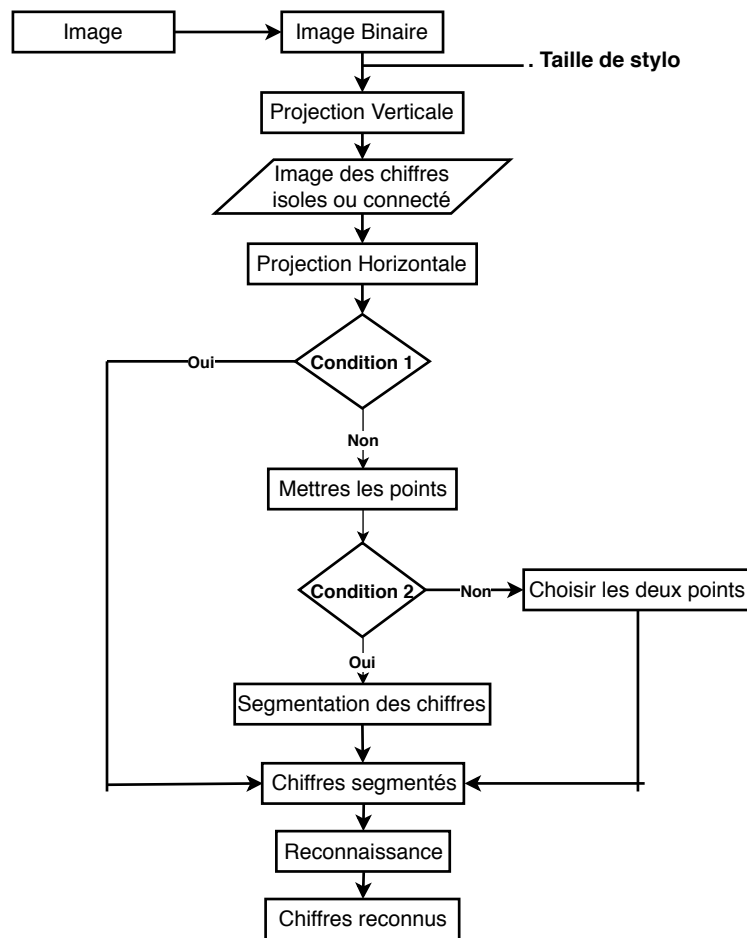
Dans les chapitres précédents, nous avons donné un aperçu des systèmes OCR en général et les systèmes de reconnaissance de chiffres en particulier, dans lesquels nous avons expliqué les différentes phases des systèmes de reconnaissance optique de chiffres. Après cela, nous avons vu les différentes méthodes et techniques de segmentation de séquence de chiffres.

Dans ce chapitre, nous verrons la structure générale de notre système, ainsi que les différentes étapes à suivre en détails. En outre, nous présentons les techniques utilisées pour vérifier la validité de la méthode proposée de segmentation et séparation des chiffres qui se chevauchent on applique en se basant sur des histogramme de projection verticale et horizontale, ainsi qu'un ensemble de points d'intérêts obtenu de l'image binaire des chiffres.

3.1 Architecture générale du système

L'idée principale du système est la segmentation de séquence de chiffres manuscrits à partir d'une image prise par camera ou scanner. Le système commence par la création d'une image binaire à partir de l'image acquise (voir la figure 3.2). Ensuite, estime la taille du stylo d'écriture de l'image binaire. par la suite, applique une projection verticale depuis laquelle il peut déterminer si il y a des chiffres chevauchés ou non. Dans le cas où il détecte un chevauchement entre les chiffres il applique une projection horizontale. Dans ce dernier, il fait un balayage vertical en calculant le nombre de pixels noir dans chaque ligne; si le nombre des pixels de tous les ligne est inférieur ou égal à la taille du stylo le segment d'image est considéré comme contenant des chiffres isolés; sinon il faut déterminer les points d'intérêts de cette segments afin de pouvoir segmenter les chiffres chevauchés. après cela, il faut voir si il n'ya pas des rouges dans la partie 2/3 de l'image

et elle ne contient pas de cycle; si oui on segmente les chiffres; sinon on choisit les deux points adjacents et on segmente les chiffres (voir la figure 3.1).



Condition1: nombres des pixels d'une ligne = Size .

Condition 2: il n'ya pas de point rouge dans la partie 2/3 de l'image et n'existe pas un cercle .

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

3.2 Génération de l'image binaire

La binarisation est la phase qui suit l'acquisition de l'image par un scanner ou une caméra, et consiste à transformer l'image en une image binaire. cette dernière est obtenu en considérant que chaque pixel noire est de la valeur 1 et chaque pixel blanche et de la valeur 0. Finalement, nous obtenant une matrice représentant l'image et qui ne contient que des 1 et 0 (voir la figure 3.2).

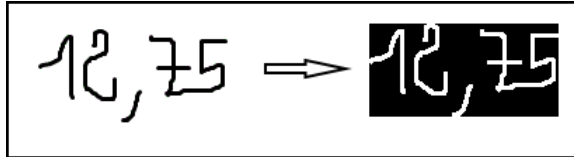


Figure 3.2: Transformation image originale en une image binaire.

3.2.1 Calcul de la taille du stylo

Cette phase consiste à balayer l'image horizontalement et à chaque étape on calcul le nombre de pixels noires. enfin, le nombre de pixels noires qui possède une grande occurrence est considéré comme la taille du stylo de l'écriture (voir l'algorithme 3.1).

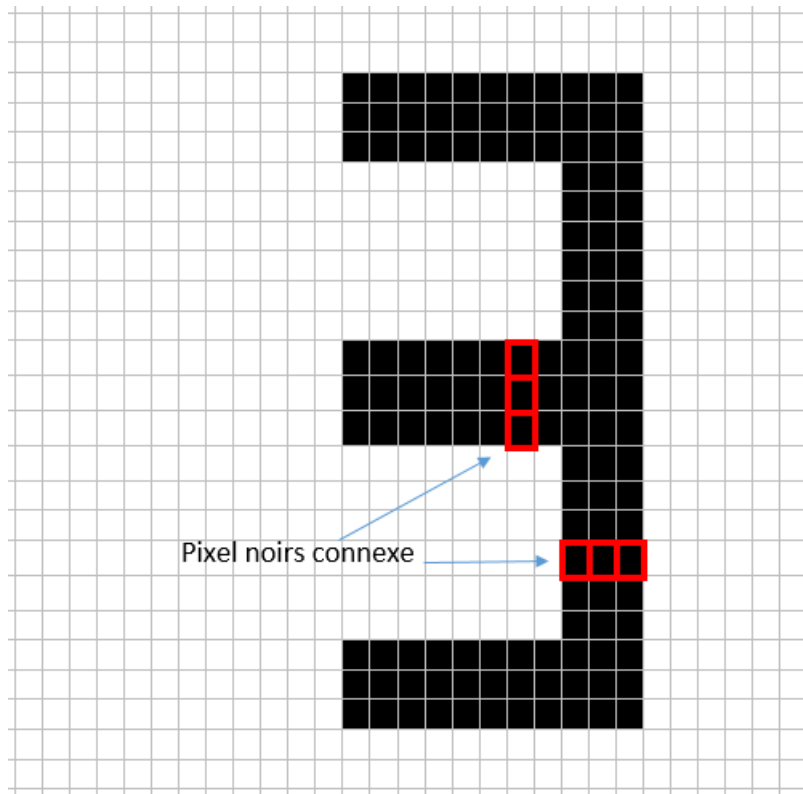


Figure 3.3: Grille de pixels d'un chiffres.

Algorithme 3.1 : Calcul de la taille du stylo

Fonction *Taille_stylo* (*img* : *image*): *Entier**L* : *List**Count* : *entier***Début****Pour** *i* ← 0 **Jusqu'à** *longueur*(*image*) **Faire****Pour** *j* ← 0 **Jusqu'à** *largeur*(*image*) **Faire****Si** ((*pixel*(*img*, *j*, *i*) == *Noir*)**Alors***Count* ← *Count* + 1**Sinon** (((*pixel*(*img*, *j*, *i*) == *black*) et ((*pixel*(*img*, *j*, *i*) == *Noir*)) et (*J* > 0)*ajout*(*Liste*, *Count*);*Count* ← 0;**FinSi****FinPour****FinPour**return (*Plus_occurent*(*List*));;**Fin**

3.3 Projection verticale

Cette phase est fondamentale avec la projection horizontale et la phase de détermination des points d'intérêts que nous allons expliquer dans les sections suivantes. L'objectif est de segmenter l'image verticalement de manière à pouvoir déterminer le nombre de chiffres existant dans l'image. Pour cela, on calcule le nombre des uns par colonne dans la matrice de l'image binaire (voir l'algorithme 3.2).

Algorithme 3.2 : Calcul de l'histogramme de projection verticale.

Fonction *Projection_Verticale* (*img* : *image*): *Tableau**Tab* : *Tableau d'entier*[]*Count* : *entier***Début****Pour** *i* ← 0 **Jusqu'à** *longueur*(*image*) **Faire***Count* ← 0**Pour** *j* ← 0 **Jusqu'à** *largeur*(*image*) **Faire***Count* ← *Count* + *Valpixel*(*img*, *j*, *i*)**FinPour***Tab*[*i*] ← *Count***FinPour**return (*Tab*);;**Fin**

Cette phase passe par deux étapes :

- Calcul de l'histogramme vertical;
- Segmentation des chiffres.

3.3.1 Calcul de l'histogramme vertical

cette étape est au cœur de toute la phase, dont lequel nous calculons les valeurs de l'histogramme verticale en calculant le nombre des uns de chaque colonne de la matrice de l'image binaire. Puis nous prenons l'index des valeurs de la table comme index de la colonne et à partir de là, nous décidons d'où segmenter l'image (voir la figure 3.4).

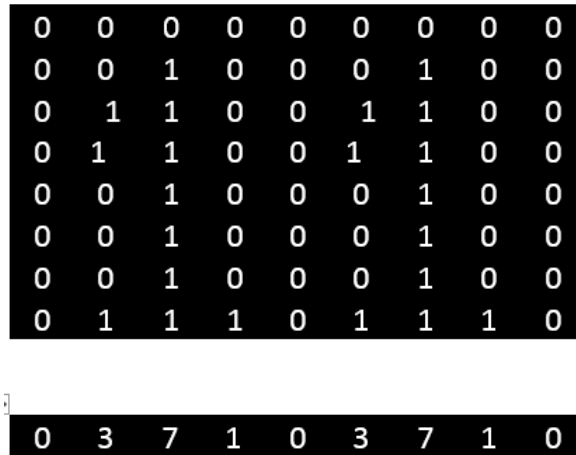


Figure 3.4: Illustration de la projection verticale de chiffres.

3.3.2 Segmentation des chiffres

La segmentation des chiffres consiste à utiliser le tableau de l'histogramme créé. Le système fait un balayage du tableau de droite à gauche et lorsque nous trouvons la valeur zéro, nous prenons cet index comme position de segmentation dans l'image binaire.

cette phase et ces deux étapes sont représentées dans l'algorithme 3.1 qui explique bien la manière de notre méthode.

3.4 Détection des chiffres isolés / connectés

Dans la phase précédente on explique comment l'image binaire est segmenter verticalement avec une matrice qui contient les pixels de l'image et le tableau qui contient la somme de chaque colonne. De la phase nous obtenons les résultats de la phase précédent et on peut distinguer deux résultats :

- images des chiffres isolés et/ou;
- images des chiffres chevauchés.

3.4.1 Images des chiffres isolés

Dans ce cas, le résultat est des images contenant des chiffres isolés, ce qui signifie que la segmentation verticale a réussi à donner des chiffres segmentés, ce qui signifie également que l'image ne contient aucun chiffres chevauchés. Même si le résultat est une image à chiffres isolés, nous devons passer à la phase suivante afin de confirmer qu'il n'y a pas de chiffres chevauchés.

3.4.2 Image des chiffres connectés

Dans ce cas, nous obtenons des images de chiffres chevauchés. Les chiffres qui se chevauchent dans l'image sont deux nombres qui partagent les mêmes pixels. Pour cela, il faut calculer l'histogramme de projection horizontale.

3.5 Calcul de l'histogramme de projection horizontale

L'objectif principal de cette phase consiste à calculer le nombre de pixels de chaque ligne et à le comparer à la taille du stylo. Donc, la détection du nombre de pixels de chaque ligne est similaire à la méthode que nous avons utilisée dans la segmentation verticale, mais dans cette phase nous n'allons pas segmenter l'image, la similarité est que nous allons additionner tout les 1s dans le binaire image qui est présentée dans une matrice. Ensuite, nous stockons la valeur de chaque ligne dans un tableau. Les valeurs stockées dans le tableau sont utilisées par la suite pour comparer si certaines d'entre elles sont égales à la taille du stylo. Après cela, nous classifions si cette image est une image à chiffres isolés, mais dans l'autre côté, si nous ne trouvons aucune valeur du tableau égale à la taille du stylo, nous classons l'image comme une image à chiffres chevaucher. Après la classification de ces images, nous passons à l'étape suivante, lorsque nous pouvons décider où la segmentation des chiffres qui se chevauchent doit être effectuée (voir l'algorithme 3.3).

Algorithme 3.3 : Calcul de l'histogramme de projection horizontale.

Fonction *Projection_Verticale* (*img* : *image*): *Tableau*

 | *Tab* : *Tableau d'entier* []

 | *Count* : *entier*

Début

 | **Pour** *i* ← 0 **Jusqu'à** *longueur*(*image*) **Faire**

 | *Count* ← 0

 | **Pour** *j* ← 0 **Jusqu'à** *largeur*(*image*) **Faire**

 | *Count* ← *Count* + *Valpixel*(*img*, *i*, *j*)

 | **FinPour**

 | *Tab*[*i*] ← *Count*

 | **FinPour**

 | return (*Tab*);;

Fin

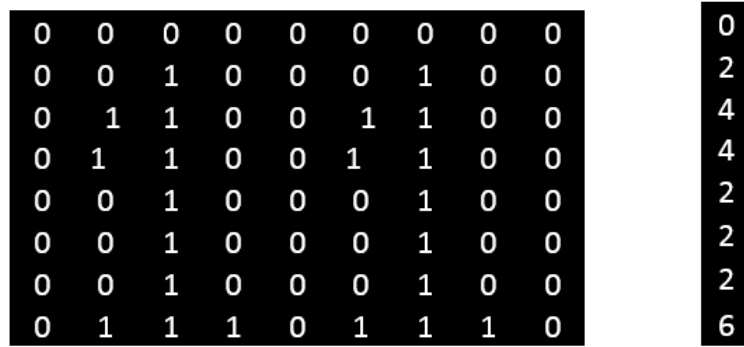


Figure 3.5: Illustration du calcul de la projection horizontale d'un chiffres.

3.6 Détermination des points d'intérêts

Cette phase est activée après la projection horizontale, si le résultat donné de la phase précédente est le nombre de pixels plus grand que la taille du stylo. Ce qui signifie que nous avons des chiffres qui se chevauchent dans l'image d'origine. En faisant des balayages des quatre côtés de l'image et en vérifions à chaque fois s'il y a un arc de cercle. S'il y a un arc, nous mettons un point au milieu de l'arc. La couleur du point se change en fonction du côté à partir duquel nous avons faire le balayage; ainsi, si c'est le côté droit, la couleur du point que nous avons placé au centre de l'arc est orange et à gauche nous avons placé un point bleu pour le côté supérieur un point vert pour le bas nous mettons un point rouge (voir la figure 3.6).

Après avoir placé les points d'intérêts dans chaque côté scanner, nous essayons de les relier pour former un cercle. L'image se fragmenté en trois parties en considérant la taille de l'image. Ensuite, nous vérifions dans la partie centrale si le point rouge existe et n'existe pas un cercle; dans ce cas, nous segmentons à partir de ce point rouge et s'il n'existe pas un point rouge et n'existe pas un cercle, nous passons à l'étape suivant appelé "choisissez deux points" selon l'algorithme 3.4.



Figure 3.6: Mettres les points sur les chiffres connectés.

Algorithme 3.4 : Construction du cycle.

Fonction *Projection_Verticale* (*img* : *image*): *Tableau*

 | *Tab* : *Tableau d'entier*[]

 | *Count* : *entier*

Début

Pour $i \leftarrow 0$ **Jusqu'à** *Taille de list* **Faire**

 | **Si** ($(point.Type == Red)$)

Alors

 | $X \leftarrow point.X;$

 | $Y \leftarrow point.Y;$

 | $Xd \leftarrow point.X;$

 | $Xg \leftarrow point.X;$

 | **Tantque** $valpixel(img, x, y) == blank$

Faire

 | **Tantque** $valpixel(img, x, y) == blank$

Faire

 | **Si** $valpixel(img, Xd, Y) == orange$

Alors

 | $orange \leftarrow Vrai;$

 | $Tab[3] \leftarrow Xd;$

 | $Tab[4] \leftarrow y;$

FinSi

 | $Xd \leftarrow Xd + 1;$

FinTantque

 | **Tantque** $valpixel(img, Xg, Y) == blank$

Faire

 | **Si** $valpixel(img, Xg, Y) == blue$

Alors

 | $blue \leftarrow Vrai;$

 | $Tab[0] \leftarrow Xg;$

 | $Tab[1] \leftarrow y;$

FinSi

 | $Xg \leftarrow Xg - 1;$

FinTantque

 | $Y \leftarrow Y - 1;$

FinTantque

 | **Pour** $i \leftarrow Y$ **Jusqu'à** $Y - 10$ **Faire**

 | **Pour** $i \leftarrow Xg$ **Jusqu'à** Xd **Faire**

 | **Si** $valpixel(img, i, Y) == Ver$

Alors

 | $Ver \leftarrow Vrai;$

FinSi

FinPour

FinPour

FinSi

FinPour

 return (*Tab*);

Fin

3.7 Segmentation des chiffres

Cette phase est où nous appliquons tout simplement la segmentation, après avoir obtenu les points et divisé l'image en trois parties. Comme nous l'avons mentionné précédemment, la segmentation est effectuée dans le point rouge. Ainsi, nous obtenons chaque chiffre dans une nouvelle image isolée. Enfin nous obtenons des chiffres segmentés.

3.8 Choisissez deux points

Dans cet étape, le système détecte qu'il y'a un cas particulier de chevauchement, où nous avons des chiffres qui se chevauchent mais ce n'est pas connecté. Pour gérer cette situation, nous avons une méthode récursive qui récupère le premier point de l'image et le dernier point et le met dans un vecteur ou une matrice, puis représente chaque chiffre dans une nouvelle image. la façon dont cet algorithme fonctionne pour obtenir le point (x, y), puis obtenir le masque qui contient x + 1 et y + 1, x-1, y-1, ces valeurs représentent les directions que prendra l'algorithme, et chaque fois un pixel dans le point spécifique est égal à 1 l'algorithme copie ce point dans la nouvelle image, et si le point pixel est égal à 0, l'algorithme arrête de passer de ce côté et continue jusqu'à ce qu'il copie chaque chiffre dans une nouvelle image. À la fin, nous pouvons dire que nous avons obtenu des chiffres segmentés (isolés), voir l'algorithme 3.5.

Algorithme 3.5 : Segmentation chevauchement

Procédure *Trace_points* (*imgA*, *imgN*: *image*; *X*, *Y*: *entier*)

Début

Si (*imgA*(*X*, *Y*) == *Noir*) et (*imgN*(*X*, *Y*) == *Blank*)

Alors

Trace_points(*imgA*, *imgN*, *x* - 1, *y* - 1);

Trace_points(*imgA*, *imgN*, *x* - 1, *y*);

Trace_points(*imgA*, *imgN*, *x* - 1, *y* + 1);

Trace_points(*imgA*, *imgN*, *x* + 1, *y* - 1);

Trace_points(*imgA*, *imgN*, *x* + 1, *y*);

Trace_points(*imgA*, *imgN*, *x* + 1, *y* + 1);

Trace_points(*imgA*, *imgN*, *x*, *y* + 1);

Trace_points(*imgA*, *imgN*, *x*, *y* - 1);

FinSi

Fin

3.9 Reconnaissance

Dans la phase de la reconnaissance nous utilisons la bibliothèque Deeplearning4j (Deep learning for java). Cette dernière, est une bibliothèque de programmation pour l'apprentissage en profondeur écrite pour Java, ainsi qu'une framework prenant en charge les algorithmes de deep learning. Elle inclut des implémentations de la machine de Boltzmann restreinte, du réseau de deep belief, de deep autoencoder, du réseau tenseur neural récuratif, de word2vec, doc2vec et GloVe. Ces algorithmes incluent tous des versions parallèles distribuées s'intégrant à Apache Hadoop et Spark. Deeplearning4j est un logiciel open source sous la licence Apache 2.0, développé principalement par un groupe d'apprentissage automatique basé à San Francisco et Tokyo et dirigé par Adam Gibson. Elle est prise en charge commercialement par la start-up Skymind, qui regroupe DL4J, Tensorflow, Keras et d'autres bibliothèques d'apprentissage en profondeur dans une distribution d'entreprise appelée Skymind Intelligence Layer. Elle est supporté par sa propre bibliothèque informatique numérique open source, ND4J, et fonctionne à la fois avec les unités de traitement centrales (CPU) et les unités de traitement graphiques (GPU).

Conclusion

Dans ce chapitre, nous avons commencé par clarifier la structure générale du système proposé. Ensuite, nous avons présenté chaque étape en détail afin de bien clarifier les différents articulation du processus de segmentation (entrelacé, connecté et isolé).

Dans le chapitre suivant, les résultats expérimentaux et démontrent l'efficacité de la technique proposée en termes de vitesse et de taux de segmentation.

Réalisation et validation

Introduction

Dans le chapitre précédent, nous avons parlé de l'architecture du système et de son fonctionnement, puis des détails de notre méthode et montré chaque phase et chaque étape. Lorsque nous obtenons des chiffres segmentés à partir d'une chaîne de chiffres, éliminant ainsi tout chevauchement entre les chiffres. Après cela, nous avons détaillé chacune des étapes et les conditions pour chacune d'entre elles et expliqué correctement le résultat obtenu à chaque étape. Enfin, le résultat obtenu sera toujours celui que nous voulions, quelle que soit la complexité de la saisie. Le résultat est un nombre réduit de chiffres lisibles par la machine.

Ce chapitre est le dernier chapitre où nous allons clarifier les outils avec lesquels nous avons travaillé pour atteindre notre objectif. Par outils, nous entendons une partie matérielle et une partie logicielle. Nous allons également parler dans ce chapitre des résultats obtenus et de la manière dont nous avons implémenté notre système à partir d'algorithmes. bien sûr, sans oublier de préciser pourquoi nous avons choisi chacun de ces logiciels que nous avons utilisés pour nous aider à atteindre ces résultats.

4.1 Choix du langage de programmation

Nous commençons donc ce chapitre en expliquant la partie logicielle où nous avons choisi le langage de programmation java et l'environnement de développement IDE. java a beaucoup de fonctionnalités intéressantes, mais celles qui nous ont fait choisir, qui sont si importantes et principales et servent notre projet:

- Langage orienté objet .
- Multithread.

- multiplateforme (Windows , Linux , Mac).
- Haute performance, Robuste, sécurisé.

4.2 Environnement de développement intégré (IDE)

L'environnement de développement intégré NetBeans (IDE) est une plate-forme open source, permettant de développer des applications avec java, C ++, PHP, JS,.. etc. NetBeans peut s'exécuter sous plusieurs systèmes d'exploitation, car il dépend essentiellement de la machine virtuelle Java (JVM), qui est considérée comme un médiateur traduisant tous les codes en code binaire que la machine peut comprendre. NetBeans a été initialement codé avec java.

Pourquoi utilisé netbeans:

- Gratuit et Open Source.
- Prise en charge dynamique du langage.
- Plate-forme extensible.
- Puissant GUI Builder.
- Support dédié disponible.

4.3 Caractéristiques de l'ordinateur utilisé

Ce programme a été créé dans un type d'ordinateur portable de périphérique (Dell) avec les spécifications suivantes

- Processeur : i5- M520 CPU @ 2.20 GHz.
- Mémoire installé (RAM) : 4.00GB.
- Type de système : système d'exploitation 64 bits.
- System d'exploitation : Windows 10 pro.

4.4 Interface et fenêtres

Lorsque l'application est lancée, la fenêtre principale apparaît dans la figure 4.1. Cette fenêtre contient une liste de boutons. Chaque bouton présente une tâche différente et certains de ces boutons mènent à une autre fenêtre. Cela entraîne donc généralement une autre fenêtre lorsque chacune de ces fenêtres a une tâche différente à effectuer.

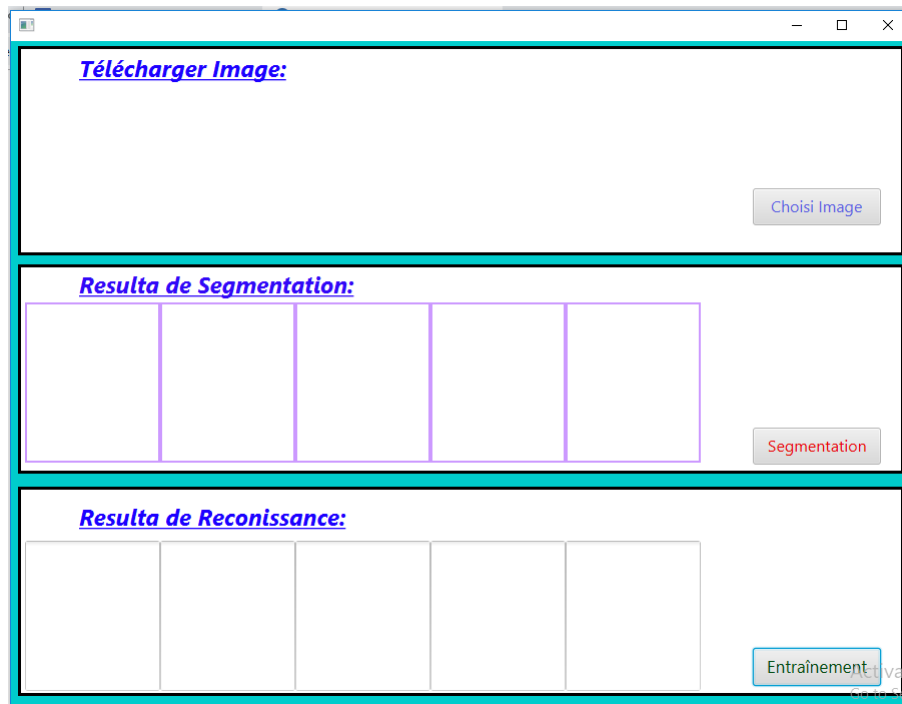


Figure 4.1: Interface principale de l'application.

La première fenêtre est divisée en trois parties : la première partie contient le détenteur de l'image d'origine sur laquelle nous allons travailler, la deuxième partie contient l'image segmentée et la troisième dernière partie contient un panneau indiquant les chiffres après la reconnaissance. Cette fenêtre contient le bouton "charger", pour charger l'image qui appartient à la première partie et le bouton "segmentation" pour segmenter l'image que nous trouvons dans la deuxième partie de la fenêtre. Enfin, cette fenêtre contient un autre bouton dans la troisième partie qui nous amène à une autre fenêtre utilisée pour la formation, voir la figure 4.7

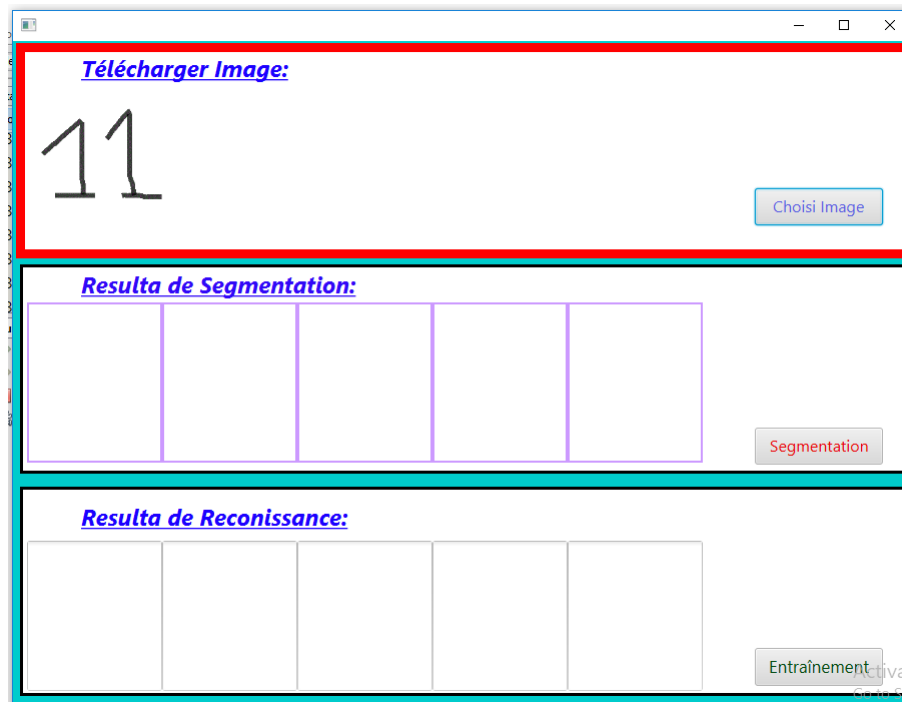


Figure 4.2: Fenêtre de téléchargement de photo.

La deuxième, dernière fenêtre est pour la formation. Cette fenêtre contient quatre boutons, deux permettant de choisir un jeu de données d'apprentissage et les deux autres permettant de choisir un jeu de données permettant de tester les résultats de la phase d'apprentissage. Cette fenêtre contient également une barre de statistiques indiquant la statistique de la phase d'apprentissage. La dernière chose dans cette fenêtre est un petit panneau qui affiche la phase de test de l'ensemble de données choisi, voir la figure 4.3.

Enfin, pour utiliser le programme, nous devons commencer par former notre programme, et la formation sera effectuée sur un ensemble de données choisi. Une fois que nous aurons suivi la formation, nous devons effectuer la partie tests, dans la même fenêtre, qui est la deuxième fenêtre voir la figure 4.5. après avoir terminé tout cela, revenons à la première fenêtre ou à la fenêtre principale et cliquez sur le bouton "Charger", puis choisissez l'image d'origine sur laquelle nous allons travailler et nous devons reconnaître les chiffres qui y sont écrits. Une fois l'image chargée, il faut cliquer sur le bouton "segmentation". en atteignant ce point, nous n'avons aucune main pour changer quoi que ce soit ou une ingérence dans les prochaines étapes, ou dans les résultats obtenus. Car le programme affichera d'abord les résultats sous forme de chiffres segmentés dans des images isolées, puis notera le chiffre de chaque image. le programme est également formé pour imprimer même la virgule si elle existe dans l'image d'origine, voir la figure 4.4.

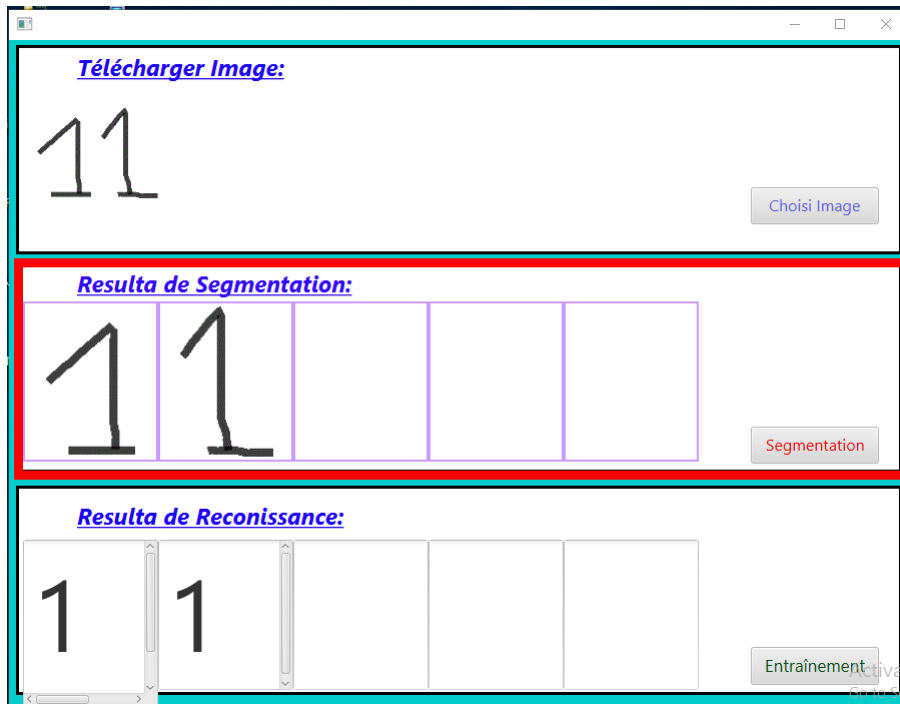


Figure 4.3: Fenêtre d'affichage résultat de segmentation .

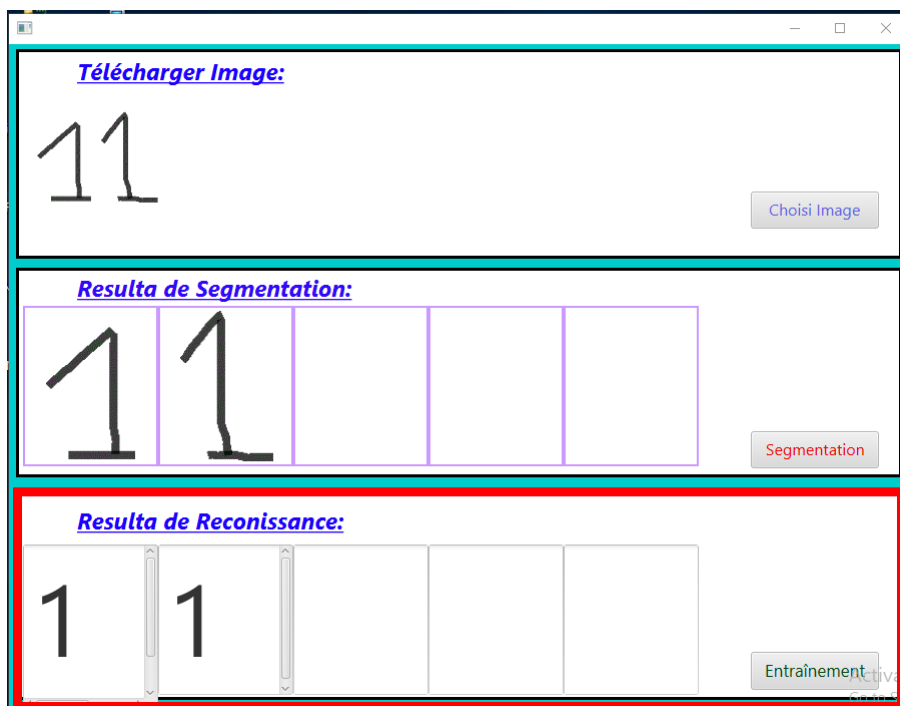


Figure 4.4: Fenêtre d'affichage résultat de reconnaissance .

4.5 Transformation image

la transformation de l'image d'origine en une image binaire peut être effectuée comme suit . d'abord nous vérifions la couleur rouge et la valeur alpha dans l'image, puis nous essayons de trouver si la valeur de la couleur rouge est absolument supérieure à 230, nous l'avons remplacée par la valeur 0, sinon si la valeur est égale ou inférieure à 230, nous

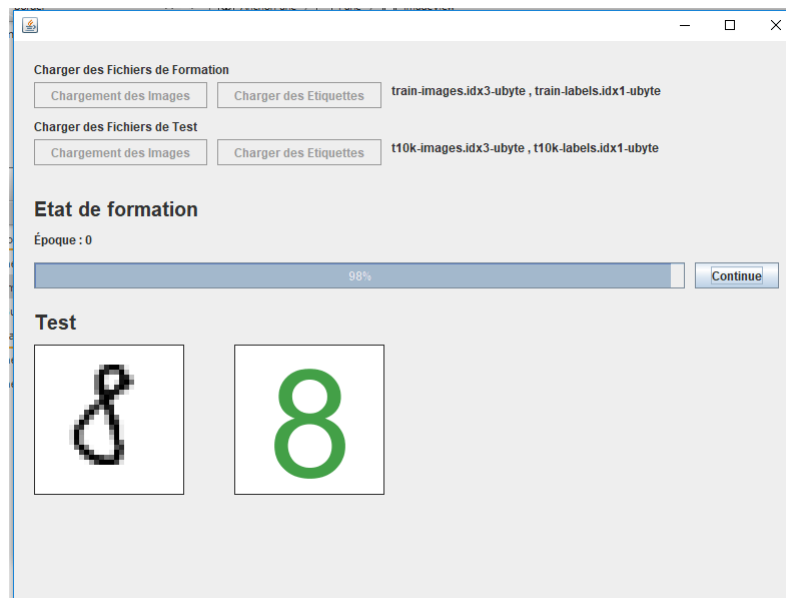


Figure 4.5: Interface d’affichage résultat de reconnaissance .

mettons la valeur 255 En remplaçant ces valeurs par 0 et 255, nous changeons l’image en noir et blanc où le blanc est 255 (1) et le noir est la valeur 0. l’étape suivante est la somme des valeurs alpha avec la valeur rouge multipliée par 3, puis nous mettre ces valeurs dans l’image binaire. ces opérations sont effectuées pour chaque pixel.

4.6 Résultat et Test

à ce point, nous pouvons prendre les résultats obtenus et les expériences acquises et les comparer. Dans cette section, nous allons présenter nos résultats de manière formelle afin qu’ils soient plus lisibles et plus clairs.

4.6.1 Segmentation

Dans ce tableau, nous présentons les résultats obtenus dans la phase de segmentation, nous avons testé notre algorithme de segmentation dans une image aléatoire différente, l’algorithme utilisé dans la segmentation est créé par nous. Le tableau ci-dessous indique le taux de segmentation et le temps d’exécution. les résultats sont divisés pour des groupes d’images, images connectées et superposées et normales, où les résultats sont parfaits et atteints à où les résultats sont parfaits et atteints à 100/100 dans l’image normale et environ 90/100 pour les images superposées et le pourcentage pour les images connectées est de 78/100. Ce que nous entendons par images normales, c’est que ces images contiennent des chiffres normaux. Les images superposées signifient que ces images contiennent des chiffres superposés, et les images connectées signifient que les images contiennent des chiffres connectés. Le rapport dans le tableau indique Le Taux de succès du méthode proposée et le test en détail.

Les catégories	Niveau de complexité	Nbr des images	Les images segmentés avec succès	Les images segmentés avec des défaillances	Le Taux de succès du méthode proposée
Ensemble d'images	Image Normale	100	100	00	100%
	Image avec Chevauchement	100	80	20	80%
	Image connecté	100	78	22	78%
Nbr globale des images testés		300	258	42	86%

Figure 4.6: Taux et vitesse de segmentation de chiffres.

4.6.2 Reconnaissance

Le tableau suivant présente les résultats obtenus pour la reconnaissance des chiffres. Notre classificateur a été formé sur un jeu de données MNIST, testé sur le même jeu de données et a obtenu des résultats d'environ 90/100. le jeu de données a été divisé en deux parties, la première pour l'entraînement et la seconde pour les tests. la division représentait 70/100 de l'ensemble de données pour l'entraînement et 30/100 pour les tests. Une fois que nous avons trouvé notre classificateur prêt, nous avons testé de manière aléatoire les images obtenues à partir de la phase de segmentation et les résultats sont présentés dans le tableau. Le tableau suivant illustre le test et le résultat de chaque chiffre.

Chiffre	Nombre exemples de test	Taux de reconnaissance
0	50	83 %
1	50	90 %
2	50	88 %
3	50	82 %
4	50	87 %
5	50	76 %
6	50	78 %
7	50	84 %
8	50	76 %
9	50	84 %

Figure 4.7: Taux de reconnaissance de chiffres.

Conclusion

Dans ce chapitre, nous avons présenté les résultats obtenus et les expériences de l'algorithme de segmentation créé et du classifieur de reconnaissance. nous avons travaillé sur différents types de chiffres comme les chiffres normaux, liés et superposés, chacun ayant donné des résultats différents. Enfin, nous avons prouvé l'efficacité de notre algorithme de segmentation et de notre classifieur de reconnaissance.

Conclusion et perspectives

dans ce travail, nous avons difficilement essayé d'obtenir un nouvel algorithme de segmentation et avons appliqué un classifieur de réseau neuronal aux résultats obtenus à partir de la segmentation, segmentation en chiffres superposés.

L'écriture est essentielle pour l'être humain, même avec le dynamisme technologique que nous vivons aujourd'hui, où les enseignants écrivent des notes sur leurs élèves, mais doivent néanmoins l'insérer dans les portes et plates-formes technologiques. ce travail peut être si dur et prend beaucoup de temps, dans ce travail nous avons essayé de le rendre plus facile.

Dans ce projet, nous avons créé un algorithme de segmentation et obtenu de bons résultats. Nous avons également créé un classificateur de reconnaissance basé sur les réseaux de neurones et obtenu des résultats satisfaisants et satisfaisants.

Pour appliquer ce projet, nous avons utilisé le langage de programmation java en raison de sa capacité à s'exécuter dans des plateformes différentes et du multi-threading. nous avons également utilisé netbeans ide pour mettre en œuvre notre projet et également pour pouvoir évaluer sa précision, et les résultats obtenus à la fois sur la segmentation et sur la reconnaissance ont été très convaincants et satisfaisants.

Même avec les bons résultats obtenus, nous gardons toujours à l'esprit le fait que nous n'avons malheureusement pas appliqué la détection en temps réel d'une vidéo ou d'une image, mais aussi la lecture du chiffre reconnu. Une autre chose que nous n'avons pas faite est de transformer tout un tableau de chiffres. dans un tableau numérique complet de chiffres dans Excel. ces choses que nous n'avons pas accomplies parce que le temps était bien sûr trop court et nous n'avons pas trouvé de jeu de données de chiffres superposés et reliés pour tester nos résultats directement sur eux. Nous avons donc été obligés de créer nos propres exemples et de les travailler.



Bibliographie

- [1] G. ABDELJALIL – “Segmentation-verification for handwritten digit recognition”, Thèse, MINISTRY OF HIGHER EDUCATION, 2016, p. 8–9.
- [2] H. AL-RASHAIDEH – “Preprocessing phase for Arabic Word Handwritten Recognition”, *INFORMATION TRANSMISSIONS IN COMPUTER NETWORKS* **6** (2006), p. 11–19.
- [3] 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.
- [4] C. CHATELAIN – “Extraction de séquences numériques dans des documents manuscrits quelconques”, Thèse, Université de Rouen, 2006.
- [5] M. CÔTÉ – “Utilisation d’un modèle d’accès lexical et de concepts perceptifs pour la reconnaissance d’images de mots cursifs. thèse de doctorat”, *École Nationale Supérieure des Télécommunications* (1997).
- [6] D. ABDELHAKIM – “La reconnaissance des chiffres manuscrits par les machines à vecteurs de support(svms)”, Thèse, Thèse de Master, Université de Tébessa, 2011.
- [7] P. DARGENTON – “contribution à la segmentation et à la reconnaissance de l’écriture manuscrite”, Thèse, Lyon, INSA, 1994, p. 174.
- [8] A. DUNDAR – “k-means classification of handwritten with convolutional networks”, Thèse, Université Paris-Est, 2016.
- [9] Z. FAOUZI, D. ABDELHAMID et B. MOHAMED CHAOUKI – “Methode structurelle a deux passes: verticale et horizontale pour la segmentation du texte arabe manuscrit”, *1st International Symposium of Informatics and its Applications ISIA*, Msila University, Algeria, 2014.
- [10] A. GATTAL – *Segmentation automatique pour la reconnaissance numérique des chèques bancaires algériens*, 2011.
- [11] N. GHANMI – “Segmentation d’images de documents manuscrits composites: application aux documents de chimie”, Thèse, Université de Lorraine, 2016.
- [12] F. GRANDIDIER – “Un nouvel algorithme de sélection de caractéristiques: application à la lecture automatique de l’écriture manuscrite”, Thèse, École de technologie supérieure, 2003.
- [13] I. GUYON, J. WESTON, S. BARNHILL et V. VAPNIK – “Gene selection for cancer classification using support vector machines”, *Machine learning* **46** (2002), no. 1-3, p. 389–422.
- [14] S. HAITAAMAR – “Segmentation de textes en caracteres pour la reconnaissance optique de l’écriture arabe”, Thèse, Université de Batna 2, 2007.
- [15] J. IBRAHIM – “Improved method of handwritten digit using the gradient histogram oriented tested on mnist data set”, *UpToDate, Waltham, MA. Accessed Dec* (2014).
- [16] 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.
- [17] — , “Improved method of handwritten digit recognition tested on mnist database”, *Image and Vision Computing* **22** (2004), no. 12, p. 971–981.
- [18] P. M. LALLICAN, C. VIARD-GAUDIN et S. KNERR – “From off-line to on-line handwriting recognition”, (2000).

- [19] C.-Y. LEE, P. W. GALLAGHER et Z. TU – “Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree”, *Artificial Intelligence and Statistics*, 2016, p. 464–472.
- [20] 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.
- [21] G. MAYRAZ et G. E. HINTON – “Recognizing handwritten digits using hierarchical products of experts”, *TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* **24** (2002).
- [22] F. MENASRI – “Contributions à la reconnaissance de l’écriture arabe manuscrite”, *UNIVERSITÉ PARIS DESCARTES, Thèse de doctorat* (2008).
- [23] Z. S. MENASRIA ABDELAALI – “Reconnaissance hors ligne des chiffres manuscrite isolé (base de donnée m.n.i.s.t)”, Thèse, Université de Tébessa, 2016, p. 10.
- [24] S. NEBTI – “Reconnaissance de caractères manuscrits par intelligence collective”, Thèse, Université de Sétif 1-Ferhat Abbas, 2013.
- [25] H. SCHAHRAZED – *Segmentation de textes en caracteres pour la reconnaissance optique de l’écriture arabe*, Mémoire, Université El-Hadj Lakhdar Batna, 2007.
- [26] C. TOUZET – *les réseaux de neurones artificiels, introduction au connexionnisme*, EC2, 1992.
- [27] M. WU et Z. ZHANG – “Handwritten digit classification using the mnist data set”, *Course project CSE802: Pattern Classification & Analysis* (2010).
- [28] Y. YAMASHITA et T. WAKAHARA – “k-nn classification of handwritten characters using a new distortion-tolerant matching measure”, *2014 22nd International Conference on Pattern Recognition*, IEEE, 2014, p. 262–267.
- [29] — , “Affine-transformation and 2d-projection invariant k-nn classification of handwritten characters via a new matching measure”, *Pattern Recognition* **52** (2016), p. 459–470.
- [30] C. YUCEF – “Traitement automatique des chèques postaux algériens”, Thèse, Faculté d’électronique et d’informatique, RSDT, 2011–2013, p. 15.
- [31] A. ZAHOUR, A. DJEMATENE, S. KEBAIRI, A. BENNASRI et B. TACONET – “Contribution à la reconnaissance de l’écriture arabe manuscrite”, *Proceedings of CIFED*, vol. 98, 1998, p. 219–227.
- [32] F. ZAIZ – “Technique basée puzzle/svm pour l’amélioration de la reconnaissance du texte arabe manuscrit”, Thèse, Université Mohamed Khider-Biskra, 2017, p. 6–10.
- [33] M. ZHAO, C. FU, L. JI, K. TANG et M. ZHOU – “Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes”, *Expert Systems with Applications* **38** (2011), no. 5, p. 5197–5204.