
IMPLÉMENTATION D'UN ALGORITHME EFFICACE DE SEUILLAGE D'IMAGE

MASTER EN SYSTÈME DE TRAITEMENT DE L'INFORMATION MULTIMEDIA

EDITER PAR

ACHRAF BEN SALAH

ENCADRÉ PAR

CAROLINE PETITJEAN



UFR DES SCIENCES ET TECHNIQUES SITE DU MADRILLET
UNIVERSITÉ DE ROUEN

2013–2014

Table des matières

1	Résumé et Introduction générale	11
1.1	Résumé	11
1.2	Introduction générale	11
2	Cadre de TER	13
2.1	introduction	13
2.2	Contexte	13
2.3	Contexte de travail	14
2.4	Objectifs	14
2.5	Organisation et plan	14
3	Etat de l'art	15
3.1	Introduction	15
3.2	Seuillage globale	15
3.2.1	Méthode d'Otsu	15
3.3	Seuillage local	16
3.3.1	Méthode de Sauvola	16
3.4	Conclusion	18
4	Expérimentation	19
4.1	Introduction	19
4.2	Base de document de validation	19
4.3	Expérimentations	20
4.4	Résultats d'évaluation	21

Liste des tableaux

4.1	Tableau d'évaluation des algorithmes de binarisation	22
4.2	Variation des taux de confiance en fonction des méthodes de binarisation	22

Table des figures

2.1	Comparaison entre la méthode Otsu et Sauvola	13
3.1	histogramme de niveaux de gris avec le seuil optimal	16
4.1	Exemples des images d'évaluation	20
4.2	verite terrain	27
4.3	Methode otsu	27
4.4	Methode sauvola	27
4.5	Methode Integral sauvola	27
4.6	Methode XORIntegralSauvola	28
4.7	Methode XOROtsu	28
4.8	Methode XORSauvola	28
4.9	verite terrain	28
4.10	Methode otsu	28
4.11	Methode sauvola	29
4.12	Methode Integral sauvola	29
4.13	Methode XORIntegralSauvola	29
4.14	Methode XOROtsu	29
4.15	Methode XORSauvola	29
4.16	verite terrain	30
4.17	Methode otsu	30
4.18	Methode sauvola	30
4.19	Methode Integral sauvola	30
4.20	Methode XORIntegralSauvola	30
4.21	Methode XOROtsu	30
4.22	Methode XORSauvola	31
4.23	verite terrain	31
4.24	Methode otsu	31
4.25	Methode sauvola	31
4.26	Methode Integral sauvola	31
4.27	Methode XORIntegralSauvola	32
4.28	Methode XOROtsu	32
4.29	Methode XORSauvola	32
4.30	verite terrain	32
4.31	Methode otsu	32
4.32	Methode sauvola	33
4.33	Methode Integral sauvola	33
4.34	Methode XORIntegralSauvola	33
4.35	Methode XOROtsu	33
4.36	Methode XORSauvola	33

Remerciements

Nous tenions à remercier tout particulièrement nos encadrants, Madame **Caroline PETITJEAN**. Elle est toujours été à notre écoute lors de nos différentes questions et nous aidé tout au long de ce TER.

Je saisis aussi cette occasion pour remercier les membres du jury d'avoir accepté de juger ce travail et j'espère qu'ils trouveront dans ce rapport les qualités de clarté et de motivation qu'ils attendent.

Ma gratitude pour tous les membres de la **Laboratoire de LITIS** et surtout Monsieur **Laurent HEUTTE** pour la bonne tout au long de ce projet.

Chapitre 1

Résumé et Introduction générale

1.1 Résumé

Ce rapport présente une étude sur les performances des algorithmes de binarisation de Sauvola et d'Otsu. L'algorithme de binarisation d'Otsu utilise une procédure de seuillage globale comme Otsu qui assure la conversion des images en niveau de gris à des images binaires. Pour obtenir les mêmes résultats, l'algorithme de Sauvola utilise une procédure de seuillage locale.

Nous commençons notre étude par la présentation de l'algorithme de méthode de seuillage. Ensuite, nous montrons l'intérêt de l'utilisation de l'algorithme de l'intégral de Sauvola qui est une amélioration de l'algorithme classique de Sauvola. Enfin, nous concluons notre rapport par une évaluation qualitative et quantitative des résultats de chaque algorithme.

Mots-clés : Seuillage, Binarisation, Sauvola, Otsu.

1.2 Introduction générale

La binarisation des images représente une étape cruciale dans la réalisation de plusieurs applications de reconnaissance de forme. La plupart des systèmes de reconnaissance optique des caractères utilisent généralement des images binaires pour extraire les caractéristiques des formes des caractères.

Plusieurs travaux de recherche ont traité le problème de binarisation. Cependant, il n'existe pas jusqu'à aujourd'hui une approche générique qui peuvent être appliquée à tous les types d'image. En effet, généralement le choix des approches de binarisation est fortement lié au contexte des images et aux conditions des prises de vue. La difficulté essentielle de l'opération de binarisation réside dans le choix de seuil optimal : celui qui réalise le meilleur compromis entre réduction de bruit et amélioration de lisibilité des formes d'éléments dans l'image.

D'autre part, l'opération de binarisation est très sensible à réaliser. En effet, la quantité d'information à traiter après l'application de la binarisation est réduite de manière importante. Le risque de perte d'information est grand dans la mesure où une mauvaise binarisation risque de supprimer une partie de l'information utile et d'introduire du bruit sur l'image.

De plus les résultats de cette opération sont utilisés généralement par d'autre couche de traitement de niveau supérieur pour réaliser des tâches bien définies. Par conséquent, la qualité des résultats de ces traitements est fortement liée à la qualité des résultats de l'opération de binarisation.

Pour comprendre la procédure de binarisation des images. Nous avons étudié dans ce travail deux algorithmes de binarisation qui ont été largement utilisés dans la littérature des méthodes de traitement d'image. Ces algorithmes sont l'algorithme d'Otsu et l'algorithme de Sauvola.

Dans la première partie de rapport, nous allons présenter le **cadre de notre travail** dans lequel nous présentons l'organisme d'accueil et l'objectif de notre travail. Dans la deuxième partie **l'Etat de l'Art** nous présentons les algorithmes de binarisation d'Otsu et de Sauvola. Ensuite, nous consacrons la troisième partie pour évaluer les résultats de chaque algorithme en utilisant plusieurs exemples d'images. Et finalement nous concluons ce rapport par une conclusion générale et les perspectives d'amélioration de notre travail.

Chapitre 2

Cadre de TER

2.1 introduction

Il est difficile de parler d'un projet avant d'avoir fait une analyse du travail à réaliser. C'est la raison pour laquelle nous allons présenter dans cette partie le cadre de notre travail ainsi que les objectifs que nous voulons atteindre.

2.2 Contexte

La binarisation des images est un thème pluridisciplinaire qui fait l'objet de nombreux travaux de recherches. Ce domaine s'intéresse à la réalisation d'un algorithme de seuillage qui permet de produire des images binaires proches de la réalité. Dans la littérature, il existe un ensemble des techniques de seuillage automatique, nous pouvons décomposer ces méthodes en deux méthodes :

- Les méthodes de seuillage globales : Otsu, Li-Lee, Cheng et Chen.
- Les méthodes locales comme : Sauvola, Bernsen, Niblack.

Dans le cadre de ce travail, nous avons étudié l'algorithme d'Otsu et de Sauvola. Nous comparons également les performances de chaque algorithme en utilisant une base d'image composée par des images de pages de documents et des images naturelles. La base des images de documents sont composées par des images en niveau de gris. Ces images sont caractérisées par des défauts images (bruit de numérisation, défaut de contraste) et physiques (défauts d'encrage, défauts de transparence). L'ensemble de ces problèmes influe sur la qualité des résultats de ces algorithmes ce qui cause parfois des erreurs de binarisation. D'autre part, les résultats de l'opération de binarisation sont utilisés généralement par d'autre couche de traitement pour assurer une opération de reconnaissance de forme par exemple. Par conséquent, les défauts de binarisation ont un impact direct sur les résultats de ces traitements. Pour s'assurer de ce constat nous évaluons ces deux algorithmes en utilisant un système de reconnaissance optique de caractère « **Abby Fine Reader version 8.0** ». Ce travail peut être considéré comme une étude préliminaire qui nous a permis d'une part de comprendre les deux algorithmes testés et d'autre part d'avoir une idée sur les pistes de leur amélioration en fonction du contexte de nos traitements. L'ensemble des algorithmes développés dans le cadre de ce TER sont présentés dans <http://achrafbensalah.webs.com/>.

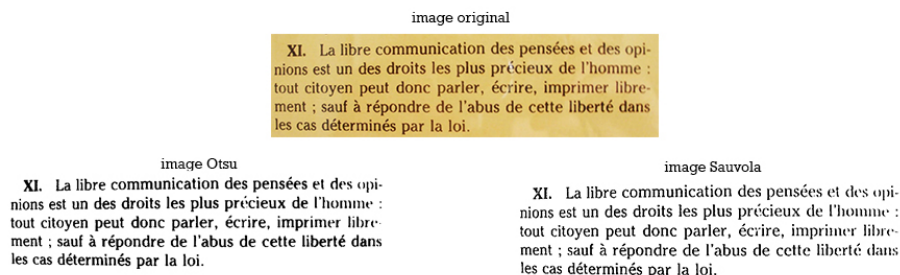


FIGURE 2.1 – Comparaison entre la méthode Otsu et Sauvola

2.3 Contexte de travail

Cette étude a été effectuée dans l'équipe "Document et Apprentissage (DocApp)" de laboratoire "LITIS" de Rouen dont un axe de recherche concerne le domaine de traitement d'image et plus spécifiquement les méthodes de binarisation des images de document. Dans le but d'implémenter les algorithmes d'Otsu et de Sauvola et de déterminer leurs avantages et inconvénients.

2.4 Objectifs

L'objectif de ce travail est d'analyser l'algorithme d'Otsu et de Sauvola pour comprendre ses modes de fonctionnement. Dans un deuxième temps, nous essayons de déterminer les avantages et les inconvénients de chaque algorithme afin de déterminer leurs champs d'application. Les propriétés que nous cherchons à vérifier sont le temps de traitement et le taux d'erreurs de binarisation. Pour atteindre nos objectifs, nous avons réalisé les étapes que suivantes :

- Implémentation des algorithmes de binarisation d'Otsu et de Sauvola,
- Construction de la vérité terrain des images binaires,
- Evaluation des algorithmes de binarisation sur des images des documents et des images naturelles afin de calculer leurs performances au niveau pixels.
- Détermination de l'effet des résultats de binarisation de ces algorithmes sur les résultats des systèmes de reconnaissance optique des caractères.

2.5 Organisation et plan

Ce rapport suit un schéma respectant dans les grandes lignes l'évolution chronologique de notre travail. Dans un premier temps, nous présenterons les algorithmes de binarisation d'Otsu et de Sauvola. Dans le chapitre 2, nous exposerons la procédure d'évaluation de ces deux algorithmes ainsi que ses performances. En conclusion, nous finissons ce rapport par une conclusion générale dans laquelle nous essayons de présenter les avantages et les inconvénients de l'algorithme d'Otsu et de Sauvola ainsi que les perspectives d'amélioration de chaque algorithme.

Chapitre 3

Etat de l'art

3.1 Introduction

Dans la littérature [AY01] [KSFV09], le sujet de la binarisation des images fait l'objet de plusieurs travaux de recherche. Nous pouvons décomposer ces travaux en deux catégories : seuillage globale, où un seul seuil est utilisé dans toute l'image pour diviser en deux classes (avant-plan et arrière-plan), et seuillage local où les valeurs des seuils sont déterminées localement, pixel par pixel ou bien région par région. D'autres approches essaient de combiner des informations globales et locales pour développer des approches mixtes pour attribuer les pixels à l'une des deux classes.

Certain état de l'art décompose les approches de binarisation des images en six catégories : méthodes basées sur la forme de l'histogramme, sur le clustering, sur l'entropie, sur les attributs d'objet, méthodes spatiales et méthodes locales.

Nous présentons dans cet état de l'art deux approches binarisation très utilisées dans la littérature. Le premier algorithme est la méthode de seuillage d'Otsu qui appartient à la famille des approches de binarisation globales. Le deuxième algorithme est la méthode de seuillage de Sauvola qui appartient à la famille des approches de binarisation locales.

3.2 Seuillage globale

3.2.1 Méthode d'Otsu

La méthode d'Otsu est algorithme de seuillage globale d'image qui essaye d'optimiser la valeur de seuil de binarisation T qui maximise la différence entre deux classes. Cette méthode utilise généralement l'histogramme normalisé des niveaux de gris pour calculer les variances inter-classes ou intra-classes.

Par conséquent, les étapes de réalisation de cette approche sont les suivantes :

- Calcul l'histogramme de l'image.
- Calcul la valeur moyenne et la variance de chaque classe.
- Fixation du seuil de binarisation de l'image

L'histogramme de l'image consiste à répartir à répartir chaque intensité de couleurs dans l'image en fonction de nombre d'apparition de chaque pixel dans un histogramme. Les histogrammes de niveau de gris que nous utilisons sont en général normalisés. Les valeurs des intensités de l'histogramme varient entre 0 et 1. Cette valeur peut être considérée comme une probabilité d'occurrence de la classe niveau de gris dans l'image.

La probabilité d'occurrence de niveau de gris X_k est définie par l'équation suivante :

$$p(X = X_k) = \frac{N_k}{N_t} \quad (3.1)$$

Avec X_k et le niveau de gris de l'image, N_k est le nombre d'occurrence d'un pixel au niveau X_k et N_t nombre total des pixels dans l'image.

A partir de cet histogramme normalisé, nous avons fixé le seuil de binarisation T de l'image I . Selon [Ots79], la méthode d'Otsu permet de binariser l'image dans laquelle la séparation entre deux classes se fait selon la calcul de la valeur de variance intra-classe a partir la calcul du poids, la valeur moyenne et l'écart type a chaque nouveaux deux classes.

La variance intra-classe est donnée par :

$$\sigma_{intra}^2 = W_b \times \sigma_b^2 + W_f \times \sigma_f^2 \quad (3.2)$$

Le poids de chaque classe est donnée par :

$$W = \frac{1}{N} \sum_{i=0}^N \text{Histog}(i) \quad (3.3)$$

La valeur moyenne de chaque classe est donnée par :

$$\text{moy} = \frac{1}{N} \sum_{i=0}^N i \times I(i) \quad (3.4)$$

L'écart type de chaque classe est donnée par :

$$\sigma = \frac{1}{N} \sum_{i=0}^N (i - \text{moy})^2 \quad (3.5)$$

Avec :

Histog : histogramme de l'image.

i : indique l'indice des distributions des pixels selon l'histogramme.

moy : La valeur moyenne des pixels.

N : Nombre total de distributions des pixels pour les deux classes dans l'image.

Cette méthode est simple à implanter et donne généralement de bons résultats en général. Cependant, la limite de cette méthode est qu'elle n'est applicable que lorsque l'image est bimodale (l'histogramme comporte deux pics). Pour résoudre ce genre de problème, plusieurs algorithmes dans la littérature ont été proposés. Par exemples, certains travaux ont porté sur l'optimisation de la valeur du seuil « T » la méthode de vallée accentuée (Valley-emphasis method) [HFN06], la méthode de vallée accentuée modifiée [JLF12]. D'autres méthodes ont recouru à des procédures de binarisation locale qui permet de choisir plusieurs seuils de binarisation d'image. Dans la section suivante, nous présentons l'algorithme de Sauvola qui est une approche de binarisation qui appartient à la famille des méthodes de seuillage local.

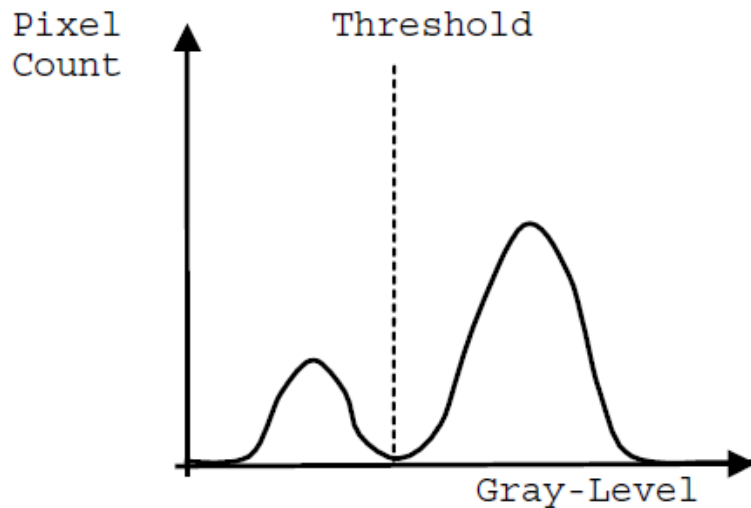


FIGURE 3.1 – histogramme de niveaux de gris avec le seuil optimal

3.3 Seuillage local

3.3.1 Méthode de Sauvola

Les méthodes de seuillage globale ne réalisent pas des bonnes performances sur des sources bruitées ou mal contrastées. Par exemple si l'encrage de la page n'est pas assez foncé, le seuillage global peut entraîner la

disparition de l'information pertinent (comme du texte) ce qui biaise énormément les résultats des systèmes d'analyse des d'image.

Par conséquent, il devient nécessaire d'appliqué des techniques de seuillage adaptatif qui permet de traiter les zones homogènes de l'image en variant la valeur du seuil de rejet.

Dans les approches locales, nous utilisons plusieurs seuils de binarisation pour classer les pixels en deux classes (arrière-plan et avant-plan). Les valeurs des seuils sont déterminées en fonction des caractéristiques des pixels et des régions voisins.

Plusieurs travaux de recherche ont été menés sur la réalisation des approches de seuillage multiple. L'algorithme de binarisation les plus connus de cette famille et l'algorithme de Sauvola.

Cette méthode de binarisation utilise la technique de fenêtre glissante centrée réduite pour déterminer la valeur du seuil de binarisation. La taille de cette fenêtre d'étude est variable en fonction de la taille moyenne des objets dans l'image. Par exemple, pour les images des pages, nous pouvons choisir la taille moyenne des caractères pour fixer la taille de fenêtre glissante de Sauvola.

Ensuite, nous procédons aux calculs de l'intensité moyenne des pixels $\mu(i, j)$ ainsi que la valeur de l'écart type de niveau de gris $\sigma(i, j)$ qui sont inclus dans la fenêtre centré en (i, j) de taille $N \times M$.

Enfin en utilisant l'équation au-dessous, nous essayons de calculer la valeur du seuil de binarisation.

$$T(i, j) = \mu(i, j) \times \left(1 + k \times \left(\left(\frac{\sigma(i, j)}{R} \right)^2 - 1 \right) \right) \quad (3.6)$$

Avec $k = 0.5$ et $R = 128$. La valeur de R est un constant utilisé généralement pour ajuster le dynamique de l'écart type. La moyenne et l'écart type des intensités de niveau de gris ne sont pas les seuls paramètres que nous utilisons pour choisir la valeur de seuil T . En effet, certains travaux de recherche utilisent d'autres critères tels que le nombre maximum des pixels voisins sur une ligne ayant les mêmes caractéristiques [YY00], la représentation probabiliste du bruit [Don95], les composants connexes de l'image pour identifier les éléments d'avant-plan et d'arrière-plan [GPH05]. Dans cette étude, nous allons traiter l'algorithme classique de Sauvola et l'algorithme de l'intégrale de Sauvola.

Par conséquent, dans notre travail, nous avons décomposé la méthode Sauvola par les étapes suivantes :

- Division de l'image en sous images à l'aide de dimension de la fenêtre (3×3).
- Binarisation de chaque sous image.

L'utilisation du seuillage de local permet de réduire le bruit de l'image et d'améliorer la représentation du détail de l'image. Généralement, il réalise des meilleures performances sur les images mal contrastées que les approches de seuillage globales. Par contre, le grand défaut de ces approches réside dans le temps de traitement important causé généralement par l'utilisation de la fenêtre glissante.

Pour résoudre ce problème, [PVMJ04] propose une amélioration de l'algorithme classique d'Otsu qui consiste à utiliser la notion de l'image intégrale pour accélérer la procédure de binarisation d'image.

L'image intégrale d'une image en niveaux de gris peut être efficacement calculée en un seul passage. En utilisant l'équation suivante :

$$\forall i \in [0M] \text{ et } \forall j \in [0N], I(x, y) = \sum_{i=0}^x \sum_{j=0}^y g(i, j) \quad (3.7)$$

La valeur de la moyenne $m(x, y)$ pour chaque fenêtre peut être calculée simplement en utilisant deux additions et une des opérations de soustraction au lieu de la sommation sur toutes les valeurs de pixels à l'intérieur de cette fenêtre. La valeur de moyenne des intensités de pixels englobés dans une fenêtre glissante de taille $w \times w$ est définie par la fonction suivante :

$$m(x, y) = (I(x + w/2, y + w/2) + I(x - w/2, y - w/2) - I(x + w/2, y - w/2) - I(x - w/2, y + w/2)) / w^2 \quad (3.8)$$

De même, si l'on considère le calcul de la variance local :

$$S^2(x, y) = \frac{1}{w^2} \sum_{i=x-w/2}^{x+w/2} \sum_{j=y-w/2}^{y+w/2} g^2(i, j) - m^2(x, y) \quad (3.9)$$

L'utilisation de l'image intégrale permet donc de réduire la complexité de l'algorithme binarisation à partir la diminution de temps du calcul. Ceci permet de rendre le temps de traitement de cet algorithme proche du temps de traitement de l'algorithme d'Otsu.

3.4 Conclusion

Nous avons présenté dans cette section deux algorithmes de binarisation qui sont l'algorithme d'Otsu et l'algorithme de Sauvola. Nous avons également décrit les avantages et les inconvénients de chaque approche. Pour s'assurer de ces constats, nous allons réaliser dans le chapitre suivant une procédure d'évaluation de chaque approche afin de vérifier leurs performances sur une base d'images de document. Ensuite, nous allons tester l'effet des résultats de binarisation des images sur les résultats de reconnaissance des caractères.

Chapitre 4

Expérimentation

4.1 Introduction

Dans cette partie, nous avons réalisé un ensemble d'expérimentations qui nous a permis de déterminer les performances des algorithmes d'Otsu et de Sauvola. Pour cela, nous avons employé sept images de document numérisées en niveau de gris et une image naturelle en couleur.

Dans un premier temps, nous avons évalué les performances des algorithmes d'Otsu et de Sauvola en niveau pixels en quantifiant les différences qui résident entre les résultats de ces algorithmes et la vérité terrain des résultats de binarisation. Ensuite, nous avons utilisé un système de reconnaissance optique des caractères pour reconnaître les textes des images des documents issues des différents algorithmes de binarisation étudiés.

Nous commençons donc cette partie par la présentation des caractéristiques d'images d'évaluation ainsi du protocole d'évaluation que nous avons adopté dans cette étude. Ensuite, nous présentons les résultats qualitatifs et quantitatifs obtenus lors de la réalisation de ces expériences. Nous finissons cette partie par une conclusion.

4.2 Base de document de validation

Dans notre processus d'évaluation nous avons utilisé sept images de documents sélectionnées de manière aléatoire à partir de sept documents différents. La figure 3.1 présente les images d'évaluation que nous avons utilisées. D'après cette figure, nous constatons la présence de certains défauts physiques et images sur ses exemples.

Certaines images sont caractérisées par des propriétés physiques particulières. Par exemple, l'image 3.1.a représente une page de garde caractérisée par la présence des défauts de contraste et de bruit de numérisation. Les exemples 3.1.b-c représentent des images de pages caractérisées par des défauts d'encrage comme la présence des tâches d'encre et des chevauchements entre le tampon et les régions textuelles. L'exemple 3.1.d présente des défauts de transparence.

D'autres images sont caractérisées par la présence simultanée des éléments graphiques et textuels caractérisés par des niveaux de gris différents. L'exemple 3.1.e illustre un exemple de carte dans lequel nous constatons la présence de quelques défauts physiques tels que des déchirures de papier et des plis de papier. L'exemple 3.1.f représente un document technique caractérisé par un défaut de contraste de côté gauche de l'image du au pli du dos document.

Nous avons également testé l'algorithme de binarisation d'Otsu et de Sauvola sur une image naturelle en couleur pour déterminer ses comportements sur ce genre d'image.

Pour chaque image de notre base de validation, nous avons produit une image de vérité terrain qui nous sert comme référence absolue pour vérifier les décisions de chaque approche. Pour produire ce genre d'image, nous avons utilisé Photoshop CS6 pour colorer les pixels de l'image de niveau de gris. Nous avons également utilisé une stratégie d'annotation qui nous a permis de représenter les pixels textuels et graphiques en noir (valeur 0) et les pixels de papier en blanc (valeur 1). Les résultats de cette procédure sont présentés dans la figure 3.2.

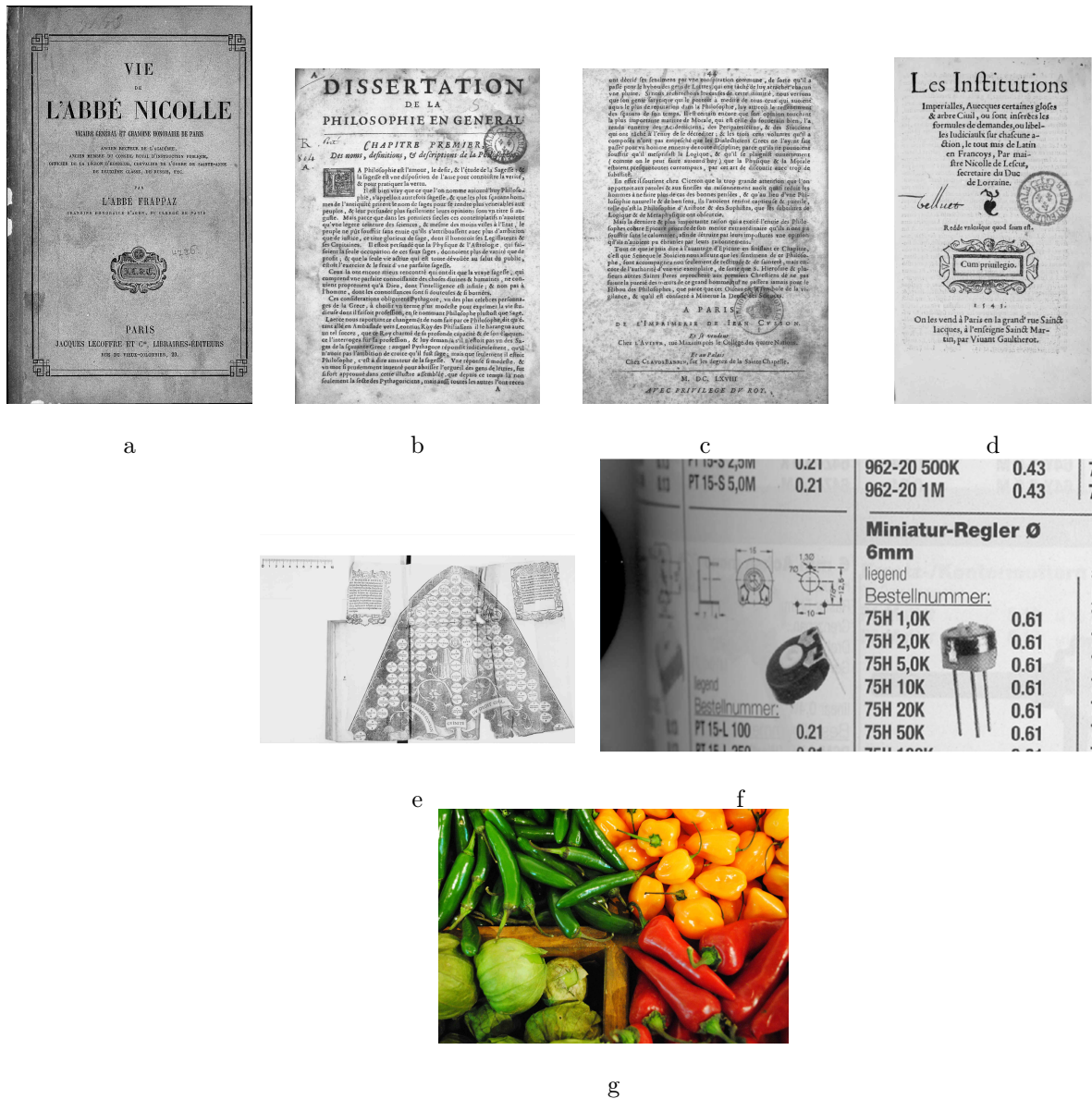


FIGURE 4.1 – Exemples des images d’évaluation

4.3 Expérimentations

Comme nous avons évoqué dans le premier chapitre l’objectif de l’opération de binarisation des images des documents est de produire des images qui conservent au maximum l’information utile. Dans notre procédure d’évaluation, l’information utile est représentée par les éléments textuels des images. Par conséquent, plus les pixels des caractères sont conservés plus la procédure de binarisation est performante.

D’autre part, nous avons montré aussi que la procédure de binarisation d’image est employée généralement par d’autre algorithme afin de réaliser des tâches de niveau supérieur. Les systèmes de reconnaissance optique des caractères utilisent des images binaires afin de normaliser les caractères des images de la page. Ceci permet d’unifier les descriptions des formes des caractères. Par conséquent, les défauts de binarisation d’images ont un impact direct sur les résultats des autres traitements.

Dans notre étude, nous avons adopté donc deux protocoles d’expériences :

- le premier protocole permet de contrôler les résultats de classification des pixels en classe arrière-plan et avant-plan.
- le deuxième protocole permet de vérifier la qualité des résultats d’OCR en utilisant les images binaires issue de l’opération de binarisation de trois méthodes de seuillage.

Nous avons commencé notre première expérimentation par l’application des trois algorithmes de binarisation sur les images de niveau de gris. Cela permet de nous fournir trois images binaires pour chaque image

en niveau de gris.

Ensuite, en utilisant l'opérateur logique « X-or » nous avons déterminé les pixels mal classés par les méthodes de binarisation. Les résultats de cette opération sont des images binaires. Les pixels blancs représentent les erreurs de binarisation des images. Les pixels noirs représentent les cas où on a une superposition entre les classes des pixels de l'image binarisée et les classes des pixels de l'image vérité terrain.

Enfin, pour qualifier quantitativement les performances de chaque méthode de binarisation, nous avons compté le nombre des pixels blanc dans l'image des résultats d'alignement des deux images. Puis, en utilisant la fonction 4.1, nous avons calculé le taux d'erreur de binarisation :

$$tauxMoyenBinar = \frac{nombre_total_erreur}{nombre_total_pixel} \quad (4.1)$$

Dans le deuxième protocole de binarisation, nous avons commencé aussi par l'application des trois algorithmes de binarisation de notre étude afin de produire pour chaque image en niveau de gris trois images binaires.

Ensuite, en utilisant les images binaires, nous avons essayé de reconnaître les caractères inscrits dans ces images. Pour cela, nous avons utilisé un système d'OCR « ABBY Fine Reader 11 » qui permet de nous fournir des résultats de transcription de l'image ainsi que des scores de confiance sur les résultats de reconnaissance des pages.

Ces scores de confiance peuvent donner une indication sur la qualité des résultats d'OCR obtenus lors de l'utilisation des images binaires produites à partir de l'algorithme d'Otsu, de Sauvola et de Sauvola image intégrale.

4.4 Résultats d'évaluation

Les images issues de l'opération d'alignement entre les images binaires produites à travers les algorithmes d'Otsu, de Sauvola et de Sauvola image intégrale sont présentées dans les figures 4.7, 4.8, 4.9, 4.14, 4.15, 4.16, 4.21, 4.22 et 4.23. D'après ces figures, nous constatons que les résultats de binarisation d'Otsu causent le plus grand nombre d'erreur de binarisation. En effet, dans les trois exemples, nous constatons qu'il y a plus de pixels blancs dans les résultats d'alignement obtenus avec les images binaires de l'algorithme d'Otsu que dans les images d'alignement obtenus avec les autres algorithmes de binarisation.

Ce constat est vérifié quantitativement à travers les taux d'erreur de binarisation inscrits dans le tableau 3.1. En effet, dans la majorité des cas étudiés le plus faible taux de biais de binarisation est obtenu avec les deux algorithmes de Sauvola. Les images issues de l'opération d'alignement entre les images binaires produites à travers les algorithmes d'Otsu, de Sauvola et de Sauvola la image intégrale sont présentées dans les figures 3.7, 3.8, 3.9, 3.14, 3.15, 3.16, 3.21, 3.22 et 3.23. D'après ces figures, nous constatons que les résultats de binarisation d'Otsu causent le plus grand nombre d'erreur de binarisation. En effet, dans les trois exemples, nous constatons qu'il y a plus de pixels blancs dans les résultats d'alignement obtenus avec les images binaires de l'algorithme d'Otsu que dans les images d'alignement obtenus avec les autres algorithmes de binarisation.

Ce constat est vérifié quantitativement à travers les taux d'erreur de binarisation inscrits dans le tableau 3.1. En effet, dans la majorité des cas testés les plus faibles taux de biais de binarisation sont obtenus avec les résultats de binarisation de Sauvola (intégrale, classique). L'algorithme d'Otsu arrive à des taux de biais de binarisation inférieurs à celui d'Otsu sur les images 4.1.e et 4.1.g. ceci peut être dû aux caractéristiques physiques de papier qui rend l'opération de binarisation de ces images très difficile. Par contre, même s'il réalise les plus faibles taux de biais de binarisation sur ces exemples, les taux d'erreurs sont très importants. Cela signifie que même la méthode d'Otsu n'est pas adaptée à la binarisation de ce genre d'images.

Le temps de traitement de l'algorithme d'Otsu est très rapide alors que l'algorithme de Sauvola prend beaucoup plus de temps pour binariser les images exprimées en niveau de gris. Par contre, nous remarquons aussi que l'utilisation de l'image intégrale a bien accélérée les traitements de l'algorithme de Sauvola. Ceci vient conforme avec les constats que nous avons évoqués dans le chapitre état de l'art.

Méthodes	Evaluation	Numéro des images						
		4.1.a	4.1.e	4.1.c	4.1.d	4.1.b	4.1.f	4.1.g
Otsu	Taux d'erreur	88.08%	6.10%	4.41%	3.30%	14.72%	20.08%	16.71%
	Temps de calcul (sec)	0.1581	0.1325	0.1361	0.1507	0.1653	0.1421	1.1858
Sauvola	Taux d'erreur	44.04%	24.07%	3.10%	2.90%	5.53%	7.31%	21.52%
	Temps de calcul (sec)	141.71	153.1069	158.6938	206.4021	143.9839	70.3464	1500.07
Integral Sauvola	Taux d'erreur	45.04%	29.07%	2.72%	2.47%	5.04%	4.37%	20.94%
	Temps de calcul (sec)	10.78	22.7305	24.8099	34.2614	21.7223	5.9683	379.6662

TABLE 4.1 – Tableau d'évaluation des algorithmes de binarisation

D'autre part, le tableau 4.2 représente les taux de confiance réalisés par un système d'OCR sur les résultats de binarisation des exemples de la figure 4.1. D'après ce tableau les meilleurs valeurs de taux de confiance sont obtenues toujours avec les résultats de binarisation de l'algorithme de Sauvola ceci exprime bien l'intérêt de l'utilisation de cette méthode sur les images des documents.

	4.1.a	4.1.c	4.1.d
Taux de confiance Otsu	60.5	72.1	53.8
Taux de confiance Sauvola	66.2	75.4	56.1
Taux de confiance Integral Sauvola	66.5	75.9	57

TABLE 4.2 – Variation des taux de confiance en fonction des méthodes de binarisation

Conclusion générale

La procédure de binarisation d'image assure la transformation des images en niveaux de gris, issue d'un scanner ou d'un appareil photo, en une image binaire. Dans la littérature, plusieurs systèmes de reconnaissance de formes utilisent des images binaires pour séparer les objets d'avant-plan de l'arrière-plan. Par conséquent, les résultats de cette procédure ont une influence directe sur les résultats des systèmes de reconnaissance.

Dans la littérature plusieurs méthodes ont été proposées pour traiter le problème de binarisation des images des documents. Nous pouvons distinguer deux familles d'approche. Les approches globales qui essaient d'optimiser la valeur d'un seuil de binarisation en traitant la totalité de l'image et les approches locales qui choisissent plusieurs valeurs de seuils afin de réaliser l'opération de binarisation de l'image. Dans ce TER, nous avons étudié deux méthodes de binarisation qui appartiennent à ces deux familles. La première méthode est l'algorithme d'Otsu et la deuxième méthode est l'algorithme de Sauvola. De plus, nous avons analysé les résultats de la méthode de Sauvola qui utilise l'image intégrale pour accélérer le temps de traitement de cet algorithme. Les résultats d'évaluation ont montré que l'algorithme d'Otsu est très rapide par rapport aux algorithmes de Sauvola que nous avons traités dans cette étude. Par contre, les algorithmes de Sauvola réalisent des performances plus meilleures que celles de l'algorithme d'Otsu. En effet, les taux de biais de binarisation obtenus avec la méthode de Sauvola sont inférieurs au taux de biais obtenus avec la méthode de binarisation d'Otsu.

D'autre part, nous avons remarqué aussi que les résultats de binarisation ont un impact direct sur la qualité des résultats de reconnaissance de caractère. En effet, les meilleurs taux de confiance sont obtenus sur les images binaires produites à travers l'algorithme Sauvola.

A partir de cette analyse nous pouvons déduire que l'utilisation de l'algorithme de Sauvola est très bénéfique sur des images des documents anciens caractérisées par des défauts physiques. Ces défauts biaisent énormément le contraste des images ce qui rend difficile la binarisation de l'image en utilisant uniquement un seul seuil. Ce travail est une étude préliminaire qui nous a permis de comprendre les principes des algorithmes de binarisation globaux et locaux. Dans les perspectives de ce travail, nous pouvons essayer de développer une approche mixte qui permet de bénéficier des avantages des approches locales et globales.

Bibliographie

- PVMJ04 P. Viola and M. J. Jones, "Robust real-time face detection", Int. Journal of Computer Vision 57(2), pp. 137â154, 2004.
- SP00 J. Sauvola and M. Pietikainen, "Adaptive document image binarization," Pattern Recognition 33(2),pp. 225â236, 2000.
- Ots79 N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Systems, Man, and Cybernetics 9(1), pp. 62â66, 1979.
- Nib86 W. Niblack, " An Introduction to Digital Image Processing ", Ed. Prentice Hall, Englewood Cliffs, 1986.
- F84 Franklin C., "Summed-Area Tables for Texture Mapping," Crow Computer Sciences Laboratory Xerox Palo Alto Research Center Computer Graphics Volume 18, Number 3 July 1984.
- BTJ11 Bilal KARABAGLI, Thierry SIMON, Jean-José ORTEU ., "Détermination automatique du seuil de binarisation des modules des gradients par modélisation de leur histogramme," ,46100 FIGEAC, France, 2011.
- ATM00 Abderrahmane Kefali, Toufik Sari, et Mokhtar Sellami, "EVALUATION DE PLUSIEURS TECHNIQUES DE SEUILLAGE DâIMAGES DE DOCUMENTS ARABES ANCIENS," ,Département Informatique, BP 12 - Université Annaba, 23000, Algérie.
- FDT09 Faisal Shafaita, Daniel Keysersa, Thomas M. Breuelb, "Efficient Implementation of Local Adaptive Thresholding Techniques Using Integral Images," ,University of Kaiserslautern D-67663 Kaiserslautern, Germany.
- Y.YANG and H.YAN., "An Adaptive logical method for binarization of degraded document image", Pattern Recognition, 788-807,2000.
- DON95 H.S.DON "A noise attribut thresholding method for document image binairization", Icdar, 1995.
- IJPH05 U.GRAIN, T.Paquet and L.Heutte "An forground background seperation low quality color document images", Icdar 2005.

Annexes

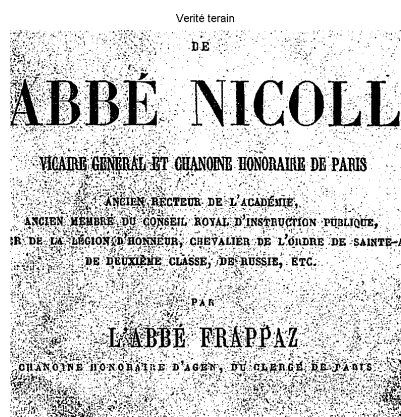


FIGURE 4.2 – verite terrain

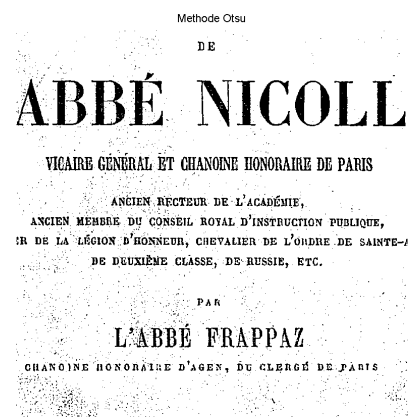


FIGURE 4.3 – Methode otsu

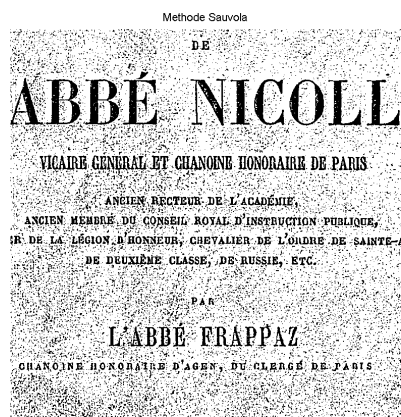


FIGURE 4.4 – Methode sauvola

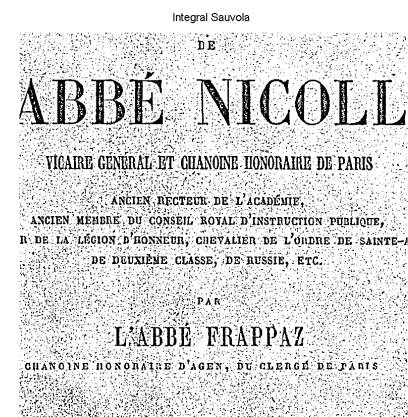


FIGURE 4.5 – Methode Integral sauvola

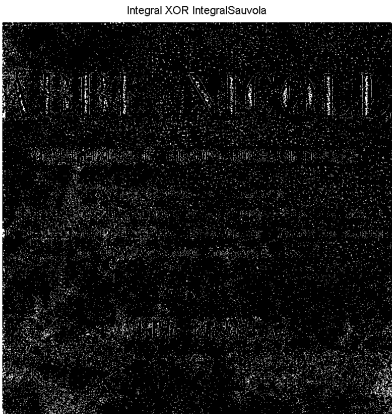


FIGURE 4.6 – Methode XORIntegralSauvola

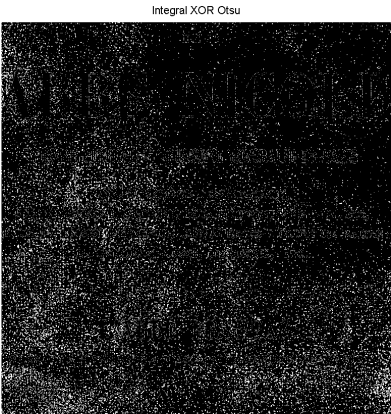


FIGURE 4.7 – Methode XOROtsu

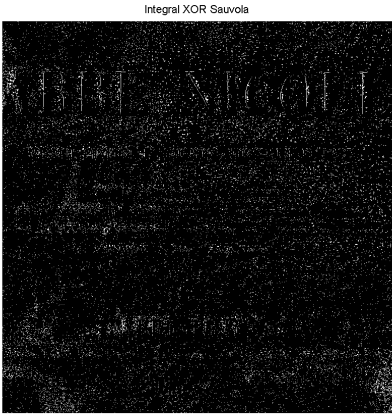


FIGURE 4.8 – Methode XORSauvola

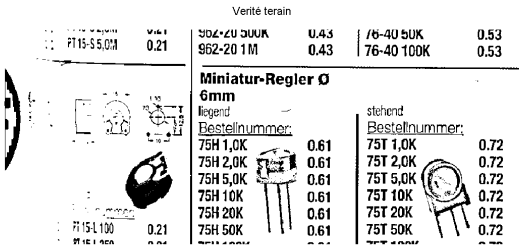


FIGURE 4.9 – verite terrain

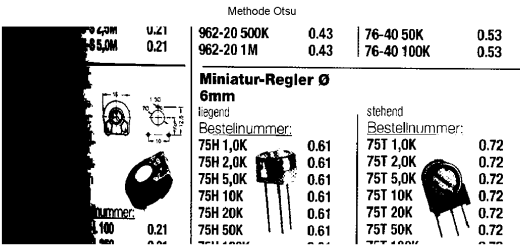


FIGURE 4.10 – Methode otsu

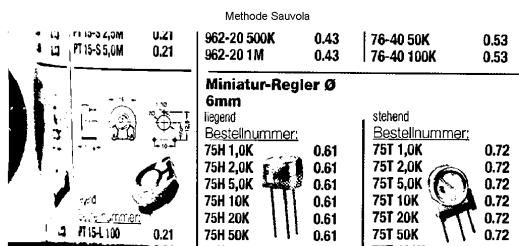


FIGURE 4.11 – Methode sauvola

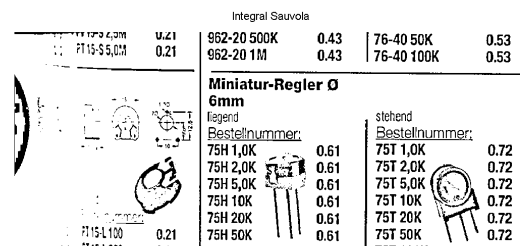


FIGURE 4.12 – Methode Integral sauvola

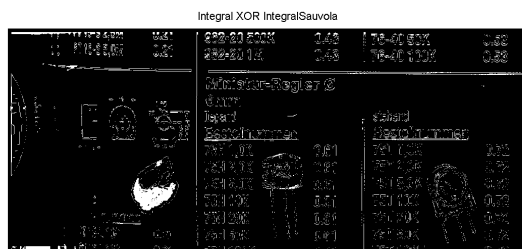


FIGURE 4.13 – Methode XORIntegralSauvola

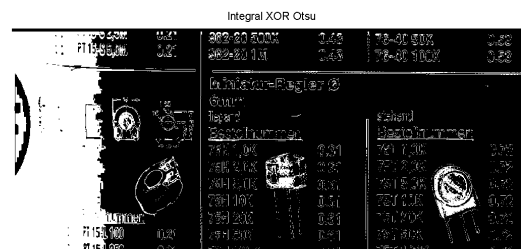


FIGURE 4.14 – Methode XOROtsu

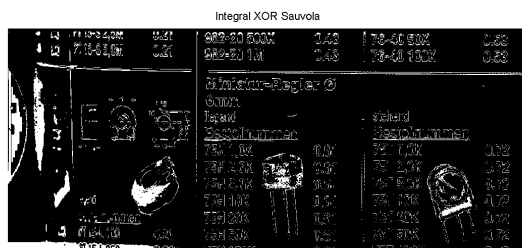


FIGURE 4.15 – Methode XORSauvola

Verite terrain

Les Institutions

Imperialles, Auecques certaines gloses
& arbre Ciuil, ou font inferées les
formules de demandes, ou libel-
les Iudiciaulx sur chascune a-
ction, le tout mis de Latin
en Francoys, Par mai-
stre Nicolle de Lescut,
secretaire du Duc
de Lorraine.

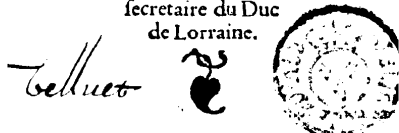


FIGURE 4.16 – verite terrain

Methode Otsu

Les Institutions

Imperialles, Auecques certaines gloses
& arbre Ciuil, ou font inferées les
formules de demandes, ou libel-
les Iudiciaulx sur chascune a-
ction, le tout mis de Latin
en Francoys, Par mai-
stre Nicolle de Lescut,
secretaire du Duc
de Lorraine.



FIGURE 4.17 – Methode otsu

Methode Sauvola

Les Institutions

Imperialles, Auecques certaines gloses
& arbre Ciuil, ou font inferées les
formules de demandes, ou libel-
les Iudiciaulx sur chascune a-
ction, le tout mis de Latin
en Francoys, Par mai-
stre Nicolle de Lescut,
secretaire du Duc
de Lorraine.



FIGURE 4.18 – Methode sauvola

Integral Sauvola

Les Institutions

Imperialles, Auecques certaines gloses
& arbre Ciuil, ou font inferées les
formules de demandes, ou libel-
les Iudiciaulx sur chascune a-
ction, le tout mis de Latin
en Francoys, Par mai-
stre Nicolle de Lescut,
secretaire du Duc
de Lorraine.

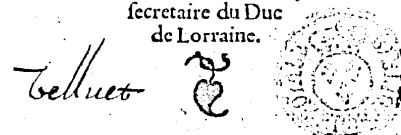


FIGURE 4.19 – Methode Integral sau-
vola

Integral XOR IntegralSauvola

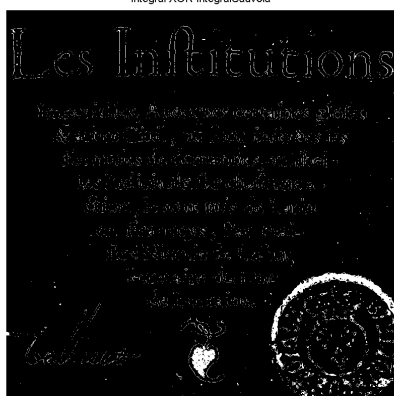


FIGURE 4.20 – Methode XORIntegralSauvola

Integral XOR Otsu

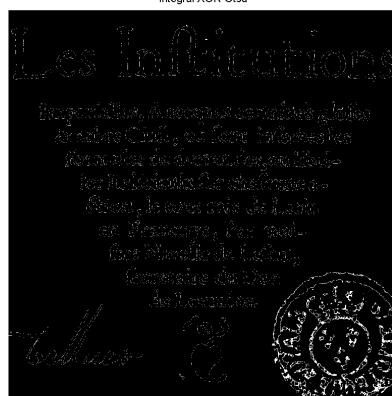


FIGURE 4.21 – Methode XOROtsu

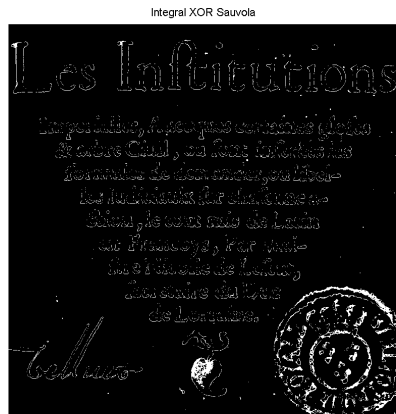


FIGURE 4.22 – Methode XORSauvola

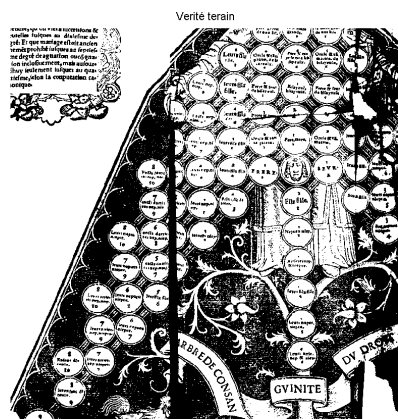


FIGURE 4.23 – verite terrain

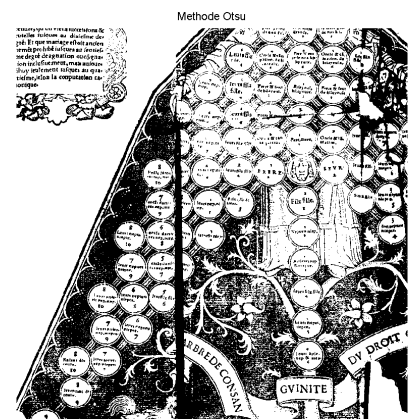


FIGURE 4.24 – Methode otsu

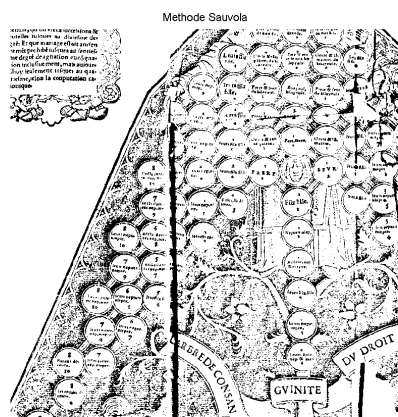


FIGURE 4.25 – Methode sauvola

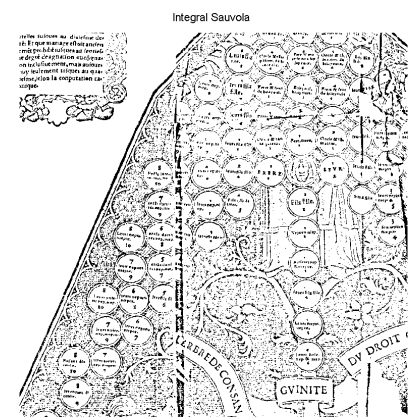


FIGURE 4.26 – Methode Integral sau-
vola

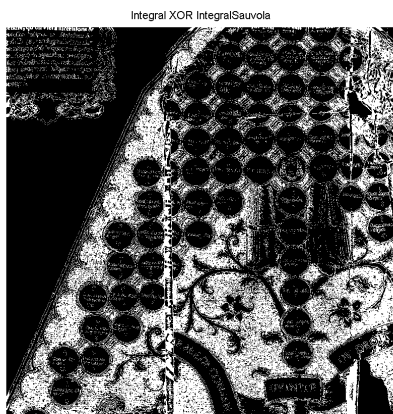


FIGURE 4.27 – Methode XORIntegralSauvola



FIGURE 4.28 – Methode XOROtsu

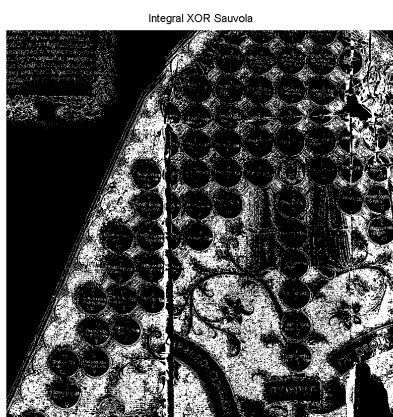


FIGURE 4.29 – Methode XORSauvola

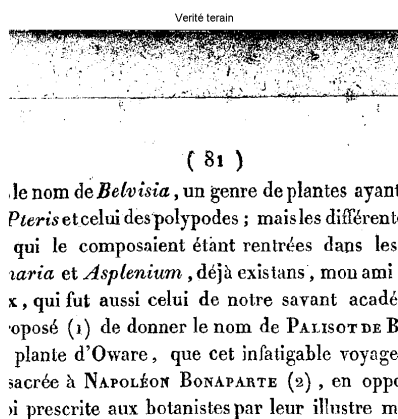


FIGURE 4.30 – verite terrain

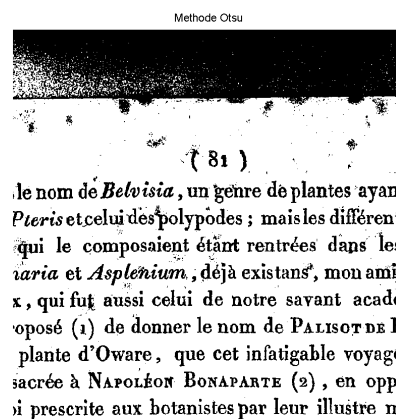


FIGURE 4.31 – Methode otsu

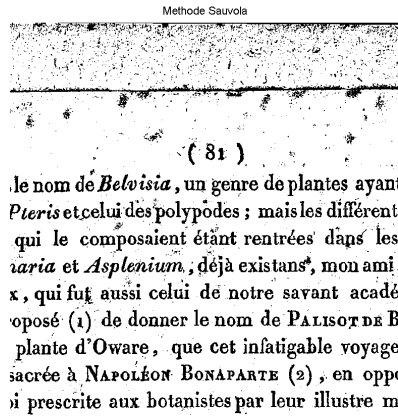


FIGURE 4.32 – Methode sauvola

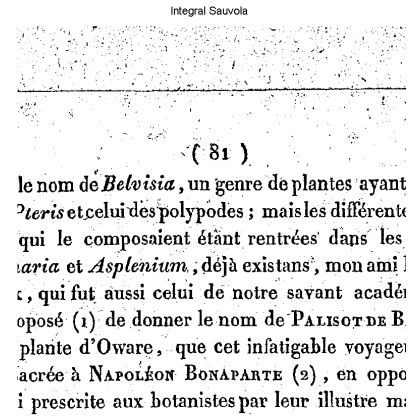


FIGURE 4.33 – Methode Integral sauvola

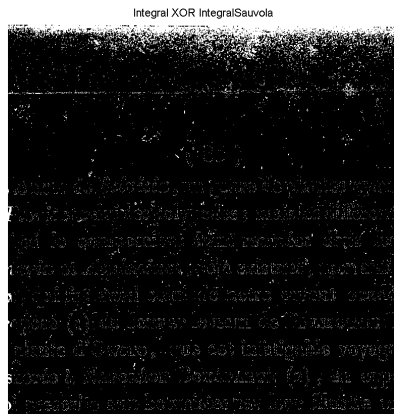


FIGURE 4.34 – Methode XORIntegralSauvola

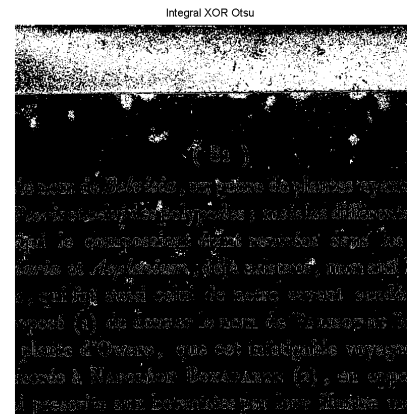


FIGURE 4.35 – Methode XOROtsu

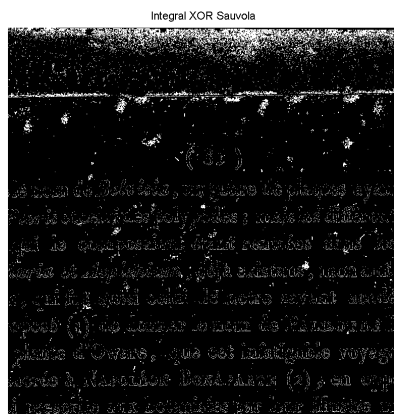


FIGURE 4.36 – Methode XORSauvola

Annexes B

```
%% UFR des sciences et techniques site du Madrillet , Universite de Rouen
%% :::::::::::::::::::::::::: Ter ACHRAF BEN SALAH ::::::::::::::::::::::::::
%% :::::::::::::::::::::::::: Seuillage d'image ::::::::::::::::::::::::::% %
% Nettoyer tous les racourcis.

clear all;

% lire image

monImage= imread('simulation3/Zola4.png');

% convertir image en niveau de gris si on travaille
% avec les images en couleurs

%monImage = rgb2gray(imread('simulation3/PeppersRGB.png'));

% Appliquer l'algorithme d'Otsu

imageAlgoOtsu = OtsuAlgo(monImage);

% Appliquer l'algorithme Sauvola

imageSauvola = SauvolaAlgo(monImage);

% Appliquer l'algorithme integrale de Sauvola

imageIntegraleSauvola = integraleSauvola(monImage);

%affichage des résultats

ImageOtsu = figure(1);
title('Methode Otsu');
imshow(imageAlgoOtsu);
saveas(ImageOtsu,'simulation3/Otsu1.jpg');
ImageS = figure(2);
title('Methode Sauvola');
imshow(uint8(imageSauvola));
saveas(ImageS,'simulation3/Sauvola1.jpg');
ImageInS = figure(3);
title('Integral Sauvola');
imshow(uint8(imageIntegraleSauvola));
saveas(ImageInS,'simulation3/IntegralSauvola1.jpg');
```

Programme principale

```
=====
function image_Otsu = OtsuAlgo(mon_image)
```

```

tic;
% Creation de l'histogramme

histo = hist(mon_image(:),0:255);
reHisto = histo(:);

%Creation niveau de gris

NGNiveau = 0:255;
NG =NGNiveau(:);

% Cette vecteur va contenir deux valeurs a chaque fois :
% l'un c'est la valeur interClasse a chaque operation et
% l'autre sa correspondance au niveau de gris.

InterClasse =zeros(size([1 256]));

for i=1:255

    [poids_p_plan,variance_p_plan]=calculSeuil(1,i);
    [poids_s_plan,variance_s_plan]=calculSeuil(i+1,255);
    InterClasse(i)=(poids_p_plan*variance_p_plan)+(poids_s_plan*variance_s_plan);

end

[Interclasse, seuilVal] = min(InterClasse);
seuilValIm2bw=(seuilVal)/256;
image_Otsu =im2bw(mon_image,seuilValIm2bw);

function [poids,variance] = calculSeuil(posI, posF)

% calcul poids

poids = sum(reHisto(posI:posF))/sum(reHisto);

% calcul moyenne

sommeNGH = sum(reHisto(posI:posF).*NG(posI:posF));
sommeNiveauGris = sum(reHisto(posI:posF));
moy = sommeNGH / sommeNiveauGris;

% calcul Variance

valeur=(NG(posI:posF)-moy).^2;

numero=sum(valeur.*reHisto(posI:posF));

variance=numero/sum(reHisto(posI:posF));

end
time = toc;
disp(time);
end

```

Methode d'Otsu

=====

```

function image_bin_sauvola = SauvolaAlgo(monimage)
tic
% prendre les dimensions de l'image

[l,c] = size(monimage);
monimage = double(monimage);

% Creation de l'image resultat.

image_bin_sauvola = zeros(l,c);

% Creation les attributs de la fenetre

longeur_fenetre = 14;
largeur_fenetre = 14;

for i=1 : l-longeur_fenetre ,
    for j=1 : c-largeur_fenetre ,

% Initialitation les dimensions de la fenetre

        ni = i+longeur_fenetre;
        nj = j+largeur_fenetre;

% Decoupage de l'image

        image_decoupage = imcrop(monimage,[j i longeur_fenetre largeur_fenetre]);

% Calcul la moyenne de la fenetre

        moy = mean(image_decoupage(:));

% Calcul l'ecartType de la fenetre

        ecart_type = std(double(image_decoupage(:)));

% Calcul la seuil de la fenetre

        seuil = seuilSauvola(moy,ecart_type);

% Seuillage l'image

        for posI = i : ni;
            for posJ = j : nj
                if (monimage(posI,posJ) > seuil)
                    image_bin_sauvola(posI,posJ)=255;
                else
                    image_bin_sauvola(posI,posJ)=0;
                end
            end
        end
    end
end
time = toc;
disp(time);
end

```

Methode de sauvola

```
=====
function imagefinal = integraleSauvola(monimage)

% Dimension d'image

[l,c] = size(monimage);

% Calculer l'image integral

image_integral = integralImage(monimage);

% Convertir les valeurs des pixels uint8 vers double

monimage=double(monimage);

% Construire la matrice des valeurs moyennes.

image_moyenne = zeros(l,c);

% Construire la matrice de ecartType.

image_ecart_type = zeros(l,c);

% Initialiser l'image resultat

imagefinal=zeros(l,c);

% Initialiser les variables

compteurDebut = 8;
cFenetreSI = 7;
cFenetreSJ = 7;
largeur_Fenetre = 14;
longueur_fenetre = 14;

% Calcul le temps d'execution

tic
for i = compteurDebut : l-cFenetreSI,
    for j = compteurDebut : c-cFenetreSJ,
        % calculer la matrice des valeurs moyennes.

        image_moyenne(i,j) = (image_integral(i + cFenetreSI, j + cFenetreSJ)
+ image_integral(i - cFenetreSI, j - cFenetreSJ)- image_integral(i + cFenetreSI,
j - cFenetreSJ) \ \ - image_integral(i - cFenetreSI, j + cFenetreSJ)) /
(cFenetreSI+cFenetreSJ+1)^2;

        % calculer la matrice de ecarttype.

        res=sum(sum(monimage(i-cFenetreSI:i+cFenetreSI,j-cFenetreSJ:j+cFenetreSJ).^2
-image_moyenne(i,j).^2));
```

```

image_ecart_type(i,j) = (1/(cFenetreSI+cFenetreSJ+1)^2)*res;
    end
end

% binariser l'image

for i = 1 : l-longueur_fenetre ,
    for j = 1 : c-largeur_Fenetre ,
        ni = i+longueur_fenetre;
        nj = j+largeur_Fenetre;

        % appelle la fonction sauvola qui permet de calculer le seuil

        seuil = seuilSauvola(image_moyenne(i,j),sqrt(image_ecart_type(i,j)));
        for posI = i : ni;
            for posJ = j : nj
                if (monimage(posI,posJ) >= seuil)
                    imagefinal(posI,posJ)=255;
                else
                    imagefinal(posI,posJ)=0;
                end
            end
        end
    end
end
end
time= toc;
disp(time);
end

```

Methode de integral Sauvola

```

=====

close all;
clear all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% verite terrain %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lire l'image original

imageOriginalCouper = rgb2gray(imread('images/PeppersRGB.jpg'));

% lire l'image verite image

imageVeriteTerrain = imread('images/veriterTerrain0087.tif');

% Appelle des algorithmes de seuillage

imageAlgoOtsu = OtsuAlgo(imageOriginalCouper);

imageSauvola = SauvolaAlgo(imageOriginalCouper);

imageIntegraleSauvola = integraleSauvola(imageOriginalCouper);

% Appliquer la fonction XOR qui permet la comparaison entre l'image verite
% terrain et l'algorithme de seuillage.

```

```

imageXORotsu = xor(imageVeriteTerrain, imageAlgoOtsu);
[vecteurErreurX, vecteurErreurY] = find(imageXORotsu==1);
tauxDErreurOtsu = length(vecteurErreurX)/(size(imageOriginalCouper,1)
*size(imageOriginalCouper,2));

imageXORSauvola = xor(imageVeriteTerrain, imageSauvola/255);
[vecteurErreurSX, vecteurErreurSY] = find(imageXORSauvola==1);
tauxDErreurSauvola = length(vecteurErreurSX)/(size(imageOriginalCouper,1)
*size(imageOriginalCouper,2));

imageXORIntegralSauvola = xor(imageVeriteTerrain, imageIntegraleSauvola/255);
[vecteurErreurISX, vecteurErreurISY] = find(imageXORIntegralSauvola==1);
tauxDErreurIntegralSauvola = length(vecteurErreurISX)/(size(imageOriginalCouper,1)
*size(imageOriginalCouper,2));

% affichage des resultats

figure;
imshow(imageVeriteTerrain); title('Verite terrain');
print('-dtiffn', 'terrain/ev7/verite.tif');
figure;
imshow(imageAlgoOtsu); title('Methode Otsu');
print('-dtiffn', 'terrain/ev7/otsu.tif');
figure;
imshow(uint8(imageSauvola)); title('Methode Sauvola');
print('-dtiffn', 'terrain/ev7/sauvola.tif');
figure;
imshow(uint8(imageIntegraleSauvola)); title('Integral Sauvola');
print('-dtiffn', 'terrain/ev7/integralSauvola.tif');
figure;
imshow(imageXORotsu); title('Integral XOR Otsu');
print('-dtiffn', 'terrain/ev7/XORotsu.tif');
figure;
imshow(imageXORSauvola); title('Integral XOR Sauvola');
print('-dtiffn', 'terrain/ev7/XORSauvola.tif');
figure;
imshow(imageXORIntegralSauvola); title('Integral XOR IntegralSauvola');
print('-dtiffn', 'terrain/ev7/XORIntegralSauvola.tif');

```

Evaluation des résultats