

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



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 :

- **BERRIM Youcef**
- **MEBARKI Youcef**

### Thème

**Accélération d'extraction des  
lignes du texte à base des modèles  
de cross résolution à deux**

Soutenue le xx-xx- 2018 Devant le jury:

M.	Belila Khaoula	MCA	Président
M.	Gherbi kaddour	MAA	Rapporteur
M.	ZAIZ Faouzi	MAA	Encadreur

Année Universitaire: 2017-2018

# 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 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

En raison de l'évolution technologique rapide dans nos vies quotidiennes, Les systèmes OCR sont de plus en plus demandé/requis dans divers domaines. À ce jour, ces systèmes sont encore en développement, en particulier les méthodes de segmentation des lignes de texte. Ces dernières années, de nombreuses recherches et études ont été menées dans le but d'améliorer les méthodes de segmentation des lignes de texte en termes de vitesse et de taux de détection. Cependant, toutes ces méthodes utilisent souvent des images haute densité allant de 150 à 300 dpi, ce qui a un impact négatif sur la vitesse de ce processus.

Dans un travail précédent [21], une technique a été proposée pour accélérer le processus d'extraction des lignes de texte en utilisant des modèles de résolution croisée (XRM: Cross-Resolution Model), basés sur l'idée de réduction d'échelle d'image, afin de gagner du temps lors de l'exécution de ce processus. Notre travail est basé sur la même idée du travail précédent, qui est basée sur le fait que le processus de segmentation de texte en lignes ne nécessite pas une image haute résolution par rapport à d'autres processus tels que la segmentation des mots ou des lettres. Les lignes sont considérées comme une unité qui peut être redimensionnée sans affecter la précision contrairement aux unités plus petites comme les mots ou les lettres, qui si elles sont redimensionnées, affecteront négativement le processus de reconnaissance. Généralement, nous avons apporté d'importantes améliorations au travail précédent en améliorant le processus de redimensionnement qui est maintenant basé sur la taille du stylo et la hauteur des lignes au lieu de la taille du stylo uniquement. En outre, plus de méthodes de segmentation de lignes de texte ont été ajoutées, et leur précision a été améliorée.

Notre méthode permet de meilleurs résultats en termes de vitesse et de taux de détection, ce qui a été prouvé par les résultats expérimentaux obtenus en utilisant sept méthodes de segmentation. Chacune de ces méthodes a été testée sur deux bases de données, la première en arabe KHATT et la seconde en anglais IAM.

**Mots Clés:** OCR, Segmentation de texte en lignes, Redimensionnement, Accélérer.

# Abstract

As a result of the rapid technological advancement in our daily lives, OCR systems has become increasingly required in various fields. Till now, these systems are still under development, particularly text lines segmentation methods. Recently, many researches and studies have been conducted with the aim of improving text lines segmentation methods in terms of speed and detection rate. However, all of these methods often use high density images ranging from 150 to 300 dpi which negatively impacts the speed of this process.

In previous work [21], a technique was proposed to speed up the process of extracting text lines by using cross-resolution models (XRM: Cross-Resolution Model), which is based on the idea of image downscaling, in order to gain processing time during this process.

Our work is based on the same idea of the previous work, of which is that the process of segmentation does not require a high resolution image compared to other processes such as the segmentation of words or letters. Where The lines are considered as a unit which can be resized without affecting the accuracy unlike smaller units like words or letters which if resized will negatively affect the recognition process. In this work, we have generally made extensive improvements to the previous work by improving the resizing process which is now based on pen size and lines height instead of pen size only, allowing for more speed and increased detection rate. In addition, more text lines segmentation methods were added, and there accuracy has been improved.

Our method allows for better results in terms of speed and detection rate, and this has been proved by the experimental results obtained using seven text lines segmentation methods. Each of these methods was tested on two databases, the first is in Arabic KHATT and the second is in English IAM.

**Keywords:** OCR, Text lines segmentation, Resizing , Speed up.

# الملخص

نتيجة التطور التكنولوجي السريع في حياتنا اليومية أصبح التعرف الضوئي على الكلمات (OCR: Optic Character Recognition) مطلوباً أكثر في شتى المجالات. و إلى يومنا هذا مازالت هذه الأنظمة قيد التطوير و بشكل خاص أنظمة تجزئة اسطر النص. بحيث تم خلال السنوات الأخيرة إجراء العديد من البحوث و الدراسات بهدف تحسين طرق استخراج خطوط النص من ناحية السرعة و معدل الكشف . لكن جميع هذه الطرق في الغالب تستخدم صور ذات كثافة عالية تتراوح ما بين 300 إلى 150 dpi و ذلك يؤثر سلباً من ناحية السرعة على هذه العملية.

في أعمال سابقة ، تم اقتراح تقنية تسمح بتسريع عملية استخراج الأسطر عن طريق نموذج الدقات المتعارضة الذي يقوم على فكرة تحجيم الصور و ذلك بهدف ربح الوقت أثناء تطبيق هذه العملية.

يقوم عملنا على نفس فكرة العمل السابق، و هي أن عملية تحديد مواقع الأسطر لا تتطلب دقة كبيرة مقارنة مع العمليات الأخرى مثل تجزئة الكلمات أو الحروف. بحيث تعتبر الأسطر كوحدة قابلة للتصغير بدون التأثير على الدقة على عكس وحدات أخرى أصغر مثل الكلمات أو الحروف التي إذا تم تصغيرها فستؤثر بشكل مباشر و سلبي على عملية التعرف.

في هذا العمل، قمنا بشكل عام بإجراء تحسينات شاملة على العمل السابق و ذلك بالتحسين في عملية التصغير عن طريق الاعتماد في التصغير في هذه المرة على كل من عرض الخط و طول الأسطر عوض عن عرض الخط فقط مما سمح بالتسريع أكثر و زيادة معدل الكشف. كذلك تم إضافة طرق أكثر للاستخراج الأسطر و التحسين فيها من حيث الدقة.

تسمح طريقتنا بالحصول على نتائج أفضل من ناحية السرعة و معدل الكشف، و قد تم إثبات هذا عن طريق النتائج التجريبية المتحصل عليها باستعمال سبع طرق لاستخراج الأسطر. تم تجربة كل من هذه الطرق على قاعدتي بيانات اثنتين عوض عن واحدة، الأولى خاصة باللغة العربية و أما الثانية فتخص اللغة الإنجليزية.

**كلمات مفتاحية:** التعرف على الكتابة OCR، تجزئة النص إلى اسطر، تحجيم، تسريع.

# Table des matières

<b>Dédicaces</b>	<b>i</b>
<b>Remerciements</b>	<b>ii</b>
<b>Résumé</b>	<b>iii</b>
<b>Liste des figures</b>	<b>ix</b>
<b>Liste des tableaux</b>	<b>xii</b>
<b>Liste des algorithmes</b>	<b>xiii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Reconnaissance de l'écriture manuscrite</b>	<b>3</b>
Introduction . . . . .	3
1.1 Problèmes liés à l'OCR . . . . .	3
1.2 Caractéristiques de l'écriture Arabe . . . . .	4
1.3 Stratégies de reconnaissance . . . . .	5
1.3.1 Stratégies holistiques . . . . .	5
1.3.2 Stratégies analytiques . . . . .	6
1.4 Architecture d'un OCR . . . . .	6
1.4.1 Mode d'acquisition . . . . .	6
1.4.2 Pré-traitements . . . . .	7
1.4.3 Segmentation . . . . .	10
1.4.4 Extraction des caractéristiques . . . . .	11
1.4.5 Classification . . . . .	12

1.5	Domaines d'applications OCR . . . . .	12
1.5.1	Industrie juridique . . . . .	12
1.5.2	Secteur bancaire . . . . .	12
1.5.3	Soins médicaux . . . . .	13
1.5.4	CAPTCHA (Completely Automatic Public Turing Test to Tell Computers and Humans) . . . . .	13
1.5.5	Reconnaissance automatique des plaques d'immatriculation . . . . .	13
	Conclusion . . . . .	13
<b>2</b>	<b>Segmentation et Extraction des lignes</b>	<b>14</b>
	Introduction . . . . .	14
2.1	Concepts de segmentation du texte . . . . .	14
2.2	Techniques de segmentation . . . . .	15
2.2.1	Segmentation à partir du squelette . . . . .	15
2.2.2	Segmentation à partir des histogrammes de projection verticales . . . . .	15
2.2.3	Segmentation basée sur les fenêtres glissantes . . . . .	16
2.3	Niveaux de segmentation . . . . .	16
2.3.1	Segmentation de page . . . . .	17
2.3.2	Segmentation en lignes . . . . .	17
2.3.3	Segmentation en mots . . . . .	18
2.3.4	Segmentation en caractères . . . . .	18
2.4	Extraction des lignes . . . . .	19
2.4.1	Caractéristiques des lignes de texte . . . . .	19
2.4.2	Méthodes d'extraction des lignes . . . . .	19
	Conclusion . . . . .	24
<b>3</b>	<b>Conception et Architecture</b>	<b>25</b>
	Introduction . . . . .	25
3.1	Architecture générale du système . . . . .	25
3.2	Extraction des caractéristiques du texte . . . . .	26
3.2.1	Taille de stylo . . . . .	28
3.2.2	Hauteur des lignes . . . . .	29
3.3	Choix de résolution cible . . . . .	30
3.4	Transformation d'échelle . . . . .	31

3.5	Extraction des lignes de texte . . . . .	32
3.5.1	Projection horizontale . . . . .	32
3.5.2	Projection partielle . . . . .	35
3.5.3	Projection par fenêtre glissante . . . . .	35
3.5.4	Projection partielle à deux fenêtre . . . . .	37
3.5.5	Méthodes Hybrides . . . . .	39
3.6	Mapping des coordonnées . . . . .	41
	Conclusion . . . . .	42
<b>4</b>	<b>Réalisation et validation</b>	<b>43</b>
	Introduction . . . . .	43
4.1	Choix du langage de programmation . . . . .	43
4.2	Bibliothèque utilisée . . . . .	44
4.3	Interface et Fenêtres . . . . .	44
4.3.1	Pré-traitement . . . . .	45
4.3.2	Extraction des lignes . . . . .	46
4.3.3	Mapping des coordonnées . . . . .	50
4.4	Résultats et tests . . . . .	50
4.4.1	KFUPM Handwritten Arabic Text Database . . . . .	50
4.4.2	IAM Handwriting Database . . . . .	53
4.5	Comparison des résultats . . . . .	56
	Conclusion . . . . .	58
	<b>Conclusion et perspectives</b>	<b>59</b>
	<b>Bibliographie</b>	<b>61</b>

# Liste des figures

<b>1</b>	<b>Reconnaissance de l'écriture manuscrite</b>	<b>3</b>
1.1	Exemple d'un texte arabe manuscrit [9]. . . . .	5
1.2	Les caractères arabes diffèrent par le nombre de points [22]. . . . .	5
1.3	Exemple d'un mot avec différentes formes [9]. . . . .	5
1.4	Les différentes formes de caractères selon la position dans le mot [22]. . .	5
1.5	Schéma général d'un système de reconnaissance de caractères[27]. . . . .	6
1.6	Effet de certaines opérations de prétraitement [24]. . . . .	7
1.7	Exemple de binarisation correcte et incorrecte [1]. . . . .	8
1.8	Exemple de squelette des mots [23]. . . . .	9
1.9	Ligne de base avant et après normalisation [9]. . . . .	9
1.10	Inclinaison des lettres par rapport à l'axe moyen du mot [11]. . . . .	9
1.11	Résultats de la normalisation [11]. . . . .	10
1.12	Exemple de localisation des lignes d'écriture [4]. . . . .	10
1.13	Histogramme de la segmentation des mots [16]. . . . .	11
1.14	Exemple d'une segmentation de mot [2]. . . . .	11
<b>2</b>	<b>Segmentation et Extraction des lignes</b>	<b>14</b>
2.1	Exemple du chevauchement de texte . . . . .	14
2.2	Exemple de squelettes de mots . . . . .	15
2.3	Exemple d'un histogramme de projection de mot . . . . .	15
2.4	Exemple d'utilisation de la fenêtre glissante sur un mot[6] . . . . .	16
2.5	Illustration de l'arborescence du processus de segmentation[15] . . . . .	16
2.6	Exemple de segmentation de page[7] . . . . .	17
2.7	Exemple de segmentation en lignes d'une structure textuelle : (a) le texte et le profil lissé de la projection horizontale avec en rouge la position de tous les maxima, (b) exclusion des fausses détections et détermination des bords de lignes[12]. . . . .	17

2.8	Exemple de segmentation en mots d'un document. Les rectangles bleus délimitent les mots extraits [12]. . . . .	18
2.9	Exemple de segmentation en caractères [12] . . . . .	18
2.10	Caractéristiques de la ligne [21] . . . . .	19
2.11	Les étapes de segmentation en lignes en utilisant une projection globale. (a) Image du document, (b) Profil de projection, (c) profil de projection lissé et (d) détection de pics et segmentation en lignes. . . . .	20
2.12	(a) Image du document découpée en 5 bandes verticales, (b) en bleu, le profil de projection de la troisième bande et en rouge le profil lissé correspondant, (c) détection des lignes de texte et d'espaces inter-lignes en se basant sur le profil, et (d) détection raffinée en utilisant un MMC. . . . .	21
2.13	de segmentation avec une technique Piece-Wise . . . . .	21
2.14	Exemple d'image de texte lissée . . . . .	22
2.15	Exemples de segmentation des caractères ligaturés [27] . . . . .	22
2.16	Des lignes de texte extraites d'une lettre de Georges Washington (réimprimée de Shi et Govindaraju [50], © [2004] IEEE). Les pixels d'avant-plan ont été étalés le long de la direction horizontale. . . . .	23
2.17	Quartiers angulaires et rectangulaires des unités ponctuelles et rectangulaires (à gauche). Quartier défini par un groupe d'unités (droit). Deux alignements A et B en conflit: une mesure de qualité choisira A et éliminera B (en bas à droite). . . . .	24
2.18	l'espacement des coLes lignes de texte extraites sur les registres de l'église (réimprimé de Feldbach [12] avec la permission de l'auteur). Composants ou une mesure de la rectitude de l'alignement. . . . .	24
<b>3</b>	<b>Conception et Architecture</b>	<b>25</b>
3.1	Architecture de système à base de résolution croisée. . . . .	26
3.2	Exemple de la diversité des poids de stylo. . . . .	27
3.3	Exemple de la diversité des hauteurs de ligne. . . . .	28
3.4	grille de pixels d'un caractères. . . . .	29
3.5	Processus de réduction d'échelle d'image [9]. . . . .	32
3.6	Segmentation basée sur la fenêtre glissante [26]. . . . .	36
3.7	Exemple de colonne de lignes max dans différents alignements de texte. . . . .	38
3.8	Segmentation par la méthode hybride [26]. . . . .	40
3.9	Processus de mapping des coordonnées à l'image original. . . . .	42
<b>4</b>	<b>Réalisation et validation</b>	<b>43</b>
4.1	Interface principale de l'application . . . . .	45
4.2	Résultats de la segmentation en utilisant le modèle de résolution croisée . . . . .	46

4.3	Extraction des lignes par projection horizontale . . . . .	46
4.4	Extraction de lignes par fenêtre glissante . . . . .	47
4.5	Extraction de lignes par fenêtre glissant hybride . . . . .	47
4.6	Extraction de lignes par fenêtre fixe . . . . .	48
4.7	Extraction de lignes par fenêtre fixe hybride . . . . .	48
4.8	Extraction de lignes par double fenêtre . . . . .	49
4.9	Extraction de lignes par double fenêtre hybride . . . . .	49
4.10	Taux de détection des lignes de base de données arabes. . . . .	51
4.11	Temps d'exécution de la méthode de projection. . . . .	51
4.12	Temps d'exécution de la méthode de fenêtre glissante. . . . .	52
4.13	Temps d'exécution de la fenêtre glissante et de la méthode à double fenêtre. . . . .	52
4.14	Temps d'exécution des méthodes hybrides. . . . .	53
4.15	Taux de détection des lignes de base de données anglais. . . . .	54
4.16	Temps d'exécution de la méthode de projection. . . . .	54
4.17	Temps d'exécution de la méthode de fenêtre glissante. . . . .	55
4.18	Temps d'exécution de la fenêtre glissante et de la méthode à double fenêtre. . . . .	55
4.19	Temps d'exécution des méthodes hybrides. . . . .	56
4.20	Moyens des résultats de la base de données arabe KHATT. . . . .	57
4.21	Moyens de résultats de la base de données anglais IAM. . . . .	57

# Liste des tableaux

<b>1</b>	<b>Reconnaissance de l'écriture manuscrite</b>	<b>3</b>
1.1	Comparaison brève entre les approches en-ligne et hors-ligne [3]. . . . .	7

# Liste des algorithmes

<b>3 Conception et Architecture</b>	<b>25</b>
3.1 Méthode de calcul de la taille de police . . . . .	29
3.2 Méthode de calcul de la hauteur des lignes . . . . .	30
3.3 Méthode de projection horizontale . . . . .	34
3.4 Méthode de projection partielle . . . . .	35
3.5 Méthode de fenêtre glissante . . . . .	37
3.6 Méthode de projection partielle à deux fenêtre . . . . .	39
3.7 Méthode hybride . . . . .	41

# Introduction

L'écriture est l'un des moyens de communication les plus importants entre les êtres humains depuis l'antiquité jusqu'à aujourd'hui en raison de son importance à travers l'histoire humaine, la découverte de l'écriture est l'une des plus grandes réalisations humaines tout au long de l'histoire, Les spécialistes ont commencé à classer les événements historiques en fonction de leur distance à ce moment articulé de la vie humain. L'écriture n'a pas reçu une telle attention à partir de rien, mais plutôt grâce à des grands services qu'elle a fournis à l'humanité.

La reconnaissance automatique de l'écriture occupe une région de recherche intensive dans le domaine de reconnaissance des formes. RC (reconnaissance de caractères) signifie transformer/traduire des images de texte à un texte en format numérique, en d'autres termes, cela représente un essai de simulation du processus de lecture humaine. Alors que la lecture représente une technique d'apprentissage fondamentale pour les êtres humains, et que cette tâche semble qu'elle est généralement facile pour lui, mais elle reste toujours une tâche très difficile pour l'appliquer par la machine, à cause de l'existence de nombreuses difficultés telles que la grande variabilité des styles et des formes manuscrites, l'incertitude de l'écriture humaine, l'inclinaison de l'écriture, la taille des mots, etc.

Aujourd'hui, une plusieurs recherches dans le domaine de la reconnaissance optique de caractères (OCR) de documents manuscrits dans une variété de langues différentes. Il existe beaucoup de données manuscrites à reconnaître. La segmentation est l'étape clé du processus OCR pour exploiter complètement ces données. la segmentation est le processus d'extraction des segments du texte du document qui sont utilisés dans le processus de reconnaissance des caractères. La segmentation de ligne est le processus de segmentation du texte du document en lignes séparables. Ensuite, la segmentation des mots et la segmentation des caractères sont effectuées.

La segmentation de l'écriture en général et l'extraction des texte en lignes en particulier restent jusqu'à aujourd'hui un thème de recherche ouvert car chaque personne a sa propre

manière d'écrire en fonction des habitudes d'écriture, la forme de la lettre, l'humeur, l'éducation et la santé de l'écrivain. C'est pourquoi il n'y a pas une méthode qui fonctionne pour tous les types de documents, puisque les lignes de texte manuscrites sont généralement asymétriques et ne sont pas clairement séparées, la segmentation de lignes de texte des documents manuscrits est toujours un problème difficile. La performance et la précision du système OCR sont fortement influencées par l'exactitude et la précision du processus d'extraction des lignes de texte.

Dans ce travail, nous présentons une contribution pour accélérer le processus d'extraction des lignes de d'une image de texte dans le cas des systèmes hors-lignes, par une méthode à base des modèles de cross résolution faible (XRM : Cross Résolution Models) à deux niveaux. Elle consiste à crée un modèle plus petit de l'image originale en se basant sur la taille du stylo et la hauteur approximée des lignes de texte, ce qui permet d'avoir une accélération dans les cycles de traitement. La méthode a été validée en utilisant des techniques de segmentation basées sur des histogrammes de projection, et sur deux bases de données: la première est la base arabe KHATT, et la seconde est la base anglaise IAM.

Le reste du travail est organisé comme suit:

Le premier chapitre présente les concepts généraux sur la reconnaissance des caractères manuscrits, et les différents aspects d'un OCR et ses problèmes ainsi que les différentes phases de processus de reconnaissance.

Le deuxième chapitre présente un état de l'art sur la segmentation de l'écriture manuscrite. Dans ce chapitre, nous présentons les différentes approches et stratégies de segmentation.

Le troisième chapitre contient le coeur de notre travail, il contient la conception de notre système proposée, on décrit en détail la méthode d'accélération proposée et les différentes techniques utilisées.

Le quatrième chapitre présente les résultats obtenus par notre méthode ainsi qu'une comparaison. De plus, nous allons voir comment fonctionne globalement notre système.

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

L'OCR est une technologie complexe qui transforme les images contenant du texte en du texte en format numérique modifiable. Il permet de traiter des livres numérisés, captures d'écran et des photos avec du texte, et d'obtenir des documents modifiables tels que des fichiers TXT, DOC ou PDF. Cette technologie est largement utilisée dans plusieurs domaines. Les systèmes OCR les plus avancés peuvent gérer presque tous les types d'images, même les plus complexes comme les pages de magazines numérisées avec des images ou des photos d'un téléphone mobile.

Dans ce chapitre, nous allons présenter une vue générale sur l'OCR, les différentes approches, les méthodes ainsi que les techniques réalisées dans le domaine de reconnaissance du manuscrit.

## 1.1 Problèmes liés à l'OCR

La tâche de l'OCR n'est pas facile, divers problèmes compliquent le processus de reconnaissance, parmi lesquels on peut citer [24]:

- La qualité du document : un document télécopié ou photocopie plusieurs fois est plus difficile à traiter que la copie originale. L'écriture peut devenir plus mince ou au contraire plus épaisse, dégradée avec des parties du texte qui manquent ou de tâches qui apparaissent, des ouvertures ou des bouchages de boucles,
- L'impression : un document composé est de meilleure qualité qu'un document dactylographié qui, à son tour, est plus clair qu'un texte issu d'une imprimante matricielle. Une imprimante à jet d'encre peut introduire des tâches d'encre et un étalement des caractères, une imprimante laser peut générer des lignes ou des fonds,

- La discrimination de la forme : selon le style de la fonte utilisée, son corps et sa graisse. . . , le caractère change de graphisme. Le nombre de formes est d'autant plus important que le nombre de styles d'écriture est élevé. De plus, plusieurs caractères présentent une forte ressemblance tels que : pour l'arabe : haa et dal ,sad et ra,meem et waw - pour le Latin : U et V , O et 0, S et 5, Z et 2 .
- L'acquisition : la numérisation en temps réel introduit souvent des distorsions dans l'image. Dans le cas hors-ligne la qualité du texte numérisé est un compromis entre les variations de la position (inclinaison, translation, rétrécissement. . . ), la propreté de la vitre du dispositif de numérisation et sa résolution.
- Les variations des dimensions : un « pitch » de 10, 12 ou de 16 . . . (10, 12 ou 16 cpi (character per inch)). Un pitch de 10 implique des caractères plus grands aussi bien en largeur qu'en hauteur que ceux d'un pitch de 12.

En plus de ces problèmes un système OCR devrait être capable de distinguer entre un texte et une figure, de reconnaître les caractères ligaturés et d'être indépendant des variations de l'espace aussi bien inter-mots que de l'interligne. Les problèmes posés par la reconnaissance optique de l'écriture manuscrite, sont plus complexes que ceux liés à l'écriture imprimée. Les erreurs de lecture dans le cas du manuscrit sont dues aux variations infinies de l'écriture de nature aléatoire qui dépendent de facteurs particuliers du scripteur et des conditions de l'écriture.

## 1.2 Caractéristiques de l'écriture Arabe

La reconnaissance de l'écriture arabe est une tâche plus difficile à cause de l'apparence similaire de certains caractères. Le choix du type et de la méthode d'extraction des caractéristiques reste l'étape importante pour atteindre une précision de reconnaissance élevée. Voici certaines caractéristiques [9]:

- la langue arabe contient 28 caractères.
- L'écriture arabe n'a pas le concept des lettres majuscules et minuscules.
- Les mots arabes se composent d'une ou plusieurs parties, chacun a un ou plusieurs caractères.
- Il y a toujours un espace entre deux mots. Cependant un espace dans un mot peut exister si une lettre non connectée est utilisée.
- le script arabe est écrit de droite à gauche d'une manière cursive.
- Beaucoup de caractères arabes sont différent seulement dans la position ou le nombre de points attachés .

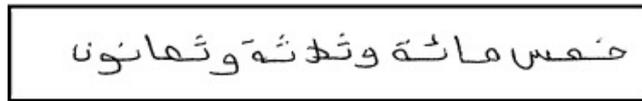


Figure 1.1: Exemple d'un texte arabe manuscrit [9].

"Yaa"	"Thaa"	"Taa"	"Baa"
ي	ث	ت	ب

Figure 1.2: Les caractères arabes diffèrent par le nombre de points [22].

- Les mots peuvent être écrits en utilisant différents styles d'écriture qui rendent les lettres et les mots ont des formes différentes qui causent une ambiguïté dans de nombreux systèmes de reconnaissance.

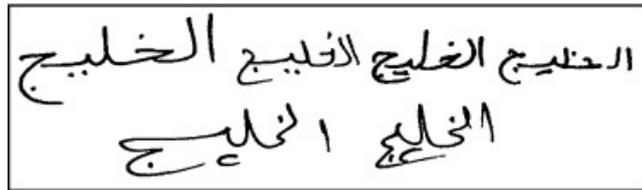


Figure 1.3: Exemple d'un mot avec différentes formes [9].

- chaque caractère peut avoir 1 à 4 formes différentes: isolées, connectées à partir de la gauche, connectées à droite et connectées à droite et à gauche :

Character	Isolated	End	Middle	Start
"Baa"	ب	ب	ب	ب
"Raa"	ر	ر	NA	NA

Figure 1.4: Les différentes formes de caractères selon la position dans le mot [22].

### 1.3 Stratégies de reconnaissance

Il existe deux approches de reconnaissance appliquées à la reconnaissance de caractères imprimés et manuscrits. Ceux-ci peuvent être classés comme suit:

#### 1.3.1 Stratégies holistiques

La reconnaissance est effectuée sur l'ensemble de la représentation des mots et lorsqu'il n'est pas nécessaire d'identifier les caractères individuellement. Cette approche présente l'avantage de garder le caractère dans son contexte avoisinant, ce qui permet une modélisation plus efficace des variations de l'écriture et des dégradations qu'elle peut subir. Cependant cette méthode est pénalisante par la taille mémoire, le temps de calcul et la complexité du traitement qui croissent linéairement avec la taille du lexique considéré, d'où une limitation du vocabulaire [11].

### 1.3.2 Stratégies analytiques

Contrairement à l'approche holistique (ou globale), les mots ne sont pas considérés comme un tout, mais comme des séquences d'unités de petite taille (caractères) et la reconnaissance n'est pas directement effectuée au niveau du mot mais à un niveau intermédiaire de ces unités [14].

## 1.4 Architecture d'un OCR

Tout système de reconnaissance de caractères passe généralement par les étapes suivantes: acquisition de données, prétraitement, segmentation, extraction de caractéristiques et enfin classification.

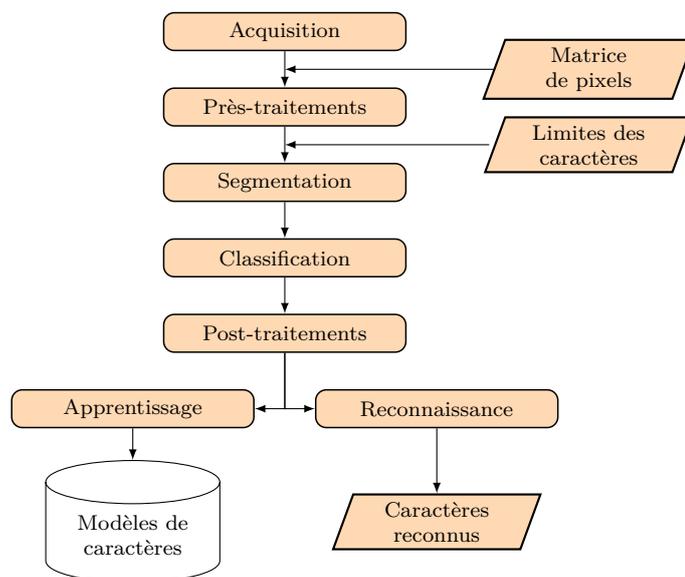


Figure 1.5: Schéma général d'un système de reconnaissance de caractères[27].

### 1.4.1 Mode d'acquisition

En général, Les méthodes de reconnaissance sont regroupées en deux catégories selon le mode d'acquisition de l'écriture, chacune avec des problèmes et des solutions différents [20]:

#### a. L'écriture en-ligne (Dynamique)

Dans la reconnaissance de caractères en ligne, les caractères sont reconnus en temps réel dès qu'ils sont écrits. Les systèmes en ligne fonctionnent mieux que la reconnaissance hors ligne car ils disposent d'informations de synchronisation et évitent l'étape de recherche initiale de localisation du caractère. Les systèmes en ligne obtiennent la position du stylo en fonction du temps directement à partir de l'interface. Ceci est généralement effectué à l'aide d'interfaces à base de stylo où l'écrivain écrit avec un stylo spécial sur une tablette électronique.

## b. L'écriture hors-ligne (Statique)

La reconnaissance de caractères hors ligne peut être classée en reconnaissance de caractères imprimés et manuscrits. Dans la reconnaissance de caractères hors ligne, le caractère imprimés ou manuscrit est typiquement scanné sous la forme d'un document papier et rendu disponible sous forme d'image binaire ou en niveaux de gris à l'algorithme de reconnaissance. En générale, la reconnaissance des caractères hors ligne est considérée comme une tâche plus difficile que sa contrepartie en ligne. En particulier La reconnaissance des caractères manuscrits est plus complexe que la reconnaissance des caractères imprimés, car l'écriture des personnes différer selon les attributs, tels que la forme, la taille, le style, etc.

Tableau 1.1: Comparaison brève entre les approches en-ligne et hors-ligne [3].

Critère de comparaison	En-ligne	Hors-ligne
Outils d'acquisition	- Stylo électronique plus tablette graphique.	- Scanner ou caméra.
Bruit d'image	- Faible	- existence d'un bruit important.
Informations disponibles	- la position, - la direction du mouvement, - les points de fin, - les points de début, - ordre des traits.	- absence d'informations contextuelles.

### 1.4.2 Pré-traitements

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

Parmi les opérations de prétraitement généralement utilisées on peut citer : l'extraction des composantes connexes, le redressement de l'écriture, le lissage, la normalisation et la squelettisation.

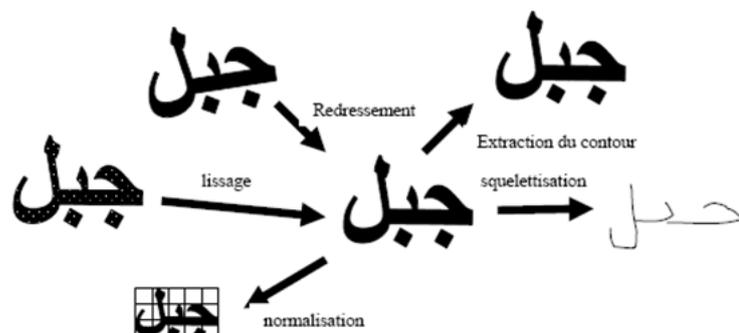


Figure 1.6: Effet de certaines opérations de prétraitement [24].

## Le Filtrage et lissage

Le filtrage est une opération de voisinage, la valeur d'un pixel est remplacée par la valeur d'une fonction appliquée à ce pixel et à ses voisins. Dans le domaine de reconnaissance de l'écriture, plusieurs types de filtrage peuvent être considérés selon le type de bruit rencontré ou le traitement désiré [24].

le lissage est une transformation locale qui ne tient pas 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].

## 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 [10]. Cette étape est très importante, car une binarisation incorrecte pose beaucoup de problèmes. Ceci est illustré ci-dessous :

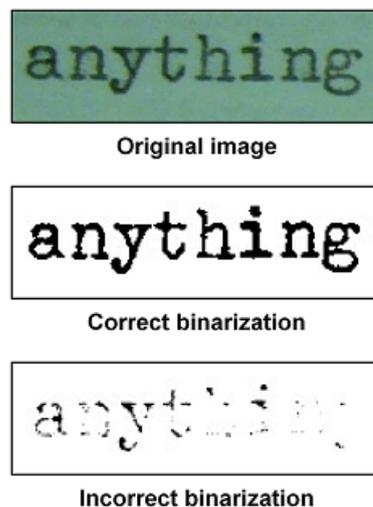


Figure 1.7: Exemple de binarisation correcte et incorrecte [1].

## La squelettisation

La squelettisation simplifie la forme du texte et réduit la quantité de données à traiter et permet d'obtenir le squelette de caractère connecté de l'image. L'épaisseur de toute ligne dans le texte après la squelettisation est égale à un pixel de large [11].

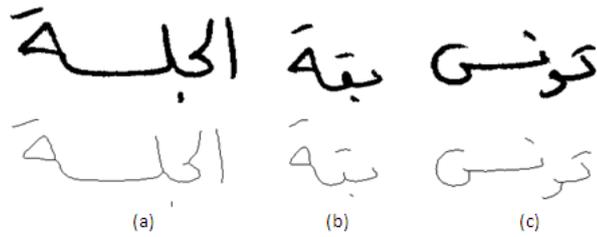


Figure 1.8: Exemple de squelette des mots [23].

### La normalisation de ligne de base

Un mauvais positionnement du document sur le scanner, conduit une inclinaison de l'image. Pour une bonne exploitation des documents, il faut donc détecter l'angle d'inclinaison et redresser les images, de telle sorte que les lignes de texte soient parallèles aux bords. L'angle du document (ou de la ligne),  $\theta$ , est celui qui correspond à l'histogramme d'entropie maximale. pour corriger cette inclinaison, il suffit d'appliquer une rotation de l'image d'angle  $\theta$  [11]:

$$\begin{cases} x' = x \cos \theta + y \sin \theta \\ y' = -y \sin \theta + x \cos \theta \end{cases}$$

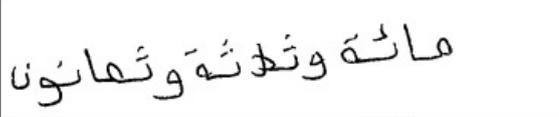
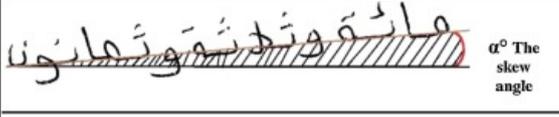
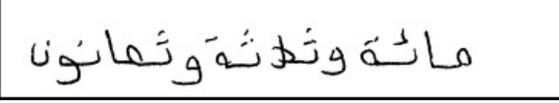
	Original Text Line
	Slope detection $\alpha^\circ$ The skew angle
	Text Line Corrected

Figure 1.9: Ligne de base avant et après normalisation [9].

### Redressement des écritures penchées

Cette technique facilite la segmentation préalable des mots en caractères. L'idée est de trouver l'angle moyen d'inclinaison puis de faire une transformation géométrique de type cisaillement de l'image.

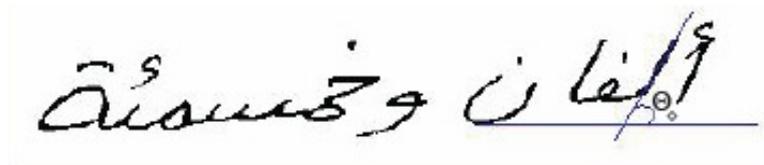


Figure 1.10: Inclinaison des lettres par rapport à l'axe moyen du mot [11].

## Normalisation de la taille

La taille des caractères peut varier d'une fonte à l'autre et même au sein d'une même fonte après agrandissement ou réduction. La normalisation de la taille permet de ramener les images de mots à la même taille.



Figure 1.11: Résultats de la normalisation [11].

### 1.4.3 Segmentation

La séparation des lignes, des mots, des pseudo-mots, des caractères constituent l'opération de segmentation de l'écriture manuscrite.

#### a. Segmentation d'un bloc de texte en lignes

Le but de cette étape est la séparation des différentes lignes du texte en s'appuyant souvent sur les histogrammes de projection horizontale. Cependant, le chevauchement et l'interférence des ascendants et des descendants des lignes voisines rendent difficile l'opération de segmentation.

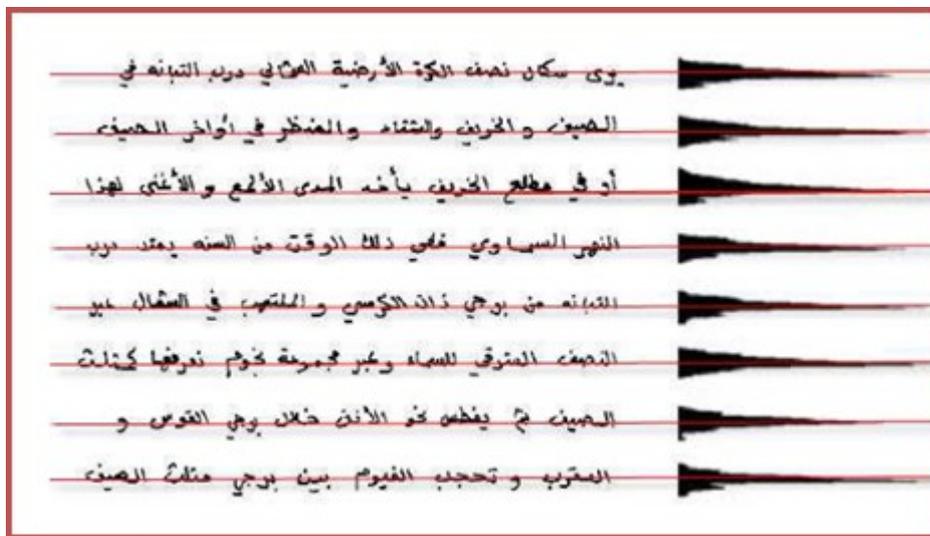


Figure 1.12: Exemple de localisation des lignes d'écriture [4].

#### b. Segmentation des lignes en mots ou en sous-mots (pseudo-mots)

elle se fait par la détection des espaces entre les mots et les pseudo-mots en utilisant les projections verticales.

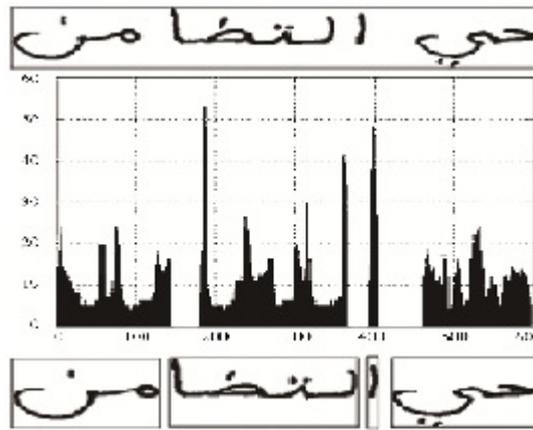


Figure 1.13: Histogramme de la segmentation des mots [16].

### c. Segmentation des mots en caractères ou graphème

La segmentation en caractères (ou en parties de caractères), constitue le problème le plus délicat de la reconnaissance de l'écriture.

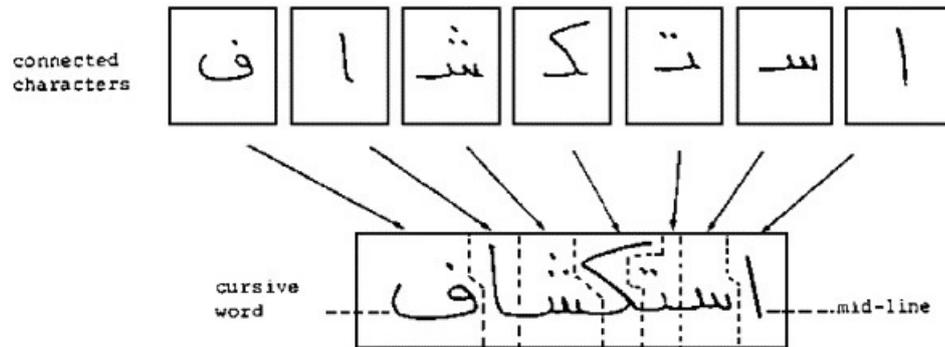


Figure 1.14: Exemple d'une segmentation de mot [2].

#### 1.4.4 Extraction des caractéristiques

Cette étape de la reconnaissance consiste à extraire des caractéristiques permettant de décrire de façon non équivoque les formes appartenant à une même classe de caractères tout en les différenciant des autres classes [11].

Les caractéristiques de caractère sont appelées primitives, sont classées en deux catégories [18] : les primitives structurelles (ou primitives locales) basées sur une représentation linéaire du caractère, la décomposition du caractère en segments de droites et courbes, contours du caractère, squelette) et les primitives statistiques (ou primitives globales) basées sur la distribution des points, les transformations et les mesures physiques faites sur le caractère (surface, moments, projections). Ces deux types peuvent être combinés formant un vecteur de primitives.

### **1.4.5 Classification**

L'étape de classification se fait en deux opérations ; l'apprentissage et la reconnaissance. L'apprentissage permet de construire un dictionnaire de prototypes. Il s'agit de regrouper plusieurs prototypes dont les primitives se rapprochent en classes.

Le processus de reconnaissance peut toujours se résumer à une décision de classification. Les réseaux de neurones constituent l'outil le plus utilisé dans les systèmes de reconnaissance grâce à leur capacité d'apprentissage. A partir de l'étape d'apprentissage, le système doit ajuster ses paramètres afin de donner une réponse lors de la phase de reconnaissance.

En plus de ces quatre phases, on pourra trouver une phase de post-traitement qu'est une opération facultative et dont le rôle consiste à vérifier et éventuellement à corriger l'hypothèse de reconnaissance générée par le module de classification [11].

## **1.5 Domaines d'applications OCR**

La reconnaissance optique des caractères a été appliquée dans l'ensemble des industries, ce qui a révolutionné le processus de gestion des documents. L'OCR est un vaste domaine avec un grand nombre d'applications tels que le Captcha, les dépôts institutionnels et les bibliothèques numériques, la reconnaissance optique musicale sans correction humaine et ni effort humain. Les industries. Certaines de ses applications sont expliquées ci-dessous [8]:

### **1.5.1 Industrie juridique**

Dans le secteur juridique, Afin d'économiser de l'espace et d'éliminer le besoin de passer par une grande quantité de boîtes de dossiers, les documents sont numérisés et saisis dans des bases de données informatiques. OCR simplifie encore le processus en rendant les documents consultables en texte, de sorte qu'ils sont plus faciles à localiser et à utiliser une fois qu'ils sont dans la base de données.

### **1.5.2 Secteur bancaire**

Une application largement connue de l'OCR est dans le secteur bancaire, où OCR est utilisé pour traiter les chèques bancaires sans intervention humaine. Un chèque bancaire peut être insérée dans une machine, l'écriture est scannée instantanément et le montant correct est transféré. Cette opération réduit le temps d'attente dans de nombreuses banques.

### 1.5.3 Soins médicaux

Les professionnels de soins médicaux doivent toujours traiter de grands volumes de formulaires pour chaque patient. Les outils de traitement de formulaires qui sont fournis par OCR sont capables de récupérer des informations à partir de formulaires et mettre ces informations dans des bases de données, Cela permet d'enregistrer rapidement les données de chaque patient. Par conséquent, les fournisseurs de soins de santé peuvent se concentrer sur la prestation du meilleur service possible à chaque patient.

### 1.5.4 CAPTCHA (Completely Automatic Public Turing Test to Tell Computers and Humans)

En raison de l'évolution rapide de l'utilisation de l'internet dans tous les domaines, CAPTCHA fonctionne comme un outil efficace contre les pirates et les programmes malveillants. CAPTCHA est un programme qui permet de générer et classer des tests que l'humain peut passer mais pas les programmeurs actuels des ordinateurs. CAPTCHA génère une image composée de séries de lettres masquées par des techniques de distorsion d'image. Ce système peut être utilisé pour supprimer ce bruit et segmenter l'image pour rendre l'image exploitable pour les systèmes OCR.

### 1.5.5 Reconnaissance automatique des plaques d'immatriculation

La reconnaissance automatique des plaques d'immatriculation (Automatic number plate recognition ou ANPR) est utilisée comme technique de surveillance en utilisant la reconnaissance optique des caractères sur les images pour identifier les plaques d'immatriculation des véhicules. Le système ANPR capture d'abord l'image du véhicule. La région de plaque d'immatriculation de véhicule est extraite en utilisant la segmentation d'image. La technique de reconnaissance de caractères optiques est utilisée pour la reconnaissance de caractères. Les données résultent sont ensuite comparées aux enregistrements d'une base de données afin de fournir des informations spécifiques telles que le propriétaire du véhicule, le lieu d'immatriculation, l'adresse, etc.

## Conclusion

Dans ce chapitre, après avoir introduit quelques aspects de la reconnaissance automatique de l'écriture, nous avons présenté un aperçu des différentes techniques employées dans le domaine. Nous avons dressé les différentes étapes d'un système de reconnaissance de caractères et de mot à savoir les prétraitements, l'extraction des caractéristiques et les méthodes de classification , De plus, nous avons présentés quelques caractéristiques de la langue arabe et montrer les difficultés qu'il faut surmonter pour réaliser un OCR arabe.

Dans le deuxième chapitre, nous allons décrire les étapes de segmentation, et les méthodes d'extraction des lignes.

# Segmentation et Extraction des lignes

## Introduction

Le processus de segmentation consiste à diviser les éléments de document en unités primaires (lignes, mots et lettres), Le processus de segmentation du texte manuscrit est une phase nécessaire pour faire améliorer le développement des systèmes de reconnaissance automatique de l'écriture manuscrite.

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

### 2.1 Concepts de segmentation du texte

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

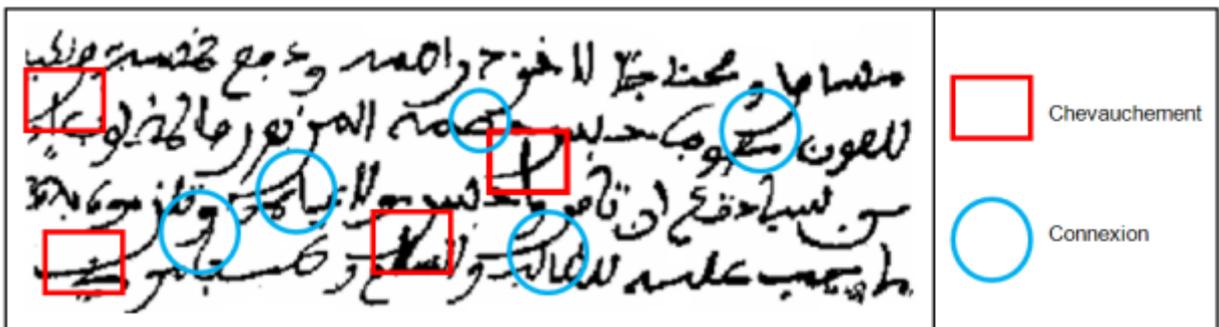


Figure 2.1: Exemple du chevauchement de texte

L'une des premières étapes de la conception d'un système de reconnaissance de l'écriture manuscrite consiste à découper le texte en lignes. Ce processus est relativement facile si le texte n'est pas incliné, 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 à écrire en arabe est la présence d'un grand nombre de points de diacritiques, 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.

## 2.2 Techniques de segmentation

### 2.2.1 Segmentation à partir du squelette

Dans ce cas, l'image du texte doit être triée par un squelette. Ensuite, à travers ce squelette, des points de connexion sont identifiés entre les caractères. 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[17].



Figure 2.2: Exemple de squelettes de mots

### 2.2.2 Segmentation à partir des histogrammes de projection verticales

Ce processus est fait en calculant le graphique de projection horizontale pour déterminer les différentes lignes de texte et le graphique de projection verticale afin de déterminer les différents caractères (89). Cette approche est proposée par le Professeur B. Yanikoglu et P. Sandon[38]



Figure 2.3: Exemple d'un histogramme de projection de mot

### 2.2.3 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 texte en fenêtres de taille fixe ou variable.



Figure 2.4: Exemple d'utilisation de la fenêtre glissante sur un mot[6]

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

## 2.3 Niveaux de segmentation

Pour détecter les informations (lettres, lignes ou mots) contenues dans l'image du texte, nous segmenter le document à l'aide de quatre étapes à savoir :

- Segmentation de page
- Segmentation en lignes
- Segmentation en mot et pseudo-mots
- et segmentation en caractères/graphèmes

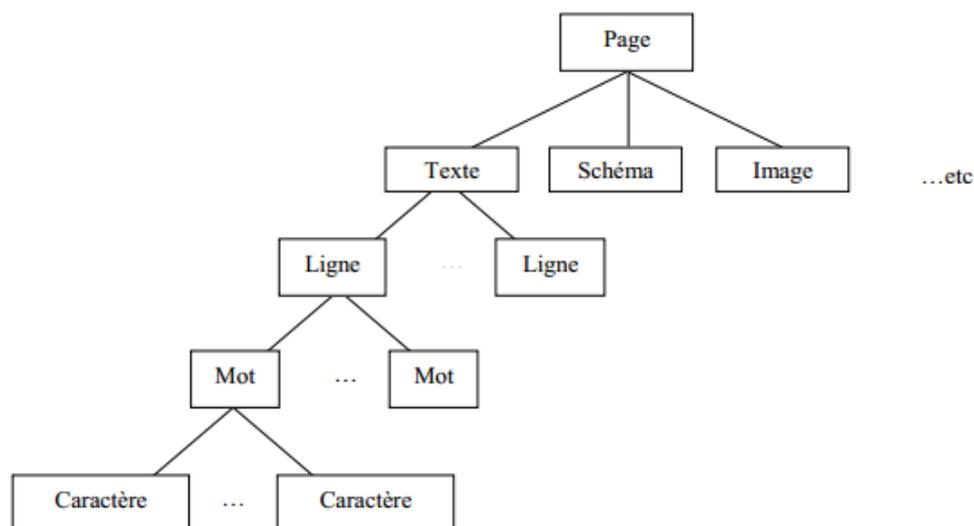


Figure 2.5: Illustration de l'arborescence du processus de segmentation[15]

### 2.3.1 Segmentation de page

Cette étape vous 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 (texte, image, dessin), ce qui permet également de diviser des régions de formes différentes et multiples[13].



Figure 2.6: Exemple de segmentation de page[7]

### 2.3.2 Segmentation en lignes

Cette segmentation consiste à examiner les structures linéaires textuelles et à affiner leur segmentation en exploitant l'information sur la hauteur des composantes connexes textuelles. L'idée consiste à utiliser la technique de projection pour segmenter toute structure linéaire susceptible de contenir plus qu'une ligne[12].

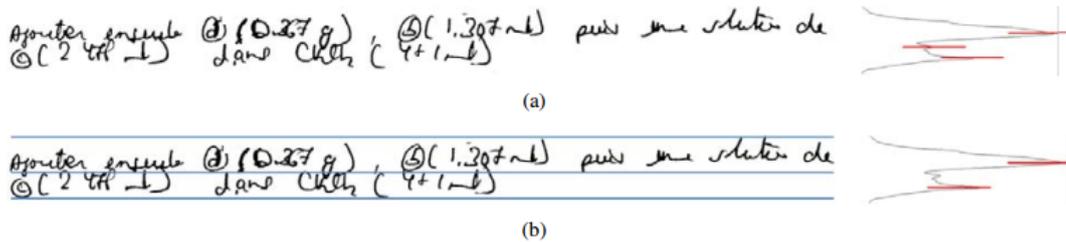


Figure 2.7: Exemple de segmentation en lignes d'une structure textuelle : (a) le texte et le profil lissé de la projection horizontale avec en rouge la position de tous les maxima, (b) exclusion des fausses détections et détermination des bords de lignes[12].

### 2.3.3 Segmentation en mots

La méthode de regroupement est utilisée en supposant que les espaces dans le mot sont plus petits que les espaces entre les mots imbriqués dans chaque ligne. Les composants associés sont ensuite extraits et ceux qui se chevauchent horizontalement et sont regroupés. Les distances horizontales entre les bords des boîtes liées sont ensuite déterminées pour les composants contigus contigus [12].

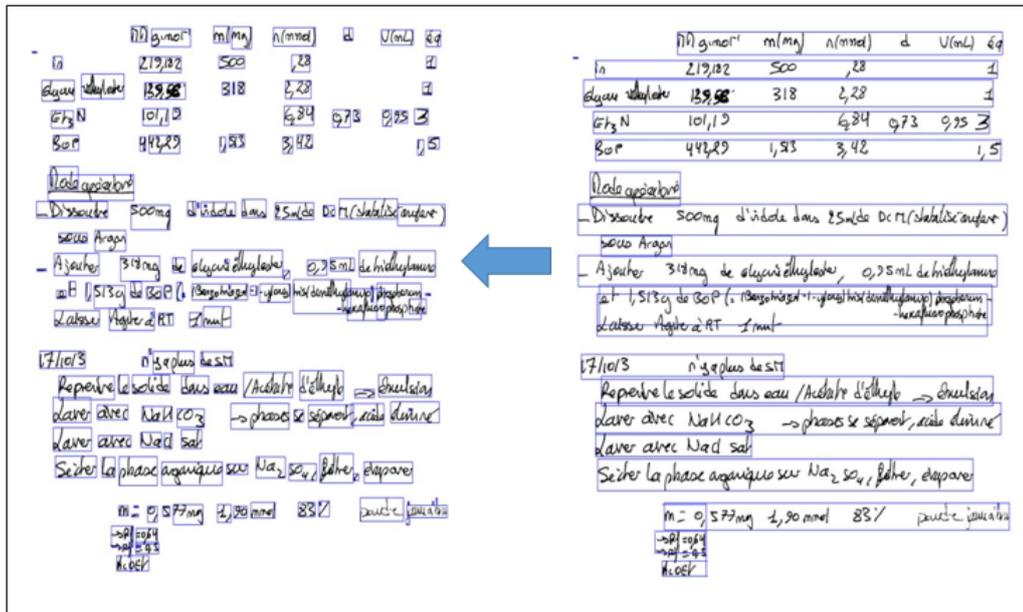


Figure 2.8: Exemple de segmentation en mots d'un document. Les rectangles bleus délimitent les mots extraits [12].

### 2.3.4 Segmentation en caractères

Les séparations de caractères sont effectuées en analysant l'image des caractères séquentiels et en la segmentation en sous-images de caractères individuels. C'est l'un des processus de décision dans un système de reconnaissance optique de caractères. Le but de ce système est de décider si l'élément isolé de l'image (lettre ou chiffre, etc.) est vrai ou non

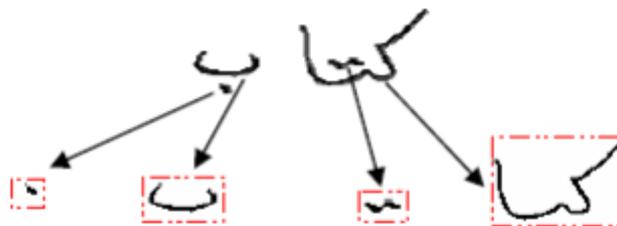


Figure 2.9: Exemple de segmentation en caractères [12]

## 2.4 Extraction des lignes

La première étape dans le traitement de texte manuscrit est d'identifier les lignes, ce qui est une étape préliminaire pour reconnaître les mots et les caractères plus tard. Ce processus est souvent facile lorsque les lignes sont inclinées ou se chevauchent.

### 2.4.1 Caractéristiques des lignes de texte

Il existe quatre types de lignes de référence : la ligne haute, la ligne de base supérieure, la ligne de base, la ligne de base inférieure.

Ces lignes de référence définissent les limites des zones importantes de la ligne de texte : la zone ascendante, la zone médiane et la zone descendante.

La ligne de base est la ligne dans laquelle les caractères sont proportionnés et n'ont pas une réduction significative dans laquelle les caractères minuscules [21] Ces lignes sont illustrées dans la figure suivante :

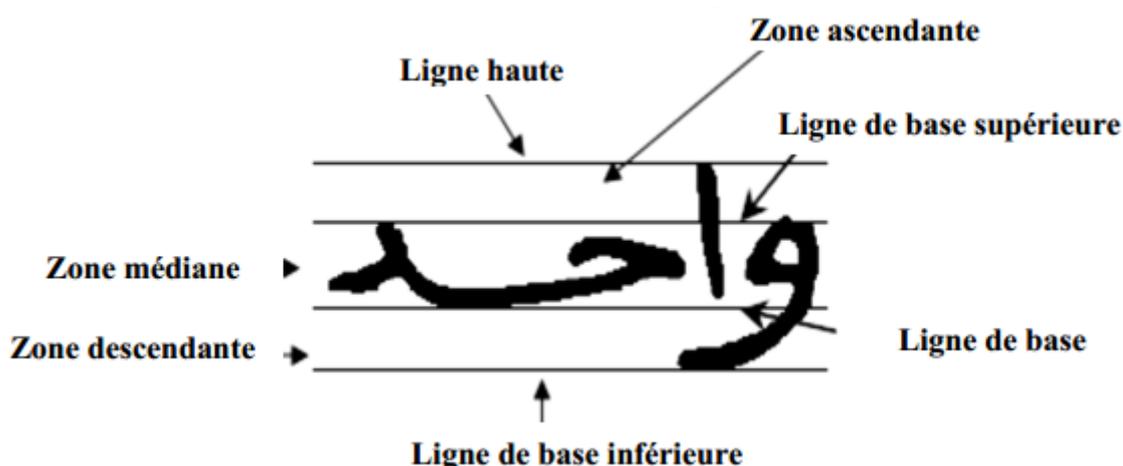


Figure 2.10: Caractéristiques de la ligne[21]

### 2.4.2 Méthodes d'extraction des lignes

L'extraction des lignes est indispensable dans les textes manuscrits qui en forment leurs seules structures. Cette extraction n'est pas facile car le texte est souvent incliné, peut comporter des ratures et des renvois. Elle est cependant indispensable pour rechercher des liges qui constituent les éléments fondateurs du texte.

#### 1. Methods descendants

Cette méthode commence par le niveau le plus élevé à savoir la page et descend d'un niveau à un autre jusqu'à arriver au niveau des composantes connexes ou au niveau pixel.

## a. Méthodes basées sur la projection

Cette méthode est utilisée pour extraire des lignes de texte en recherchant des pics de courbe en projection par la direction du texte dans le document. Généralement, dans le cas de la définition maximale, elle correspond à une ligne et dans le cas du minimum entourant les espaces inter-lignes. (en-dessus et en-dessous). Ainsi, les composantes connexes situées entre ces deux minima sont regroupées dans la même ligne. Selon le type de lignes, droites ou fluctuantes, la projection est effectuée globalement ou par morceaux. Par exemple, dans [Manmatha2005], les auteurs utilisent une projection globale, appliquée sur une image en niveaux de gris, pour extraire les lignes dans les manuscrits de Georges Washington. Le profil est lissé en utilisant un filtre passe-bas (gaussien) afin d'éliminer les faux pics et de réduire sa sensibilité au bruit. Enfin, les maxima locaux sont obtenus en déterminant les points où la dérivée du profil s'annule. Ces étapes sont illustrées dans la figure extraite de [Manmatha2005].

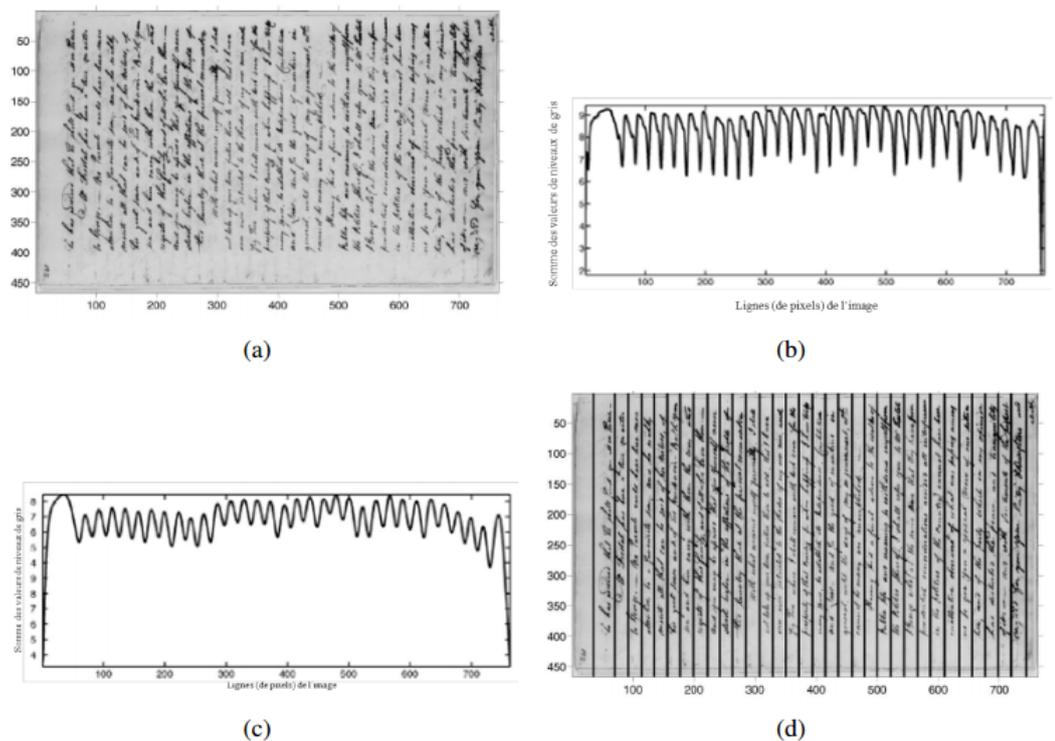


Figure 2.11: Les étapes de segmentation en lignes en utilisant une projection globale. (a) Image du document, (b) Profil de projection, (c) profil de projection lissé et (d) détection de pics et segmentation en lignes.

Dans le cas de l'écriture inversée et en italique, le document est divisé en bandes verticales de sorte que l'effet d'inclinaison est ignoré.

Lorsque le document contient des marges contenant un petit texte, elles sont ignorées si la densité de pixels est inférieure à un certain seuil.

Une autre méthode de projection [Marti2001] utilise les transitions noir-blanc au lieu des pixels. Pour chaque ligne de l'image, le nombre de transitions est compté et l'histogramme

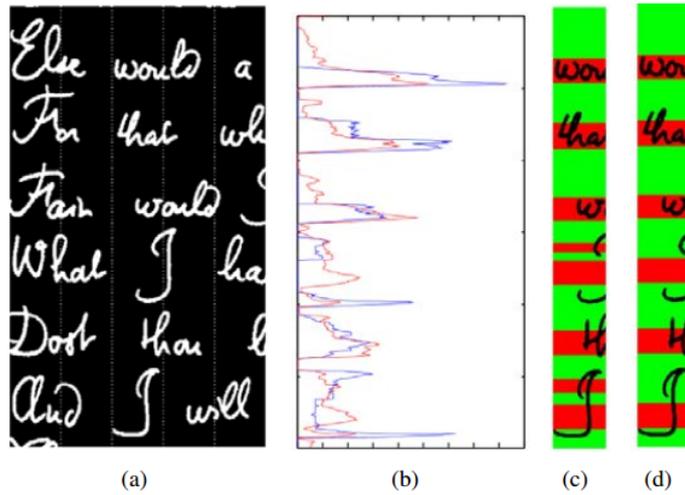


Figure 2.12: (a) Image du document découpée en 5 bandes verticales , (b) en bleu, le profil de projection de la troisième bande et en rouge le profil lissé correspondant, (c) détection des lignes de texte et d'espaces inter-lignes en se basant sur le profil, et (d) détection raffinée en utilisant un MMC.

correspondant est construit. Il est ensuite lissé en utilisant un filtre médian. Les bords (haut et bas) des lignes de texte correspondant aux minima de l'histogramme, sont déterminés. Selon la position de leur centre de gravité, les composantes connexes intersectées par les bords des lignes sont affectées à la ligne au-dessus ou en-dessous[12].

### b. Méthodes basées sur segmentation basée peinture Piece-Wise

Cette technique est proposée le but de segmenter en lignes des textes perses manuscrits. Le texte est divisé verticalement en segments parallèles. Chaque ligne de chaque bande est colorée par une intensité de gris, qui est la valeur d'intensité moyenne des valeurs de gris de tous les pixels présents dans cette bande de lignes. Par conséquent, les bandes colorées sont converties à une représentation lissées en deux couleurs. Les espaces blancs / noirs dans chaque bande de l'image complétée sont analysés pour une courte ligne de séparation PPSL (Piece-wise Potential Separating Line) entre deux espaces noirs consécutifs[27].



Figure 2.13: de segmentation avec une technique Piece-Wise

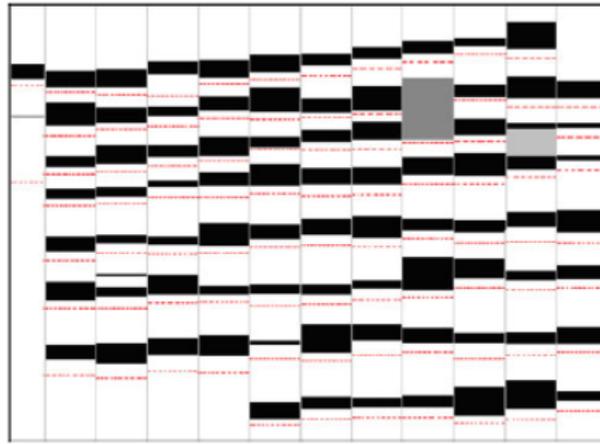


Figure 2.14: Exemple d'image de texte lissée

## 2. Méthodes ascendantes

Cette méthode commencent par le niveau le plus bas et remontent d'un niveau à un autre jusqu'à compléter la page.

### a. Méthodes basées sur Segmentation par écoulement d'eau

L'algorithme d'écoulement d'eau suppose que des débits d'eau hypothétique à partir de gauche à droite et de haut en bas. Dans le cas où l'eau s'écoule de gauche à droite, la situation est montrée par la figure (a). De plus, ce flux d'eau hypothétique devrait combler les écarts entre les lignes du texte consécutives. et les régions non mouillées restantes sur l'image indiquent les lignes du texte.

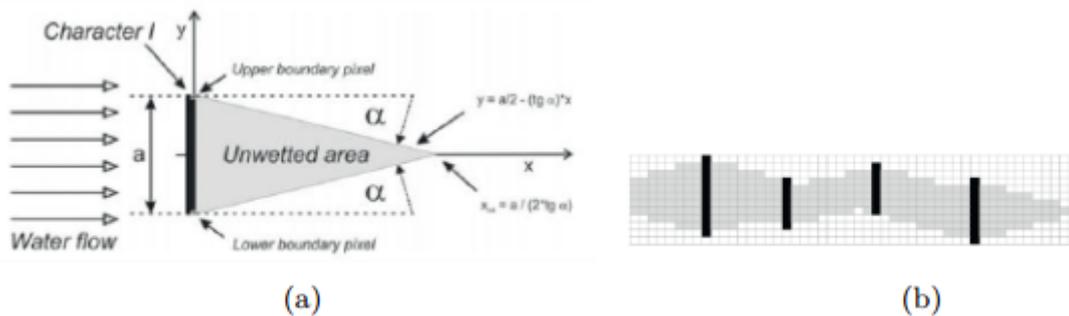


Figure 2.15: Exemples de segmentation des caractères ligaturés[27]

L'image va être divisée en deux types de bandes. Le premier contient les lignes du texte. Le second présente l'espacement entre les lignes. L'angle d'écoulement de l'eau hypothétique peut être obtenu en utilisant une fonction mathématique selon l'application. L'union des régions non mouillées est présentée par la figure (b)[27].

## b. Méthodes basées sur le regroupement

Ces méthodes consistent à construire des alignements en agrégeant des unités dans une stratégie ascendante. Les pixels peuvent être plus hauts ou plus élevés, tels que des blocs ou des blocs connectés. Les unités sont ensuite combinées pour former un alignement. Le schéma d'assemblage repose sur des critères locaux et globaux, utilisés pour vérifier la cohérence locale et globale, respectivement.

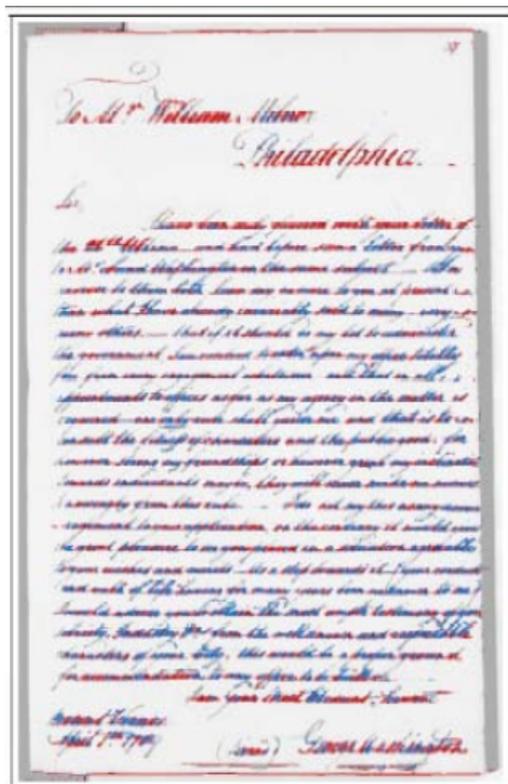


Figure 2.16: Des lignes de texte extraites d'une lettre de Georges Washington (réimprimée de Shi et Govindaraju [50], © [2004] IEEE). Les pixels d'avant-plan ont été étalés le long de la direction horizontale.

un simple schéma de jointure avec le plus proche voisin échoue souvent à regrouper des unités manuscrites complexes, car le voisin le plus proche appartient souvent à une autre ligne. Les critères d'assemblage utilisés dans les méthodes décrites ci-dessous sont adaptés au type des unités et aux caractéristiques des documents étudiés. Mais chaque méthode doit faire face à ce qui suit :

- initiation des alignements: une ou plusieurs graines pour chaque alignement.
- définir le voisinage d'une unité pour atteindre l'unité suivante. C'est généralement une zone rectangulaire ou angulaire (Fig. 7).
- résoudre les conflits. Comme une unité peut appartenir à plusieurs alignements en construction, un choix doit être fait: jeter un alignement ou garder les deux, en coupant l'unité en plusieurs parties..

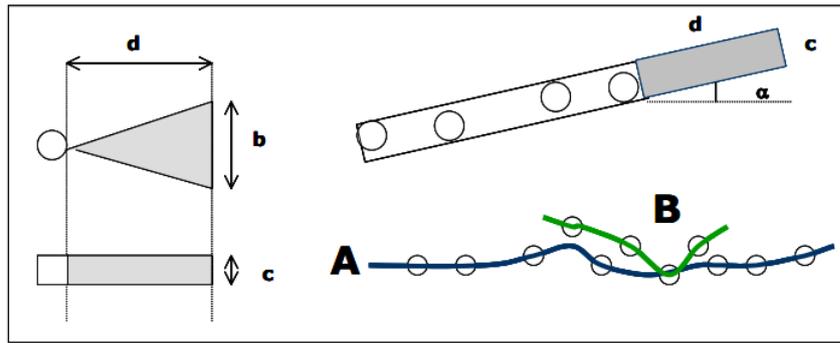


Figure 2.17: Quartiers angulaires et rectangulaires des unités ponctuelles et rectangulaires (à gauche). Quartier défini par un groupe d'unités (droit). Deux alignements A et B en conflit: une mesure de qualité choisira A et éliminera B (en bas à droite).

Les mesures de qualité comprennent généralement la force de l'alignement, c'est-à-dire le nombre d'unités incluses. D'autres éléments de qualité peuvent concerner la taille des composants[17].

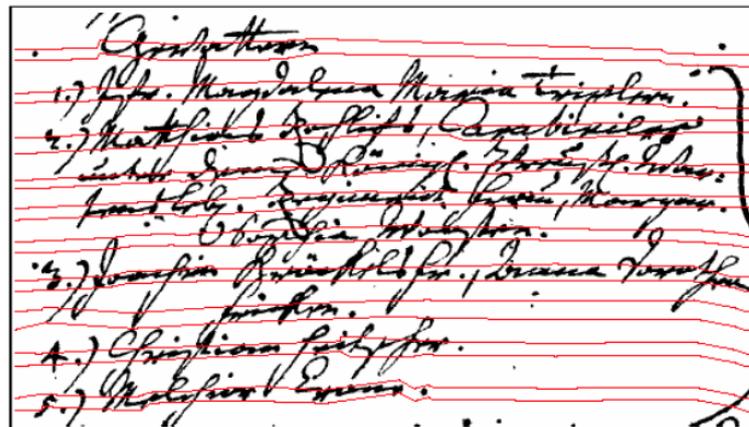


Figure 2.18: l'espacement des coLes lignes de texte extraites sur les registres de l'église (réimprimé de Feldbach [12] avec la permission de l'auteur).mposants ou une mesure de la rectitude de l'alignement.

## Conclusion

Dans ce chapitre, nous avons vu comment extraire des lettres, des mots et des lignes du document manuscrit. En plus, nous avons conclu que la méthode de séparation des composants du texte se présente par deux approches: approche descendante qui commence par la ligne et la divise en mots et caractères, et approche ascendante qui commence par des unités élémentaires et reconstruit le texte graduellement de bas en haut.

## Conception et Architecture

### Introduction

Dans les chapitres précédents, nous avons donné un aperçu des systèmes OCR et leur architecture générale dans laquelle nous avons expliqué les différentes phases des systèmes OCR. Par la suite, nous avons vu les différentes méthodes de segmentation des lignes et leurs types et techniques.

Dans ce chapitre, nous verrons l'architecture globale de notre système, ainsi que les étapes impliquées en détails. En plus, nous présentons les techniques et les approches utilisées pour valider notre méthode proposée pour l'accélération de l'extraction de lignes de texte à base des modèles de cross résolution XRM, qui est basée sur l'utilisation d'une résolution basse au lieu de la résolution d'origine qui est souvent de 150 à 300 dpi.

### 3.1 Architecture générale du système

L'idée principale du système consiste à appliquer les méthodes d'extraction des lignes sur une image à basse résolution (appelée XRM: Cross Résolution Model) au lieu l'image originale (qui est souvent d'une résolution de 150 ou de 300 dpi), dans le but de gagner du temps des cycles de traitement. Le choix du modèle XRM approprié est réalisée en se basant sur la taille du stylo et la hauteur approximée des lignes de texte. Après avoir faire l'extraction des lignes et obtenir les coordonnées des lignes de texte il est possible d'appliquer une fonction de mapping pour obtenir les coordonnées des lignes dans l'image originale.

D'abord, le système commence par l'acquisition de l'image (scanné ou imprimé) à traiter. Ensuite, prépare l'image aux traitements futur en appliquant des opérations de prétraitement (binarisation), cela génère une image binaire. Ensuite, le système calcule les caractéristiques du texte (la taille du stylo et la hauteur approximée des lignes de texte). En fonction de ces caractéristiques calculées, un facteur d'échelle est calculé (voir

la section 3.4), puis une nouvelle image de basse résolution est générée à partir de l'image à haute résolution originale en fonction. Après cela, une méthode de segmentation de ligne est appliquée sur l'image à basse résolution dans le but d'avoir les coordonnées des lignes de texte. Enfin, les résultats de la segmentation sont mappés sur l'image à haute résolution en utilisant une fonction de mapping (voir la section 3.6).

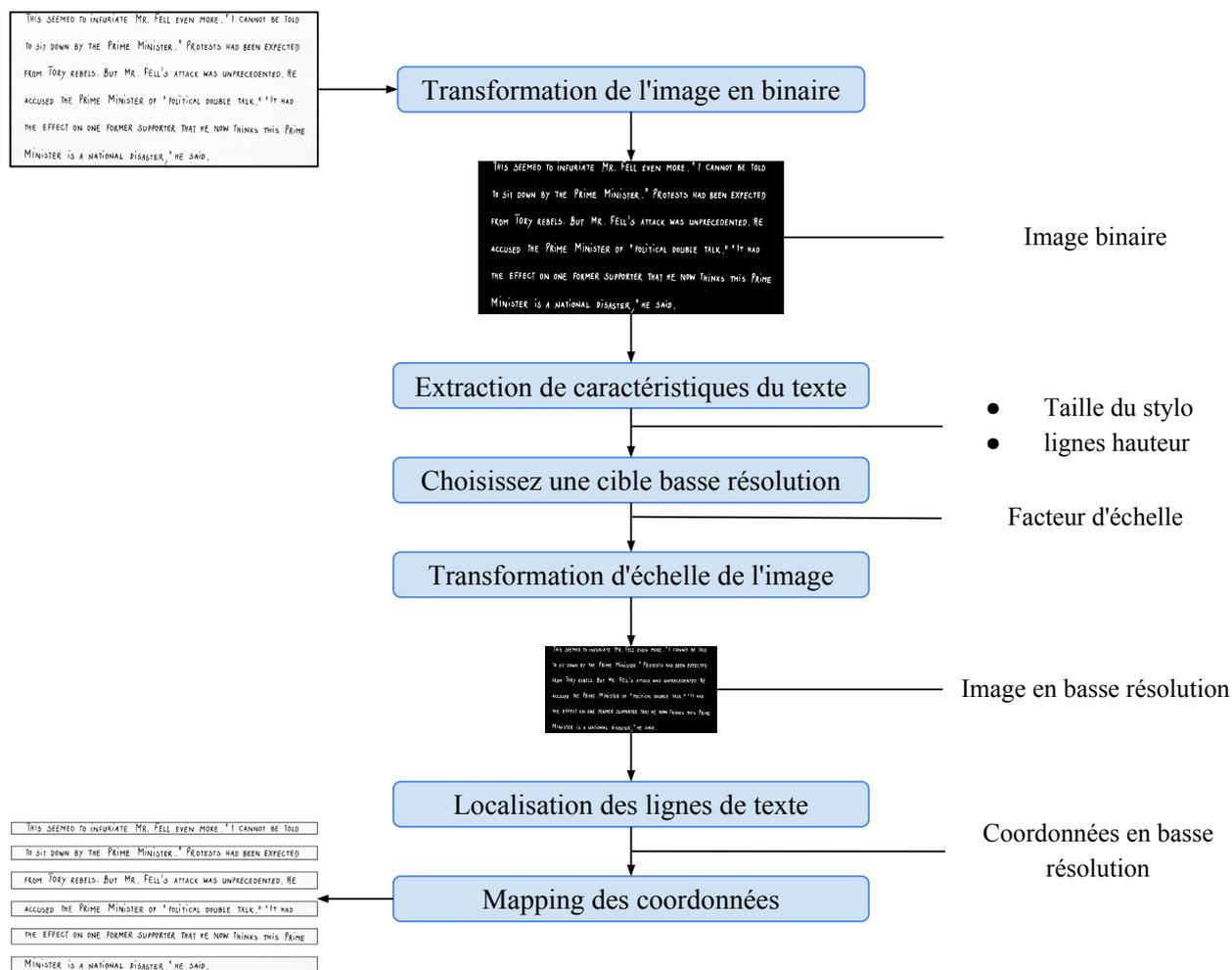


Figure 3.1: Architecture de système à base de résolution croisée.

## 3.2 Extraction des caractéristiques du texte

Cette phase, consiste à déterminer les caractéristiques selon lesquelles va pouvoir choisir la bonne résolution basse cible. Selon l'étude faite durant ce travail nous avons remarqué que deux facteurs principal peuvent influencés le choix de cette résolution, à savoir :

1. **la taille du stylo** : parce que chaque texte(imprimé ou manuscrit) est écrit avec des poids de police différents (voir la figure 3.2 ci-dessous ) et chaque poids de police doit être redimensionné différemment. par exemple, un petit poids de police ne doit pas être redimensionné trop de fois sinon le texte va disparaître. contrairement à un gros poids de police qui peut supporter/tolérer un facteur de redimensionnement

plus grand. alors que d'autre part, si nous redimensionnons seulement quelques fois, nous sacrifions les performances, C'est pour ça, nous devons estimer la taille et redimensionner en conséquence pour éviter tous les problèmes précédemment.

2. **la hauteur approximée des lignes de texte:** la hauteur des lignes est également quelque chose que nous devons considérer, car il diffère d'un document manuscrit à l'autre (voir la figure ci-dessous), surtout dans le cas d'écriture sans contrainte (voir la figure 3.3 ci-dessous). C'est pourquoi chaque hauteur des lignes doit être redimensionnée différemment. par exemple, Les lignes de texte avec une petite hauteur ne doivent pas être redimensionné trop de fois ou les lignes de texte seront fusionnées. Contrairement aux lignes de texte avec une grande hauteur qui peut supporter/tolérer un facteur de redimensionnement plus grand. De cette façon, nous devons également estimer cette hauteur et redimensionner en conséquence pour éviter tout les problèmes.

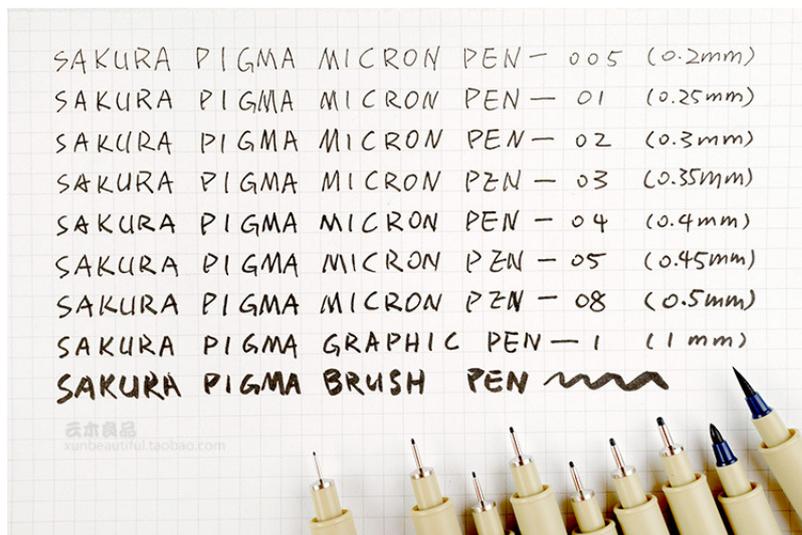


Figure 3.2: Exemple de la diversité des poids de stylo.

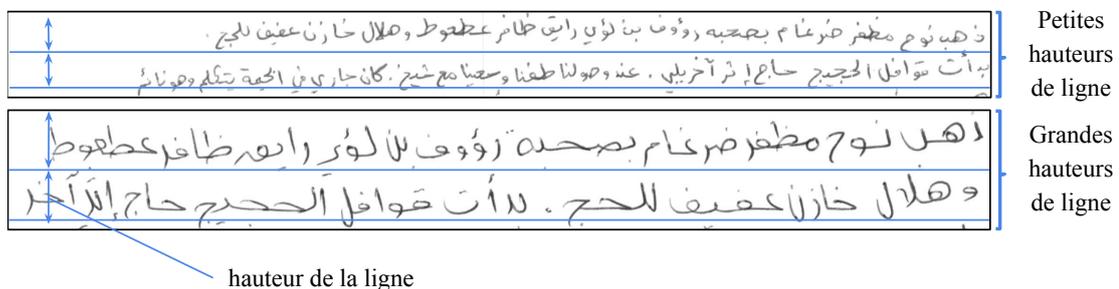


Figure 3.3: Exemple de la diversité des hauteurs de ligne.

Des problèmes de segmentation de ligne liés à la perte d'informations peuvent se produire en raison d'un redimensionnement incorrect, en particulier dans le cas de petites tailles de stylo ou de hauteur de ligne. le processus de segmentation échouera certainement dans cette situation en cas de redimensionnement manuel du document image avec une échelle fixe. C'est pourquoi nous avons opté pour une solution de redimensionnement automatique en exploitant les propriétés du texte (taille de la police, hauteur de la ligne) afin d'obtenir les meilleurs résultats possibles dans la segmentation des lignes dans les images de plus basse résolution avec un contenu limité.

### 3.2.1 Taille de stylo

La taille d'une police particulière est l'épaisseur des contours de caractères qui est simplement représentée par un ensemble de pixels. Donc, afin d'approximer cette taille. Premièrement nous appliquons un scan horizontal pour obtenir la largeur horizontale puis nous appliquons un autre scan vertical séparé pour obtenir la largeur verticale. dans chaque scan vertical ou horizontal, nous comptons le nombre de pixels noirs successifs. Ensuite, on calcule la taille moyenne et considérez-la comme la taille du stylo. La figure et l'algorithme ci-dessous illustrent cette étape.

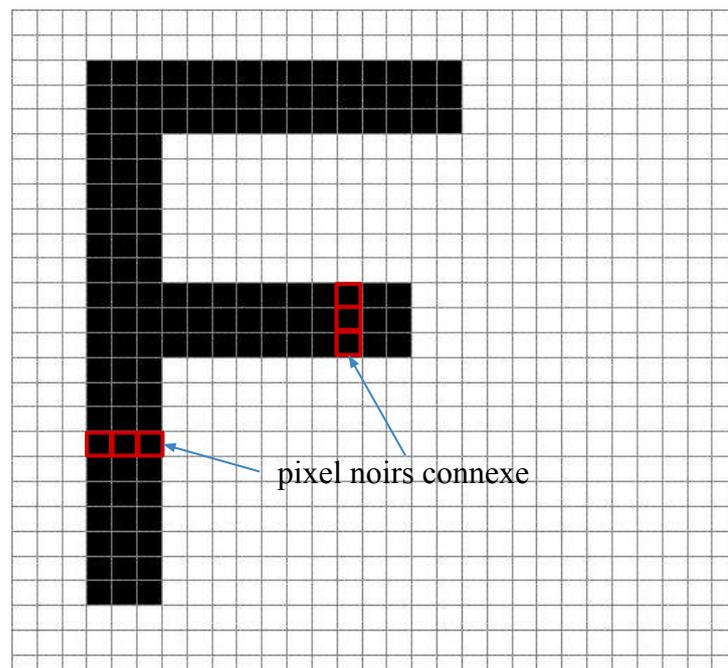


Figure 3.4: grille de pixels d'un caractères.

**Entrée:**

*img*: image binaire;  
*W, N, som*: entier;

**Début**

Partitionner la largeur de l'image en 05 parties;

**Pour** chaque pixel *p* de colonne du milieu **Faire**

**Si** *valeur(p)* > 0 **Alors**

    | *som*++;

**Sinon**

    | **Si** *som* > 0 **Alors** Charger la *som* dans la liste de somme **LS**;

    | *som* ← 0;

**FinSi**

**FinPour**

*tailleDuStylo* ← somme totale de LS / taille de LS;

**Fin**

---

### 3.2.2 Hauteur des lignes

Il est évident que dans n'importe quel texte la colonne avec le nombre maximum de lignes est la colonne la plus à gauche (en cas d'alignement de droite à gauche) ou la colonne la plus à droite (en cas d'alignement de gauche à droite). Par conséquent, ils sont le meilleur endroit pour obtenir la hauteur de la ligne. Mais nous pouvons pas connaître vraiment la colonne avec le nombre maximum de lignes jusqu'à diviser l'image en cinq colonnes, puis appliquer une projection partielle sur la première et la dernière colonne, puis on extrait la hauteur de la ligne du colonne avec le nombre maximal de lignes. Malheureusement, cette solution peut influencer sur les performances du processus de redimensionnement. Donc pour cette raison, afin d'approximer la hauteur de la ligne sans sacrifier la performance, nous avons divisé l'image en cinq colonnes, puis nous avons appliquer une seule projection partielle au lieu de deux sur la colonne du milieu, ensuite on extrait la hauteur des lignes. De cette façon, nous pouvons obtenir la hauteur des lignes indépendamment de la langue (voir l'algorithme 3.2 ci-dessous). Les principaux étapes de ce processus sont:

1. partitionner l'image en cinq colonnes.
2. appliquer une projection partielle sur la colonne du milieu.
3. extraire les lignes de la colonne (colonne du milieu).
4. Itérer chaque ligne dans la liste des lignes et calculer la hauteur en soustrayant la position de la ligne actuelle de la position de la ligne précédente.
5. calculer enfin la moyenne comme la hauteur des lignes.

**Entrée:**

*img*: image binaire;  
*W, N, som*: entier;

**Début**

Partitionner la largeur de l'image en 05 parties;  
Appliquer une projection partielle sur la colonne du milieu;  
Calculer le seuil puis extraire les lignes de la colonne du milieu;  
// itérer sur les lignes trouvées dans colonne du milieu

**Pour** *i* de 2 à *N - 1* **Faire**

*p1* ← coordonnées de la ligne *lignes*[*i*];  
*p2* ← coordonnées de la ligne *lignes*[*i - 1*];  
*h* ← *p1 - p2*;  
Charger *h* dans la liste des hauteurs **LH**;

**FinPour**

*lignesHauteur* ← somme totale de LH / taille de LH;

**Fin**

---

### 3.3 Choix de résolution cible

Le choix de la basse résolution cible (XRM: Cross Resolution Model) est une tâche très critique dans notre système qui a un impact direct sur les résultats de la segmentation. c'est pourquoi il doit être fait correctement pour éviter tout les problèmes.

Cette opération est faite automatiquement en utilisant une formule prédéfinis. Cette formule est composée des deux facteurs de base la taille du stylo et la hauteur approximée des lignes, afin de choisir la plus petite résolution possible en fonction des entrées tout en préservant en même temps les caractéristiques importantes du texte. Ces deux entrées sont calculées à partir de l'image à haute résolution après plusieurs testes expérimentaux:

$$S = \left( \frac{f_{min}}{f} \right) + \left( \frac{h_{min}}{h} \right) \quad (3.1)$$

Où:

- *S* : le facteur d'échelle.
- *f* : la taille du stylo.
- *h* : la hauteur des lignes.
- *f<sub>min</sub>* : la taille minimale du stylo qui est égale à 1.0 .
- *h<sub>min</sub>* : la hauteur minimale des lignes qui est égale à 6.0 .

- $f_{min}$  et  $h_{min}$  sont des constantes qui sont choisies après des expérimentation.

Après avoir effectué plusieurs expérimentations au cours du développement sur un ensemble d'environ 600 images de la base de données KHATT, nous avons trouvé que les meilleures valeurs qui permettent la meilleure minimisation possible de la taille de l'image pour la taille de stylo minimale est 1.0 et la hauteur de la ligne minimale est 6.0 .

Il convient de noter que ces tailles peuvent ne pas être réalistes, mais elles permis d'obtenir les meilleurs résultats en termes de minimisation.

### 3.4 Transformation d'échelle

Après avoir choisi l'échelle de la basse résolution cible,nous pouvons redimensionner l'image en utilisant cette résolution, et cela est fait dans notre cas en utilisant une API (Application Programming Interface) de bibliothèque tierce externe pour gagner du temps et faciliter ce processus.

Lors du redimensionnement d'une image bitmap (généralement représenté par une grille rectangulaire de pixels), une nouvelle image avec un nombre de pixels supérieur ou inférieur est générée. Dans le cas de la réduction d'échelle , cela se traduit généralement par une diminution du nombre de pixels accompagnée par une perte de la qualité des éléments visible (voir la figure 3.5).

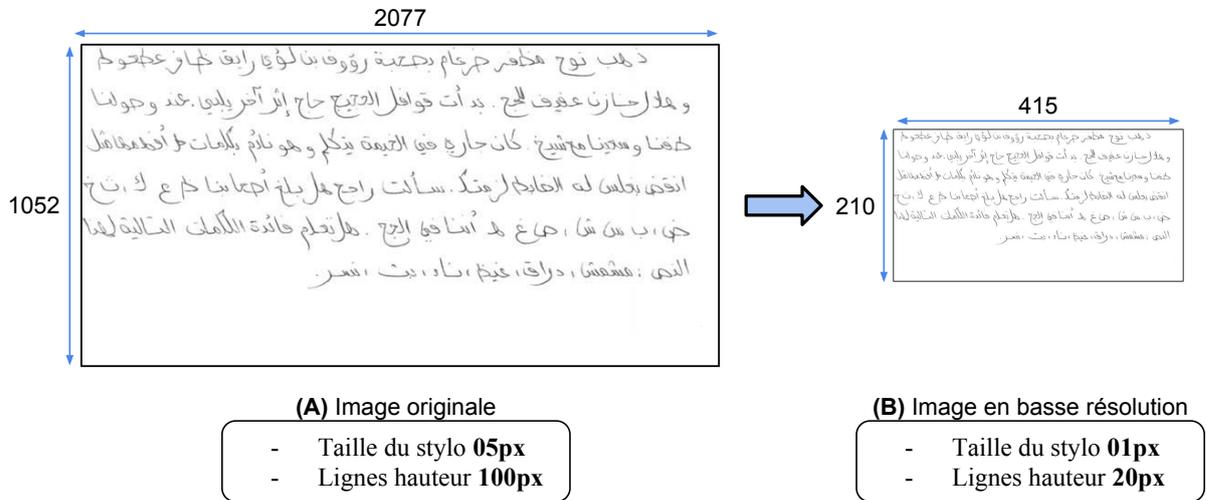


Figure 3.5: Processus de réduction d'échelle d'image [9].

## 3.5 Extraction des lignes de texte

A ce stade, nous appliquant l'une des sept méthodes qui sont toutes basées sur une technique de segmentation de la ligne descendante appelée profil de projection horizontale. il est considéré comme l'une des méthodes les plus rapides dans la segmentation de texte en lignes. Les sept méthodes mises en œuvre sont les suivantes:

- Méthode de projection horizontale .
- Méthode de fenêtre glissante .
- Méthode hybride(horizontale et par fenêtre glissante).
- Méthode de projection partielle.
- Méthode hybride (horizontale et par projection partielle).
- Méthode de projection partielle à deux fenêtre.
- Méthode hybride (horizontale et par projection partielle à deux fenêtre).

### 3.5.1 Projection horizontale

C'est l'approche la plus utilisée par les systèmes de reconnaissance, en raison de sa grande vitesse de récupération des résultats, mais elle est sensible aux chevauchements et fluctuations des lignes du texte. Elle consiste à faire une segmentation du texte en utilisant la projection horizontale (voir l'équation 3.2 ci-dessous).

$$f(y) = \sum_{x=0}^W I(x, y) \quad (3.2)$$

La méthode prend comme entrée l'image binaire (résultat de l'étape de prétraitement), puis, calcul une projection horizontale et utilise des seuils afin d'identifier les lignes de texte. Cette méthode extrait uniquement les lignes droites. Elle utilise les étapes suivantes (voir l'algorithme 3.3) :

1. Calculer la somme des pixels noires de chaque ligne de l'image .
2. Calculer le seuil, qui est égale à la moyenne des sommes totals .
3. Extraire toutes les lignes qui sont plus grandes que le seuil.
4. Tracer les lignes extraites.

**Entrée:**

*img*: image binaire;  
*W, H, seuil, som*: entier;  
*PL*: Tableau[H] d'entiers;

**Début**

*W* ← largeur de l'image;  
*H* ← hauteur de l'image;  
// calculer la somme pour chaque ligne de l'image  
**Pour** chaque ligne *L* de l'image **Faire**  
    *som* ← 0 ;  
    **Pour** chaque pixel *p* dans *L* **Faire**  
        **Si** *valeur(p)* > 0 **Alors** *som*++;  
    **FinPour**  
    Charger le somme dans la liste de projection **PL**;  
**FinPour**  
  
// calculer le seuil  
*seuil* ← somme totale de **PL** / taille de **PL**;  
// extraire la liste des lignes **LL**  
**Pour** chaque valeur *x* dans **PL** **Faire**  
    **Si** *x* > *seuil* **Alors**  
        Charger *x* dans une liste temporaire **LT**;  
    **Sinon**  
        **Si** **LT** taille > 0 **Alors**  
            Charger la liste temporaire **LT** dans la liste des lignes **LL**;  
            Effacer la liste **LT**;  
        **FinSi**  
    **FinSi**  
**FinPour**  
  
// localiser les coordonnées des lignes  
**Pour** chaque ligne *L* dans **LL** **Faire**  
    *positionDeLigne* ← Calculer le maximum de la liste **L**;  
    Charger la position de ligne dans la liste des coordonnées de ligne;  
**FinPour**

**Fin**

---

### 3.5.2 Projection partielle

Elle se base sur la projection partielle, et consiste à diviser l'image du texte en colonnes, et par la suite applique localement une projection horizontale dans chaque colonne. Elle ce résume par les étapes suivantes (voir l'algorithme 3.4):

1. Subdiviser l'image en plusieurs colonnes selon le choix, nous avons utilisé 05 colonnes.
2. Pour chaque colonne appliquer une projection horizontale.
3. Enfin les lignes sont assemblées en regroupant des segments de ligne similaires. la direction de l'assemblage des lignes est basée sur l'alignement du texte. Dans le cas de l'arabe, nous commençons par la dernière colonne de droite jusqu'à la première colonne de gauche. La même chose peut être dite pour les langues latines, sauf dans la direction opposée.

---

**Algorithme 3.4:** Méthode de projection partielle

---

**Entrée:**

*img*: image binaire;  
*W,N*: entier;

**Début**

*N* ← 05;  
*W* ← largeur de l'image;  
Partitionner la largeur de l'image en *N* parties;  
// appliquer une projection partielle sur chaque partie

**Pour** *chaque partie* **Faire**

    Calculer le somme de pixel pour cette partie;  
    Calculer le seuil locale;  
    Extraire la liste des lignes **LL** dans cette partie;

**FinPour**

**Fin**

---

### 3.5.3 Projection par fenêtre glissante

Elle consiste à subdiviser l'image du texte en colonnes par une fenêtre de taille fixe (comme peut être variable). Cette fenêtre est déplacée avec un pas souvent la moitié de la fenêtre. Ensuite, un diagramme de projection horizontale est calculé à chaque déplacement de la fenêtre dans le but de détecter les lignes du texte (voir la figure 3.6).

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. En plus, quand il y'a beaucoup de segments liés, on risque de joindre deux entités différentes. les étapes de la méthode sont les suivantes (voir l'algorithme 3.5):

1. Subdiviser l'image en plusieurs colonnes selon le choix, nous avons utilisé 08 colonnes pour les meilleurs résultats.
2. Pour chaque trois colonnes, On considère chacune des deux colonnes (deux à gauche et à droite) dans les trois comme une seule colonne, puis nous appliquons à chaque colonne une projection horizontale.
3. Enfin les lignes sont assemblées en regroupant des segments de ligne similaires. la direction de l'assemblage des lignes est basée sur l'alignement du texte. Dans le cas de l'arabe, nous commençons par la dernière colonne de droite jusqu'à la première colonne de gauche. La même chose peut être dite pour les langues latines, sauf qu'on utilise la direction opposée.

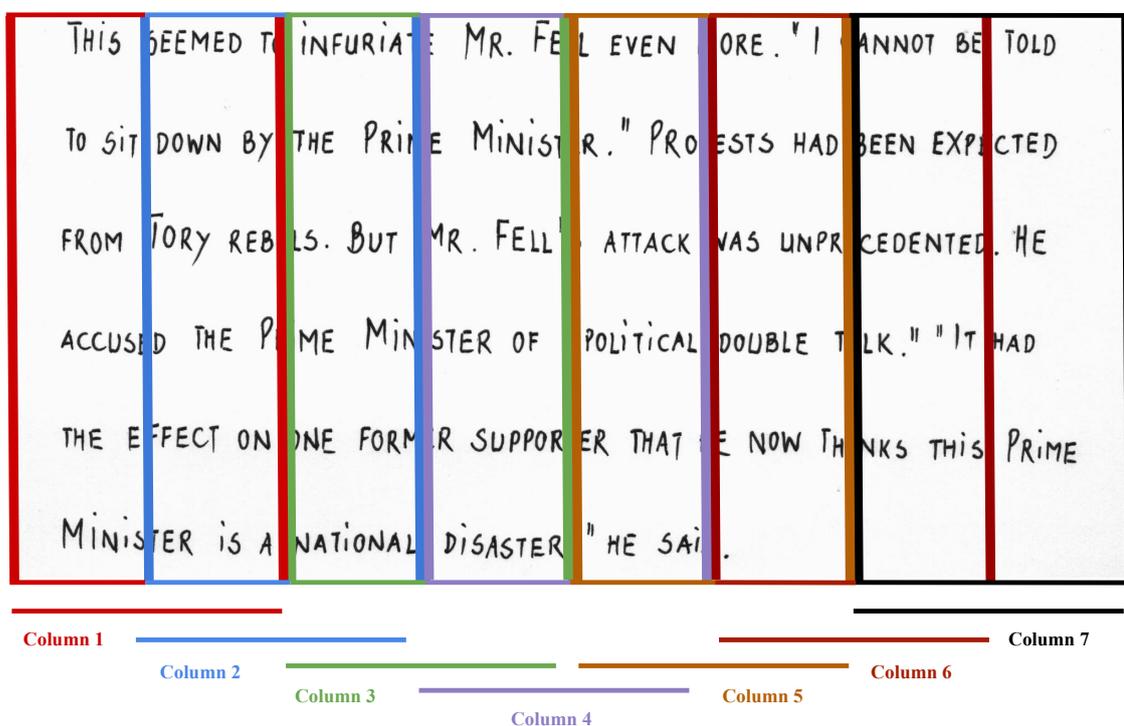


Figure 3.6: Segmentation basée sur la fenêtre glissante [26].

**Entrée:**

*img*: image binaire;  
*W, N*: entier;

**Début**

*N* ← 08;  
*W* ← largeur de l'image;  
Partitionner la largeur de l'image en *N* parties;  
(la largeur de chaque partie =  $W / N$ );

// appliquer une projection partielle sur chaque partie

**Pour** *i* de 1 à *N* - 3 **Faire**

*partieGauche* ← Les deux parties *partie*[*i*] et *partie*[*i*+1];  
    *partieDroite* ← Les deux parties *partie*[*i*+1] et *partie*[*i*+2];  
    // appliqué une projection partielle pour chaque deux parties  
    Calculer la projection horizontale locale des parties gauche et droite;  
    Calculer le seuil locale des parties gauche et droite;  
    Extraire la liste des lignes **LL** des parties gauche et droite;

**FinPour****Fin**

---

### 3.5.4 Projection partielle à deux fenêtre

similaire à la méthode précédente, elle est basée sur la projection partielle, elle consiste à diviser l'image du texte en cinq colonnes, ensuite appliquer une projection partielle sur la première et la dernière colonne, puis diviser l'image en deux moitiés. après ça appliquer une projection partielle sur l'une des deux moitiés en fonction de l'alignement du texte (voir la figure 3.7 ci-dessous). les étapes de cette méthode est comme suit (voir l'algorithme 3.6):

1. diviser l'image en cinq colonnes.
2. appliquer une projection partielle uniquement sur la première et la dernière colonne.
3. choisir parmi les deux la colonne avec le plus grand nombre de lignes.
4. diviser l'image en deux moitiés.
5. si la colonne avec le plus grand nombre de lignes est la dernière colonne de droite. alors appliquer une projection partielle uniquement sur la moitié gauche. sinon si la colonne avec le plus grand nombre de lignes est la première colonne de gauche. alors appliquer une projection partielle uniquement sur la moitié droite.



**Entrée:**

*img*: image binaire;  
*W, N*: entier;

**Début**

Partitionner la largeur de l'image en 05 parties;  
Appliquer une projection partielle pour la première partie gauche (*partie[1]*);  
Appliquer une projection partielle pour la dernière partie droite (*partie[5]*);  
Calculer le seuil puis extraire les lignes pour les deux parties locales;

*part1NumDeLignes*  $\leftarrow$  nombre de lignes de la première partie;  
*part5NumDeLignes*  $\leftarrow$  nombre de lignes de la dernière partie;  
Partitionner la largeur de l'image en deux moitiés;

**Si** *part1NumDeLignes* > *part5NumDeLignes* **Alors**

    Appliquer une projection partielle sur la moitié droite;  
    Calculer le seuil, puis extraire les lignes de la moitié droite;

**Sinon**

    Appliquer une projection partielle sur la moitié gauche;  
    Calculer le seuil, puis extraire les lignes de la moitié gauche;

**FinSi****Fin**

---

### 3.5.5 Méthodes Hybrides

pour améliorer la précision de détection et la performance du processus de segmentation, le principe de cette méthode est de combiner les deux méthodes: la première est la projection horizontale suivie par la classification des types de lignes en fonction de leurs hauteurs et la seconde est l'une des méthodes secondaires qui sont la projection partielle, fenêtre glissante, projection à deux fenêtres. le processus de classification des lignes permet d'extraire les lignes droites par projection et de laisser les autres lignes à extraire par l'une des méthodes secondaires avancées qui n'est appliquée que partiellement sur les segments de ligne non normaux plutôt que sur l'image entière.

Selon cette méthode, le document acquis par scanner contient deux types de lignes:

1. Lignes normales : elles possèdent des poids moyens.
2. Lignes chevauchées ou inclinées: elles possèdent des grandes hauteurs, ce sont le résultat du chevauchement des caractères ascendants et descendants, mais aussi des caractères collés des lignes adjacentes.

Il est à noter que la méthode de projection est essentielle pour le processus de classification des lignes, c'est pourquoi il est toujours appliquée contrairement à la seconde méthode, qui peut ou ne peut pas être appliquée selon le type de lignes qui existent dans l'image de texte (voir la figure 3.8 ).

les étapes de la méthode sont les suivantes (voir l'algorithme 3.7 ):

1. premièrement appliquer une projection horizontale sur toute l'image.
2. classifier les lignes en fonction de leurs hauteurs (lignes normales et non normales).  
Nous choisissons une hauteur fixe pour la classification.
3. appliquer l'une des méthodes secondaires uniquement sur les segments de ligne avec de grandes hauteurs.

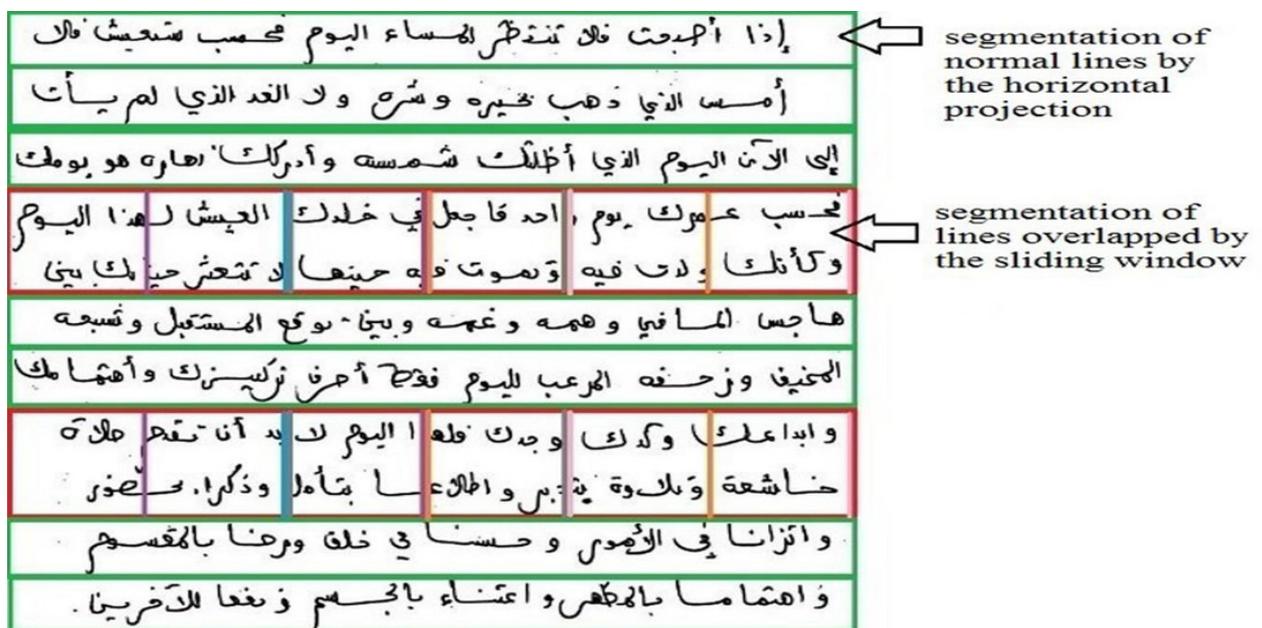


Figure 3.8: Segmentation par la méthode hybride [26].

en raison de la similitude des algorithmes des méthodes hybrides à l'exception de la méthode secondaire qui est différente pour chacun. Pour cette raison, l'algorithme suivant est pour les trois méthodes hybrides (horizontale et par fenêtre glissante, ou par projection

partielle, ou par projection partielle à deux fenêtre):

---

**Algorithme 3.7:** Méthode hybride

---

**Entrée:**

*img*: image binaire;  
*Ligne\_Normale*  $\leftarrow$  135: entier;  
*N*: entier;

**Début**

```
// appliquer une projection horizontale à l'image complète
Calculer le somme de pixel (projection) ;
Calculer le seuil;
Extraire la liste des lignes LL;

// classer les lignes en fonction de leurs hauteurs
Pour chaque ligne dans LL Faire
    Si taille de ligne > Ligne_Normale Alors
        //la hauteur de la ligne est grande qui signifie la ligne
        est biaisée ou en chevauchement
        Charger la ligne dans la liste des lignes anormales LA;
    Sinon
        //la hauteur de la ligne est normale qui signifie la ligne
        est droite
        Charger la ligne dans la liste des lignes normales LN;
    FinSi
FinPour
Pour chaque ligne dans LA Faire
    Appliquer l'une des méthodes secondaires de segmentation des lignes sur
    cette segment de ligne;
FinPour
Fin
```

---

### 3.6 Mapping des coordonnées

C'est la dernière étape de notre système, qui consiste à mapper les coordonnées obtenues dans la résolution basse aux coordonnées résolution haute d'origine. Ceci est fait en divisant les coordonnées Y des lignes extraites de la basse résolution par un facteur d'échelle. Ce facteur d'échelle est calculé par la division de la hauteur d'image basse résolution sur la hauteur de l'image originale (voir la figure 3.9).

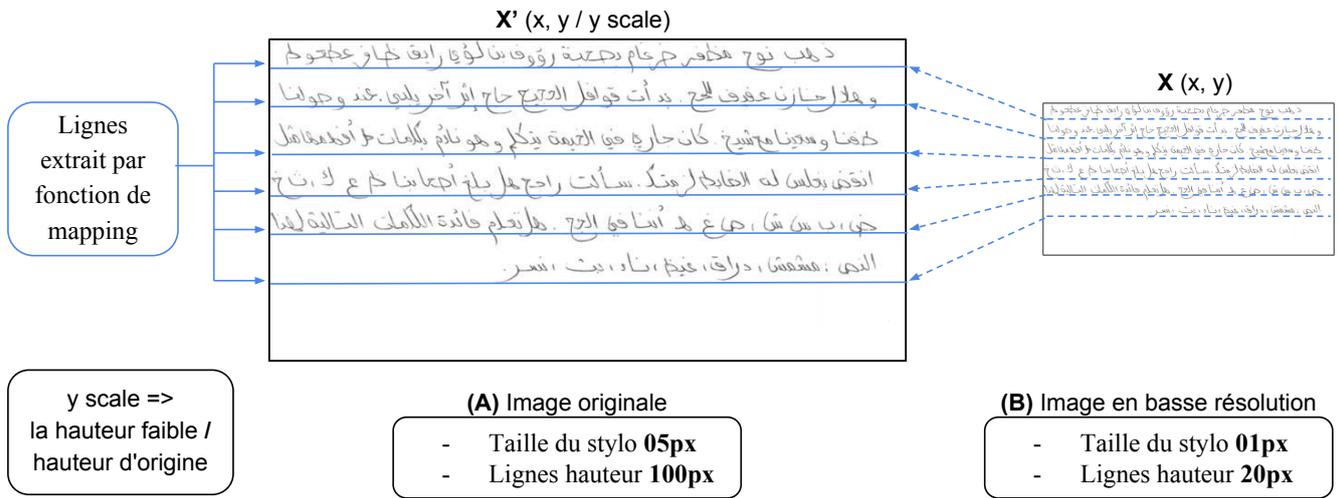


Figure 3.9: Processus de mapping des coordonnées a l'image originale.

## Conclusion

Dans ce chapitre, nous avons commencé par illustrer l'architecture globale du système proposé. Dans lequel nous avons expliqué en détail les étapes du système. Nous avons également présenté notre méthode d'accélération d'extraction des lignes de texte à base des modèles de cross résolution (XRM). Ensuite Nous avons vu comment les caractéristiques du texte sont utilisées par notre système afin de déterminer la meilleure résolution basse cible. Enfin, nous avons examiné comment les coordonnées sont transformées par l'utilisation d'une fonction de mapping des coordonnées.

Dans le chapitre suivant Les résultats expérimentaux et la comparaison avec les sept méthodes de segmentation existantes démontrent l'efficacité de la technique proposée en termes de vitesse et de taux de détection.

# Réalisation et validation

## Introduction

Dans le chapitre précédent, d'abord nous avons vu l'architecture générale du système et présenté en détail la conception du système et la méthode d'accélération proposée, où laquelle nous avons vu comment le système exploite les caractéristiques du texte pour produire une image de plus petite taille que l'original. Ensuite, nous avons vu comment le système utilise les coordonnées de basse résolution pour extraire les lignes de texte. Après cela, nous avons fait un aperçu rapide sur les méthodes mises en œuvre et les résultats de la segmentation qu'elles produisent. Enfin, nous discuterons et comparerons les résultats de différentes méthodes de segmentation des lignes.

Dans ce chapitre, nous allons présenté au premier lieu le langage de programmation C# choisit, ainsi que la bibliothèque utilisée Open CV. Ensuite, nous allons décrire les interfaces et fenêtres principales de l'application. Finalement, nous mettons l'accent sur les résultats obtenus en les comparant dans les différents cas possibles.

## 4.1 Choix du langage de programmation

Dans ce travail, nous avons utilisé l'environnement de programmation .NET C# principalement pour sa rapidité, sa facilité de développement et sa flexibilité, ainsi que pour les avantages suivants :

- Langage orienté objet
- Libération automatique de la mémoire
- Multitâches simplifiée
- Presque tous les contrôles de Windows (boutons, listes déroulantes, menus ... etc.) qui sont représentés par des classes.

## 4.2 Bibliothèque utilisée

Nous avons utilisé Open CV (Open Source Computer Vision) qui est une bibliothèque de source ouverte et de fonctions de programmation multi-plateforme pour la vision artificielle en temps réel afin de nous aider dans le processus de développement de notre application.

Les principaux avantages de la bibliothèque sont :

- La vitesse
- Portabilité
- Facilité d'utilisation
- Gestion de la mémoire
- Faible utilisation de la mémoire

## 4.3 Interface et Fenêtres

Lors du lancement de l'application, l'interface principale représentée dans la figure 4.1 ci-dessous est affichée. Elle contient une liste déroulante de méthodes et des onglets de contrôle. Ces onglets contiennent deux pages chacune pour un mode d'utilisation différent.

Le premier page est pour le mode de segmentation des lignes normale sans utiliser la résolution croisée. Elle contient deux boîtes d'images, l'une pour afficher les résultats de segmentation et l'autre pour afficher l'histogramme de projection.

La deuxième page concerne le mode de segmentation des lignes avancées en utilisant la résolution croisée. Elle contient trois boîtes d'images, deux servent à afficher les résultats de segmentation en résolutions basse et haute, et l'autre pour afficher l'histogramme de projection. Elle contient également quatre zones de texte, deux pour l'affichage de la résolution normale et basse, les deux autres pour l'affichage des propriétés du texte.

Afin d'utiliser le programme, nous devons d'abord sélectionner une méthode dans la liste déroulante, puis sélectionner l'un des deux modes de fonctionnement, ensuite glisser et déposer une image de document, après ce point, aucun travail supplémentaire n'est requis de l'utilisateur, le programme exécutera automatiquement le nécessaire opérations et enfin afficher les résultats de la segmentation des lignes.

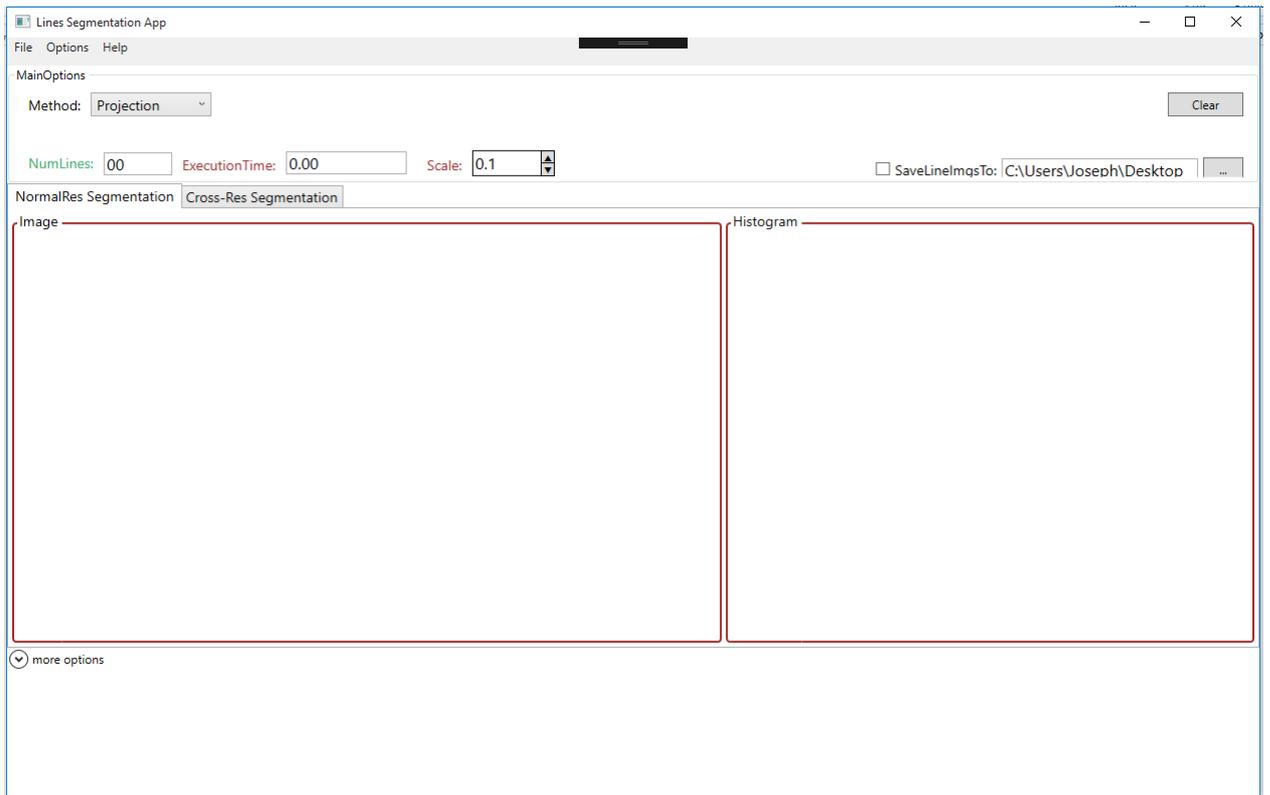


Figure 4.1: Interface principale de l'application

### 4.3.1 Pré-traitement

une fois que l'image est glissé et déposé, dans le cas du mode avancé, le programme convertit automatiquement l'image en binaire, puis calcule la taille du stylo et la hauteur des lignes. Cela préparera l'image pour la prochaine étape de la segmentation des lignes.

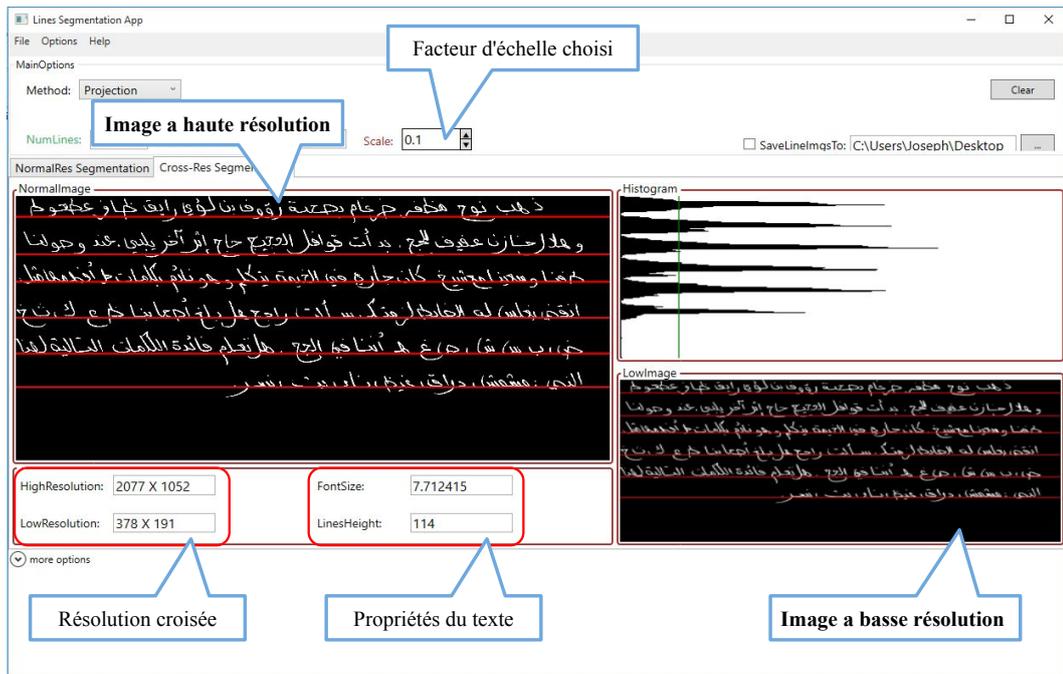


Figure 4.2: Résultats de la segmentation en utilisant le modèle de résolution croisée

#### 4.3.2 Extraction des lignes

Dans cette étape, le programme va appliquer la méthode d'extraction choisie, dans le cas du mode avancé la méthode d'extraction est appliqué sur l'image a basse résolution.



Figure 4.3: Extraction des lignes par projection horizontale

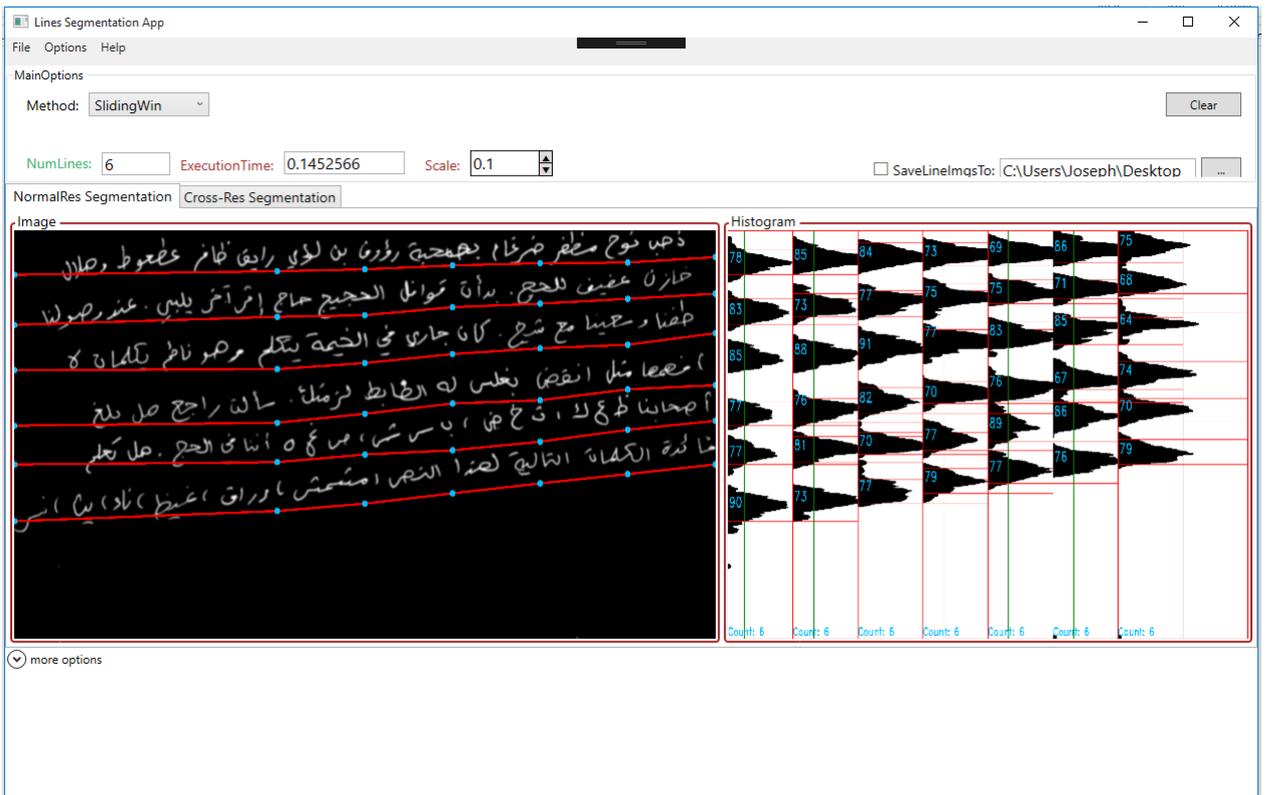


Figure 4.4: Extraction de lignes par fenêtre glissante

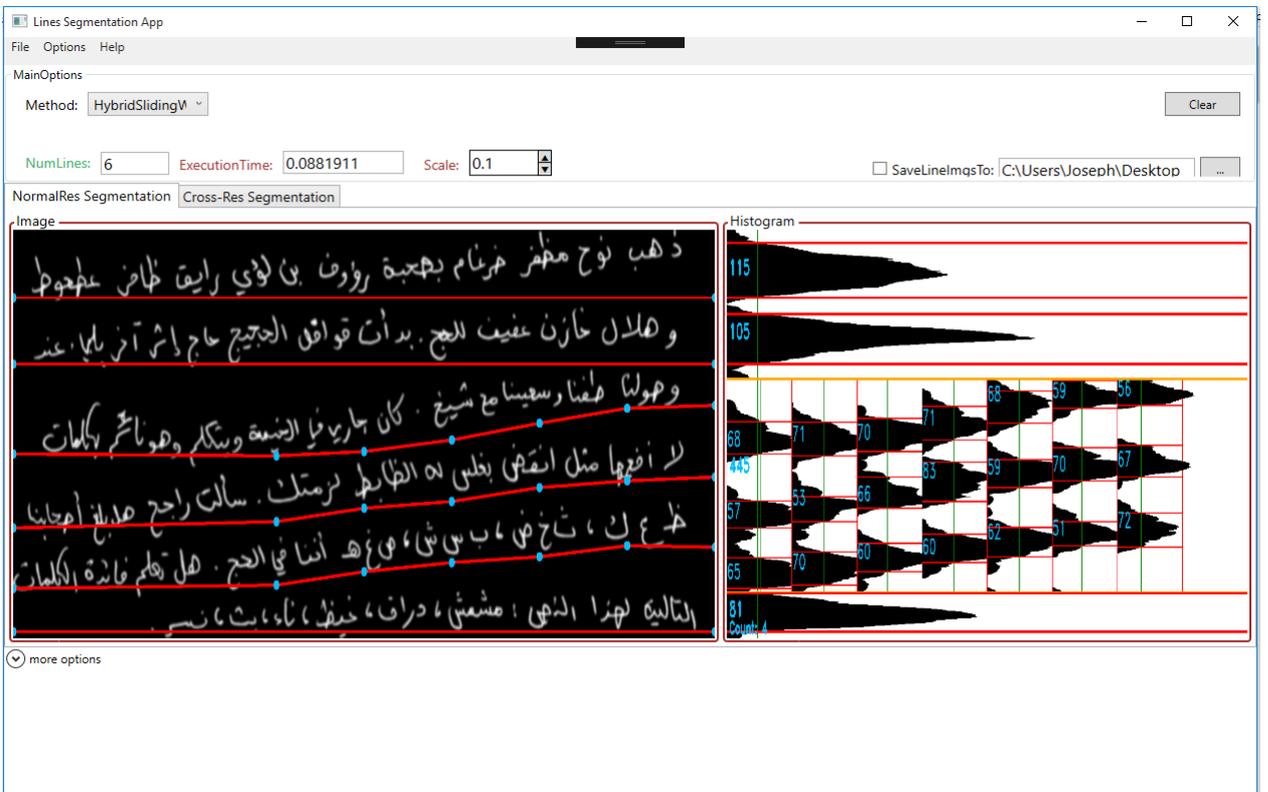


Figure 4.5: Extraction de lignes par fenêtre glissant hybride

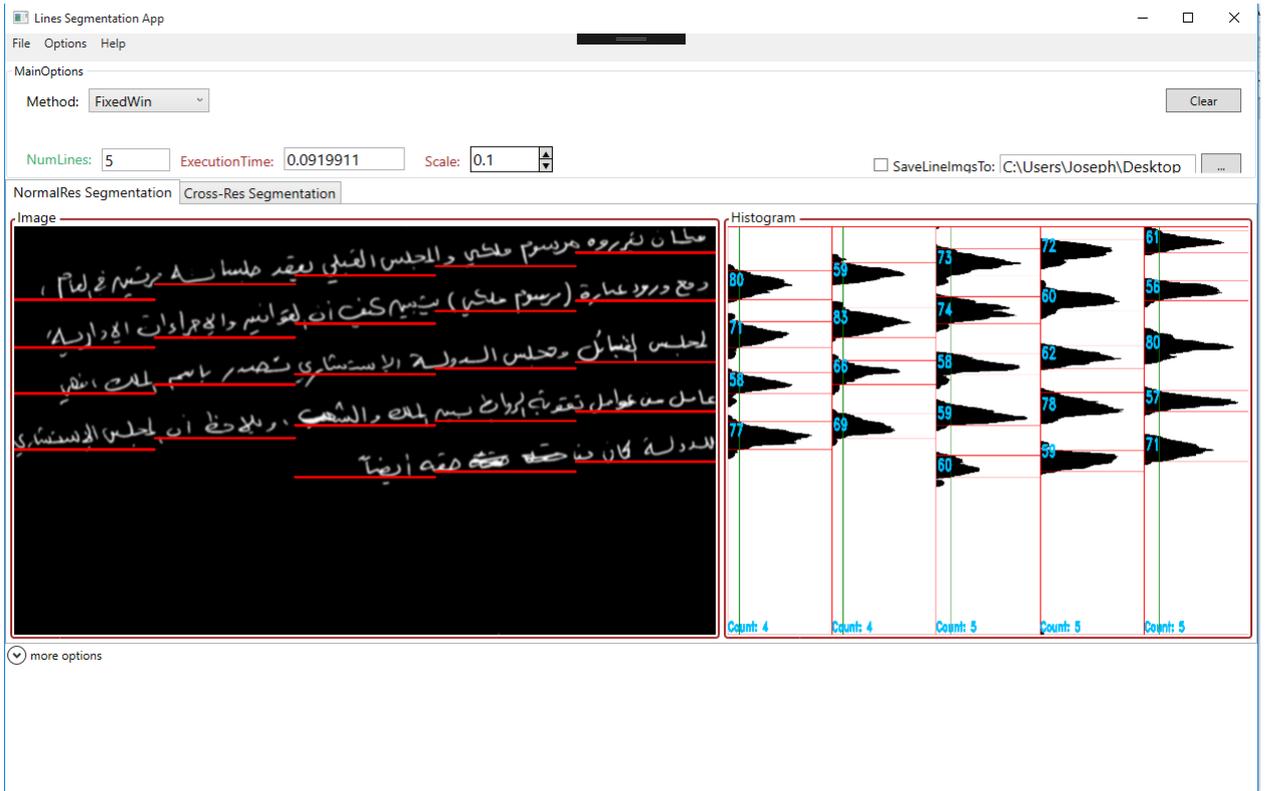


Figure 4.6: Extraction de lignes par fenêtre fixe

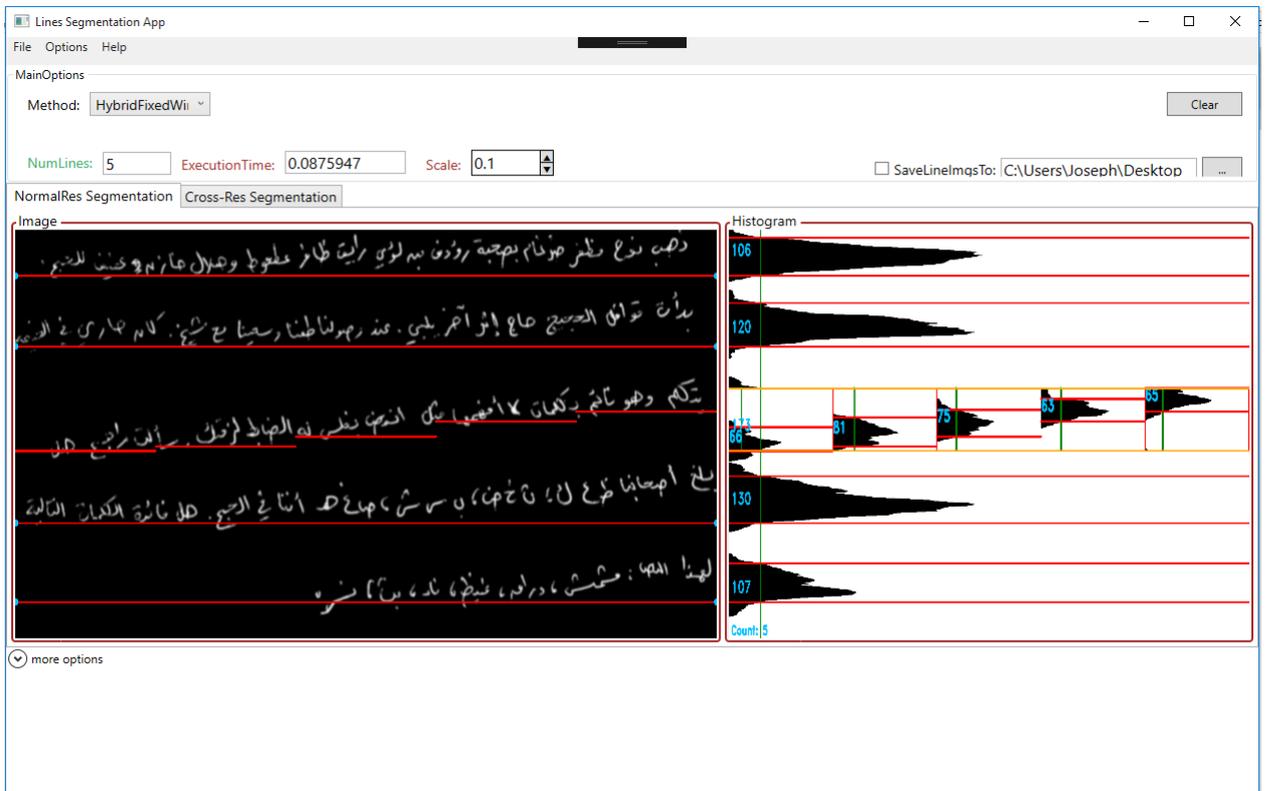


Figure 4.7: Extraction de lignes par fenêtre fixe hybride

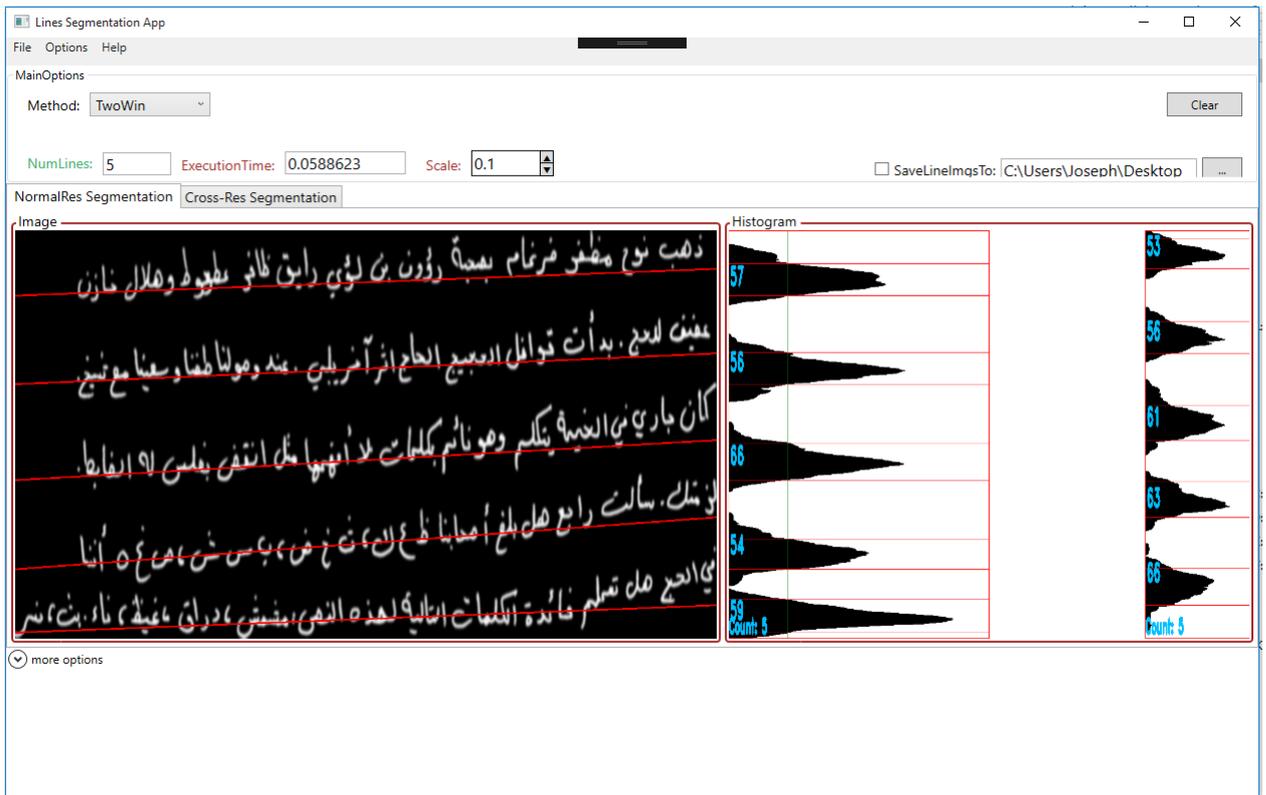


Figure 4.8: Extraction de lignes par double fenêtre

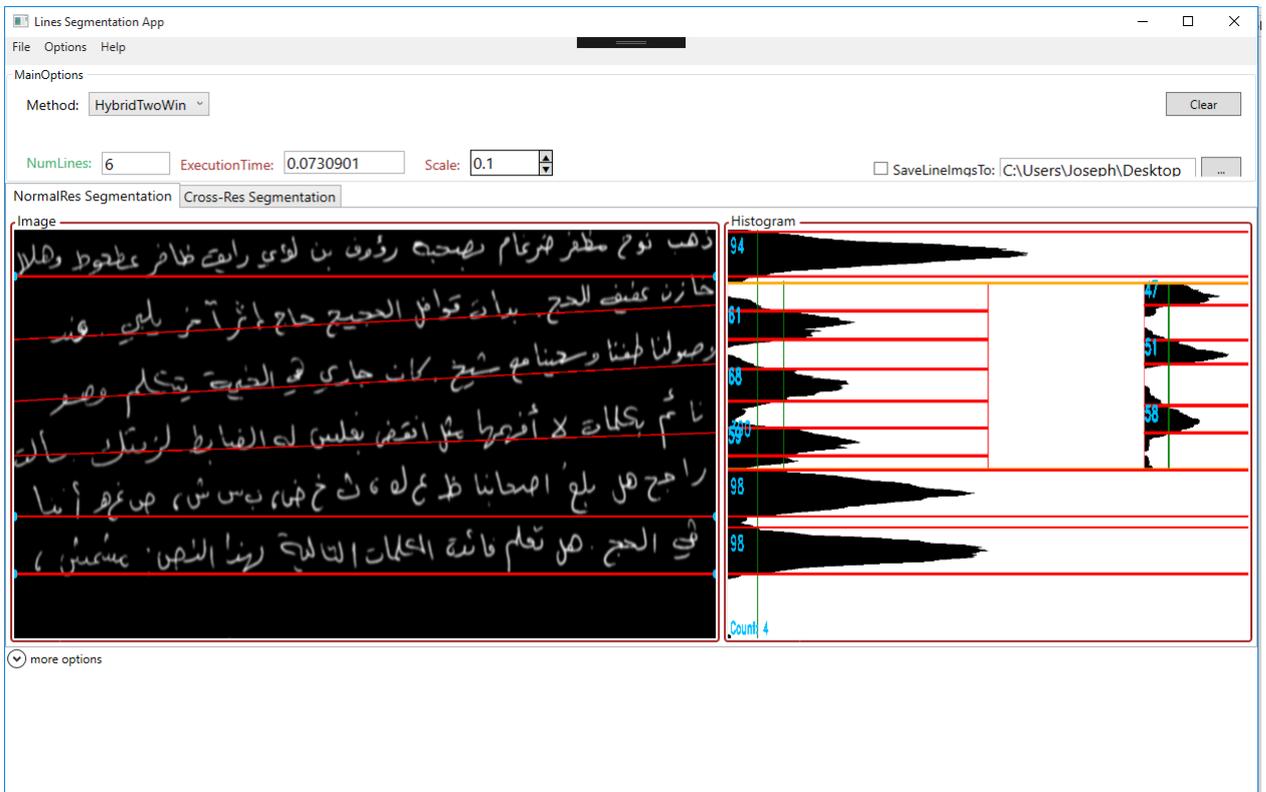


Figure 4.9: Extraction de lignes par double fenêtre hybride

### 4.3.3 Mapping des coordonnées

Une fois que le programme atteint cette étape, tout ce qui reste est de transformer les coordonnées de résolution basse en coordonnées de résolution normale. Enfin les résultats globaux de la segmentation seront affichés à l'utilisateur.

## 4.4 Résultats et tests

Pour tester l'efficacité du modèle XRM. Nous avons utilisé des images de texte de deux bases de données différentes d'arabe et d'anglais manuscrits.

Afin d'évaluer la performance des algorithmes de segmentation des lignes de texte. Deux tests sont effectués, le premier est un test de taux de détection et le second est un test de vitesse d'exécution. chaque test est appliqué selon deux résolutions différentes, une résolution normale et une résolution basse.

### 4.4.1 KFUPM Handwritten Arabic Text Database

Dans ce cas, Les résultats des tests sont obtenus en utilisant des images de texte manuscrites arabes de la base de données de la Fondation KHATT (KFUPM Handwritten Arabic Text) menée par l'Université King Fahd du Pétrole et des Minéraux (KFUPM). C'est une base de données de texte arabe manuscrit sans contrainte écrite par 1000 auteurs différents. L'ensemble de données de test se compose de 2000 manuscrits avec le nombre total de 11562 lignes de texte. Il comprend des images difficiles en termes de segmentation des lignes de texte. Ceux-ci incluent des images avec des lignes de longueurs différentes, des degrés d'inclinaison et des chevauchements. Le texte a été écrit sur papier sans lignes directrices. Les images ont été scannées à une résolution de 300 dpi.

#### A. Taux de détection

On peut voir que la première méthode de projection montre les résultats les plus faibles par rapport aux autres méthodes, cela est parce que la sensibilité de la méthode à l'inclinaison et au chevauchement, déformation qui existe dans la plupart des lignes de cette base de données.

Généralement, nous remarquons que le taux de détection minimal est de 80% à travers toutes les méthodes, ce qui indique la puissance et la qualité du taux de détection. Cependant, il existe des différences mineures entre le taux de détection de basse et de haute résolution causé par la perte d'information.

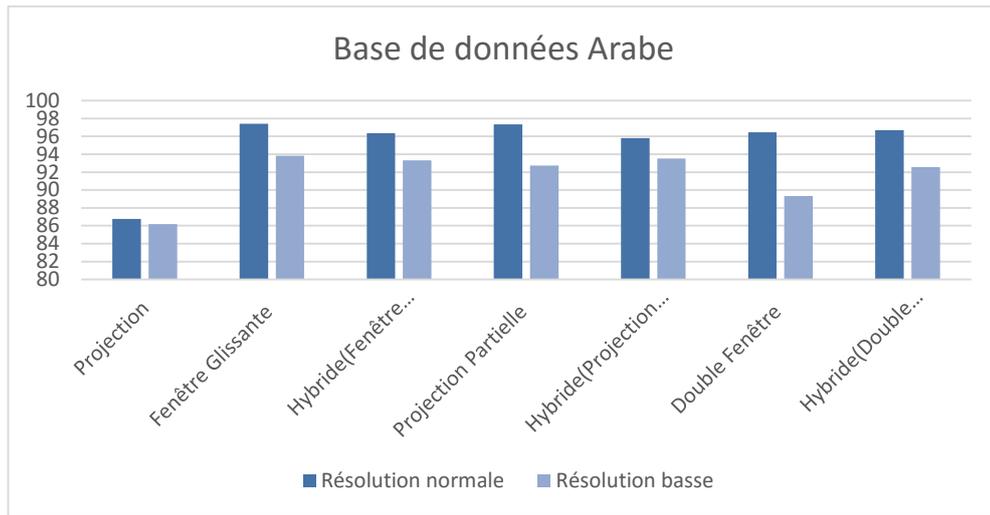


Figure 4.10: Taux de détection des lignes de base de données arabes.

## B. Vitesse d'exécution

Les courbes suivantes, nous permettent de comparer la vitesse d'exécution à différentes résolutions d'images, pour la base de données arabe KHATT:

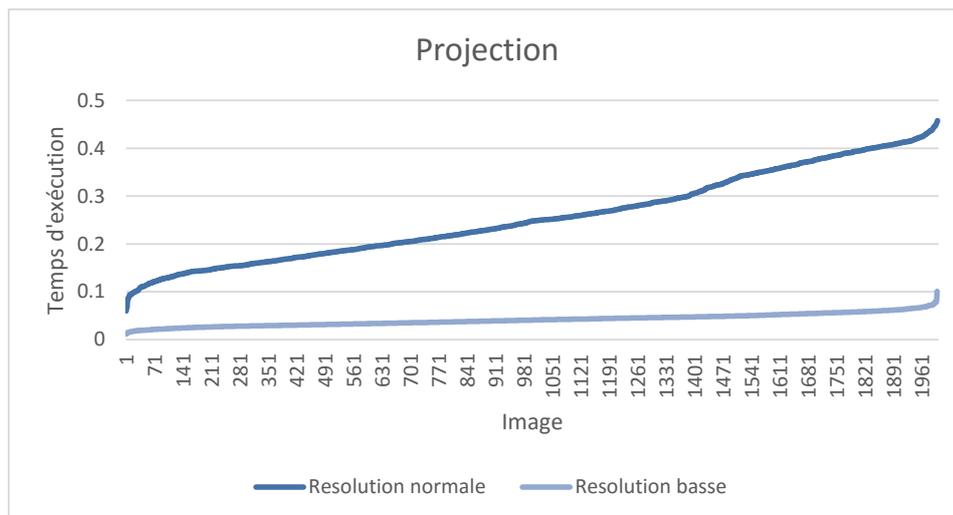


Figure 4.11: Temps d'exécution de la méthode de projection.

La première figure de la méthode de projection montre qu'il y a une grande différence de 6 fois entre les deux résolutions. Cette différence est due à la taille de la petite image qui est minimisée plusieurs fois.

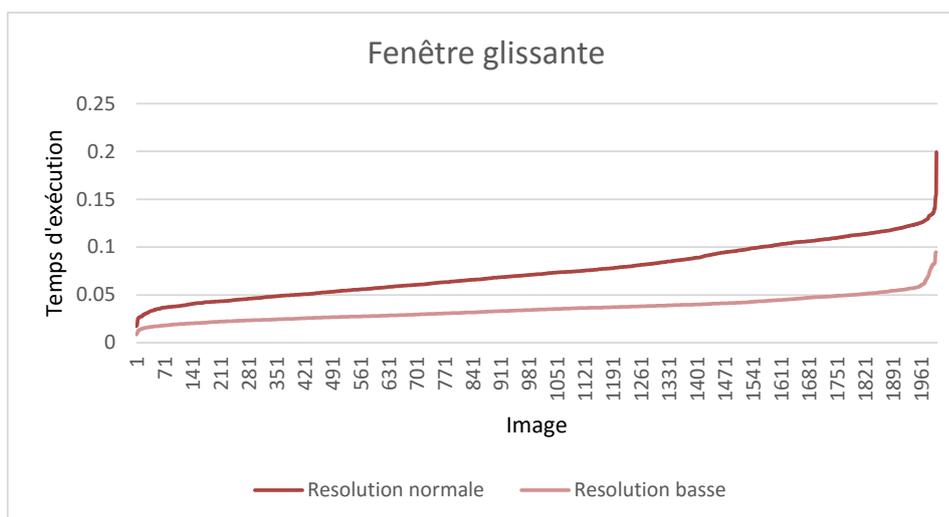


Figure 4.12: Temps d'exécution de la méthode de fenêtre glissante.

La deuxième figure de la méthode de fenêtre glissante montre une légère différence de 2,1 fois, ce qui signifie que la basse résolution est seulement deux fois plus rapide que la haute résolution. Il convient de noter que la taille de la fenêtre joue un rôle important dans la détermination de la vitesse globale et de la précision de cette méthode.

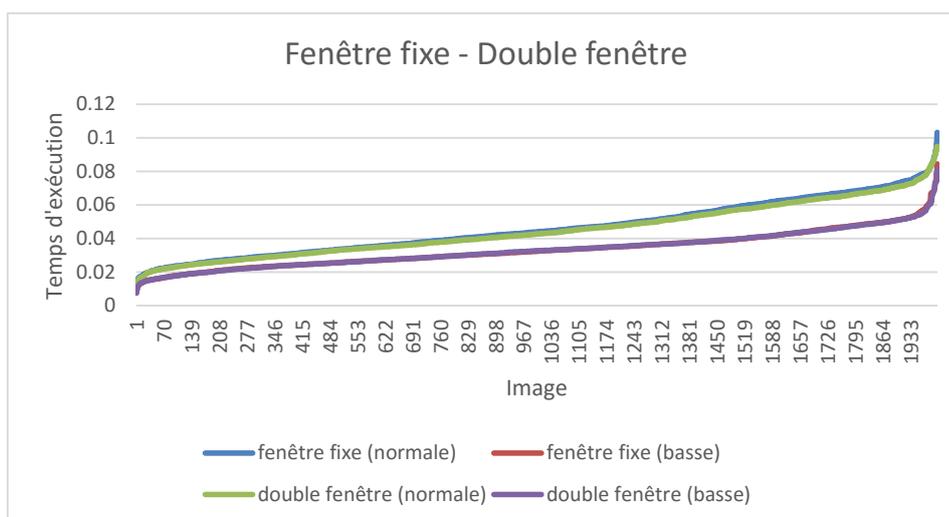


Figure 4.13: Temps d'exécution de la fenêtre glissante et de la méthode à double fenêtre.

La troisième figure de la fenêtre fixe et de la méthode à deux fenêtres montre des résultats proches et une différence mineure de 1,3 fois, ce qui signifie que le temps de résolution faible est approximativement le même que le temps en haute résolution. La différence insignifiant de vitesse peut s'expliquer par le fait que les méthodes sont trop rapides pour remarquer la différence dans les deux cas. cela peut être prouvé si on regarde le pire temps d'exécution qui est juste 0,08 ms dans le cas de la résolution normale. La similitude des résultats est due au fait que le processus de projection qui est effectué par chacun d'eux est presque le même dans les deux méthodes avec seulement de légères différences. il convient de noter que la différence entre eux est de savoir comment chacun extrait des

lignes après le processus de projection.

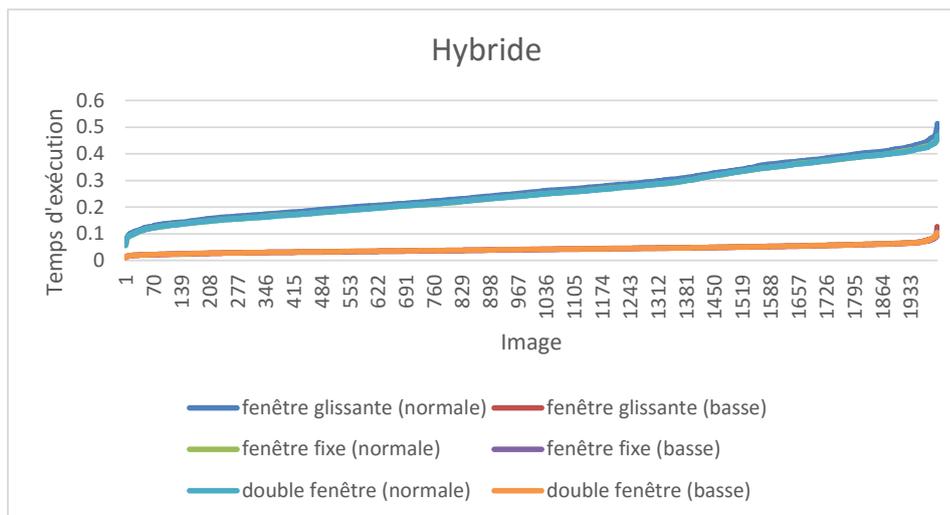


Figure 4.14: Temps d'exécution des méthodes hybrides.

La quatrième figure des trois méthodes hybrides montre des résultats très proches, donc la différence est la même pour tous, qui est de 6,3 fois. la différence est due à la taille de la petite image. La similitude des résultats dans ces trois diagrammes est due au fait que ces méthodes hybrides sont une combinaison de deux méthodes. La première est la méthode de projection et la seconde est l'une des autres méthodes. La méthode qui s'exécute le plus entre les deux est la méthode de projection. Les autres méthodes secondaires s'exécutent rarement, et même si elles s'exécutent, elles ne s'exécutent que sur de petits segments, c'est pourquoi elles ont un impact mineur sur les performances. cela a rendu la méthode de projection très influente. cela signifie qu'il joue un rôle important dans la détermination de la vitesse globale de ces méthodes hybrides.

#### 4.4.2 IAM Handwriting Database

Dans ce cas, les résultats des tests sont obtenus en utilisant la base de données anglaise de l'Université de Berne, Suisse. La base de données contient des formes de texte manuscrit sans contrainte, écrites par 657 auteurs différents. il comprend principalement des lignes droites avec un petite chevauchement dans certains cas. la base de données se compose de 1539 images de texte scanné avec un total de 13353 lignes de texte. Les images ont été scanné à une résolution de 300 dpi et enregistrées sous forme d'images PNG en niveaux de gris.

#### A. Taux de détection

La première méthode de projection montre les résultats les plus élevés par rapport aux autres méthodes, parce que la plupart des lignes dans cette base de données sont des lignes droites avec juste un peu de chevauchement dans certain cas.

La troisième méthode de la fenêtre glissante et la quatrième méthode de projection partielle montrent les résultats les plus faibles par rapport aux autres méthodes, avec un taux de détection de 60% pour la fenêtre glissante et de 53% pour la projection partielle. Les résultats de détection faibles sont dus à des erreurs qui se produisent dans l'assemblage des segments de ligne dans les deux méthodes.

Comme pour toutes les autres méthodes, nous remarquons généralement, qu'il y a une différence mineure entre les taux de détection de faible et haute résolution qui est causée par la perte de l'information.

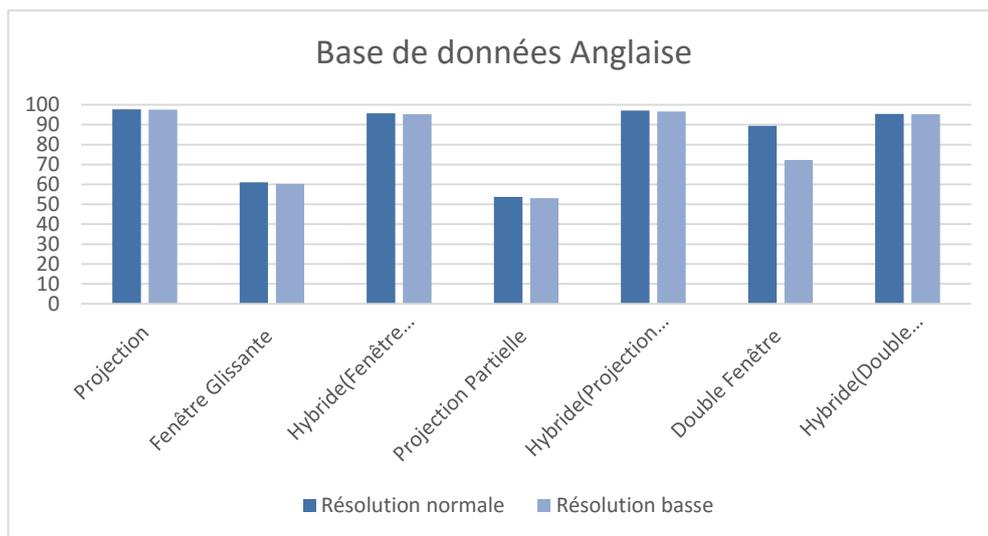


Figure 4.15: Taux de détection des lignes de base de données anglais.

## B. Vitesse d'exécution

Dans les courbes suivants, nous comparerons la vitesse d'exécution à différentes résolutions d'images, pour la base de données anglaise:

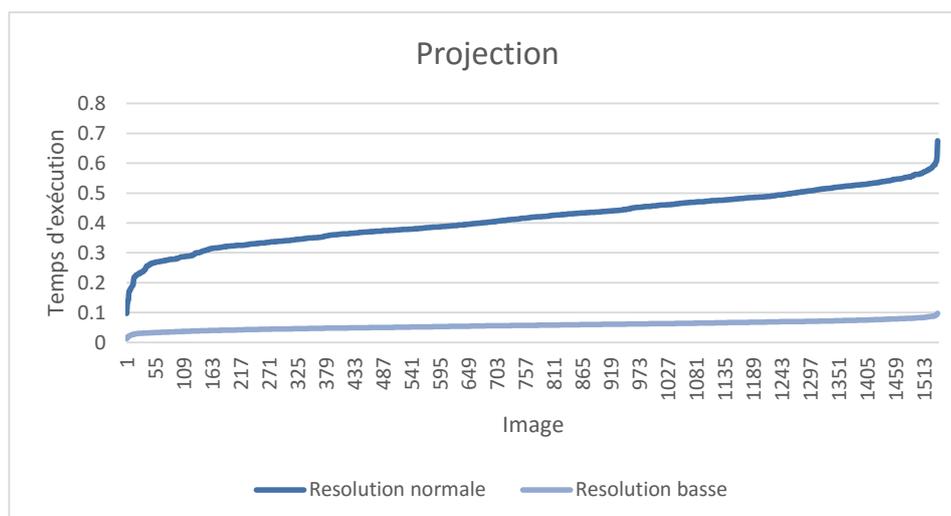


Figure 4.16: Temps d'exécution de la méthode de projection.

La première figure de la méthode de projection montre une très grande différence de 7,1 fois entre les deux résolutions. Ce qui est une grande différence évidente. Cette différence est due à la taille petite de l'image qui est minimisée plusieurs fois.

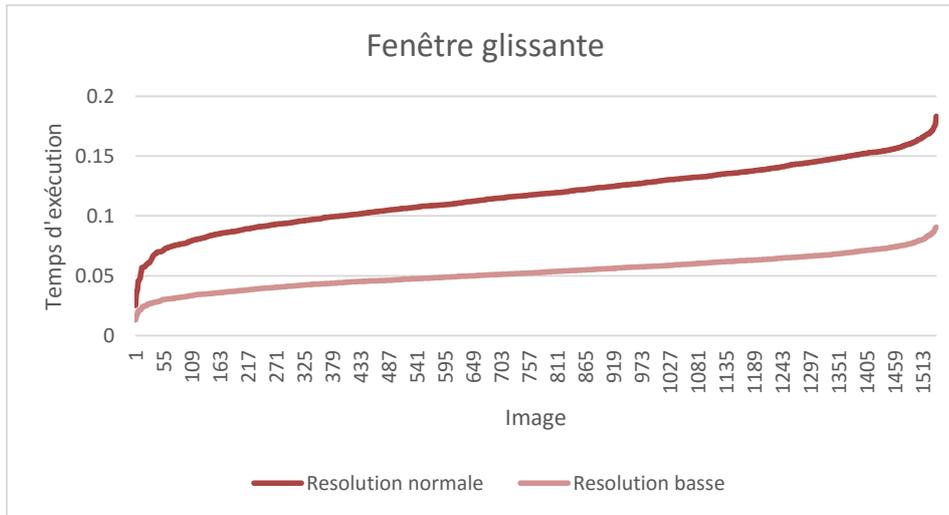


Figure 4.17: Temps d'exécution de la méthode de fenêtre glissante.

La deuxième figure de la méthode de la fenêtre glissante ne montre pas beaucoup de différence importante qui n'est que 2,1 fois comme la dernière fois, ce qui signifie que la basse résolution est seulement deux fois plus rapide que la haute résolution. la différence est remarquablement similaire à ce que nous avons discuté précédemment dans la fenêtre glissante dans le cas de la base de données arabe.

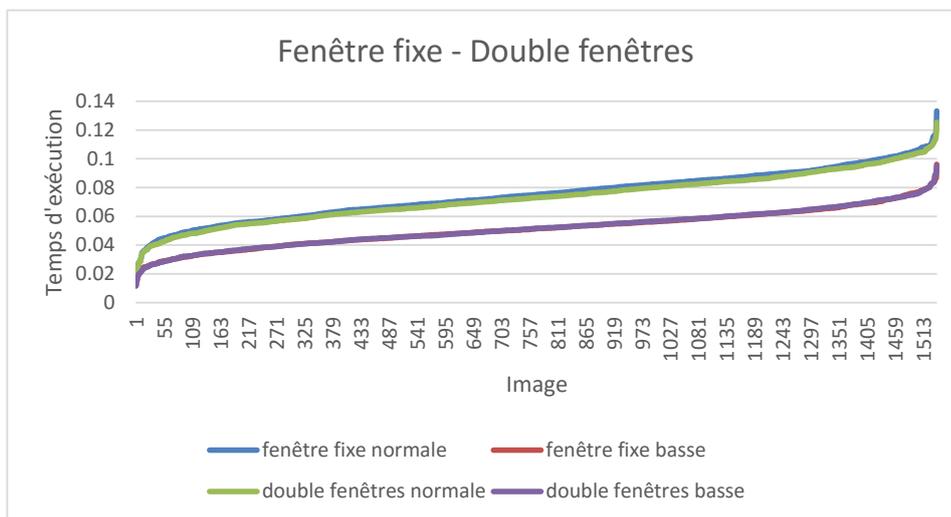


Figure 4.18: Temps d'exécution de la fenêtre glissante et de la méthode à double fenêtre.

La troisième figure de la fenêtre fixe et de la méthode à deux fenêtres montre des résultats proches et une petite différence de 1,4 fois, ce qui signifie que le temps de la basse résolution est presque le même que le temps de la haute résolution. La différence insignifiante de vitesse peut être expliquée comme la dernière fois par le fait que les

méthodes sont trop rapides pour remarquer la différence dans les deux cas (résolution haute, basse).

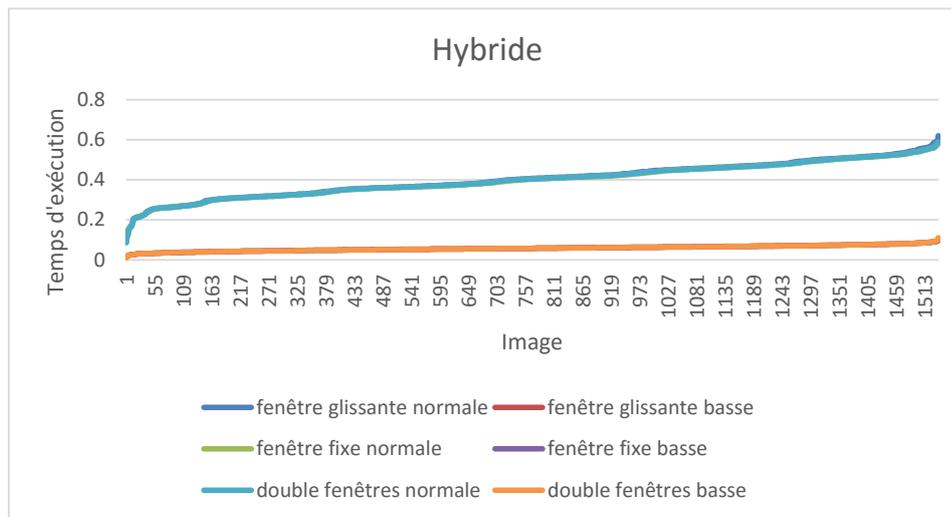


Figure 4.19: Temps d'exécution des méthodes hybrides.

La quatrième figure des trois méthodes hybrides montre des résultats très proches comme discuté précédemment, mais cette fois, la différence est de 6,8 fois. la différence est grande et c'est évidemment à cause de la taille de la petite image.. La similitude des résultats est due au fait que la méthode de projection est la méthode dominante qui est exécutée dans la plupart des cas..

## 4.5 Comparaison des résultats

A ce niveau, Nous pouvons comparer en même temps la vitesse d'exécution et le taux de détection entre toutes les méthodes dans les deux cas, par les résultats moyens présenté dans les tableaux ci-dessus:

Méthode	Résolution normale		Résolution basse	
	Taux de détection (%)	Temps d'exécution (ms)	Taux de détection (%)	Temps d'exécution (ms)
Projection	86.7	0.25	86.1	0.041
Fenêtre Glissante	97.4	0.074	93.8	0.035
Fenêtre Glissante (Hybride)	96.3	0.26	93.3	0.041
Fenêtre fixe	97.3	0.046	92.7	0.033
Fenêtre Fixe (Hybride)	95.8	0.25	93.5	0.041
Double Fenêtre	96.4	0.044	89.3	0.033
Double Fenêtre (Hybride)	96.6	0.25	92.5	0.042

Figure 4.20: Moyens des résultats de la base de données arabe KHATT.

Globalement, nous remarquons de bons taux de détection pour toutes les méthodes dans les deux cas (normale et basse), sauf pour la méthode de projection où la vitesse d'exécution est significative, mais au prix d'avoir le taux de détection le plus faible parmi toutes les autres méthodes.

Méthode	Résolution normale		Résolution basse	
	Taux de détection (%)	Temps d'exécution (ms)	Taux de détection (%)	Temps d'exécution (ms)
Projection	97.7	0.41	97.5	0.057
Fenêtre Glissante	61	0.11	60.2	0.052
Fenêtre Glissante (Hybride)	95.7	0.40	95.1	0.058
Fenêtre fixe	53.7	0.07	53.1	0.051
Fenêtre Fixe (Hybride)	97	0.39	96.6	0.058
Double Fenêtre	89.3	0.07	72.2	0.051
Double Fenêtre (Hybride)	95.3	0.39	95.3	0.058

Figure 4.21: Moyens de résultats de la base de données anglais IAM.

Nous observons dans la fenêtre glissante, fixe et la fenêtre double dans les deux cas de résolution (normale et basse), les taux de détection sont très proches à l'exception de la vitesse d'exécution où il existe une différence entre la fenêtre glissante et les deux autres méthodes. Où, la vitesse d'exécution dans la méthode de fenêtre glissante en cas de la résolution normale est deux fois plus rapide que les deux autres méthodes.

Globalement, les meilleures méthodes selon les statistiques en termes de taux de détection et de temps d'exécution sont les méthodes hybrides.

## **Conclusion**

Dans ce chapitre, nous avons présenté les résultats expérimentaux obtenus à partir des sept méthodes et à travers l'utilisation des bases de données arabes et anglaises. Nous avons discuté les résultats en les comparant les uns aux autres, ce qui nous a finalement démontré l'efficacité du modèle XRM proposé pour accélérer le processus de segmentation des lignes de texte, ce qui ouvre à son tour de nouvelles perspectives et possibilités dans le domaine de la segmentation des lignes.

## Conclusion et perspectives

L'objectif principal de notre travail est d'améliorer le temps d'exécution de la segmentation des lignes en utilisant des modèles de résolution croisée qui accélèrent considérablement le temps de détection des lignes.

L'inconvénient majeur de la segmentation des lignes pour les approches existantes est le temps d'exécution de cette opération qui affecte directement la performance globale des systèmes OCR. Pour remédier à ce problème, nous avons proposé une nouvelle approche basée sur le modèle XRM. Le système que nous proposons exploite l'idée que toutes les méthodes de segmentation des lignes utilisent souvent des images avec une haute résolution de 150 à 300 dpi. Donc la réduction de cette taille aura un impact direct sur le temps d'exécution du processus de segmentation. Mais le processus de réduction d'échelle ne devrait pas être en utilisant une échelle fixe. donc, pour des résultats optimaux, le système approxime les propriétés du texte et minimise la taille en conséquence.

De cette étude, nous avons apporté une contribution qui profite de l'idée des modèles de résolution croisée pour surmonter le problème du temps de calcul perdu lors de l'extraction des lignes de texte.

Nous avons mis en œuvre l'approche que nous avons proposée en C#. Pour évaluer les performances de notre système en termes de taux de détection et de vitesse d'exécution, nous avons mené une étude expérimentale sur des manuscrits arabes et anglais. Les résultats obtenus montrent que notre système a permis d'accélérer significativement l'extraction des lignes et donne des résultats proche qu'avec l'image en résolution de 150 ou 300 dpi.

Le travail que nous avons fait au cours de ce mémoire est un grand pas dans le domaine de la segmentation des lignes. Cependant nous pensons qu'il peut être amélioré, et étendu par les points suivants :

- Utiliser une unité de mesure indépendante comme le pourcentage pour la hauteur des lignes, la taille du stylo et le seuil de classification au lieu des pixels.
- Extraire les propriétés de texte en utilisant d'autres méthodes améliorées.
- Utiliser le modèle XRM à des densités inférieures à 300 dpi en améliorant l'opération de redimensionnement pour prendre en compte la taille de l'image (largeur et hauteur) avant le processus de redimensionnement.
- Utiliser l'idée du modèle XRM dans d'autres types de méthodes de segmentation des lignes.
- Étendre l'idée du modèle XRM en l'utilisant dans l'étape suivante du segmentation des mots.



# Bibliographie

- [1] “Optical character recognition (ocr) – how it works” – <https://www.nicomsoft.com/optical-character-recognition-ocr-how-it-works>, 2012 (visité 2 Février, 2018).
- [2] A.CHEUNG\*M.BENNAOUN\*N.W.BERGMANN\* – “An arabic optical character recognition system using recognition-based segmentation”, **34** (2001), no. 2, p. 215–233.
- [3] H. AL-RASHAIDEH – “Preprocessing phase for Arabic Word Handwritten Recognition”, *INFORMATION TRANSMISSIONS IN COMPUTER NETWORKS* **6** (2006), p. 11–19.
- [4] O. ALAQL – “Segmentation for handwritten documents”, <http://slideplayer.com/slide/7563572/>, Accès: 2018-02-4.
- [5] A. AMIN – *Recognition of printed and handwritten arabic characters*, Springer Verlag, 2005.
- [6] M. AMROUCH – “Reconnaissance de caractères imprimés et manuscrits, textes et documents basée sur les modèles de markov cachés”, Thèse, Université Ibn Zohr Faculté des Sciences d’Agadir, 2012.
- [7] A. BELAÏD – “Conception automatisée de modèles de page en vue de leur utilisation en reconnaissance de documents”.
- [8] A. M. A. M. A. M. DR. YUSUF PERWEJ, SHAIKH ABDUL HANNAN – “An overview and applications of optical character recognition”, **3** (2014), no. 7.
- [9] S. K. EL MOUBTAHIJ HICHAMA, HALLI AKRAMB – “Using features of local densities, statistics and hmm toolkit (htk) for offline arabic handwriting text recognition”, *Journal of Electrical Systems and Information Technology* **4** (2017), no. 3, p. 387–396.
- [10] A. A. ELKARIM – *Proposition d’une architecture multiagents pour la reconnaissance de l’écriture manuscrite arabe*, Mémoire, Université de Jijel, 2009.
- [11] M. I. ET GEUDDA ADEL – *Contribution à la reconnaissance d’écriture arabe manuscrite en utilisant la classification neuronale*, Mémoire, UNIVERSITE KASDI MERBAH, OUARGLA, 2016.
- [12] N. GHANMI – “Segmentation d’images de documents manuscrits composites : applica documents de chimie”, Thèse, Université de Lorraine, 2016.
- [13] S. HAITAAMAR – *Segmentation de textes en caracteres pour la reconnaissance optique de l’écriture arabe*, Mémoire, Université de Batna 2, 2007.
- [14] P. S.-P. W. HORST BUNKE – *Handbook of character recognition and document image analysis*, 2nd éd., World Scientific Pub Co Inc, 1997.
- [15] M. B. G. IMANE – *Proposition d’un modèle de classificateur logique, application à la reconnaissance du texte arabe imprimé*, Mémoire, UNIVERSITÉ D’EL-OUED, 2014.
- [16] Y. LASRI – *Contribution à la reconnaissance optique (ocr) du texte arabe imprimé lieu de stage*, Mémoire, UNIVERSITÉ Sidi Mohamed Ben Abdellah, 2014.
- [17] B. T. LAURENCE LIKFORMAN-SULEM\*, ABDERRAZAK ZAHOUR\*\* – “Text line segmentation of historical documents: A survey”, (2007).
- [18] M. M. E. MAMOUN – *modèles svm multi-classes application à la reconnaissance des caractères arabes*, Mémoire, Université d’Oran Mohamed BOUDIAF, 2012.

- [19] J. B. M.D. GANIS, C.L. WILSON – “Neural network-based systems for handprint ocr applications”, **7** (1998), no. 8, p. 1097 – 1112.
- [20] M. J. NASRIWALA – “Introduction to optical character recognition”, <http://srimca.edu.in/srijan/Srijanjan2011/IT/IntroductionToOCR.htm>, 2011, Accès: 2018-02-1.
- [21] G. OUASSILA – *Accélération d’extraction des lignes du texte à base des modèles de cross résolution*, Mémoire, UNIVERSITÉ D’EL-OUED, 2017.
- [22] P. D. S. A. A. M. RANDA I. M. ELANWAR, PROF. DR. MOHSEN A. A. RASHWAN – “Using features of local densities, statistics and hmm toolkit”, *Journal of Electrical Systems and Information Technology* (2007), p. 167–192.
- [23] V. G. SAFWAN WSHAH, ZHIXIN SHI – “Segmentation of arabic handwriting based on both contour and skeleton segmentation”, (2009).
- [24] H. SCHAHRAZED – *Segmentation de textes en caracteres pour la reconnaissance optique de l’écriture arabe*, Mémoire, Université El-Hadj Lakhdar Batna, 2007.
- [25] L. C. Y. C. S. C. S.L. CHANG, T. TAIWAN – “Automatic license plate recognition”, **5** (2004), no. 1, p. 42—53.
- [26] M. YOUNES et Y. ABDELLAH – “Segmentation of arabic handwritten text to lines”, *Procedia Computer Science* **73** (2015), p. 115–121.
- [27] F. ZAIZ – “Technique basée puzzle/svm pour l’amélioration de la reconnaissance du texte arabe manuscrit”, Thèse, Université Mohamed Khider – BISKRA, 2017.

