

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جامعة سعد دحلب البليدة
UNIVERSITE SAAD DAHLEB DE BLIDA



كلية الهندسة - دائرة الإلكترونيك
FACULTE DES SCIENCES DE L'INGENIEUR
DEPARTEMENT D'ELECTRONIQUE

**MEMOIRE DE
PROJET DE FIN D'ETUDES
POUR L'OBTENTION DU DIPLOME DE
MASTER EN ELECTRONIQUE**
Option : Théorie de l'information et systèmes électroniques

THÈME

**Segmentation d'image de document
latin imprimé par l'approche
ascendante "BOTTOM-UP"**

Présenté par:

M. BELGROUNE Nadir.

M. ZAMECHA Billel.

Proposé et dirigé par:

Promoteur: M. A.NAMANE

Co-Promoteur: M. M.AREZKI

Promotion: 2010/2011

Remerciements

Nous tenons à remercier en premier lieu dieu le tout puissant qui nous a dotés de toute la force nécessaire à l'aboutissement de ce mémoire.

Nos remerciements vont aussi à notre promoteur le D^r A.NAMANE qui nous a fait bénéficier de ses nombreuses et riches connaissances durant toute la durée de ce travail.

Sans Oublier tous ceux qui ont contribué, de près ou de loin à mener à terme ce travail. Notamment nos enseignants qui nous ont instruits Tout au long de notre long parcours.

Dédicaces

J'ai le grand plaisir de dédier le fruit de mes années d'études aux êtres qui me sont les plus chers, mes parents.

A ma mère qui s'est donnée tant de mal pour moi et qui ma offerte amour et soutien depuis mon premier souffle.

A mon cher père pour avoir mis tous les moyens à ma disposition et d'avoir été à mes cotés tout au long de ces années d'études.

A mes chers frères et ma chère sœur ainsi que toute ma famille.

A mon binôme Billel.

A mes chers amis Hillel, Lyes, Mahdi, Amine, Sidali, Bachir, Zaki, et Amel, et à tous les membres de mon équipe de robotique 'Piratech', ainsi qu'à tous mes amis d'étude.

Et particulièrement Wassila, pour me faire constamment sentir à quel point je suis unique...

Nadir.

Dédicaces

Je dédie ce modeste travail

À Ma très chère et douce mère et Mon très cher père pour l'éducation qu'ils m'ont prodigué, avec tous les moyens et au prix de toutes les sacrifices qu'ils ont consentis à mon égard, pour le sens du devoir qu'ils m'ont enseigné depuis mon enfance.

À mes chères sœurs.

À mes chers frères.

À mon binôme Nadir.

À tous mes amis.

À tout ceux qui ont su m'apporter aide et soutien aux moments propices.

Billel.

الملخص:

في هذا المشروع ، نقتراح أسلوب تجزئة صورة الوثيقة اللاتينية المطبوعة على أساس منهج من أسفل إلى أعلى. يتم تطبيق هذا التحليل على صورة من الوثيقة المسوحة ضوئيا مقياس 4 (A4) في 300 نقطة في البوصة. هذه الوثائق هي من أنواع مختلفة مثل النصوص العلمية، المجالات و سلسلة التمارين.

هذا الأسلوب يهدف إلى تقسيم الصورة الرقمية للوثيقة إلى كتل متماسكة مختلفة مرورا بمرحلتين من التجزئة و هي: تجزئة الأسطر و تجزئة الكتلة. هذه الإنقسامات تستند إلى تقدير المسافات بين أحرف النص و الخطوط .

الكلمات الرئيسية: منهج من الأسفل إلى الأعلى، تجزئة الصورة الرقمية للوثيقة، تحليل الوثيقة المطبوعة.

Résumé :

Dans ce projet, nous proposons une méthode de segmentation d'images de document latin imprimé basée sur une approche ascendante " Bottom-Up". Cette analyse est appliquée à des images de document papier de format A4 scanné sous 300 dpi. Ces documents sont de différents types tels que; les articles scientifiques, les revues et les séries d'exercices.

Cette méthode consiste à répartir l'image numérique du document en différents blocs cohérents en passant par deux étapes de segmentations, notamment, la segmentation en ligne et la segmentation en blocs. Ces deux segmentations nécessitent l'estimation de l'espace inter caractères et l'espace interligne.

Mots clé : méthodes ascendantes (Bottom-up), segmentation d'image de document, analyse de document imprimé.

Abstract:

In this project, we propose an image segmentation method of printed latin document based on a bottom-up approach. This analysis is applied to images of A4 paper document scanned at 300 dpi. These documents are of different types such as, scientific papers, magazines and series of exercises.

This method aims to divide the digital image of a document in different coherent blocks by passing through two stages of segmentation, namely, line segmentation and block segmentation. These two segmentations require the estimation of the spaces between characters and lines text.

Keywords: Ascending methods (Bottom-up), segmentation of image of document, analysis of printed document.

SOMMAIRE

INTRODUCTION GENERALE	2
CHAPITRE I : GENERALITES SUR LE TRAITEMENT D'IMAGE.	4
I. INTRODUCTION :	4
II. NOTIONS FONDAMENTALES :	6
II.2. Définition de l'image :	6
II.2.1. Définition :	6
II.2.2. Connexité :	7
II.2.3. Résolution d'une image :	7
II.2.4. Image en niveaux de gris :	8
II.2.5. Image binaire :	8
II.2.6. Image en couleurs :	9
II.2.7. Bruit dans l'image :	11
II.2.8. Histogramme :	12
II.3. Etapes d'analyse des images :	13
II.4. Quelques techniques du traitement d'images :	13
II.4.1. Modification d'histogramme :	13
II.4.2. Binarisation par seuillage :	15
III. CONCLUSION :	16
CHAPITRE II : PRETRAITEMENT.	17
I. INTRODUCTION :	17
II. FILTRAGE:	18
II.1. Les filtres linéaires :	18
II.1.1. Le filtre moyen:	19
II.1.2. Le filtre gaussien :	20
II.2. Les filtres non linéaires :	21
Le filtre médian :	22
II.3. Les filtres morphologiques :	24
II.3.1. La dilatation :	24
II.3.2. L'érosion :	24
II.3.3. L'ouverture :	24
II.3.4. La fermeture :	24
III. LE SEUILLAGE :	25
III.1. Seuillage global :	25
III.2. Seuillage local :	25
IV. LA BINARISATION :	26
IV.1. Définition :	26
IV.2. Binarisation globale :	26
IV.3. Binarisation locale :	26

IV.4. Méthode d'Otsu :	26
IV.4.1 définition :	26
IV.4.2. Les étapes de l'algorithme :	27
V. CONCLUSION :	30
CHAPITRE III : SEGMENTATION D'IMAGES DE DOCUMENTS.....	56
I. INTRODUCTION :	31
II. DEFINITION :	31
III. LES DIFFERENTES APPROCHES DE LA SEGMENTATION :	32
III.1. Segmentation ascendante :	32
III.1.1. Algorithme RLSA	32
III.1.2. Utilisation d'heuristiques.....	33
III.2 Segmentation descendante :	34
X-Y-Cut.....	35
IV. INTER CARACTERES, INTER MOTS ET INTER LIGNES :	36
IV.1. Segmentation en ligne :	36
IV.2. Segmentation en bloc :	38
V. L'ETIQUETAGE :	39
V.1. Définition :	39
V.2. Etiquetage parallèle de régions :	39
V.3. Etiquetage séquentielle de régions :	39
V.4 Algorithme d'étiquetage :	39
VI. CONCLUSION :	44
CHAPITRE IV :.....	56
IMPLEMENTATION ET RESULTATS.....	56
I.INTRODUCTION :	45
II.PRESENTATION DES LANGAGES DE PROGRAMMATION UTILISES :...45	
II.1. Le langage MATLAB R2009a :	45
II.2. Le langage Builder C++ 5 :	46
III. FONCTIONNEMENT :	48
IV.DESCRPTION D'ESPACE DE TRAVAIL :	49
IV.1. Menu principale :	50
IV.1.1. Le menu fichier :	50
IV.1.2. Le menu prétraitement :	51
IV.1.3. Le menu fenêtre :	54
IV.1.4. Le menu affichage :	55
IV.1.5. Le menu analyse :	56
IV.1.6. Le menu aide ? :	56
IV.2. La barre d'outils :	58
IV.2.1. Les boutons :	58
IV.2.2. Le groupe Box :	59
IV.2.3. Les édit :	59

V. RESULTATS ET DISCUTION :	60
V.1. Résultats du filtrage :	60
V.2. Résultats du binarisation :	61
V.3. Résultats de la segmentation en ligne:	62
V.4. Résultats de la segmentation en blocs :	63
V.4. Résultats d'étiquetage :	65
VI. CONCLUSION :	66
CONCLUSION GENERALE :	67
REFERENCES BIBLIOGRAPHIQUES	68
ANNEXE	71

LISTE DES FIGURE

Figure I. 1. <i>Relations du traitement d'images avec d'autres disciplines.</i>	5
Figure I. 2. <i>Les Lettre A et B écrites avec un groupement de pixel.</i>	6
Figure I. 3. <i>Voisinage d'un pixel Contraste.</i>	7
Figure I. 4. <i>Valeurs des niveaux de gris et teintes correspondantes.</i>	8
Figure I. 5. <i>Représentation en variant des couleurs (24 bits).</i>	10
Figure I. 6. <i>Exemple d'une palette de couleurs.</i>	10
Figure I. 7. <i>Image avant et après le bruit.</i>	11
Figure I. 8. <i>Une image et son histogramme.</i>	12
Figure I. 9. <i>Principe de la modification d'histogramme.</i>	14
Figure I. 10. <i>Choix du seuil de binarisation sur l'histogramme.</i>	15
Figure II. 1. <i>Schéma synoptique de procédure de prétraitement.</i>	17
Figure II. 2. <i>Filtre moyenne de différente taille.</i>	19
Figure II. 3. <i>Image d'une gaussienne</i>	20
Figure II. 4. <i>Fonction d'une gaussienne en 3D.</i>	20
Figure II. 5. <i>Exemple du filtre gaussien pour les images.</i>	21
Figure II. 6. <i>Principe de fonctionnement d'un filtre médian.</i>	22
Figure II. 7. <i>Filtre médian de différente taille.</i>	23
Figure II. 8. <i>Image binarisée par la méthode d'Otsu.</i>	27
Figure II. 9. <i>Image binarisée par Otsu.</i>	29
Figure III. 1. <i>Image originale</i>	33
Figure III. 2. <i>Image segmenté avec RLSA.</i>	33
Figure III. 3. <i>Exemple de document à problème.</i>	35
Figure III. 4. <i>Estimation et élimination de l'espace inter-caractères.</i>	36
Figure III. 5. <i>Organigramme qui explique le calcul de la distance entre les caractères et les mots.</i>	37
Figure III. 6. <i>Organigramme qui explique l'élimination des espaces entre les caractères et les mots.</i>	37
Figure III. 7. <i>Estimation et élimination de l'espace interlignes.</i>	38
Figure III. 8. <i>Exemple de l'Etiquetage.</i>	39
Figure III. 9. <i>L'image binaire à étiqueté</i>	40
Figure III. 10. <i>L'image étiquetée</i>	40
Figure III. 11. <i>L'image binaire à étiqueté</i>	40
Figure III. 12. <i>L'image pointée sur le pixel (1,1).</i>	40
Figure III. 13. <i>L'image pointée sur le pixel (1,4), crée une nouvelle étiquette.</i>	41
Figure III. 14. <i>L'image après le balayage de la première ligne.</i>	41
Figure III. 15. <i>L'image après le balayage de la deuxième ligne.</i>	42
Figure III. 16. <i>L'image au pixel (3,4.)</i>	42
Figure III. 17. <i>L'image après la quatrième ligne</i>	42
Figure III. 18. <i>L'image après la ligne 5</i>	42
Figure III. 19. <i>L'image après la ligne 6.</i>	42
Figure III. 20. <i>L'image après la 7</i>	42
Figure III. 21. <i>L'image après la ligne 8.</i>	42
Figure III. 22. <i>Le nettoyage de l'image étiqueté</i>	43
Figure III. 23. <i>L'image nettoyée à l'étiquette 1.</i>	43

Figure III. 24. <i>L'image nettoyée jusqu'à la ligne 7.</i>	43
Figure III. 25. <i>Résultat final d'étiquetage.</i>	44
Figure IV. 1. <i>Le langage MATLAB.</i>	46
Figure IV. 2. <i>Le programme avec le langage MATLAB.</i>	46
Figure IV. 3. <i>Le langage Builder C++ 5.</i>	47
Figure IV. 4. <i>L'image numérique de document imprimé.</i>	48
Figure IV. 5. <i>L'espace de travail de notre application.</i>	49
Figure IV. 6. <i>Le menu principal.</i>	50
Figure IV. 7. <i>Le menu fichier.</i>	50
Figure IV. 8. <i>Le menu prétraitement.</i>	51
Figure IV. 9. <i>Filtrage Gaussien de l'image.</i>	51
Figure IV. 10. <i>Binarisation de l'image par la méthode d'Otsu.</i>	52
Figure IV. 11. <i>Menu soudage.</i>	52
Figure IV. 12. <i>Fusion horizontale de l'image.</i>	53
Figure IV. 13. <i>Fusion verticale de l'image.</i>	54
Figure IV. 14. <i>Menu fenêtre.</i>	54
Figure IV. 15. <i>Menu affichage.</i>	55
Figure IV. 16. <i>La rotation de l'image.</i>	55
Figure IV. 17. <i>Menu analyse.</i>	56
Figure IV. 18. <i>Les résultats obtenus avec une image qui possède deux étiquettes.</i>	56
Figure IV. 19. <i>Menu aide ?</i>	56
Figure IV. 20. <i>La boîte de dialogue A propos.</i>	57
Figure IV. 21. <i>La barre d'outils.</i>	58
Figure IV. 22. <i>Groupe box pour visualiser les différents paramètres résultants.</i>	59
Figure IV. 23. <i>Exemple de visualisation des paramètres.</i>	59
Figure IV. 24. <i>Édit pour afficher le nombre d'étiquettes de l'image segmentée.</i>	59
Figure IV. 25. <i>Les résultats de filtrage.</i>	60
Figure IV. 26. <i>L'image après binarisation</i>	61
Figure IV. 27. <i>Résultat de la segmentation en ligne.</i>	62
Figure IV. 28. <i>Exemple d'un article scientifique ou il y a que des textes.</i>	63
Figure IV. 29. <i>Article scientifique avec des images à l'intérieur.</i>	63
Figure IV. 30. <i>Image texte simple.</i>	64
Figure IV. 31. <i>Image qui contient des tableaux.</i>	64
Figure IV. 32. <i>Image qui contient des graphes.</i>	65
Figure IV. 33. <i>Image étiquetée avec MATLAB.</i>	65

Liste des Annexe

Annexe. 1. Différentes techniques de binarisation	71
Annexe. 2. Tableau récapitulatif des approches ascendantes.....	73
Annexe. 3. Tableau récapitulatif des différentes approches descendantes.....	74

INTRODUCTION GENERALE :

INTRODUCTION GENERALE

L'analyse automatique de documents est un champ vaste d'applications potentielles compte tenu de la quantité énorme de documents produits par l'activité humaine. L'utilisation effective de système automatique d'analyse de documents dépend évidemment des capacités à localiser les zones informatives et à les reconnaître.

Les résultats obtenus aujourd'hui en analyse d'image de document ne sont satisfaisants qu'au prix de conditions très restrictives si l'on considère la très grande variété des documents existants, comme les articles scientifiques, les revues, les séries d'exercices, les journaux,etc.

En particulier, le texte doit être en caractères imprimés. Il doit être de bonne qualité et disposé régulièrement sur un fond uniforme. Or beaucoup d'applications potentielles sortent de ces limites : orientations variables des lignes, fond incontrôlé, écriture manuscrite, mélange de textes, de graphiques, de photographies. C'est le cas des documents techniques, des cartes et plans, des magazines où la mise en page peut être très complexe, ou encore des documents postaux où l'adresse peut n'être qu'une faible fraction de l'information présente.

Dès lors une étape essentielle d'un système complet d'analyse, en amont de la lecture elle-même, consiste à localiser et extraire le plus correctement possible toutes les entités à traiter par le système de reconnaissance.

Les approches utilisées jusqu'à aujourd'hui pour traiter ce type de problèmes consistaient à binariser dès l'acquisition d'image du document en utilisant un algorithme de binarisation. Suivant le type de document et la méthode employée [1], le résultat n'est pas toujours bon.

A partir de l'image binaire obtenue, des méthodes de segmentation sont alors mises en œuvre.

Deux types d'approches sont classiquement rencontrés. Le premier est intéressant lorsque des connaissances sur la structure du document sont disponibles. C'est le cas des méthodes dites descendantes dont l'algorithme RXYC (Réursive XY Cut) basé sur des profils de projection est l'une des références en la matière [2].

La deuxième catégorie d'approche est de type ascendant, est plutôt guidée par les données : elle procède par regroupements progressifs d'entités élémentaires partant du

pixel pour construire les structures physiques du document. Une des grosses difficultés est de savoir alors définir les règles permettant de contrôler les fusions [3].

De plus, la plupart des méthodes évoquées ci-dessus sont destinées à traiter des documents imprimés. C'est dans ce cadre que se situent les travaux présentés dans notre projet. Il s'agit donc de concevoir une application sous Builder C++ 5 pour la segmentation d'images de documents imprimés.

A fin de décrire et mettre en évidence les différents étapes suivies lors de notre projet, nous avons adopté la répartition suivante pour la rédaction de notre mémoire :

Chapitre I présente des généralités sur le monde de traitement d'image.

Chapitre II aura pour but d'introduire la phase de prétraitement. En commençant par le filtrage, la binarisation et en suite la phase de la segmentation.

Chapitre III aura pour but de détailler la phase de la segmentation d'images de documents.

Chapitre IV sera consacré à notre application, et montre comment les techniques présentées dans le chapitre II peuvent être appliquées réellement. Ce chapitre présente aussi l'approche proposée sur des images de documents papier de format A4 scannés sous 300 dpi.

CHAPITRE I :

Généralités sur le traitement
d'image.

I. INTRODUCTION :

Le traitement d'images est l'ensemble des méthodes qui permettent de rendre une image plus intelligible pour l'être humain ou plus facile à manipuler par la machine, en mettant en exergue certaines de ses caractéristiques, en vue d'une application donnée.

C'est un domaine qui fait appel à plusieurs disciplines : La théorie du signal, la théorie des systèmes, l'analyse numérique, les statistiques, la théorie de l'information, la neurophysiologie et la psychophysique, l'optique, l'électronique et l'informatique pour ne citer que les principales (figure I.1).

Les différentes relations avec des domaines variés engendrent une grande diversité dans les traitements que l'on peut effectuer sur des images. Pour illustrer cette diversité, on peut citer quelques exemples :

- ❖ Rendre nette une image floue.
- ❖ Améliorer le contraste d'une image.
- ❖ Filtrer des parasites sur une image
- ❖ Comprimer le nombre d'échantillons d'une image numérique par un facteur de 100 ou plus et d'une séquence d'images (TV numérique) par un facteur 1000 ou plus.
- ❖ Reconnaître un texte imprimé, magazine, journal ; etc.....
- ❖ Reconnaître et compter des bactéries dans des préparations microscopiques.
- ❖ Reconnaître des visages, empreintes digitales et des signatures.
- ❖ Aider les pilotes et les conducteurs à s'entraîner.
- ❖ Diriger un véhicule sans chauffeur.
- ❖ Surveiller les machines-outils.
- ❖ Aider les robots mécaniques à voir ; etc.....

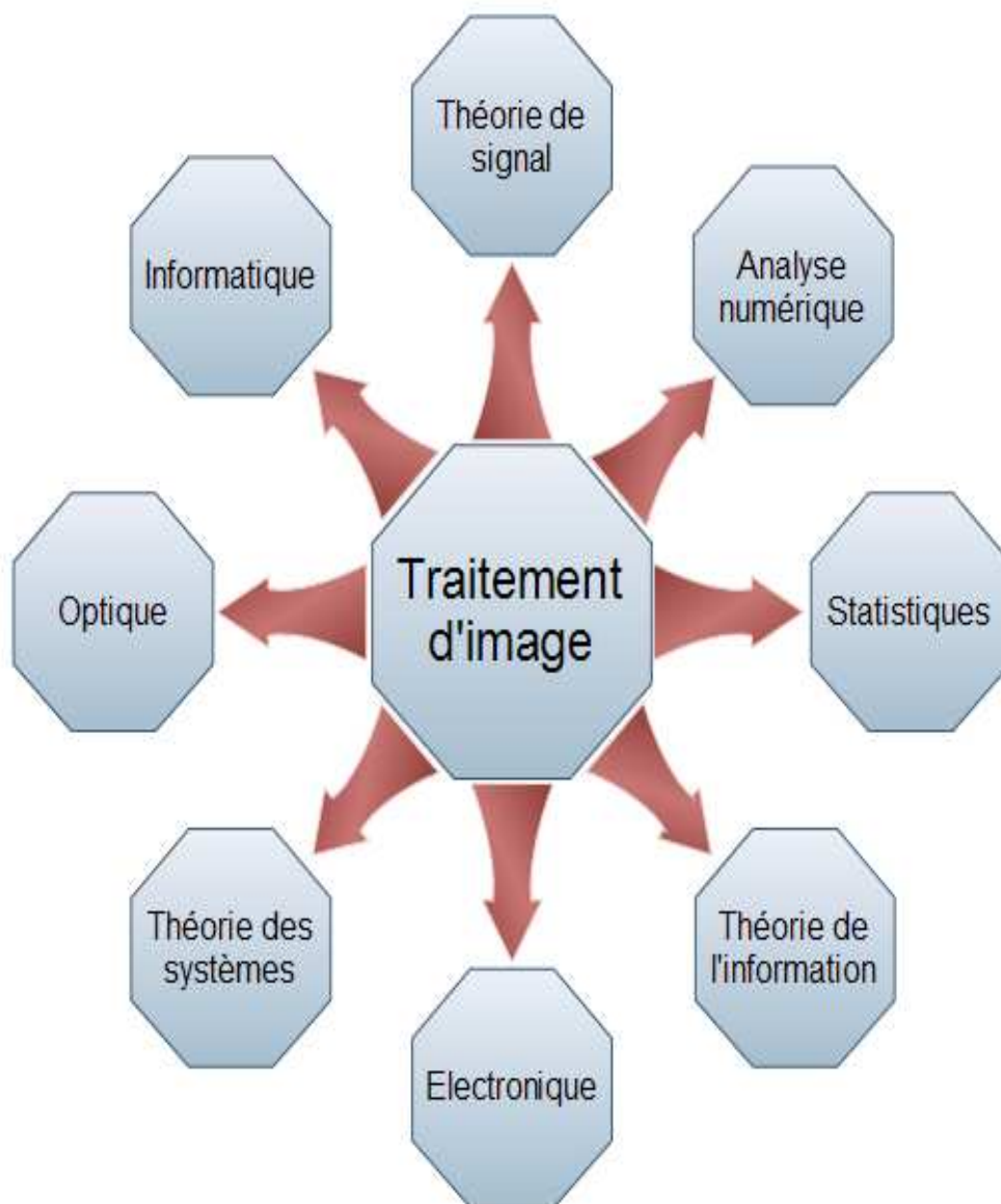


Figure I. 1. Relations du traitement d'images avec d'autres disciplines.

Il faut noter aussi que la principale difficulté dans la réalisation d'une application quelconque de traitement d'images est l'absence d'un traitement idéal universel qui résoudrait plusieurs problèmes à la fois. Ainsi, chaque application particulière nécessite une étude en soi [4].

Les algorithmes de traitement d'images impliquent des volumes de calcul assez importants. L'essor de l'implémentation de ces algorithmes est fortement lié à l'avènement sur le marché des processeurs de traitement du signal (DSP, acronyme de Digital Signal Processor) et des circuits FPGA (Field Programmable Gate Array).

II. NOTIONS FONDAMENTALES :

II.2. Définition de l'image :

II.2.1. Définition :

Une image est une représentation numérique bidimensionnelle d'une scène analogique réelle située en général dans un espace tridimensionnel. Elle possède l'information pour chaque point de son intensité lumineuse fournie par les capteurs (appareils photo, cameras, scanner).

On peut la décrire par une fonction $f(x, y)$ où :

f : est la fonction d'intensité lumineuse définie dans un domaine borné.

x, y : coordonnées cartésiennes d'un point de l'image.

$f(x, y)$: niveau de gris en ce point [5].

Le pixel (contraction de « Picture élément ») est l'unité de base constituant l'image, si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression [5].

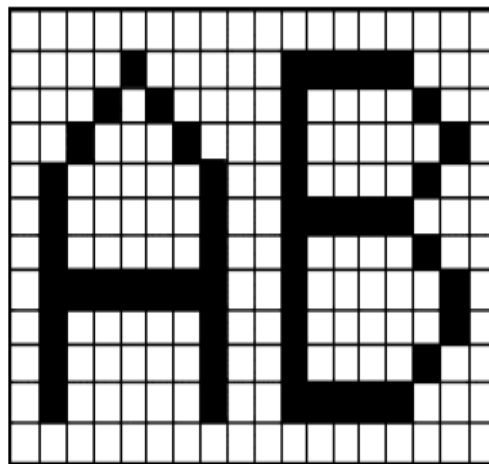


Figure I. 2. Les Lettre A et B écrites avec un groupement de pixel.

II.2.2. Connexité :

La connexité entre les pixels d'une image est un concept très important utilisé surtout dans l'établissement des frontières des objets et l'identification des composants d'une région dans une image.

Pour établir si deux pixels sont connectés, nous devons déterminer s'ils sont adjacents quelques parts (par exemple, s'ils appartiennent à un 4-voisinage) et si leurs niveaux de gris respectent un certain critère de similarité [5].

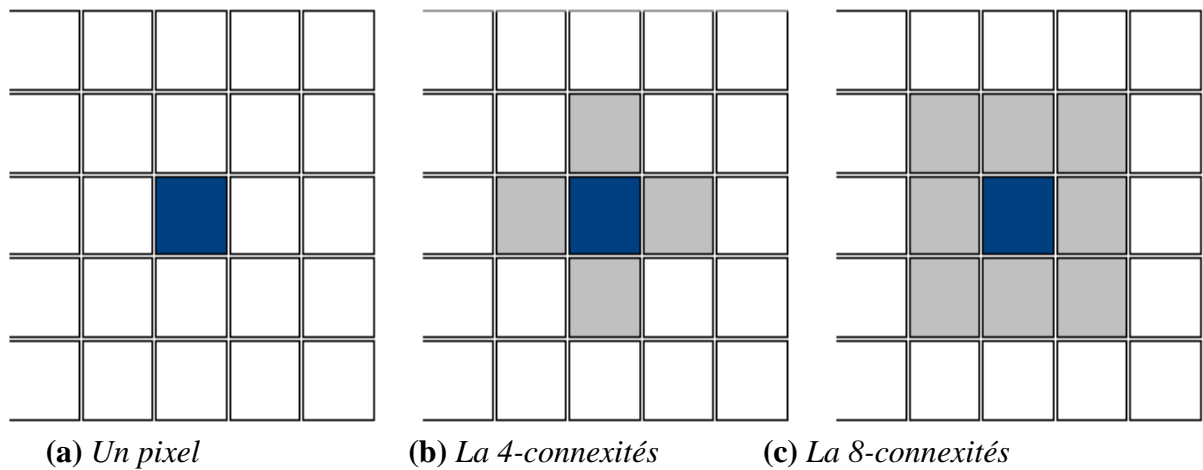


Figure I. 3. Voisinage d'un pixel Contraste.

C'est l'opposition marquée entre deux régions juxtaposées d'une image, plus précisément les régions sombres et les régions claires de cette image [6]. Il est défini en fonction des luminances L_i et L_{i+1} de deux zones de l'image par l'équation :

$$C = \frac{L_i - L_{i+1}}{L_i + L_{i+1}} \dots \dots \dots (1)$$

II.2.3. Résolution d'une image :

La résolution d'une image est définie par un nombre de pixels par unité de longueur de la structure à numériser (classiquement en dpi (dots per inches) ou ppp (points par pouce)).

Ce paramètre est défini lors de la numérisation et dépend principalement des caractéristiques du matériel utilisé lors du processus de numérisation. Plus le nombre

de pixels est élevé par unité de longueur de la structure à numériser, plus la quantité d'information qui décrit cette structure est importante et plus la résolution est élevée[6].

II.2.4. Image en niveaux de gris :

Dans une image en niveaux de gris, la couleur d'un pixel peut prendre des valeurs allant du noir au blanc, en passant par un nombre fini de niveaux intermédiaires.

En général, les images en niveaux de gris renferment 256 teintes de gris. Par convention la valeur zéro représente le noir (intensité lumineuse nulle) et la valeur 255 le blanc (intensité lumineuse maximale). Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la « couleur » de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux [7].

0... 20... 40... 50... 80... 100... 120... 140... 160... 180... 200... 220... 240... 255



Figure I. 4. Valeurs des niveaux de gris et teintes correspondantes.

II.2.5. Image binaire :

Ces images sont constituées de pixels qui ne peuvent avoir que deux états : noir ou blanc. On les appelle des images « au trait ». On trouve parmi elles des pages de texte, des signatures, des plans, des dessins, ...etc. [8].

Il existe de nombreux type d'équipement de reproduction d'image qui ne sont capables de reproduire que deux niveaux de gris dans les cas d'image monochrome et 3 teintes dans le cas d'image en couleurs.

Toutes ces techniques sont basées sur le principe du seuillage comme le montre :

$$I_2(X, Y) = \begin{cases} 0 & \text{si } I_n(X, y) < T \\ 1 & \text{si } I_n(X, y) > T \end{cases} \dots\dots\dots (2)$$

I_n : Intensité à n niveaux de chaque point d'image.

I_2 : représente l'intensité à 2 niveaux correspondante.

T : le seuil de binarisation.

II.2.6. Image en couleurs :

Une image en couleurs est censée représenter le mieux possible la réalité. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités.

En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs, . . .) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.) [7].

➤ **Représentation en couleurs réelles (Images en "vraies couleurs" (ou 24 bits)) :**

Elle consiste à utiliser 24 bits pour chaque point de l'image (figure I.5). Huit bits sont employés pour décrire la composante rouge (R), huit pour le vert (V) et huit pour le bleu (B), ce qui fait que les images en vraies couleurs sont des images très « lourdes ». Il est ainsi possible de représenter environ 16,7 millions de couleurs différentes simultanément. Par ailleurs l'œil humain n'est pas capable de distinguer autant de couleurs.



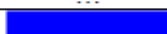



Valeur R	Valeur V	Valeur B	Couleur correspondante	Commentaire
0	0	0		Noir
0	0	1		Un peu moins noir (nuance impossible à discerner du noir par l'œil humain)
...
0	0	255		Bleu
...
0	255	0		Vert
...
255	0	0		Rouge
...
128	128	128		Couleur intermédiaire (gris)
255	255	255		blanc

Figure I. 5. Représentation en variant des couleurs (24 bits).

➤ **Représentation en couleurs indexées :**

Afin de diminuer la charge de travail nécessaire pour manipuler des images en 24 bits, on peut utiliser le mode de représentation en couleurs indexées. Le principe consiste à déterminer le nombre de couleurs différentes utilisées dans l'image, puis à créer une table de ces couleurs en attribuant à chacune une valeur numérique correspondant à sa position dans la table. La table, appelée palette, comporte également la description de chacune des couleurs, sur 24 bits.

➤ **Autres modèles de représentation :**

Le modèle R.V.B. représentant toutes les couleurs par l'addition de trois composantes fondamentales, n'est pas le seul possible. Il en existe de nombreux autres. L'un d'eux est particulièrement important. Il consiste à séparer les informations de couleurs (chrominance) et les informations d'intensité lumineuse (luminance).







Palette avec les codes RVB				Couleur correspondante
Couleur	Code RVB			
0	255	255	255	
1	255	255	204	
2	255	255	153	
3	255	255	102	
4	255	255	51	
5	255	255	0	
...

Figure I. 6. Exemple d'une palette de couleurs.

II.2.7. Bruit dans l'image :

Le bruit dans une image est défini comme étant un phénomène de brusque variation d'un pixel isolé par rapport à ses voisins, il affecte la qualité de l'image et il provient soit du dispositif d'acquisition (scanner, caméra, amplification, quantification, . . .) soit de la scène elle-même (poussières, rayures, . . .) [9].



Figure I. 7. Image avant et après le bruit.

Le bruit est une perturbation qui peut affecter une image provenant :

➤ **Des matériels de saisie :**

Il y a des parasites qui troublent la réception des signaux suite à un mauvais réglage du capteur. En vision par exemple, un mauvais réglage de l'objectif de la caméra peut conduire à des erreurs de parallaxe ou de flou ; un mauvais calibrage de la caméra peut entraîner des erreurs de mesure et d'interprétation des données.

Pour résumer, la mauvaise qualité de capteur et quand il est mal adapté à l'acquisition de la forme.

➤ **De l'environnement :**

Le mauvais éclairage, le positionnement des objets à saisir, etc. par exemple, un défaut d'éclairage dans la saisie de caractère peut donner lieu à une image incomplète où l'information peut manquer des endroits spécifiques de la forme.

Une technique appelée *filtrage logique* consiste à combler les petits trous dans la matrice du caractère en examinant les huit voisins d'un point noir.

➤ *De l'objet lui-même :*

Qui peut être considéré comme un cas pathologique parce qu'il ne présente pas les mêmes caractéristiques que l'ensemble des autres de la même famille.

II.2.8. Histogramme :

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image.

Pour diminuer l'erreur de quantification ou pour comparer deux images obtenues sous des éclairages différents ou encore pour mesurer certaines propriétés de l'image, on modifie souvent l'histogramme correspondant.

L'histogramme permet de donner un grand nombre d'informations sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée (figure I.8).

L'histogramme peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci [10].

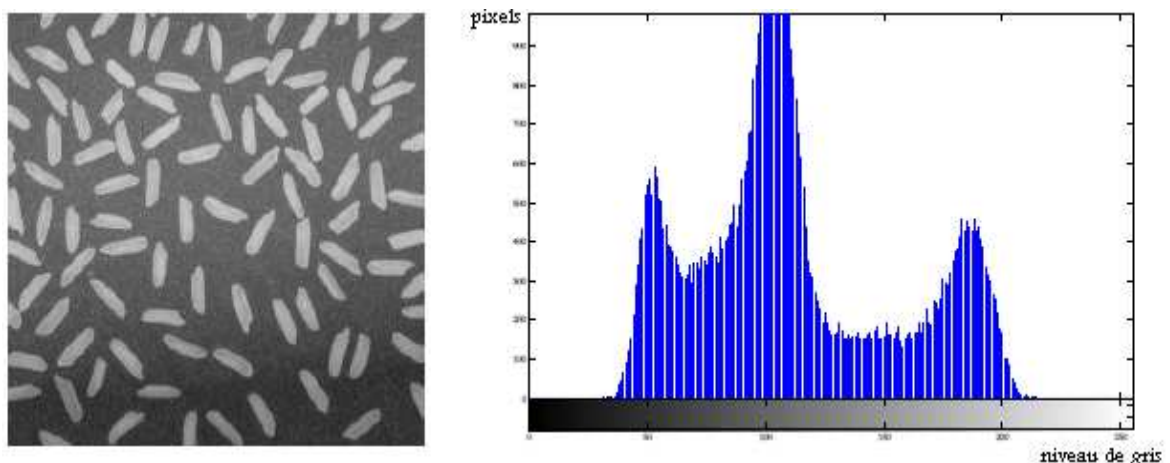


Figure I. 8. Une image et son histogramme.

II.3. Etapes d'analyse des images :

La plupart des applications du traitement d'images passent par les étapes suivantes :

- ❖ Acquisition : Echantillonnage, Quantification.
- ❖ Analyse globale de l'image et transformations ponctuelles : Histogramme, Statistiques (moyenne, écart-type, ...etc.), transcodage (palette de couleur [LUT]) et classification.
- ❖ Opérations entre images : Indices, ratio, différences, opérations logiques, masques et seuillage.
- ❖ Amélioration, filtrage et segmentation : Opérations de convolution (Lissage, rehaussement, détection de contours) ; squelettisation, vectorisation.
- ❖ Interprétation et sémantique : Cartographie thématique (Classification automatique et supervisée), cartographie vecteur, représentation des graphes et de la topologie.

II.4. Quelques techniques du traitement d'images :

Avant d'extraire les objets et d'analyser une image, il est souvent nécessaire d'améliorer sa qualité en lui appliquant quelques traitements (modification d'histogramme, binarisation, amélioration du contraste, élimination de bruits, ...etc.).

Nous allons présenter dans ce qui suit quelques techniques de base.

II.4.1. Modification d'histogramme :

Certaines images sont initialement trop claires, foncées ou bien peu contrastées, cela est dû respectivement au fait que les niveaux de gris de l'image sont tassés vers le haut de l'échelle, vers le bas ou bien sont regroupés dans un intervalle étroit. Ces défauts sont très visibles sur l'histogramme. Le but est de redistribuer les niveaux de gris de l'image afin de leur faire occuper toute la bande de nuances possibles. Le principe est donné sur la figure I.9.

Cette méthode est basée sur les transformations ponctuelles d'intensité. C'est-à-dire, à chaque pixel d'intensité I_s on associe une intensité $I'_s = T(I_s)$. [7]

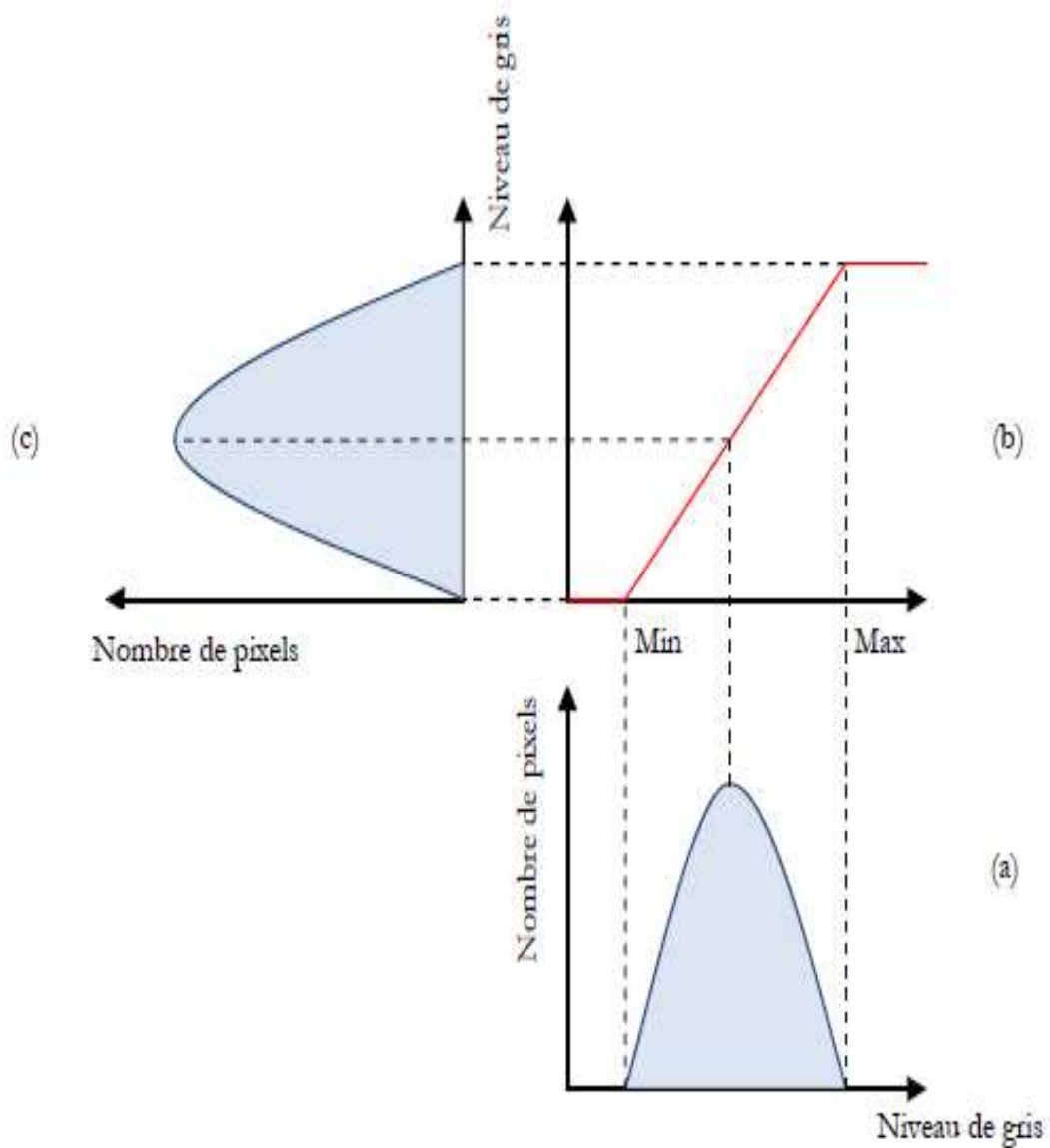


Figure I. 9. Principe de la modification d'histogramme.

(a) Histogramme de l'image originale.

(b) Fonction de transformation.

(c) Histogramme recadré.

II.4.2. Binarisation par seuillage :

Cette technique consiste à réduire la dynamique d'une image (plusieurs niveaux de gris) à deux niveaux (noir et blanc). Ceci revient à séparer les pixels de l'image en deux classes, la première ayant un niveau maximal (typiquement 255) et la seconde un niveau minimal (0).

La méthode consiste à calculer un seuil d'après l'histogramme de l'image, ce seuil doit être le Plus adéquat possible pour ne pas altérer les informations pertinentes de l'image (figure I.10) [7].

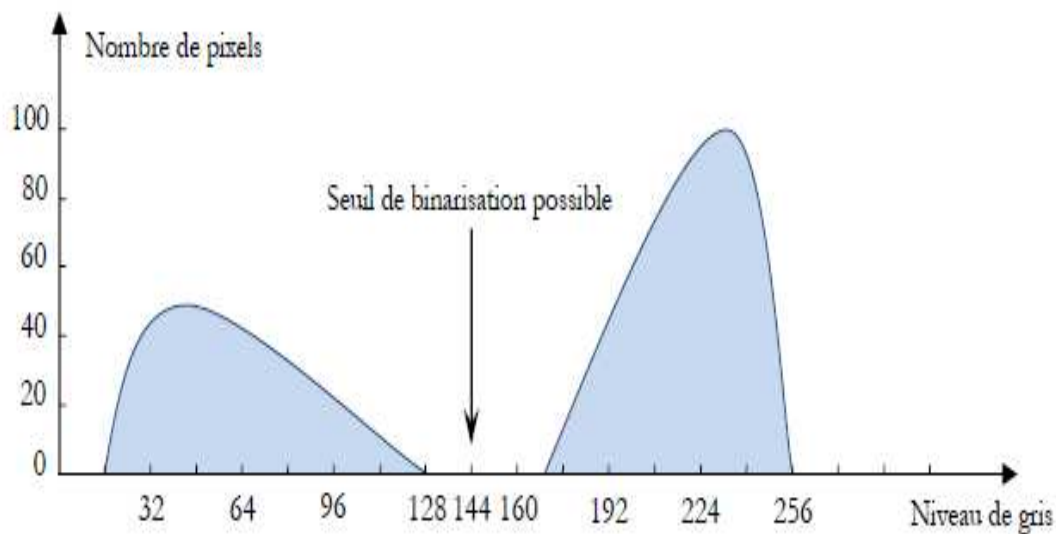


Figure I. 10. Choix du seuil de binarisation sur l'histogramme.

III. CONCLUSION :

On a vu dans ce chapitre les différentes notions de base du traitement d'image et aussi quelque méthode utilisée en segmentation de document plus précisément les approches ascendantes et les approches descendantes.

Le chapitre suivant décrit les étapes du prétraitement d'une image numérique.

CHAPITRE II :

Prétraitement.

I. INTRODUCTION :

Le rôle du prétraitement est de «préparer» les documents reçus du scanner à la phase suivante d'analyse consacrée à la segmentation de document. Cette phase n'est possible et surtout fiable que si les documents du scanner possèdent de bruit, et corrigées de leurs erreurs éventuelles.

Toutes les techniques élaborées dans ce sens se gardent de modifier les propriétés essentielles des formes, ce qui pourrait conduire, dans le cas contraire, à de graves erreurs d'analyse.

Dans la suite, nous allons présenter les techniques nécessaires pour préparer l'image à la classification.

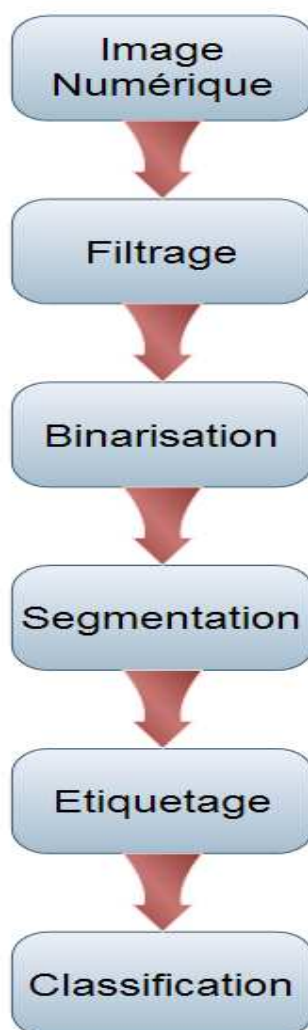


Figure II. 1. Schéma synoptique de procédure de prétraitement.

II. FILTRAGE:

Le but du filtrage est essentiellement l'atténuation du bruit et l'accentuation des discontinuités [24].

On peut citer, entre autre, les classes de filtres suivants :

- ❖ Les filtres linéaires.
- ❖ Les filtres non linéaires.
- ❖ Les filtres morphologiques.

II.1. Les filtres linéaires :

Ce type de filtrage est caractérisé par une opération de convolution de sorte que la transformation de chaque pixel est le fruit d'une combinaison linéaire des pixels voisins [24].

$$I'(i,j) = I(i,j) * \text{filtre}(i,j) \dots\dots\dots(3)$$

$$I'(i,j) = \sum_u \sum_v I(i-u, j-v) \cdot \text{filtre}(i,j) \dots\dots\dots(4)$$

$$A = 1/\sum X_M * \begin{bmatrix} X & X & X \\ X & X & X \\ X & X & X \end{bmatrix} \dots\dots\dots (5)$$

M : la taille de la matrice A.

A : masque de convolution.

X : les coefficients de la matrice A, qui varient d'un filtre à un autre.

Remarque :

- Pour obtenir une image filtrée dont la valeur des pixels varie entre 0 et 255, il faut diviser tous les coefficients du masque de convolution par leur somme.
- Il faut choisir une taille impaire pour que le masque de pixel à traiter soit au centre.

II.1.1. Le filtre moyen:

Ce filtre considère chaque pixel d'image et fait la moyenne avec les pixels voisins en utilisant un masque. On obtient une image adoucie en réduisant les variations brusques de niveaux de gris [24].

La taille du masque est un paramètre variable. Plus le masque sera de grande dimension, plus l'effet du filtrage est apparent.



Image originale



filtre de moyenne : taille 3



filtre de moyenne : taille 5



filtre de moyenne : taille 7

Figure II. 2. *Filtre moyenne de différente taille.*

On utilise le masque suivant :

$$H = \frac{1}{N} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \dots\dots\dots (6)$$

Où : N est un coefficient de normalisation =somme des coefficients non nuls.

Chaque pixel est multiplié par le coefficient correspondant, les masques les plus courants Sont :

$$H_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, H_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, H_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \dots\dots\dots(7)$$

On déplace le masque sur toute l'image et le pixel affecté par la transformation sera le pixel central du masque [25].

II.1.2. Le filtre gaussien :

Pour appliquer ce filtre, il suffit d'une convolution entre l'image initiale et une Gaussienne G(x; y; σ) à deux dimensions données par la formule :

$$G(x; y; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left(-\frac{x^2+y^2}{2\sigma^2} \right) \dots\dots\dots(8)$$

Ce filtre effectue la moyenne des pixels voisins avec une pondération par des coefficients discrétisés d'une gaussienne.

Il a l'avantage de régler le degré de filtrage par le paramètre σ [7].

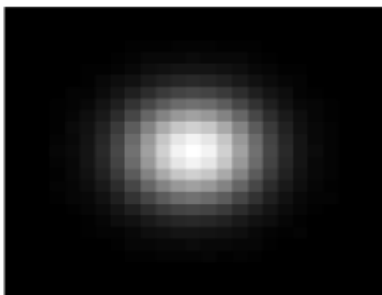


Figure II. 3. Image d'une gaussienne.

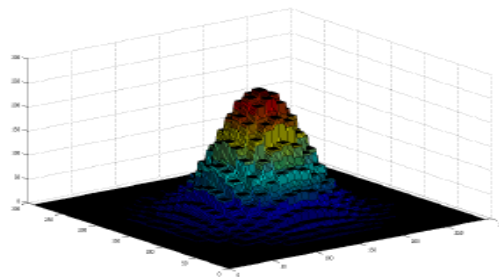


Figure II. 4. Fonction d'une gaussienne en 3D.



Image originale



Gaussien 3x3



Gaussien 5x5



Gaussien 7x7

Figure II. 5. Exemple du filtre gaussien pour les images.

II.2. Les filtres non linéaires :

Ces opérateurs ont été développés pour pallier les insuffisances des filtres linéaires, surtout la mauvaise conservation des contours.

Les pixels voisins interviennent suivant une loi non linéaire dans la détermination du pixel central. Un des exemples de ces filtres est le filtre médian.

Le filtre médian :

Le niveau de gris du pixel central est remplacé par la valeur médiane de tous les pixels de la fenêtre d'analyse centrée sur le pixel. La taille du masque dépend de la variance du bruit et de la taille des détails significatifs de l'image traitée.

La figure II.6 illustre le fonctionnement d'un filtre médian de fenêtre d'analyse 3×3 sur un exemple. Ici le pixel central ayant le niveau de gris (12) aura le niveau de gris (15).

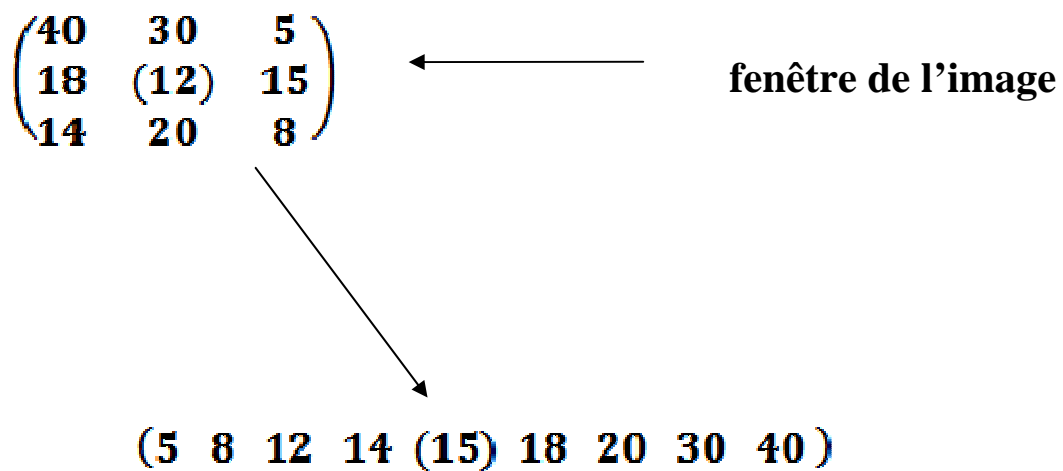


Figure II. 6.Principe de fonctionnement d'un filtre médian.

Le filtre médian garde la netteté de l'image pour les éléments de dimensions importantes par rapport au noyau du filtre, mais élimine les détails fins de manière irrémédiable [7].



Image originale



Filtre médian : taille 3



Filtre médian : taille 5



Filtre médian : taille 7

Figure II. 7. *Filtre médian de différente taille.*

II.3. Les filtres morphologiques :

Les filtres morphologiques sont souvent utilisés pour éliminer des pixels isolés considérés comme un bruit dans une image binarisée. Ces méthodes utilisent un élément structurant.

Parmi ces opérations morphologiques il y a la dilatation, l'érosion, l'ouverture et la fermeture mathématiques.

II.3.1. La dilatation :

Elle permet d'éliminer les pixels blancs isolés mais ajoute des pixels noirs au contour des objets présents dans l'image. Le résultat de cette opération est l'augmentation de la taille de ces objets.

II.3.2. L'érosion :

Elle permet d'éliminer les pixels noirs isolés au milieu des parties blanches de l'image. Le résultat de cette opération est la diminution de la taille des objets présents dans l'image.

II.3.3. L'ouverture :

L'ouverture est constituée par une opération d'érosion suivie d'une dilatation. Elle permet de retrouver les taches noires dans l'image.

II.3.4. La fermeture :

La fermeture est l'opération inverse de l'ouverture, qui consiste à faire subir à l'image une dilatation suivie d'une érosion. Elle permet d'éliminer les blancs qui se trouvent dans l'objet [9].

III. LE SEUILLAGE :

Le seuil du seuillage ou binarisation consiste à transformer l'image codée sur 6, 8 ou 16 bits, en une image binaire créée sur 1 bit.

Le seuillage permet de sélectionner les parties de l'image qui intéressent l'opérateur, par exemple 2 types de grains (blancs et gris) dans un mélange. On peut donc, par exemple, attribuer à tous les pixels de l'image numérique qui ont un niveau de gris compris entre deux valeurs i_1 et i_2 , choisies par l'opérateur, la valeur 1; à tous les autres pixels est attribuée la valeur 0.

$$I_b(i, j) = \begin{cases} 1 & \text{si } I_b(i, j) \geq s \dots\dots\dots(9) \\ 0 & \text{si non} \end{cases}$$

Après seuillage, les parties des images sélectionnées seront traduites en noir et blanc.

III.1. Seuillage global :

La première solution pour fixer le seuil est de procéder par tâtonnement à partir de l'image en noir et blanc résultante censée mettre en évidence les contours. Le seuil correspond à une valeur réelle dans l'intervalle [0,1]. Au préalable, il est nécessaire de normaliser l'image du module du gradient afin que tous les pixels se trouvent également dans l'intervalle [0, 1]. Cette méthode est très simple mais peu efficace [26].

III.2. Seuillage local :

Dans cette technique de seuillage contrairement aux précédentes, le traitement n'est pas identique en tout point de l'image. On s'intéresse ici aux pixels avoisinant les contours les plus significatifs de l'image. L'idée est de garder les contours les plus forts de l'image mais en essayant d'assurer leur continuité. Deux seuils sont nécessaires pour implanter la technique : un seuil haut S_h et un seuil bas S_b . Le seuil haut va servir à sélectionner les contours les plus significatifs dans l'image du module du gradient. Ces contours sont contenus dans l'image résultante en noir et blanc. Le seuil bas permet de mettre en évidence des contours moins forts de l'image [27]. Ces contours sont conservés dans l'image résultante seulement s'ils sont situés dans le voisinage des contours les plus significatifs mis en évidence par le seuillage avec S_h . Généralement le voisinage est défini par les huit voisins.

IV. LA BINARISATION :

Le but de la binarisation d'une image est d'affecter un niveau uniforme aux pixels pertinents et d'éliminer les autres.

IV.1. Définition :

La binarisation est une technique où les pixels de l'image sont partagés par un seul seuil s en deux classes : ceux qui appartiennent au fond et ceux qui appartiennent à l'objet. L'image est alors séparée en deux classes de façon à ce que l'information comprise entre 0 et s est retenue et l'autre non, ou vice-versa, voir Annexe 1.

IV.2. Binarisation globale :

Le seuil est calculé de façon à être unique pour tous pixels de l'image. Cette méthode utilise un seul seuil, parmi ces méthodes on trouve la méthode d'Otsu qui est basée sur la variance inter-classe.

IV.3. Binarisation locale :

Dans ce cas le seuil est différent d'un pixel à l'autre, et dépend essentiellement du voisinage. On fait balayer une fenêtre de taille impaire à travers l'image, et pour chaque pixel central on calcule son seuil, parmi ces méthodes on trouve la méthode de Niblack.

IV.4. Méthode d'Otsu :

IV.4.1 définition :

En vision par ordinateur et traitement d'image, la méthode d'Otsu est utilisée pour effectuer un seuillage automatique à partir de la forme de l'histogramme de l'image, ou la réduction d'une image à niveaux de gris en une image binaire. L'algorithme suppose que l'image à binariser ne contient que deux classes de pixels, (c'est-à-dire le premier plan et l'arrière-plan) puis calcule le seuil optimal qui sépare ces deux classes afin que leur variance intra-classe soit minimale [28].



Image originale



Image binarisée par l'algorithme d'Otsu

Figure II. 8. Image binarisée par la méthode d'Otsu.

Dans la méthode d'Otsu, le seuil qui minimise la variance inter-classe est recherché à partir de tous les seuillages possibles :

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \dots \dots \dots (10)$$

Les poids ω_i représentent la probabilité d'être dans la n-ième classe, chacune étant séparée par un seuil t . Finalement, les σ_i^2 sont les variances de ces classes.

Otsu montre que minimiser la variance inter-classe revient à maximiser la variance intra-classe:

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t) [\mu_1(t) - \mu_2(t)]^2 \dots \dots \dots (11)$$

Qui est exprimée en termes des probabilités de classe ω_i et des moyennes de classes μ_i qui à leur tour peuvent être mises à jour itérativement. Cette idée conduit à un algorithme efficace.

IV.4.2. Les étapes de l'algorithme :

Cette méthode de binarisation nécessite au préalable le calcul de l'histogramme. Puis, la séparation en deux classes est effectuée.

- Calcul de l'histogramme

Le calcul de l'histogramme est très simple.

On initialise un tableau hist avec des 0. Généralement, ce tableau est constitué de 255 cases correspondant aux 255 niveaux de gris d'une image.

Ensuite, si $p(i, j)$ représente la valeur du pixel au point (i, j) , on balaye toute l'image et on compte le nombre de fois où un niveau de gris apparaît.

- Séparation en 2 classes

La séparation se fait à partir des moments des deux premiers ordres : la moyenne et l'écart-type. Pour que le procédé soit indépendant du nombre de points dans l'image N , on normalise l'histogramme : $p_i = n_i/N$ où n_i représente le nombre de pixels de niveau i . On peut calculer alors les deux moments utilisés :

$$\mu(k) = \sum_{i=1}^{i=k} i \cdot P_i \quad \text{et} \quad \omega(k) = \sum_{i=1}^{i=k} P_i$$

On note $\mu_T = \mu(256)$, où 256 est le nombre totale de niveaux de gris.

Si on appelle ω_0 la probabilité de la classe C_0 et ω_1 la probabilité de la classe C_1 , alors:

$$\omega_0 = \omega(k^*) \quad \text{où } k^* \text{ représente le niveau de seuil.}$$

$$\text{Et } \omega_1 = 1 - \omega(k^*).$$

Si on note de même μ_1 et μ_0 avec :

$$\mu_0 = \mu(k^*) / \omega(k^*) \quad \mu_1 = (\mu_T - \mu(k^*)) / (1 - \omega(k^*))$$

Or l'image totale conserve certaines propriétés, d'où on peut tirer les relations :

$$\omega_0 \mu_0 + \omega_1 \mu_1 = \mu_T \quad \text{et} \quad \omega_0 + \omega_1 = 1$$

En introduisant un paramètre pour évaluer la qualité du niveau de seuillage, on obtient :

$$S^2 = \omega_0 * \omega_1 (\mu_1 - \mu_0)^2 . \text{ La valeur précédente est fonction de } k.$$

On calcule donc cette valeur pour les 256 niveaux de gris de l'image. En fait on peut déjà enlever les valeurs 0 et 255 qui correspondent à affecter tous les pixels à la même classe.

A partir de $\omega(k)$ et $\mu(k)$, on calcule donc :

$$S^2(k) = \omega(k) * (1 - \omega(k)) * (\mu_T * \omega(k) - \mu(k))^2$$

La valeur du seuil k^* est obtenue pour le maximum de S^2 .

Il ne reste plus qu'à comparer la valeur de tous les pixels de l'image au seuil ainsi trouvé. Les résultats de l'application de cette méthode sont représentés par la figure II.9.



Image initiale.

Image binaire.

Figure II. 9. Image binarisée par Otsu.

On remarque la présence d'un bruit important dans le fond de l'image. Cependant on arrive à distinguer convenablement les caractères [29].

V. CONCLUSION :

On a présenté dans ce chapitre les différentes parties qui précède un système de segmentation de document notamment ; le filtrage, binarisation.

Le chapitre qui suit va détailler la segmentation d'image document, ainsi que l'algorithme d'étiquetage suivi.

CHAPITRE III :

Segmentation d'images de
documents.

I. INTRODUCTION :

La segmentation est nécessaire pour pouvoir traiter correctement les images numériques de documents. De nombreuses techniques ont été trouvées, certaines plus performantes que d'autres, mais comme nous allons le voir, le plus souvent destinées à un domaine particulier. La multitude de techniques rend leur présentation difficile, la réalisation d'une simple énumération est un piège qu'il faut éviter. La présentation des différentes techniques est séparée en fonction de l'approche, afin de rendre plus clair les différences qui existent entre les techniques, mais parfois ce découpage n'est pas correct car certaines techniques combinent les approches. Différentes méthodes de segmentation existent voir les Annexe .2 et Annexe .3, mais aucune ne permet de mettre parfaitement en avant les différences [17].

II. DEFINITION :

La segmentation est l'un des éléments les plus importants dans le traitement d'image car elle permet l'extraction des entités et des objets pour des traitements ultérieurs telle la reconnaissance de formes.

La segmentation est une tâche des plus délicates et des plus complexes. Elle a comme objectif de délimiter toutes les régions d'intérêt de l'image, avec un degré de finesse variable. Ainsi, il s'agit de délimiter les régions homogènes appelées blocs (textes, figures, notes en bas de page, etc.), puis, à l'intérieur de ces régions, les lignes de texte, les mots et enfin des caractères.

La détection des filets (lignes de séparation, horizontales ou verticales) et des cadres fait également partie de la segmentation. Une telle étape est surtout utile pour l'analyse de formulaires contenant souvent des grilles et des cases à remplir. Les cadres et les filets ainsi détectés ne servent souvent qu'à des fins de filtrage et de localisation des autres informations, car ils n'ont pas de signification propre.

III. LES DIFFERENTES APPROCHES DE LA SEGMENTATION :

III.1. Segmentation ascendante :

La catégorie des algorithmes de segmentation ascendante est caractérisée par le fait que l'analyse part de composants de bas niveau (comme les pixels) pour essayer de les fusionner en utilisant des heuristiques. Il existe de nombreuses variantes, certaines se focalisent sur la caractérisation des composants de bas niveaux, tandis que d'autres cherchent à bien modéliser les règles de fusion des régions. La grande majorité des techniques utilisent la notion de composants connexes. Ce sont des composants de bas niveau ayant les mêmes propriétés (comme un même niveau de gris) et qui sont liés spatialement. Il existe plusieurs définitions de cette notion de « relation spatiale » comme la 4-connexité ou la 8-connexité. La première méthode étudiée est encore très utilisée car relativement simple à mettre en œuvre. Il s'agit de la méthode dite RLSA (Run Length Smoothing Algorithm) [18].

III.1.1. Algorithme RLSA

Il s'agit d'une méthode itérative basée sur des opérations morphologiques de traitement d'image. Le principe de cette méthode est de noircir toute séquence de pixels blancs comprise entre deux pixels noirs, de longueur inférieure à un seuil donné. En pratique l'algorithme est appliqué horizontalement et verticalement sur l'image binaire originale avec des seuils éventuellement différents pour l'horizontale et la verticale, puis une opération «ET logique » est appliquée entre les deux images lissées obtenues. L'extraction des composantes connexes de l'image résultante permet d'obtenir les entités de la structure physique sur un niveau hiérarchique donné. On peut ainsi, en répétant la procédure avec des seuils de lissage horizontal et vertical différents, extraire itérativement les blocs de l'image, puis les lignes de texte et les mots. Ces seuils de lissage sont les seuls paramètres de l'algorithme RLSA. Ils contrôlent la manière dont les composantes sont fusionnées sur un niveau de segmentation prédéterminé. Par exemple pour segmenter des lignes de texte horizontales, droites et bien espacées, on pourra utiliser un seuil de lissage vertical nul et un seuil de lissage horizontal suffisamment grand pour combler les espaces inter-caractères et inter-mots.

Cet algorithme a l'avantage, en plus d'être très simple à mettre en œuvre, de ne requérir qu'un nombre limité d'itérations pour atteindre la segmentation complète du document. La principale difficulté dans l'utilisation de cet algorithme est le réglage des seuils de lissage qui est délicat.

Il est nécessaire de déterminer les seuils adéquats pour chaque niveau de segmentation. De tels seuils ne peuvent être déterminés qu'empiriquement. De plus, pour une itération donnée, ces seuils sont constants. Il faut donc que pour le niveau de segmentation considéré, les espaces entre les composantes de l'image à fusionner soient aussi constants. Antonacopoulos [19] proposa une autre méthode basée, elle aussi, sur une approche en composantes, dont la grande différence est le regroupement en composants des espaces plutôt que des caractères. En effet, ce sont généralement les espaces qui séparent les différents objets d'une image. En détectant les espaces, l'approche permet d'isoler les régions, sans avoir besoin de les caractériser. Cependant, la méthode reste sensible au bruit du fond et marche mal si les caractères ou les lignes se touchent.

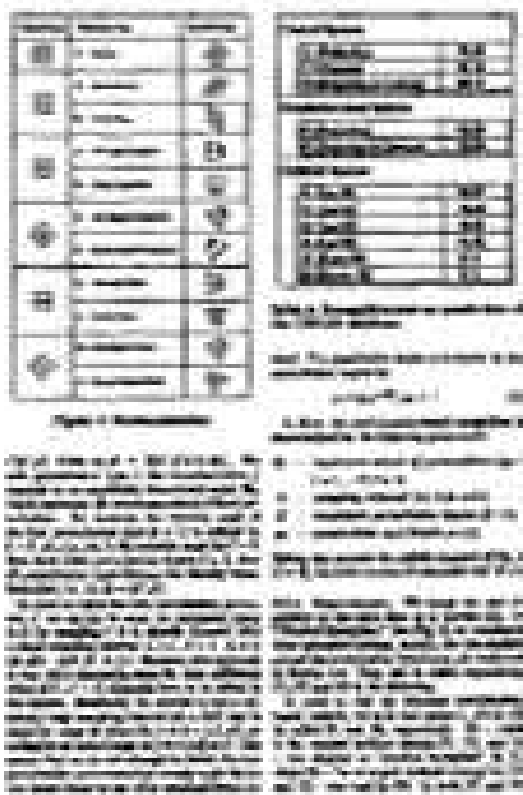


Figure III. 1. Image originale.

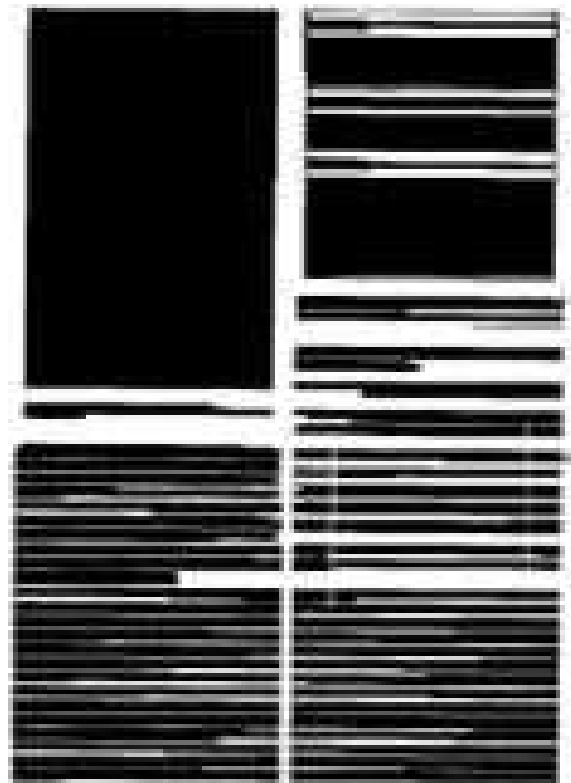


Figure III. 2. Image segmenté avec RLSA.

III.1.2. Utilisation d'heuristiques

L'inconvénient des méthodes précédentes est du même type que la binarisation à seuil global: elle est trop générale pour s'adapter à des documents ayant des caractéristiques variées. De même que pour la binarisation, les recherches se sont alors tournées vers des méthodes adaptatives.

Ainsi, la méthode de Messelodi [20] prend en compte différents paramètres pour fusionner les composantes connexes. Afin d'obtenir les composantes connexes, l'image est

d'abord binarisée pour ensuite regrouper les pixels en groupes connexes. La méthode de binarisation utilisée est un seuillage global, à réaliser deux fois afin d'extraire le texte noir sur fond clair et le texte clair sur fond sombre. Les deux images sont traitées séparément. Ensuite, en partant de l'hypothèse que les composantes connexes contiennent tout le texte mixé avec du bruit, des heuristiques permettent de trier les bonnes composantes des mauvaises. La première heuristique sert à détecter le bruit et se base sur le nombre des pixels de la composante. En effet, le bruit produit de petites tâches, il suffit de fixer un seuil au-dessous duquel la composante est classée comme bruit (en faisant attention aux ponctuations). Ensuite, la taille de la composante comparée avec la taille de l'image permet de pondérer la probabilité d'être en présence du texte. D'autres critères sont utilisés, comme la densité ou le contraste, pour essayer de supprimer au maximum les composantes n'étant pas du texte. Ensuite, les composantes sont regroupées en ligne en utilisant d'autres heuristiques, comme la distance, l'alignement ou la taille comparée.

Bien que cette méthode soit efficace pour trouver le texte dans une image sans a priori sur l'orientation du texte, la binarisation utilisée empêche la détection de texte dans le cas de documents mal éclairés. En 2006, Wang [21] proposa une nouvelle méthode permettant de mieux traiter la segmentation, tout en conservant cette approche basée sur des heuristiques. La méthode se place en amont de la détection de zones de textes. Cette approche est intéressante pour la segmentation car elle propose de nombreuses heuristiques pour caractériser les boîtes heuristiques qui pourront être réutilisées pour améliorer les performances d'une technique de segmentation. Le but du projet est de classer correctement le type des blocs dans neuf classes, dont le texte « petit » (moins de 18 points), le texte « grand » (plus de 19 points), les formules mathématiques, les logos, etc. Afin de réaliser la classification, un ensemble de 69 caractéristiques est calculée, parmi lesquelles le « run length », l'auto-corrélation ou encore le ratio entre la longueur et la largeur, sur lesquels un arbre binaire de classification étiquette le bloc. Le but n'est pas de segmenter l'image, cependant, la méthode est facilement adaptable à une approche ascendante, et permettrait d'ajouter des caractéristiques pour traiter plus efficacement les fusions de régions.

III.2 Segmentation descendante :

La famille de techniques de segmentation descendante essaie d'avoir une approche globale pour affiner les régions. Apparue légèrement après les méthodes ascendantes, elles ne connurent pas un grand développement malgré les très bonnes performances obtenues. La

plus connue d'entre elles est certainement la méthode X-Y Cut, présenté par Nagy et Seth en 1984 [22].

X-Y-Cut

Le principe de la méthode du X-Y-Cut est d'utiliser une projection horizontale et verticale afin de trouver les espaces interlignes. Une projection verticale (resp. horizontale) est la somme des valeurs des niveaux de gris des pixels sur une ligne (resp. colonne). Ces projections représentent donc, pour une ligne donnée, l'intensité totale des pixels. Si la valeur de la projection est faible, c'est qu'il y a beaucoup de pixels foncés sur la ligne. La méthode X-Y Cut utilise cette propriété afin de segmenter un document [22]. L'image est dans un premier temps projetée horizontalement et découpée en bandes, là où la projection a les plus faibles valeurs. Sur chacune des bandes, une projection verticale est faite, et la bande est découpée en colonnes, en suivant le même principe que précédemment. L'algorithme réitère ce processus jusqu'à ce qu'il n'y ait plus de creux dans les projections ou jusqu'à ce que les blocs aient atteint une taille inférieure à un seuil. Parallèlement à ces segmentations, un arbre est créé, dont les fils représentent les sous-catégories (bande ou colonne), ce qui permet de garder l'information de « qui contient qui ».



Figure III. 3. Exemple de document à problème.

Cependant, cette technique a le gros inconvénient de poser des problèmes avec des documents dont la mise en page est particulière. De même, dans le cas où les lignes ne seraient pas bien alignées, ou que le document serait légèrement tourné, la méthode ne peut plus trouver de creux dans les projections. Ces limitations ont conduit quelques recherches pour améliorer la technique, dont une correction de l'inclinaison des lignes, mais les résultats ne sont pas très convaincants [23].

IV. INTER CARACTERES, INTER MOTS ET INTER LIGNES :

IV.1. Segmentation en ligne :

Il s'agit ici de l'estimation d'espace entre les caractères et les mots utilisés dans notre méthode de fusion pour regrouper les signes en mots. Pour que l'estimation soit réalisée, on doit incrémenter un compteur pour sommer le nombre des pixels blancs entre deux pixels noirs et on met ces valeurs dans un vecteur, la valeur maximale dans ce vecteur représente la distance maximum entre les caractères et les mots. À partir de là on éliminera cet espace en remplaçant cette espace blanc par des pixels noirs.

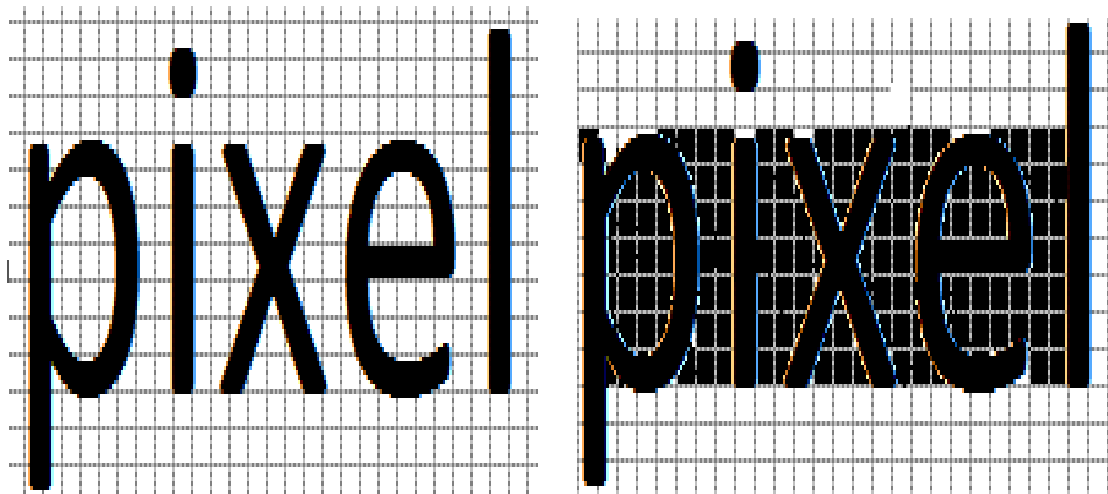


Figure III. 4. Estimation et élimination de l'espace inter-caractères.

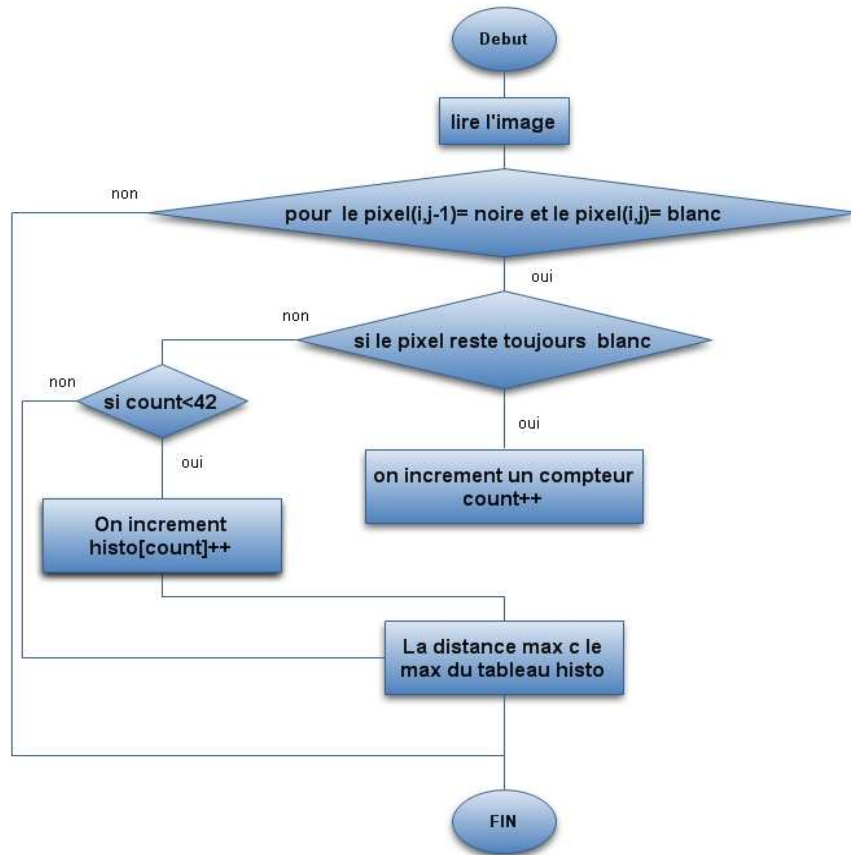


Figure III. 5. Organigramme qui explique le calcul de la distance entre les caractères et les mots.

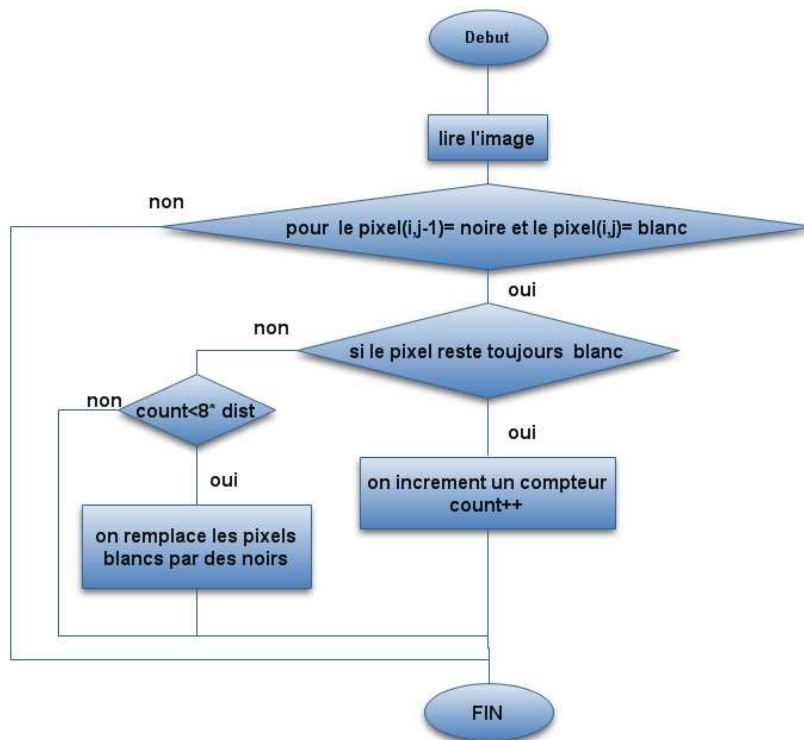


Figure III. 6. Organigramme qui explique l'élimination des espaces entre les caractères et les mots.

IV.2. Segmentation en bloc :

Si en typographie, interligne désigne l'espace entre deux lignes de texte consécutives, en informatique, il désigne généralement l'espace entre deux lignes de base consécutives. Cette dernière définition de l'interligne présente une propriété intéressante : celle d'être constante dans un document monospace. L'interligne, ainsi défini, par l'estimation des nombres de pixels blancs entre les lignes.

Cet estimation est réalisée en calculant le nombre de pixels entre deux lignes consécutives, on met ces valeurs dans un vecteur, la valeur maximale dans ce vecteur représente la distance maximum entre les lignes. À partir de la on éliminera cet espace en le remplaçant par des pixels noirs pour obtenir des blocs.

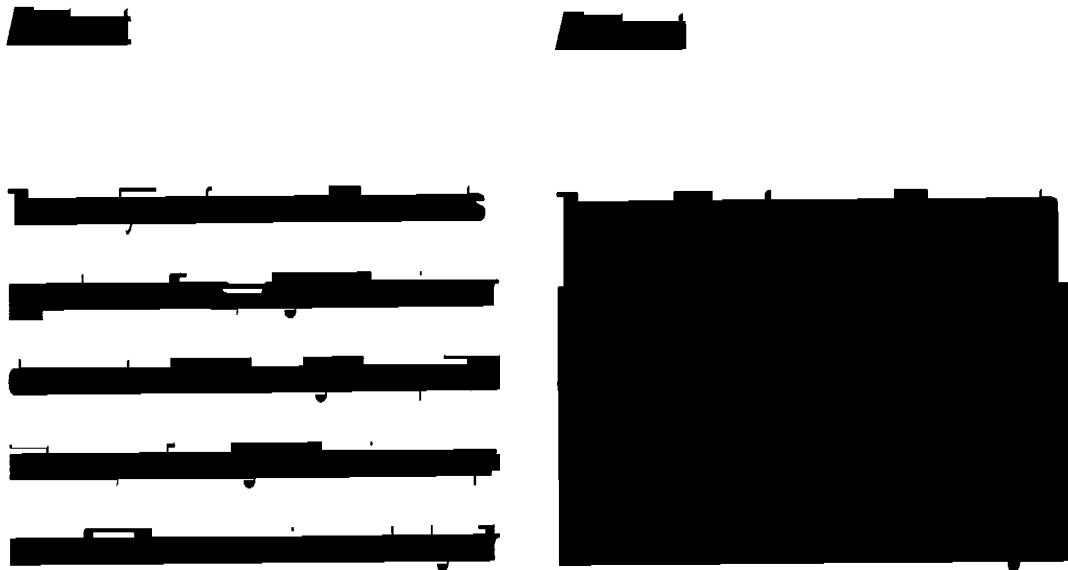


Figure III. 7. Estimation et élimination de l'espace interlignes.

V. L'ETIQUETAGE :

V.1. Définition :

L'étiquetage de régions, est une opération fondamentale du traitement d'images numériques de documents. Ce traitement affecte un numéro d'identification, ou étiquette, à chaque bloc de l'image soudée [12, 16].

L'étiquetage des régions n'est pas un traitement local, car le voisinage d'un point de l'image n'est pas suffisant pour lui affecter un numéro d'identification.

Cet opérateur a fait l'objet de nombreuses études sur les aspects algorithmiques et architecturaux qui ont mis en évidence deux types d'exécution des traitements : parallèle ou séquentielle.



Figure III. 8. Exemple de l'Etiquetage.

V.2. Etiquetage parallèle de régions :

Dans ce type d'exécution, l'itération de traitement utilise un parallélisme massif non compatible avec des systèmes embarqués [13,15].

V.3. Etiquetage séquentielle de régions :

Pour le cas séquentiel, les images sont traitées ligne par ligne [11, 14] et nécessitent deux passes. Ce double balayage a l'inconvénient de contraindre le traitement à mémoriser une image intermédiaire.

V.4 Algorithme d'étiquetage :

L'algorithme étudié se base sur les travaux de A. Rosenfeld et nécessite deux balayages de l'image [11].

Dans l'exemple suivant nous allons détailler comment fonctionne l'algorithme d'étiquetage des composants connexes et les différentes étapes suivies.

On a travaillé sur une image binaire simple. Supposons que l'image binaire est la suivante:

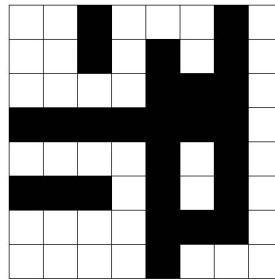


Figure III. 9. L'image binaire à étiqueté

Les blocs représentent un pixel. Le blanc représente le 1 et le noir représente le 0. Nous sommes intéressés à l'étiquetage des pixels non-nuls. Idéalement, il vous suffit de boucler l'image et arriver à un système d'étiquetage possible:

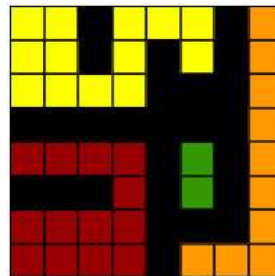


Figure III. 10. L'image étiquetée

Les étiquettes 1, 2, 3, 4, ...etc. sont juste réelles. Ici, les couleurs sont juste une représentation visuelle des étiquettes réelles.

- La première passe :

Nous vérifions les pixels en haut et à gauche. On ne trouve aucun pixel, alors nous avons besoins de créer une nouvelle étiquette. Cette étiquette prendra la couleur jaune.

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	1	1	0	0	0	1
1	1	1	1	0	1	1	1

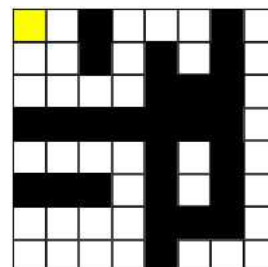


Figure III. 11. L'image binaire à étiqueté.

Figure III. 12. L'image pointée sur le pixel (1.1).

Ensuite on se positionne sur la première ligne de la matrice, le pixel (1,2) ne dispose que d'un seul pixel à sa gauche, nous avons donc copié son étiquette.

Le pixel suivant (1, 3) est un pixel de fond. Il ne nous intéresse pas, donc on l'ignore, il sera noir. Ensuite vient le pixel (1, 4), il n'a aucun pixel au-dessus, mais le pixel à sa gauche est un pixel de fond. Nous avons donc créé une nouvelle étiquette. Cette étiquette prendra la couleur verte.

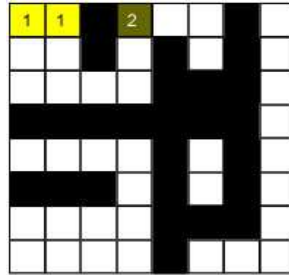


Figure III. 13. *L'image pointée sur le pixel (1,4), crée une nouvelle étiquette.*

Les pixels, (1, 5) et (1, 6) auront des pixels à leur gauche. Ainsi, ils obtiendront l'étiquette 2, le pixel (1, 7) est donc un pixel de fond, alors nous l'ignorons et le pixel (1,8) va prendre une nouvelle étiquette car le pixel à sa gauche est un pixel de fond. Cette étiquette prendra la couleur jaune foncé.

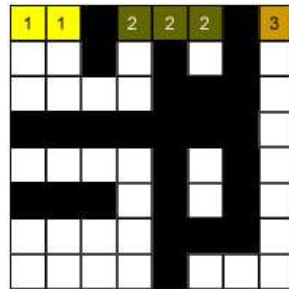


Figure III. 14. *L'image après le balayage de la première ligne.*

Après que le balayage de la première ligne est achevé.

Le pixel (2, 1) n'a aucun pixel à sa gauche, mais il a un pixel juste au dessus. Son étiquette est donc 1.

Donc, le résultat ressemble à ceci :

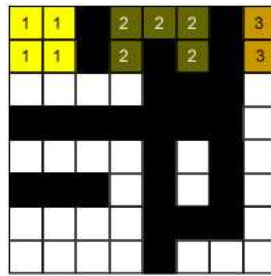


Figure III. 15. L'image après le balayage de la deuxième ligne.

Dans la troisième rangée, les pixels (3, 1) (3,2) et (3, 3) sont assez simple. Ils obtiennent l'étiquette 1. Le cas du pixel (3, 4) est délicat. Nous avons des pixels à la fois au dessus et à gauche. Les deux ont des étiquettes différentes. La on va prendre la petite étiquette dans ce cas 1 et mettre ça sur (3, 4). Et on retient que 2 est un enfant de 1.

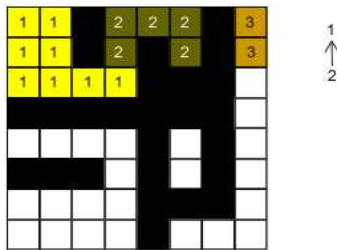


Figure III. 16. L'image au pixel (3,4).

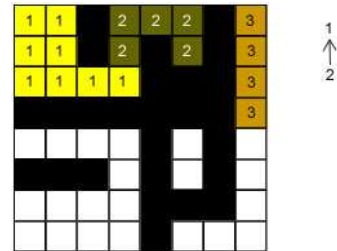


Figure III. 17. L'image après la quatrième ligne.

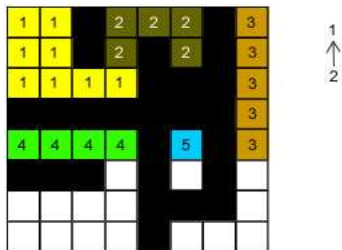


Figure III. 18. L'image après la ligne 5.



Figure III. 19. L'image après la ligne 6.

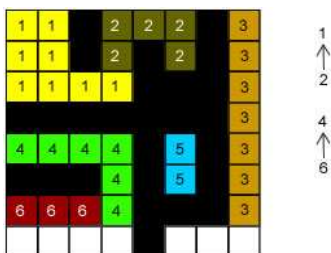


Figure III. 20. L'image après la 7.



Figure III. 21. L'image après la ligne 8.

- La seconde passe :

Maintenant, nous pouvons voir que cette technique de marquage a créé plusieurs étiquettes pour la même région ce qui pose un problème.

Pour ce la, on va de nouveau passer par tous les pixels un par un. On commence par pixel (1, 1) et vérifier la structure "union-find" donnée de l'étiquette 1. En vérifiant que l'étiquette 1 n'est pas un enfant de toute autre étiquette. C'est une racine d'elle-même. Donc, on se déplace au pixel voisin, de même pour le pixel (1,2).

Le pixel (1, 3) n'est donc pas concernés, car c'est un pixel de fond. Le pixel (1,4) vérifie la structure de donnée pour l'étiquette 2. On remarque que l'étiquette 2 est un enfant de l'étiquette 1. Et on sait aussi que l'étiquette 1 est une racine. Ainsi, il remplace l'étiquette de (1, 4) avec l'étiquette 1.

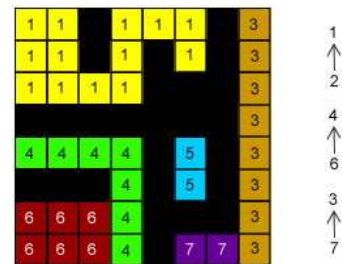
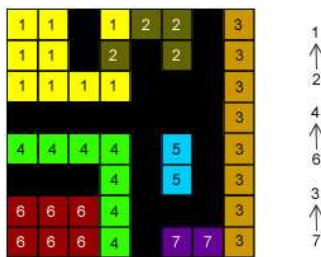


Figure III. 22. Le nettoyage de l'image étiqueté.

Figure III. 23. L'image nettoyée à l'étiquette 1.

La région connectée en haut à droite est complètement nettoyé.

La ligne 5 reste inchangée. Les étiquettes 3, 4 et 5 sont des racines. Ainsi que l'étiquette 6 reste également inchangée.

La ligne 7 subie des changements:

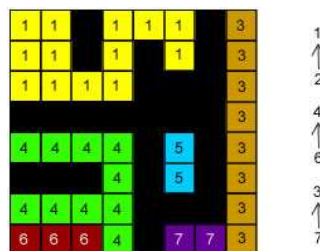


Figure III. 24. L'image nettoyée jusqu'à la ligne 7.

Et après avoir balayé la ligne 8, voici le résultat final:

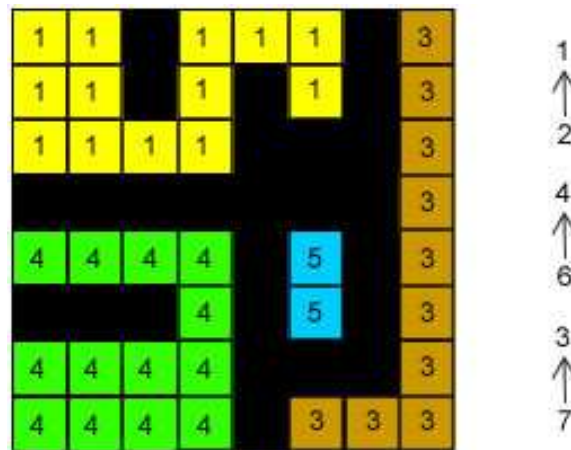


Figure III. 25. Résultat final d'étiquetage.

Le résultat final est d'avoir chaque région reliée exactement à la même valeur.

VI. CONCLUSION :

On a présenté dans ce chapitre les différentes approches de segmentation de document ainsi que les détails de la technique de segmentation utilisée et l'algorithme d'étiquetage.

Le chapitre qui suit sera consacré à la description des étapes de réalisation de notre application développée en langage C++.

CHAPITRE IV :

Implémentation et résultats

I.INTRODUCTION :

L'analyse et la segmentation d'images numériques de documents désignent une discipline scientifique qui regroupe un ensemble de techniques informatiques dont le but est de reconstituer le contenu d'un document à partir de son image.

Alors qu'elle est longtemps restée cantonnée dans la problématique de la reconnaissance de caractères, elle vise aujourd'hui des objectifs beaucoup plus larges, allant de la simple classification de documents à l'interprétation complète du contenu en passant par des techniques de segmentation qui donne plus de précision aux résultats, et dans notre projet on a utilisé la méthode ascendante « Botton- up ».

Notre application traite et analyse les images numériques de documents. Pour le développer, nous avons utilisé les langages de programmation le MATLAB R2009a et le C++ Builder 5.

II.PRESENTATION DES LANGAGES DE PROGRAMMATION UTILISES :

Pour l'implémentation et le développement de l'application on a utilisé les langages suivants :

II.1. Le langage MATLAB R2009a :

MATLAB est à la fois un langage de programmation et un environnement de développement, développé et commercialisé par la société américaine MathWorks. Il est utilisé dans les domaines de l'éducation, de la recherche et de l'industrie pour le calcul numérique mais aussi dans les phases de développement de projets, Il peut être utilisé aussi pour le traitement du signal, l'analyse statistique et le *traitement d'images*.

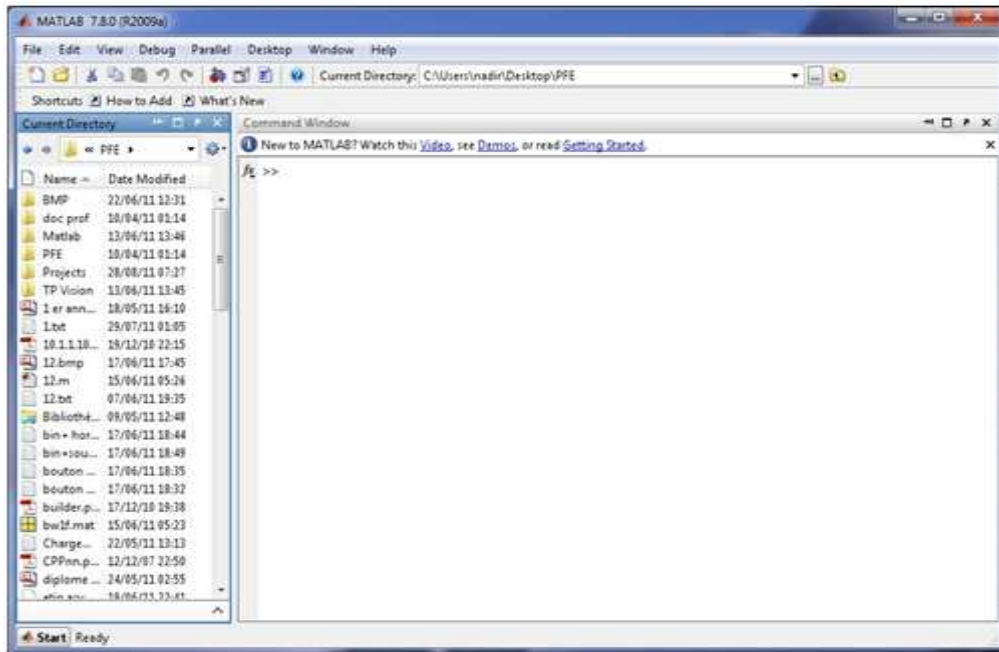


Figure IV. 1. Le langage MATLAB.

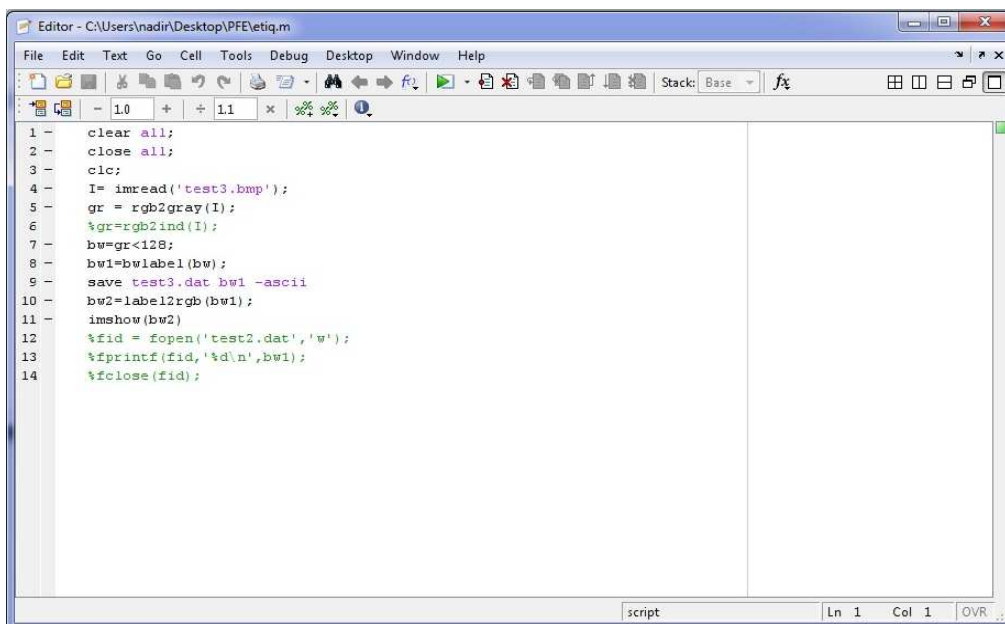


Figure IV. 2. Le programme avec le langage MATLAB.

II.2. Le langage Builder C++ 5 :

C++Builder est un environnement de programmation visuel orienté objet pour le développement rapide d'applications (RAD). En utilisant C++Builder, nous pouvons créer des applications Windows 32 bits très efficaces, avec un minimum de codage manuel. C++Builder fournit tous les outils qui sont nécessaires pour développer, tester, déboguer et

déployer des applications, incluant une importante bibliothèque de composants réutilisables, un ensemble d'outils de conception, des modèles d'applications et de fiches, ainsi que des experts de programmation. Ces outils simplifient le prototypage et réduisent la durée du développement.

Pourquoi le Builder C++5 ?

- Développé des programmes Windows C++ avec plus de facilité et de rapidité qu'auparavant.
- Créé des applications console Win 32 ou des programmes Win 32 GUI (Interface Utilisateur Graphique).
- Analyse, exploration et visualisation des données.

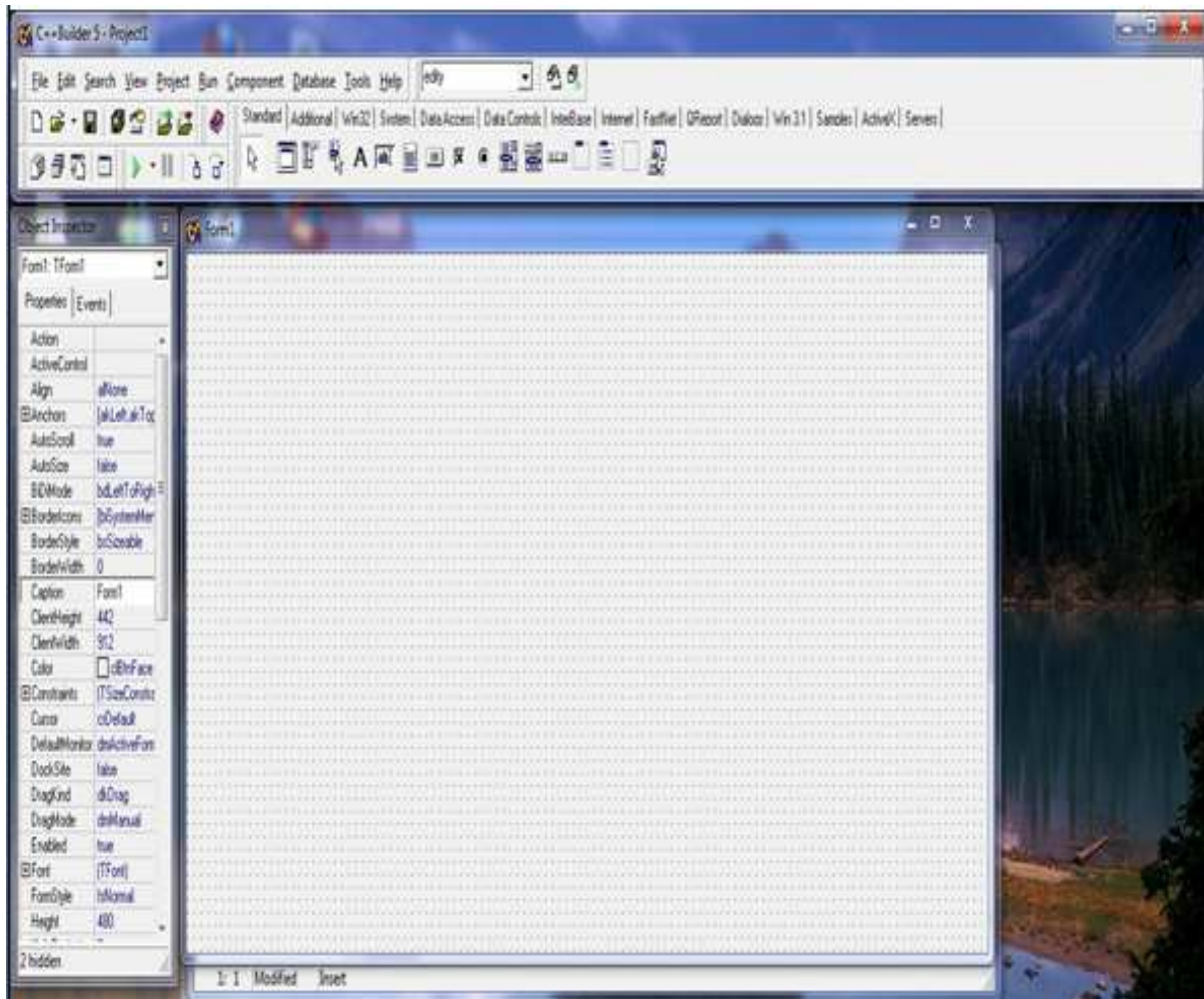


Figure IV. 3. Le langage Builder C++ 5.

III. FONCTIONNEMENT :

- On part de l'image numérique de document latin imprimé réalisée par un scanner optique d'une page (document imprimé, feuillet, dactylographié, etc....) dont on veut analyser cette image.
- L'image doit être assez contrastée et de format A4 scannée avec un scanner de haute résolution 300dpi.
- Le principe de notre programme est de lire le document puis analyser et traiter l'image de document afin de segmenter et étiqueter cette dernière pour avoir les différentes zones ou étiquettes de l'image.



Figure IV. 4. L'image numérique de document imprimé.

IV.DESCRPTION D'ESPACE DE TRAVAIL :

Nous allons découvrir l'espace de travail de notre application (figure III.5). Dans la partie supérieure nous avons les différents menus et les boutons qui nous permettent d'entamer les différentes procédures de l'analyse de notre document.

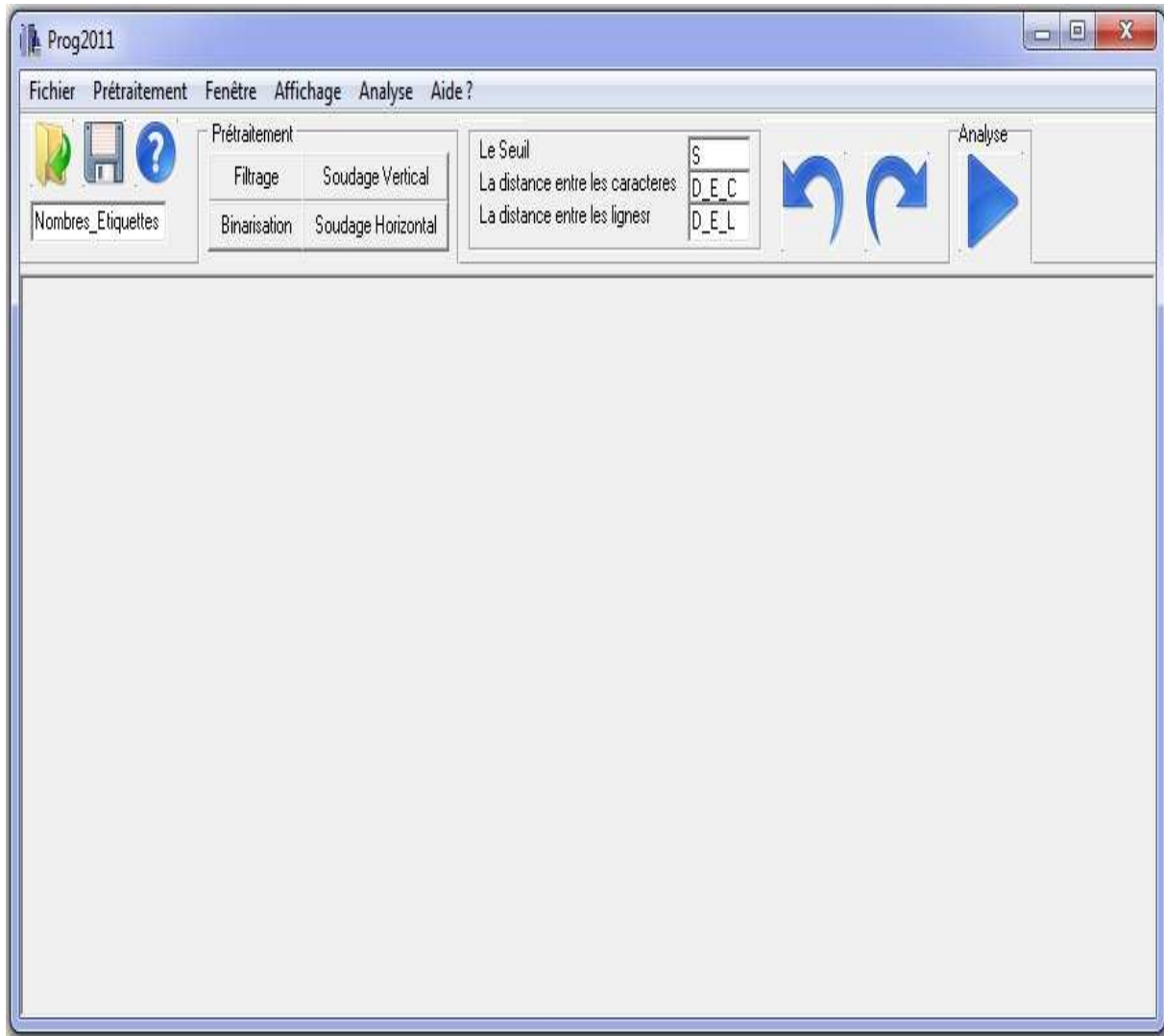


Figure IV. 5. *L'espace de travail de notre application.*

IV.1. Menu principale :

Le menu principal permet d'accéder aux différentes fonctions de notre application, dans ce qui suit nous allons détailler chaque sous menu.



Figure IV. 6. Le menu principal.

IV.1.1. Le menu fichier :



Figure IV. 7. Le menu fichier.

- **Ouvrir** : obtenir l'image à analyser, raccourci clavier Ctrl+O.
- **Enregistrer Sous** : sert à enregistrer l'image sélectionnée dans un fichier, raccourci clavier Ctrl+S
- **Quitter** : quitter l'exécution.

IV.1.2. Le menu prétraitement :

C'est le sous menu qui permet de faire prétraitement de notre image de document.



Figure IV. 8. Le menu prétraitement.

- **Filtrage :** Lance le filtrage de notre image, on utilise dans notre application le filtrage Gaussien, raccourci clavier Ctrl+F.



Figure IV. 9. Filtrage Gaussien de l'image.

- **Binarisation :** Lance la binarisation de notre image, on utilise dans notre application la binarisation par la méthode d'Otsu, raccourci clavier Ctrl+B.

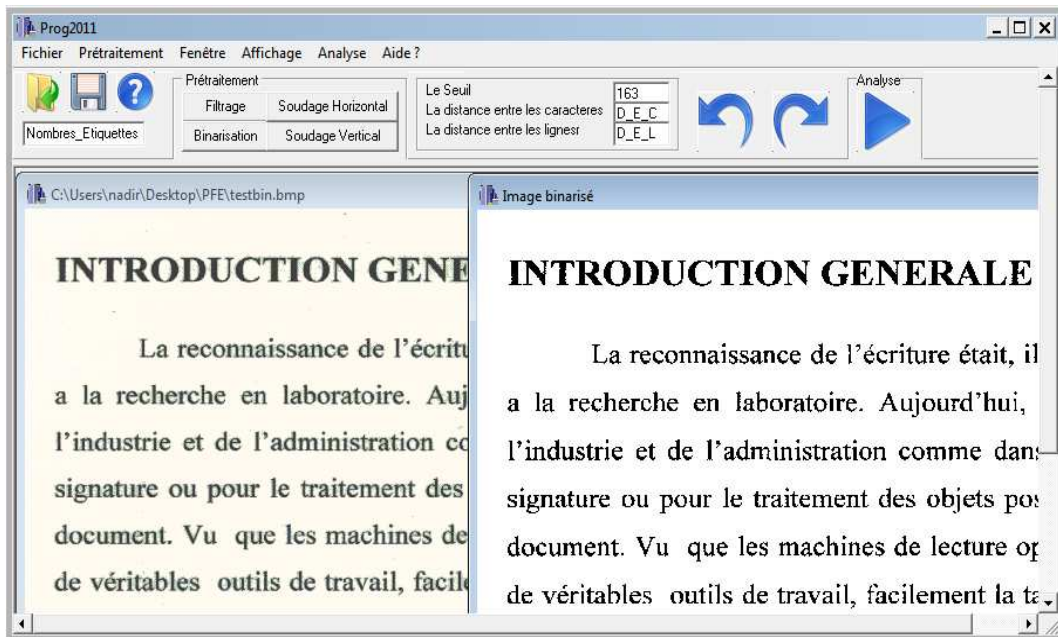


Figure IV. 10. Binarisation de l'image par la méthode d'Otsu.

- **Fusion :**

L'objectif du soudage est d'obtenir les paragraphes de l'image de documents souder on bloc.

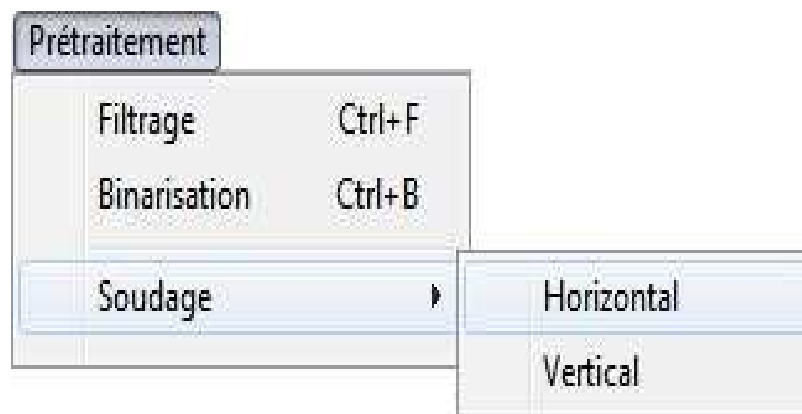


Figure IV. 11. Menu Fusion.

- *Fusion horizontale* : Dans cette phase on suit deux étapes, on calcule la distance maximum entre les caractères, et à partir de ça on va fusionner les lignes des paragraphes, on obtient le résultat suivant.

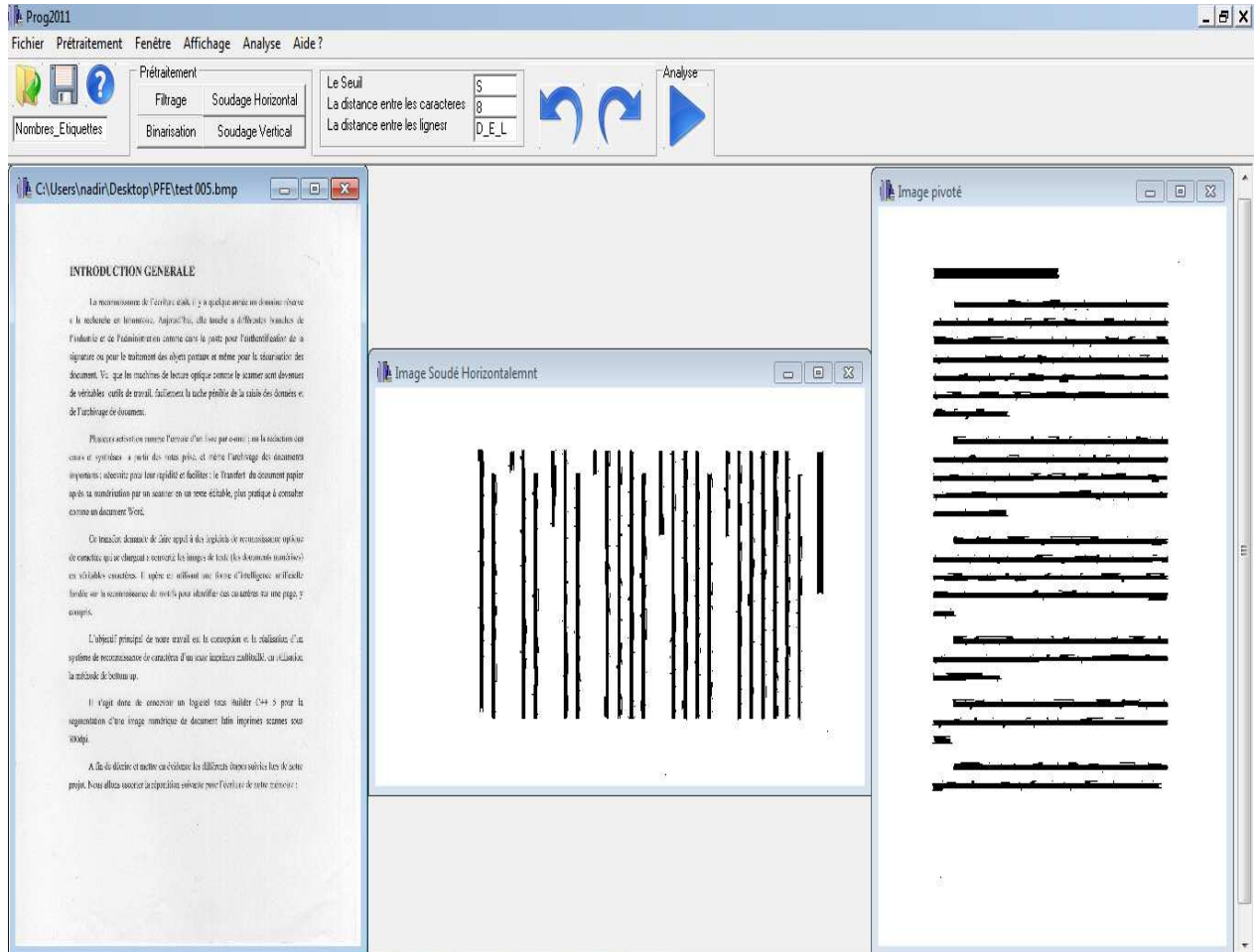


Figure IV. 12. *Fusion horizontale de l'image.*

- *Fusion verticale* : Dans cette phase on va pivoter à 90° notre image obtenue par la fusion horizontale, puis on va calculer la distance entre les lignes, et là on va souder les entre lignes pour avoir des blocs de chaque paragraphe.

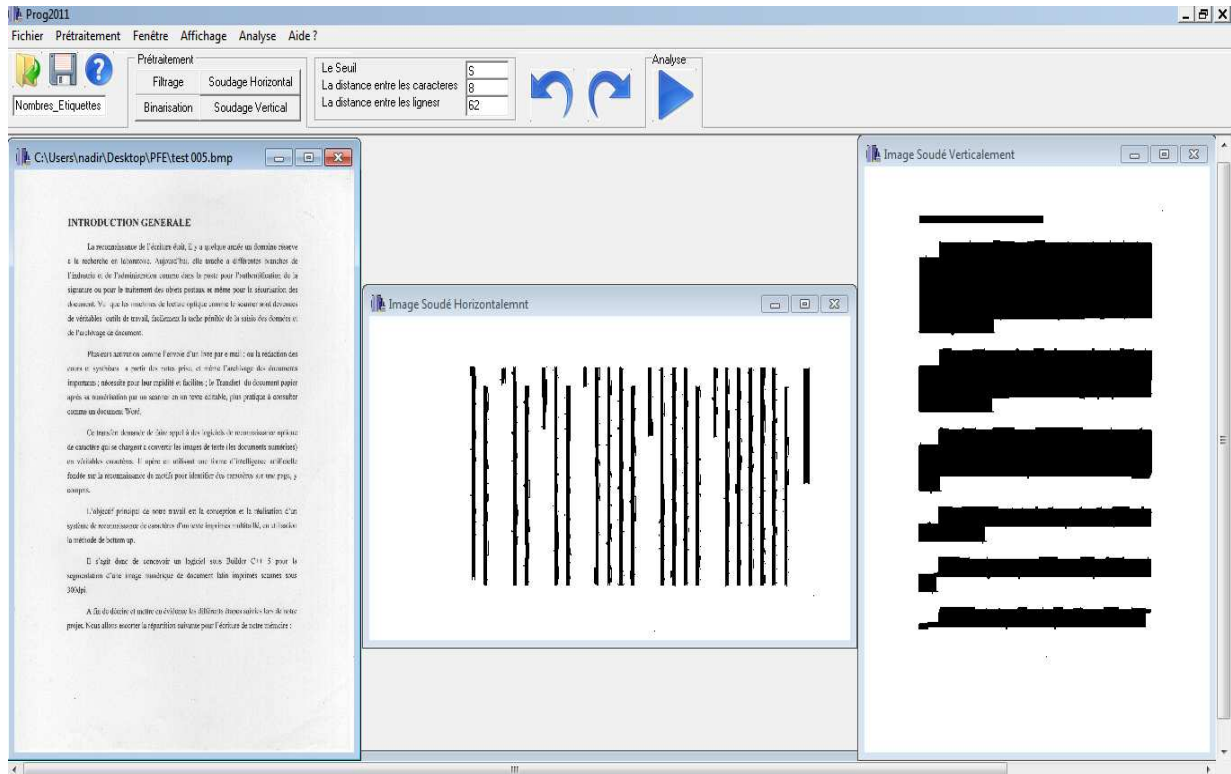


Figure IV. 13. Fusion verticale de l'image.

IV.1.3. Le menu fenêtre :



Figure IV. 14. Menu fenêtre.

- **Titre** : pour choisir la fenêtre par titre.
- **Cascade** : pour choisir la fenêtre en cascade.
- **Arranger tout** : pour réorganiser toutes les fenêtres.

IV.1.4. Le menu affichage :

C'est le sous menu qui permet de faire la rotation de notre image.

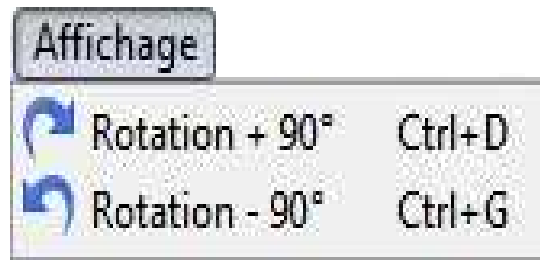


Figure IV. 15. Menu affichage.

- **Rotation + 90°** : Lance la rotation à droite (rotation + 90°) de notre image, raccourci clavier Ctrl+D.
- **Rotation - 90°** : Lance la rotation à gauche (rotation - 90°) de notre image, raccourci clavier Ctrl+G.

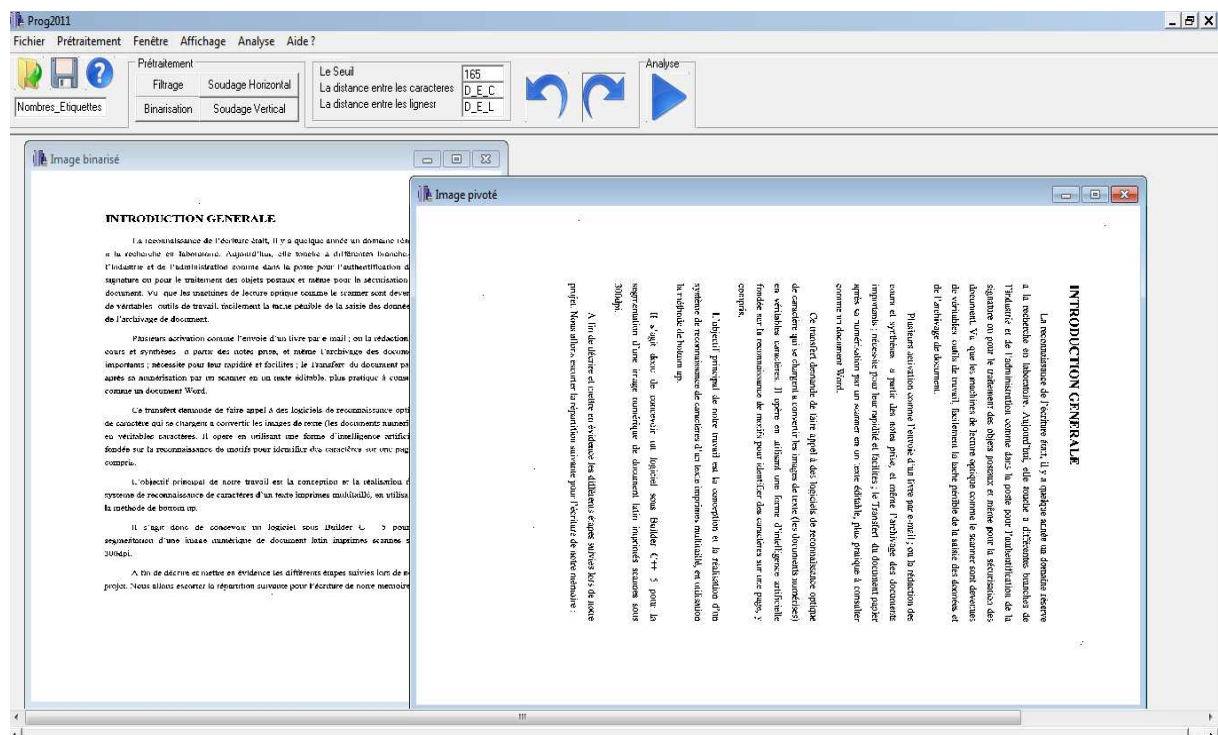


Figure IV. 16. La rotation de l'image.

IV.1.5. Le menu analyse :

C'est le sous menu le plus important c'est a ce niveau qu'on fait notre analyse de l'image numérique scannée, cette phase contient l'étiquetage



Figure IV. 17. Menu analyse.

➤ Etiquetage :

Ce sous menu lance :

- Lecture de l'image déjà étiqueté par un programme MATLAB et la mettre sous forme d'une matrice.
- Calcul et affichage du nombre d'étiquettes dans cette image déjà lu.
- Détermination du maximum et du minimum de chaque bloc.
- Calcul de l'inter ligne dans notre image initiale.

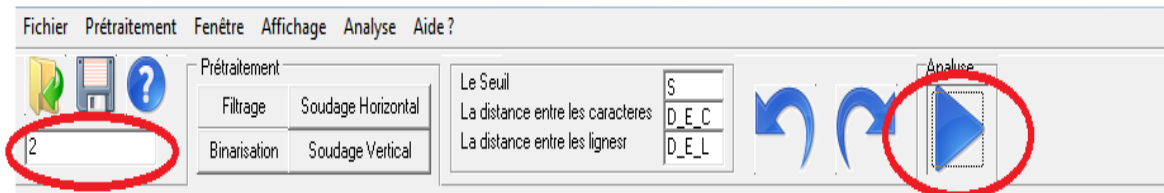


Figure IV. 18. Les résultats obtenus avec une image qui possède deux étiquettes.

IV.1.6. Le menu aide ? :

C'est le sous menu qui permet d'accéder à la boîte de dialogue A propos.



Figure IV. 19. Menu aide ?

- **A propos** : Lance la boîte de dialogue A propos qui nous donnent des informations sur notre application.



Figure IV. 20. La boîte de dialogue A propos.

IV.2. La barre d'outils :

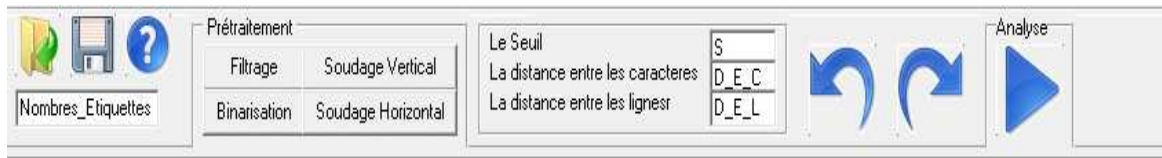
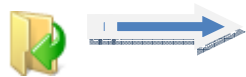


Figure IV. 21. La barre d'outils.

IV.2.1. Les boutons :



Bouton d'ouvrir notre image.



Bouton d'enregistrement de l'image obtenue.



Bouton d' A propos.



Bouton de filtrage Gaussien.



Bouton de binarisation.



Bouton de soudage horizontal.



Bouton de soudage vertical.



Bouton rotation à gauche (rotation - 90°).



Bouton rotation à droite (rotation + 90°).



Bouton donne le nombre d'étiquettes, le maximum et le minimum de chaque bloc étiqueté.

IV.2.2. Le groupe Box :

Ce groupe box est consacré à visualiser les différents paramètres résultants du prétraitement :

- Le seuil de la binarisation S.
- La distance entre les caractères de notre image de document D_E_C.
- La distance entre les lignes de notre image de document D_E_L.

Le Seuil	S
La distance entre les caracteres	D_E_C
La distance entre les lignes	D_E_L

Figure IV. 22. Groupe box pour visualiser les différents paramètres résultants.

Le Seuil	165
La distance entre les caracteres	8
La distance entre les lignes	62

Figure IV. 23. Exemple de visualisation des paramètres.

IV.2.3. Les édit :

Cet édit permet de savoir le nombre d'étiquettes de l'image segmentée.

Nombres_Etiquettes

Figure IV. 24. Édit pour afficher le nombre d'étiquettes de l'image segmentée.

V. RESULTATS ET DISCUSSION :

Les tests ont été réalisés sur différents types d'images de documents imprimés, tels que ; articles de revues scientifiques, polycopié de TP et série d'exercices. Dans ce qui suit nous allons tout d'abord présenter les différentes étapes que constitue notre système ; notamment, le filtrage, la binarisation, segmentation en ligne et finalement la segmentation en blocs.

V.1. Résultats du filtrage :

Dans cette étape l'image de document scannée est filtrée avec un filtre gaussien, dans le but de réduire le bruit. Le résultat du filtrage de cette étape est représenté dans la figure suivante.

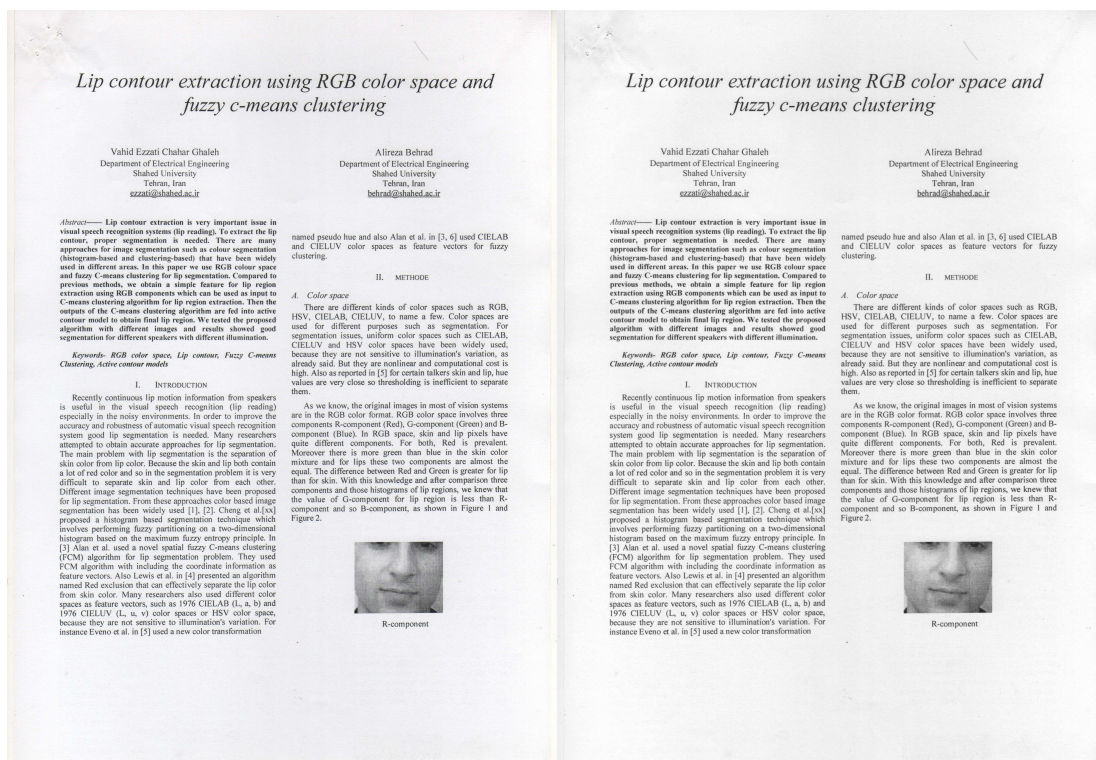


Figure IV. 25. Les résultats de filtrage.

V.2. Résultats de la binarisation :

Dans cette phase on binarise l'image de document pour avoir une image à deux niveaux codée sur 1 bit en utilisant la méthode d'Otsu, la figure suivante représente l'image binarisée.

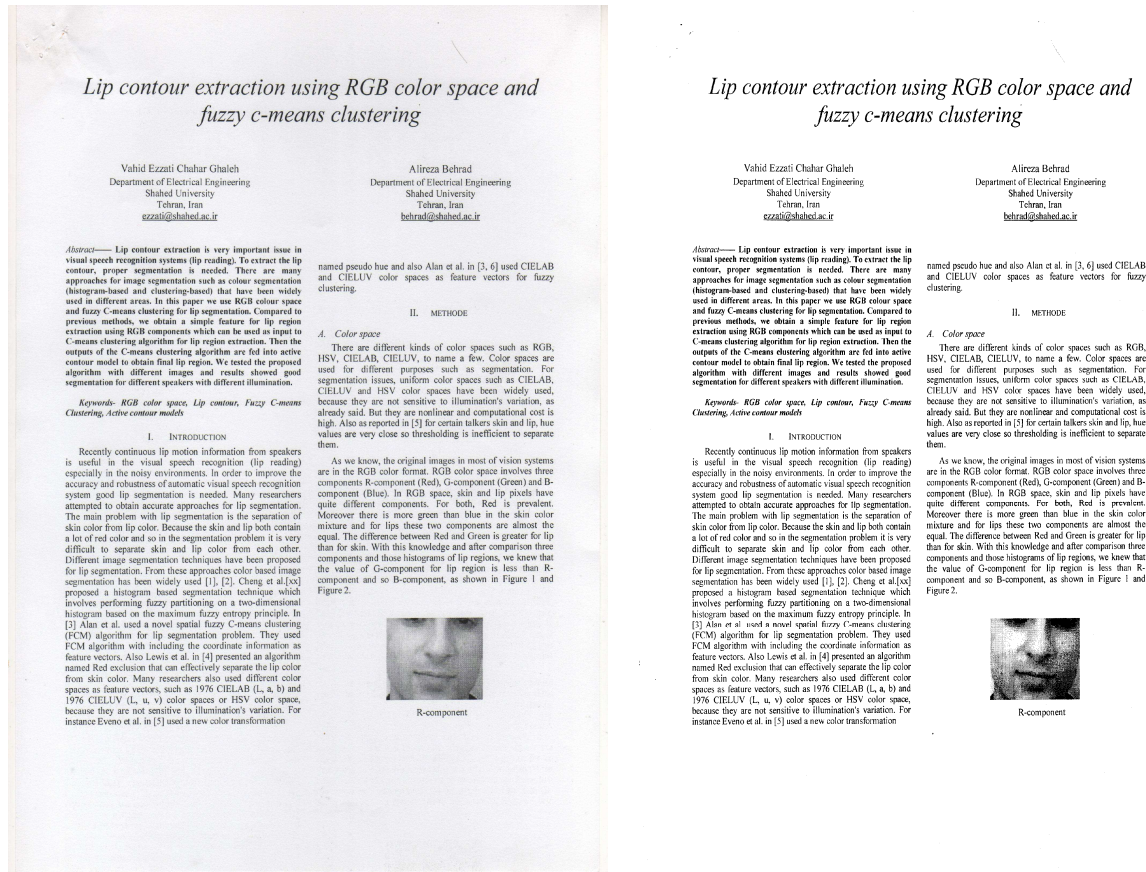


Figure IV. 26. L'image après binarisation

V.3. Résultats de la segmentation en ligne:

Dans cette étape on a estimé l'espace entre les caractères et entre les mots et on estimé aussi la distance maximale entre les caractères dans les lignes. En utilisant ces distances, tous les espaces dans la même ligne sont remplis par des pixels noirs comme l'indique la figure suivante.

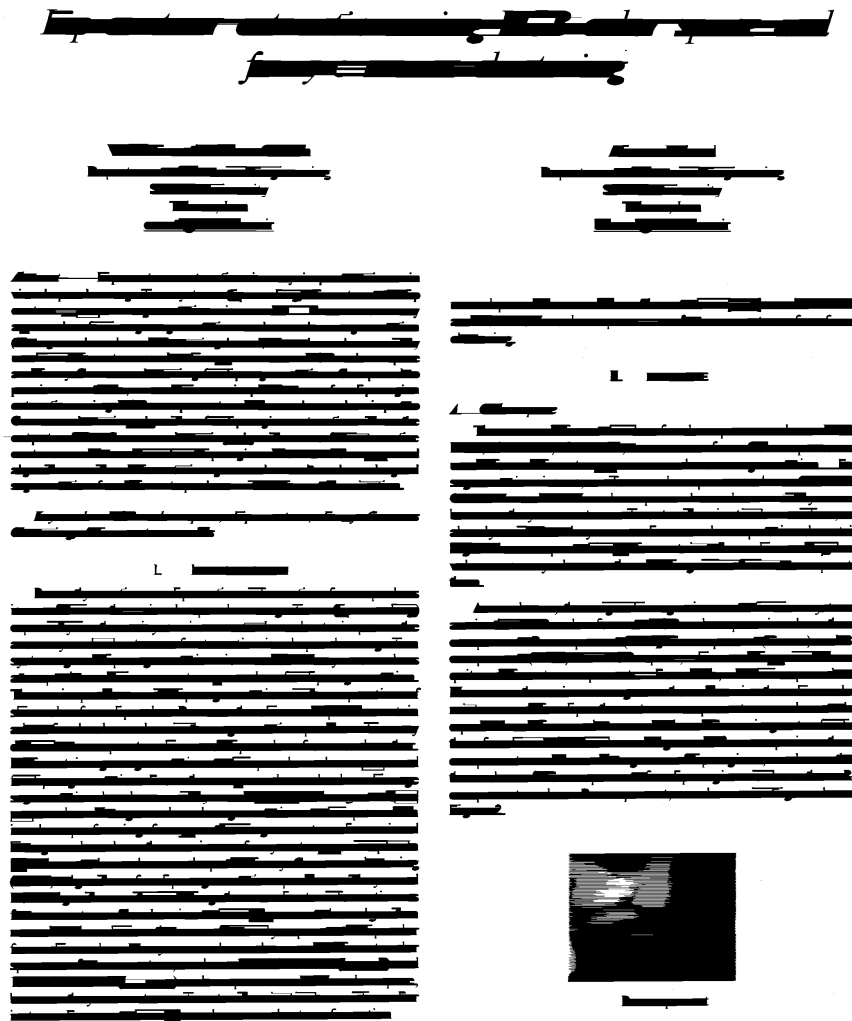


Figure IV. 27. Résultat de la segmentation en ligne.

V.4. Résultats de la segmentation en blocs :

A cette étape on a estimé l'interligne. Les interlignes sont éliminées en les remplaçant par des pixels noirs. Après des testes sur plusieurs types d'images de document on a obtenu les résultats suivants. Pour mieux illustrer cette phase on a superposé le texte initial sur l'image segmentée en blocs.

A Semantic-Based Text Classification System
Abdullah Hawaiki and Mourad Oussafik
School of Engineering
Department of Electronic, Electrical and Computer Engineering
University of Birmingham
a.h100@bham.ac.uk

Abstract
This paper presents a system that performs automatic semantic-based text classification. Using Probabilistic WordNet, a series of induced methods were implemented to extract semantic features from text and utilize them to decide how similar a document is to different topics. In addition, a bag-of-words method incorporating no knowledge from WordNet is implemented in the system as a baseline to compare different WordNet-based approaches with. This paper describes the system and report on a simple analysis performed to evaluate the different implemented methods. At the end, a discussion on the limitations of this study and the future work to optimize the system is presented.

Keywords: Semantic Similarity, WordNet, Categorization, Topic Classification, Information Retrieval, Word Sense Disambiguation

1. Introduction
Most of the available data nowadays exist in the form of electronic structured text information. The job of the data and to report growth makes it challenging for users to analyze, organize and access the required information efficiently. Systems that provide text categorization offer a promising approach for tackling this problem [1]. Text classification, or categorization, is the task of automatically assigning a class label to an unlabeled document.
During the last decades, a large number of text categorization systems have been proposed such as Support Vector Machines [2], Naïve Bayes [3], Term Frequency and Inverse Document Frequency [4] and others. While a number of approaches have been implemented in these systems, most of them use the bag-of-words model or vector space representation by having words (or word stems) and representative features for the document content [5].
With most of the bag-of-words approaches present an excellent performance on many machine-learning tasks, it is still limited to the type of training data that was introduced into the system. The results are

strongly more heavily influenced by the quality of the training data and its domain. It is thus possible to gain better results using multiple datasets by utilizing an external semantic thesaurus. However, WordNet [6] defines an upper-level of relationship between most of the terms in the training data. Our system performs semantic text classification through semantic-based approaches relying on semantic similarities between the training data and the classes.

The next section gives more background and details our work with similar existing semantic-based systems. In section 3, we give an overview on our system, its main stages and how it works. In section 4, we discuss the evaluation performed on different tests for the implemented methods in the system. In section 5, we present some ideas for future work and how the system can be improved.

2. Background

In the next few years, a number of researchers have implemented systems incorporating features derived from the semantic domain. Semantic WordNet for the English language containing semantic relations between groups of words [6]. Every word stands for a single word prototype that refers to a group of words that share the same (semantic) meaning. In addition to sharing a set of relations in WordNet, features such as these [7]. Speech tags have been considered by [8].

Our system uses WordNet as the underlying reference ontology for computing the semantic similarity between documents and classes. The hypothesis that was investigated by our system is that when computing the semantic similarity between sentences, it is optimal to compute as many words (including nouns and verbs) from the first sentence with a more terms from the second. Since WordNet does not contain "is-a" relations, directly linking nouns to verbs, an initial work for transferring verbs into nouns in our system to achieve the comparison between documents sentences and classes. A score was computed for each class to highlight how close it is related to the documents, the higher the score, the

Reconstruction of Fingerprint Images

Abdullah Hawaiki and Mourad Oussafik
School of Engineering
Department of Electronic, Electrical and Computer Engineering
University of Birmingham
a.h100@bham.ac.uk

Abstract
This paper presents a system that performs automatic reconstruction of fingerprint images. The system uses a series of induced methods to extract semantic features from text and utilize them to decide how similar a document is to different topics. In addition, a bag-of-words method incorporating no knowledge from WordNet is implemented in the system as a baseline to compare different WordNet-based approaches with. This paper describes the system and report on a simple analysis performed to evaluate the different implemented methods. At the end, a discussion on the limitations of this study and the future work to optimize the system is presented.

Keywords: Semantic Similarity, WordNet, Categorization, Topic Classification, Information Retrieval, Word Sense Disambiguation

1. Introduction
Most of the available data nowadays exist in the form of electronic structured text information. The job of the data and to report growth makes it challenging for users to analyze, organize and access the required information efficiently. Systems that provide text categorization offer a promising approach for tackling this problem [1]. Text classification, or categorization, is the task of automatically assigning a class label to an unlabeled document.
During the last decades, a large number of text categorization systems have been proposed such as Support Vector Machines [2], Naïve Bayes [3], Term Frequency and Inverse Document Frequency [4] and others. While a number of approaches have been implemented in these systems, most of them use the bag-of-words model or vector space representation by having words (or word stems) and representative features for the document content [5].
With most of the bag-of-words approaches present an excellent performance on many machine-learning tasks, it is still limited to the type of training data that was introduced into the system. The results are

strongly more heavily influenced by the quality of the training data and its domain. It is thus possible to gain better results using multiple datasets by utilizing an external semantic thesaurus. However, WordNet [6] defines an upper-level of relationship between most of the terms in the training data. Our system performs semantic text classification through semantic-based approaches relying on semantic similarities between the training data and the classes.

The next section gives more background and details our work with similar existing semantic-based systems. In section 3, we give an overview on our system, its main stages and how it works. In section 4, we discuss the evaluation performed on different tests for the implemented methods in the system. In section 5, we present some ideas for future work and how the system can be improved.

2. Background

In the next few years, a number of researchers have implemented systems incorporating features derived from the semantic domain. Semantic WordNet for the English language containing semantic relations between groups of words [6]. Every word stands for a single word prototype that refers to a group of words that share the same (semantic) meaning. In addition to sharing a set of relations in WordNet, features such as these [7]. Speech tags have been considered by [8].

Our system uses WordNet as the underlying reference ontology for computing the semantic similarity between documents and classes. The hypothesis that was investigated by our system is that when computing the semantic similarity between sentences, it is optimal to compute as many words (including nouns and verbs) from the first sentence with a more terms from the second. Since WordNet does not contain "is-a" relations, directly linking nouns to verbs, an initial work for transferring verbs into nouns in our system to achieve the comparison between documents sentences and classes. A score was computed for each class to highlight how close it is related to the documents, the higher the score, the

Figure IV. 28. Exemple d'un article scientifique ou il y a que des textes.

Lip contour extraction using RGB color space and fuzzy c-means clustering

Walid Elzafar Chahar Ghaleb
Department of Electrical Engineering
Shahed University
Tehran, Iran
ezzar@shahed.ac.ir

Alireza Behrad
Department of Electrical Engineering
Shahed University
Tehran, Iran
behrad@shahed.ac.ir

Abstract
Lip contour extraction is very important issue in visual speech recognition systems (lip reading). To extract the lip contour, proper segmentation is needed. There are many approaches for image segmentation such as color segmentation (histogram-based and clustering-based) that have been widely used in different areas. In this paper we use RGB color space and fuzzy C-means clustering for lip segmentation. Compared to previous methods, we obtain a simple feature for lip region extraction using RGB components which can be used as input to C-means clustering algorithm for lip region extraction. Then the output of the C-means clustering algorithm can be fed into active contour model to obtain final lip region. We tested the proposed algorithm with different images and results showed good segmentation for different operators with different characteristics.

Keywords: RGB color space, Lip contour, Fuzzy C-means Clustering, Active contour model

1. Introduction
Recently continuous lip motion information from speakers is useful in the visual speech recognition (lip reading) especially in the noisy environments. In order to improve the accuracy and robustness of automatic visual speech recognition system good lip segmentation is needed. Many researchers attempted to obtain accurate approaches for lip segmentation. The main problem with lip segmentation is the separation of skin color from lip color. Because the skin and lip both contain a lot of red color and so in the segmentation problem it is very difficult to separate skin and lip color from each other. Different image segmentation techniques have been proposed for lip segmentation. From these approaches color based image segmentation has been widely used [1], [2]. Cheng et al [3] proposed a histogram based segmentation technique which involves performing fuzzy partitioning on a two-dimensional histogram based on the maximum fuzzy entropy principle. In [4] Alan et al used a novel spatial fuzzy C-means clustering (FCM) algorithm for lip segmentation process. They used FCM algorithm with including the coordinate information as feature vectors. Also Levin et al [5] presented an algorithm named Red exclusion that can effectively separate the lip color from skin color. Many researchers also used different color spaces as feature vectors, such as 19% CIE-L*a*b) and 19% CIE-L*u*v) color space of HSV color space, because they are not sensitive to illumination variation. For instance Eizen et al [6] used a new color transformation

named pseudo hue and also Alan et al in [3,6] used CIE-L*a*b) and CIE-L*u*v) color spaces as feature vectors for fuzzy clustering.

II. METHOD

A. Color space
As we know, there are different kinds of color spaces such as RGB, HSV, CIE-L*a*b), CIE-L*u*v), G-component (Green) and B-component (Blue) in RGB space, skin and lip pixels have quite different components. For both, Red is prevalent. Moreover there is more green than blue in the skin color mixture and for lip these two components are almost the equal. The difference between Red and Green is greater for lip than for skin. With this knowledge and after comparison three components and their histograms of lip region, we know that the value of G-component for lip region is less than R-component and B-component, as shown in Figure 1 and Figure 2.



Figure IV. 29. Article scientifique avec des images à l'intérieur.

Distance-Invariant Sign Detection in High-Definition Video

Abdullah Hawaiki and Mourad Oussafik
School of Engineering
Department of Electronic, Electrical and Computer Engineering
University of Birmingham
a.h100@bham.ac.uk

Mediterranean Microwave Symposium MMS'2009
 November 15-17 Tangiers, Morocco
<http://www.ieee.ma/MMS2009/index.html>
 Abstract submission: June 30, 2009
 Acceptance notification: July 30, 2009
 Registration: September 1, 2009
 Final manuscript: September 30, 2009

The 5th International Conference on Signal Image Technology and Internet Based Systems (SITIS'09)
 November 29 – December 1 Marrakech, Morocco
<http://www.bourgoigne.fr/SITIS09/index.html>
 Paper Submission (extended): June 30th
 Acceptance/Reject notification: August 15th
 Camera ready: September 1st
 Author registration due date: September 1st

The 2nd INTERNATIONAL CONFERENCE ON COMPUTER AND ELECTRICAL ENGINEERING (ICCEE 2009)
 December 28 – 30 Dubai, UAE
<http://www.iccee.org/index.html>
 Final Paper Submission: July 15, 2009
 Notification of Acceptance: August 25, 2009
 Final Paper Submission: September 15, 2009
 Author registration: September 25, 2009

TP N°2: REDRESSEMENT-FILTRAGE

La plupart des montages électroniques nécessitent pour leur alimentation une (ou plusieurs) source continue (courant ou de tension) produite par un ensemble d'accumulateurs. Comme l'utilisation du réseau domestique présente une solution pratique de recharge très avantageuse puisque à moindre coût avec un encombrement moindre, la transformation d'un signal alternatif en un signal continu passera par les étapes d'abaissement de filtrage et enfin de stabilisation.

PREPARATION

1- REDRESSEMENT MONO-ALTERNANCE
 Soit le circuit de la figure (1). La tension d'entrée crête (ou maximale) $V_{acm} = 15V$.
 Calculer et inscrire dans un tableau la valeur des grandeurs suivantes:
 Tension crête aux bornes de R : V_{Rm}
 Tension moyenne aux bornes de R : V_{Rmoy}
 Courant moyen dans la diode : I_{Dmoy}
 Fréquence d'ondulation aux bornes de R (ou D) : f_o

2- REDRESSEMENT DOUBLE-ALTERNANCE
 Soit le circuit de la figure (2). La tension d'entrée crête (ou maximale) $V_{acm} = 15V$.
 Calculer et inscrire dans un tableau la valeur des grandeurs suivantes:
 V_{Rm} , V_{Rmoy} , I_{Dmoy} , f_o

3- REDRESSEMENT DOUBLE-ALTERNANCE AVEC FILTRAGE
 Soit le circuit de la figure (3). La tension d'entrée crête (ou maximale) $V_{acm} = 15V$, $R = 1k\Omega$ et $C = 47\mu F$. Calculer et inscrire dans un tableau la valeur des grandeurs :
 V_{Rm} , V_{Rmoy} , I_{Dmoy} , f_o
 ΔV_{Rmoy} représente la variation d'ondulation aux bornes de R.

MANIPULATION

1- REDRESSEMENT MONO-ALTERNANCE

- 1) Réaliser le montage de la figure (1). Observer les signaux aux bornes des différents composants.
- 2) Mesurer la valeur moyenne de la tension aux bornes de R. En déduire la valeur du courant dans le circuit.
- 3) Mesurer V_{Rm} et f_o .
- 4) Inscrire toutes les valeurs dans un tableau. Conclusion.

2- REDRESSEMENT DOUBLE-ALTERNANCE

- 1) Réaliser le montage de la figure (2). Observer à l'oscilloscope les signaux aux bornes du générateur et de la résistance R.
- 2) Mesurer et inscrire la valeur des grandeurs suivantes : V_{Rm} , V_{Rmoy} , I_{Dmoy} , f_o .
- 3) Conclusion.

Figure IV. 30. Image texte simple.

TP N°3: STABILISATION PAR DIODES ZENER

On voudrait utiliser une diode Zener pour stabiliser une tension aux bornes d'une charge. Cette étude sera faite dans le cas d'une entrée variable pour une charge fixe en sortie, et dans le cas d'une entrée fixe pour une charge variable en sortie.

A- PREPARATION

1- ETUDE AVEC ENTREE VARIABLE ET CHARGE FIXE EN SORTIE:
 (a) (1k):
 Soit le montage de la figure (1).
 Pour chaque diode zener de valeurs 6V, 9V et 15V, déterminer la valeur de R à partir de laquelle la diode commence à réguler.

2- ETUDE AVEC ENTREE FIXE (15V) ET CHARGE VARIABLE:
 Calculer pour le montage de la figure (1) les résistances de charge R_{min} et R_{max} pour lesquelles la tension de sortie est régulée.

B- STABILISATION D'UNE TENSION AVEC DIODE ZENER APRES FILTRAGE:
 Soit le montage de la figure (2).
 La tension d'entrée crête à crête est égale à 15V.
 Calculer et inscrire dans un tableau, pour $R = 1k\Omega$, $R_1 = 470\Omega$, $R_2 = 1k\Omega$, $R_3 = 47k\Omega$, les grandeurs suivantes:
 - V_c : Tension aux bornes du condensateur.
 - V_{Rm} : Tension d'ondulation crête à crête aux bornes du condensateur.
 - V_{R1} : Tension aux bornes de R_1 .
 - V_{R2} : Tension d'ondulation crête à crête.

B- MANIPULATION

1- ETUDE AVEC ENTREE VARIABLE ET CHARGE FIXE EN SORTIE:
 Réaliser le montage de la figure (1), pour chaque diode zener : 6V, 9V ET 15V.
 Relever et tracer le graphique de la tension de sortie en fonction de la tension d'entrée. La valeur crête à crête sera 0 et 15V, par pas de 1V. Quelle est le rôle de la diode zener ?

2- ETUDE AVEC ENTREE FIXE (15V) ET CHARGE VARIABLE:
 Réaliser le montage de la figure (1) et ajouter en série avec la diode zener, un ampèremètre.
 Pour chaque diode zener, faire ce qui suit :
 - Relever aux bornes de la diode zener la tension à crête à crête qui la traverse pour les différentes valeurs de la charge R qui suivent.

R (kΩ)	I (mA)	6V	9V	15V
0,02				
0,05				
0,1				
0,2				
0,5				
1				
2				
5				
10				

Tracer les variations de la tension relevée aux bornes de la zener en fonction de la valeur de R.
 Expliquer le fonctionnement du dispositif en interprétant vos résultats.

4 - Loi des mailles ou loi des tensions

Schéma de montage:

Matériel:

- U_g: tension continue 6V
- R1 = 100 Ω
- R2 = 200 Ω
- R3 = 60 Ω
- R4 = 20 Ω
- Un multimètre utilisé en voltmètre.

Fig.2

Mode théorique:
 Calculer les tensions aux bornes des résistances R1, R2, R3, R4 du montage de la figure 1.

Travail préparatoire:

- 1) Représenter sur le multimètre les bornes d'utilisation en volt-mètre, ainsi que le bon calibre.
- 2) Réaliser le schéma ci-dessus en vérifiant le sens du courant et les tensions aux bornes de chaque résistance.
- 3) Réaliser (à l'aide du multimètre) le schéma ci-dessus avec la position du multimètre permettant les mesures successives des tensions aux bornes de R1, R2, R3, R4.
- 4) Faire valider par le professeur les trois points précédents.

Méthode:

- 1) Réaliser le montage en plaçant le multimètre de telle façon qu'il mesure la tension U1 aux bornes de R1, puis U2, U3, U4, U5, U6 de sorte le circuit sous tension.
- 2) Mesurer alors U1 puis recommencer pour U2, U3, U4, U5, U6.
- 3) En observant les valeurs de ces tensions déterminer les égalités entre:

① U_g = U1 + U2
 ② U_g = U1 + U3 + U4
 ③ U_g = U1 + U5 + U6

NB: L'accès au laboratoire est conditionné par la présentation de la partie théorique au chargé de TP. Tout binôme ou trinôme n'ayant pas préparé cette partie sera renvoyé automatiquement de la séance de TP. Le compte rendu doit être remis à la fin de la séance de TP.

Figure IV. 31. Image qui contient des tableaux.

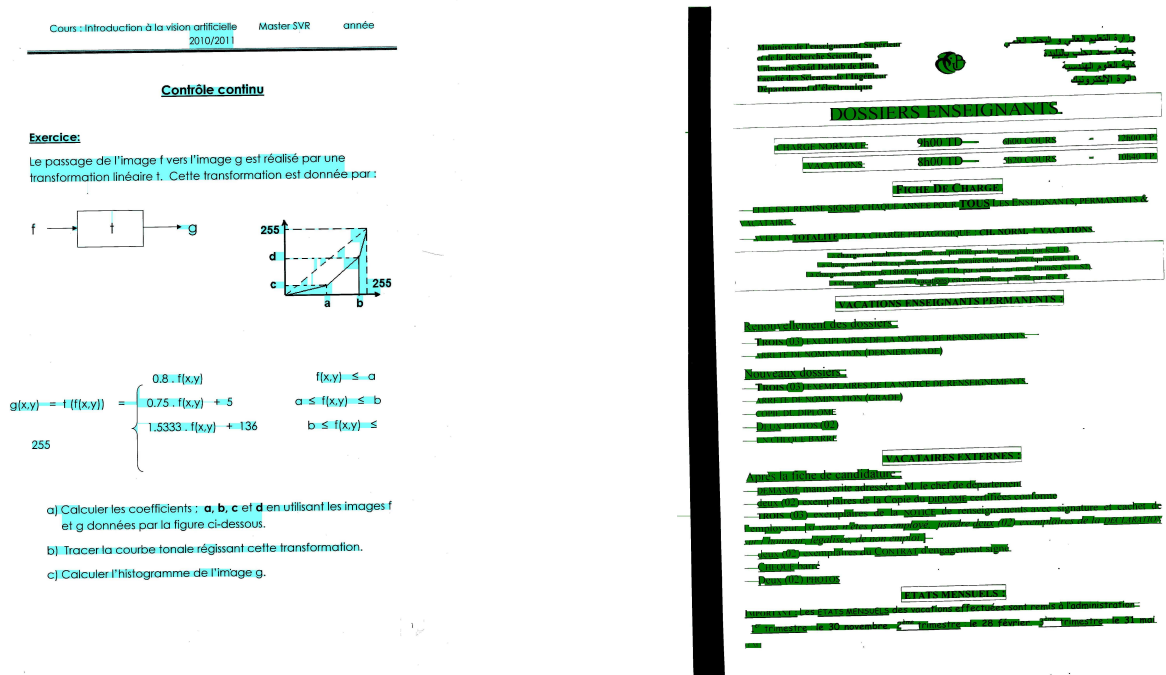


Figure IV. 32. Image qui contient des graphes.

V.4. Résultats d'étiquetage :

Dans cette phase on a utilisé un programme MATLAB pour l'étiquetage puis lire cette image étiquetée avec notre application en Builder C++. Afin de pouvoir déterminer le nombre des étiquettes et estimer l'espace inter ligne, l'image suivante présente une image document étiquetée.

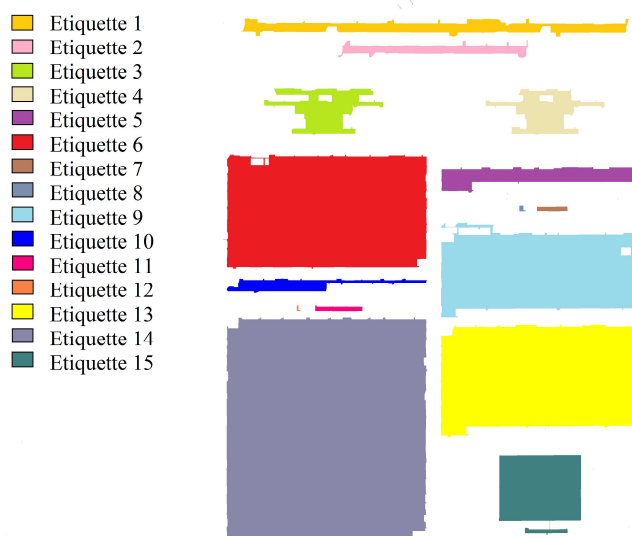


Figure IV. 33. Image étiquetée avec MATLAB.

VI. CONCLUSION :

D'après les différents tests et les résultats obtenus on constate qu'on a pu segmenter nos différents types d'images document avec succès. Avec ces résultats on peut par la suite classifier les différents blocs de l'image document en texte, image, ou même illustration.

CONCLUSION

GENERALE :

Bien que le nombre de techniques différentes puisse faire croire à un sujet fermé, la segmentation d'image est loin d'être résolue. En effet, le nombre de documents ayant des caractéristiques différentes rend difficile l'élaboration d'une technique non-supervisée capable de travailler sur l'ensemble des documents. Il est ainsi fréquent de trouver une solution très efficace pour, par exemple, la détection de texte dans une vidéo et incapable de segmenter un document manuscrit.

Au cours de notre travail, nous avons fait la réalisation d'une application d'un système de segmentation d'image de document papier de format A4 scanné sous 300 dpi, en utilisant l'approche ascendante « Bottom-Up ».

En premier lieu, nous avons introduit quelques généralités sur le traitement d'images, ainsi que les différentes méthodes de segmentation du document numérique et aussi la technique de l'étiquetage.

En deuxième lieu, nous avons présenté les différents procédés de prétraitement, dans le but d'avoir une meilleure qualité d'image de document acquise pour l'analyse.

A la fin nous avons décrit l'espace de travail de notre application, et mode fonctionnement, ainsi que les différents résultats obtenus.

Le travail que nous avons fait a permis de découvrir de vaste domaine comme l'informatique en manipulant le langage de programmation « Builder C++ 5 », et on a utilisé également le langage MATLAB R2009a.

Nous proposons comme continuité à ce travail et ce dans le cadre d'un projet de fin d'études de faire le test de chaque bloc de l'image segmentée afin de l'affecter à trois classes qui sont ; le texte, l'image ou le graphique.

Références bibliographiques

- [1] A . Chéhikian, « *Binarisation d'images, deux solutions à ce problème* », traitement du Signal, Vol . 6, n° 1, 1989, p. 59-68 .
- [2] G. Nagy, S . Seth, « *Hierarchical representation of optically scanned documents*», Proc. Seventh ICPR, Montreal, Canada, 1984, pp . 347-349 .
- [3] F. Hönes, J. Litcher, « *Layout extraction of mixed mode documents*», Machine Vision and Applications, Vol . 5, 1994, pp . 237-246.
- [4] M. KOCHER. Murat KUNT, G. GRANLUND, « *Traitement numérique des images* », volume 2. Presse polytechnique et universitaires normandes, 1993.
- [5] Jean-Jaeques Toumazet, « *Traitement de l'image* » par l'exemple Edition SYBEX Paris, Juillet 1990.
- [6] Eric TRINH, Rank LEBOUCEOIS & Nuria MARTIN, « *Numérisation des documents* », Laboratoire de reconnaissance des fromes et vision Institue national des sciences appliquées Lyon.
- [7] BOUABIDA Naima, « *Segmentation des tissus cérébraux sur des images par résonance Magnétique* », Projet de fin d'études, Electronique, ENP, Alger, 2007.
- [8] FRADJ Mohammed Karim, LOUKAL Abdelkader, « *Segmentation par étiquetage des artères coronaires avec implémentation sur DSP C6000* », Projet de fin d'études, Electronique, ENP, Alger, 2007.
- [9] DOUMANDJI Samah, BOULFANI Yasmine, « *Implémentation sur DSP TMS320C5000 de filtres optimaux appliqués aux images et introduction de réseaux neuronaux* », Projet de fin d'études, Electronique, ENP, Alger, 2004.

- [10] KADDOUR Chakib, AISSA BRAHIM Salim, « *Généralités sur le traitement d'images* ». [Http: //www.kaddour.com/chap1/chap1.htm](http://www.kaddour.com/chap1/chap1.htm).
- [11] A.ROSENFELD, J. L. PFALTZ , « *Sequential operations in digital picture processing* », Journal of ACM, vol. 13, n° 4, pp.471-494, 1966
- [12] A.ROSENFELD , « *Digital Picture Processing* », Academic Press, 1982
- [13] J.M.CHASSERY, « *Deux algorithmes orientés parallélisme : courbes de niveau et étiquetage* », 5ème congrès RFIA, Grenoble, pp.541-548, Novembre 1985
- [14] J.QUESNE, « *Architectures data-flow pour le traitement des images en temps réel* », Thèse, Vision robotique, pp. 65-82, Janvier 1992
- [15] E.MOZEF, S.WEBER, J.JABER, E.TISSERAND, « *Architecture dédiée à l'algorithme parallèle $O(n.\log 2n)$ d'étiquetage de composantes connexes* », GRETSI Symposium on Signal and Image Processing, Juan les Pins, pp.83-89, Septembre 1995
- [16] J.P.COCQUEREZ, S.PHILIPP , « *Analyse d'images : filtrage et segmentation* », Masson, pp.61-63, 1996
- [17] K. I. Kim, K. Jung, and A. K. Jain, « *Text information extraction in images and video : a survey* », Pattern Recognition, 37(5) :977–997, 2003. (Cité page 15.)
- [18] K.Y. Wong, R.G. Casey, and F.M. Wahl, « *Document analysis system. IBM Journal of Research and Development* », 26(6) :647–656, 1982. (Cité pages 24 et 29.)
- [19] A. Antonacopoulos, « *Page segmentation using the description of the background. Computer Vision and Image Understanding* », 70(3) :350–369, 1998. (Cité pages 25, 29 et 39.)
- [20] S. Messelodi and C.M. Modena, « *Automatic identification and skew estimation of text lines in real scene images* », Pattern Recognition, 32(5) :791–810, 1999. (Cité pages 25 et 29.)

- [21] Y.Wang, I. T. Phillips, and R. M. Haralick, « *Document zone content classification and its performance evaluation*», Pattern Recognition, 39(1) :57–73, 2006. (Cité pages 26, 29 et 39.)
- [22] G. Nagy and S. Seth, «*Hierarchical representation of optically scanned documents*». International conference on Pattern Recognition, 7 :347–349, 1984. (Cité pages 30 et 32.)
- [23] O.T. Akindele and A. Belaid, «*Page segmentation by segment tracing*», International Conference on Document Analysis and Recognition, 2 :341–344, 1993. (Cité page 31.)
- [24] R. BOUABBANA. M. BOURAOUI, « *Conception et réalisation d'un OCR multitaille* », Projet de fin d'étude, électronique, USDB, BLIDA 2009.
- [25] S.PHILIPP J.COCQUEREZ, « *Analyse d'image : filtrage et segmentation* », Éditions MASSON, 1995.
- [26] V.Wu and R. Manmatha, «*Document image clean-up and binarization*», Proceedings of IS&T/SPIE Symposium on Electronic Imaging, 3305 :263–273, 1998, (Cité pages 16 et 23.)
- [27] J. Bernsen, «*Dynamic thresholding of grey-level images*», In Proc. Eighth Int 'l Conf. on Pattern Recognition, pages 1251–1255, 1986. (Cité pages 17 et 23.)
- [28] N.Otsu. «*A threshold selection method from gray-level histograms*», IEEE Transactions on Systems, Man and Cybernetics, 9(1) :62–66, 1979. (Cité pages 20, 22 et 23.)
- [29] N. Otsu, «*A threshold selection method from grey scale histogram*», IEEE Trans. on Syst. Man and Cyber, vol 1, pp 62-66, 1979
- [30] K.Y. Wong, R.G. Casey, and F.M. Wahl. «*Document analysis system*», IBM Journal of Research and Development, 26(6) :647-656, 1982 (cité page 24 et 29).

ANNEXE

Annexe. 1. Différentes techniques de binarisation

Nom	Année	Type	Principe	Inconvénients
Otsu	1979	Seuillage global	D'après l'histogramme, cherche à maximiser la variance intra-classe du « texte » et du « fond ».	Problèmes pour les documents mal éclairés.
Bernsen	1986	Seuillage local	Estime la valeur du seuil en faisant la moyenne de la plus haute et la plus basse valeur de la fenêtre.	Le seuil est trop bas lorsque la fenêtre est centrée sur du fond.
Niblack	1986	Seuillage local	Amélioration de : prise en compte de la variance et de la moyenne.	Même problème que : apparition de bruit sur les zones uniformes.
Trier	1995	Seuillage local	« Ternarise » l'image en fonction du gradient puis utilise une heuristique pour réduire à deux classes.	L'utilisation de seuils définis empiriquement rend la méthode peu robuste.
Wu	1998	Seuillage global	« Floute » l'image pour mieux Séparer l'histogramme et utilise une méthode de seuillage global .	Problèmes lorsqu'il n'y a pas deux modes distincts sur l'histogramme.
Sauvola	2000	Seuillage local	Insère des constantes dans la méthode de afin d'améliorer la méthode sur les zones uniformes.	Les constantes à ajuster empêchent la méthode de traiter parfaitement des documents non uniformes.
Kim	2002	Seuillage local	Utilise la valeur des pixels comme courbes de niveaux pour simuler une montée des eaux.	Difficultés à adapter le débit de l'eau ou le nombre d'itérations au document.
Wolf	2002	Seuillage local	Utilise les champs de Markov pour savoir où se trouvent les caractères.	L'utilisation de rend la technique victime des

				mêmes limitations que pour Sauvola.
Garain	2005	Seuillage local	Utilise les composantes connexes pour créer un graphe d'adjacence qui est ensuite réduit.	Marche mal si l'image à traiter contient des illustrations.
Gatos	2006	Seuillage local	Cherche à estimer le fond pour ensuite faire un seuillage sur la différence entre le fond et l'image d'origine.	Très bonnes performances.

Annexe. 2. *Tableau récapitulatif des approches ascendantes.*

Nom	Année	Principe	Inconvénients
Wong	1982	Noirci les espaces blancs entre deux pixels noirs verticalement et horizontalement puis fait un « ET logique » entre les deux images noircies.	Nécessite une orientation horizontale du texte.
Antonacopoulos	1998	Utilise des tuiles sur le fond pour estimer les interlignes.	Très sensible au bruit de fond.
Messelodi	1999	Utilisation d'heuristiques pour classer les composants connexes.	La binarisation utilisée n'est pas très efficace.
Lienhart	2000	Accroissement de régions où les frontières se déplacent en fonction du gradient.	plus adapté à la segmentation de vidéos.
Wang	2006	Utilisation d'un vecteur de 69 caractéristiques, réduction à 23 par un algorithme de classification pour ensuite identifier le type de la boîte.	L'intérêt de la méthode n'est pas de segmenter mais de classifier.
Nicolas	2006	Utilisation des champs de Markov pour caractériser le texte.	L'apprentissage ne rend la méthode valable que pour un type de document à la fois.
Caponetti	2007	Utilise deux réseaux de neurones flou pour segmenter une image.	La phase d'apprentissage est très lourde à mettre en place en raison du type de réseau.

Annexe. 3. *Tableau récapitulatif des différentes approches descendantes*

Nom	Année	Principe	Inconvénients
Horowitz	1972	Commence par découper l'image en quatre, récursivement puis fusionne les zones de caractéristiques proches.	Nécessite une organisation horizontale de l'image.
Nagy	1984	Découpe l'image horizontalement puis verticalement, récursivement. Le découpage se fait dans le creux des projections.	Il existe des documents impossibles à segmenter.
Kim	1996	Utilise les pics de l'histogramme pour sélectionner les pixels de couleurs proches. Regroupe ensuite les composants proches et utilise des heuristiques pour classer les composantes.	L'utilisation de l'histogramme est trop générale pour donner de bons résultats.
Kim	1996	Réduit le nombre de couleurs en fonction des couleurs les plus proches. Regroupe ensuite les composantes de couleurs proches et utilise des heuristiques pour classer les composantes.	Pas de prise en compte spatiale des couleurs.