



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة محمد البشير الإبراهيمي برج بوعريريج
Université Mohamed El Bachir El Ibrahimi B.B.A.

كلية الرياضيات و الاعلام الالي
Faculté de mathématiques et d'informatique
قسم الاعلام الالي

Département d'informatique

Mémoire

En vue de l'obtention du Diplôme de Master

Domaine Des mathématiques et d'informatique

Filière : informatique

Spécialité : Réseaux et Multimédia

Thème

Conception et réalisation d'un OCR

Présenté par : Mecheri zakaria
Hacine_gherbi Abdellah

Devant le jury :

Président : Dr. Attia Abdelouahab , MCA à l'université de Bordj Bou Arréridj.

Examineur: Mr. Saifi Abdelhamid, MAA à l'université de Bordj Bou Arréridj.

Encadrant: Dr. Bouziane Abderraouf, MCA à l'université de Bordj Bou Arréridj)

Année universitaire : 2019/2020

DEDICACE

Je vous dédie ce travail avec tous

Mes vœux de bonheur, de santé et de réussite

À mes parents

À mes frères Abdenour, Ahcine, Bilal et Amine

À mes sœurs

A toute ma famille

À tous mes amis et collègues

À tous mes enseignants

À tous les étudiants de la promotion 2019/2020

A tous ceux qui, par un mot, m'ont donné la force de continuer

ABDELLAH

DEDICACE

Je vous dédie ce travail avec tous

Mes vœux de bonheur, de santé et de réussite

À mes parents

À mes deux frères Brahim et Youcef

À ma sœur et son fils abdel_hay

A ma grand-mère, et toute ma famille

À tous mes amis et collègues

À tous mes enseignants

À tous les étudiants de la promotion 2019/2020

A tous ceux qui, par un mot, m'ont donné la force de continuer

ZAKI

ملخص :

هذه الرسالة مكرسة لتحقيق نظام التعرف الضوئي على الحروف يعتمد بشكل أساسي على طريقة الرسم البياني للتدرج

الموجه (HOG) ، المصنف (K-NN (KPPV). مساهماتنا هي:

المعالجة المسبقة للصور (الترميز الثنائي ، التصفية ، المحاذاة) ، تقنيات التجزئة (تجزئة الخط ، تجزئة الكلمات) ،

التوصيف (LBP + HOG) ، التعرف.

الكلمات المفتاحية: التعرف الضوئي على الحروف ، HOG ، KPPV.

Résumé

Ce mémoire est consacrée à la réalisation d'un système de reconnaissance optique de caractère en s'appuyant principalement sur la méthode d'histogramme de gradient Orienté (HOG), le classifieur K plus proches voisins K-NN (KPPV). Nos contributions sont:

Le prétraitement de l'image (binarisation, filtrage, alignement), les techniques de Segmentation (Segmentation des lignes, Segmentation des mots), Caractérisation (LBP + HOG), la Reconnaissance.

Mots clés : reconnaissance optique de caractère, HOG, KPPV.

Abstract:

This dissertation is devoted to the realization of an optical character recognition system based mainly on the Oriented Gradient Histogram (HOG) method, the K-classifier K-NN (KPPV). Our contributions are:

Image preprocessing (binarization, filtering, alignment), Segmentation techniques (Line segmentation, Character segmentation), Characterization (LBP + HOG), Recognition.

Key words: optical character recognition, HOG, K-NN.

Table des matières

1	Introduction générale et problématique	1
1.1	Contexte	2
1.2	Problématique et objectif	3
1.3	Contribution	3
1.4	Plan de mémoire	4
2	Etat de l’art	5
2.1	les modèles existants	6
2.1.1	Mayraz et Hinton (2002)	6
2.1.2	Kussul et Baidyk (2004)	6
2.1.3	Wu et Zhang (2010)	7
2.1.4	Ebrahim et Jampour (2014)	7
2.1.5	Wakahara et Yamashita (2014)	7
2.1.6	Lee et al (2015)	8
2.1.7	Wakahara et Yamashita (2016)	8
2.1.8	Dundar et al (2016)	9
2.1.9	Quelque résultat expérimental obtenu par les auteurs . . .	9
2.2	Description sommaire du modèle proposé	10
2.3	Objectifs du modèle proposé	10
2.4	Conclusion	11

3	Architecture et Modélisation	13
3.1	Introduction	14
3.2	Description du modèle	14
3.2.1	Les étapes de système proposé	14
3.2.1.1	Phase d'acquisition	15
3.2.1.2	Phase de pré-traitement	15
3.2.1.3	Phase de segmentation	21
3.2.1.4	Phase de post-traitement : " la reconnaissance " . .	22
3.3	La conception	28
3.3.1	Présentation d'UML	28
3.3.2	Diagramme des cas d'utilisation	28
3.3.3	Diagramme des classes	29
3.4	Conclusion	30
4	Implémentation et Bilan	31
4.1	Introduction	32
4.2	Les outils et langage utilisés	32
4.2.1	Langage de calcule technique MATLAB	32
4.2.2	Guide MATLAB (Graphical User Interface Development Environment)	32
4.2.3	Environnement de développement MATLAB (Matlab R2016a)	34
4.3	Présentation les interfaces de l'application	35
4.3.1	L'acquisition	36
4.3.2	Pré-traitement	36
4.3.3	La segmentation	40
4.3.4	La reconnaissance	41
4.4	Evaluation	42
4.4.1	Bases de données	42

4.4.2	Le taux moyen de bonne reconnaissance	43
4.5	Conclusion	43
5	Conclusion générale	45
5.1	Conclusion générale	46
5.1.1	Synthèse des chapitres	46
5.1.2	Perspectives	47
	Bibliographie	49

Table des figures

3.1	Schéma qui résume L'exécution de notre système.	14
3.2	Exemple de binarisation.	15
3.3	Effet de seuillage sur la qualité des caractères.[16]	16
3.4	Résultat de la binarisation de différentes images par le même seuil S=120.[16]	17
3.5	Le masque d'un filtrage moyen	18
3.6	Un exemple d'un filtrage moyen	18
3.7	Un exemple pour calculer la matrice de convolution gaussienne avec $\sigma = 0.8$	19
3.8	Étapes du filtre médian	20
3.9	Exemple de deux types de redressement.	21
3.10	Etapes du Descripteur HOG.	23
3.11	Diviser l'image en blocs	24
3.12	Calcul de l'histogramme de bloc	24
3.13	Méthode de K plus proches voisins.	26
3.14	Diagramme des cas d'utilisation	29
3.15	Diagramme des classes	30
4.1	Logo de MATLAB.	32
4.2	Fenêtre principale du GUIDE.	33
4.3	Inspecteur de propriété.	34

4.4	Interface MATLAB R2016a.	35
4.5	Interface de notre application.	35
4.6	Pour charger l'image.	36
4.7	Pour la binaritaion «automatique».	37
4.8	Pour la binaritaion «avec niveau »	37
4.9	Pour le filtrage «avec filtre moyennneur »	38
4.10	Pour le filtrage «avec filtre gaussien»	38
4.11	Pour le filtrage «avec filtre median »	39
4.12	L'image « avant» l'alignement.	39
4.13	L'image « après» l'alignement.	40
4.14	La segmentation en lignes.	40
4.15	La segmentation en caractères.	41
4.16	Le résultat obtenu.	41
4.17	Exemples de la base de données MNIST.	42
4.18	Le taux moyen de bonne reconnaissance.	43

Chapitre

1

Introduction générale et problématique

Sommaire

1.1	Contexte	2
1.2	Problématique et objectif	3
1.3	Contribution	3
1.4	Plan de mémoire	4

1.1 Contexte

Le développement des systèmes de reconnaissance de l'écriture manuscrite a commencé dans les années 1950, lorsque des opérateurs humains avaient pour tâche de convertir les données de divers documents en format électronique, ce qui rendait le processus assez long et souvent affecté par des erreurs. La reconnaissance automatique de texte vise à limiter ces erreurs en utilisant des techniques de prétraitement d'image qui apportent une vitesse et une précision accrues à l'ensemble du processus de reconnaissance. La reconnaissance de l'écriture manuscrite a été l'un des domaines de recherche les plus fascinants et les plus stimulants dans le domaine du traitement d'image et de la reconnaissance de formes ces dernières années. Il contribue énormément à l'avancement des processus d'automatisation et améliore l'interface entre l'homme et la machine dans de nombreuses applications. La reconnaissance optique des caractères est un domaine d'étude qui peut englober de nombreuses techniques de résolution différentes. Les réseaux de neurones (Sandhu et Leon, 2009), les machines vectorielles de support et les classificateurs statistiques semblent être les solutions préférées au problème en raison de leur précision éprouvée dans la classification de nouvelles données [1].

La reconnaissance optique de caractères est en fait un convertisseur qui traduit des images de texte manuscrites en texte basé sur une machine. En général, la reconnaissance de l'écriture manuscrite est classée en deux types : hors ligne et en ligne. Dans la reconnaissance hors ligne, l'écriture est généralement capturée optiquement par un scanner et l'écriture terminée est disponible sous forme d'image. En d'autres termes, le texte manuscrit hors ligne est lorsque le texte écrit à la main est numérisé par un scanner dans un format numérique. Mais, dans le système en ligne, les coordonnées bidimensionnelles des points successifs sont représentées en fonction du temps et de l'ordre des traits effectués par l'écrivain. En d'autres termes, les coordonnées X-Y sont données comme un résultat qui indique l'emplacement du stylo et la force appliquée par l'utilisateur pendant l'écriture et la vitesse également. Le texte manuscrit en ligne est écrit par un stylet sur une tablette. Il existe également un troisième procédé qui n'est pas aussi connu que les deux premiers procédés mentionnés ci-dessus dans lesquels des dispositifs laser, à jet d'encre, peuvent être utilisés pour obtenir un texte imprimé par machine [2].

Il y a un travail considérable dans le domaine de la reconnaissance de l'écriture manuscrite et un certain nombre de revues existent. Les méthodes en ligne se sont avérées supérieures à leurs homologues hors ligne dans la reconnais-

sance des caractères manuscrits en raison des informations temporelles disponibles avec les premières [3][4]. Cependant, plusieurs applications telles que le tri du courrier, le traitement bancaire, la lecture de documents et la reconnaissance d'adresses postales nécessitent des systèmes de reconnaissance d'écriture manuscrite hors ligne. De plus, dans les systèmes hors ligne, les réseaux neuronaux et les machines à vecteurs de support ont été utilisés avec succès pour produire des niveaux de précision de reconnaissance comparativement élevés. En conséquence, la reconnaissance de l'écriture manuscrite hors ligne continue d'être un domaine de recherche actif en vue d'explorer les nouvelles techniques susceptibles d'améliorer la précision de la reconnaissance [5] [6].

1.2 Problématique et objectif

La reconnaissance optique de caractères (OCR en anglais) est un processus qui permet de convertir un texte présenté par une image numérique en un texte modifiable.

Par conséquent, les documents papier sont convertis en fichiers informatiques modifiables. Et c'est précisément l'intérêt de la reconnaissance de texte : saisir des textes dans votre ordinateur sans les retaper. Après tout, aussi efficaces que soient les ordinateurs, vous devez d'abord saisir vos informations.

Les applications doivent être claires maintenant : capturer des documents sans les retaper, donner à vos documents papier une nouvelle vie. Si vous avez déjà retapé un rapport de 15 pages ou un grand tableau de chiffres, vous savez à quel point cela peut être fastidieux et long ! Le besoin de recapture des données est en fait beaucoup plus important que vous ne le pensez au début.

En conclusion, l'OCR est peut-être l'outil le plus efficace, il n'y a donc aucune raison pour que vous ne vous déchargiez pas de la tâche fastidieuse de retaper du texte et des données.

1.3 Contribution

Le but de notre projet est la réalisation d'un système de reconnaissance optique des caractères analyse optiquement un texte et en produit une version informatique, sous forme d'un fichier texte, comme s'il avait été saisi sur un ordinateur. Parfois les documents à traiter peuvent être dégradé physiquement ou lors de leurs acquisition pour cela l'étape de prétraitement est donc indispensable afin de rendre fiable l'étape de conversion de l'image vers un texte,

Les filtres linéaires pour le traitement du bruit, tel que le filtre gaussien, moyen ...etc. permettent de lisser l'image et ainsi diminuer le bruit qui pourrait impacter négativement sur le résultat de reconnaissance sans garantir la conversion des contours. Pour améliorer les résultats de l'OCR, nous allons utiliser quelques prétraitements pour améliorer la performance de l'OCR.

1.4 Plan de mémoire

Ce mémoire présente une technologie qui permet de convertir différents types de documents tels que les documents papiers scannés ou les photos numériques vers des formats modifiables et exploitables. Sur le plan méthodologique. Il est organisé comme suit :

- Le chapitre 1 présente une étude bibliographie sur le système de reconnaissance automatique OCR.
- Le chapitre 2 présente un état de l'art sur différentes méthodes de reconnaissance automatique en partant de la description d'une chacune méthode.
- Le chapitre 3 présente les différentes étapes suivies par la conception de notre solution.
- Le chapitre 4, a été alloué pour présenter les interfaces et les résultats obtenus.

Chapitre 2

Etat de l'art

Sommaire

2.1	les modèles existants	6
2.1.1	Mayraz et Hinton (2002)	6
2.1.2	Kussul et Baidyk (2004)	6
2.1.3	Wu et Zhang (2010)	7
2.1.4	Ebrahim et Jampour (2014)	7
2.1.5	Wakahara et Yamashita (2014)	7
2.1.6	Lee et al (2015)	8
2.1.7	Wakahara et Yamashita (2016)	8
2.1.8	Dundar et al (2016)	9
2.1.9	Quelque résultat expérimental obtenu par les auteurs . . .	9
2.2	Description sommaire du modèle proposé	10
2.3	Objectifs du modèle proposé	10
2.4	Conclusion	11

2.1 les modèles existants

Pour concevoir un système de reconnaissance d'écriture manuscrite, il faut avoir les différents travaux ultérieurs qu'ils ont réalisés sur la base de données M.N.I.S.T. Dans ce chapitre, nous décrivons de ces travaux. En précisant la performance et le résultat obtenu.

2.1.1 Mayraz et Hinton (2002)

G. Mayraz et G. E. Hinton. 2002 utilisent le produit d'une procédure d'apprentissage d'experts qui permet de découvrir un ensemble de caractéristiques binaires stochastiques qui constituent un modèle génératif non linéaire d'images d'écriture manuscrite. La qualité des modèles génératifs formés peut être évaluée par l'apprentissage d'un modèle distinct pour chaque catégorie d'écriture, puis en comparant les probabilités non normalisées d'images de test sous les 10 différents modèles de classe spécifique. Pour améliorer les performances discriminatoires, une hiérarchie de modèles distincts peut être formée pour chaque classe d'écriture. Chaque modèle de la hiérarchie apprend une couche de détecteurs de caractéristiques binaires qui modélisent la distribution de probabilité des vecteurs d'activité des détecteurs de caractéristique dans la couche inférieure. Les modèles de la hiérarchie sont formés séquentiellement et chaque modèle utilise une couche de détecteurs de caractéristiques binaires pour former un modèle génératif des modèles d'activités caractéristiques dans la couche précédente. Après la formation, chaque couche de détecteurs de caractéristiques est produite séparée, non normalisée de score de probabilités. Avec trois couches de détecteurs de caractéristiques pour chacune des classes de 10 d'écriture, une image de test produisent scores 30 qui peuvent être utilisés comme entrants dans un réseau de 24 classification supervisée et logistique qui est formé sur des données distinctes. Sur la base de données MNIST, ce système démontrant que le produit d'une procédure d'apprentissage d'experts peut produire des hiérarchies efficaces de modèles génératifs de données de grande dimension. Le taux de reconnaissance atteint 98.3% sur la base de données MNIST qui constitue 60 000 images d'apprentissage et 10 000 images de test [7].

2.1.2 Kussul et Baidyk (2004)

Kussul et Baidyk ont développé un nouveau classifieur neuronal Limited Receptive Area (LIRA) pour la reconnaissance d'image. Le LIRA classifieur contient

trois couches de neurones : entrée, associative et sortie. La couche d'entrée est reliée à la couche associative sans connexion aléatoire modifiable et la couche associative est reliée à la couche de sortie avec des connexions de l'apprentissage. Le processus d'apprentissage converge suffisamment rapide. Ce classificateur n'utilise pas d'opérations en virgule flottante et de multiplication. Le classifieur a été testé sur deux bases de données. La première base de données est la base de données MNIST. Elle contient 60.000 images d'écritures manuscrites pour l'apprentissage et 10.000 images d'écritures manuscrites pour le test. La seconde base de données contient 441 images de micro dispositif d'assemblage. Le classifieur LIRA donne un taux de reconnaissance de 99.59% [8].

2.1.3 Wu et Zhang (2010)

Wu et Zhang ont utilisé les caractéristiques de direction extraites pour la réduction de la dimensionnalité. K-ième voisin le plus proche, les modèles de mélanges gaussiens et SVM se révélant les meilleurs candidats pour les caractéristiques. Le taux de reconnaissance atteint 98.81% sur la base de donnée MNIST qui constitue 60 000 images d'apprentissage et 10 000 images de test. En utilisant 3-NN [9].

2.1.4 Ebrahim et Jampour (2014)

Ebrahim et Jampour ont proposé une approche à base de fonction de l'appariement qui traite des données en utilisant l'Histogramme de gradient Orienté (HOG). HOG est un descripteur de caractéristique très efficace pour les chiffres 25 manuscrits, qui est stable sur des variations d'éclairage, car il est à base d'un descripteur de gradient. En outre, SVM linéaire a été employé comme classificateur qui a de meilleurs résultats que les noyaux RBF et sigmoïde. Le taux de reconnaissance atteint 97,25% sur la base de données MNIST qui constitue 60 000 images d'apprentissage et 10 000 images de test.

2.1.5 Wakahara et Yamashita (2014)

Global affine transformation (GAT) est proposée par Wakahara et al. C'est une méthode de modèle qui base sur l'appariement (matching) qui peut compenser la transformation affine incorporé dans un modèle d'entrée. La méthode de corrélation GAT a démontré une haute performance dans la reconnaissance de

caractères et le matching des objets. Une nouvelle mesure d'adaptation appelée Nearest Neighbors Distance of Equi-Gradient Direction (NNDEGD) utilisé en coopération avec la méthode de corrélation GAT. Le NNDEGD est juste le paramètre de la fenêtre de la fonction gaussienne utilisée dans la méthode de corrélation GAT, qui est égale à la distance minimale moyenne entre un point dans une image et un autre point dans l'autre image avec la même direction de gradient. Ensuite ils utilisent cette valeur comme une nouvelle mesure d'appariement. Deuxièmement, ils ont étendu la méthode de corrélation GAT pour gérer le changement de largeur de trait en plus de la transformation affine. Enfin, ils appliquent la version originale et étendues de la méthode de corrélation GAT à la classification de FC-NN expériences en utilisant la base de données MNIST. Ces expériences sont réalisées de manière efficace pour la première fois parce que ils ont réduit considérablement la complexité et la mémoire de charge de calcul impliqués par rapport à la méthode de corrélation de GAT originale. Le taux de reconnaissance atteint 99.51% sur la base de donnée MNIST qui constituer 60 000 images d'apprentissage et 10 000 images de test[10].

2.1.6 Lee et al (2015)

Dans L'article [11] Lee et al ont amélioré les réseaux de neurones profonds en généralisant les opérations de pooling qui jouent un rôle central dans les architectures actuelles. Ils ont poursuivi une exploration minutieuse des approches pour permettre la pooling d'apprendre et de s'adapter à des modèles complexes et variables. Les deux directions principales se trouvent dans (1) l'apprentissage d'une fonction de pooling via 26 (deux stratégies) combinant des max et la moyenne de pooling, et (2) l'apprentissage d'une fonction de pooling sous la forme d'une fusion à structure arborescente des filtres de pooling eux-mêmes appris. Dans leurs expériences chaque opération de pooling généralisée, ils explorent à améliorer les performances lorsqu'il est utilisé à la place de max pooling. Le taux de reconnaissance atteint 99.71% sur la base de donnée MNIST qui constituer 60 000 images d'apprentissage et 10 000 images de test.

2.1.7 Wakahara et Yamashita (2016)

Le modele de la reconnaissance basée sur l'appariement (matching) ne nécessite pas un processus d'apprentissage et le résultat de matching fournit des information intuitive et géométrique. Wakahara et al. Ont proposé Global Affine Transformation (GAT) pour la corrélation d'appariement, ce qui peut com-

penser les transformations affines imposées à un modèle. La corrélation d'appariement de GAT avec une méthode d'accélération et une nouvelle mesure d'apparement, appelée Nearest-Neighbor Distance of Equi-Gradient Direction (NNDEGD), atteint de hautes performances dans des expériences en utilisant la base de données MNIST. La mesure d'apparement GAT a été étendue à global projection transformation de mesure d'apparement (GPT) de permettre une déformation par transformation de projection 2D. Leur travail consiste d'abord à développer une méthode d'accélération pour GPT d'apparement de corrélation. Deuxièmement, à améliorer les performances de reconnaissance, ils ont appliqué la courbure (curvature) des contours en traits à la mesure d'apparement. la Courbure (curvature) est souvent utilisée comme une caractéristique des caractères. Toutefois, ils avaient utilisé comme un poids dans le NNDEGD. Troisièmement, pour une enquête sur les performances des méthodes proposées, ils ont appliqué l'apparement de l'image et la reconnaissance des bases de données MNIST et ITPP pour kPPV (k -NN). Dans l'expérience avec la base donnée MNIST, l'apparement de corrélation GPT avec la mesure d'apparement de la courbure pondéré NNDEGD a achevé le taux de reconnaissance le plus élevée de 99.70% avec les méthodes k -NN [12].

2.1.8 Dundar et al (2016)

Dundar et al Ont proposé de former un réseau de convolution profond basé sur une version améliorée des algorithmes k-means, ce qui réduit le nombre de paramètres 27 corrélés dans la forme de filtres similaires, et ainsi augmente la précision de la catégorisation de test. Les expériences montrent que l'algorithme proposé surpasse les autres techniques qui apprennent les filtres non supervisés. Plus précisément, Le taux de reconnaissance atteint 98.6% sur la base de donnée MNIST qui constituer 60 000 images d'apprentissage et 10 000 images de test [13].

2.1.9 Quelque résultat expérimental obtenu par les auteurs

Dans les sections précédentes on a présenté différentes méthodes et algorithmes a compris le principe de mise en œuvre. Le tableau au-dessus, illustre les performances et l'approche utilisée pour chaque méthode.

Année	EDITEUR	Taux (%)	METHODE
2002	Mayraz et Hinton	98.3	Produit expert
2004	Kussul et Baidyk	99.59	LIRA
2010	Wu et Zhang	98.81	Caractéristique de direction +SVM
2014	Ebrahimzadeh et Jampour	97.25	HOG + SVM
2014	Wakahara et Yamashita	99.51	GAT NNDEGD
2015	Lee et al	99.71	Réseaux de neurones
2016	Wakahara et Yamashita	92.5	NNDEGD+GPT
2016	Dundar et al	98.6	réseaux convolutionnel profond + Kmeans

TABLE 2.1 – Quelque résultats rapportés dans la littérature.

2.2 Description sommaire du modèle proposé

Histogramme de gradient Orienté (Histogram of Oriented Gradients HOG) a été proposé par Dalal et Triggs pour la détection de corps humain, mais il est maintenant l'un des descripteurs utilisés avec succès et populaires dans la vision d'ordinateur et reconnaissance de formes. Le descripteur HOG compte les occurrences d'orientation gradient en partie d'une image où il est un descripteur d'apparence. Il divise l'image d'entrée en petites cellules carrées (par exemple 9×9), puis calcule l'histogramme des directions de gradient ou les directions de bord sur la base des différences centrales. Pour améliorer la précision, les histogrammes locaux ont été normalisés sur la base du contraste et ceci est la raison pour laquelle HOG est stable sur la variation d'éclairage. En le comparant avec SIFT et LBP, HOG est plus rapide et plus stable sur la variation d'illumination en raison de calculs simples., il a été également montré que les caractéristiques du HOG sont des descripteurs avec succès pour la détection [14].

2.3 Objectifs du modèle proposé

L'idée importante est que l'apparence et la forme locale d'un objet dans une image peuvent être décrites par la distribution de l'intensité du gradient ou la direction des contours. Ceci peut être fait en divisant l'image en des régions adjacentes de petite taille, appelées cellules, et en calculant pour chaque cellule l'histogramme des directions du gradient ou des orientations des contours pour les pixels à l'intérieur de cette cellule. La combinaison des histogrammes forme alors le descripteur HOG. Pour de meilleurs résultats, les histogrammes locaux

sont normalisés en contraste, en calculant une mesure de l'intensité sur des zones plus larges que les cellules, appelées des blocs, et en utilisant cette valeur pour 30 normaliser toutes les cellules du bloc. Cette normalisation permet une meilleure résistance aux changements d'illuminations et aux ombres.

2.4 Conclusion

Dans ce chapitre nous avons présenté un aperçu sur état de l'art des méthodes et techniques des segmentations, d'écriture manuscrite en fonction de l'approche utilisée et le résultat obtenu. Ensuite nous avons présenté une description sur le modèle proposé et leurs objectifs.

Architecture et Modélisation

Sommaire

3.1	Introduction	14
3.2	Description du modèle	14
3.2.1	Les étapes de système proposé	14
3.2.1.1	Phase d'acquisition	15
3.2.1.2	Phase de pré-traitement	15
3.2.1.3	Phase de segmentation	21
3.2.1.4	Phase de post-traitement : " la reconnaissance "	22
3.3	La conception	28
3.3.1	Présentation d'UML	28
3.3.2	Diagramme des cas d'utilisation	28
3.3.3	Diagramme des classes	29
3.4	Conclusion	30

3.1 Introduction

Le but de notre projet est la réalisation d'une application utilisant la méthode d'histogramme de gradient Orienté (Histogram of Oriented Gradients HOG) pour la reconnaissance de caractère, nous allons intégrer des prétraitements pour améliorer les résultats. Dans ce qui suit nous présentons les différentes étapes suivies par la conception de notre solution.

3.2 Description du modèle

Lors de l'utilisation de notre system, au lieu d'avoir recoure directement à l'histogramme de gradient Orienté l'utilisation pourra utiliser des prétraitements pour améliorer la qualité d'image avant de passer à la segmentation et la reconnaissance de caractère. L'exécution de notre système suivra donc l'enchaînement suivant :

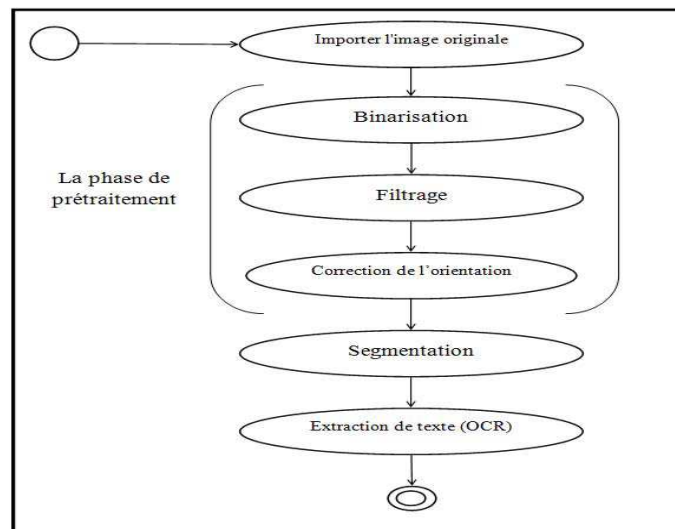


FIGURE 3.1 – Schéma qui résume L'exécution de notre système.

3.2.1 Les étapes de système proposé

Un système de reconnaissance automatique (OCR) s'effectue généralement en quatre étapes :

3.2.1.1 Phase d'acquisition

La phase d'acquisition base essentiellement sur la capture de l'image de l'écriture manuscrite au moyen des capteurs physiques et de la convertir en élévations numériques adaptés au système de traitement informatisés avec un minimum de dégradation possible. Dans le cas où l'information est disponible sur un support souvent papier, les capteurs physiques sont des scanners ou des caméras numériques. Durant cette phase, malgré la bonne qualité des systèmes d'acquisition, des bruits parasites peuvent apparaître et causer une hétérogénéité du fond [15]. Ceci est dû à la nature de la texture, l'aire de travail et de son éclairage.

3.2.1.2 Phase de pré-traitement

C'est une phase nécessaire, importante et très efficace dans un système de reconnaissance automatique (OCR), pour arriver à une rapidité de traitement et atteindre des meilleurs résultats. C'est l'étape du prétraitement (preprocessing step), qui comprend plusieurs et diverses méthodes et techniques dépendant dans leur utilisation de la nature et le contenu de l'image. Cette étape nous intéresse dans notre mémoire parce qu'elle permet de faciliter tous les traitements ultérieurs tels que la segmentation, l'extraction de primitives et la classification. Parmi les opérations de prétraitements généralement utilisées, citons :

1. La binarisation : Les systèmes de reconnaissance nécessitent une étape de binarisation qui vise à séparer les pixels de texte des pixels de l'arrière-plan de l'image traitée. En fait, la plupart des systèmes ne fonctionnent que sur des images binaires.



FIGURE 3.2 – Exemple de binarisation.

La plus simple façon pour obtenir une image binaire est de choisir une valeur seuil, puis de classer tous les pixels dont les valeurs sont au-dessus de ce seuil comme étant des pixels d'arrière-plan, et tous les autres pixels

comme étant des pixels de texte. Soit l'image $I(M \times N)$, supposons que $f(x,y)$ représente le niveau de gris du pixel aux coordonnées (x,y) , $0 \leq x \leq M, 0 \leq y \leq N$ et s est le seuil choisi, les pixels de l'objet sont ceux ayant le niveau de gris inférieur à s et les autres ayant le niveau de gris supérieur à s sont des pixels du fond. Alors, l'image binarisée g est déterminée par les pixels (x,y) dont la valeur est donnée par l'équation :

$$g(x,y) = 1 \text{ si } f(x,y) < s \quad (3.1)$$

$$g(x,y) = 0 \text{ si } f(x,y) \geq s \quad (3.2)$$

Dans la pratique, cette situation idéale ne se rencontre que très rarement. Les niveaux de gris associés au fond et aux objets présents sur l'image sont supposés être suffisamment différents pour qu'une bonne discrimination puisse être faite. Cependant, cette dichotomie n'est évidemment pas parfaite en raison de défauts d'éclairage ou de bruits introduits par l'opération d'acquisition lui-même. Par conséquent, un mauvais choix d'un seuil de binarisation peut détruire une grande part d'information utile contenue dans l'image en dégradant notamment la qualité des caractères à reconnaître par l'OCR, ces caractères peuvent ainsi être fragmentés ou fusionnés.[16]

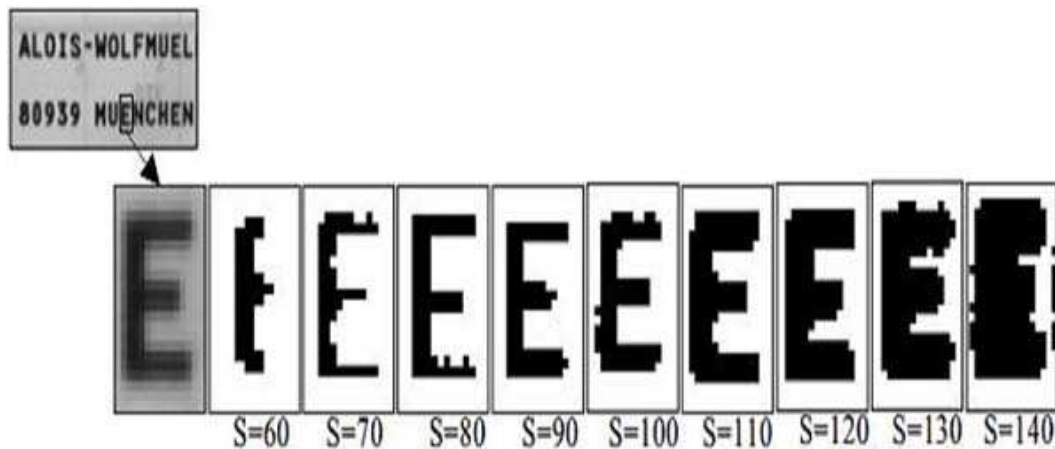


FIGURE 3.3 – Effet de seuillage sur la qualité des caractères.[16]



FIGURE 3.4 – Résultat de la binarisation de différentes images par le même seuil $S=120$. [16]

2. Le filtrage : Nous allons intéressé à du filtrage spatial : c'est-à-dire un filtrage qui s'applique sur un voisinage d'un pixel dans une image. Parmi les différents types de filtrage, certains sont linéaires, s'exprimant sous forme de convolution, d'autres sont non-linéaires (filtrage conservatif, filtrage médian, etc.). Les filtres peuvent effectuer plusieurs types d'opérations comme du lissage ou du rehaussement de contours.

Le principe du filtrage est de modifier la valeur des pixels afin d'améliorer la qualité visuelle de l'image et d'obtenir une image proche à la réalité qui aurait pu être obtenue si le système de capture était parfait.

Le principe général des filtres est pour chaque pixel, le filtre utilise les valeurs des pixels voisins pour calculer la valeur finale du pixel.

- (a) Filtrage linéaire : Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon l'opération « convolution ».

- Filtre moyennneur : Le filtre moyennneur est un filtre passe-bas permettant ainsi d'éliminer les hautes fréquences, correspondant au bruit. Son inconvénient est qu'il élimine également les hautes fréquences correspondant aux détails de l'image : il rend ainsi l'image moins bruitée mais plus floue.

Principe : Le niveau de gris du pixel central est remplacé par la moyenne des niveaux de gris des pixels environnants. Voici le masque 3×3 :



FIGURE 3.5 – Le masque d'un filtrage moyen

La figure suivante montre un exemple de calcul de la moyennes :

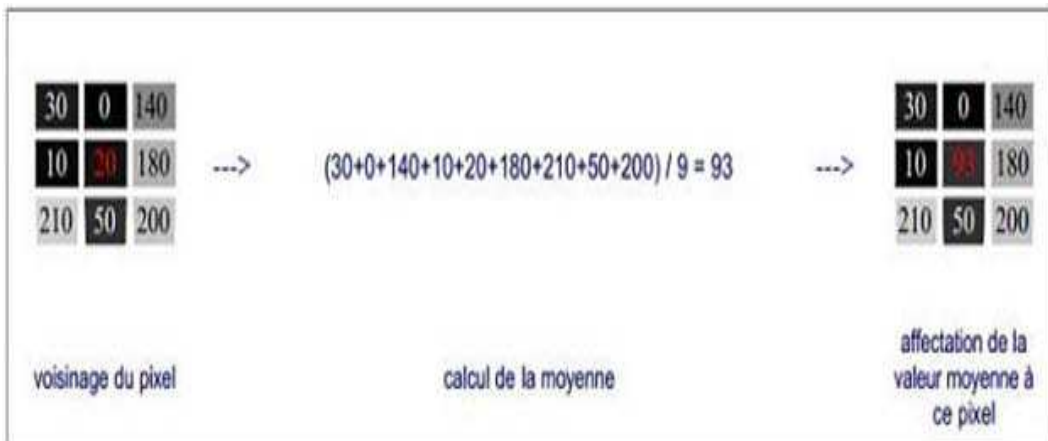
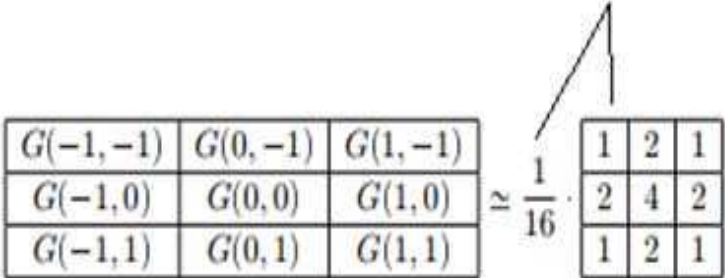


FIGURE 3.6 – Un exemple d'un filtrage moyen

- Filtre gaussien : C'est également un filtre passe-bas. Une gaussienne à deux dimensions est donnée par l'expression suivante :

$$g(x;y) = \frac{1}{2\pi\sigma^2} \exp(-(x^2 + y^2)/(2\sigma^2)) \quad (3.3)$$

Si par exemple $\sigma = 0.8$ on a le filtre 3x3 suivant :

$$\frac{1}{2\pi(0.8)^2} \exp\left(-\frac{(-1)^2 + (-1)^2}{2(0.8)^2}\right) = \frac{1}{16}$$


$G(-1, -1)$	$G(0, -1)$	$G(1, -1)$
$G(-1, 0)$	$G(0, 0)$	$G(1, 0)$
$G(-1, 1)$	$G(0, 1)$	$G(1, 1)$

 $\simeq \frac{1}{16} \cdot$

1	2	1
2	4	2
1	2	1

FIGURE 3.7 – Un exemple pour calculer la matrice de convolution gaussienne avec $\sigma = 0.8$

Le filtre gaussien donne un meilleur lissage et une meilleure réduction du bruit que le filtre moyenne. En général un filtre gaussien avec $\sigma = 1$ est utilisé pour réduire le bruit, et si $\sigma > 1$ c'est dans le but de fabriquer une image qu'on va utiliser pour faire un « masque flou » personnalisé. Il faut noter que plus σ est grand, plus le flou appliqué à l'image sera marqué.

- (b) Filtrage non-linéaire : Le filtrage local est dit non linéaire si nous ne pouvons pas exprimer un filtre par une combinaison linéaire. Ces filtres sont plus complexes à mettre en œuvre que les filtres linéaires. Cependant leurs résultats obtenus sont très souvent de meilleure qualité que ceux obtenus par les filtres linéaires. L'un des filtres non-linéaire les plus connus est le filtre médian, qui prend la valeur médiane des niveaux de gris de voisinage (3x3, 5x5, etc.). Ce type de filtre permet d'avoir un meilleur contraste de l'image par rapport aux filtres gaussien en assurant une bonne réduction de bruit. D'autres filtres existent aussi comme les filtres morphologiques, filtres FAS, filtres de Nasgao, etc.
- Filtre médian : Le filtre médian consiste à remplacer un pixel par la médiane de ses voisins. Ainsi, même si plusieurs pixels voisins sont bruités, on peut corriger le pixel courant. Ce filtre induit cependant un lissage puisque même des pixels corrects peuvent être modifiés. De plus, ce filtrage est plus coûteux car nécessite d'effectuer un tri des voisins pour chaque pixel. Plus le voisinage considéré est plus, plus l'algorithme sera coûteux. On pensera donc, lors de l'implémentation, à utiliser un algorithme de tri rapide tel que le quick sort. Le filtre médian permet d'obtenir de bons résultats sur du bruit

poivre et sel.

La figure suivante montre un exemple de calcul de la valeur médiane d'un voisinage d'un pixel :

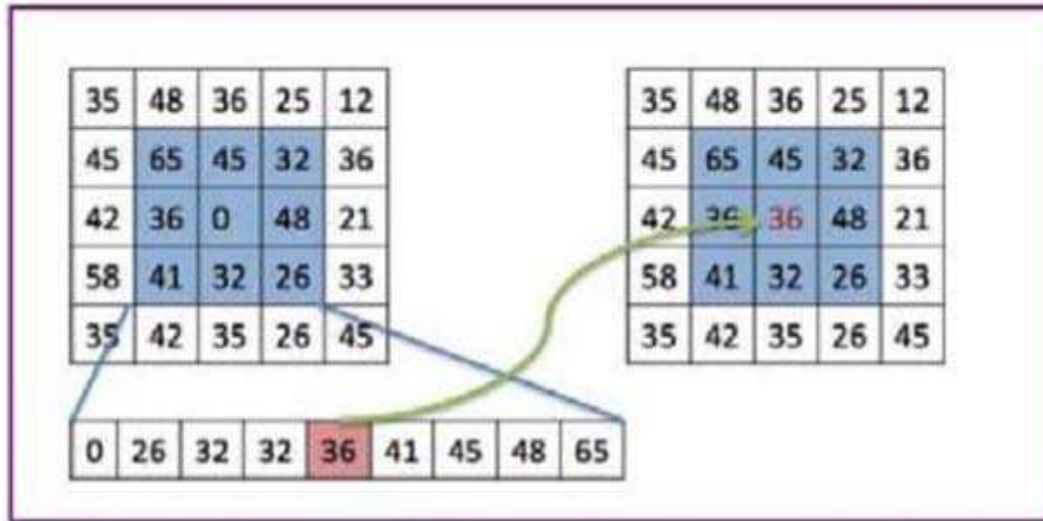


FIGURE 3.8 – Étapes du filtre médian

3. Correction de l'orientation (Redressement) : Le redressement (également appelée correction de « skew »), est une technique de correction de l'inclinaison qui a 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 qui peuvent apparaître.

En effet, la plupart des algorithmes de reconnaissance ne fonctionnent bien que si le document est bien redressé. Le redressement se fait selon deux niveaux [17] :

- Le redressement de la ligne de base : L'idée est de rendre horizontaux les chiffres à l'aide d'une transformation géométrique de type rotation isométrique des points de l'image.
- Le redressement des écritures penchées : Cette technique 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.

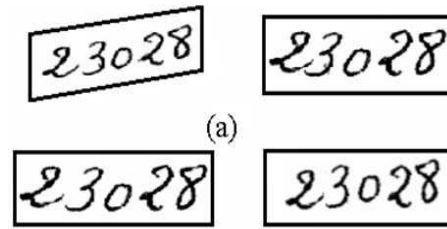


FIGURE 3.9 – Exemple de deux types de redressement.

3.2.1.3 Phase de segmentation

La segmentation est le processus consistant à décomposer l'image d'un texte en entités(mots, caractères) qui font partie d'un alphabet prédéfini selon le but visé. Par ailleurs, elle permet de réduire la complexité des modules de traitements utilisés par la suite [18]. Dans les systèmes de reconnaissance automatique la segmentation est une étape importante dans l'extraction des informations d'un document image, car un mauvais repérage des de l'information mènera probablement à un accroissement de la difficulté lors de l'étape de reconnaissance. La segmentation peut être effectuée à plusieurs niveaux :

1. Segmentation de la page : Cette étape permet de localiser les blocs d'information d'homogènes et de les classer en fonction de leur contenu : texte ou non texte. Cette séparation permet d'écarter les zones de graphique et de photographie du processus de reconnaissance de texte [19].
2. Segmentation d'un bloc de texte en lignes : Cette étape consiste à séparer les différentes lignes du texte pour en extraire les mots puis les caractères composants les mots. La segmentation EN LIGNES est réalisée en déterminant l'histogramme des projections horizontales d'un bloc de texte pour détecter les espaces entre les lignes et pouvoir les séparer [20].
3. Segmentation des lignes en mots : La segmentation en mots est réalisée en déterminant l'histogramme des projections verticales des lignes pour détecter les espaces entre les mots et pouvoir les séparer. Cependant cette technique peut ne pas être efficace dans certains cas où les mots se chevauchent (cas par exemple de l'écriture arabe). Dans ce cas d'autres techniques sont utilisées telles que : le suivi du contour, détermination du squelettes ou la détermination des composantes connexes
4. Segmentation des mots en caractères : La segmentation des caractères est une opération qui tente de décomposer une image de séquence de caractères

tères (mot) en sous-images de symboles individuels. C'est l'un des processus de décision dans un système de reconnaissance automatique. Son but est de décider si un motif isolé d'une image (caractère ou autre entité identifiable du mot) est correct ou non [21].

3.2.1.4 Phase de post-traitement : " la reconnaissance "

1. L'extraction de caractéristiques : Dans un système de reconnaissance automatique Le but ultime d'extraction de caractéristiques est obtenir le volume d'informations la plus pertinentes qui sera fourni au système. C'est une étape critique lors de la construction d'un système de reconnaissance. L'une des raisons pour laquelle cette étape pose un problème est qu'une plusieurs techniques d'extraction s'accompagne d'une perte d'information. De ce fait, il faut effectuer un compromis entre la quantité et la qualité de l'information [22]. dans notre projet on va utiliser la méthode de l'histogramme de gradient Orienté (Histogram of Oriented Gradients HOG).
 - (a) L'histogramme de gradient Orienté " HOG " : Le descripteur HOG a été proposé pour la première fois par Dalal et Triggs [23] comme descripteur d'image utilisé en vision par ordinateur et en traitement d'image pour la détection d'objets. L'idée de base de ce descripteur est que l'apparence et la forme des objets locaux peuvent être assez bien caractérisées par la distribution des gradients d'intensité locaux. Le filtre dégradé est appliqué dans les deux directions x et y de l'image. Les deux images obtenues sont ensuite transformées en gradients d'amplitude et d'orientation. Ensuite, ils sont divisés en petites régions spatiales (cellules). Pour chaque cellule, chaque pixel a une amplitude de gradient qui accumule la distribution au niveau du bin correspondant à sa valeur d'orientation. La concaténation de ces histogrammes donne l'histogramme HOG. Par exemple, si le nombre de cases d'orientation espacées de 0° à 180° est de 9 ($180^\circ / 20^\circ$) et que l'image est divisée en 3×4 cellules (12 est le nombre total de cellules), on obtient alors un histogramme de G avec $3 \times 4 \times 9 = 108bins$. En fait, l'histogramme obtenu n'est pas authentique car le cumul des cases n'atteint pas le nombre total de pixels. Un histogramme est finalement obtenu avec la normalisation $\sqrt{L2}$ [23].

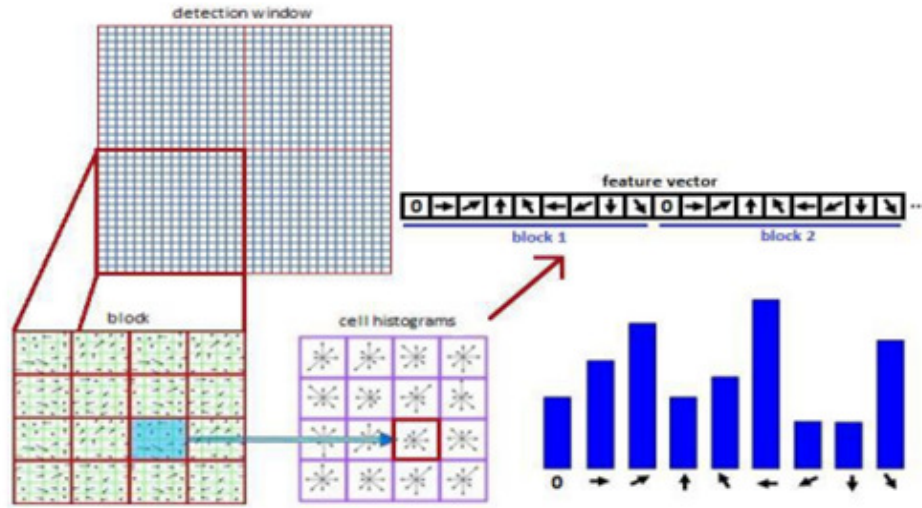


FIGURE 3.10 – Etapes du Descripteur HOG.

(b) Construction du descripteur :

- i. Calcul du gradient : Dans la première étape pour calculer un descripteur HOG, nous devons d'abord calculer les gradients horizontaux et verticaux, Ceci est réalisé en filtrant l'image avec les noyaux comme Sobel et Prewitt, Roberts... etc. Par la suite en calcul l'amplitude (g) et la direction (orientation O) du gradient comme suit :

$$g = \sqrt{dx^2 + dy^2} \quad (3.4)$$

$$O = \tan^{-1} \frac{dy}{dx} \quad (3.5)$$

On termine par calcul de l'histogramme des gradients.

- ii. Construction de l'histogramme : La seconde étape est la création des histogrammes de l'orientation des gradients. Ceci est fait sur des cellules carrées de petite taille (de 4x4 à 12x12 pixels).

Bloc 2*2 cellule

B1	B2	B3	B4
B5	B6	B7	B8
B9	B10	B11	B12
B13	B14	B15	B16

FIGURE 3.11 – Diviser l'image en blocs

Chaque pixel de la cellule vote alors pour une classe de l'histogramme, en fonction de l'orientation du gradient à ce point. Le vote du pixel est pondéré par l'intensité du gradient en ce point. Les histogrammes sont uniformes de 0 à 180° ou de 0 à 360°. Mais de bien meilleur résultat sont obtenus avec un histogramme à 9 bins[24].

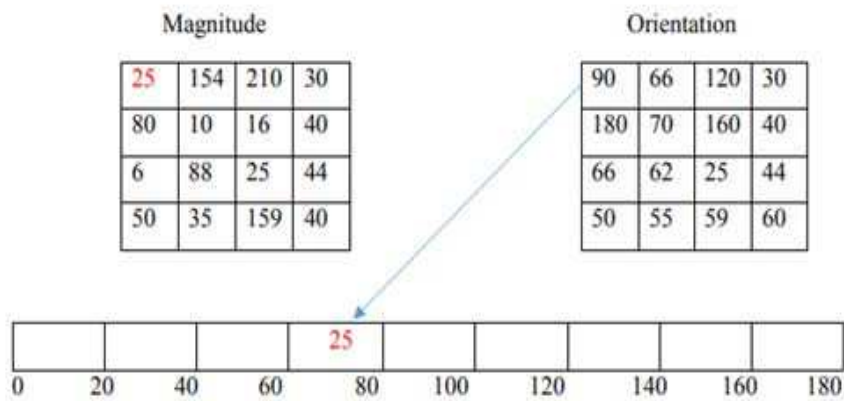


FIGURE 3.12 – Calcul de l'histogramme de bloc

- iii. Formation des blocs : Pour tenir compte des changements dans l'éclairage et le contraste, les forces de gradient doivent être normalisées au niveau local, ce qui nécessite le regroupement des cellules ensemble dans de plus grands blocs, spatialement connectés. Le descripteur de HOG est alors le vecteur concaténé des composants des histogrammes normalisés de cellules provenant de toutes les régions du bloc. Ces blocs se chevauchent généralement, ce qui signifie que chaque cellule contribue plus d'une fois au descripteur

final. Deux principales géométries de blocs existent : des blocs rectangulaires R-HOG et circulaires blocs C-HOG. R-blocs HOG sont généralement des grilles carrées, représentée par trois paramètres : le nombre de cellules par bloc, le nombre de pixels par cellule et le nombre de canaux par l'histogramme de la cellule[14].

- iv. Normalisation des blocs : Une étape importante est la normalisation des descripteurs afin d'éviter les disparités dues aux variations d'illumination. Cette étape introduit également de la redondance dans le descripteur. Pour cela, il est important de regrouper plusieurs cellules dans un bloc, qui est l'unité sur laquelle est effectuée la normalisation. Les blocs se recouvrent, donc une même cellule participe plusieurs fois au descripteur final, comme membre de blocs différents .

Dalal et Triggs ont exploré quatre méthodes différentes pour le bloc de normalisation. Soit v le vecteur non normalisé contenant tous les histogrammes dans un bloc donné, $\|v\|_k$ être son k -norme par $K = 1, 2$ et ϵ est une petite constante. Ensuite, le facteur de normalisation peut être l'un des éléments suivants :

$$L2 - norme : f = \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}} \quad (3.6)$$

$$L1 - norme : f = \frac{v}{\sqrt{\|v\|_1 + \epsilon}} \quad (3.7)$$

$$L1 - racine : f = \sqrt{\frac{v}{\sqrt{\|v\|_1 + \epsilon}}} \quad (3.8)$$

Une quatrième norme L2-hys, consistant à calculer v tout d'abord par la L2-norme, puis limiter les valeurs maximales de v à 0.2, et ensuite à normaliser. Les normes L2-Hys, L2-norme, et L1-racine obtiennent des performances similaires, tandis que L1-norme obtient de moins bons résultats, mais toutefois bien meilleurs que l'absence de normalisation [14].

2. Classification : L'étape finale dans le processus de détection d'objet est l'utilisation des descripteurs HOG pour entraîner un classifieur supervisé. Nous utiliserons le classifieur K-PPV pour fournir la décision de la classification.

- (a) K plus proches voisins K-NN (KPPV) : K plus proches voisins est une méthode de raisonnement à partir de cas. Elle part de l'idée de prendre des décisions en recherchant un ou des cas similaires déjà résolus en mémoire. Contrairement aux autres méthodes de classification (Arbres de Décision, Réseaux de Neurones, Algorithmes Génétiques). C'est l'une des méthodes les plus simples d'apprentissage automatique supervisé.
- (b) Principe de classifieur KPPV : Etant donnée une base d'apprentissage d'images, pour prédire la classe d'un nouveau cas, le classifieur KPPV cherche les K plus proches voisins de ce nouveau cas et prédit la réponse la plus fréquente de ces K plus proches voisins. La méthode utilise donc deux paramètres : le nombre K et la fonction de similarité pour comparer le nouveau cas aux cas déjà classés. Le principe est donné par :
- Choix d'un entier k (souvent $k = \text{nombre d'attributs} + 1$, où $k = \sqrt{n}$).
 - Calcul des distances (ex : distance euclidienne, c'est la distance la plus populaire).
 - Retenir les k observations pour lesquelles ces distances sont les plus petites (les k plus proches voisins).
 - Compter les nombres de fois où ces k observations apparaissent dans chacune des classes (déterminer les classes correspondantes).
 - Choisir la classe la plus représentée.

Ces étapes sont résumées comme suit : Pour une forme inconnue x à classer, nous allons examiner la distance de x à tous les échantillons (qui définissent toutes les classes), puis nous sélectionnons les K plus proches échantillons et nous affectons x à la classe majoritaire parmi ces K échantillons.[25] Dans l'exemple de la figure suivante et pour cinq voisins on classerait x dans la classe w1 :

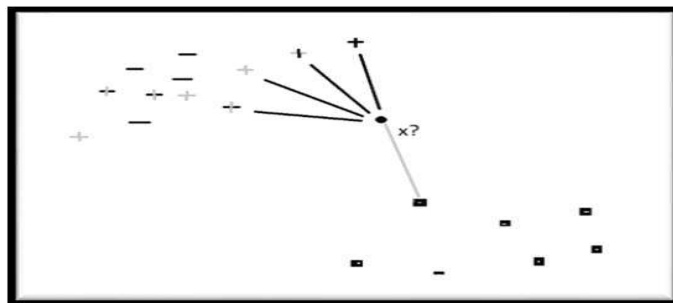


FIGURE 3.13 – Méthode de K plus proches voisins.

- (c) Choix du nombre K : Le choix du paramètre K dans la règle des KPPV a une influence directe sur le style du classifieur ainsi déterminé. Une faible valeur de K va donner un classifieur de bonne résolution (définir des frontières compliquées entre classes) mais très sensible au bruit sur les échantillons et sur le vecteur à classer. Une valeur grande de K aura un comportement inverse, lissant les frontières mais peu sensible au bruit.[26]
- (d) Définition de la distance : Le choix de la distance est primordial au bon fonctionnement de la méthode. Quoique les distances les plus simples permettent d'obtenir des résultats satisfaisants. Une distance doit avoir quatre propriétés pour tous les vecteurs a, b et c :[27]
- $D(a,b) \geq 0$ (non-négativité)
 - $D(a,b)=0$ si et seulement si $a=b$ (réflexivité)
 - $D(a,b)=D(b,a)$ (symmetrie)
 - $D(a,b)+D(b,c) \geq D(a,c)$. (Triangle inégalité)

On peut cependant noter qu'un point « a » peut avoir un plus proche voisin « b » Tandis que « b » possède de nombreux voisins plus proches que « a ». Le choix de la distance se fait en fonction des connaissances préalables du problème. Il est possible de choisir la distance en faisant varier cette distance et, pour chacun des choix, estimer l'erreur réelle. On choisit alors la distance donnant la meilleure erreur réelle estimée. Plusieurs types pour les distances :

- Distance Euclidienne.
- Distance "city block" ou "Manhattan" (somme des valeurs absolues).
- Distance de Tchebycheff.
- Distance de Mahalanobis.
- etc.

Dans notre cas nous avons utilisé la mesure de similarité cosinus qui permet de calculer la similarité entre deux vecteurs à n dimension en déterminant le cosinus de l'angle entre eux.

$$\cos \theta = \frac{A.B}{\|A\| \|B\|} \quad (3.9)$$

- (e) Sélection de la classe : La méthode la plus simple est de rechercher le cas le plus proche et de prendre la même décision. C'est la méthode 1-PPV (1-NN) du plus proche voisin. Si cette méthode peut fournir de bons résultats sur des problèmes simples pour lesquels les points sont bien répartis en groupes denses de même classe, en règle générale, il faut considérer un nombre de voisins plus important pour obtenir de

bons résultats. Une première façon de combiner les k classes des k voisins les plus proches est le vote majoritaire. Elle consiste simplement à prendre la classe majoritaire. Une seconde façon est le vote majoritaire pondéré. Chaque vote, c'est-à-dire Chaque classe d'un des k voisins sélectionnés, est pondéré. Le poids est inversement proportionnel à la distance entre le cas à classer et les autres k plus proches voisin classes (Une telle pondération s'appelle un noyau). Dans les deux cas précédents, il est possible de définir une confiance dans la classe attribuée égale au rapport entre les votes gagnants et le total des votes chaque classe d'un des k voisins sélectionnés, est pondéré. Le poids est inversement proportionnel à la distance entre le cas à classer et les autres k plus proches voisin classes (Une telle pondération s'appelle un noyau). Dans les deux cas précédents, il est possible de définir une confiance dans la classe attribuée égale au rapport entre les votes gagnants et le total des votes.

3.3 La conception

3.3.1 Présentation d'UML

Unified Modeling Language est un langage unifié de modélisation objets. Ce n'est pas une méthode, il ne donne pas de solution pour la mise en œuvre d'un projet. C'est avant tout un formalisme graphique issu de notations employées dans différentes méthodes objets. Nous essayerons dans ce qui suit de décrire les différentes fonctionnalités qu'offrira notre système à l'aide des différents types de diagrammes UML.

3.3.2 Diagramme des cas d'utilisation

Un cas d'utilisation est un résumé des scénarios pour un but ou une tâche unique. Un acteur est la personne ou l'objet qui engage les événements impliqués dans cette tâche. Notre diagramme de cas d'utilisation sera présenté comme suit :

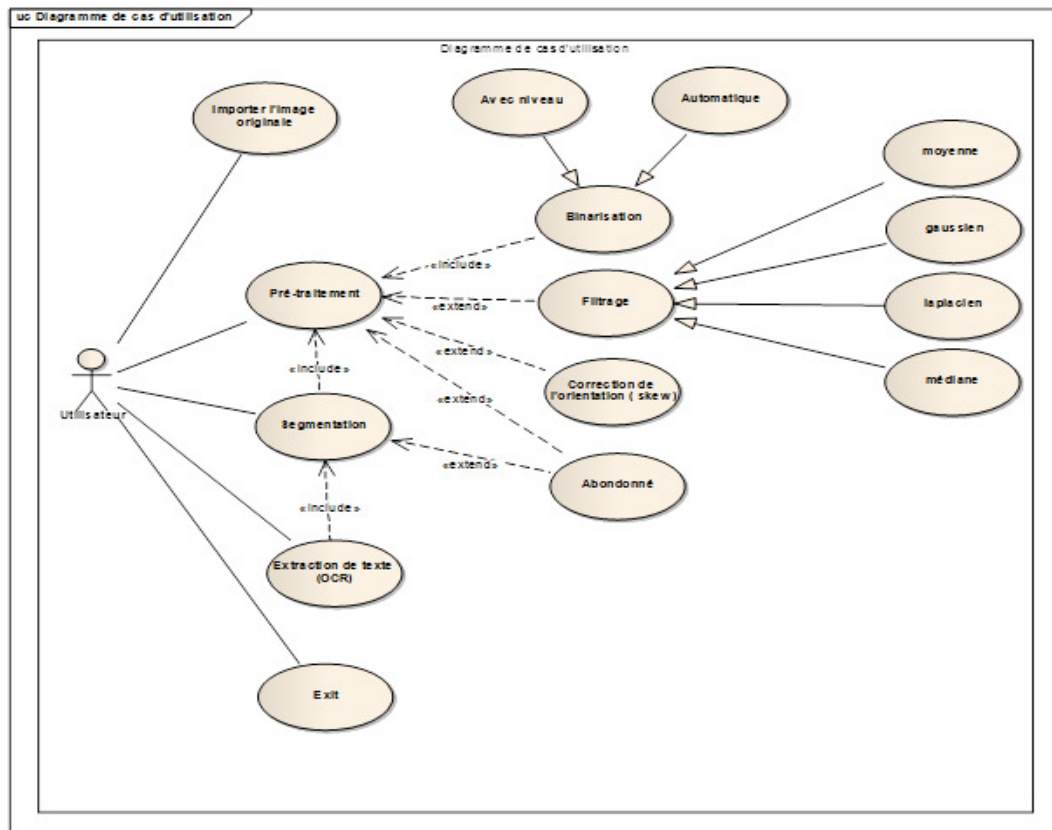


FIGURE 3.14 – Diagramme des cas d'utilisation

3.3.3 Diagramme des classes

Le diagramme de classes permet de définir la structure de toutes les classes qui constituent un système. Une classe est définie en plus de son nom par des attributs et des méthodes.

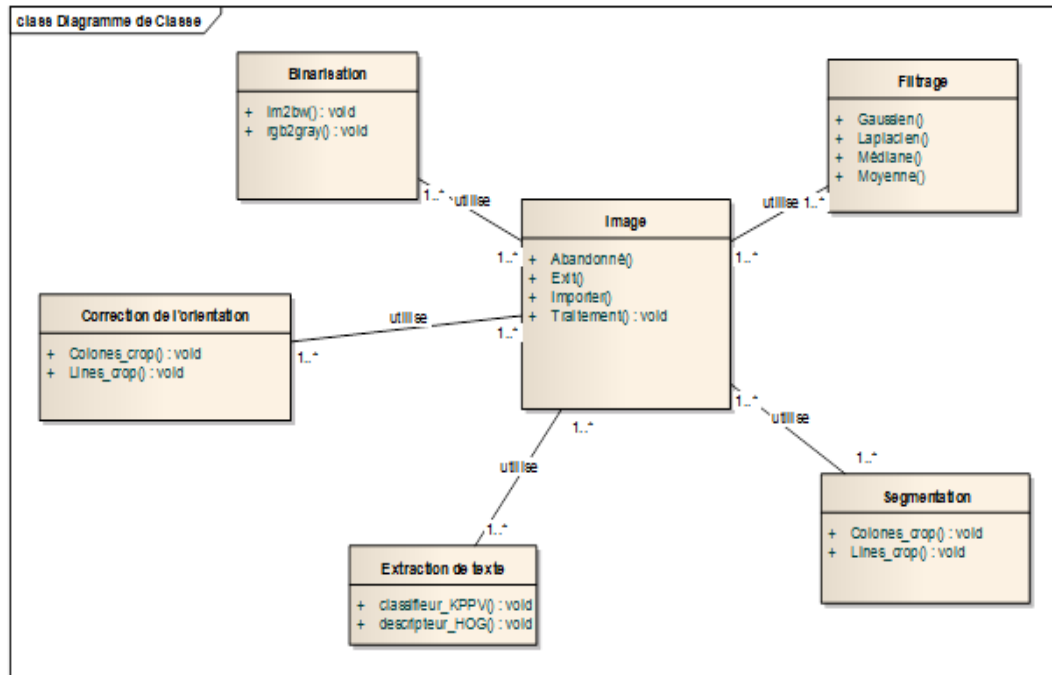


FIGURE 3.15 – Diagramme des classes

3.4 Conclusion

Dans ce chapitre nous avons illustré la logique de notre approche ainsi que les étapes de cette dernière après nous avons construit le diagramme de cas d'utilisation et le diagramme de classe tout en s'appuyant sur le langage UML. Le chapitre suivant sera consacré à la partie de la réalisation de l'application MATLAB et les résultats expérimentaux de notre méthode.

Implémentation et Bilan

Sommaire

4.1	Introduction	32
4.2	Les outils et langage utilisés	32
4.2.1	Langage de calcule technique MATLAB	32
4.2.2	Guide MATLAB (Graphical User Interface Development Environment)	32
4.2.3	Environnement de développement MATLAB (Matlab R2016a)	34
4.3	Présentation les interfaces de l'application	35
4.3.1	L'acquisition	36
4.3.2	Pré-traitement	36
4.3.3	La segmentation	40
4.3.4	La reconnaissance	41
4.4	Evaluation	42
4.4.1	Bases de données	42
4.4.2	Le taux moyen de bonne reconnaissance	43
4.5	Conclusion	43

4.1 Introduction

Ce chapitre est le dernier de notre travail, il sera consacré à la mise en œuvre de notre application en indiquant les outils de développement, les langages de programmation utilisés pour son implémentation. Nous présentons ensuite les résultats de nos tests expérimentaux obtenus grâce à notre application.

4.2 Les outils et langage utilisés

4.2.1 Langage de calcul technique MATLAB

MATLAB une abréviation de Matrix LABoratory, écrit à l'origine, en Fortran. par C Moler, est un langage de programmation de quatrième génération émulé par un environnement de développement du même nom ; il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre des algorithmes, de créer des interfaces utilisateurs, et peut s'interfacer avec d'autres langages comme le C, C++, Java, et Fortran. Les utilisateurs de MATLAB (environ un million en 2004) sont de milieux très différents comme l'ingénierie, les sciences et l'économie dans un contexte aussi bien industriel que pour la recherche. Matlab peut s'utiliser seul ou bien avec des *toolbox* (boîte à outils).

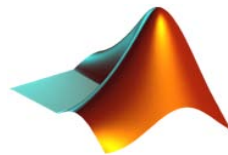


FIGURE 4.1 – Logo de MATLAB.

4.2.2 Guide MATLAB (Graphical User Interface Development Environment)

Les interfaces graphiques (ou interfaces homme-machine) sont appelées GUI (Graphical User Interface) sous MATLAB. Elles permettent à l'utilisateur d'interagir avec un programme informatique, grâce à différents objets graphiques

(boutons, menus, cases à cocher...). Ces objets sont généralement actionnés à l'aide de la souris ou du clavier. Malgré le fait que les interfaces graphiques semblent secondaires par rapport au développement du coeur d'une application, elles doivent néanmoins être conçues et développées avec soin et rigueur. Leur efficacité et leur ergonomie sont essentielles dans l'acceptation et l'utilisation de ces outils par les utilisateurs finaux. Une bonne conception et un développement maîtrisé permettent également d'en assurer une meilleure maintenabilité. MATLAB possède un outil dédié à la création des interfaces graphiques appelé GUIDE (pour Graphical User Interface Development Environment). Le GUIDE est un constructeur d'interface graphique qui regroupe tous les outils dont le programmeur a besoin pour créer une interface graphique de façon intuitive. Il s'ouvre, soit en cliquant sur l'icône, soit en tapant guide dans le Command Window de MATLAB.

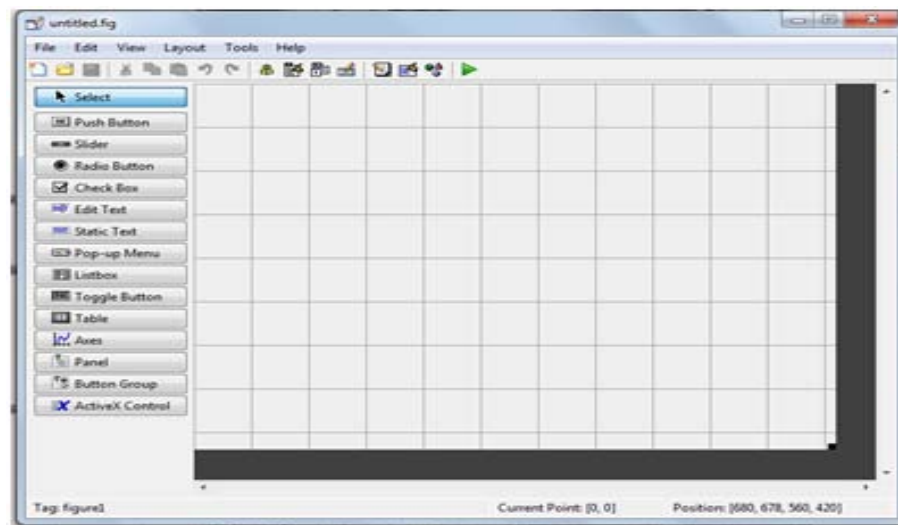


FIGURE 4.2 – Fenêtre principale du GUIDE.

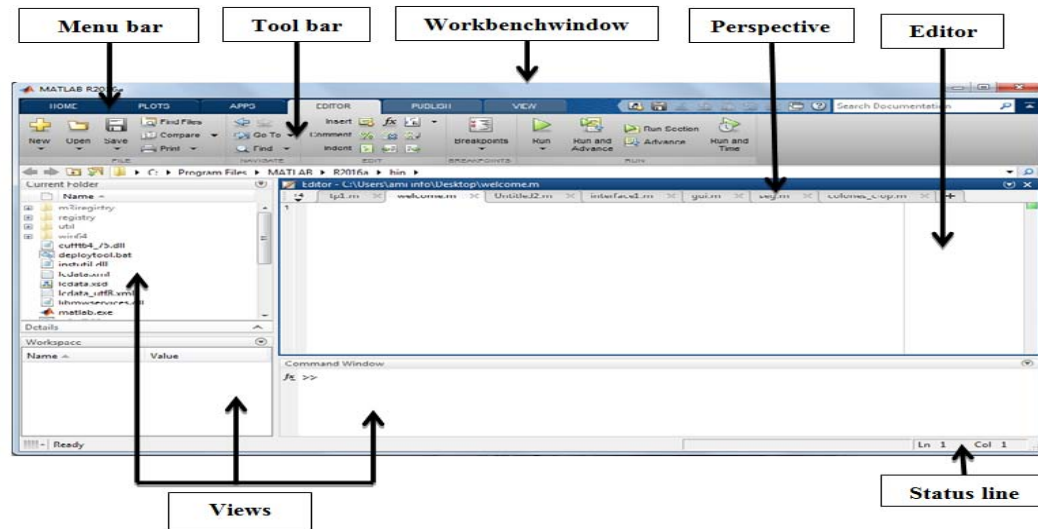


FIGURE 4.4 – Interface MATLAB R2016a.

4.3 Présentation les interfaces de l'application

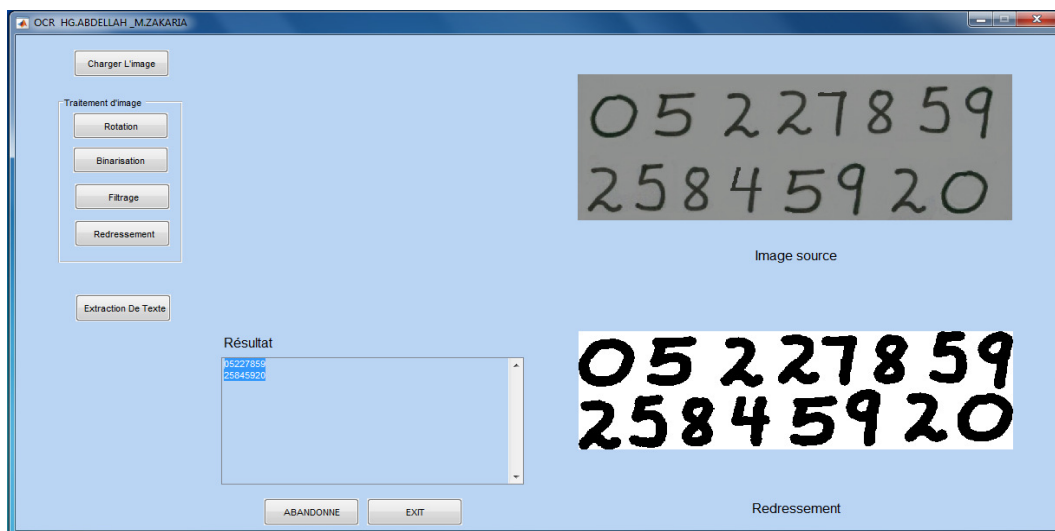


FIGURE 4.5 – Interface de notre application.

Dans ce qui suit nous allons détaillé ces différentes phases :

4.3.1 L'acquisition

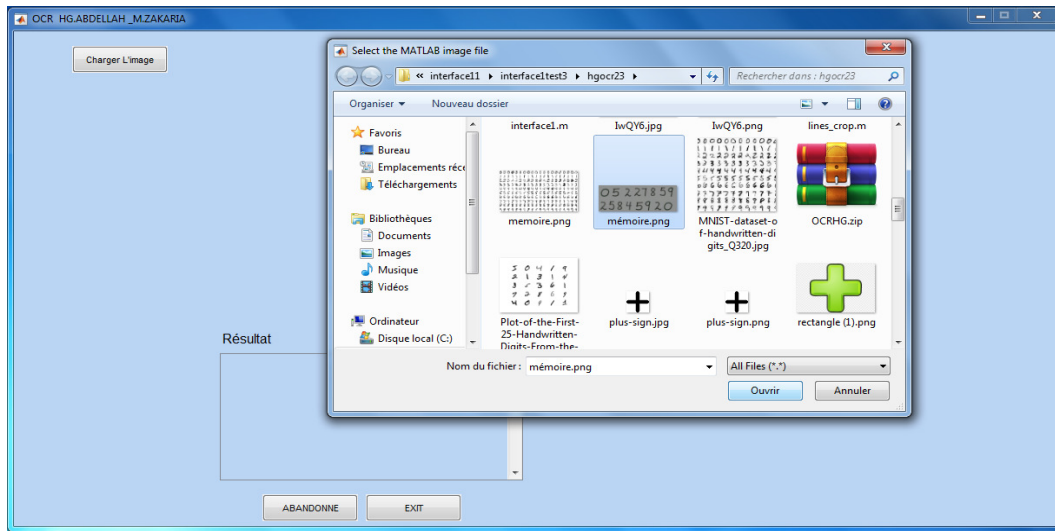


FIGURE 4.6 – Pour charger l'image.

4.3.2 Pré-traitement

Le prétraitement se compose de :

1. La binarisation : Notre application offre 2 possibilités, L'utilisateur a le choix entre les paramètres par défaut (automatique) ou ajuster lui même le niveau de binarisation.

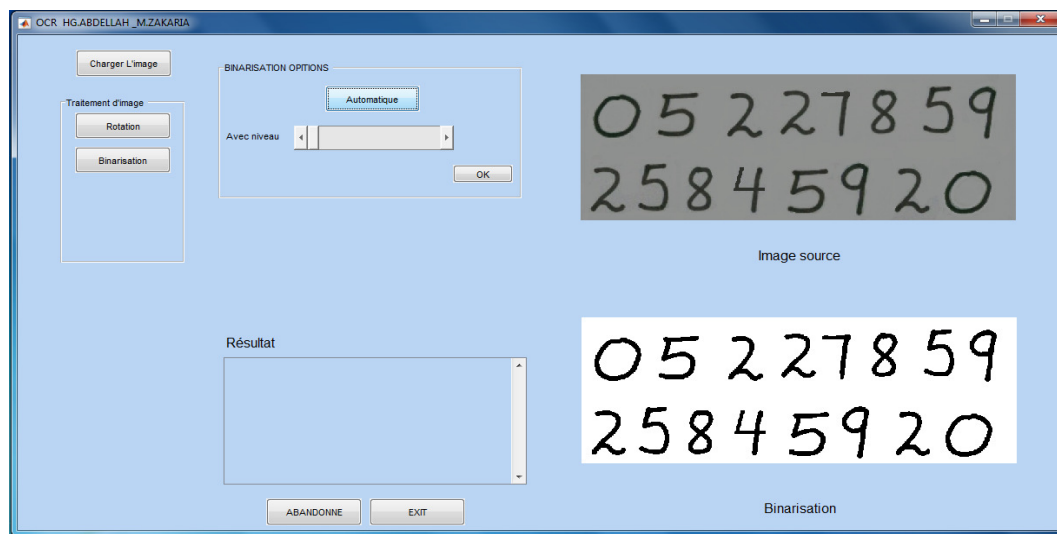


FIGURE 4.7 – Pour la binarisation «automatique».

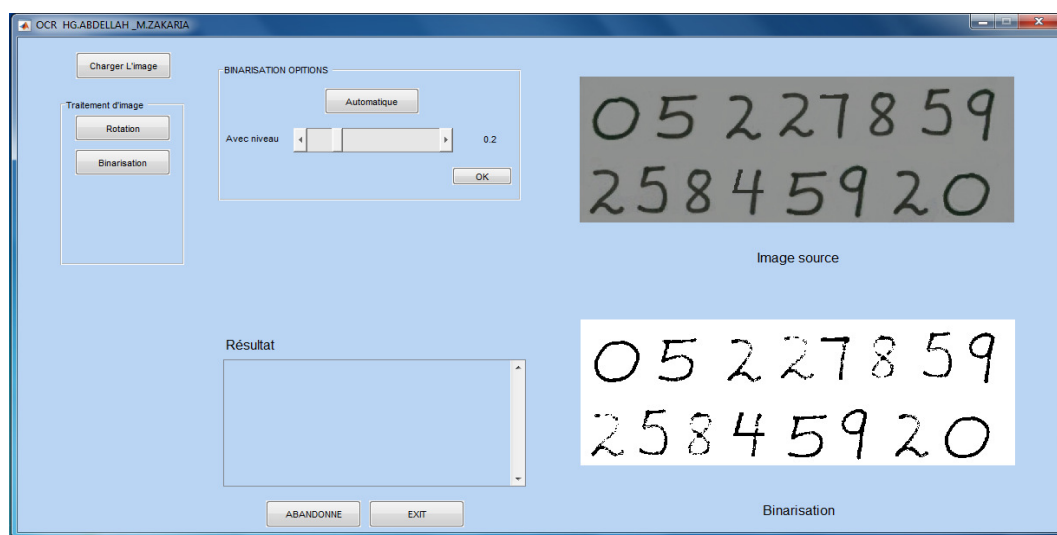


FIGURE 4.8 – Pour la binarisation «avec niveau »

2. Le filtrage : Les différents filtres disponibles dans notre application sont :

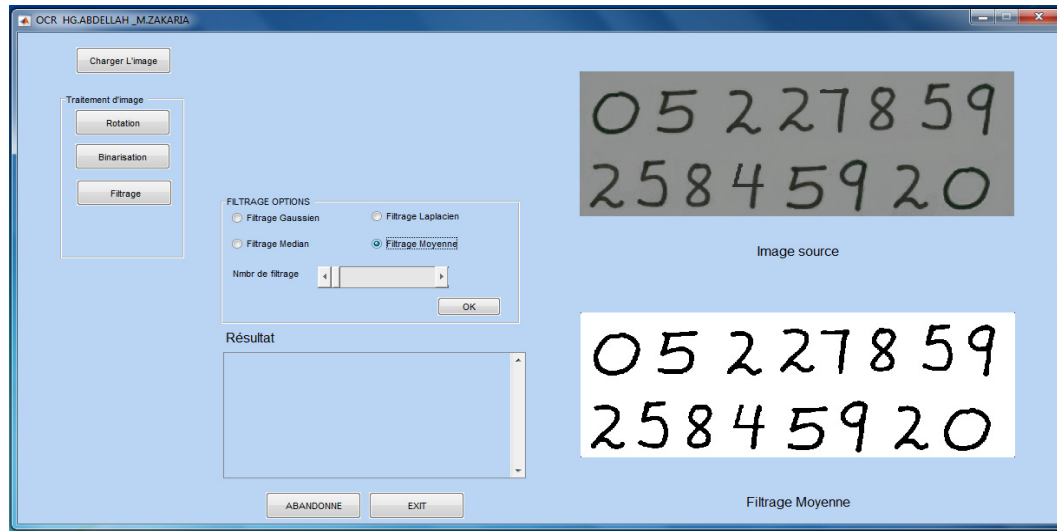


FIGURE 4.9 – Pour le filtrage «avec filtre moyennneur »

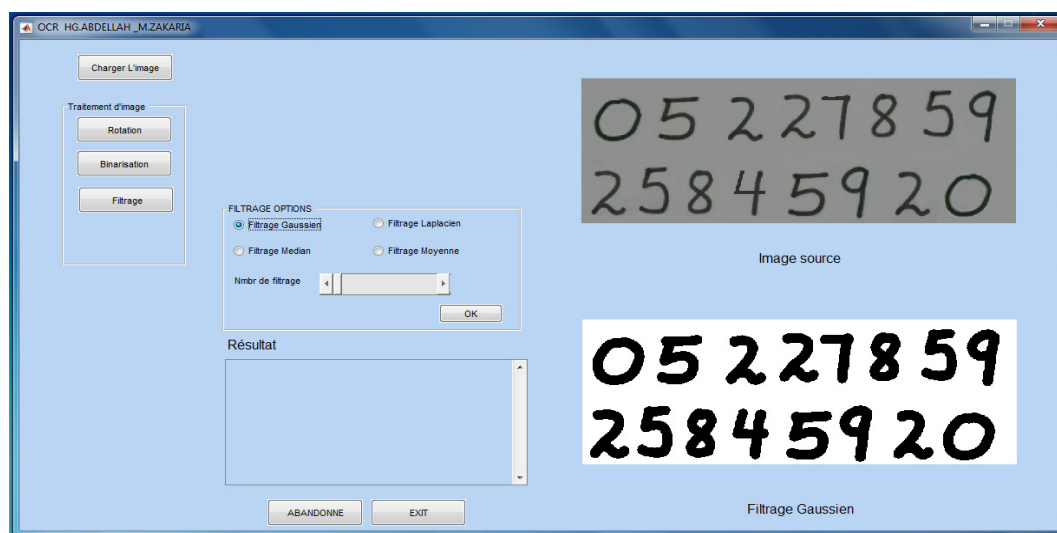


FIGURE 4.10 – Pour le filtrage «avec filtre gaussien»

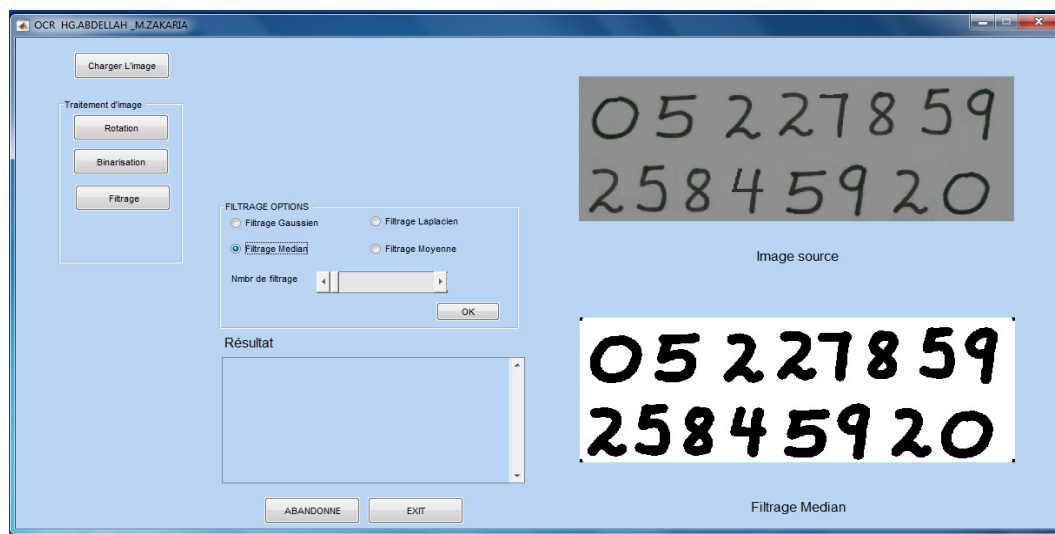


FIGURE 4.11 – Pour le filtrage «avec filtre median »

3. Le Redressement : Dans notre application, il possible d'ajuster l'alignement du texte (lignes) :

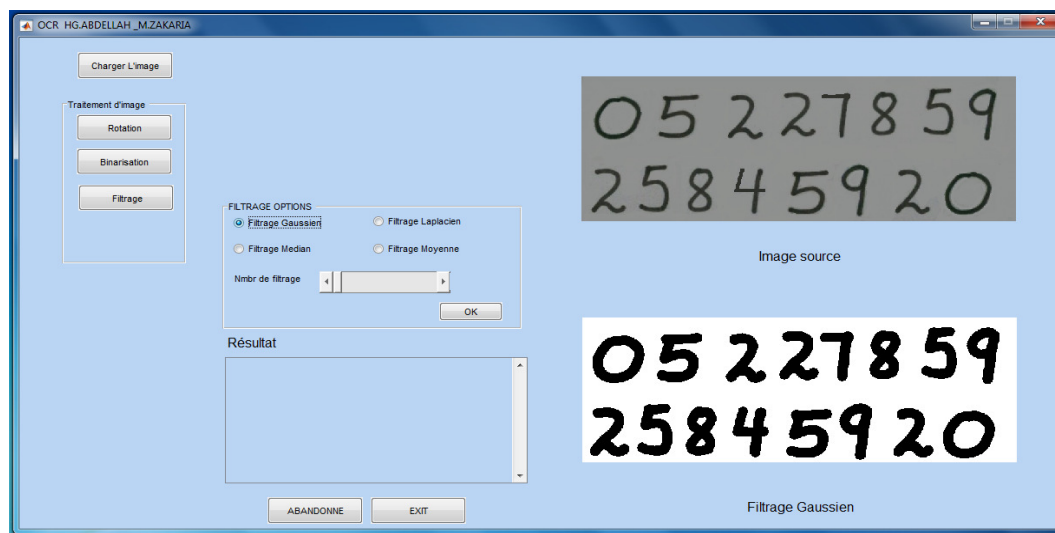


FIGURE 4.12 – L'image « avant » l'alignement.

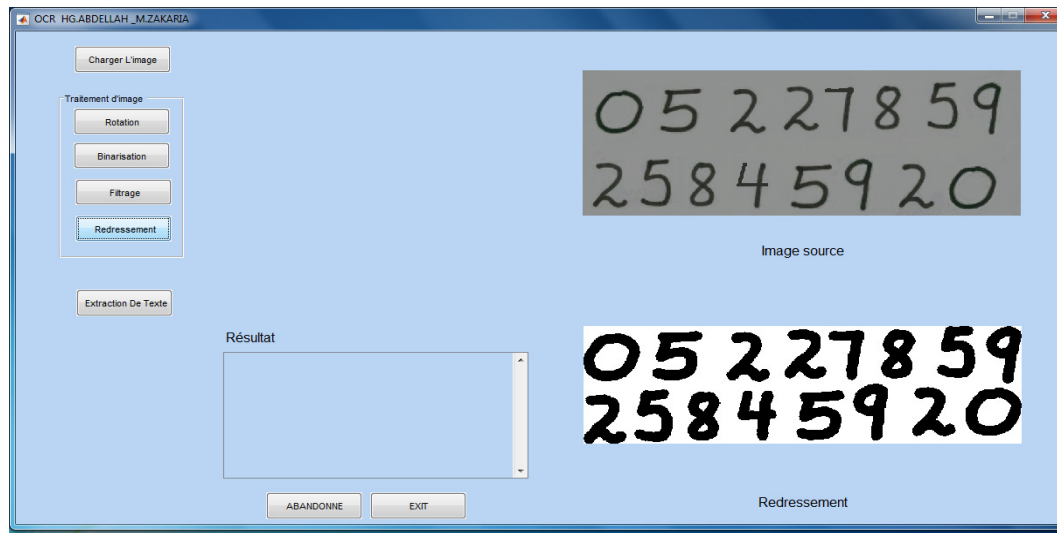


FIGURE 4.13 – L'image « après » l'alignement.

4.3.3 La segmentation

Dans notre application, nous segmentons d'abord les différentes lignes et pour chaque ligne nous extrayons les différents caractères.

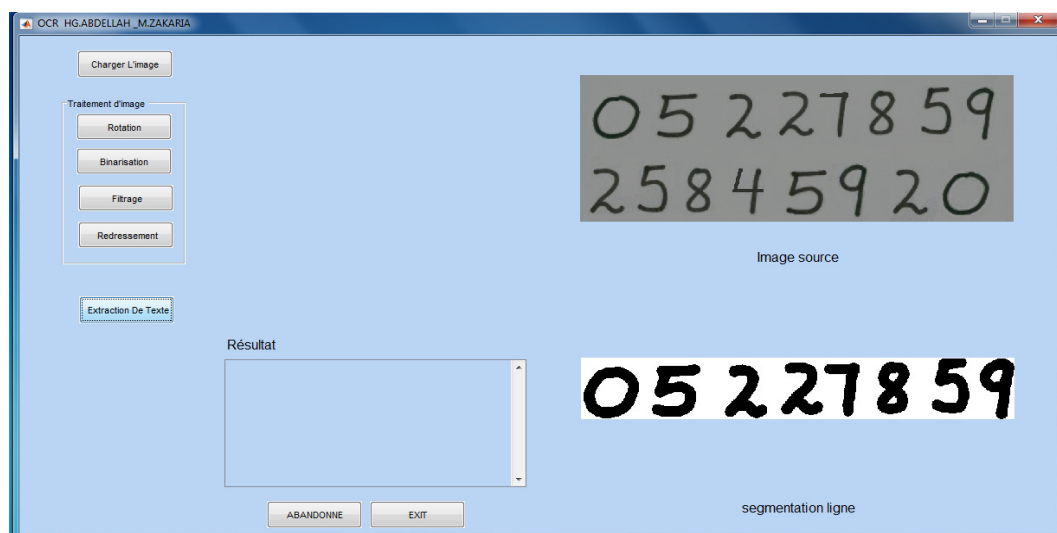


FIGURE 4.14 – La segmentation en lignes.

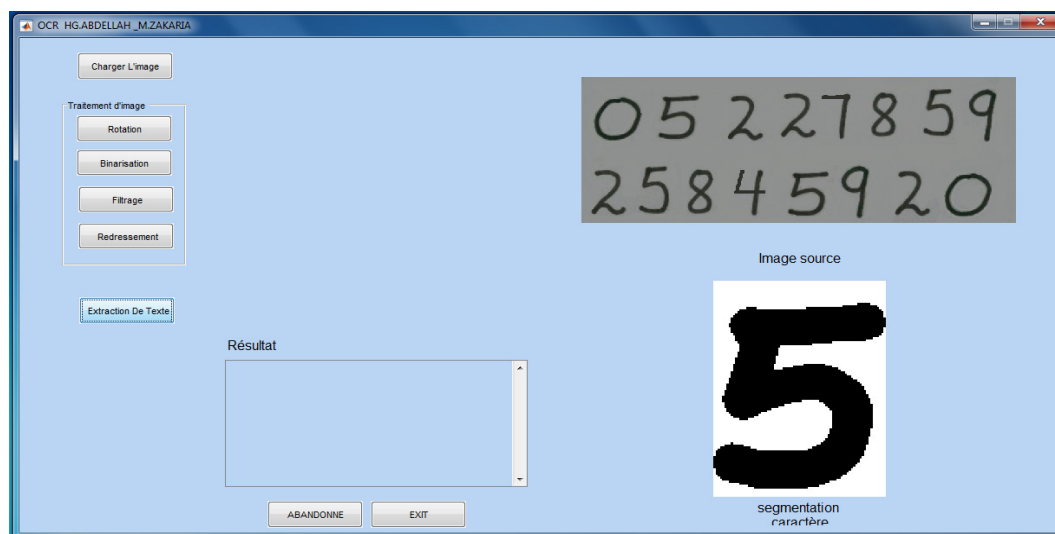


FIGURE 4.15 – La segmentation en caractères.

4.3.4 La reconnaissance

Après les pré traitements et la segmentation nous pouvons extraire le texte à partir de l'image que nous pouvons l'exporter vers un éditeur de texte tel que Word, Notepad... :

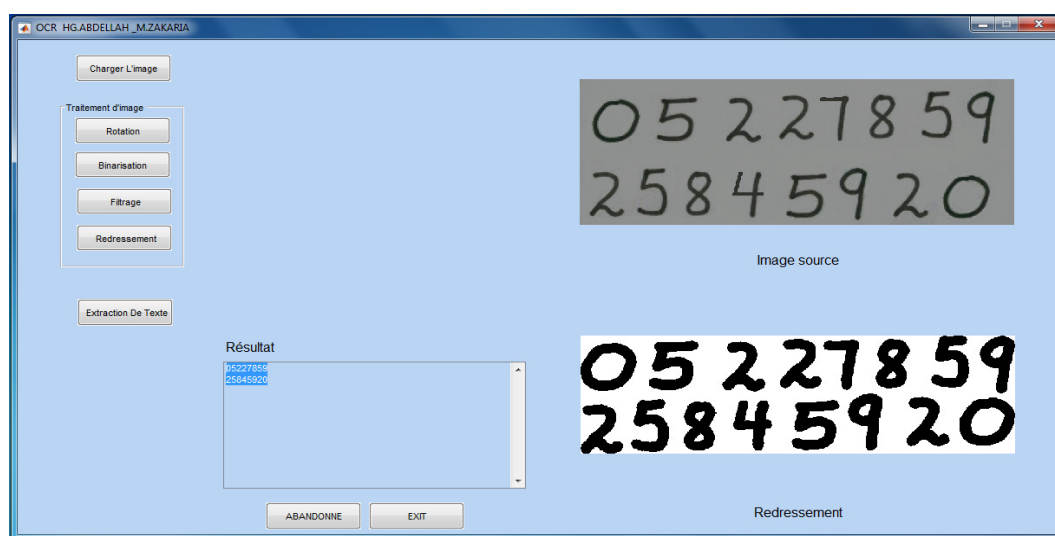


FIGURE 4.16 – Le résultat obtenu.

4.4 Evaluation

4.4.1 Bases de données

Les expériences ont été conduites sur une base de données composée de chiffres manuscrits isolés sous forme d'images binaires (figure 26.3). Cette base, appelée la base de données MNIST (Mixed National Institute of Standards and Technology) est une grande base de chiffres manuscrits qui est couramment utilisé pour divers systèmes de traitement d'image.[28] La base de données est également largement utilisée pour l'apprentissage et les tests. Elle a été créé par "re-mélange" les échantillons provenant des ensembles de données d'origine du NIST. En outre, les images en noir et blanc du NIST ont été normalisées insérer dans une boîte 28x28 pixel englobant.[29] La base de données MNIST contient 60.000 images d'apprentissage et de 10.000 images de test.

Classes	Apprentissage	Test	Total
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
Total	60000	10000	70000

TABLE 4.1 – Répartition de la base MNIST pour les chiffres



FIGURE 4.17 – Exemples de la base de données MNIST.

4.4.2 Le taux moyen de bonne reconnaissance

le taux moyen de bonne reconnaissance (TMBR) qui est défini comme étant le nombre de chiffres reconnus sur le nombre totale du chiffres exprimé en (

$$TMBR = \frac{\text{nombre de chiffres reconnus}}{\text{nombre total de chiffres}} \quad (4.1)$$

$$TMBR = \frac{\text{sum}(CT == CLT)}{\text{length}(CLT)} \quad (4.2)$$

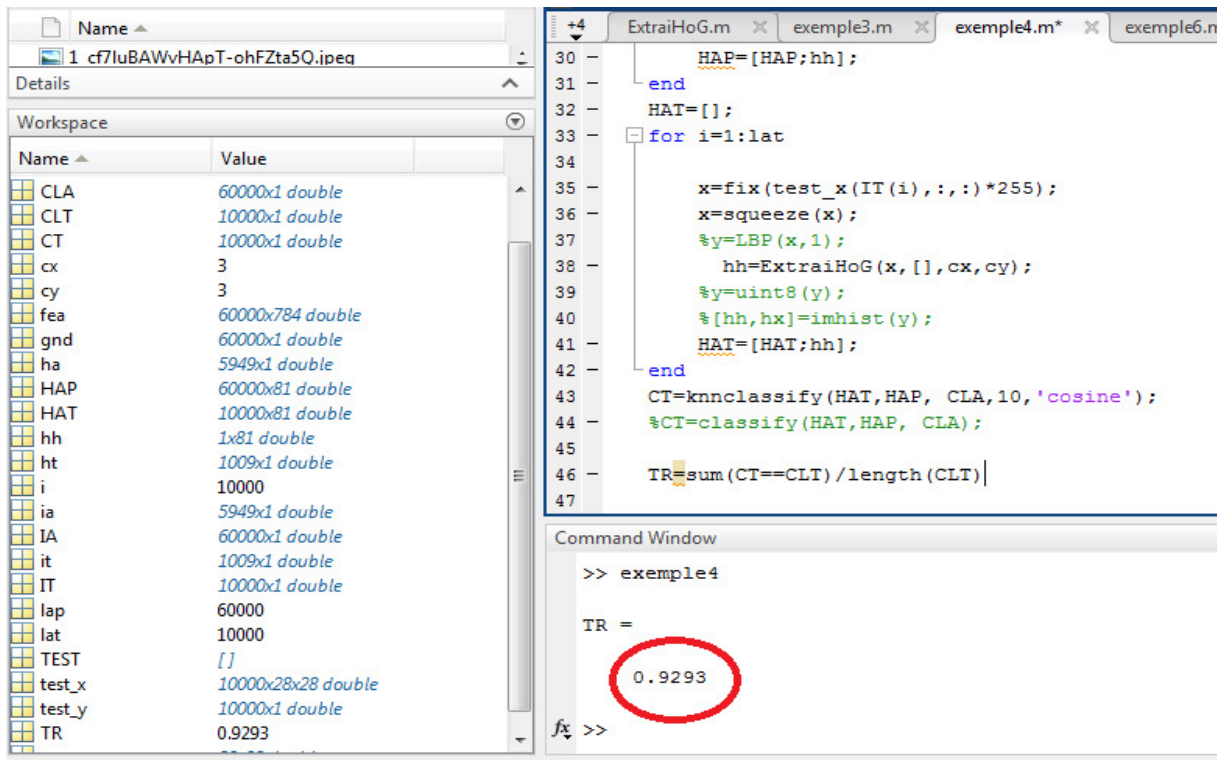


FIGURE 4.18 – Le taux moyen de bonne reconnaissance.

4.5 Conclusion

Dans ce chapitre nous avons présenté les démarches que nous avons suivies pour le développement de la plateforme de système de reconnaissance automatique, les différentes étapes de la réalisation de notre application, et à la fin nous avons présenté les différents résultats expérimentaux de méthode développée.

Chapitre 5

Conclusion générale

Sommaire

5.1	Conclusion générale	46
5.1.1	Synthèse des chapitres	46
5.1.2	Perspectives	47

5.1 Conclusion générale

Dans cette mémoire, nous avons présenté nos contributions dans le cadre de la reconnaissance d'écriture manuscrite. Nous avons adopté principalement sur la méthode d'histogramme de gradient Orienté (Histogram of Oriented Gradients HOG).

5.1.1 Synthèse des chapitres

Dans le chapitre 1, nous avons abordé une présentation d'étude bibliographique sur le système de reconnaissance automatique de caractère OCR, nous avons également introduit certains facteurs principaux à la nécessité du développement et de la création de ce système. Enfin, nous avons présenté nos contributions et nos axes d'amélioration ainsi que nos objectifs. La variabilité de l'écriture manuscrite ainsi que les différentes applications, fait que la reconnaissance des caractères demeure un domaine de recherche très vivace. Ce qui nous a conduits dans le chapitre 2, à dresser un état de l'art sur les méthodes ultérieures qui 'ils ont réalisé sur la base de données M.N.I.S.T. La revue de littérature du domaine de reconnaissance des caractères manuscrits, permet de constater la grande diversité des techniques d'extraction de caractéristiques, et malgré les efforts et le progrès important fournis par la communauté des chercheurs, aucun système actuel n'est jugé fiable à 100%. Nous avons ainsi consacré le 3ème chapitre aux différentes étapes suivies par la conception de notre solution. Après une étude approfondie des techniques utilisées pour la reconnaissance automatique de caractère OCR, nous avons présenté les quatre principaux étapes du système de reconnaissance automatique de caractère, en commençant par l'étape d'acquisition, prétraitement de l'image, puis segmentation, et l'étape de post-traitement « l'extraction des caractéristiques et la classification ». L'étape d'acquisition repose essentiellement sur la capture de l'image de l'écriture manuscrite au moyen des capteurs physiques et de la convertir en élévations numériques adaptés au système de traitement informatisés avec un minimum de dégradation possible. L'étape de prétraitement a été le premier défi que nous avons rencontré lors de la mise en œuvre du programme. C'est une étape nécessaire, importante et très efficace, parce qu'elle permet de faciliter tous les traitements ultérieurs tels que la segmentation, l'extraction de primitives et la classification. Parmi les opérations de prétraitements : La binarisation pour séparer les pixels de texte des pixels de l'arrière-plan de l'image traitée, le filtrage pour améliorer la qualité visuelle de l'image et d'obtenir une image proche à la réalité, le redressement pour cor-

riger le défaut de positionnement du papier dans le scanner et aussi il dresser l'inclinaison interne de certaines parties du document qui peuvent apparaître. L'étape de segmentation est composée de trois phases, à savoir la segmentation de la page pour localiser les blocs d'information d'homogènes et de les classer en fonction de leur contenu texte ou non texte, la segmentation en lignes pour séparer les différentes lignes du texte, nous avons utilisé la technique de la projection horizontale, la segmentation en mots pour séparer les différentes mots du ligne, nous avons utilisé la technique de la projection verticale, la segmentation en caractères de décomposer une image de séquence de caractères (mot) en sous-images de symboles individuels. En ce qui concerne l'étape de post-traitement, elle passe par deux phases, à savoir L'extraction de caractéristiques et la classification : L'extraction des caractéristiques a pour but de décrire une donnée (image d'un caractère) par un vecteur de caractéristiques. La similarité ou dis similarité entre les caractères est alors mesurée sur la base de leurs vecteurs de caractéristiques. C'est une étape importante d'un système efficace de reconnaissance de caractères. Nous avons utilisé la méthode de l'histogramme de gradient Orienté « HOG » pour l'extraction des caractéristiques. La phase de classification consiste à trouver la classe la plus appropriée à l'image de caractères d'entrée. . Nous avons utilisé le classifieur K-PPV pour fournir la décision de la classification. Enfin, nous avons présenté les différentes fonctionnalités de notre système à l'aide des différents types de diagrammes UML. Dans le 4ème chapitre, nous avons présenté les différentes interfaces de notre application, et les différents résultats expérimentaux de méthode développée.

5.1.2 Perspectives

Les perspectives ouvertes par ces travaux sont très nombreuses et peuvent se résumer en plusieurs points :

- Essayez d'améliorer le prétraitement de l'image, par exemple en utilisant des autres technique..
- Utiliser d'autre descripteurs ou fusionner plusieurs descripteurs ensemble.

Bibliographie

Bibliographie

- [1] F. Leon E. Tautu. Optical character recognition system using support vector. page 13, 07 2012.
- [2] Jyoti Mahajan and Rohini Mahajan. Designing an intelligent system for optical handwritten character recognition using ann. *International Journal of Computer Applications*, 91, 03 2014.
- [3] R. Plamondon and S. N. Srihari. Online and off-line handwriting recognition : a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1) :63–84, 2000.
- [4] Nafiz Arica and Fatos Yarman Vural. An overview of character recognition focused on off-line handwriting. *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, 31 :216 – 233, 06 2001.
- [5] Ujjwal Bhattacharya and Bidyut Chaudhuri. Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals. *IEEE transactions on pattern analysis and machine intelligence*, 31 :444–57, 04 2009.
- [6] Umapada Pal, T Wakabayashi, Nabin Sharma, and F Kimura. Handwritten numeral recognition of six popular indian scripts. pages 749–753, 10 2007.
- [7] Guy Mayraz and Geoffrey Hinton. Recognizing hand-written digits using hierarchical products of experts. 13, 02 2001.
- [8] Ernst Kussul and Tatyana Baidyk. Improved method of handwritten digit recognition tested on mnist database. *Image Vision Comput.*, 22 :971–981, 01 2004.

- [9] Ming Wu and Zhen Zhang. Handwritten digit classification using the mnist data set. 09 2020.
- [10] Yukihiro Yamashita. Toru wakahara ; k-nn classification of handwritten characters using a new distortion-tolerant matching measure. In ., *Pattern Recognition (ICPR), 2014, 2nd International Conference on*, Stockholm.
- [11] Chen-Yu Lee, Patrick Gallagher, and Zhuowen Tu. Generalizing pooling functions in convolutional neural networks : Mixed, gated, and tree. 09 2015.
- [12] Yukihiro Yamashita and Toru Wakahara. Affine-transformation and 2d-projection invariant k-nn classification of handwritten characters via a new matching measure. *Pattern Recognition*, 52, 10 2015.
- [13] Aysegul Dundar, Jonghoon Jin, and Eugenio Culurciello. Convolutional clustering for unsupervised learning. 11 2015.
- [14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection". In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, page 886–893.
- [15] Guy LORETTE Jean-Pierre CRETTEZ. Reconnaissance de l'écriture manuscrite. *Signal Processing*, pages 49–77, 10 1998.
- [16] Djamel GACEB. *Contributions au tri automatique de documents et de courrier d'entreprises*. PhD thesis, 2009.
- [17] Fares Menasri. *Contributions 'a la reconnaissance de l'écriture arabe manuscrite*. PhD thesis, 2008.
- [18] Abdel Belaïd et Yolande Belaïd. *Reconnaissance des formes : méthodes et applications*. InterEditions, Paris.
- [19] Abdel Belaïd. Analyse de document : de l'image à la représentation par les normes de codage. 01 1997.
- [20] Badr Al-Badr and Sabri Mahmoud. Mahmoud, s.a. : Survey and bibliography of arabic optical text recognition. signal process. 41(1), 49-77. *Signal Processing*, 41 :49–77, 01 1995.

- [21] R.G. Casey and E. Lecolinet. A survey of methods and strategies in character segmentation ». *IEEE Transactions on pattern analysis and machine intelligence*, 18(7) :690–7.
- [22] Frédéric Grandidier. *Un nouvel algorithme de sélection de caractéristiques - Application à la lecture automatique de l'écriture manuscrite*. PhD thesis, 01 2003.
- [23] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05*, pages 886–893,, San Diego, USA.
- [24] Website. <https://www.learnopencv.com/histogram-of-oriented-gradients/>, accessed 14/09/2020.
- [25] Adrian Bilski. A review of artificial intelligence algorithms in document classification. *International Journal of Electronics and Telecommunications*, 57 :263–270, 09 2011.
- [26] M. Milgram. *Reconnaissance des formes méthodes numériques et connexionnistes* ». Edition Armand colin, Paris.
- [27] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification* ». Second edition.
- [28] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *Signal Processing Magazine, IEEE*, 29 :141–142, 11 2012.
- [29] John C. Platt. Using analytic qp and sparseness to speed training of support vector machines". In *Advances in neural information processing systems*,(1999, page 557–563.

