École doctorale : InfoMaths Numéro d'ordre : 2012-ISAL0044

Thèse

Segmentation et classification dans les images de documents numérisés

présentée devant
L'Institut National des Sciences Appliquées de Lyon

pour obtenir
le grade de docteur

par

Asma OUJI

Soutenue le 01/06/2012 devant la Commission d'examen

Jury

Rapporteur	J.-M. OGIER	Professeur, Université de La Rochelle
Rapporteur	C. VIARD-GAUDIN	Professeur, IUT de Nantes
Directeur de thèse	A. BASKURT	Professeur, INSA de Lyon
Co-directeur de thèse	F. LeBOURGEOIS	Maître de conférence, INSA de Lyon
Examinateur	P. LAMBERT	Professeur, Polytech'Savoie Annecy
Membre invité	P.-F BESSON	Ingénieur, SPIGRAPH

Laboratoire :
Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS)

INSA Direction de la Recherche - Ecoles Doctorales – Quinquennal 2011-2015

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **http://www.edchimie-lyon.fr** Insa : R. GOURDON	M. Jean Marc LANCELIN Université de Lyon – Collège Doctoral Bât ESCPE 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72.43 13 95 directeur@edchimie-lyon.fr
E.E.A.	**ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE** **http://edeea.ec-lyon.fr** Secrétariat : M.C. HAVGOUDOUKIAN eea@ec-lyon.fr	M. Gérard SCORLETTI Ecole Centrale de Lyon 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60 97 Fax : 04 78 43 37 17 Gerard.scorletti@ec-lyon.fr
E2M2	**EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION** **http://e2m2.universite-lyon.fr** Insa : H. CHARLES	Mme Gudrun BORNETTE CNRS UMR 5023 LEHNA Université Claude Bernard Lyon 1 Bât Forel 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cédex Tél : 04.72.43.12.94 e2m2@biomserv.univ-lyon1.fr
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTE** **http://ww2.ibcp.fr/ediss** Sec : Safia AIT CHALAL Insa : M. LAGARDE	M. Didier REVEL Hôpital Louis Pradel Bâtiment Central 28 Avenue Doyen Lépine 69677 BRON Tél : 04.72.68 49 09 Fax :04 72 35 49 16 Didier.revel@creatis.uni-lyon1.fr
INFOMATHS	**INFORMATIQUE ET MATHEMATIQUES** http://infomaths.univ-lyon1.fr	M. Johannes KELLENDONK Université Claude Bernard Lyon 1 INFOMATHS Bâtiment Braconnier 43 bd du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél : 04.72. 44.82.94 Fax 04 72 43 16 87 infomaths@univ-lyon1.fr
Matériaux	**MATERIAUX DE LYON** Secrétariat : M. LABOUNE PM : 71.70 –Fax : 87.12 Bat. Saint Exupéry Ed.materiaux@insa-lyon.fr	M. Jean-Yves BUFFIERE INSA de Lyon MATEIS Bâtiment Saint Exupéry 7 avenue Jean Capelle 69621 VILLEURBANNE Cédex Tél : 04.72.43 83 18 Fax 04 72 43 85 28 Jean-yves.buffiere@insa-lyon.fr
MEGA	**MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE** Secrétariat : M. LABOUNE PM : 71.70 –Fax : 87.12 Bat. Saint Exupéry mega@insa-lyon.fr	M. Philippe BOISSE INSA de Lyon Laboratoire LAMCOS Bâtiment Jacquard 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél :04.72.43.71.70 Fax : 04 72 43 72 37 Philippe.boisse@insa-lyon.fr
ScSo	**ScSo*** M. OBADIA Lionel Sec : Viviane POLSINELLI Insa : J.Y. TOUSSAINT	M. OBADIA Lionel Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Tél : 04.78.69.72.76 Fax : 04.37.28.04.48 Lionel.Obadia@univ-lyon2.fr

*ScSo : Histoire, Geographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Remerciements

J'adresse mes sincères remerciements aux Professeurs Christian Viard-Gaudin et Jean-Marc Ogier pour avoir tenu le rôle de rapporteur de ma thèse avec tout l'investissement que représente sa lecture et son annotation.

Je remercie également le Professeur Patrick Lambert pour avoir participé à mon jury de thèse et M. Pierre-François Besson pour avoir été mon correspondant en entreprise.

Je tiens à exprimer ma reconnaissance à la société SPIGRAPH d'avoir été si généreuse et accueillante durant toute ma thèse. Merci à toute l'équipe d'Aix en Provence particulièrement.

Naturellement, j'exprime ma gratitude au Professeur Atilla Baskurt pour avoir dirigé ma thèse.

Je suis très reconnaissante envers Frank Le Bourgeois pour ses conseils techniques, son encadrement et son aide inestimable.

Je ne trouve pas les mots pour exprimer ma reconnaissance envers Yann Leydier pour toutes les connaissances que j'ai acquises grâce à lui et pour avoir suivi mes travaux durant toute leur progression.

Enfin, mes pensées vont à toutes les personnes qui m'ont apporté leur soutien. Merci à Nabil Bizid, Jérôme Revaud, Jean Duong, à ma mère et mon père ainsi que toute ma famille et mes amis.

À ma mère
pour son amour
et son soutien illimités

Résumé

Les travaux de cette thèse ont été effectués dans le cadre de l'analyse et du traitement d'images de documents imprimés afin d'automatiser la création de revues de presse.

Les images en sortie du scanner sont traitées sans aucune information a priori ou intervention humaine. Ainsi, pour les caractériser, nous présentons un système d'analyse de documents composites couleur qui réalise une segmentation en zones colorimétriquement homogènes et qui adapte les algorithmes d'extraction de textes aux caractéristiques locales de chaque zone.

Les informations colorimétriques et textuelles fournies par ce système alimentent une méthode de segmentation physique des pages de presse numérisée. Les blocs issus de cette décomposition font l'objet d'une classification permettant, entre autres, de détecter les zones publicitaires.

Dans la continuité et l'expansion des travaux de classification effectués dans la première partie, nous présentons un nouveau moteur de classification et de classement générique, rapide et facile à utiliser. Cette approche se distingue de la grande majorité des méthodes existantes qui reposent sur des connaissances *a priori* sur les données et dépendent de paramètres abstraits et difficiles à déterminer par l'utilisateur.

De la caractérisation colorimétrique au suivi des articles en passant par la détection des publicités, l'ensemble des approches présentées ont été combinées afin de mettre au point une application permettant la classification des documents de presse numérisée par le contenu.

Mots clés : images scannées bruitées, analyse colorimétrique, segmentation physique, classification, classement.

Abstract

In this thesis, we deal with printed document images processing and analysis to automate the press reviews.

The scanner output images are processed without any prior knowledge nor human intervention. Thus, to characterize them, we present a scalable analysis system for complex documents. This characterization is based on a hybrid color segmentation suited to noisy document images. The color analysis customizes text extraction algorithms to fit the local image properties.

The provided color and text information is used to perform layout segmentation in press images and to compute features on the resulting blocks. These elements are classified to detect advertisements.

In the second part of this thesis, we deal with a more general purpose : clusternig and classification. We present a new clustering approach, named ACPP, which is completely automated, fast and easy to use. This approach's main features are its independence of prior knowledge about the data and theoretical parameters that should be determined by the user.

Color analysis, layout segmentation and the ACPP classification method are combined to create a complete processing chain for press images.

Key words : noisy digitized document images, color analysis, layout segmentation, classification, clustering.

Table des matières

Introduction

Table des matières

1 Contexte de nos travaux

1.1 Convention CIFRE

Le présent mémoire rend compte des travaux de recherche effectués de concert au sein de la société Spigraph et du LIRIS autour de la thématique d'analyse et traitement des images de document numérisé.

Cette thèse présente un aspect applicatif aussi bien que théorique dans la mesure où nous nous plaçons dans un contexte industriel, vis-à-vis de l'entreprise, d'une part et un cadre de recherches scientifiques d'autre part.

Nous veillons donc à étudier et expliquer le fondement théorique de chaque fonctionnalité présentée ainsi qu'à obtenir des résultats exploitables dans un contexte industriel (des performances élevées et un temps d'exécution acceptable).

1.2 Projet MediaBox

Certaines entreprises ou particuliers ont besoin d'un accès facile et rapide aux articles de presse les intéressant. Or, devant le volume des publications partout dans le monde, il est impossible de recueillir manuellement, chaque jour, tous les articles qui concernent une société donnée pour constituer une revue de presse. L'automatisation de cette tâche s'avère donc particulièrement rentable.

C'est dans cette optique qu'est né MediaBox : un projet OSEO réunissant plusieurs acteurs industriels spécialisés en numérisation, traitement d'images (dont Spigraph) ou en Web.

Ce projet vise à automatiser la création des revues de presse, de la numérisation (capture) des journaux et magazines jusqu'à la mise en ligne des articles. Il comporte des volets en traitement de document numérique (Web, Pdf, Xml, *etc.*), comme par exemple la captation du Web, et des volets en traitement d'images de documents numérisés (scannés).

Le cadre du sous-projet assigné à Spigraph comporte les fonctionnalités mentionnées ci-dessous.

1.2.1 Détection de défauts de numérisation

La prise de vue d'importants volumes de documents est une tâche fastidieuse et répétitive. Les opérateurs sont nécessairement amenés à commettre des erreurs et les machines peuvent se dérégler en cours de l'utilisation.

La vérification manuelle n'étant pas suffisante, la détection automatique, par traitement d'images, des défauts de numérisation, avant OCR, est le seul moyen de réagir au plus vite lorsque la numérisation est défaillante.

1.2.2 Automatisation de la chaîne de traitement en lots

Les logiciels de reconnaissance optique de caractères (OCR) donnent des résultats très variables en fonction du type de document (roman, prospectus, *etc.*), d'une part, et de la qualité de la prise de vue d'autre part. Selon son type, un document peut être de fond texturé ou uniforme, contenir des zones graphiques riches en couleurs ou ne contenir que de texte, *etc.* Par ailleurs, la qualité de l'image varie selon les paramètres de réglage du scanner qui peuvent respecter les couleurs d'origine (présentes dans le support papier) et introduire du bruit à des degrés variables.

Une caractérisation colorimétrique automatique de chaque image permettrait de guider le choix des traitement optimaux qui sont adaptés à son type et ses propriétés locales. La caractérisation permettrait également de filtrer le bruit de numérisation dès sa détection. Ces traitements préalables, comme par exemple une binarisation locale et adaptative,

permettraient donc à l'OCR d'atteindre de hautes performances.

1.2.3 Détection automatique des publicités

La mise en ligne des articles de presse ne peut prendre forme sans une phase de filtrage éliminant les éléments indésirables au préalable. Ces éléments peuvent être sous forme de zones publicitaires infiltrés à l'intérieur d'articles réguliers. Dans d'autres cas de figure, il est nécessaire de vérifier la présence de ces éléments pour des objectifs de marketing.

À l'heure actuelle, ces filtrages / vérifications se font manuellement, ce qui peut s'avérer parfois coûteux. Nous tâcherons donc, dans le cadre de ce projet, d'automatiser l'ensemble de ces tâches.

1.2.4 Suivi des articles

Le suivi des articles consiste en la recomposition automatique des blocs de texte dans l'ordre naturel de lecture. Il est, bien entendu, nécessaire d'effectuer ce formatage avant la mise en ligne des articles ; autrement, le contenu sémantique manquerait de cohérence.

Cette phase devrait intervenir suite à une étape de segmentation physique décomposant chaque page en un ensemble de blocs homogènes ainsi que la phase d'OCR procurant l'information sémantique.

2 Caractérisation des images en sortie du scanner

Une multitude de traitements peut être appliquée à une image en sortie d'un scanner. La nature de ces traitements devrait s'adapter au le contenu de l'image (texte, figures, photos, etc.), le fait qu'elle soit en couleur, en niveaux de gris ou en noir et blanc, *etc.*. En effet, certains traitements ne sont applicables que sous certaines conditions ; par exemple, nous ne pouvons pas analyser les valeurs de teinte dans une image en noir et blanc puisque cette mesure n'est pas définie pour ces pixels.

Réaliser la typologie ou la caractérisation des images permettrait donc de réserver le traitement idoine à chaque image (ou partie d'image). Ces outils de caractérisation devraient, par exemple, décider si une image est une page de document contenant du texte ou une photo, si une image est en couleur ou en niveaux de gris, *etc.* Pour récapituler, ces outils visent à déterminer, automatiser et améliorer la suite de traitements susceptibles d'être lancés sur une image de document numérisé.

3 Mise en adéquation de la chaîne de traitement vis à vis des images

Les chaînes de traitement classiques se séparent en deux catégories :

- les traitements de masse opérant la même suite d'opérations quel que soit la nature de l'image en entrée, comme par exemple les logiciels d'OCR qui opèrent systématiquement une binarisation avant de segmenter en blocs même si le texte est immergé dans une photo,
- et les chaînes où l'utilisateur intervient pour caractériser l'image manuellement et diriger ainsi la suite de traitements subséquents.

Dans le premier cas, une perte d'information, et donc de performances est inéluctable. En effet, comme nous l'avons mentionné dans la section précédente, il est important de réserver à chaque image le traitement approprié. La transgression de cette règle, comme par exemple la binarisation d'une photo qui contient du texte, implique certainement des conséquences indésirables.

L'intervention de l'utilisateur est certes efficace en termes de performances mais cela implique un temps de réponse inacceptable dans le cadre des lots de traitements massifs sur des données conséquentes.

Ainsi, la caractérisation d'images peut différencier les documents qui peuvent être traités automatiquement de ceux qui nécessitent une supervision par un opérateur. La juste répartition des documents entre la chaîne automatique et la chaîne de traitement supervisée, garantit une performance maximale du système pour un temps d'exécution minimal.

4 Traitement des images en couleurs

Les travaux sur les images de documents en couleurs sont extrêmement rares. En effet, la communauté de chercheurs en images de document a depuis toujours traité des images binaires ou, plus rarement, en niveaux de gris. Ainsi, la plupart des applications existantes (segmentation physique, redressement, OCR, *etc.*) ne s'appliquent généralement qu'à des images bitonales. Quelques travaux récents ont permis d'adapter certains de ces traitements aux images couleurs ou en niveaux de gris. Ces derniers ne couvrent cependant pas l'ensemble des besoins en traitement d'images de documents.

Comme nous l'avons mentionné précédemment, la binarisation (ainsi que le passage en niveaux de gris) implique une perte d'information pénalisante sur certains documents. La caractérisation colorimétrique, quant à elle, permet une binarisation locale et conditionnelle sans perte qui stimule ainsi les performances des traitements ultérieurs.

Il existe, dans la littérature très peu de travaux traitant de l'analyse colorimétrique dans les images de document. Il serait donc intéressant de mettre au point une méthode de classification adaptée à ce type d'images. Par ailleurs, cela enrichira le volet 'traitement d'images' du projet Mediabox de façon considérable.

5 Classification d'images et dans les images de document

La classification est un domaine de recherche très vaste qui englobe différents cas d'utilisation et qui répond à des besoins divers. Dans le cadre du projet Mediabox, par exemple, nous avons besoin de classifier le contenu colorimétrique, structurel et sémantique des pages ou articles de presse.

La caractérisation colorimétrique dont nous venons de discuter constitue une forme de classification concernant les images de documents, puisqu'il s'agit d'étiqueter les images ou zones d'images selon leur contenu colorimétrique. À travers cette caractérisation, nous faisons donc un pas dans le domaine de classification d'images et dans les images de documents.

De nombreuses méthodes de classification sont conçues pour traiter plusieurs types d'objets et peuvent être appliquées dans différents contextes (classification de visages, de mouvements, d'images naturelles, *etc.*). Cependant, ces approches requièrent souvent des connaissances *a priori* sur les données comme par exemple le nombre ou la forme des classes. Par ailleurs, elles ne sont pas toujours efficaces sur les données issues d'images de document. Nous avons donc besoin d'une nouvelle approche à la fois générique, efficace sur notre corpus et adaptée au contexte industriel avec lequel nous interagissons, autrement dit, rapide et facilement utilisable par un utilisateur non-spécialiste. Cela nous permettra d'intervenir dans le projet Mediabox à différents niveaux : classification de bloc de presse, détection de publicités, classification de polices, *etc.*

6 Organisation du mémoire

Dans la première partie du présent mémoire, nous nous intéresserons à l'aspect colorimétrique des pages de documents numérisés.

Dans le cadre de la caractérisation des images, nous proposerons un système d'analyse colorimétrique permettant de détecter les régions chromatiques dans une page donnée, de déterminer les zones quantifiables sans perte et les régions que l'on peut binariser sans réduire la qualité de l'image.

Cette analyse assurera une quantification et une binarisation adaptatives et conditionnelles qui permettent de surmonter les difficultés intrinsèques à la variabilité des images de documents.

Par ailleurs, au cours de chaque phase de la décomposition colorimétrique, le bruit sera détecté et filtré et certains défauts dus à la numérisation seront corrigés.

Cette caractérisation ouvre l'accès à différentes applications. La fonctionnalité la plus immédiate et la plus facilitée est la détection de texte. En effet, le nombre de couleurs utilisées pour imprimer une région donnée nous donne une idée *a priori* sur son contenu (texte, image, *etc.*).

Nous utiliserons l'information colorimétrique et textuelle acquises pour mettre au point un algorithme de segmentation en blocs que nous opérerons sur des images de presse. Ces entités seront ensuite classées selon leurs contenus respectifs (texte, graphique, *etc.*) en employant des descripteurs basés sur le même type d'information (colorimétrique et textuelle). Ce classement permettra, entre autres, de détecter les blocs publicitaires qui peuvent s'infiltrer à l'intérieur des articles de journaux et magazines.

En résumé, ces applications et d'autres encore sont envisageables grâce à la caractérisation par le système de décomposition colorimétrique. Elles seront présentées dans le chapitre II.

Dans la continuité et l'expansion des travaux de classification effectués dans la première partie et afin de combler les lacunes inhérentes à ces derniers, nous proposerons un nouveau moteur de classification et de classement générique, rapide et facile à utiliser. Cette approche se distingue ainsi de la grande majorité des méthodes de classification qui reposent sur des connaissances *a priori* sur les données et dépendent de paramètres abstraits difficiles à déterminer par l'utilisateur. Or, notre moteur de classification (chapitre III) est indépendant de tout paramètre nécessitant une connaissance profonde des données.

Nous évaluerons les performances de cette approche dans différents cadres d'utilisation et ce en mode non-supervisé puis supervisé (chapitre IV).

À partir de l'ensemble des travaux réalisés durant cette thèse, nous présenterons une application de synthèse qui consiste en une chaîne de traitements complète des images de presse numérisée (caractérisation, reconnaissance de polices, classement d'articles de journaux, *etc.*).

Chapitre I

Caractérisation colorimétrique

Table des matières

1 Introduction et motivations

1.1 Quel traitement pour quelle image ?

Les outils de caractérisation visent à déterminer, automatiser et améliorer la suite de traitements susceptibles d'être lancés sur l'image en sortie du scanner.

Focalisons-nous tout d'abord sur l'aspect colorimétrie. Nous avons à faire à de plus en plus d'images à couches colorimétriques superposées grâce à la PAO (Publication Assistée par Ordinateur). Néanmoins, les rares travaux de recherches traitant de telles images ciblent des applications bien spécifiques telles que la compression MRC (Mixed Raster Content). Or, nombre d'applications, comme l'OCR (Optical Character Recognition) et la segmentation structurelle, sont beaucoup moins efficaces sans analyse colorimétrique préalable sur certaines images. Une telle étude s'avère même indispensable pour certaines applications comme la catégorisation d'images de documents récents, et plus particulièrement la détection de publicités.

1.2 Dégradations dans les images de documents

Impression, numérisation, compression, les images scannées sont souvent issues d'une série de traitements plus ou moins standards qui altèrent les couleurs originelles du document et y introduisent du bruit indésirable. Nous proposons de remonter le temps afin de restituer au document ses couleurs d'origine : celles qui ont été voulues par son auteur. Ce type de restauration permet d'améliorer la netteté de l'image, d'éliminer le bruit et rendre plus efficaces des applications comme la détection de texte, la classification, *etc.*

L'impression est la première source de distorsion qui affecte les documents modernes. En effet, les imprimantes simulent souvent les couleurs d'origine en utilisant un tramage de 4 couleurs (cyan, magenta, jaune et noir). L'œil humain ne perçoit pas très bien cette altération ; pourtant le scanner y est sensible et la détecte souvent. En effet, selon la largeur du tramage et la résolution de numérisation, des artefacts très gênants peuvent apparaître.

Les documents numérisés sont davantage altérés par des distorsions introduites par les scanners qui ne perçoivent pas forcément les couleurs exactement telles qu'elles sont (cela peut être partiellement corrigé grâce à un calibrage rigoureux). Des dégradations supplémentaires apparaissent lorsque les numériseurs sont mal-paramétrés. On distingue trois types de bruits potentiellement présents dans les images scannées :

- Le bruit de saturation : il s'agit de pixels chromatiques (colorés) introduits au voisinage de traits noirs ou dans des zones originellement achromatique (en niveaux de gris). Ce type de bruit est plus particulièrement engendré par les caméras linéaires (voir Figure I.1).

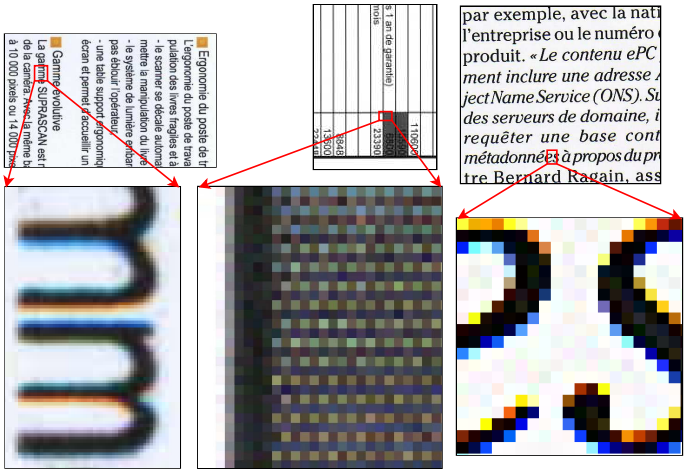


FIGURE I.1 – bruit de saturation

- Le bruit de teinte : il s'agit de pixels chromatiques dont la couleur a été altérée (voir Figure I.2). Ce genre de distorsion est généralement dû à une inadéquation entre la résolution de numérisation et le tramage.

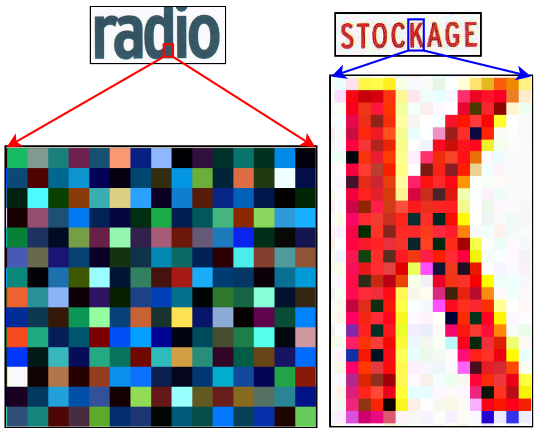


FIGURE I.2 – bruit de teinte

- Le bruit de luminance généralement causé par les basses résolutions. Ce bruit consiste en des traits originellement noirs qui paraissent gris dans l'image scan-

née. La binarisation de formes affectée par ce type de bruit risque de causer leur disparition.

Toutes les distorsions causées par les bruits décrits précédemment sont intensifiées par les différents formats de compression avec perte (JPEG, JPEG 2000, *etc.*)

Il s'avère parfois difficile de distinguer visuellement les couleurs originales du document des bruits introduits par la chaine de numérisation. La figure I.3 montre un exemple de ces distorsions. Dans cette dernière illustration, le théorème d'échantillonnage de Nyquist-Shannon n'est pas respecté vis à vis du tramage ; il est difficile de savoir si le 'L' est chromatique ou pas.

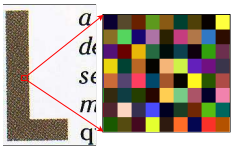


FIGURE I.3 – tramage multicolore

1.3 Solution proposée

Nous présenterons, dans ce chapitre, des outils performants permettant d'identifier la présence de régions colorées dans une image donnée et de localiser précisément ces zones. Cette étape permet, en l'occurrence, de distinguer les vraies couleurs des bruits et artéfacts présents dans l'image. Une segmentation colorimétrique sera ensuite appliquée aux zones où la quantification n'engendre aucune perte d'information (comme les zones de texte). Les zones multi-chromatiques (photos) sont identifiées et restent inchangées. De même, les zones achromatiques sont séparées en régions niveaux de gris et en zones en noir et blanc ; ces dernières peuvent être binarisées sans risque de perte d'information.

Le système de segmentation que nous proposons est complètement automatisé et ne nécessite aucune intervention humaine. Pour ce faire, nous introduisons une mesure préliminaire permettant d'estimer l'épaisseur moyenne des traits d'une image et, ainsi, d'automatiser l'ensemble du processus.

2 Vue d'ensemble

Dans cette section, nous donnerons une vue d'ensemble sur notre système de segmentation. Ce dernier est principalement composé de trois phases indépendantes et complémentaires, à savoir : la séparation chromatique / achromatique, la segmentation chromatique et la segmentation achromatique.

2.1 Séparation chromatique-achromatique

Ce procédé consiste à délimiter les zones de couleur dans une image quelconque. Un pixel est dit chromatique s'il a une teinte définie (rouge, vert, bleu, jaune, *etc.*) ; autrement, il est dit achromatique (niveaux de gris, y compris noir et blanc).

Les pixels chromatiques requièrent un traitement différent des pixels achromatiques. En effet, il serait insensé d'appliquer des procédés analysant la teinte à des pixels achromatiques puisque leur teinte est indéfinie ou imprévisible [60]. De ce fait, la séparation chromatique / achromatique est une phase indispensable à notre système.

À ce stade, le bruit de saturation est filtré.

2.2 Segmentation chromatique

Il s'agit de segmenter les zones monochromatiques tout en gardant les régions multi-chromatiques intactes :

- *une zone monochromatique* est composée d'un ensemble d'éléments de teintes discrètes (comme par exemple le texte en couleurs). La quantification d'une telle zone n'implique aucune perte significative d'informations.
- *une zone multi-chromatique* est généralement composée d'un large spectre de couleurs. L'exemple le plus représentatif de ces zones serait les photos naturelles. Leur quantification engendrerait une perte significative d'informations.

Le bruit de teinte est filtré au cours de la séparation chromatique.

2.3 Segmentation achromatique

La segmentation achromatique vise à distinguer les zones en noir et blanc des zones grises.

- Une zone en noir et blanc est binarisable sans perte d'information. Loin de l'altérer, la binarisation d'une zone bitonale permet d'améliorer son contraste.
- Une zone en gris correspond souvent à un graphique qui ne doit jamais être binarisé pour ne pas perdre de façon inéluctable les informations importantes..

Le bruit de luminance est éliminé à cette étape.

2.4 Bilan

Le schéma de la figure I.4 illustre les différentes classes colorimétriques issus de ce système de segmentation.

Le système de segmentation proposé ne requiert aucune information *a priori* ni de modèle. Par ailleurs, chacune des phases du système est réutilisable indépendamment des

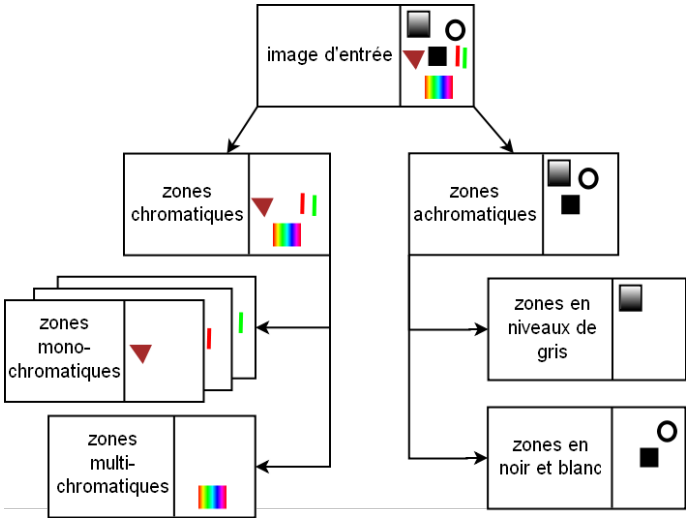


FIGURE I.4 – couches issues de la segmentation colorimétrique

autres. Si, par exemple, on sait d'avance qu'une image ne contient pas de couleur, la phase segmentation achromatique suffit.

Les différentes composantes de notre système de segmentation colorimétrique seront développées au cours des prochaines sections.

3 État de l'art

Cette section s'attache à donner un aperçu des principales méthodes de segmentation couleur dont le but se rapproche le plus possible du nôtre. Un grand nombre ces méthodes ont été développées dans le cadre de l'analyse d'images naturelles. Les objectifs et spécificités étant très différents, ces méthodes ne sont pas directement applicables à notre problématique et ne seront donc pas exhaustivement décrites. Qui plus est, les images naturelles comptent un nombre réduit de pixels par rapport aux images de documents qui en contiennent plusieurs millions pour une résolution moyenne de 300 ou 400dpi. Donc beaucoup de méthodes de l'état de l'art seront trop lentes pour que l'on puisse les adopter.

3.1 Séparation chromatique / achromatique

La séparation chromatique / achromatique est une phase cruciale dans le traitement des images de document couleur. Néanmoins, la littérature contient très peu d'articles traitant de ce problème.

Une façon intuitive pour séparer les zones en couleur des régions en niveaux de gris a été proposée et appliquée aux images naturelles [62]. Elle consiste à seuiller le canal Saturation, dans le système de représentation TSV (Teinte Saturation Valeur), avec un

seuil fixe. De même, Yang *et al.* [122] procèdent par simple seuillage de la saturation dans des images de couverts de magazines de presse. Loin de le filtrer, de telles méthodes attribuent le bruit de saturation à la catégorie des zones chromatiques. Or, nous avons besoin d'une méthode plus robuste au bruit pour traiter notre corpus.

Le système de représentation TSV a également été seuillé en s'appuyant sur un modèle de perception humaine [60], et ce sur des images issues du Web. Nous tâchons de restituer au document ses couleurs d'origine (avant impression et numérisation) ; or comme le montrent les figures I.1 et I.2, l'œil humain ne perçoit pas les couleurs de la même façon selon la distance à laquelle l'observateur se place par rapport à l'image. Le recours au modèle de perception humaine ne peut donc pas nous être utile. Par ailleurs, les images du Web ne présentent pas les mêmes distorsions que les images de documents. En effet, elles sont généralement exemptes des bruits introduits par la chaine d'impression et numérisation. Ainsi, la détection des zones couleur est plus simple sur de telles images.

3.2 Segmentation chromatique

Une fois les couches chromatique et achromatique séparées, chacune sera segmentée à son tour. La segmentation de la couche chromatique est similaire à une quantification sans pour autant en être une, puisque seules les zones monochromatiques sont concernées. C'est pour cette raison que nous passerons en revues certaines méthodes de quantification dans cette section.

La littérature comprend une gamme large et variée de méthodes de quantification [11, 103] conçues pour les images naturelles mais inadaptées aux images de document. Les méthodes de segmentation d'images de documents se répartissent en deux grandes familles : les méthodes globales qui appliquent un traitement identique à tous les pixels de la page et les méthodes hybrides qui prennent en considération le contexte local.

3.2.1 Méthodes globales

Classification des couleurs Il est possible d'appliquer un algorithme de classification non supervisé tel que k-means [1] [24] sur une image complète mais le résultat dépendra fortement des germes choisis et du nombre de classes (choisi *a priori*). Par ailleurs, l'application de cet algorithme sur plusieurs millions de pixels nécessite un temps de traitement considérable.

Une version élaborée et appropriée aux images de documents a été proposée [68, 67] mais nécessite toujours des informations *a priori* sur le document dont nous ne disposons

1. L'algorithme fut nommé nuées dynamiques par son auteur mais il est connu de la plupart sous son appellation anglophone.

pas.

Une autre version de k-means dans l'algorithme [114] propose de créer des classes dynamiquement. À chaque fois qu'une classe se voit affecter trop de points, une nouvelle est créée. L'algorithme commence avec une classe unique (qui a pour centre la moyenne des pixels de l'image). Afin de bien répartir les pixels dans les classes, ces derniers ne sont pas injectés séquentiellement mais dans un ordre aléatoire. Cette méthode nécessite tout de même de fixer un certain nombre de paramètres (nombre maximal de classes, nombre maximal d'itérations, nombre maximal de pixels affectés à une classe, *etc.*). Cet algorithme part de plus de l'hypothèse que la répartition des pixels dans les classes est équiprobable, ce qui n'est pas le cas dans les images de documents.

D'autres méthodes de quantification déterminent automatiquement le nombre de classes dans des images naturelles [45, 120, 91, 5] : les couleurs visuellement similaires sont regroupées ensemble. Les couleurs peu présentes dans l'image sont, toutefois, mal gérées par ces méthodes. Ces faux classements engendrent une sur-segmentation.

Pujol *et al.* [94, 95] présentent une méthode de quantification qui optimise le nombre de classes. Cette approche permet de fusionner les classes de couleurs similaires tout en conservant les couleurs faiblement représentées et susceptibles d'apporter de l'information (*e.g.* : un petit oiseau dans le ciel, un bouton d'arrêt d'urgence sur un panneau métallique...). Les algorithmes de réduction de couleurs conviennent aux images naturelles. En revanche, leur application aux images de documents bruitées engendre des classes superflues qui risquent de correspondre aux bruits de numérisation.

D'autres méthodes de réduction de couleurs ont été proposées dans le but de faciliter l'extraction de texte. Ces approches sont parfois basées sur un réseau de neurones avec rétroaction en mode non-supervisé [108] ou sur les graphes théoriques [96].

Smigiel *et al.* [106] proposent une méthode de segmentation en quatre classes (fond, texte, texte coloré et texte du verso) par classement via des cartes de Kohonen. Le réseau neuronal est entraîné sur une portion représentative d'une page et est ensuite utilisé pour classifier les pixels de chaque page du livre. La méthode est adaptée aux documents anciens et permet de traiter le cas de la visibilité du verso en transparence. En revanche, une telle approche n'est pas adaptée aux images de documents récents de notre corpus (comme par exemple les magazines de mode). Par ailleurs, la classification supervisée n'est pas robuste aux variations de luminosité et de teinte sur une même page, et encore moins d'une page à une autre.

Sur les images de documents anciens et dégradés, on recourt souvent à une restauration qui s'apparente à une quantification [25]. Il va de soi que notre corpus ne présente pas les mêmes dégradations que les manuscrits anciens et qu'une telle restauration engendrerait des pertes significatives.

Analyse d'histogrammes Une analyse d'histogrammes appliquée à des couleurs (après quantification) permet de segmenter automatiquement une image sans connaissance *a priori* sur le document [93]. Cette technique est particulièrement adaptée aux documents contemporains très colorés mais l'analyse étant globale, elle n'a ni la finesse nécessaire au traitement de nos documents à structure complexe, ni la possibilité de réserver un traitement particulier à chaque type de zones (monochromatique et multi-chromatique).

[122, 84] décrivent une réduction colorimétrique adaptée aux images de document et basée sur Mean-shift [34] et un filtre de lissage préservant les contours. Cette dernière approche est efficace sur des images de texte mais engendre une certaine perte sur les images contenant des éléments graphiques colorés.

De nombreuses autres méthodes de segmentation se basent sur les histogrammes [112, 126, 43]. Partant du principe qu'une classe est associée à un mode dans l'histogramme des couleurs, il est possible d'extraire les modes des trois histogrammes rouge, vert et bleu. La composition des résultats permet de déterminer les prototypes des éventuelles classes. La recherche de modes peut aussi être menée directement sur l'histogramme des couleurs en trois dimensions.

3.2.2 Méthodes hybrides

Les méthodes d'agrégation de régions permettent de se passer d'une phase d'apprentissage. Ces dernières sont basées sur des calculs locaux (notamment lors du calcul des germes) mais l'agrégation des régions ne peut pas être considérée comme locale. Nous considérons donc ces méthodes comme hybrides.

En utilisant une vision pyramidale (approche perceptive) d'une image, il est possible d'agglomérer les pixels semblables, puis les régions semblables. Le choix judicieux des paramètres d'agrégation permet de séparer le texte de fonds colorés complexes [74]. Néanmoins, ceci ne permet pas de filtrer les bruits d'impression et numérisation.

Afin de mieux contrôler l'agglomération, il est possible de sélectionner les germes des régions de façon biaisée. Une méthode s'appuyant sur des composantes connexes extraites directement de l'image en couleur a été proposée [35]. Les composantes sont ensuite triées dans un ou plusieurs arbres qui sont classifiés par un algorithme de type k-means.

Cette méthode dépend d'un paramètre décrivant la distance maximale entre deux couleurs d'une même composante. Ce dernier paramètre est calculé globalement pour chaque image en analysant la distance entre chaque couple de pixels. Ainsi, sur une même image de nombreuses zones sont sur-segmentées alors que d'autres sont sous-segmentées.

Une analyse récursive des modes de l'histogramme utilisant un nouvel espace de représentation perceptuel a été proposée [105]. Rappelons que les modèles perceptuels sont

mieux adaptés aux images naturelles.

DjVu [10] est un logiciel de compression des images de documents. Cette méthode est basée sur une segmentation de l'image en deux couches : texte et fond. Les éléments chromatiques du fond sont quantifiés via une décomposition multi-résolutions basée sur une cascade de classificateurs. Ce logiciel est assez performant sur les images propres et de résolution suffisante. Il s'avère cependant inefficace dès que te texte est affecté par des distorsions chromatiques ou achromatiques ou que ce dernier ne présente pas un contraste suffisant par rapport au fond. Dans tels cas, les éléments textuels sont quantifiés de la même façon que le fond ce qui paralyse sa lisibilité.

3.3 Segmentation achromatique

Notre système sépare les zones *Noir & Blanc* (qui seront exclusivement binarisées sans risque de perte d'information) des régions en *Gris* qui resteront inaltérées. Notre problématique consiste à savoir s'il faut binariser, non pas "comment binariser". Or la plupart des travaux liés à notre champ d'intérêt recherchent la façon optimale de binariser une image en entier en gardant le maximum d'information. C'est pour cette raison que notre étude de l'existant sera sommaire dans cette section.

En effet, un grand nombre de méthodes de binarisation qui s'adaptent au contexte local de chaque pixel [111, 72, 56, 87, 36, 17] existe dans la littérature. Ces algorithmes sont également adaptés aux images de documents dégradées. Or, étant donné que nous optons pour une binarisation sélective, un algorithme de binarisation basique pourrait suffire.

Il est possible de prétraiter [78] l'image ou de la restaurer [26] avant de la binariser afin d'assurer de meilleurs résultats. Néanmoins la binarisation des zones graphiques reste toujours pénalisante.

Une alternative serait d'appliquer une décomposition structurelle en amont du processus. Une segmentation logique [55, 13, 20] permettrait d'identifier les zones texte, les graphiques, les tableaux, *etc.* Seules les plages de texte seraient binarisées. Une telle approche donnerait des résultats intéressants localement sur le texte. Cependant, obtenir la structure logique nécessite une segmentation préalable ...

L'application d'approches structurelles serait toutefois coûteuse en termes de temps d'exécution d'une part, et largement dépendante du type de document d'autre part. Par ailleurs, il existe potentiellement des zones graphiques bitonales et des titres noirs écrits en gros qui seraient privés de la binarisation puisqu'ils n'appartiennent pas à la bonne classe.

3.4 Conclusion

Le seuillage de la saturation semble être la façon la plus appropriée pour localiser les zones chromatiques. Notre approche sera donc basée sur une mesure inspirée de la saturation et enrichie d'un ensemble de filtres qui éliminent le bruit.

Les traitements globaux n'étant pas suffisamment précis, nous avons opté pour une approche hybride basée sur les histogrammes de teinte pour assurer la séparation chromatique. Cette méthode permettra également de balayer une partie considérable du bruit de teinte.

La séparation achromatique sera également assurée par le biais d'histogrammes de luminosité à une échelle à la fois locale et globale.

4 Préambule : estimation de l'échelle

Nous allons présenter dans cette section une mesure de l'épaisseur moyenne des traits dans une image ou un corpus donnés et d'automatiser ainsi l'ensemble des traitements suivants.

Étant un prétraitement en soi, l'approche proposée est particulièrement rapide et ne nécessite aucun calcul préalable lourd, là où des méthodes aux objectifs similaires nécessitent au moins une binarisation.

4.1 Motivations

Les algorithmes de traitement d'images de document dépendent souvent d'un paramètre étroitement lié à la résolution de l'image (ex : taille d'un élément structurant, taille d'une matrice de convolution, *etc.*).

Pour automatiser ces traitements, il semblerait intuitif de se renseigner sur la résolution de l'image. Toutefois, même si la résolution de numérisation est figée (certains proclament que 300ppp est une résolution universellement adéquate), chaque document possède ses propres caractéristiques typographiques qui ne sont pas forcément communes à toutes les images du lot. Les lettres modernes et les factures sont généralement écrites avec une police de taille 10 ou 12 points. En revanche, les publicités, les dépliants et les journaux présentent des typographies très variables.

Ainsi, pour une taille donnée, les traits d'un même caractère sont susceptibles de présenter des épaisseurs différentes en fonction de la police de caractères choisie (voir Figure I.5).

Pour que les algorithmes dépendants d'un paramètre typographique soient robustes, nous avons donc choisi d'écarter la résolution de numérisation et de baser nos calculs sur

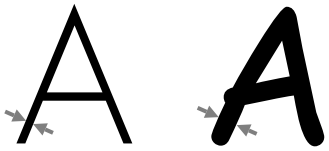


FIGURE I.5 – Deux caractères de même taille et d'épaisseurs différents.

l'épaisseur moyenne des traits du document.

4.2 Notre proposition

Nous avons opté pour une façon simple et efficace pour estimer l'épaisseur des traits : l'auto-corrélation le long de l'axe horizontal et respectivement vertical.

Les calculs seraient *a fortiori* plus précis si on déterminait l'orientation principale des traits au préalable mais cette évaluation prendrait un temps considérable pour un gain quasi-imperceptible.

4.2.1 Énoncé

Nous proposons une approche statistique et globale dont le but n'est pas de représenter toutes les spécificités d'une page de document donnée mais d'indiquer un ordre de grandeur de l'épaisseur moyenne des traits dans le texte.

Soit $T_h(I, \delta)$ la translation d'une image I en niveau de gris (définie dans un plan Ω) de δ pixels suivant l'axe horizontal. La suite $\left(\mathcal{D}^h(I)_n\right)_n$ est définie par :

$$\begin{aligned} \mathcal{D}^h(I)_0 &= 0 \\ \mathcal{D}^h(I)_n &= \textstyle\sum_{(x,y)\in\Omega} \|I(x,y) - T_h(I,n)(x,y)\| \end{aligned} \tag{I.1}$$

La suite $\left(\mathcal{D}^h(I)_n\right)_n$ est asymptotique.

L'estimation de la largeur moyenne des traits $\mathcal{S}_w$ repose sur l'évaluation itérative de $\left(\mathcal{D}^h(I)_n\right)_n$ jusqu'à ce que $n = n_m$, lorsque le covariogramme atteint un palier. On obtient ainsi $\mathcal{S}_w = n_m$.

L'estimation de la hauteur moyenne des traits $\mathcal{S}_h$ va de même : si $T_v(I, \delta)$ est la translation de I suivant l'axe vertical, la séquence $(\mathcal{D}^v(I)_n)_n$ est donnée par :

$$\begin{aligned} \mathcal{D}^v(I)_0 &= 0 \\ \mathcal{D}^v(I)_n &= \textstyle\sum_{(x,y)\in\Omega} \|I(x,y) - T_v(I,n)(x,y)\| \end{aligned} \tag{I.2}$$

Après itération, on obtient $\mathcal{S}_h$.

4.2.2 Évaluation

Notre approche a été testée sur une grande variété d'images de document (voir Figure I.6).

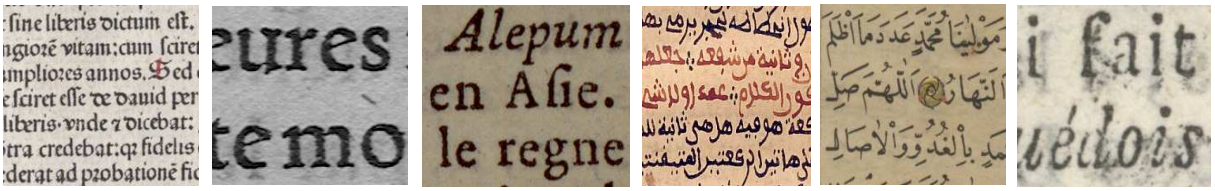


FIGURE I.6 – Exemples d'images utilisées pour tester l'estimation de l'épaisseur des traits.

Nous avons mesuré manuellement la largeur et la hauteur des traits : l'échelle de mesures varie de 2 à 10 pixels. L'erreur moyenne de $\mathcal{S}_w$ est de $1,25$ pixels. En raison de la proéminence des traits verticaux dans le texte, l'erreur moyenne de $\mathcal{S}_h$ est de $1,75$ pixels.

Ce résultat est très bon étant donné que nos images ne sont pas binarisées. En effet, étant donné que les traits sont toujours lissés dans une image en niveaux de gris, il est impossible de mesurer leurs épaisseurs avec précision. Par conséquent, une erreur inférieure à 2 pixels prouve que notre estimateur est suffisamment précis.

La présence d'éléments graphiques affecte naturellement les mesures. Les expérimentations témoignent, toutefois, que l'erreur occasionnée par ces éléments n'est jamais supérieure à 1 pixel. Dans une image contenant du texte de polices et tailles multiples, l'épaisseur moyenne correspond à celle de la classe la mieux représentée. En effet, le corps du texte (généralement écrit en plus petit) présente de nombreuses frontières horizontales (resp. verticales) qui influent sur l'auto-corrélation de façon significative tout en limitant l'impact du texte de grande taille, comme les titres.

Comme nous ne disposons pas d'information *a priori* sur l'orientation de la page, nous utiliserons, dans les sections suivantes l'estimateur
$\mathcal{S}_t = \max(\mathcal{S}_w, \mathcal{S}_h)$. $\mathcal{S}_t$ est désormais notre estimateur de l'épaisseur des traits dans une image (ou un corpus) donnée.

4.3 Conclusion

Nous avons présenté dans cette section une estimation fiable de l'échelle basée sur l'épaisseur des traits. La plupart des paramètres intervenant dans les algorithmes présentés dans les sections suivantes seront calculés automatiquement grâce à cette mesure.

5 Séparation chromatique / achromatique

Nous nous attachons dans cette section à identifier la présence des zones chromatiques dans une image donnée et de localiser précisément ces zones. Ce procédé permet, en l'occurrence, de distinguer les vraies couleurs du bruit de saturation et artéfacts introduit dans l'image.

La détection des régions chromatique repose, entre autres, sur le seuillage d'une nouvelle grandeur que nous nommons pseudo-saturation qui sera définie ci-après.

Le résultat de cette segmentation est un masque où chaque entrée indique si le pixel correspondant est chromatique ou achromatique. La création du masque final $\mathcal{M}_F$ passe par une étape intermédiaire : un masque approximatif noté $\mathcal{M}_A$. Ce dernier ne permet pas de retracer les frontières précises des zones chromatiques mais de nous renseigner sur leur présence et leur position.

Des tests à grande échelle ont été menés pour évaluer cette segmentation. Les résultats seront exposés et commentés.

5.1 Pseudo-saturation

5.1.1 Motivations

La séparation chromatique / achromatique est traditionnellement assurée par le seuillage de la saturation : les pixels chromatique affichent généralement des valeurs de saturation plus élevées que les pixels achromatiques.

Rappelons les différentes formules de saturation dans les trois systèmes de représentation existants.

Les deux représentations de points du modèle RVB dans un système de coordonnées cylindriques les plus communes sont HSL (Hue Saturation Lightness) et HSV (Hue Saturation Value).

HSI (Hue Saturation Intensity) est également un modèle colorimétrique utilisant des coordonnées cylindriques et généralement utilisé dans le domaine de vision par ordinateur.

La définition du modèle colorimétrique cylindrique TSV n'a jamais été standardisée et peut faire référence à l'un des trois modèles HSV, HSB ou HSI de façon équivoque.

Soit I une image définie dans l'espace RVB par :

$$\begin{aligned} I : \quad & \Omega \to \mathbb{N}^3 \\ & p \mapsto (R_p, V_p, B_p) \end{aligned} \tag{I.3}$$

Pour tout pixel p exprimé dans l'espace $[0,1]^3$, on définit dans $[0,1]$:

$$\begin{aligned} M(p) &= max(R_p, V_p, B_p) \\ m(p) &= min(R_p, V_p, B_p) \\ C(p) &= M(p) - m(p) \end{aligned} \tag{I.4}$$

La saturation dans les trois espaces HSL, HSV et HSI est respectivement définie par S_{HSL}, $S_{HSV} = S$, S_{HSI}.

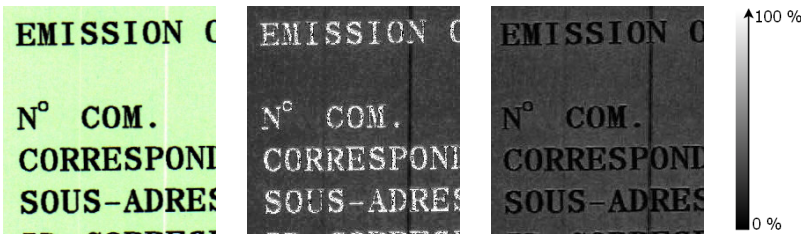
$$\begin{aligned} S_{HSL} : \quad & \Omega \to [0,1] \\ & p \mapsto \frac{C(p)}{1-|M(p)+m(p)-1|} \end{aligned}$$

$$\begin{aligned} S_{HSV} : \quad & \Omega \to [0,1] \\ & p \mapsto \frac{C(p)}{M(p)} \end{aligned} \tag{I.5}$$

$$\begin{aligned} S_{HSI} : \quad & \Omega \to [0,1] \\ & p \mapsto \frac{C(p)}{1-\frac{3\ m(p)}{(R_p+V_p+B_p)}} \end{aligned}$$

Comme l'attestent les définitions I.5 les trois formules classiques de saturation impliquent des divisions par des valeurs susceptibles d'être nulles.

En effet, lorsque la luminance est faible, la somme $(R+V+B)$ ainsi que M et m sont nulles ou proches de zéro. La saturation se comporte donc de façon chaotique pour les pixels sombres ou noirs. En illustration, la figure I.7.b montre que des pixels achromatiques (noirs) présentent des valeurs de S_{HSV} plus élevées que les pixels chromatiques (verts), ce qui est absurde.



a. b. c. d.

FIGURE I.7 – a. Image couleur, b. Saturation, c. Pseudo-saturation, d. Légende

5.1.2 Notre proposition

Les formules classiques de saturation n'étant pas satisfaisantes, nous en proposons une nouvelle que nous nommerons pseudo-saturation.

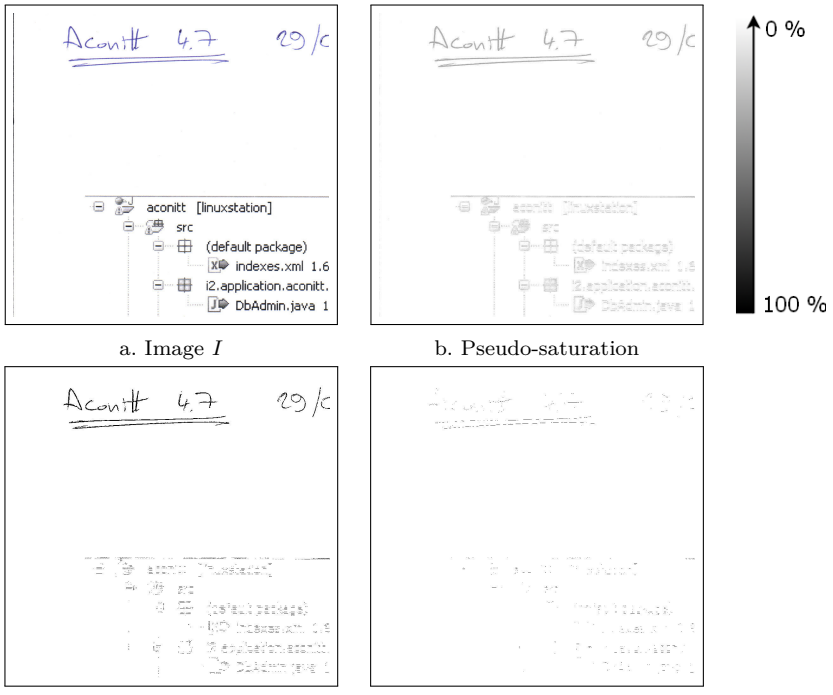
En tout point p d'une image I, la pseudo-saturation S_I^* est définie par :

$$\begin{aligned} S_I^* : \quad & \Omega \to [0, 255] \\ & p \mapsto \max(|R_p - G_p|, |R_p - B_p|, |G_p - B_p|) \end{aligned} \tag{I.6}$$

Comme le montre la figure I.7.c, la pseudo-saturation est valide aussi bien sur les pixels achromatiques que sur les pixels chromatiques.

Quelque soit la formule de saturation employée, il est impossible d'éliminer le bruit colorimétrique par un simple seuillage.

En effet, l'analyse de la figure I.8 montre qu'aucun seuil global ne permet de ressortir les zones chromatiques tout en filtrant le bruit. Le masque I.8.c laisse voir qu'un seuil faible de S^* permet d'extraire la ligne de texte bleu mais ne filtre pas le bruit de saturation se trouvant plus bas dans l'image. Un seuil plus élevé détériore la région chromatique sans pour autant éliminer le bruit (Figure I.8.d).



a. Image I b. Pseudo-saturation

c. S_I^* seuillée à 25% d. S_I^* seuillée à 40%

FIGURE I.8 – Seuillage de la pseudo-saturation (échelle inversée).

Le bruit de saturation sera éliminé durant une phase de pré-segmentation qui sera décrite dans la section suivante.

5.2 Pré-segmentation

Nous allons proposer ici une série de filtres permettant de créer $\mathcal{M}_A$, masque exempt du bruit de saturation et indiquant les positions des zones chromatiques de façon approximative. Une détection plus ponctuelle sera assurée par le masque final $\mathcal{M}_F$.

Cette segmentation est intrinsèquement utile dans la mesure où il est possible de se passer de $\mathcal{M}_F$ si, par exemple, l'utilisateur cherche à séparer les images contenant de la couleur des images n'en contenant pas.

Le calcul de $\mathcal{M}_A$ passe par deux étapes : un ré-échantillonnage Gaussien et une fermeture morphologique.

5.2.1 Ré-échantillonnage Gaussien

Un ré-échantillonnage permet de réduire la taille d'une image et subséquemment d'éliminer une partie du bruit qu'elle contient. Par ailleurs, le temps de traitement est d'autant plus faible que l'image est de taille réduite.

La première étape menant à $\mathcal{M}_A$ sera donc un ré-échantillonnage Gaussien de facteur d'échelle $\mathcal{S}_t$ (*cf.* section 4) effectué sur l'image d'origine.

Plus le facteur de réduction est grand, plus les artéfacts sont éliminés mais les traits fins ou les caractères colorés sont moins bien détectés. Le pas de ré-échantillonnage de réduction que nous avons choisi permet d'assurer un compromis entre l'élimination des artéfacts et la détection des traits colorés fins, puisqu'il est adapté à chaque document.

5.2.2 Fermeture morphologique

Les caméras linéaires introduisent souvent du bruit de saturation au voisinage des pixels de texte noir. Une fermeture morphologique [104] s'avère être le moyen le plus adéquat pour éliminer ce type de distorsions.

En effet, disposant d'un élément structurant de taille appropriée, la fermeture remplace le pixel bruités par des traits de texte régulier sans ronger les zones chromatiques. Il s'agit, bien entendu, d'une fermeture des points plus foncés que leur voisins.

Comme nous l'avons signalé précédemment, le bruit de saturation est connexe aux frontières des caractères [70]. Baird [6] affirme, en effet, que le bruit est une courbe exponentielle inversée qui décroît en partant des contours des traits.

La taille des images étant déjà normalisée par le ré-échantillonnage Gaussien, le choix d'un élément structurant commun à toutes nos images de document est trivial. Nous avons donc opté pour une taille minimale de l'élément structurant, à savoir une croix 3×3 (Figure I.9).

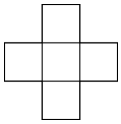


FIGURE I.9 – Élément structurant

L'image réduite et dilatée est appelée I^r (voir Figure I.11.b). Cette dernière sera seuillée pour créer le masque $\mathcal{M}_A$.

5.2.3 Seuillage

Nous calculons l'image de pseudo-saturation sur I^r. L'image résultante $S^*_{I^r}$ est ensuite seuillée pour en déduire le masque $\mathcal{M}_A$. Les éléments dont la pseudo-saturation est supérieure au seuil sont chromatiques, les zones restantes étant achromatiques.

L'estimation du seuil est basée sur l'analyse de l'histogramme de $S^*_{I^r}$, plus particulièrement son premier pic. La figure I.10 illustre l'algorithme d'estimation du seuil.

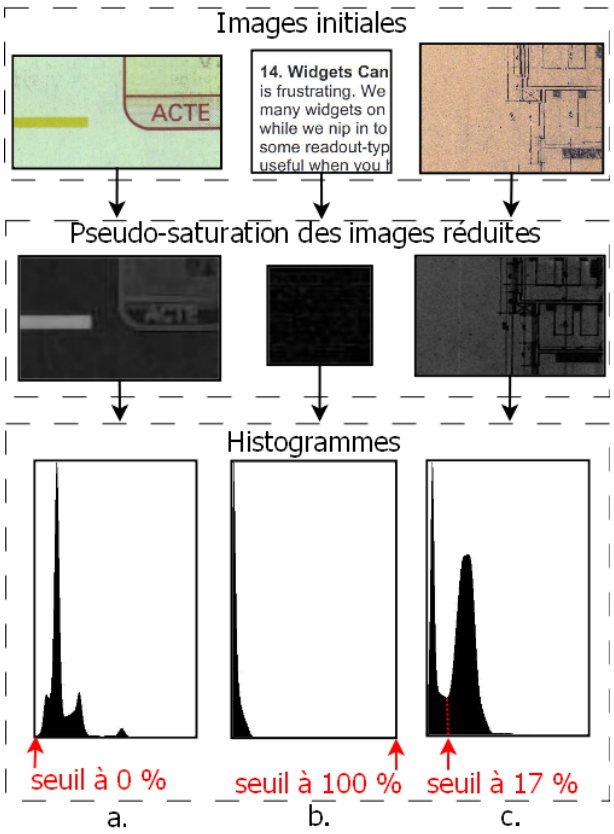


FIGURE I.10 – Seuillage d'une image : a. chromatique, b. achromatique c. mixte

Algorithme

- Les images entièrement chromatiques présentent des valeurs élevées de pseudo-saturation. Ainsi, si la position du premier mode de l'histogramme est sensiblement supérieur à zéro (tel que la figure I.10.a), l'image est jugée entièrement chromatique ; nous assignons donc au seuil de pseudo-saturation la valeur minimale (0%).
- À l'opposé du cas précédent, les pixels d'une image niveaux de gris sont faiblement saturés. Par conséquent, si les positions respectives de tous les modes de l'histogramme sont proches de zéro, l'image est considérée entièrement achromatique et le seuil est fixée à la valeur maximale (Figure I.10.b).
- Si les deux conditions précédentes ne sont pas remplies, le seuil de S^* est donné par la position du premier minima local qui suit le premier mode (voir Figure I.10.c).

Le masque $\mathcal{M}_A$ dans la figure I.11.c résultant du seuillage de l'image I.11.b montre que le bruit de saturation est radicalement filtré. La segmentation sera affinée dans la prochaine section.

a. I b. I^r c. $\mathcal{M}_A$

FIGURE I.11 – $\mathcal{M}_A$ sur une image contenant une ligne manuscrite bleue en haut de la page et du texte imprimé noir.

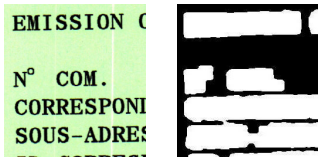
5.3 Détourage

Maintenant que le bruit de saturation est éliminé, nous pouvons nous pencher sur l'extraction ponctuelle des formes chromatiques, ce qui donnera lieu à la création du masque $\mathcal{M}_F$.

Un détourage efficace ne peut être effectué à une sous-échelle. Nous combinons ainsi $\mathcal{M}_A$ remis à l'échelle 1 : 1 avec un masque intermédiaire $\mathcal{M}_I$ qui résulte du seuillage de l'image de pseudo-saturation en pleine échelle. L'algorithme de seuillage employé est celui défini dans la section 5.2.3.

Il semblerait de prime abord que cette combinaison puisse naturellement être assurée par l'intersection logique entre les maques $\mathcal{M}_A$ et $\mathcal{M}_I$. Cependant, cette opération est inadaptée aux images à arrière-plan chromatique (*cf.* Figure I.12).

Pour remédier à ce dernier problème, nous proposons de construire le masque final

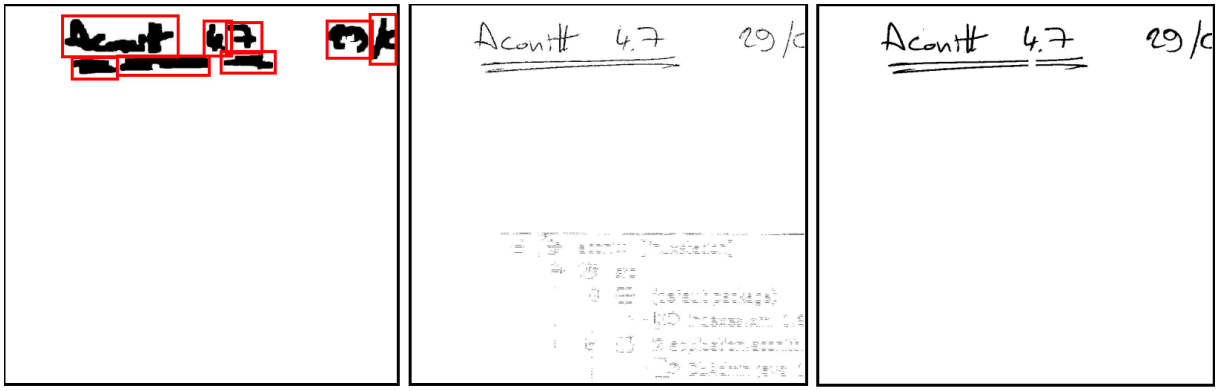


a. b.

FIGURE I.12 – a. Image à fond vert, b. Masque donné par l'intersection logique de $\mathcal{M}_A$ et $\mathcal{M}_I$

à partir de l'intersection logique entre $\mathcal{M}_I$ et les rectangles englobants [44] de $\mathcal{M}_A$ (en pleine échelle).

Les rectangles rouges dans la figure I.13.a correspondent aux rectangles englobants extraits sur $\mathcal{M}_A$. Le masque $\mathcal{M}_F$ déployé dans Figure I.13.c montre les formes chromatiques sont parfaitement détourées et que le bruit est radicalement filtré.



a. b. c.

FIGURE I.13 – Les masques intervenant dans la séparation chromatique / achromatique : a. $\mathcal{M}_A$, b. $\mathcal{M}_I$, c. $\mathcal{M}_F$

Une évaluation quantitative qui appuiera les résultats visuels sera fournie au cours de la prochaine section.

5.4 Évaluation

Dans le cadre d'évaluation quantitative de notre approche, nous avons mené des expérimentations sur une base composée d'une grande variété de documents numérisés (magazines, journaux, plans, manuscrits, *etc.*).

5.4.1 Protocole expérimental

Comme nous ne disposons pas des sources électroniques des images (même si cela avait été le cas, la phase mise en vis à vis aurait été problématique !), nous avons étiqueté 320

images manuellement en encadrant chaque connexité chromatique. De la même manière que les fragments d'images dans les figures I.1 et I.14), 30% des images de notre base de test sont extrêmement bruitées.

Les résultats seront exprimés en terme de précision P et rappel R. La précision est définie par la surface d'intersection entre les zones chromatiques de la vérité terrain et celles que notre approche a détectées divisée par la somme des surfaces chromatiques correctement détectées. Le rappel est donné par le rapport entre cette dernière surface d'intersection et la somme des surfaces chromatique dans la vérité terrain.

$$R = \frac{surface \bigcap}{surface\ de\ la\ vérité\ terrain} \quad P = \frac{surface \bigcap}{surface\ détectée} \tag{I.7}$$

5.4.2 Résultats et commentaires

Les résultats produits par notre méthode sont exposés et comparés avec une méthode plus basique dans le tableau I.1. Cette dernière consiste à seuiller l'image de saturation à une valeur fixe [62], à savoir 20% de du maximum possible.

Méthode	R	P
Kim 2009 [62]	93.26	70.03
Ouji 2011 [90]	91.54	99.88

TABLE I.1 – Résultats P/R

Comme le montre le tableau I.1 notre méthode atteint une précision quasi-parfaite. En effet, le filtrage du bruit de saturation élimine les fausses détections et augmente la précision de l'algorithme. D'ailleurs, notre système parvient à éliminer le bruit présenté dans Figure I.1. Un bruit d'aspect différent a été éliminé dans la figure I.14.

Les deux méthodes présentées affichent de bonnes valeurs de rappel signifiant que presque toutes les zones chromatiques ont été correctement extraites.

La figure I.15.d montre que le détourage des formes est extrêmement précis.

Les zones qui paraissent grises dans le masque présenté dans la figure I.15.b correspondent un tramage fin de couleur rose pâle (voir Figure I.15.c) qui a été correctement extraites.

Certaines images sont si mal numérisées que le texte y est composé, exclusivement de bruit. De telles images sont heureusement très rares et peuvent pénaliser la segmentation chromatique / achromatique (voir Figure I.16).

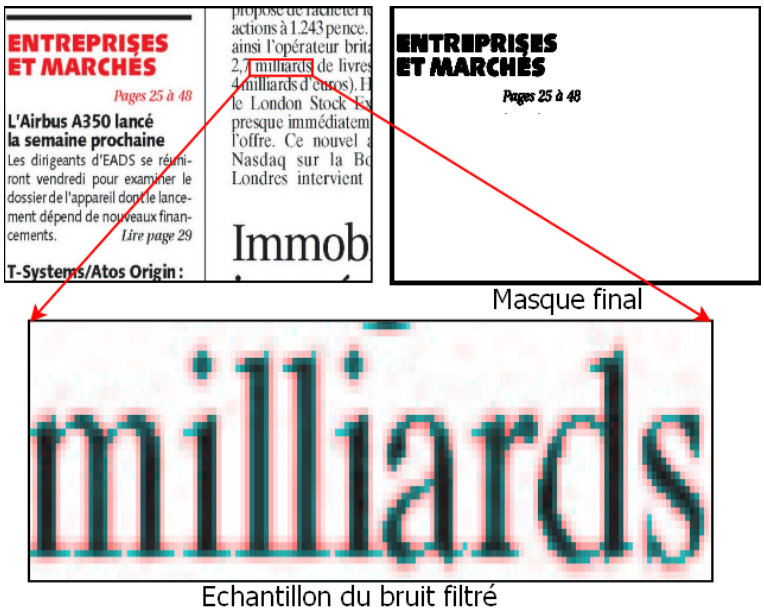


FIGURE I.14 – Fragment de $\mathcal{M}_F$ sur une image bruitée

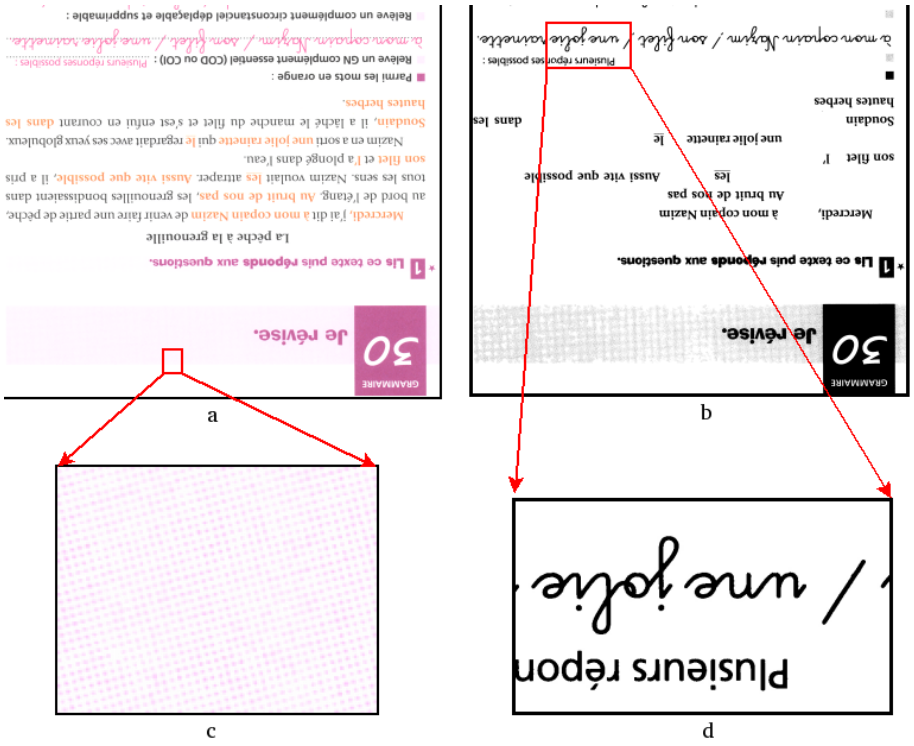


FIGURE I.15 – a. Image I, b. $\mathcal{M}_F$ (), c. Tramage chromatique, d. Texte chromatique.

5.5 Bilan

Nous avons présenté dans cette section un système générique permettant l'élimination du bruit de saturation si bien que la détection des zones chromatiques atteint des résultats remarquables.

a. Fragment de I b. Fragment de $\mathcal{M}_F$

FIGURE I.16 – Résultat sur une image excessivement bruitée

La méthode proposée est hybride (chaque pixel est analysé localement en considérant son proche voisinage tandis que le seuil de saturation est estimé sur l'image globale) et indépendante du modèle du document à traiter : aucune information *a priori* n'est nécessaire.

Grâce à l'estimation préalable de l'épaisseur des traits, aucun paramètre n'intervient dans notre processus.

Une nouvelle formule de la saturation a prouvé sa pertinence et ses atouts par rapport aux formules de saturation classiques.

Des tests lancés sur une base d'images variées et particulièrement bruitées attestent de l'efficacité de la segmentation et, notamment, de sa grande précision.

Les couches chromatiques et achromatiques séparées dans cette section seront segmentés, à leur tour, dans les deux prochaines sections.

6 Séparation chromatique

Une agglomération de pixels chromatiques forme une zone soit monochromatique ou bien multi-chromatique. Dans cette section, nous montrerons comment séparer les couleurs unies des régions monochromatiques en différentes régions tout en gardant les zones multi-chromatiques (comme les photos) intactes. Notons que les éléments de texte écrits sur un fond multicolore sont considérés comme faisant partie d'une zone multi-chromatique.

C'est également à ce stade que le bruit de teinte, causé par une inadéquation entre la résolution de numérisation et le tramage, sera filtré (*cf.* Figure I.2).

6.1 choix et motivations

Comme nous l'avons mentionné précédemment, loin de nous intéresser à la façon dont un document est perçu par l'œil humain, nous nous focalisons sur la manière dont il fut conçu par son auteur. L'usage de modèles perceptuels, tel que La*b*, serait donc hors de propos dans notre cadre d'étude.

Nous nous plaçons dans l'espace colorimétrique TSV. Dans ce cadre, la teinte est le canal le plus discriminant en terme de séparation colorimétrique [117, 80]. Notre segmentation colorimétrique sera donc basée sur cet axe.

La teinte $\mathcal{T}$ est une fonction circulaire, c'est-à-dire que $\mathcal{T}(0°) = \mathcal{T}(360°)$. Rappelons sa formule ramenée à l'intervalle $[0, 360]$ dans le domaine TSL. En utilisant les définitions de la formule I.4, $\mathcal{T}$ est définie en tout point p par :

$$\mathcal{T}(p) = \begin{cases} 0, & \text{si } C(p) = 0 \\ \frac{V_p - B_p}{C(p)} \mod 6, & \text{si } M(p) = R_p \\ \frac{B_p - R_p}{C(p)} + 2, & \text{si } M(p) = V_p \\ \frac{R_p - V_p}{C(p)} + 4, & \text{si } M(p) = B_p \end{cases} \tag{I.8}$$

Il serait impossible d'analyser l'histogramme de teinte global d'une image comportant, à la fois, des zones monochromatiques et des régions multi-chromatiques. En effet, Les informations provenant des différentes zones locales se superposent dans l'histogramme global et il devient donc impossible de les séparer. Notre approche sera donc locale dans un premier temps. Une analyse hybride sera effectuée ultérieurement afin de consolider la segmentation.

6.2 Définition des zones de traitement local

Une zone de traitement de taille trop petite ne contient pas assez d'information pour caractériser la zone et la classer correctement. Une zone trop grande, en revanche, risque d'englober des éléments disparates qui ne doivent pas être affectés à la même classe. Ainsi, il existe une taille pour laquelle la quantité d'information est maximale.

Une première possibilité consiste à associer une zone de traitement à chaque connexité du masque $\mathcal{M}_F$. Néanmoins, ces composantes connexes correspondent souvent à des zones locales isolées (un caractère isolé ou une portion d'un graphique). Ces zones isolées ne comprennent pas une quantité d'information suffisante pour les représenter. Nous proposons donc de choisir les connexités du masque $\mathcal{M}_A$ (voir Figure I.13.a) à la place. Une composante connexe de ce masque correspond souvent à un ensemble de caractères formant un mot, à une portion de ligne ou de paragraphe ou à une zone graphique. Le contenu d'une telle zone est donc cohérent et homogène.

6.3 Détection des zones multi-chromatiques

Une zone locale dont l'histogramme de teinte présente des pics larges correspond à une zone multi-chromatique tandis que les pics fins révèlent la présence de couleurs pures (zones monochromatiques).

Ainsi, Si l'histogramme de teinte $\mathcal{H}_i^l$ relatif à une zone de traitement i présente un (ou plusieurs) pic de largeur supérieure à 10% du spectre colorimétrique, la zone i est classée multi-chromatique.

6.4 Segmentation dans les zones monochromatiques

Les couleurs pures des zones monochromatiques seront séparées en différents masques. Ces derniers peuvent être, éventuellement, quantifiés si l'application qui découle de la segmentation colorimétrique l'exige (comme par exemple une compression MRC).

À partir d'un modèle colorimétrique construit à partir des histogrammes locaux, chaque pixel est assigné à l'une des classes du modèle, c'est-à-dire à l'un des masques monochromatiques.

Les histogrammes $\left\{\mathcal{H}_i^l\right\}_i$ relatifs aux zones multi-chromatiques seront, désormais, ignorés dans cette section.

6.4.1 Création du modèle colorimétrique

Un modèle colorimétrique est construit à partir des modes significatifs de l'histogramme global $\mathcal{H}^g$.

$\mathcal{H}^g$ est créé en additionnant les histogrammes locaux $\left\{\mathcal{H}_i^l\right\}_i$. Les zones multi-chromatiques étant exclues de l'histogramme global, ce dernier ne contient que des couleurs discrétisées (pures).

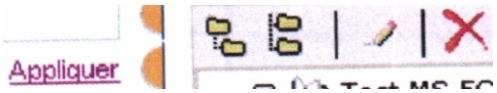
Une classe colorimétrique est associée à chacun des modes de $\mathcal{H}^g$.

6.4.2 Classement des pixels

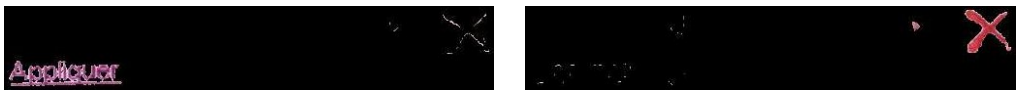
Deux pixels de couleurs différentes peuvent être présents côte à côte dans une même zone (connexité). Par conséquent, chaque pixel sera assigné à l'une des classes du modèle de façon individuelle. Par ailleurs, seul un classement au niveau pixel permet de filtrer les points bruités qui peuvent être parfaitement mêlés à leur voisins d'une même connexité.

Chaque pixel sera classé après une double validation faisant intervenir son histogramme local $\mathcal{H}_i^l$ ainsi que $\mathcal{H}^g$.

Justification de la double validation La plupart des zones monochromatiques contiennent du bruit de teinte. En illustration, la figure I.17 [2] montre deux masques résultant d'un simple algorithme de classement selon lequel chaque pixel est assigné à son plus proche voisin appartenant au modèle issu de $\mathcal{H}^g$. Dans cette figure, les pixels violets entourant la croix rouge illustrent bien l'insuffisance de cette approche à éliminer le bruit.



Un fragment d'image quelconque



Deux couches monochromatiques

FIGURE I.17 – Exemple de bruit de teinte non filtré par une simple validation.

Hypothèse Il est fort probable qu'un pixel bruit p soit placé loin des positions des modes dans l'histogramme global. De même, dans un contexte local, il serait assigné à un pic aléatoire de $\mathcal{H}_i^l$. Par ailleurs, $\mathcal{H}^g$ contient probablement plus de modes que $\mathcal{H}_i^l$; p est donc susceptible d'être attribué à une classe différente dans l'histogramme global.

Ainsi, si une incohérence est décelée lors du classement d'un pixel entre les niveaux local et global, nous considérons que ce dernier fait partie du bruit de teinte ; autrement, il est assigné à la classe du mode le plus proche dans $\mathcal{H}^g$.

Algorithme Soient $\mathcal{M}^g(p)$ le mode le plus proche d'un pixel p dans $\mathcal{H}^g$ et $\mathcal{M}^l(p)$ son plus proche mode dans $\mathcal{H}_i^l$. Les variations colorimétriques entre les différentes régions d'une même image font qu'un mode de l'histogramme local est souvent légèrement différent de son correspondant dans $\mathcal{H}^g$. Il serait donc imprudent de comparer les valeurs de $\mathcal{M}^g(p)$ et $\mathcal{M}^l(p)$ directement. Nous proposons donc de procéder comme suit :

1. calculer $\mathcal{M}^g(\mathcal{M}^l(p))$, le plus proche mode de $\mathcal{M}^l(p)$ dans $\mathcal{H}^g$
2. si $\mathcal{M}^g(p) = \mathcal{M}^g(\mathcal{M}^l(p))$ alors p est assigné à la classe associée à $\mathcal{M}^g(p)$; sinon p est un pixel bruit.

Illustrons notre approche par l'exemple de la figure I.18 :

- cas du pixel bleu p de teinte 150 : $\mathcal{M}^g(p) = 159$, $\mathcal{M}^l(p) = 160$, $\mathcal{M}^g(\mathcal{M}^l(p)) = 159$; $\mathcal{M}^g(p) = \mathcal{M}^g(\mathcal{M}^l(p))$; p est donc assigné au masque bleu ;
- cas du pixel rose p' de teinte 5 : $\mathcal{M}^g(p') = 8$, $\mathcal{M}^l(p') = 21$, $\mathcal{M}^g(\mathcal{M}^l(p')) = 21$; cette fois, $\mathcal{M}^g(p') \neq \mathcal{M}^g(\mathcal{M}^l(p'))$; p' est détecté comme bruit.

2. Les images sont de fond noir pour des raisons de visibilité.

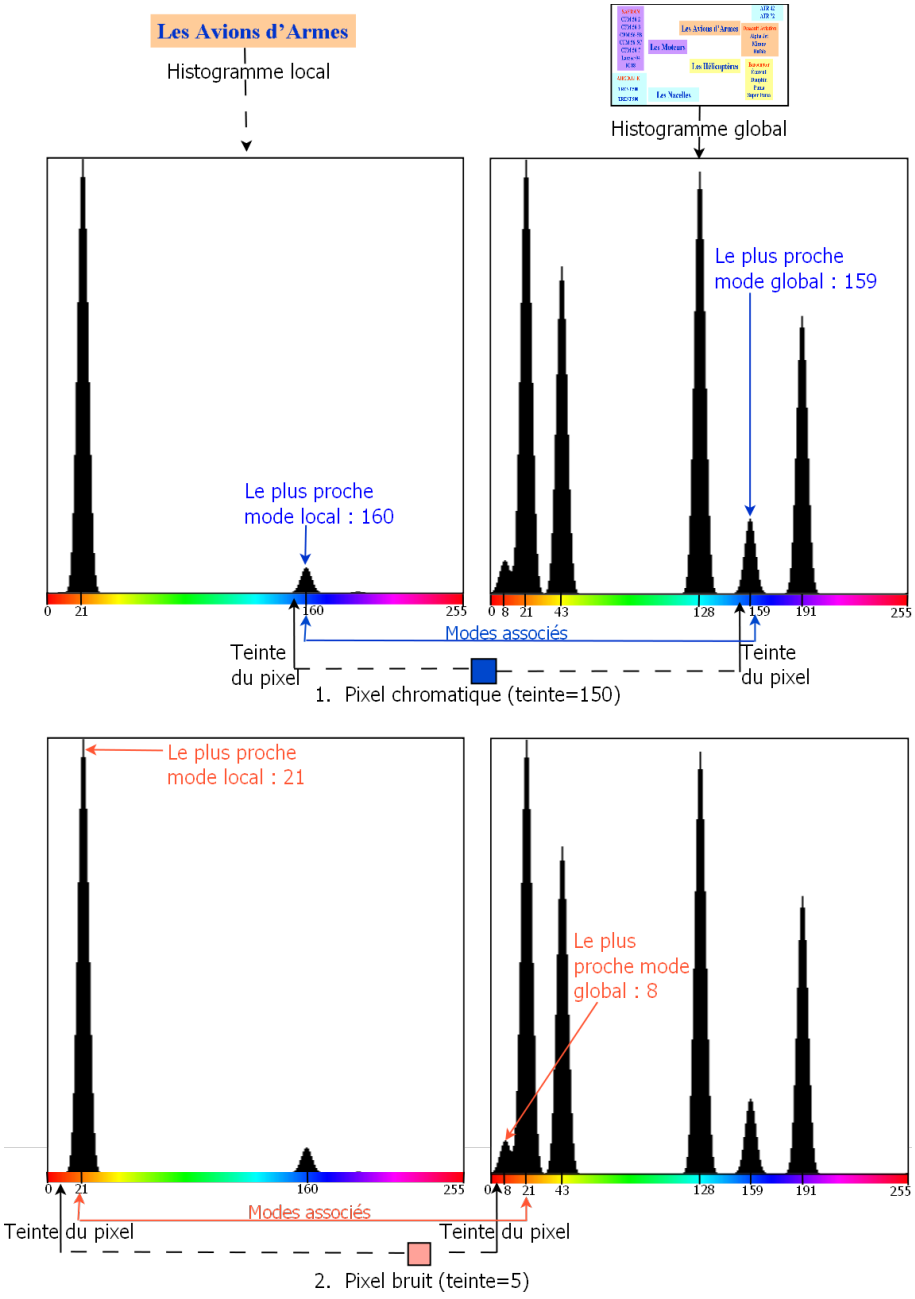


FIGURE I.18 – Classement d'un pixel monochromatique et un pixel bruit respectivement

6.5 Résultats

La mise en place d'une évaluation quantitative de notre segmentation chromatique s'avère difficile, voire irréalisable faute de disposer de toute source électronique d'origine.

Une validation indirecte via des applications se basant sur notre système de séparation colorimétrique sera fournie dans le prochain chapitre.

Nous avons visuellement évalué les résultats sur environ 300 images provenant de sources variées. Quelques échantillons seront présentés dans cette section.

FIGURE I.19 – Classes de teinte

Comme illustré dans les figures I.20 et I.19 [3], les résultats sont globalement très satisfaisants.

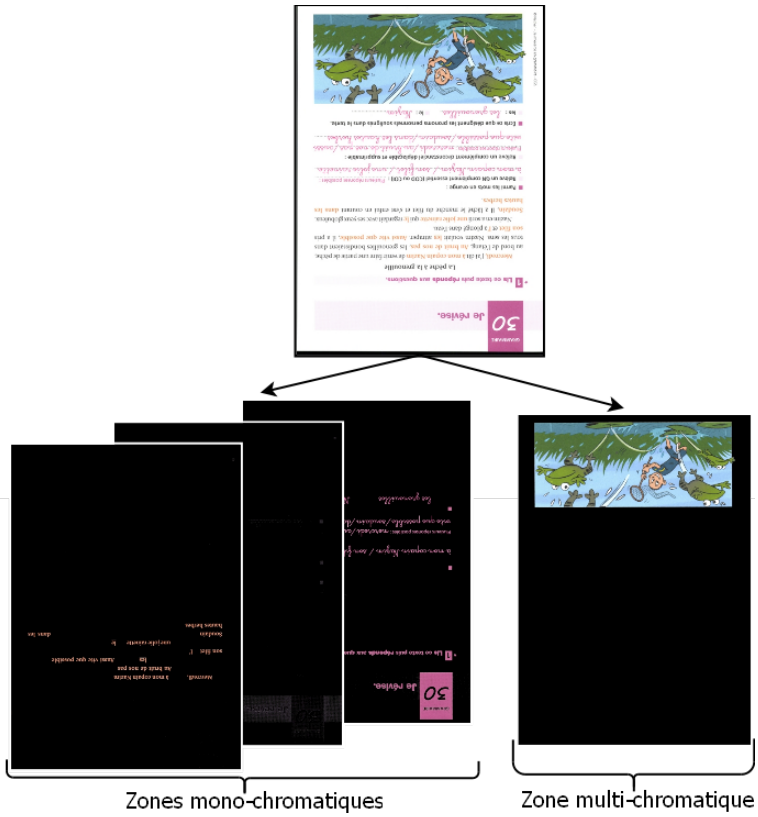


FIGURE I.20 – Résultat d'une segmentation chromatique

La figure I.20 montre un exemple de zone multi-chromatique correctement détectée.

3. Pour des raisons de visibilité, toutes les couches chromatiques sont affichées sur un fond noir.

6.6 Conclusion

Dans cette section, la couche chromatique a été segmentée selon une approche nouvelle et propre aux images de documents. Une partie considérable du bruit de teinte a été filtrée.

La couche achromatique sera segmentée, à son tour, durant la prochaine section.

7 Séparation achromatique

La couche achromatique est séparée en deux classes : le masque noir et blanc $N\&B$ et le masque $Gris$. Nous proposons, dans cette section, une approche similaire à celle que nous avons proposée pour réaliser la segmentation chromatique, à la différence que nous utilisons, cette fois, des histogrammes de luminance. C'est également le bruit de luminance qui sera filtré dans cette partie.

Rappelons d'emblée la formule de la luminance, canal sur lequel seront basés nos calculs. En tout pixel RVB p, la luminance $\mathcal{L}$ est définie dans le modèle TSV par :

$$\mathcal{L}_p = \frac{1}{3}(R_p + V_p + B_p) \tag{I.9}$$

La segmentation achromatique passe par trois principales étapes :

1. une segmentation partielle qui assignera les pixels blancs de l'image à la classe $N\&B$; il s'agit d'un classement simple et rapide appliqué au niveau pixel,
2. un classement local qui permettra la catégorisation des pixels restants ; cette étape s'effectuera au niveau régional ; les zones de traitement sont directement déduites du résultat du classement de la phase précédente ;
3. un post-traitement permettant une vérification contextuelle des zones classées en s'appuyant sur leurs voisinages respectifs ainsi qu'une détection et un reclassement des filets (*cf.* paragraphe 7.3.2).

7.1 Niveau global

Afin d'optimiser le temps d'exécution, les pixels triviaux qui affichent les plus fortes valeurs de luminance sont immédiatement assignés à la classe $N\&B$.

Pour ce faire, nous identifions, dans l'histogramme global de luminance lissé (lissage uniforme), deux seuils : $\mathcal{B}$ et $\mathcal{N}$. La valeur $\mathcal{B}$ est celle du dernier mode (le plus grand) significatif de l'histogramme. Tel que les zones blanches dans la figure I.21.c, les pixels de luminosité supérieure à ce dernier seuil sont considérés blancs ; ils sont donc assignés à la classe $N\&B$.

$\mathcal{N}$ correspond au premier mode de l'histogramme. Ce seuil sera utilisé au cours des prochaines sections.

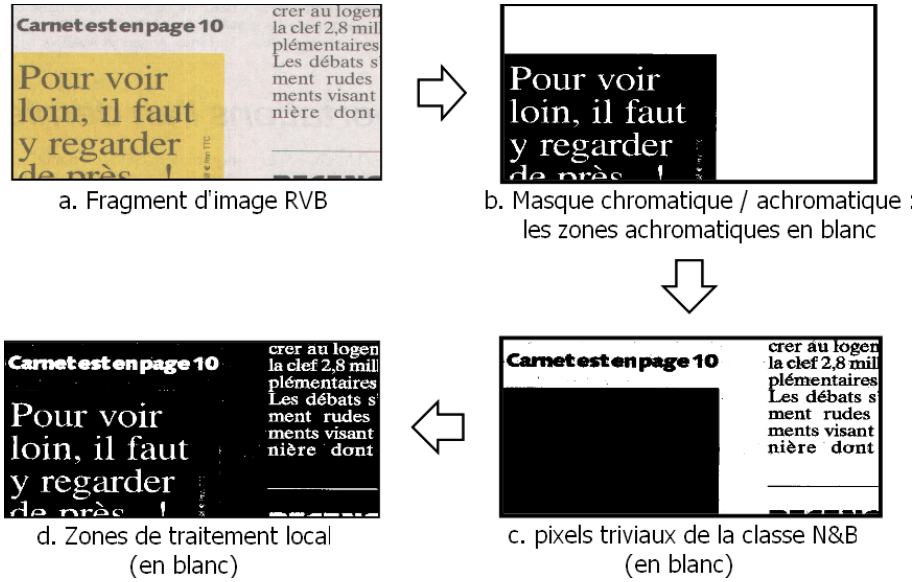


FIGURE I.21 – Classement partiel et zones de traitement local

7.2 Niveau local

La numérisation, notamment dans de mauvaises conditions d'éclairage, altère souvent les traits noirs si bien qu'ils apparaissent gris dans l'image produite. Ainsi, le classement individuel d'un pixel, indépendamment de son voisinage, serait biaisé par ce bruit de luminance. C'est pour cette raison que nous optons pour une catégorisation de niveau local.

Les zones de traitement sont simplement données par l'agglomération en connexités des pixels achromatiques non classés lors de la phase précédente (les pixels non blancs). En illustration, chaque zone blanche dans la figure I.21.d correspond à une zone de traitement.

Connexités de grande taille Contrairement aux nombreuses petites connexités qui représentent souvent des caractères de texte noir, les zones achromatiques spacieuses [4] correspondent probablement à des graphiques en niveaux de gris. Nous considérons donc que ces dernières zones forment un cas particulier nécessitant un traitement approprié.

Les zones de grande taille peuvent faire l'objet de gros titres noirs. De telles régions sont plus robustes au bruit de luminance : elles contiennent un faible taux de pixels gris.

4. La taille d'une zone correspond à la surface de son rectangle englobant. Le seuil sur la taille des zones vaut 6075 $\mathcal{S}_t$

Ainsi, si une grande connexité est composée de plus de 90% de pixels dont la luminance est inférieure à $\mathcal{N}$, cette zone est assignée à la classe $N\&B$; autrement, elle est évidemment affectée à la classe $Gris$.

Connexités de tailles moyennes ou petites Un vecteur bidimensionnel de caractéristiques est associé à chacune des zones de traitement restantes. Cette signature est composée du premier (le plus petit) mode de valeur non négligeable ainsi que de la largeur du pic correspondant à ce mode. La dernière caractéristique reflète la dispersion des valeurs de luminosité dans la zone ; ainsi, ses grandes valeurs correspondent probablement à des zones grises.

Maintenant qu'une partie des zones de l'image sont classées, nous recalculons le seuil $\mathcal{N}$ afin de l'affiner.

En effet, $\mathcal{N}$ étant calculé sur l'histogramme global, toutes les zones interviennent lors de l'estimation de ce seuil. Or, les traits fins sont généralement plus lumineux que les grandes zones (cela est dû à une négligence généralisée du théorème d'échantillonnage de Shannon-Niquist).

Il est donc plus pertinent d'estimer une nouvelle valeur de $\mathcal{N}$ à partir d'un histogramme ne faisant intervenir que les zones candidates. Nous recalculons ainsi $\mathcal{N}$ sur l'histogramme de luminance des zones petites ou moyennes.

La catégorisation est accomplie en assignant une zone à la classe $N\&B$ si et seulement si les valeurs de ses deux caractéristiques sont respectivement inférieures à $\mathcal{N}$ et la largeur du pic correspondant à $\mathcal{N}$ dans l'histogramme.

7.3 Post-traitements

7.3.1 Vérification contextuelle

À ce stade, tous les pixels achromatiques sont classés. Une information contextuelle est donc établie entre les entités voisines. Nous profitons de cette information pour recatégoriser les petites connexités qui s'avèrent mal classées.

Une région contenant de nombreuses connexités $N\&B$ correspond probablement à une zone de texte. De ce fait, si une connexité de petite taille et de type $Gris$ avoisine plusieurs zones noires, cette dernière est réaffectée à la classe $N\&B$. De même, une petite connexité de type $N\&B$ avoisinant plusieurs connexités grises se voit réassignée à la classe $Gris$.

7.3.2 Détection des filets

Les Filets consistent en des zones filiformes comme par exemple les bordures de tableaux, les cadres, les lignes, *etc.* De tels éléments sont souvent formés de traits fins et

dégradés. Ils contiennent donc très peu de noir et beaucoup de gris. Ainsi, l'étape précédente les affecte à la classe *Gris*. Cette étape vise à détecter ces zones filiformes afin de les réassigner à la classe $N\&B$.

L'algorithme ci-dessous est appliqué à toutes les grandes connexités de la classe $Gris$:

1. Une érosion morphologique est appliquée aux zones concernées. La taille de l'élément structurant est expérimentalement fixée à $(2{\cdot}\mathcal{S}_t){\times}(2{\cdot}\mathcal{S}_t)$. L'érosion devrait remplacer les traits fins noirs par des pixels de la même couleur que le fond local de l'image.
2. Si tous les pixels sombres (dont la luminance est inférieure à $\mathcal{N}$) ne subsistent pas à cette dernière transformation, un filet est détecté et la zone est donc réassignée à la classe $N\&B$.

7.4 Résultats

Nous rappelons que nous ne disposons d'aucune source électronique d'origine et que toute évaluation quantitative sera donc inaccessible dans ce cas. De même que la segmentation chromatique, la séparation achromatique sera évaluée indirectement via des applications basées dessus.

Quelques échantillons d'images résultats sont affichés dans la figure I.22. Pour des raisons de clarté de l'affichage, les zones classées $N\&B$ sont binarisées et présentées sur un fond blanc ; les pixels de la classe $Gris$ superposent un fond bleu. Ces images démontrent de l'efficacité de la méthode proposée, même sur les documents les plus dégradés.

Comme le montre le dernier échantillon de cette dernière figure, notre approche assigne les zones ambigües, comme par exemple du texte superposé, à la classe *Gris*. Ceci permet d'éviter une éventuelle perte d'information lors de la binarisation du masque $N\&B$.

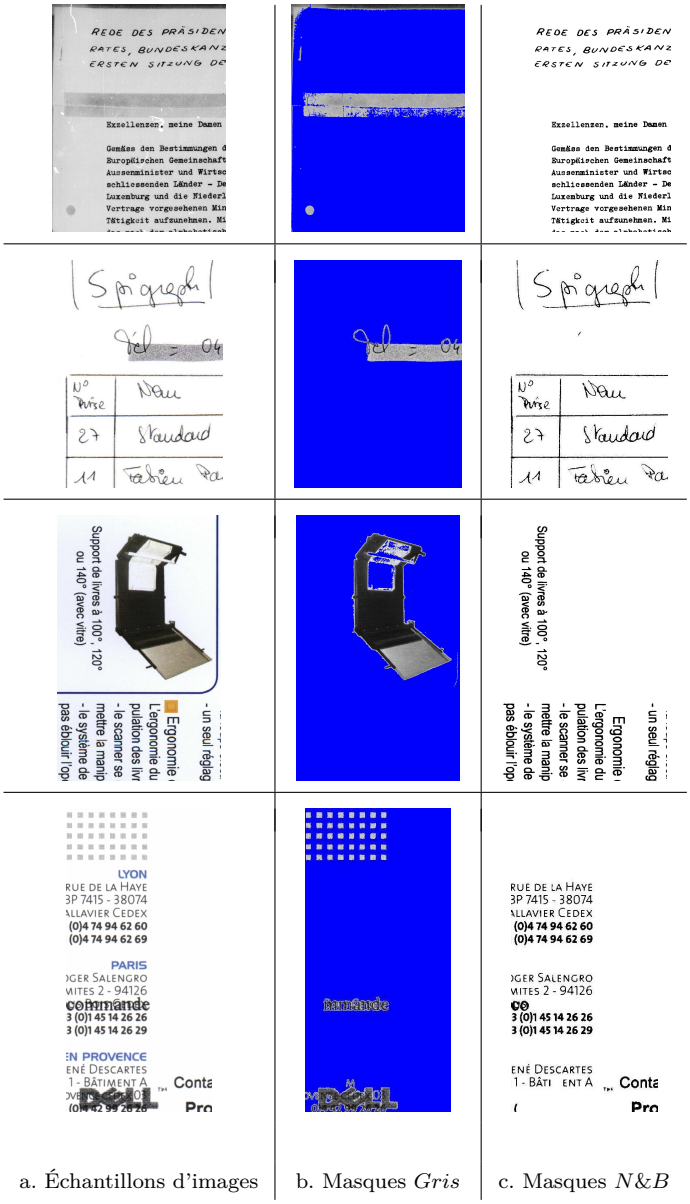
7.5 Conclusion

Dans cette section, nous avons abordé la problématique nouvelle et prometteuse d'une segmentation qui prépare à une binarisation intelligente. Ce traitement permet d'améliorer la netteté et la lisibilité de l'image et de simplifier l'analyse et la reconnaissance des contenus.

8 Bilan des résultats

Nous présenterons, dans cette section, des images représentatives et variées issues de notre système de segmentation colorimétrique global.

Certains de nos résultats seront comparés à ceux issus de deux méthodes de quantification de la littérature conçues pour les images de documents.



a. Échantillons d'images | b. Masques *Gris* | c. Masques *N&B*

FIGURE I.22 – Exemples de résultats de la séparation achromatique.

8.1 Évaluation

8.1.1 Qualité de la segmentation

Nous recomposons chaque image à partir des différentes classes colorimétriques issues des séparations achromatique et achromatique. La recomposition consiste en une simple addition des masques résultats. Les imagettes de la figures I.23 montrent quelques exemples d'images issues de notre système de segmentation.

Cette dernière figure montre des images nettes et de meilleure qualité que les originales. Cette amélioration est due au fait que nous avons pu déterminer automatiquement les zones binarisables sans perte d'information.

1. 2.

FIGURE I.23 – 1. Échantillons d'images, 2. Images recomposées

Le texte net et lisible ainsi que les détails colorimétriques conservés dans les imagettes de la dernière ligne du tableau I.26 témoignent de la capacité de cette approche à assurer une quantification sans perte d'information. Par ailleurs, le texte correctement assigné à une même classe colorimétrique prépare le terrain à l'OCR et assure donc une reconnaissance de texte efficace.

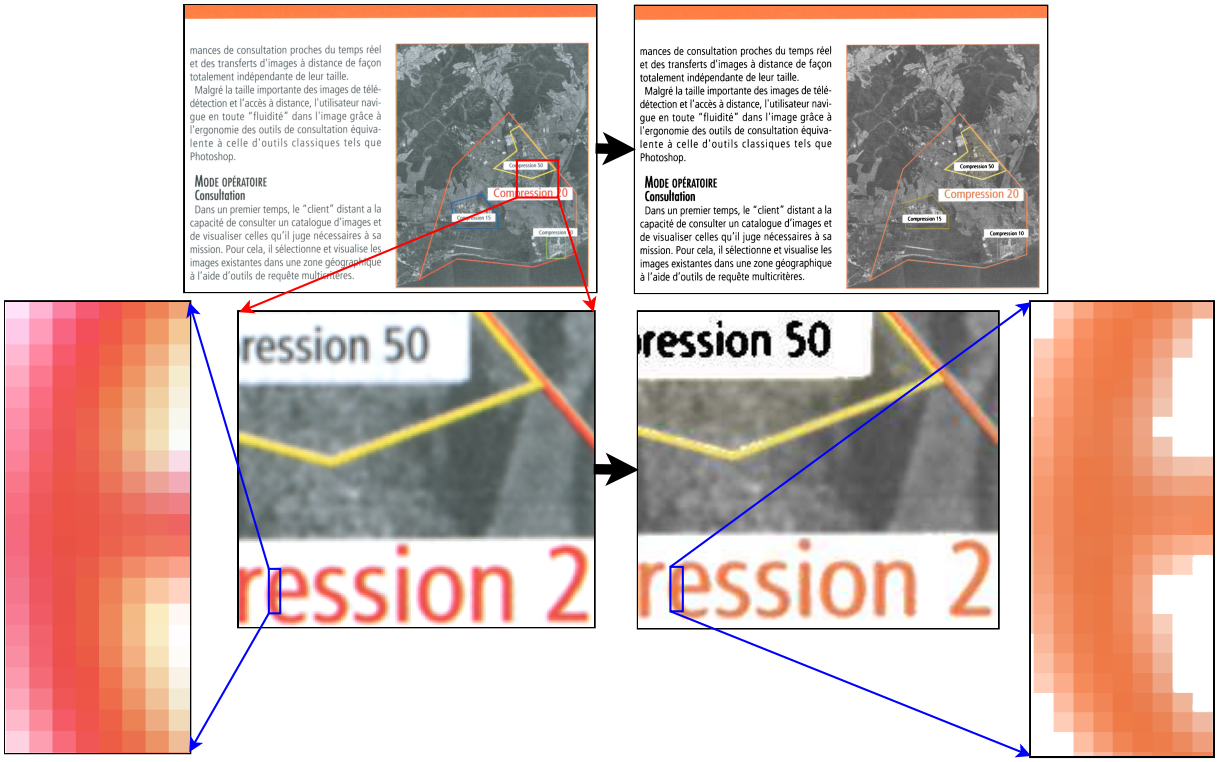


FIGURE I.24 – Recomposition d'un dégradé de teintes.

Dans nos images recomposées, certaines couleurs paraissent parfois différentes des originales (voir Figure I.24). En effet, nous rappelons que le système de séparation chromatique calcule la teinte relative à chaque zone de façon différente et indépendante du modèle de perception humaine.

L'image I.25 présente des zones de texte noir et coloré ainsi que des graphiques. Malgré le fond bruité, le texte noir est correctement séparé du fond. Néanmoins, certains pixels bleus sont rendus achromatique (voir la zone agrandie). Ces pixels correspondent à du bruit de teinte issu d'un tramage numérisé en basse résolution.

8.1.2 Temps d'exécution

Les masques $N\&B$ sont binarisés en utilisant l'algorithme Otsu [89]. Nous avons choisi cette méthode pour sa rapidité. En effet, étant donné que nous ne binarisons que les zones binaires dans le support original, non perturbées par la présence d'éléments graphiques

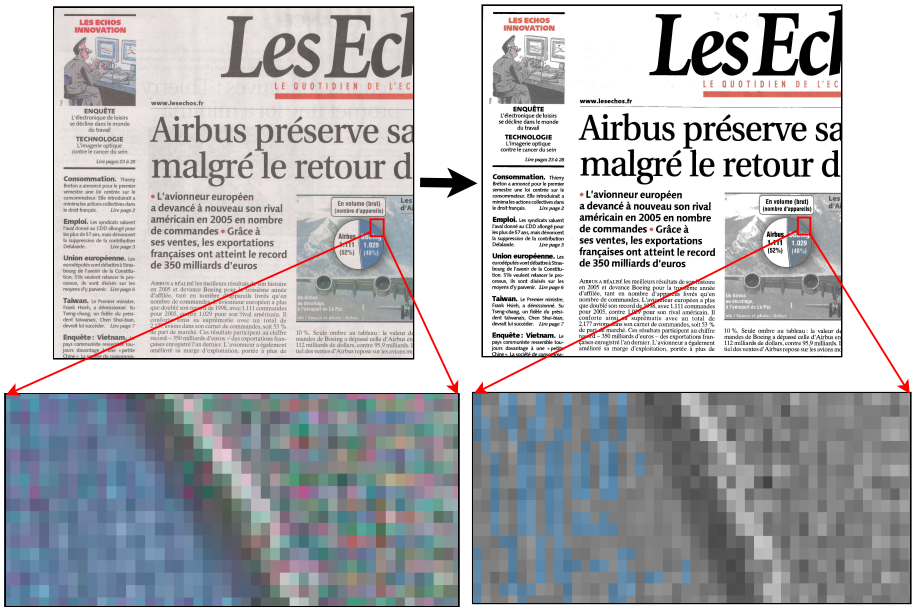


FIGURE I.25 – Tramage numérisé en basse résolution.

Phase	Séparation chromatique/achromatique	Séparation chromatique	Séparation achromatique
Quota temporel	13%	11%	76%

TABLE I.2 – Temps d'exécution par phase

polychromes, il est justifié d'appliquer une telle méthode de seuillage global. Si, toutefois, le temps d'exécution ne constitue pas une contrainte, une méthode adaptative permet toujours de mieux détourer les caractères dans les images de faible résolution, comme par exemple le premier et le troisième exemples de la figure I.23.

En utilisant une machine à 2.8 GHz de fréquence du processeur et de 3.48 Go de RAM, le traitement d'une image de page A4 et de résolution 300 *dpi* dure 5 secondes environ, ce qui revient à 0.05 ms par unité de 1000 pixels.

La répartition du temps d'exécution entre les différentes phases de notre système est consignée dans le tableau I.2.

L'épaisseur des traits ($\mathcal{S}_t$) est une mesure indépendante de la segmentation colorimétrique et réutilisable pour d'autres applications. C'est pour cette raison que nous n'avons pas pris en compte le temps requis par cette phase lors de la mesure du temps d'exécution global.

Le temps d'exécution varie selon le contenu de l'image : l'algorithme est plus rapide sur les pages multicolores, telles que les couverts des magazines de mode, que sur les simples images de texte en niveaux de gris.

Par ailleurs, certaines phases peuvent être omises si l'on dispose d'une information *a priori* : il est possible, par exemple, de faire abstraction de la $3^{ème}$ étape si on sait que toutes les zones achromatiques correspondent à du texte et peuvent être binarisées sans

aucune perte.

8.2 Comparaisons

Comme nous l'avons mentionné précédemment, nous ne disposons d'aucun moyen de mesure de qualité des résultats sans passer par des applications annexes. Nous nous contentons donc d'une comparaison visuelle.

Notre approche a été comparée avec deux méthodes de finalités similaires. La première est une approche connue et communément utilisé par les systèmes MRC depuis plusieurs années, à savoir DjVu [42]. La seconde consiste en une méthode de quantification récente conçue pour les images de documents [84].

La deuxième ligne du tableau I.26 est relative aux résultats de la méthode de quantification décrite en [84]. Les résultats donnés par DjVu sont consignés dans la troisième ligne de cette figure. La dernière ligne de ce même tableau reflète les performances de notre système.

Nous pouvons déduire de cette figure I.26 que la méthode de quantification [84] segmente souvent les zones monochromatiques en plusieurs classes d'une part (comme la lettre 'e' bleue dans la première colonne) et engendre une perte d'information dans les régions multi-chromatiques d'autre part (exemple de zones photo dans la figure I.27).

En raison des distorsions dans les images scannées, la méthode de quantification [84] aussi bien que DjVu [42] ne permettent pas de séparer la couleur du texte de celle du fond efficacement. Le mot agrandi "soit" dans la figure I.27 montre que plusieurs classes colorimétriques sont aperçues dans un même caractère. Cette sur-segmentation pénaliserait la phase d'extraction de texte que nous mettrons en place ultérieurement.

9 Conclusion

Nous avons présenté dans ce chapitre un système de segmentation colorimétrique particulièrement efficace sur les images bruitées de par son aptitude à réparer les distorsions introduites par la chaîne de numérisation et de compression. Il s'agit, par ailleurs, d'une approche pragmatique adaptée au contexte industriel de par sa rapidité et son efficacité.

Tous les paramètres intervenant dans ce système sont évalués automatiquement grâce à une nouvelle mesure permettant d'estimer l'épaisseur des traits dans une image contenant du texte. Par ailleurs, la séparation colorimétrique proposée est non-supervisée et générique, donc applicable à tout type de document.

Ce système comporte trois phases complémentaires et indépendantes : chacune est réutilisable indépendamment des autres. La séparation chromatique / achromatique se base, notamment, sur une nouvelle formulation de la saturation ainsi qu'un ensemble

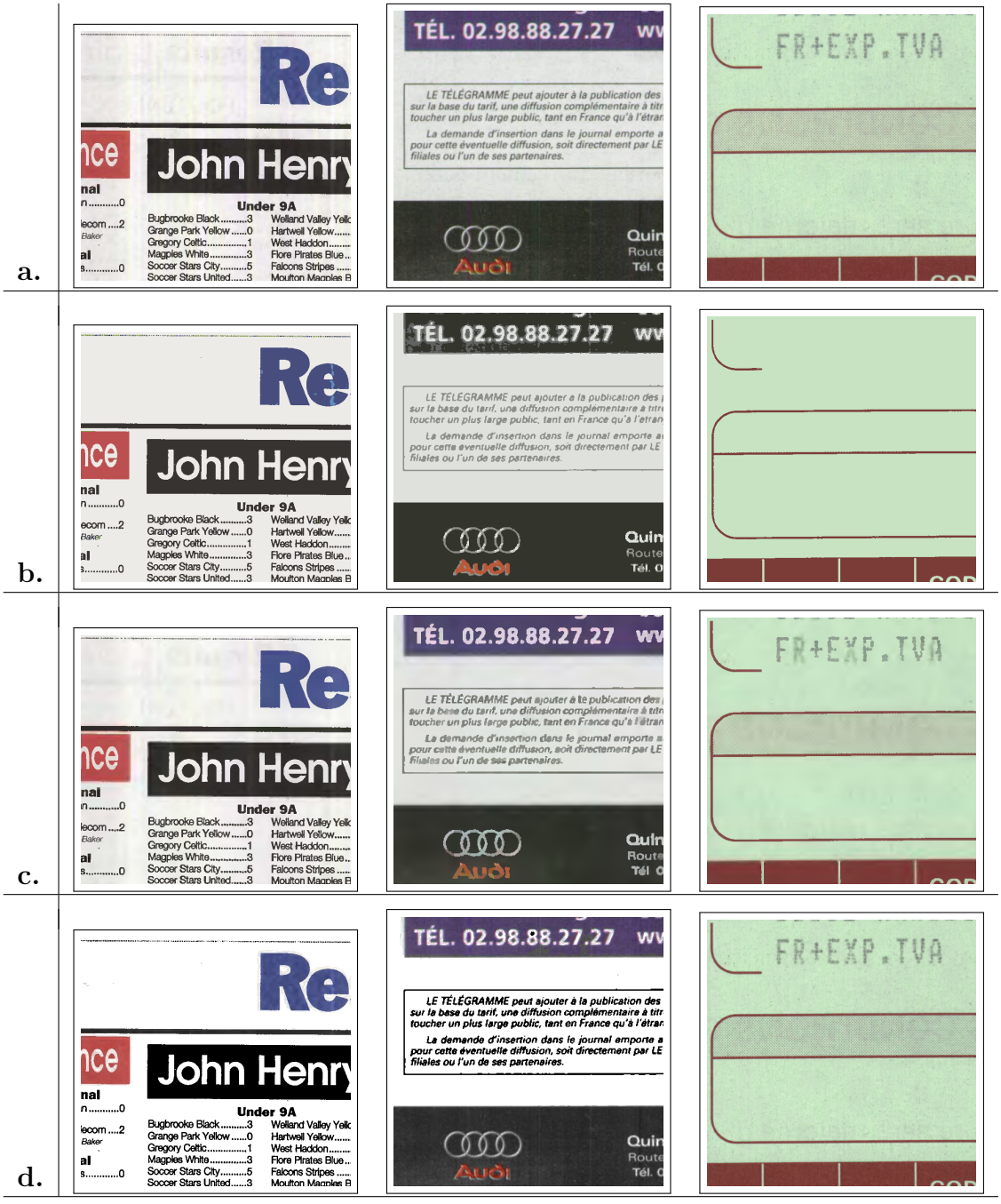
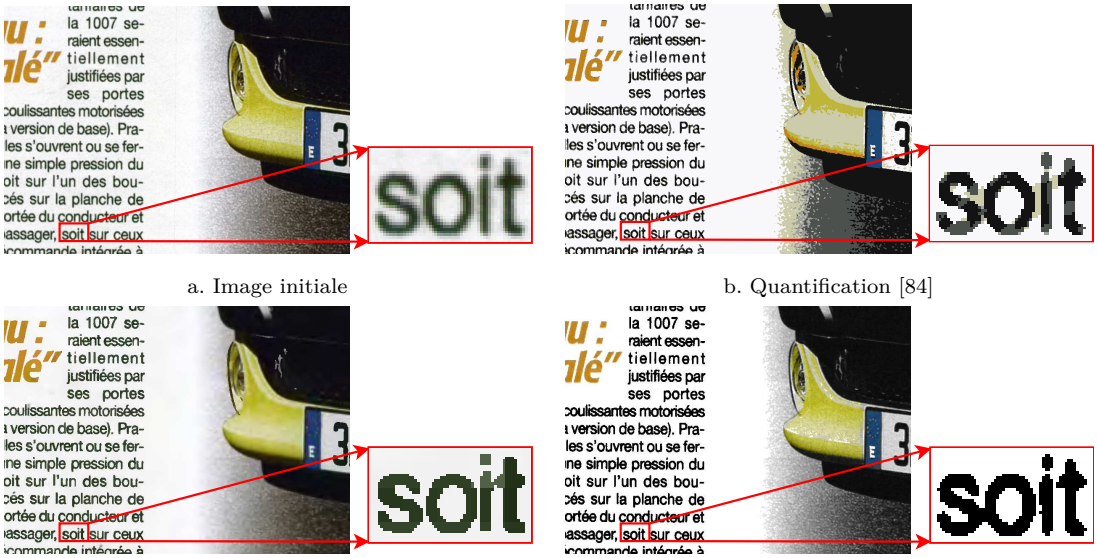


FIGURE I.26 – Résultats et comparaisons : a. Échantillons d'images, b. Méthode [84], c. DjVu [42], d. Notre approche.



a. Image initiale b. Quantification [84]

c. DjVu [42] d. Notre méthode

FIGURE I.27 – Comparaison de résultats.

de filtres éliminant le bruit de saturation. À la suite de cette phase, les segmentations chromatique et achromatique sont lancées parallèlement. Ces dernières reposent sur une double validation utilisant des histogrammes locaux et globaux pour le classement.

Aucune perte pénalisante d'information n'est engendrée par cette approche. En effet, seules les zones estimées de couleurs uniformes sont concernées par la segmentation (ou binarisation). Les zones multi-chromatiques ou grises où les nuances de couleurs apportent de l'information sont préservées.

Des tests sur une variété d'images bruités ont été lancés. L'aspect des images traitées témoigne de l'efficacité de la méthode à filtrer le bruit et à produire des images nettes. Une évaluation quantitative faisant appel à des applications directes et indirectes de ce système sera rapportée dans le prochain chapitre.

Chapitre II

Applications à la segmentation colorimétrique

Table des matières

1 Introduction et motivations

Les techniques récentes en multimédia donnent naissance à des images de plus en plus sophistiquées : une avalanche de couleurs et de structures complexes et non régulières sont souvent employées afin d'attirer l'attention du lecteur. De ce fait, maintes applications, comme par exemple l'extraction de texte, la compression MRC, la catégorisation de blocs de magazines et notamment la détection des publicités, seraient de rendu médiocre sans information sur les couleurs. D'autres tâches comme la catégorisation de blocs de magazines et notamment la détection des publicités sont inconcevables sans cette information.

Nous avons choisi deux applications importantes pour tirer parti de notre système de segmentation colorimétrique : Une extraction de texte permettant d'améliorer les résultats de l'OCR et une segmentation d'images de journaux et magazines en blocs suivie d'une classification qui permet de détecter les publicités.

Ces applications constituent, par ailleurs, une contribution majeure au projet Mediabox. Nous rappelons, en effet, que ce projet vise à automatiser la création des revues de presse, ce qui ne peut être réalisé sans connaissances sur le contenu structurel et textuel.

1.1 Extraction de texte pour améliorer les performances de l'OCR

Les OCR les plus performants, tel que le produit commercial Abbyy FineReader, sont parfois incapables d'extraire le texte écrit en inverse vidéo sur un fond coloré. Ces derniers sont encore moins efficaces quand le fond est irrégulier ou multicolore ou quand le texte est faiblement contrasté par rapport à ce dernier.

FineReader (même les dernières versions de ce produit) traque les éléments textuels dans des fenêtre de tailles prédéfinies en effectuant des binarisations localement adaptatives. De ce fait, les titres dont la taille dépasse les seuils fixés ne sont jamais détectés par l'OCR le plus puissant sur le marché actuel. Une localisation de texte suivie d'une réduction de taille appliquée, exclusivement, aux zones de texte de grande taille permettrait donc à l'OCR d'extraire et ainsi de reconnaitre des éléments de texte supplémentaires.

Une fois extraits, les éléments de texte sont facilement regroupés en mots, lignes ou paragraphes. Les blocs de texte formés permettent de réaliser une meilleure segmentation logique. Ce dernier point fait l'objet de travaux de recherche continus, et ce depuis une trentaine d'années, car aucune solution générique n'a été proposée à ce jour.

Les blocs de texte correspondant à des articles de presse peuvent être également mis en ligne afin de faciliter leur accès à un plus grand public. Pour ce faire, nous avons besoin d'un système de segmentation rapide, efficace et générique : une solution applicable à tout type de document, quel que soit sa structure physique, sa langue et sa catégorie.

1.2 Détection de publicités

La mise en ligne des articles de presse ne peut avoir lieu sans une classification préalable des blocs de texte extraits des images de presse. En effet, il serait inconvenant de laisser s'infiltrer une publicité parmi les articles réguliers, faute de la détecter.

La détection de publicités présente deux principaux intérêts :

- **Un objectif de filtrage** : comme nous l'avons mentionné précédemment, il est nécessaire de détecter les zones publicitaires afin d'éviter de les confondre avec les articles réguliers. La confusion d'un article examinant un produit donné avec une publicité sur le même produit est particulièrement problématique.
- **Un objectif de pointage** : nombre d'acteurs commerciaux s'acquittent des espaces publicitaires dans des journaux et magazines. Leur fournir des moyens automatiques et rapides pour accéder aux publicités les concernant leur économiserait donc un temps considérable.

De façon plus générale, la classification de blocs dans les images de presse facilite l'indexation des articles et accroît ainsi leur accessibilité. L'automatisation de cette tâche permet d'épargner le temps de filtrage et de triage à différents utilisateurs.

1.3 Solutions proposées

L'information colorimétrique obtenue par notre approche décrite dans le précédent chapitre joue un rôle majeur aussi bien dans la détection de texte que dans la segmentation en blocs et la classification de ces derniers. Notre analyse colorimétrique permet aussi d'améliorer les résultats de l'OCR sur les textes colorés imprimés sur des fonds de couleurs et de fournir une information textuelle correcte qui est nécessaire à des applications avancées comme la catégorisation de documents et la détection de publicités par exemple.

Nous présenterons dans la prochaine section une méthode simple et efficace basée sur le groupement *intra*-classe de connexités ainsi que sur la technique de gradients cumulés

qui permet d'extraire des lignes de texte dans des images de natures diverses et variées. Nous passerons en revue les travaux antérieurs concernant la détection de texte avant de présenter notre approche dans la section 2. Nous évaluerons ensuite les performances de l'OCR FineReader sur les images résultant de notre méthode d'extraction de texte. Cette évaluation permettra de mesurer l'apport de notre contribution aux produits existants les plus performants.

Les lignes de texte sont regroupées pour former des articles. Les éléments graphiques sont détectés et regroupés en blocs homogènes. Toutes ces unités forment des candidats à une classification qui permettra, entre autres, de détecter les publicités. Le classement repose sur des descripteurs intuitifs utilisant l'information colorimétrique et textuelle acquises antérieurement.

2 Extraction de texte

Nous recherchons les zones de texte dans chacune des classes produites par le système de segmentation colorimétrique. Pour ce faire, nous proposerons une méthode simple et efficace basée principalement sur le regroupement de connexités homogènes. Le choix de simplicité et rapidité est guidé par des contraintes industrielles d'une part et par notre résolution de mettre à l'épreuve notre méthode de classification de couleurs d'autre part. L'évaluation quantitative via cette application enrichira le jeux de tests présenté dans le chapitre précédent.

Après une étude de l'état de l'art qui permettra de situer notre contribution par rapport l'existant, nous présenterons notre approche d'extraction de texte. Les performances de FineReader en matière de segmentation de lignes seront ensuite mesurées et commentées.

2.1 État de l'art

La littérature comprend une grande variété d'approches allant de la détection à la reconnaissance de texte dans des images ou des vidéos, et ce pour répondre à besoins divers comme la segmentation logique d'une page de document, l'amélioration de la lisibilité, le tri automatique de courrier postal, *etc.* Chaque application concerne, en général, un type de document bien déterminé et la méthode employée pour y parvenir est difficilement réutilisable pour d'autres catégories de documents. Ainsi, malgré un état de l'art dense et varié [57], l'étude de l'existant ne révèle aucune méthode multi-usages ni complètement automatisée [82].

Selon le type d'image à traiter, nous regroupons les méthodes d'extraction de texte en

deux catégories. Les approches structurelles concernent les images binaires ou facilement binarisables de documents réguliers. La segmentation structurelle consiste à localiser récursivement toutes les zones contenant des données homogènes : blocs de texte, zones graphiques, tableaux, *etc.* La seconde catégorie d'approches s'attache à extraire le texte écrit sur un fond non-uniforme : structuré ou multicolore.

2.1.1 Approches structurelles

Nous distinguons deux familles d'approches menant à une segmentation de la structure physique : les approches descendantes procédant par découpage récursif à partir des espaces blancs (où il n'y a pas de texte) et les méthodes (descendantes) de segmentation par fusion récursive des objets entre eux de proche en proche.

Les méthodes de segmentation par fusion Les méthodes de segmentation par fusion consistent à regrouper les objets élémentaires (pixels, composantes connexes, groupes de connexités faiblement espacés) récursivement en suivant des règles de fusion des blocs de proche en proche. Ces méthodes sont simples à mettre en œuvre et ne nécessitent pas un modèle sur le contenu des images.

Les méthodes ascendantes sont regroupée en trois catégories :

Approches par regroupement de connexités : les différentes catégories de connexités d'une image binaire sont différenciées par leurs tailles et leurs alignements (les connexités alignées et de hauteur moyenne représentent des caractères ou des groupes de caractères qui peuvent être assemblées en lignes de texte, les connexités non alignés quel que soit leurs tailles ne peuvent pas être du texte, *etc.*). À partir de ces critères, ces composantes sont regroupées en zones de texte (mots, lignes, paragraphes...) ou en éléments graphiques [30, 20].

Le Run Length Smoothing or Smearing Algorithm (RLSA) est un filtre qui agglomère (smoothing / smearing) les séquences (Run-Length) de pixels noirs en fonction de la longueur des espaces. La longueur des séquences dépend d'un paramètre appelé Contrainte C. Avec des valeurs croissantes de C, les caractères sont regroupés en mots, les mots en lignes... Des variantes ont été introduites [113, 4, 108] pour éviter de fixer des seuils arbitraires sur ce paramètres.

Les méthodes de segmentation structurelle, entre autres le RLSA, ne sont applicables que sur des images binaires. Pour remédier à cette contrainte, Strouthopoulos *et al.* quantifient l'image en utilisant un réseau de neurones non supervisé et appliquent ensuite cette

approche structurelle sur chacune des images masques associées aux couleurs résultant de la réduction colorimétrique.

La segmentation par changement d'échelle [86] revient à appliquer des filtres de flou Gaussien et un sous-échantillonnage progressif des images. Sur les images binaires, il suffit de prendre la valeur majoritaire des pixels sur une zone de taille (dx, dy) pixels et la copier dans une image plus petite à une échelle $(1/dx, 1/dy)$ de l'image initiale. Si les valeurs des paramètres sont correctement choisies, ce ré-échantillonnage permet de coller les entité homogènes, comme par exemple les mots de la même ligne, et ainsi de les extraire.

Les méthodes de segmentation par découpage Ces méthodes reposent sur un découpage récursif de l'image en analysant les espaces plutôt que les traits.

Dans cette famille d'approches, nous distinguons les deux sous-familles suivantes :

Le découpage récursif X-Y [83, 61] consiste à découper récursivement une image binaire, à l'aide de l'analyse des répartitions des pixels représentant la couleur de l'encre, horizontalement et verticalement. À chaque étape de la récursivité, des projections sont appliquées selon les axes X et Y dans le bloc en cours d'analyse. Ces projections atteignent des pics le long des lignes de texte et forment des vallées autour des espaces de séparation entre les blocs de texte. On peut, ainsi, s'arrêter au niveau de segmentation désiré (paragraphe, ligne, mot ou caractère).

La segmentation par l'analyse des espaces [92] est basée sur la détection des grandes zones d'espaces entre les blocs imprimés. Une approche originale [12] repose sur l'analyse des lignes équidistantes de tous les objets de l'image obtenues par la squelettisation des espaces à l'aide d'outils de morphologie mathématique.

Approches mixtes Les méthodes de segmentation par fusion et par découpage ont chacune des avantages et des inconvénients. Si les approches descendantes sont généralement plus rapides, les méthodes de segmentation par regroupement sont plus adaptées à la grande variabilité des documents alors que les méthodes de segmentation par découpage se limitent à des documents contraints ou pour lesquels le modèle est connu. Pour pallier ce problème, quelques auteurs ont développé des approches mixtes par fusion et découpage en essayant de tirer parti des avantages des deux méthodes. Les approches mixtes [59, 110, 107, 73] permettent de ne pas partir de tous les objets élémentaires de l'image à pleine résolution et accélérer les méthodes ascendantes. De plus les approches

mixtes permettent de traiter des documents moins contraints que pour les méthodes purement descendantes.

2.1.2 Séparation texte / fond

Nous nous penchons, dans cette section, sur les approches permettant d'extraire le texte écrit sur un fond non uniforme et texturé, tel que certaines images de publicités. Les lignes de texte contenues dans de telles images sont généralement peu nombreuses et ne présentent aucune régularité. Les approches structurelles sont donc inapplicables dans ce cas de figure.

La binarisation est largement utilisée dans la littérature pour atteindre cet objectif. Une méthode de seuillage du niveau de gris locale et adaptative a été récemment proposée [18]. Cette méthode extrait efficacement le texte des images texturées. Néanmoins, l'information colorimétrique n'étant pas exploitée, de telles approches s'avèrent inefficaces face aux fonds multicolores et faiblement contrastés.

Selon le type de descripteur utilisé pour détecter et extraire le texte des fonds non-uniformes, nous regroupons les approches existantes en deux grandes familles. Les approches locales tirent profit des différences chromatiques et achromatiques entre les éléments de texte et le fond. Ces dernières sont séparées en deux sous-familles : les approches basées sur le regroupement des connexités et les approches basées sur la détection des bords. La seconde famille est composée des méthodes basées sur la texture.

Regroupement de composantes connexes L'extraction de connexités sur un fond coloré et texturé est la phase la plus problématique pour ce type d'approches. L'étape suivante se réduit, en effet, à appliquer des règles géométriques simples aux connexités extraites et éventuellement une série de filtrages pour éliminer les fausses détections.

Sur les images en niveaux de gris, une binarisation locale et adaptative suffit : les connexités noires (ou blanches) alignés et de tailles similaires forment probablement des éléments de texte [88].

Sur les images en couleur, on recourt généralement à une quantification qui donne lieu à plusieurs images binaires (une image par classe colorimétrique). Une extraction suivie d'un regroupement de connexités sont ensuite effectués dans chacun des masques résultant de la quantification.

L'analyse des histogrammes du modèle RVB [52, 126] ou L*a*b* [43] est la technique de quantification la plus communément utilisée dans le but de prétraiter l'extraction de texte. Une méthode de segmentations chromatiques et achromatiques effectuant de simples seuillages des canaux TSV a été proposée [122]. Une telle méthode ne filtre pas les bruits qui entraînent des classes superflues biaisant ainsi la phase d'agglomération de connexités.

D'autres méthodes de classification de couleurs, comme par exemple les graphes théoriques [96], sont utilisées pour regrouper les composantes connexes de la même classe colorimétriques en d'éventuelles entités de texte.

Extraction de bords Une photo incrustée possède des transitions plus douces et plus aléatoires de nuances de gris entre le blanc et le noir, alors qu'une zone de texte est représentée par des transitions brutales des niveaux de gris correspondant aux contours francs des caractères. C'est sur ce constat que s'appuient les méthodes de détection de texte à base de détection de bords [124, 82].

L'image de gradient est calculée en soustrayant les valeurs de gris de chaque paire de pixels consécutifs dans le sens de lecture. Cette image est équivalente à une dérivée de l'image dans une direction donnée. Les valeurs les plus élevées de cette dérivée sont atteintes autour des variations les plus fortes, c'est à dire le long des contours des caractères. En sommant dans le sens de lecture, les valeurs de cette carte de gradients dans une fenêtre de taille donnée, on obtient des valeurs maximales du filtre le long des lignes de texte que l'on peut comparer à un seuil qui règle la sensibilité de la détection. Ce filtre de « gradients cumulé », initialement développé pour la localisation de textes dans les images vidéos [65], a été utilisé pour la localisation des titres dans les vidéos non contraintes comme les archives télévisuelles [119] et la segmentation de l'imprimé composite couleur sans binarisation [66]. Un post-traitement morphologique est appliqué pour isoler les formes allongées entre elles. Cette méthode est cependant sensible à la densité des traits ; ainsi un ou deux très grands caractères isolés ne représentent pas une densité suffisante de variations locales pour être détecté par le filtre. Cette limite est toutefois négligeable puisque les paragraphes de textes composés de moins de 3 caractères sont extrêmement rares.

La plupart des méthodes existantes utilisent le filtre de Canny pour détecter les bords [1, 85], positions potentielles de textes. Les traits détectés sont rassemblés et fusionnés et un ensemble d'heuristiques est appliqué pour éliminer les fausses détections. L'algorithme de détection de bord de Canny est principalement basé sur l'estimation du gradient de l'image après un lissage Gaussien pour réduire le bruit. Cette méthode est conçue pour les images en niveaux de gris ce qui limite son intérêt sur les images en couleurs.

Méthodes utilisant la texture Les éléments textuels (les caractères) présentent des textures similaires entre eux d'une part et différentes de celle du fond d'autre part. La représentation de la texture s'effectue souvent dans le domaine fréquentiel. Les techniques basées sur les filtres de Gabor [96], les ondelettes [48], la transformée en cosinus discrète

(DCT) [71], la transformée de Fourier (FFT) [21], *etc.* sont utilisées pour détecter la texture locale des zones de texte dans une image.

Ces techniques requièrent souvent un apprentissage via des machines à vecteurs de support (SVM), des réseaux de neurones [21], *etc.* pour classer les différentes textures observées et détecter ainsi les zones de textes.

Les méthodes les plus rapides de cette famille présentent une complexité en $\mathcal{O}(n\log(n))$. Elles sont donc spécialement coûteuses en ressources, notamment en termes de temps d'exécution. Cette charge élevée est due à des balayages multiples de l'image. Par ailleurs, l'apprentissage rend ces méthodes dépendantes d'un modèle et non-génériques.

2.1.3 Conclusion

Nous avons présenté dans cette section un aperçu sur les diverses méthodes permettant la détection et l'extraction de texte dans de différents types d'images. Les méthodes structurelles sont adaptées aux images binaires tandis que les approches à base de détection de bords ou de texture permettent d'extraire le texte sur un fond gris. Seules les approches à base de regroupement de connexités gèrent les images couleur aussi bien que les images en noir et blanc. Qui plus est, ces méthodes sont simples et rapides. Cependant, les méthodes de quantification employées par ces approches sont sensibles aux bruits, aux distorsions ou à la résolution de l'image. Cette limite risque de provoquer une sous-segmentation ou une sur-segmentation qui pénalise la phase de regroupement de connexités et ainsi l'extraction de texte.

Nous proposons donc une méthode mixte basée sur le regroupement de connexités dans les plans monochromatiques d'une part et sur la détection des bords ainsi que le regroupement de connexités dans les zones multi-chromatiques ou grises d'autre part.

2.2 Notre proposition

À l'exception des deux classes multi-chromatique et *Gris*, chacune des couches monochromatiques et $N\&B$ issues de notre système de séparation colorimétrique est assimilée à un masque binaire. Comme nous l'avons mentionné dans la section précédente, la méthode d'extraction de texte choisie est tenue de s'adapter à la nature de l'image et son contenu.

Nous présentons, dans cette section, une approche adaptative qui prend en compte les particularités des zones traitées. Cette approche repose sur le groupement de connexités dans les images binaires et sur une adaptation de la technique des gradients cumulés aux images RVB.

2.2.1 Extraction de texte dans les classes monochromatiques

Nous nous intéressons dans cette section à la classe $N\&B$ ainsi que toutes les régions monochromatiques. Nous associons à chacune de ces classes une image binaire. Les approches structurelles y sont parfaitement applicables.

L'agglomération de connexité s'avère être une approche efficace et rapide. Cette méthode permet donc d'atteindre nos objectifs tout en satisfaisant nos contraintes industrielles.

Algorithme Dans nos images de presse actuelle, trois contraintes régissent l'agglomération des composantes connexes en lignes de texte :

- l'alignement horizontal : les connexités d'une même ligne doivent présenter un recouvrement vertical suffisant (au moins le quart de chaque connexité doit être en recouvrement)
- des hauteurs similaires : deux connexités sont colinéaires si aucune des deux n'est plus grande d'un ratio de 2 que l'autre.
- un espacement horizontal suffisamment faible : l'espacement entre chaque paire de connexités consécutives doit être inférieur au double de la hauteur moyenne de la ligne correspondante.

Cette approche permet de détecter les lignes horizontales ou légèrement inclinées. Cette limite est acceptable dans la mesure où ces lignes non détectées sont extrêmement rares dans les documents de notre corpus d'une part et où l'algorithme est très rapide d'autre part. De plus, les OCR ne reconnaissent que les lignes horizontales. Cet algorithme reste toutefois modulable pour détecter les lignes verticales, en agglomérant les connexités dans la direction orthogonale, ou les lignes obliques si l'angle d'inclinaison est connu *a priori*.

Dans certaines images, notamment les pages de presse, les lignes de texte sont organisées en colonnes. Si les colonnes sont horizontalement proches, notre algorithme est susceptible d'associer des connexités appartenant à des colonnes différentes à une même ligne. Toutefois, nous nous contentons, à ce stade d'une simple extraction de texte sous la forme de lignes "physiques" puisque nous n'avons pas besoin d'extraire les lignes "logiques".

2.2.2 Extraction de texte dans les couches multi-chromatiques et *Gris*

La technique des gradients cumulés a prouvé son efficacité sur des images et des vidéos en niveaux de gris. Le recours aux gradients couleurs [64] permet d'adapter cette technique aux zones en couleurs sans besoin de les transformer en niveaux de gris. Ce nouveau gradient permet donc de tirer profit de l'information apportée par les variations

colorimétriques et facilite ainsi l'extraction de texte dans les régions les plus faiblement contrastées.

Nous proposons l'algorithme ci-dessous pour extraire le texte dans les zones multi-chromatiques.

Algorithme Les lignes de textes étant quasiment toujours horizontales, nos calculs seront basés sur la dérivée horizontale [1].

Soit $p(x, y)$ un point de l'image défini dans l'espace RVB par (R_p, V_p, B_p). La dérivée horizontale d_x est définie en p par :

$$d_x(p) = \mathcal{M} \ / \quad |\mathcal{M}| = max\{\left|\frac{\partial R_p}{\partial x}\right|, \left|\frac{\partial V_p}{\partial x}\right|, \left|\frac{\partial B_p}{\partial x}\right|\} \tag{II.1}$$

Les valeurs de d_x sont cumulées dans des fenêtres horizontales de taille $(k \cdot \mathcal{S}_t \times 1)$; k est un paramètre dont nous avons fixé la valeur à 45. Cette largeur de la fenêtre correspond à la largeur moyenne de trois caractères. Il est à noter que l'algorithme est robuste aux variations de k. En effet, nous avons effectué des tests avec k variant de 15 (largeur d'un caractère) à 300 et nous avons constaté que les résultats ne varient pas de façon significative.

L'image de gradients cumulés (Fig. II.1.b) est ensuite seuillée en appliquant une binarisation rudimentaire, comme par exemple la méthode Otsu [89]. Les zones émergentes sont quantifiées (Figure II.1.c) par le biais d'une méthode simple et rapide : une classification qui s'apparente à la méthode MeanShift [34] appliquée dans le cube RVB réduit. Selon cette approche, les classes retenues sont données par les centres des agglomérations de fortes densités. Notons que le recours à d'autres méthodes de quantification est également possible.

La quantification sert à séparer le texte du fond, localement dans l'image. Dans l'exemple de la figure II.1, la quantification donne lieu à trois classes colorimétriques, c'est-à-dire à trois images binaires où nous pouvons appliquer des approches structurelles pour localiser le texte. L'algorithme détaillé dans la section 2.2.1 est donc appliqué à chacune des couches résultantes afin d'y extraire les éventuels éléments de texte.

En remplaçant le gradient couleur par le gradient niveau de gris classique, nous calculons l'image de gradient cumulé dans la couche $Gris$. Dans cette couche, la quantification est évidemment remplacée par une simple binarisation.

1. Il est possible de remplacer la dérivée horizontale par la dérivée verticale si la direction de l'écriture l'exige.

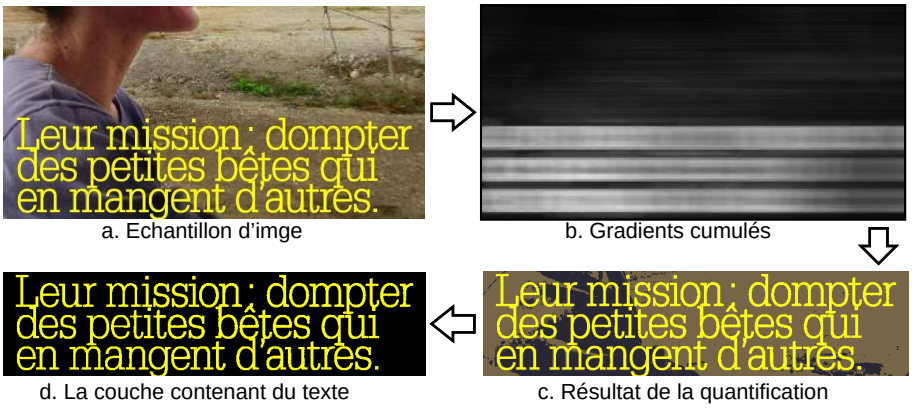


FIGURE II.1 – Extraction de texte d'une photo

2.2.3 Conclusion

Nous avons présenté dans cette section une approche simple et générique permettant d'extraire le texte dans des images quelconques. Cette méthode est simple grâce à la segmentation colorimétrique (présentée dans le chapitre I) qui lui fournit des classes homogènes et libérées de tout bruit colorimétrique.

La méthode proposée s'accommode localement à la nature de la zone : nous appliquons une approche structurelle dans les régions uniformes tandis que les zones polytonales sont traitées par des méthodes plus sophistiquées comme le gradient cumulé. C'est cette mixité qui assure la généricité de l'approche.

2.3 Évaluation

Nous allons mesurer l'impact de notre méthode d'extraction de texte sur les performances de l'OCR Abbyy FineReader 8.1 en termes de segmentation en lignes. Le taux de reconnaissance de l'OCR n'entre pas dans notre cadre d'étude.

2.3.1 Choix et motivations

Pourquoi utiliser FineReader ? Comme nous l'avons mentionné précédemment, aucune méthode de segmentation structurelle n'est applicable à tout type de document. En revanche, les OCR passent nécessairement par une phase de segmentation en mots, lignes, paragraphes, *etc.* avant de reconnaitre le texte, et ce indépendamment de la nature de l'image. Ces logiciels nous procurent donc une segmentation physique générique, objective et facilement reproductible en vue de comparer les résultats.

Nous avons choisi d'évaluer la segmentation en lignes, non pas en caractères, en mots ou en paragraphe. Ce choix constitue un compromis entre la complexité du protocole expérimental (l'étiquetage manuel de la base d'évaluation) et la précision des résultats.

FineReader s'avère être l'OCR le plus performant sur le marché actuel. Améliorer les performances de ce dernier logiciel constitue donc un challenge de poids que nous avons tenté de relever.

Choix de la base de test FineReader est parfaitement efficace sur les images de documents réguliers comme par exemple les romans, les revues scientifiques, les documents techniques, *etc.* L'application de notre méthode à ces images serait donc superflue. En revanche, cet OCR peine à extraire le texte des photos et fonds multicolores. Un traitement préalable de ces zones permettrait donc au logiciel d'atteindre de meilleures performances. C'est pour cette raison que la littérature comprend une variété de méthodes assurant ce prétraitement, à savoir l'extraction de texte des zones picturales.

Les magazines de modes contiennent des blocs de texte réguliers aussi bien que des pages de publicités de structures variables et des photos naturelles incorporant du texte. De par leur mixité et leurs structures complexes, ces images s'avèrent être le meilleur moyen pour évaluer la généricité de notre approche.

2.3.2 Protocole expérimental

Nous obtenons une image de texte globale en additionnant les résultats d'agglomération de connexités provenant des différentes couches chromatiques et achromatiques. L'image résultante est de fond blanc uniforme [2]. Nous comparons ensuite les performances de l'OCR sur, respectivement, ces dernières images et les images de documents brutes.

Nous avons manuellement encadré toutes les lignes de texte des numéros '3321' du magazine 'Elle' ainsi que '73 02 10' de 'Glamour'. Notre base de test est donc composée de 448 pages contenant 14303 lignes de texte globalement. Ces lignes proviennent de blocs de texte réguliers ainsi que de zones graphiques.

Les résultats de la segmentation en lignes seront présentés en termes de précision P et rappel R ; R étant le nombre de lignes de la vérité terrain correctement détectées (par l'OCR) divisé par le nombre de lignes dans la vérité terrain ; P est donné par le rapport entre le nombre de lignes correctement détectées et le nombre de lignes détectées au total.

$$R = \frac{nb\ lignes\ correctement\ détectées}{nb\ lignes\ dans\ la\ vérité\ terrain} \quad P = \frac{nb\ lignes\ correctement\ détectées}{nb\ total\ de\ lignes\ détectées} \tag{II.2}$$

Notre protocole expérimental suit la méthode d'évaluation Deteval [118]. Cette mesure stipule que :

- une ligne de la vérité terrain est considérée correctement détectée s'il existe une ligne segmentée dont le rectangle englobant recouvre, au minimum, 80% de sa surface.

2. Les connexités blanches sont inversées afin d'être décelable du fond blanc.

- une ligne segmentée est considérée comme fausse détection si au moins 40% de la surface couverte par son rectangle englobant ne recouvre aucun rectangle dans la vérité terrain.

2.3.3 Résultats et commentaires

Le tableau II.1 affiche les résultats de segmentation en lignes par FineReader sur les images brutes et respectivement nos images de texte recomposées.

Type d'image en entrée	R	P
FineReader appliqué aux images d'origines	81.63	93.70
FineReader appliqué à nos images de texte	91.03	90.89

TABLE II.1 – Résultats de segmentation sur une base de 14303 lignes

Analyse des valeurs de rappel Le rappel reflète la capacité de la méthode à retrouver les lignes de textes ; les fausses détections n'influent donc pas sur R. L'amélioration apportée par notre méthode vis-à-vis de cette mesure est principalement due à l'extraction du texte des zones graphique (exemple de résultats dans la figure II.2).

Nous avons constaté que certains éléments textuels que notre méthode a correctement extraites ne sont pas pour autant correctement segmentées par l'OCR. Ces éléments sont, en effet, imprimés avec des polices non gérées par FineReader. Ce logiciel n'est donc pas suffisamment générique pour segmenter tous les éléments textuels. Cette défaillance affecte négativement les valeurs de rappel qui atteindraient 96% sur nos images de texte si toutes les polices étaient correctement gérées.

Par ailleurs, FineReader ne détecte également pas les titres écrits en gros caractères. Cette limite s'ajoute à la précédente pour baisser les valeurs de rappel. Il est toutefois envisageable de redimensionner les grosses connexités de façon à ce qu'elles aient la taille suffisamment petite pour être reconnues par le logiciel. Ce redimensionnement permettrait conséquemment d'améliorer le rappel.

Analyse des valeurs de précision De façon complémentaire au rappel, la précision reflète le taux d'erreurs (de fausses détections) engendré par le système de segmentation. Le tableau II.1 témoigne une certaine baisse des valeurs de précision sur nos images de texte par rapport aux images brutes. Cette baisse est due à deux principales raisons :

- Les fausses détections sous forme de textures régulières et de couleurs uniformes dans les zones picturales (exemple : des fenêtres dans l'image d'un immeuble). Ces fausses détections incombent à notre système d'extraction de texte qui est dépourvue d'une

a. b.

FIGURE II.2 – a. Exemples de résultats par FineReader, b. Légende

phase de filtrage. Le recours à certaines heuristiques géométriques permettrait de les filtrer et d'améliorer ainsi la précision de façon significative.

- L'OCR associe parfois des entités textuelles provenant de paragraphes différents, voire de zones de natures différentes, à une même ligne. Cette sous-segmentation pénalise la précision considérablement.

Comparaisons Il nous est difficile de nous comparer à l'état de l'art car les approches existantes ciblent des images de natures différentes de notre corpus.

Les approches structurelles ne permettent pas d'extraire le texte dans les zones picturales ; elles n'apportent donc aucune améliorations aux performances de FineReader.

En revanche, les approches non structurelles sont toutes conçues pour extraire le texte exclusivement des zones graphiques. Ces méthodes sont donc inapplicables aux images de magazines qui contiennent naturellement des blocs de texte réguliers de fond uniforme. Les images traitées par ces méthodes sont notamment les couvertures de magazines et les vidéos. Les images de tests renferment donc un nombre réduit de lignes de texte, ce qui rend la comparaison de résultats encore moins objective.

Par ailleurs, ces dernières pages présentent un aspect irrégulier, si bien qu'il est difficile d'y regrouper les mots en lignes de texte. De ce fait, les résultats de recherche dans ce cadre sont souvent exprimés en termes de taux d'extraction de mots, jamais en taux de segmentation en lignes.

À titre d'exemple, une méthode d'extraction de texte récente [18] atteint $R = 99.2$ et $P = 99.4$ (sans passer par aucun OCR) sur une base de test composée de 65 images contenant peu de mots. Cette méthode présente l'inconvénient de dépendre de plusieurs paramètres tout comme la plupart des méthodes existantes. De surcroît, l'information colorimétrique n'est pas mise à profit par cette approche.

Notons, par ailleurs, que les travaux de recherches liés à ce domaine [108, 43] affichent souvent leurs résultats en termes d'images illustratives et ne procurent pas de valeurs numériques.

2.4 Conclusion

Nous avons présenté dans cette section une méthode d'extraction de texte qui a la particularité et l'avantage d'être applicable à toute image de document, quelque soit sa nature. Cette méthode s'appuie fortement sur nos résultats de segmentation chromatique et achromatique ainsi que sur la technique des gradients cumulés que nous avons adaptée afin d'utiliser l'information colorimétrique.

Nous avons ensuite mesuré l'apport de notre approche aux performances de l'OCR FineReader en matière de segmentation en lignes de texte. Nous avons donc pu relever une amélioration de 10% des valeurs de rappel pour une perte de 3% de la précision. Une phase de filtrage subséquante à l'étape d'extraction de texte permettrait d'améliorer la précision de façon significative.

À l'instar de l'information chromatique ou achromatique, cette information textuelle acquise alimentera un classificateur de blocs de presse qui permet de détecter les zones publicitaires.

3 Détection des publicités

Nous nous intéressons dans cette section au problème de détection des zones publicitaires dans les images de presse numérisée.

Pour ce faire, nous effectuons un classement de blocs homogènes découpés dans les pages de journaux et magazines. Ces blocs sont issus d'une segmentation physique s'appuyant sur les résultats d'extraction de texte (section 2).

Une étude de l'état de l'art (présentée dans la section 3.1) montre que les descripteurs

image, colorimétriques et structurels, sont les mieux adaptés pour identifier les blocs publicitaires.

Nous avons testé deux classificateurs, k-NN et AdaBoost [33], pour regrouper les blocs en respectivement quatre et deux catégories. Les résultats seront rapportés dans la section 3.3.

3.1 Étude de l'existant

La détection de publicités dans les images de presse numérisée constitue une nouvelle thématique de recherche. En effet, quelques travaux sur le classement d'images de documents ou sur la détection des zones publicitaires dans des bases de données très différentes de notre corpus existent dans la littérature mais aucun travail existant ne permet d'aboutir à la détection de publicités dans des images de périodiques et magazines.

Dans cette section, nous allons passer en revue les méthodes les plus adaptables à notre application et dont nous pouvons nous inspirer. Les approches non applicables dans notre cadre d'utilisation seront présentées de façon sommaire.

3.1.1 Classement d'images de document

Les méthodes de classement de documents répondent à des besoins divers et variés comme par exemple l'indexation, la préparation à des traitement d'images spécialisés, la détection d'articles non réguliers (comme les blocs publicitaires), *etc.*

Les descripteurs et la méthode d'apprentissage dépendent du corpus, des classes d'images à distinguer et du problème à traiter [16]. À titre d'exemple, le tri automatique des courriers passe par une binarisation suivie d'un classement basé sur des descripteurs sémantiques [3]. En revanche, la détection d'images publicitaires dans les pages web est principalement basée sur l'information colorimétrique et structurelle [69, 39].

Le classement de blocs de documents repose souvent sur des caractéristiques simples et directement calculées sur l'image, tel que la densité de pixels noirs, ou extraites à partir des résultats de segmentation physique comme par exemple le nombre de lignes de texte, les dimensions du bloc, *etc.* [16].

Les caractéristiques sémantiques sont également utilisées et sont généralement déduites des résultats de l'OCR.

La texture et entre autres les bords sont des informations souvent utilisées, dans différents types de médias [115], pour caractériser les échantillons à classer. L'extraction de telles caractéristiques est souvent coûteuse en terme de temps d'exécution. Par ailleurs, ces dernières ne sont pas assez discriminantes pour les images de presse actuelle.

3.1.2 Détection de zones publicitaires

Certains travaux concernant la détection ou la reconnaissance de zones publicitaires dans les images Web et les vidéos existent dans la littérature. Cependant, l'état de l'art ne révèle aucune approche permettant la localisation de ces espaces commerciaux dans les images de presse numérisée qui présentent des caractéristiques différentes des images web et des vidéos en différents points :

- la détection de publicités dans les images de presse revient à un classement d'articles de journaux. Une phase de segmentation en blocs est donc nécessaire au préalable. Les résultats de classement sont sensiblement affectés par la finesse de la segmentation physique. Les images Web et les vidéos sont exemptées de cette segmentation : dans ces corpus, ce sont les images entières qui font l'objet du classement.
- Les descripteurs sémantiques facilement obtenus à partir du code HTML sont souvent employés [69, 39, 98] pour classer les images Web et détecter ainsi les publicités. Or, nos blocs de presse ne nous procurent aucune information sémantique.
- L'information spatio-temporelle est souvent utilisée pour détecter les publicités dans les vidéos [99, 125, 116]. Des descripteurs audio sont également employés pour extraire les passages publicitaires dans ce type de médias. Le format image ne nous procure évidemment pas ces informations spécifiques au format vidéo.

Les publicités présentent toutefois certaines caractéristiques partagées par la plupart des formats médiatiques. Il s'agit de l'information colorimétrique et structurelle mentionnées dans la section 3.1.1.

Le recours à l'information sémantique engage l'acquisition d'un dictionnaire permettant de rechercher les termes révélant la présence d'une éventuelle zones publicitaire. Or, une telle base de données restreindrait la détection à certaines publicités bien ciblées sur lesquelles nous disposons d'une connaissance *a priori* comme par exemple le nom de l'enseigne, certains termes commerciaux fréquemment employés, *etc.* À défaut de cette information, il est impossible de tirer profit des descripteurs sémantiques pour détecter les publicités.

Si les descripteurs varient énormément afin de s'adapter aux différents types de médias à traiter, certaines méthodes d'apprentissage sont suffisamment matures pour classer des objets de natures différentes. Les classificateurs les plus communément utilisés dans le cadre de détection de publicités sont SVM [23], AdaBoost [33] et les réseaux de neurones [49].

3.1.3 Conclusion

L'étude de l'état de l'art nous révèle que les descripteurs colorimétriques et structurels sont les plus appropriés pour détecter les publicité dans notre cadre d'utilisation. Les premières caractéristiques sont directement calculées sur l'image ou à l'issue d'une séparation de couleurs. Les descripteurs structurels sont simplement déduits des résultats d'une segmentation physique. Le recours à d'éventuelles caractéristiques sémantiques restreindrait inéluctablement le champ d'application du système de détection de publicités.

3.2 Notre proposition

Nous présentons dans cette section une approche inédite permettant de classer des images de documents dans le but de détecter les zones publicitaires.

Une publicité couvre rarement la totalité d'une page : une image de presse est généralement composée d'un ensemble d'articles, d'illustrations graphiques et éventuellement de zones publicitaires de formats variables (graphique, texte, *etc.*). Une phase de segmentation en blocs homogènes s'avère donc nécessaire en amont du classement.

Après une sélection minutieuse des descripteurs de classement, nous présenterons les résultats obtenus ainsi que les commentaires associés.

3.2.1 Segmentation en blocs et pré-classement

La segmentation physique de la page n'étant pas notre objectif final, nous en proposons une approche simple et ultra-rapide basée sur le regroupement d'entités homogènes.

Cette méthode donne lieu à trois types de blocs :

- Les blocs de type *texte* sont exclusivement composés de lignes de texte. Les lignes de texte (section 2) présentant un recouvrement horizontal suffisant (au moins la moitié de la largeur de chaque ligne), qui sont suffisamment proches verticalement (la distance entre deux lignes consécutives doit être, au plus, deux fois plus grande que le hauteur de chaque ligne) et de hauteurs similaires (le rapport de hauteurs maximum toléré entre chaque paire de lignes du bloc est de 2) forment un bloc de texte.
- Les blocs de type *graphique* ne contiennent aucun élément textuel. Ces blocs sont donnés par les rectangles englobants ne contenant pas de texte dans les classes multi-chromatiques et *Gris* ainsi que toutes les entités non textuelles des classes complémentaires. Si les rectangles englobants de deux blocs *graphiques* entrent en intersection, ces derniers sont naturellement fusionnés.
- Si un élément textuel se trouve à l'intérieur d'un bloc *graphique*, ce dernier est considéré de type *graphique&texte*

La figure II.3 présente un exemple des résultats de segmentation sur un échantillon d'image : chaque bloc identifié est colorié en une teinte aléatoire.

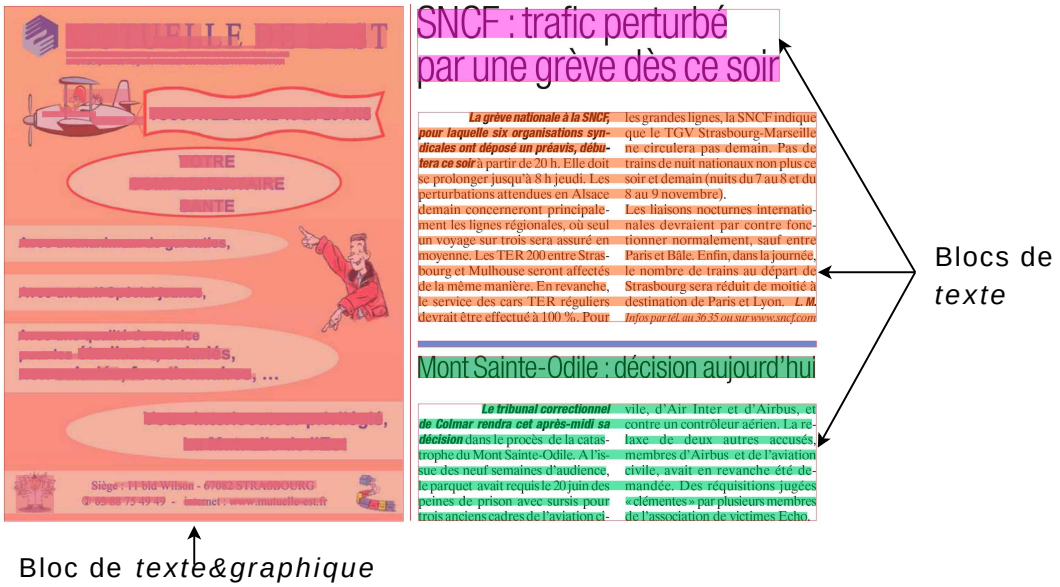


FIGURE II.3 – Résultat de segmentation en blocs

Cette méthode engendre parfois la sur-segmentation de certains articles composés de plusieurs colonnes. Les titres sont également souvent considérés comme des blocs à part entière (figure II.3). Néanmoins, cette insuffisance reste acceptable puisque la méthode est complètement automatisée et ne requiert aucun modèle ni apprentissage. Par ailleurs, les blocs de texte correspondent rarement à des publicités : ce sont les candidats de type *graphique&texte* qui sont les plus susceptibles de l'être.

3.2.2 Sélection des descripteurs

Afin d'évaluer nos méthode de segmentation colorimétrique et structurelle de façon objective, tous les descripteurs guidant le classement sont directement déduits des résultats de ces approches de décomposition, à l'exclusion de toute caractéristique sémantique.

L'observation de publicités provenant de différents magazines et journaux permet de constater que les zones publicitaires présentent souvent les caractéristiques suivantes :

- des éléments textuels ou / et un fond colorés,
- présence de plusieurs teintes ou de régions multi-chromatiques,
- un alignement irrégulier (lignes de texte de largeur ou / et de hauteurs variables),
- peu de texte,
- le bloc publicitaire n'est pas noyé dans un article.

Il est malheureusement peu probable d'observer toutes ces caractéristiques conjointement dans un même bloc.

Ces observations sont traduites en l'ensemble de descripteurs numériques ci-dessous :

- taux respectifs de pixels dans les parties noires, grises, monochromatiques, et multi-chromatiques dans le bloc,
- nombre de teintes différentes,
- pourcentages et densités respectifs de lignes de texte noir, blanc, gris, monochromatique et multi-chromatique [3],
- variances respectives des hauteurs et largeurs des lignes de texte,
- taux de surface d'intersection avec les autres blocs de la page.

Ces derniers descripteurs donnent lieu à un vecteur de caractéristiques de dimension 21.

3.2.3 Classificateurs

Nous avons sélectionné deux classificateurs pour catégoriser les blocs candidats et détecter ainsi les publicités.

K-NN L'algorithme des k-plus proches voisins (k-NN) est une méthode de classement fondamentale. Pour classer un candidat donné, il suffit de calculer sa distance par rapport à chacun des échantillon de la base de connaissance. Les k plus proches voisins correspondent aux k échantillons dont les distances sont les plus faibles. Le candidat est assigné à la classe la plus représentée par les k échantillons. $K \in [1..N[$; N étant le nombre d'échantillons dans la base de connaissance.

Cette méthode présente l'avantage d'être simple et générique : elle est applicable à n'importe quelle base de données, qu'elle soit équilibrée ou pas. Par ailleurs la modification (l'ajout ou la suppression d'échantillons) de la base étiquetée n'affecte en rien le procédé. Cependant, il s'agit d'une approche lente ; sa complexité étant en $\mathcal{O}(N)$.

AdaBoost AdaBoost [33] est une méthode d'apprentissage automatique reposant sur la sélection itérative de classificateurs faibles en fonction d'une distribution des exemples d'apprentissage. Chaque exemple est pondéré en fonction de sa difficulté avec le classificateur courant.

Algorithme :

- Soit une base d'apprentissage : $(x_1, y_1), ..., (x_m, y_m)$ où $x_i \in X$ sont les échantillons et $y_i \in Y = \{-1, 1\}$ les étiquettes.

3. Une ligne de texte est dite multi-chromatique si elle est incluse dans une zone multi-chromatique

- Initialement, on associe à chaque échantillon i de la base d'apprentissage une valeur $D_1(i) = \frac{1}{m}, i = 1, .., m$.
- T étant le nombre d'itérations, pour $t = 1, ..., T$:
  1. Trouver dans H, famille des classificateurs faibles [4], le classificateur $h_t : X \rightarrow \{-1, 1\}$ qui minimise l'erreur de classement ϵ_t : $h_t = \text{argmin}_{h \in H \epsilon_t}$ avec $\epsilon_t = \min_{h \in H} \sum_{i=1}^{m} D_t(i)[y_i \neq h(x_i)]$
  2. Si $\epsilon_t < 0.5$ le classificateur est sélectionné, sinon l'algorithme s'arrête
  3. On détermine alors le poids du classificateur : $\alpha_t \in \mathbb{R}$, avec $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$
  4. $D_{t+1}(i) = \frac{D_t(i) \mathrm{e}^{-\alpha_t y_i h_t(x_i)}}{Z_t}$, Z_t étant un facteur de normalisation.
- Le classificateur global résultant de ce processus est : $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$
- Pour classer un candidat x_c, il suffit de calculer $H(x_c)$; si la valeur renvoyé est positive, il est assignée à la classe dont l'étiquette est 1 ; sinon le candidat est affecté à la deuxième classe.

L'algorithme décrit ci-dessus renvoie une valeur binaire, c'est-à-dire que seulement deux classes peuvent être séparées. D'autres variantes adaptées à un plus grand nombre de classes existent toutefois dans la littérature. Cependant, ces dernières approches sont particulièrement couteuses en terme de temps d'exécution.

Le classificateur AdaBoost est capable d'évaluer le pouvoir discriminatoire de chaque descripteur indépendamment des autres et de faire valoir ainsi les caractéristiques les plus performantes dans chaque base en leur associant les poids les plus élevés.

Par ailleurs, un échantillon x est classé plus rapidement par AdaBoost que par k-NN puisqu'il suffit, pour ce faire, d'évaluer $H(x)$.

Cette méthode requiert, cependant, l'équilibrage de la base d'apprentissage (les échantillons positifs doivent être présents dans la base de façon similaire aux échantillons négatifs). Or, notre cadre d'utilisation, cette contrainte ne reflète pas les conditions réelles : les publicités sont beaucoup moins présentes que les articles réguliers dans notre corpus.

3.2.4 Conclusion

Nous avons défini dans cette section tous les éléments nécessaires à un apprentissage supervisé, à savoir : les candidats potentiels au classement, les descripteurs numériques et les classificateurs.

Les blocs candidats résultent de la segmentation physique de la page. La méthode employée à cet effet est une application directe aux résultats de la décomposition colo-

4. Un classificateur faible est défini par un axe a ($a \in [1..21]$ dans notre cas de figure)de l'espace des descripteurs, un seuil fixe s et un sens ($<$ ou $>$). Il s'agit d'une fonction binaire renvoyant la valeur 1 si la projection de l'échantillon selon a est inférieure (ou supérieure, selon le sens du classificateur) à s et -1 sinon.

rimétrique et de l'extraction de texte. Les caractéristiques de classement sont également calculées à partir de ces derniers résultats.

Nous avons choisi deux classificateurs différents pour évaluer ces descripteurs. Les résultats respectifs de chaque méthode seront reportés et commentés dans la prochaine section.

3.3 Résultats et commentaires

3.3.1 Protocole expérimental

Base de données Notre base de données est issue d'environ 400 pages simples et doubles provenant de journaux et magazines européens variés. Il s'agit donc d'une base multilingue composée d'images de qualités et résolutions variables procurées par notre partenaire commercial.

Ces images proviennent, plus précisément, de 5 documents choisis pour leurs aspects différents et complémentaires. Nous estimons que ces images sont donc assez représentatives de notre corpus :

- les documents numérotés 1 et 2 présentent des magazines politiques contenant des zones colorés aussi bien à l'intérieur des blocs publicitaires que des blocs de type *graphique&texte* ou *graphique*,
- documents 3 et 4 sont des journaux contenant peu de zones colorées et un nombre considérable de publicités imprimées en niveaux de gris,
- document 5 est un quotidien dont la charte graphique est particulièrement colorée : même les blocs de texte régulier sont souvent écrits sur un fond chromatique.

La segmentation physique de ces pages donne lieu à un grand nombre de blocs. Or, il est parfois immédiat de déterminer automatiquement le type de certains blocs comme les filets (section 7.3.2) et les titres d'articles ; il est improbable que ces derniers correspondent à des blocs publicitaires. Le nombre effectif de candidats au classement tombe donc à 3458. Le tableau II.2 décrit la répartition des données entre les documents de notre base.

Document 1	Document 2	Document 3	Document 3	Document 5
9%	8%	42%	30%	11%

Table II.2 – Répartition des données entre les cinq documents

Nous distinguons 4 types de blocs. L'étiquetage manuel de la base complète révèle que cette dernière comprend :

- 8% de *publictités*,
- 43% de blocs de type *texte*,

- 39% de blocs de type *graphique*,
- et 10% de blocs de type *graphique&texte*.

Apprentissage Nous répartissons les 3458 blocs entre la base d'apprentissage et la base de test selon un quota variable : dans les différentes expérimentations, la base d'apprentissage peut compter de 30% jusqu'à 70% du total des échantillons.

Une base d'apprentissage plus conséquente (resp. plus réduite) induirait une base de test trop réduite (resp. trop volumineuse) pour conduire des expérimentations statistiquement solides.

En fixant le quota de répartition entre les bases d'apprentissage et de test, nous effectuons une validation par sous-échantillonnage aléatoire répété 100 fois (100-Repeated random sub-sampling validation) : il s'agit de choisir aléatoirement les échantillons d'apprentissages et de test et ce 100 fois de suite ; nous considérons ensuite les valeurs moyennes des résultats de ces 100 expérimentations.

Évaluation Les résultats relatifs à l'algorithme des k-plus proches voisins seront présentés en terme de matrices de confusion. Une ligne de cette matrice représente les résultats associés à une classe donnée :

- la première ligne est associée à la classe *texte*,
- la seconde à la classe *graphique&texte*,
- la $3^{ème}$ ligne est associée à la classe *publicité*,
- et la $4^{ème}$ à la classe *graphique*.

À titre d'exemple, la troisième ligne de la première matrice (en haut à gauche) dans le tableau II.4 signifie que 1.11% des publicités sont assignées à la classe *texte*, 28.11% à la classe *graphique&texte*, 55.22% sont correctement classées et 15.56% sont affectées à la classe *graphique*.

Il est à noter que dans une matrice de confusion $M = (m_{i,j})_{1 \leq i,j \leq 4}$:

- chacune des valeurs $m_{i,i}$ de la diagonale correspond au rappel d'une classe i (le rappel relatif à la détection de publicité est donné par $m_{3,3}$),
- la précision relative à une classe i est donnée par $\frac{m_{i,i}}{\sum_{j=1}^{4} m_{i,j}}$.

Les résultats relatifs à AdaBoost seront présentés sous une forme différente puisque cette méthode est de réponse binaire : il est seulement possible de décider si un échantillon donné appartient à la classe *publicité* ou pas. En l'occurrence, nous présenterons les résultats de cette approche en terme de précision P et rappel R. Soit $N_{effectf}$ le nombre effectif de publicités dans la base de test, $N_{correct}$ le nombre de publicités correctement

classées et N_{pub} le nombre de blocs assignés à la classe *publicité* dans cette base.

$$P = N_{correct}/N_{pub} \quad R = N_{correct}/N_{effectf} \tag{II.3}$$

Par ailleurs, la méthode AdaBoost requiert l'équilibrage de la base d'apprentissage pour chaque expérimentation. Cet ensemble doit donc contenir 50% de publicité et 50% d'échantillons négatifs (16,6% de blocs *texte*, 16,6% de *texte&graphique* et 16,6% de *graphique*). Les blocs restants figurent naturellement dans la base de test.

3.3.2 Classificateur k-NN

Le tableau II.3 affiche les valeurs moyennes de précision et rappel atteintes par 3-NN sur une base répartie selon un quota de 50% entre l'ensemble de connaissances et celui de test (chacune des deux bases compte 50% des publicités). Les valeurs de la première ligne de ce tableau sont obtenues en équilibrant la base de connaissance (cette base est composée de 25% de chacune des quatre classes). La seconde ligne présente les valeurs atteintes en utilisant une base inéquilibrée : le quota de répartition de 50% est appliqué à chaque classe indépendamment des autres.

Les valeurs P/R moyennes sont calculées en prenant en compte le nombre total de blocs dans chaque document.

		Texte	*Texte&graphique*	***Publicité***	*Graphique*	**Global**
Équilibrée	*R*	95.15	59.05	**69.86**	87.27	**86.44**
	P	97.74	51.68	**54.57**	93.56	**88.05**
Non-équilibrée	*R*	98,38	58.65	**61.76**	93.82	**89.70**
	P	97.09	63.63	**65.35**	92.47	**89.40**

TABLE II.3 – Résultats P/R globaux obtenus par 3-NN sur une base de connaissance équilibrée *vs* non-équilibrée et de quota 50%

Nous pouvons constater à partir de ce tableau que l'équilibrage de la base de connaissances permet d'atteindre un meilleur rappel au détriment de la précision vis-à-vis de la détection des publicités. Ainsi, l'utilisateur peut choisir le mode de répartition des classes dans cette base en fonction de l'application ciblée. Par exemple, le rappel a une plus grande importance dans le cadre d'une utilisation visant le filtrage des publicités.

Nous présentons les résultats 3-NN détaillés sur des bases de connaissance équilibrées dans le tableau II.4, les quotas respectifs attribués à ces bases étant de 70%, 50% et 30% (70% des blocs dans la base de connaissance et 30% dans la base de test, puis 50% des blocs dans chacune des deux bases, puis 70% dans l'ensemble d'apprentissage et 30% dans celui de test).

Quota attribué à la base de connaissance	70%	50%	30%
document 1	$\begin{pmatrix} 93.68 & 2.41 & 1.54 & 2.37 \\ 7.74 & 48.68 & 34.24 & 9.34 \\ 1.11 & 28.11 & 55.22 & 15.56 \\ 1.32 & 8.84 & 5.18 & 84.66 \end{pmatrix}$	$\begin{pmatrix} 92.51 & 2.79 & 1.29 & 3.40 \\ 7.14 & 49.12 & 34.74 & 9.00 \\ 0.79 & 31.79 & 53.07 & 14.36 \\ 1.13 & 6.92 & 5.00 & 86.95 \end{pmatrix}$	$\begin{pmatrix} 84.71 & 9.21 & 2.52 & 3.55 \\ 7.10 & 46.06 & 38.31 & 8.52 \\ 1.53 & 34.11 & 51.37 & 13.00 \\ 1.09 & 10.78 & 5.49 & 82.65 \end{pmatrix}$
document 2	$\begin{pmatrix} 87.22 & 6.88 & 2.61 & 3.29 \\ 4.79 & 45.71 & 23.29 & 26.21 \\ 0.71 & 23.29 & 60.00 & 16.00 \\ 0.52 & 5.80 & 1.30 & 92.37 \end{pmatrix}$	$\begin{pmatrix} 86.11 & 6.82 & 2.78 & 4.29 \\ 6.21 & 47.84 & 21.53 & 24.42 \\ 1.00 & 30.08 & 58.67 & 10.25 \\ 0.48 & 7.05 & 1.47 & 91.00 \end{pmatrix}$	$\begin{pmatrix} 81.27 & 7.91 & 6.78 & 4.04 \\ 7.75 & 44.71 & 28.63 & 18.92 \\ 3.06 & 31.82 & 56.41 & 8.71 \\ 2.61 & 13.65 & 2.07 & 81.66 \end{pmatrix}$
document 3	$\begin{pmatrix} 99.34 & 0.16 & 0.36 & 0.15 \\ 2.38 & 60.62 & 25.33 & 11.67 \\ 0.39 & 15.00 & 74.56 & 10.06 \\ 0.39 & 7.81 & 4.07 & 87.73 \end{pmatrix}$	$\begin{pmatrix} 98.24 & 0.93 & 0.59 & 0.23 \\ 2.58 & 60.32 & 27.01 & 10.08 \\ 0.41 & 16.07 & 74.72 & 8.79 \\ 0.45 & 9.30 & 4.32 & 85.92 \end{pmatrix}$	$\begin{pmatrix} 96.79 & 1.17 & 1.65 & 0.39 \\ 2.94 & 59.99 & 28.69 & 8.39 \\ 0.17 & 18.78 & 73.80 & 7.24 \\ 0.49 & 14.22 & 3.87 & 81.42 \end{pmatrix}$
document 4	$\begin{pmatrix} 97.48 & 0.77 & 1.56 & 0.19 \\ 1.07 & 71.73 & 18.27 & 8.93 \\ 3.05 & 11.35 & 75.45 & 10.15 \\ 0.32 & 5.40 & 5.48 & 88.81 \end{pmatrix}$	$\begin{pmatrix} 96.59 & 1.63 & 1.36 & 0.42 \\ 1.37 & 72.02 & 16.71 & 9.90 \\ 3.26 & 12.64 & 73.62 & 10.48 \\ 0.49 & 5.97 & 4.84 & 88.70 \end{pmatrix}$	$\begin{pmatrix} 96.38 & 1.59 & 1.24 & 0.79 \\ 1.47 & 71.75 & 17.38 & 9.40 \\ 3.80 & 13.74 & 71.62 & 10.84 \\ 0.69 & 6.87 & 4.65 & 87.79 \end{pmatrix}$
document 5	$\begin{pmatrix} 89.49 & 7.05 & 0.68 & 2.78 \\ 13.72 & 45.11 & 29.39 & 11.78 \\ 1.00 & 22.71 & 67.43 & 8.86 \\ 0.97 & 6.31 & 4.55 & 88.17 \end{pmatrix}$	$\begin{pmatrix} 88.12 & 9.84 & 0.30 & 1.74 \\ 18.76 & 35.10 & 34.38 & 11.76 \\ 2.12 & 26.35 & 62.94 & 8.59 \\ 3.72 & 5.09 & 5.16 & 86.03 \end{pmatrix}$	$\begin{pmatrix} 85.29 & 10.50 & 1.45 & 2.76 \\ 17.21 & 34.92 & 41.75 & 6.13 \\ 2.60 & 19.85 & 72.50 & 5.05 \\ 3.20 & 6.94 & 6.20 & 83.66 \end{pmatrix}$

TABLE II.4 – Résultats 3-NN sur des bases de connaissances équilibrées construites à partir des documents 1 à 5

Conformément à nos attentes, les classes *texte* et *graphique* sont bien reconnues et séparées des autres. Les classes *publicité* et *texte&graphique* sont, en revanche, assez confuses. En effet, ces dernières présentent souvent des aspects similaires.

Nous obtenons des résultats similaires avec 3-NN et 1-NN. Des valeurs plus grandes de k permettent d'obtenir des résultats meilleurs mais il est nécessaire de disposer d'une base de connaissance suffisamment volumineuse dans ce cas d'utilisation.

Nous avons choisi des valeurs impaires de k afin d'éviter les cas indécidables.

3.3.3 Classificateur AdaBoost

Le tableau II.5 affiche les résultats obtenus à l'aide du classificateur Adaboost sur des bases d'apprentissage équilibrées construites à partir des documents 1 à 5. Ces dernières comptent respectivement 70%, 50% et 30% du total des échantillons.

Les valeurs de rappel montrent que les publicités sont mieux reconnues en employant ce classificateur que la méthode des plus proches voisins. En effet, AdaBoost permet de mettre en avant les caractéristiques les plus discriminantes pour chaque document de telle façon que le taux de reconnaissance soit plus élevé.

Nous constatons, en revanche, une chute des valeurs de précision en comparaison avec celles obtenues à l'aide de k-NN.

Par ailleurs, les poids calculés par AdaBoost permettent de conclure que les 21 descrip-

		Base d'apprentissage de quota 70%	Base d'apprentissage de quota 50%	Base d'apprentissage de quota 30%
document 1	*R*	74.33	77.86	84.16
	P	36.08	32.86	38.86
document 2	*R*	71.86	72.00	69.06
	P	38.51	39.70	31.09
document 3	*R*	87.28	84.03	85.83
	P	50.08	46.26	41.73
document 4	*R*	85.67	85.91	84.03
	P	43.64	44.73	36.86
document 5	*R*	83.60	81.24	88.08
	P	36.15	33.18	34.87
Global	*R*	83.99	82.77	84.05
	P	44.43	42.63	34.40

TABLE II.5 – Résultats AdaBoost sur des bases d'apprentissage / test de 3 tailles différentes

teurs sont globalement aussi utiles les uns que les autres, un sous-ensemble de descripteurs différent étant sélectionné sur chaque document.

3.3.4 Analyse des résultats

Évaluation La figure II.4 présente des échantillons représentatifs de publicités, d'articles de journaux réguliers et de blocs de type *texte&graphique*. Ces images laissent voir qu'il est très difficile de distinguer les échantillons positifs des négatifs, même à l'œil nu, notamment en l'absence de toute information sémantique (de la transcription par OCR). Ceci explique les faibles valeurs de rappel et notamment de précision obtenues.

Les taux de confusion élevés entre les deux classes *publicité* et *texte&graphique* sont intuitivement expliqués : les blocs publicitaires sont souvent composés d'éléments de texte et de graphiques. Nous approchons par là les limites de la reconnaissance de formes sans analyse de texte !

Si AdaBoost permet d'atteindre un bon taux de rappel, les résultats donnés par k-NN sont plus précis. Le choix entre les deux méthodes doit être guidé par l'application finale visée par l'utilisateur : AdaBoost est plus approprié aux finalités de pointage tandis que k-NN est mieux adapté aux applications de filtrage.

Les expérimentations révèlent qu'il n'y a pas besoin d'une base d'apprentissage volumineuse pour atteindre des résultats satisfaisants. En effet, étiqueter la moitié d'un numéro de magazine suffit à détecter les publicités dans les numéros suivants.

(a) Un bloc de *publicité*

(b) Un bloc de *texte&graphique*


ADVERTISEMENT

Warning Goes Out To All Women Taking Thyroid Hormones

If you are a woman who has recently been prescribed thyroid hormones or have been taking thyroid hormones for years and still have the same thyroid symptoms then you may be afflicted with a devastating disease process... that is going on inside your body right now...UNDIAGNOSED.To receive a free report chronicling how women on thyroid medication are one by one being mis-diagnosed and how this health care flaw is having devastating effects on women's health day after day...year after year. Call toll free 1-800-762-4779 or visit www.freewomensthyroidreport.com Yvoune Petrie D.C.

(c) Un bloc de *publicité*


Russian Rights Advocate, 82, Attacked in Moscow

By CLIFFORD J. LEVY

MOSCOW — Lyudmila M. Alexeyeva, a prominent Russian human-rights activist who is 82 years old, was punched in the head on Wednesday night while she was visiting a memorial in a subway station that was bombed by terrorists on Monday.

Ms. Alexeyeva suffered a minor injury, and the assailant was immediately arrested, she said in a telephone interview. She said she had gone to the Park Kultury station to pay her respects and place flowers on the memorial.

"Some young man came out who obviously had been waiting for me, and said, 'Are you still alive?' And then he hit me on the head," she said.

Ms. Alexeyeva, chairwoman of the Moscow Helsinki Group, has long helped lead human rights and political opposition campaigns in Russia and gained attention recently when a photo of her being detained by the police at a Moscow march was widely publicized.

She said she believed that the attack on her on Wednesday night was planned. "This man obviously had an assignment," she said.

The police told the Interfax news agency that the assailant was being questioned.

Suicide bombers attacked Park Kultury and another station, Lubyanka, during rush hour on Monday morning, killing 39 people and wounding scores of others. Large memorials have been established at both stations.

(d) Un bloc de *texte*

FIGURE II.4 – Échantillons positifs et négatifs ; (a) et (b) proviennent du même numéro de magazine et il va de même pour (c) et (d)

Résultats sur une base nettoyée Étant facilement reconnaissables sans classement, nous avons automatiquement éliminé les blocs de type *texte* et *graphique* de notre base à l'issue de la phase de segmentation en blocs. Pour ce faire, il suffit d'éliminer les blocs exclusivement composés de texte ou d'éléments graphiques. 547 blocs contenant du texte et des graphiques sont désormais candidats au classement. Cette nouvelle base compte 63% de publicités.

Le tableau II.6 montre les valeurs moyennes des résultats AdaBoost sur les 5 documents, la base d'apprentissage comptant 50% des blocs. Nous n'avons pas réévalué k-NN sur cette base puisqu'elle ne présente que deux classes de confusion : AdaBoost est donc plus propice à y atteindre de meilleurs résultats.

R	P
91.30	82.75

TABLE II.6 – Résultat AdaBoost sur la base réduite de 547 blocs, 50% se trouvant dans la base d'apprentissage

Nous pouvons remarquer une nette amélioration des résultats dans cette nouvelle base par rapport à la précédente. En effet, en présence d'échantillons variés, l'information principale permettant de distinguer les publicités des échantillons négatifs se trouve noyée parmi tant d'autres. La réduction du nombre de classes est donc similaire à une opération de débruitage de la base.

L'image II.5 illustre les résultats de classement sur 5 échantillons. Nous pouvons constater que le taux de détection est très bon vu la ressemblance entre les deux classes

à séparer et notamment l'absence de toute information colorimétrique.

Un échantillon positif correctement classifié

Un échantillon négatif correctement classifié

Erreur de classification sur un échantillon négatif

FIGURE II.5 – Résultat de classement sur des pages de magazine comportant 5 blocs candidats

Comparaison avec l'état de l'art Le classement d'images Web [69], en 'publicité' ou 'autre' en employant AdaBoost aboutit à un rappel de 87.66% et une précision de 72.33%. La base d'expérimentations compte 22.4% d'échantillons positifs tandis que notre base initiale compte seulement 8% de blocs publicitaires. La comparaison de nos résultats à ceux obtenus sur des images Web ne peut être objective en raison des différences manifestes

entre les deux corpus. En effet, ces images ne sont sujettes ni aux erreurs de segmentation physique ni aux bruits de numérisation, contrairement à nos blocs.

La détection de publicités dans les vidéos de football atteint un rappel de 92.55%. Étant donné que les descripteurs vidéo sont beaucoup plus riches que les nôtres, grâce à l'information spatio-temporelle, il est encore une fois difficile de comparer ces résultats aux nôtres.

Par ailleurs notre base comprend de multiples classes de confusion (*texte*, *graphique*, *publicité* et *texte&graphique*)tandis que les méthodes existantes n'en comptent que deux.

Considérant finalement que ces méthodes nécessitent un découpage 80-20 ou 90-10 entre les ensembles d'apprentissage et de test (tandis qu'une base d'apprentissage beaucoup plus réduite, à 30-70, suffit à notre méthode), nous pouvons considérer que nos descripteurs sont suffisamment génériques.

3.4 Conclusion

Si certains travaux sur la détection et la reconnaissance de publicités dans les images Web ou les vidéos existent dans la littérature, les images de presse n'ont jamais fait l'objet de base d'expérimentation pour de tels travaux. Notre contribution dans ce domaine de recherche s'avère donc nouvelle et prometteuse.

La détection de publicité dans les images de presse passe par le découpage des pages en blocs homogènes puis le classement de ces blocs. Nous employons, pour la segmentation physique de la page, une méthode simple et rapide basée sur l'agglomération d'entités de texte homogènes et guidée par l'information colorimétrique. Contrairement à la plupart des méthodes existantes, cette méthode ne requiert aucun modèle ou information *a priori*.

Nous avons calculé les descripteurs de classement à partir de l'information colorimétrique et textuelle acquise lors des phases précédentes. Ces caractéristiques ont été évaluées à l'aide de deux classificateurs qui s'avèrent complémentaires : k-NN est plus propice aux applications de filtrage de publicités tandis qu'AdaBoost est plus performant en pointage.

Nous avons tiré ces dernières conclusions suite aux expérimentations que nous avons menées sur une base représentative d'images de journaux et magazines de diverses provenances. Ces expérimentations montrent, par ailleurs, que les caractéristiques utilisées sont globalement aussi discriminantes les unes que les autres.

4 Conclusion

Nous avons présenté dans ce chapitre deux applications majeures de notre système de décomposition colorimétrique défini dans le chapitre précédent.

L'application la plus immédiate étant l'extraction de texte, nous avons commencé par la mise en place d'une méthode de détection de lignes de texte élaborée afin d'accéder à l'information textuelle, de préparer la segmentation physique et d'améliorer ainsi les résultats de l'OCR. Pour ce faire, nous avons opté pour une méthode hybride basée sur le regroupement de connexités homogènes dans les zones monochromatiques de l'image et sur la combinaison de la méthode des gradients cumulés avec une approche par fusion dans les régions polytonales.

Cette méthode a été évaluée indirectement en mesurant les performances du produit Abby Finereader sur les images de texte et en les comparant à celles atteintes sur les images brutes. Les expérimentations montrent une nette amélioration des valeurs de rappel en matière de segmentation en lignes par l'OCR. Cette amélioration est principalement due à l'extraction des éléments de texte écrit sur un fond non-uniforme.

L'information colorimétrique enrichie de l'information structurelle récemment acquise alimentent un moteur de classement d'images de journaux et magazines visant à détecter les blocs publicitaires. En effet, les descripteurs de classement sont calculés à partir de simples statistiques sur les aspects chromatiques et structurels des blocs concernés. Ces derniers sont issus d'une segmentation en bloc des pages de presse basée sur ces mêmes informations.

Nous avons comparé les résultats de classement de deux approches différentes dont l'efficacité est prouvée. K-NN permet de regrouper les blocs candidats en 4 classes dont la classe 'publicité'. Cet algorithme est particulièrement simple à implémenter et permet d'atteindre une précision de classement satisfaisante.

Pour sa part, le classificateur AdaBoost est capable de sélectionner les caractéristiques les plus discriminantes dans chaque base d'expérimentation. Nous atteignons de meilleures valeurs de rappel en employant cette méthode qu'avec la précédente.

De ce fait, l'utilisateur peut choisir d'employer AdaBoost s'il vise des applications de pointage ou k-NN s'il est plus intéressé par des applications de filtrage de zones publicitaires.

Les opérations de filtrage et pointage inhérentes à la détection de publicités par ces derniers classificateurs (AdaBoost et k-NN) ne sont pas fiables en raisons des valeurs de précision et rappel qui restent insuffisantes. En effet, ces dernières sont parfois au dessous de 50%.

Par conséquent, nous estimons qu'il serait intéressant de mettre au point une nouvelle méthode de classement qui soit plus robuste vis-à-vis de ce type de données.

Ainsi, les images de blocs de presse seront reclassées à l'aide de nouveaux classificateurs qui seront définis dans le prochain chapitre. Nous y présenterons également une méthode de classification non-supervisée conçue pour appréhender différents types d'objet.

Chapitre III

Classification auto-contrôlée

Table des matières

1 Introduction

1.1 Classification / Classement

1.1.1 Classification

La classification (*clustering* en anglais) consiste à regrouper les individus en classes. D'autres termes comme segmentation, quantification vectorielle, taxonomie numérique et apprentissage non-supervisé sont également employés pour désigner ce même procédé. La terminologie employée dépend souvent du cadre d'utilisation.

En effet, la classification peut alimenter des applications diverses et variées dans les domaines d'exploration, d'analyse des données, *etc.* Ces applications vont de la segmentation d'images et la classification de formes à l'indexation des documents ainsi que d'autres utilisation de type Data Mining.

1.1.2 Supervision

La classification est dite supervisée si l'on dispose de connaissances concernant l'appartenance des observations (comme par exemple : x_2 et x_{10} appartiennent à la même classe ou x_{36} et x_7 ne sont pas de la même classe) *a priori*. Autrement, la classification est dite non-supervisée.

Traditionnellement, l'emploi de certaines connaissances comme le nombre de classes ou le facteur d'échelle n'est pas considéré comme une supervision. Or, nous considérons que c'en est une.

En effet, les algorithmes de classification peuvent suivre deux tendances : l'émulation ou la découverte. Aux extrémités se trouvent les systèmes experts (émulation) et l'émergence (découverte). Entre les deux s'étend une multitude d'algorithmes de classement et classification. Ces derniers peuvent être caractérisés par leur aspect supervisé ou non-supervisé.

La supervision ne constitue donc pas une caractéristique binaire d'un algorithme. Elle peut en effet être présente de manière plus ou moins subtile ou impérative. Nous définissions alors les niveaux de supervision suivants :

- **Supervision 'assumée' :** en se plaçant plus du côté émulation que découverte, on trouve les algorithmes supervisés au sens fort du terme. On décrit à la machine le résultat que l'on souhaite obtenir mais on la laisse découvrir comment y arriver.
  L'exemple le plus immédiat est celui des réseaux de neurones. On fournit à la machine un ensemble de données (la base d'apprentissage sous forme de vecteurs de nombres réels), le classement escompté et un algorithme spécifique détermine la succession d'opérations simples à effectuer pour calculer une valeur indiquant la possibilité d'appartenance d'un échantillon à chaque classe.
- **Supervision 'candide' :** si l'on se place plus du côté de la découverte, on rencontre des algorithmes traditionnellement qualifiés de non-supervisés. On peut cependant discuter d'une telle qualification car la supervision n'est pas nécessairement visible. Prenons par exemple le cas des nuées dynamiques. Cette méthode est capable d'inventer des classes à partir de données pour lesquelles une distance et le centre sont calculables. Le seul paramètre à fournir à l'algorithme est le nombre de classes souhaité. Même si cela n'est pas considéré comme de la supervision *stricto sensu*, ce paramètre guide fortement l'algorithme vers un résultat donné. De plus, dans sa ver-

sion originale, la première étape des nuées dynamiques est de choisir aléatoirement un représentant pour chacune des classes (ce représentant aléatoire sera progressivement remplacé par des moyennes réellement représentatives des classes). Une modification triviale de l'algorithme consiste à choisir soi-même les représentants initiaux des classes. Cela permet de diriger la création des classes vers un résultat en particulier. Ainsi, un algorithme considéré comme non-supervisé peut très aisément devenir une machine suivant un modèle imposé par le concepteur.

- **Supervision 'sous-jacente' :** les algorithmes de classement et classification travaillent généralement sur des espaces vectoriels et/ou métriques. Les opérateurs ainsi mis en œuvre (addition, moyenne, distance, *etc.*) doivent être définis si les données ne sont pas de simples listes de valeurs numériques. Le choix dans la construction de ces opérateurs influe inévitablement sur le comportement des algorithmes susmentionnés.
  S'il est trop complexe de redéfinir ces opérateurs, il convient alors d'associer à chaque échantillon un vecteur de nombres réels qui le représentera lors du classement ou de la classification. Là aussi, le choix de ces vecteurs de caractéristiques n'est pas anodin.
  Prenons par exemple le cas d'un classificateur totalement non-supervisé que l'on appliquerait sur un ensemble de formes géométriques (figure III.1).

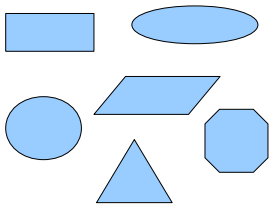


FIGURE III.1 – Formes géométriques quelconques

Si la caractéristique retenue est le nombre de sommets, on obtiendra la classification illustré par la figure III.2.
Cependant, si la caractéristique choisie est le rapport hauteur/largeur, on obtient quelque chose de totalement différent (figure III.3).
L'algorithme était pourtant le même et aucun paramètre ne lui a été fourni quant au nombre de classes où à leur taille. Le simple choix des caractéristiques peut donc très fortement influencer les résultats sans que l'on parle pourtant de supervision.

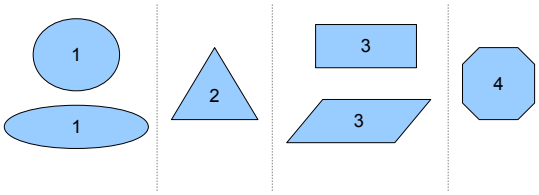


FIGURE III.2 – Classification selon le nombre de sommets

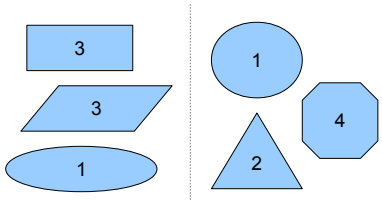


FIGURE III.3 – Classification selon le rapport hauteur/largeur

1.1.3 Classement

Le classement (*classification* en anglais) consiste à affecter des individus à des classes préexistantes. Ces classes peuvent éventuellement être obtenues par classification. On dit alors qu'il y a un apprentissage. La supervision est, dans ce cas, assumée. En revanche, dans le cadre d'une classification, la supervision est souvent candide ou sous-jacente.

1.1.4 Quelques définitions sous-jacentes

La classification et le classement s'appliquent à :

- des objets vectoriels sur lesquels on peut calculer des moyennes, des distances, *etc.*,
- ou des objets métriques pour lesquels seule la distance est définie.

Outre la multitude de types d'objets gérés, les méthodes de classification et classements varient selon différents critères. Par exemple, selon l'empiètement autorisé entre les différentes classes, nous distinguons :

- la classification sans recouvrement (partition) : les méthodes binaires exclusives selon lesquelles un objet appartient à une et une seule classe,
- la classification avec recouvrement : les approches binaires permissives selon lesquelles un objet appartient à une ou plusieurs classes
- approches de logique floue selon lesquelles un objet appartient à toutes les classes avec un degré d'appartenance défini pour chacune. Ainsi, on sait si un objet appartient plus fortement à une classe qu'à une autre et parfois s'il appartient à une seule

classe.

Dans la suite de cette étude, nous nous intéresserons plus particulièrement aux méthodes de classification produisant des classes disjointes et capables de traiter des objets vectoriels et requérant une supervision candide (choix de critères et des méthodes) voire sous-jacente (choix des distances, des métriques...). De telles approches ne passent naturellement par aucune phase d'apprentissage.

1.1.5 Évaluation

Les résultats de classification dépendent énormément de la nature des objets manipulés et même sur la même base de données, les performances varient selon les descripteurs utilisés (*cf.* figures III.2 et III.3).

Aucun algorithme ne peut être efficace dans tous les cas d'utilisation possibles : une méthode donnée peut être très performante sur certaines bases de données et beaucoup moins sur d'autres. En effet, l'évaluation de la qualité de classification est une notion subjective : deux utilisateurs opérants dans des domaines différents peuvent choisir de partitionner l'espace de façons différentes (voir Figures III.2 et III.3). C'est pour cette raison que la littérature compte une profusion de méthodes de classification.

Parmi les propriétés que nous pouvons considérer pour évaluer un algorithme de classification, nous citons :

- le type de données que la méthode peut gérer,
- son extensibilité à des bases volumineuses,
- sa capacité à gérer les espaces de dimensions élevées,
- sa robustesse et sa capacité de généralisation sur des classes un peu mélangées et pas complètement séparables,
- son aptitude à identifier des classes de formes irrégulières (non convexes),
- sa sensibilité aux échantillons isolés,
- son temps d'exécution (sa complexité),
- et notamment sa dépendance à des connaissances *a priori* ou des paramètres prédéfinis par l'utilisateur.

La plupart des méthodes de classification existantes requièrent une certaine connaissance des particularités de la technique de classification employée ainsi que des données à traiter. Sans ces pré-requis, le choix des paramètres de l'algorithme poserait un sérieux problème. En effet, les méthodes existantes les plus performantes sont lourdement paramétrées afin de s'adapter aux particularités de chaque jeu de données. Par ailleurs, ces techniques sont souvent de moindre performance à mesure que la dimension de l'espace grandit (malédiction de la dimension).

Pour toutes ces raisons, nous proposons une nouvelle approche de classification indépendante de tout paramètre abstrait, capable de traiter des données de grande dimension, qui soit simple d'utilisation, suffisamment générique et extrêmement rapide pour pouvoir s'utiliser facilement dans n'importe quel contexte industriel.

1.2 Positionnement

À l'origine de notre démarche était une volonté de généralisation de la notion d'accumulation d'indices. Cette accumulation passe généralement par une discrétisation de l'espace si les objets sont représentées dans un domaine continu.

Une discrétisation régulière de l'espace entraînerait des effets de bords inéluctables, c'est-à-dire des amas d'échantillons homogènes morcelés entre plusieurs classes. Cette sur-segmentation de l'espace engendrerait des groupements dont la taille n'est pas suffisamment grande pour déclencher une détection.

Le fondement de notre approche sera donc de découper l'espace de manière irrégulière, en fonction des données. Cela se rapproche des méthodes de classification basées sur la connectivité qui partitionnent l'espace de façon récursive. Il s'agit d'une approche en cascade qui subdivise l'espace de façon graduelle en changeant d'espace par une projection soit au départ soit à chaque récursion de façon à s'adapter aux données.

Dans un premier temps, nous avons étendu et redéfini le découpage en arbre à la manière des quadtree [29] et des octree [37] en le combinant à des techniques de bi-partitionnement dynamiques et complètement automatisées.

Nous avons ensuite appliqué ces mêmes techniques de partitionnement en insérant les données dans un arbre de type Kd-tree [8]. Cet arbre binaire permet de travailler dans des espaces de dimensions plus grandes. Cependant, ces traitements supposent que les axes sont décorrélés.

Pour remédier à ce problème, nous associons à nos deux modèles de représentation des données des techniques d'analyse et de décorrélation des données qui permettent une meilleure segmentation de l'espace. Il convient de préciser que cette analyse se passe de façon transparente vis-à-vis de l'utilisateur. Ce dernier n'a besoin d'avoir aucune connaissance *a priori* des données et ce tout au long du processus de classification.

Nous proposons de nommer notre méthodologie de classification/classement PPCAC (Partitionnement par Projection en Cascade Auto-Contrôlé) ou ACPP (*Automated Cascading Projection and Partitionning*).

1.3 Plan du chapitre

Après une étude de l'état de l'art qui couvre les différentes catégories d'approches de classification, nous présenterons notre méthode ainsi que ses différentes variantes dans la section 3.

Notre approche permet de traiter différents types de données. Or, comme nous nous intéressons aux images de documents, nous mesurerons ses performances sur une base de caractères numériques. Nous illustrerons ensuite l'étendue de son champ d'applications sur des bases d'images variées.

Dans la section 5, nous présenterons une façon d'adapter notre méthode de classification pour l'utiliser en mode classement.

2 État de l'art

Nous rappelons que les termes classification (*clustering*) et classement (supervision assumée) sont souvent confondus en raison de l'amalgame avec l'anglais. Jusqu'à mention du contraire, nous traiterons désormais uniquement de classification.

La littérature compte une profusion de méthodes de classification que nous pouvons regrouper selon différents critères, comme par exemple le nombre de partitions que la méthode produit, la forme des groupements construits, *etc.* La catégorisation des méthodes de classification n'est donc pas canonique : les différentes classes de méthodes s'enchevêtrent ; c'est-à-dire qu'une approche de classification donnée peut appartenir à plusieurs sous-ensembles de méthodes conjointement.

L'étude de l'existant nous a permis de relever différents critères permettant de séparer les familles d'approches. Les catégories citées ci-dessous donnent un petit aperçu sur cette diversité :

- **Approches hiérarchiques *vs* approches non-hiérarchiques :** les méthodes hiérarchiques, qui constituent une classification basée sur la connectivité, regroupent les données de façon graduelle en amas tandis que la seconde catégorie d'approches donne lieu à une unique partition de l'espace contrairement aux approches hiérarchiques qui forment des partitions imbriquées.
- **Utilisation séquentielle *vs* simultanée des caractéristiques :** le critère de séparation de types de méthodes se rapporte, dans ce cas, à la façon dont les vecteurs de caractéristiques interviennent dans le calcul de la mesure de similarité entre les données : les différents descripteurs peuvent être utilisés de façon séquentielle ou simultanée. La grande majorité des approches existantes font partie du second type de méthodes (*polythetic* en anglais). En effet, souvent, tous les descripteurs participent au calcul de la distance entre deux entités ; la décision de classification

étant fondée sur cette distance. En revanche, dans le cadre d'une méthode du type "utilisation séquentielle" (*monothetic* en anglais), chaque classe est définie par une conjonction de propriétés logiques agissant sur une seule variable parmi l'ensemble des descripteurs : un échantillon $X = (x_0, x_1, .., x_n)$ appartient à la classe c faisant intervenir l'axe i si et seulement si sa projetée x_i selon cet axe satisfait cet ensemble de propriétés. Un algorithme illustratif [15] de cette catégorie d'approches sera décrit dans la section 2.1.2.

- **Méthodes paramétriques *vs* méthodes optimisant un critère :** les méthodes dites paramétriques sont basées sur l'estimation de paramètres comme par exemple la distribution de probabilités dans le cas d'un mélange de Gaussiennes (EM). L'algorithme k-means est un exemple de méthodes qui optimisent un critère ; dans ce cas le critère optimisé est la variance intra-classes.

Nous nous attachons à étudier les méthodes de classification les plus représentatives de leurs domaines respectifs. Cette étude ne se veut pas exhaustive mais couvre l'ensemble de méthodes les plus liées à notre proposition. En effet, une étude exhaustive de l'état de l'art dans ce domaine requerrait des milliers de pages !

Nous avons choisi de regrouper les méthodes de classification en cinq catégories présentées dans les sous-sections ci-dessous, à savoir :

- classification basée sur la connectivité
- approches probabilistes
- méthodes basées sur les centres des classes
- approches basées sur la densité
- et classification basée sur les frontières des classes.

Nous décrirons quelques exemples d'algorithmes phares de ces catégories. Il existe, bien entendu, des approches hybrides ou qui ne peuvent pas être cataloguées selon le modèle établi. Citons par exemple l'accumulation de preuves qui consiste à considérer le résultat de classification de plusieurs algorithmes de classification, ou les approches de classification conçues pour les espaces de très grandes dimensions. Ces dernières approches seront présentées de façon plus sommaire.

2.1 Approches basées sur la connectivité

Les algorithmes de classification basée sur la connectivité sont souvent hiérarchiques. Ces derniers génèrent un ensemble de partitions imbriquées. L'arbre représentant cette hiérarchie de partitions est appelé dendrogramme. À chaque niveau de l'arbre correspond une partition composée de classes de taille de plus en plus réduite à mesure que l'on s'approche des feuilles.

Ce type d'algorithmes est avantageusement flexible par rapport au niveau de granularité : c'est-à-dire qu'il est toujours possible de couper le dendrogramme au niveau de la partition désirée. Par ailleurs, de tels modèles de représentation sont compatibles avec différents types de données d'où leur généricité.

Un algorithme de classification hiérarchique est de type ascendant (par fusions) ou descendant (par subdivisions). Une méthode ascendante fusionne récursivement les classes les plus similaires en partant d'un ensemble de singletons. Une approche descendante, quant à elle, découpe récursivement les nœuds de l'arbre en des classes de population de plus en plus réduite en partant d'une classe initiale contenant tous les échantillons de la base ; le processus de découpage s'arrête quand une condition d'arrêt (généralement le nombre k de classes) est atteinte.

2.1.1 Approches ascendantes

Algorithmes par liaisons Les approches de classification les plus classiques sont basées sur des métriques de liaison. Ces approches partent d'une matrice de similarité de taille $N \times N$ (N étant le nombre d'échantillons) appelée matrice de connectivité. Un graphe $G(X, E)$, dont les sommets X correspondent aux échantillons et les arêtes E aux mesures de similarités, est associé à cette matrice.

Selon la manière dont ce graphe est partitionné, nous distinguons les approches par liaison unique et les approches par liaison complète. Les méthodes par liaison unique considèrent que la distance entre deux classes correspond à la distance minimale entre toutes les paires d'individus inter-classes (chaque paire compte deux individus appartenant à des classes différentes). Dans le cadre de la méthode par liaison complète, c'est la distance maximale entre les paires d'individus qui est considérée.

Dans les deux cas, les classes les plus proches, selon un critère de distance minimale, sont fusionnées.

À l'issue de la classification, les classes formées par un algorithme par liaison complète sont compactes tandis que celles données par une méthode par liaison unique sont de formes allongées ; ces dernières classes sont affectées par l'effet de chaîne [51].

D'un point de vue pragmatique, les expérimentations montrent que les algorithmes par liaison complète produisent des hiérarchies de partitions plus utiles dans le cadre de nombreuses applications [54].

Classes de forme quelconque Les métriques de liaison utilisant la distance euclidienne forment naturellement des classes de forme convexe [63]. Or, les classes sont souvent de formes plus complexes dans la pratique.

Certaines méthodes ont été proposées à cet effet. Citons, à titre d'exemple l'algorithme CURE [40] qui est conçu pour gérer les bases de données volumineuses en data mining et qui donne lieu à des classes de tailles et formes variables.

Cette approche fait partie de la catégorie de méthodes où chaque classe est représentée par k prototypes. Il s'agit d'un nombre fixe d'individus dispersés autour du barycentre de la classe et distants de ce dernier d'une longueur α. Les deux classes présentant les deux représentants les plus similaires sont itérativement fusionnées.

L'algorithme CURE présente cependant l'inconvénient de dépendre de nombreux paramètres : α, k, le nombre de partitions ainsi que la taille des données. Par ailleurs, la forme des classes doit être connue *a priori* pour un choix pertinent des k représentants.

2.1.2 Approches descendantes

Les méthodes de classification hiérarchique descendante les plus connues sont des taxonomies binaires du type "caractéristiques utilisées séquentiellement" que nous avons introduit précédemment. De telles approches sont employées dans diverses applications, notamment la classification de documents, pour leur rapidité ainsi que leur aptitude à manier des bases de données volumineuses.

Un exemple d'algorithmes hiérarchiques descendants [15] est détaillé en annexe B. Les expérimentations montrent que cette méthode est efficace sur certaines bases de données (comme Iris de Fisher) et qu'elle l'est moins sur d'autres où les caractéristiques ne sont pas indépendantes.

Ce type d'algorithmes souvent nommé PDDP (Principal Direction Divisive Partitionning) a été initialement introduit par Boley [9].

Selon cette approche, une ACP est appliquée au niveau de chaque nœud d'un arbre binaire afin de déterminer la direction principale. Un hyperplan orthogonal à cette direction coupe ensuite l'espace en deux parties pour donner naissance à deux nœuds fils. Le nœud à découper est unique à chaque niveau du dendrogramme ; il est choisi en employant une mesure de variance. Ainsi, l'espace est récursivement bi-partitionné jusqu'à ce que le nombre maximal de classes K soit atteint.

Cet algorithme permet de produire une hiérarchie de partitions qui peut contenir des objets de natures diverses (documents, pixels, couleurs, *etc.*). Cette approche assez générique nécessite cependant l'intervention de l'utilisateur pour définir la profondeur de l'arbre ou le nombre total de nœuds dans l'arbre.

Ainsi, une série d'améliorations [109] a été apportée à cette approche au fil des années, principalement par la communauté du *data mining* en raison de sa capacité à gérer les grandes dimensions. Ces améliorations concernent principalement la façon de choisir le nœud à partitionner à un niveau donné, la façon de le partitionner et l'automatisation de

l'algorithme, c'est-à-dire tenter de déterminer automatiquement le paramètre K.

La plupart des algorithmes hiérarchiques dépendent également de ce même paramètre. C'est pour cette raison que certaines références bibliographiques [100] proposent des méthodes calculant le nombre optimal de classes dans une hiérarchie de partitions. Ces approches procèdent, souvent, par la recherche d'un point de forte courbure dans une courbe présentant une mesure de qualité de la classification (comme la variance intra-classes) en fonction du nombre de classes. Cette courbe est naturellement décroissante. En effet, une classe devient généralement plus homogène si sa population est réduite.

Le point recherché correspond au nombre optimum de classes. La courbe devrait présenter un aspect fortement croissant à gauche de ce point et un quasi-palier à sa droite, d'où sa forte courbure.

2.2 Méthodes probabilistes

Les méthodes probabilistes s'appuient sur un point de vue conceptuel : chaque classe est caractérisée par un modèle inconnu *a priori* dont les paramètres sont déterminés par une approche probabiliste. En d'autres termes, ces méthodes considèrent que les données sont issues d'un mélange de plusieurs populations dont les fonctions de distribution sont à caractériser.

Il existe plusieurs méthodes permettant de déterminer les paramètres d'une composition de distributions de probabilités. La plupart des travaux répondant à cette problématique considèrent que les composantes du mélange de distributions sont gaussiennes. À titre d'exemple, dans la figure III.4, la courbe verte est issue du mélange de trois gaussiennes de couleurs respectives bleu, jaune et rouge, et ce dans un espace de dimension 2.

L'estimation des paramètres passe, traditionnellement, par la maximisation itérative d'une fonction de vraisemblance.

L'algorithme EM présenté en annexe A est une méthode phare de ce domaine.

Les méthodes probabilistes donnent lieu à des classes intuitivement interprétables. Par ailleurs, disposant d'une représentation par classe, il est possible de calculer facilement et rapidement une fonction de distribution objective et globale caractérisant la partition formée (comme la vraisemblance logarithmique introduite en annexe A).

2.3 Méthodes basées sur les centres des classes

Les méthodes de type 'ré-estimation itérative des barycentres' représentent une classe par un point (un individu) unique.

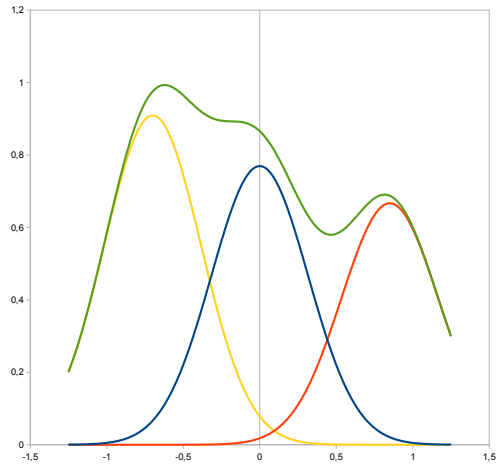


FIGURE III.4 – Mélange de trois Gaussiennes

Une fois les barycentres identifiés, chaque classe est identifiée par l'ensemble d'individus les plus similaires au centre qui lui est associé. La fonction objective consiste donc en une mesure de similarité entre un point et le barycentre de la classe à laquelle il appartient.

Ces approches sont généralement simples à implémenter, assez génériques puisqu'elles permettent de traiter différents types de données, et peu sensibles aux cas isolés.

Parmi les approches de ce type, l'algorithme K-means [14] constitue la méthode de classification la plus populaire. Cet outil est utilisé dans différents domaines aussi bien industriels que scientifiques pour sa simplicité, sa généricité et son efficacité.

2.3.1 Algorithme K-means

La méthode des nuées dynamiques (K-means) s'attache à scinder l'espace en un nombre K de classes fixé *a priori*.

Soient $\{C_1, ..., C_K\}$ l'ensemble de ces classes et $c_1, ..., c_K$ leurs barycentres respectifs. Le procédé K-means repose sur le déplacement des centres c_i ($i \in \{1, .., K\}$).

Initialisation K individus distincts $c_1^{(0)}, .., c_K^{(0)}$ sont choisis arbitrairement et donnent naissance à K classes $C_1^{(0)}, ..., C_K^{(0)}$ réduites à un unique élément (initialement).

Récurrence À chaque itération i, $i \geq 1$,

1. chaque individu x_e, $e = \{1, .., N\}$ est affecté à la classe C_j telle que

$$j = \text{Argmin}_{g=\{1,...,K\}}\{d(x_e, c_g^{(i-1)}\}; \tag{III.1}$$

d étant une mesure de distance ;

2. les barycentres $c_1^{(i)}, ..., c_K^{(i)}$ des nouvelles classes $C_1^{(i)}, ..., C_K^{(i)}$ sont ensuite calculés ;
3. L'algorithme s'arrête quand

$$c_g^{(i)} = c_g^{(i-1)} \; \forall g \in \{1, ..., K\}. \tag{III.2}$$

La partition $\{C_1, ..., C_K\}$ formée à la dernière itération constitue le résultat de l'algorithme.

2.3.2 K-means Intelligent

Les principaux inconvénients de l'algorithme des nuées dynamiques réside dans sa dépendance du paramètre K ainsi que du choix des centres initiaux.

Nombre d'approches ont été proposées pour sélectionner les échantillons d'initialisation ou pour déterminer le nombre optimal K de classes [19, 53].

Considérons, par exemple, l'approche 'Intelligent K-Means' [77] dont le principal apport réside dans l'initialisation optimale, entre autres la détermination de K, de l'algorithme k-means. L'approche 'Anomalous Pattern' décrite ci-dessous est employée pour assurer cette initialisation.

Algorithme Anomalous Pattern (AP) Soient $\mathcal{O}$ l'origine de l'espace, $\mathcal{E}$ une population et S un ensemble initialement vide. Les étapes suivantes définissent l'algorithme AP.

1. Trouver l'individu I le moins similaire à $\mathcal{O}$. Ajouter ensuite I dans S. Nous avons donc initialement $c = I$, c étant le barycentre de S.
2. Calculer les distances $d(x, c)$ et $d(x, \mathcal{O})$ $\forall x \in \mathcal{E}$. Si $d(x, c) < d(x, \mathcal{O})$ alors x est affecté à S.
3. Recalculer le barycentre $c^{(i)}$ de S à l'itération i.
4. Si $c^{(i)} \neq c^{(i-1)}$, commencer une nouvelle itération à partir de la deuxième étape (itération $i + 1$) ; sinon l'algorithme s'arrête.
5. $\mathcal{E} \leftarrow \mathcal{E} - S$.

Algorithme K-means intelligent Cet algorithme diffère de la méthode K-means classique uniquement par la phase d'initialisation.

Cette étape consiste à appeler itérativement la procédure AP jusqu'à ce que l'ensemble $\mathcal{E}$ soit vide. Les centroïdes initiaux associés à k-means sont définis par les barycentres

respectifs des ensembles $S^{(i)}$ formés lors des itérations i, $i = 1, .., T$; T étant le nombre d'itérations.

L'algorithme K-means intelligent est résumé ci-dessous.

1. Initialement (à l'itération $i = 0$), l'ensemble $\mathcal{E}^{(0)}$ compte tous les échantillons.
2. Appliquer le procédé AP à $\mathcal{E}^{(i-1)}$ pour former $S^{(i)}$. Ensuite, $\mathcal{E}^{(i)} \leftarrow \mathcal{E}^{(i-1)} - S^{(i)}$
3. Si $S^{(i)} \neq \phi$ et $\mathcal{E}^{(i)} \neq \phi$, $i \leftarrow i + 1$ et revenir à l'étape numéro 1.
4. Éliminer tous les ensembles $S^{(i)}$ tels que $\text{card}(S^{(i)}) = 1$. Soit K le nombre des ensembles $S^{(i)}$ restants et $c_1, ..., c_K$ leurs barycentres respectifs.
5. Appliquer l'algorithme K-means en l'initialisant avec $c_1, ..., c_K$.

Des études comparatives [19] de nombre de variantes de l'algorithme des nuées dynamiques révèlent que la méthode k-means intelligent permet d'atteindre les meilleurs résultats sur différentes bases de données.

2.4 Méthodes basées sur la densité

Le concept de densité spatiale évoque les notions de voisinage et de frontière. Ces concepts sont étroitement liés à l'idée du plus proche voisin. En effet, une classe définie comme étant un ensemble de composantes voisines se propage dans la direction et le sens induits par les fortes densités.

Ainsi, les méthodes de classification basées sur le calcul de densité [101, 34] peuvent donner lieu à des classes de formes irrégulières (voir figure III.5). Par ailleurs, de telles approches sont naturellement protégées des cas isolés.

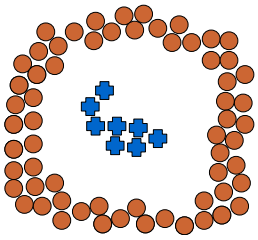


FIGURE III.5 – Exemples de classes de formes irrégulières

Il convient toutefois de préciser que ces approches présentent certains inconvénients considérables :

- de tels algorithmes ne permettent pas de détecter les classes composées de zones de densités non homogènes (au delà d'un certain seuil d'homogénéité) ; or les partitions de répartition uniforme sont extrêmement rares ;

- de par leur incapacité de gérer des classes de densités variables, ces méthodes sont sensiblement dépendantes d'un seuil de voisinage. De surcroît, ce paramètre est généralement difficile à déterminer sans connaissance *a priori* suffisante sur les données ;
- les classes de formes irrégulières sont souvent difficiles à interpréter, notamment par un utilisateur non informaticien.

L'algorithme Mean-Shift [34] est un exemple d'approches basées sur la densité. Il s'agit d'une méthode itérative appliquée dans divers domaines comme la vision par ordinateur, la quantification colorimétrique, *etc.*

Cette méthode ne requiert pas la connaissance *a priori* du nombre de classes K et permet de séparer des classes de formes diverses et irrégulières.

Il est cependant nécessaire de fixer pour cet algorithme un noyau $\mathcal{K}(x)$ et un rayon (facteur d'échelle) h.

2.4.1 Définition

Dans un espace réel de dimension d, un noyau est une fonction satisfaisant les propriétés suivantes :

- $$\int_{\mathbb{R}^d} \Phi(x)\,\mathrm{d}x = 1,$$
- $$\Phi(x) \geq 0 \,\forall x \in \mathbb{R}^d.$$

Le noyau le plus communément utilisé est le noyau Gaussien défini par

$$\Phi(x) = e^{-\frac{x^2}{2\sigma^2}}. \tag{III.3}$$

À titre indicatif, il existe d'autres types de noyaux comme le noyau rectangulaire uniforme :

$$\Phi(x) = \left\{ \begin{array}{ll} \frac{1}{b-a} & \text{si} \quad a \leq x \leq b \\ 0 & \text{sinon} \end{array} \right\} \tag{III.4}$$

ou le noyau de Epanechnikov :

$$\Phi(x) = \left\{ \begin{array}{ll} \frac{3}{4}(1-x^2) & \text{si} \quad |x| \leq 1 \\ 0 & \text{sinon} \end{array} \right\}. \tag{III.5}$$

La figure III.6 illustre les trois noyaux cités ci-dessus.

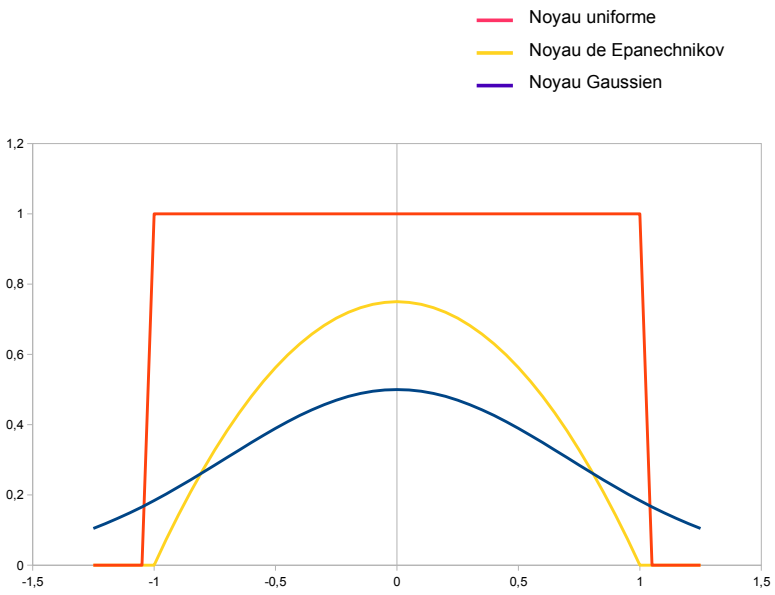


FIGURE III.6 – Exemples de noyaux

2.4.2 Algorithme Mean-Shift

Soit un ensemble $\{x_1, ..., x_N\}$ de N échantillons, h un rayon fixe et $\mathcal{K}$ un noyau. Les étapes suivantes sont appliquées à chacun des points x_e ; $e = \{1, ..., N\}$.

1. À l'itération $i = 0$, initialiser le barycentre y avec l'individu x_e : $y^{(0)} \leftarrow x_e$.
2. Calculer la valeur Mean-Shift $m_{h,\mathcal{K}}(y^{(i)})$

$$m_{h,\mathcal{K}}(x) = \frac{\sum\limits_{x_j \in \mathcal{V}(x)} x_j \, \mathcal{K}(\| \, x - x_j \, \|^2)}{\sum\limits_{x_j \in \mathcal{V}(x)} \mathcal{K}(\| \, x - x_j \, \|^2)} - x; \tag{III.6}$$

$\mathcal{V}(x)$ étant la fenêtre de Parzen centrée en x et définie par :

$$\mathcal{V}(x) = \{x_j \quad / \quad d(x, x_j) < h\}. \tag{III.7}$$

3. Déplacer le point p : $y^{(i+1)} \leftarrow y^{(i)} + m_{h,\mathcal{K}}(y^{(i)})$.
4. Incrémenter i.
5. Reprendre l'algorithme en 3 jusqu'à la convergence vers le point y tel que $\| \, m_{h,\mathcal{K}}(y) \, \| < \epsilon$

L'algorithme Mean-Shift consiste donc à déplacer itérativement tous les points de la base jusqu'à la convergence vers le point de densité maximale y.

Le nombre de classes de la partition finale correspond au nombre de points stationnaires y différents.

2.5 Classification basée sur les frontières des classes

Les réseaux de neurones artificiels [46] sont utilisés par un large public pour la classification et notamment pour le classement.

La méthode SOM (Self-Organizing Map), ou carte auto adaptative ou carte de Kohonen, est un réseau de neurones particulièrement populaire dans le domaine de la quantification vectorielle. Cet algorithme adopte une représentation intuitive des données multidimensionnelles en employant un carte bi-dimensionnelle, ce qui permet une bonne visualisation des classes résultantes.

Cette méthode permet de sélectionner les descripteurs les plus discriminants et de les normaliser via les poids associés aux différents neurones. Or, une mauvaise initialisation de ces derniers affecte les résultats de la classification sensiblement.

Par ailleurs, la convergence de l'algorithme SOM dépend de nombreux paramètres comme le taux d'apprentissage, le rayon de voisinage, *etc.*

2.6 Outils connexes

2.6.1 Accumulation de preuves

L'accumulation de preuves (*evidence accumulation*) consiste a appliquer plusieurs algorithmes de classification et à déduire le résultat final en se basant sur un vote.

Considérons, par exemple, l'approche [32] qui consiste à construire la matrice de similarité à partir de plusieurs partitionnements (en employant k-means avec différentes valeurs de k par exemple). Cette matrice, donne lieu à un graphe qu'il suffit de seuiller pour obtenir la classification finale. Cela revient au final à une approche de type 'classification basée sur la connectivité'.

Une autre approche initialement proposée par Ho et *al.* [47] et perfectionnée récemment [81] consiste à considérer le résultat de vote de plusieurs arbres de partitions aléatoires.

De telles approches sont naturellement plus coûteuses en termes de temps d'exécution.

2.6.2 Réduction de dimensionnalité

La plupart des approches existantes sont affectées par la malédiction de la dimensionalité : à partir d'une certaine dimension de l'espace, les performances diminuent. Si l'on considère l'algorithme k-means, par exemple, ce dernier devient moins performant dès que la dimension dépasse quelque centaines. Par ailleurs, ces méthodes sont de plus en plus lentes à mesure que la dimension augmente.

En revanche, les méthodes analysant les caractéristiques de façon séquentielles, qui sont souvent hiérarchiques, sont moins sensibles à ce problème. Ces approches basées sur la connectivité sont, de surcroît, souvent les moins coûteuses en termes de temps d'exécution. Cependant, ces dernières ne sont pas très performantes puisque les caractéristiques sont rarement décorrélées et indépendantes.

Pour tenter de résoudre ces problèmes, certaines approches réduisent la dimension de l'espace en utilisant une ACP (Analyse en Composantes Principales), par exemple. Or l'ACP peut être pénalisante sur certains jeux de données [2]. La comunauté du *data mining* propose un ensemble de techniques (DBSCAN, OPTICS, *etc.*) [58] afin d'affronter la malédiction de la dimensionnalité.

2.6.3 Approches évolutionnaires

Les approches évolutionnaires peuvent être utilisées pour estimer les paramètres d'un algorithme de classification. Ces approches sont inspirées par l'évolution naturelle : la partition finale est obtenue à partir d'une série d'opérations génétiques, à savoir la sélection, la recombinaison et la mutation, appliquées à une partition initiale arbitraire.

Les solutions possibles sont assimilées à des chromosomes. Les opérations génétiques transforment ces derniers en d'autres partitions progressivement jusqu'à la convergence, c'est-à-dire jusqu'à ce que la vraisemblance de survie d'un chromosome à une nouvelle génération soit suffisante.

Les techniques évolutionnaires les plus populaires sont les algorithmes génétiques (GAs) [38]. Les solutions données par ces approches sont sous forme de codes binaires dont les bits peuvent changer d'une génération à l'autre.

L'algorithme k-means a été combiné avec la méthode GAS en remplaçant les centres par des vecteurs binaires [102]. Chaque vecteur définit d codes génétiques, un code étant associé à chaque coordonnées de l'espace. Ces chromosomes sont itérativement modifiés via des opérations de croisement génétique et de mutation jusqu'à la convergence vers la solution optimale.

Une telle approche nécessite donc, entre autres, la définition d'une fonction de vraisemblance (*fitness*) permettant de juger de la qualité de chaque chromosome.

L'emploi de l'algorithme génétique permet de détecter des classes de tailles variables et de formes allongés (contrairement à l'algorithme k-means classique).

Encore une fois, l'inconvénient majeur de ce type d'approches réside dans sa dépendance de certains paramètres décisifs comme la fonction de vraisemblance, le nombre de classes, *etc.*

2.7 Bilan et conclusion

Nous avons présenté, dans cette section, une classification des méthodes de classification les plus populaires et les plus liées à notre proposition.

Nous pouvons examiner le problème de classification sous différents angles : selon le modèle de représentation des classes, la mesure de similarité adoptée, la forme des classes, *etc.*

En se basant sur une combinaison de critères, nous distinguons les approches basées sur la connectivité ou hiérarchiques, les méthodes probabilistes, celles basées sur les barycentres ou la densité locale et les réseaux de neurones SOM.

Les méthodes hiérarchiques se divisent en approches ascendantes et approches descendantes. Ces dernières sont plus populaires. Elles sont basées sur la mise à jour itérative d'une matrice de similarité. La littérature compte plusieurs méthodes hiérarchiques descendantes dont le seul point de divergence réside dans la manière dont cette matrice est appréhendée.

Le principe général des approches hiérarchiques descendantes consiste à subdiviser itérativement les classes représentés par les nœuds d'un arbre en partant d'une classe unique contenant toutes les données.

Les approches hiérarchiques sont avantageusement peu paramétrées, d'où leur généricité. En revanche, ces méthodes donnent lieu à une multitude de partitions imbriquées. L'utilisateur a donc besoin de déterminer la partition optimale qui lui convient. Il existe néanmoins des métriques permettant de mesurer la pertinence d'une cascade de partitions et d'automatiser ainsi le choix de la partition optimale.

Les approches non-hiérarchiques donnent lieu à une partition unique des données ou de l'espace.

Les méthodes par réestimation itérative des barycentres, notamment les variantes de la méthode k-means, sont les plus communément employées. Ces approches partitionnent les données en k classes de formes convexe en se basant sur la distance des échantillons par rapport aux barycentres des k-classes. Ces algorithmes sont simples et rapides. Cependant,

ils requièrent souvent la connaissance *a priori* du nombre de classes k quoiqu'il existe des solutions permettant d'estimer ce nombre automatiquement.

Les méthodes probabilistes, telles que la méthode EM, donnent des résultats très similaire à ceux associés à la méthode des nuées dynamiques, c'est-à-dire des classes isotropes de forme convexe.

Contrairement aux approches non-hiérarchiques citées ci-dessous, les méthodes par maximisation de la densité sont aptes à détecter des classes de formes irrégulières. En revanche, il est souvent nécessaire de connaître la forme des classes et leurs tailles *a priori* afin de définir les paramètres adéquats notamment la fonction noyau, le rayon de classe, *etc.*

La littérature compte d'autres types d'approches non-hiérarchiques, comme par exemple les cartes de Kohonen, qui peuvent séparer les données de façon optimale à condition que les paramètres des algorithmes associés soient bien choisis.

Ne disposant d'aucune connaissance *a priori* sur les données à traiter et visant un moteur de classification générique, nous éviterons les approches lourdement paramétrées.

À la manière des méthodes hybrides [121, 32] qui permettent de tirer profit des points forts de différents approches, nous proposerons un algorithme par partitionnement selon la principale direction en cascade inspiré des approches hiérarchiques descendantes ainsi que de certaines méthodes probabilistes ou basées sur les barycentres...Cette approche est, par ailleurs, assez robuste vis à vis du problème de la dimensionnalité.

3 Notre contribution

Nous optons pour une approche hiérarchique descendante qui présente certains points de similitude avec les méthodes dites PDDP [9, 109] (partitionnement selon la principale direction). Nous pouvons également considérer que notre approche est une généralisation des algorithmes PDDP.

Après une description globale et sommaire de la méthode dans la section 3.1, nous détaillerons ses différentes composantes : la nature de l'arbre utilisé pour représenter la hiérarchie de partitions, la méthode de subdivision d'un nœud donné de l'arbre ainsi que l'espace vectoriel dans lequel les données sont représentées.

3.1 Vue d'ensemble

Soit Ω notre espace de représentation et d sa dimension. Au niveau de chaque nœud de l'arbre, les données sont projetées selon m axes, $m \in [\![1..d]\!]$. Ces axes sont définis par

un 'analyseur-projecteur' que nous spécifierons ultérieurement. Ce dernier peut être une méthode d'analyse des données comme, par exemple, l'ACP (Analyse en Composantes Principales).

Ces axes sont ensuite manipulés indépendamment les uns des autres. Nous avons donc besoin d'assurer que les différents axes de projection sont effectivement décorrélés et indépendants. Le changement d'espace de représentation peut être effectué, au choix, une seule fois sur l'ensemble de la population, au niveau du nœud origine, ou alors de façon répétée au niveau de chaque nœud non terminal.

Un 'partitionneur' associe à tout ou partie de ces axes des hyperplans qui leur sont respectivement orthogonaux. Il s'appuie sur une méthode de bi-partitionnement, comme par exemple 2-means ou EM, qui détermine le point d'intersection entre chaque axe et l'hyperplan qui lui est associé. Les axes concernés par ces calculs sont sélectionnés par ce même partitionneur. Ce dernier peut refuser de partitionner selon un axe donné si :

- les données projetées selon cet axe sont suffisamment homogènes ; aucune séparation n'est donc nécessaire (selon cet axe),
- ou alors le nombre d'individus est trop réduit. Dans ce cas, le nœud en cours d'analyse est jugé une feuille (aucun axe n'est sélectionné).

Soit d_s le nombre de bi-partitionnements effectués (d'axes sélectionnés). À l'issue de cette suite d'opérations, l'espace est subdivisé en 2^{d_s} régions disjointes. Un nœud fils est associé à chacune des régions non vides. Les individus du nœud sont alors répartis entre ses fils.

3.2 Modèles hiérarchiques proposés

3.2.1 Arbre non-contraint

Dans un espace de dimension d, nous définissons une structure de données de type arbre dans laquelle chaque nœud peut compter jusqu'à 2^d fils.

Chaque nœud de l'arbre, caractérisé par d_s axes [1] sélectionnés, est subdivisé en 2^{d_s} régions par d_s hyperplans respectivement orthogonaux aux d_s axes. Un nœud fils est associé à chacune des régions non vides de l'espace.

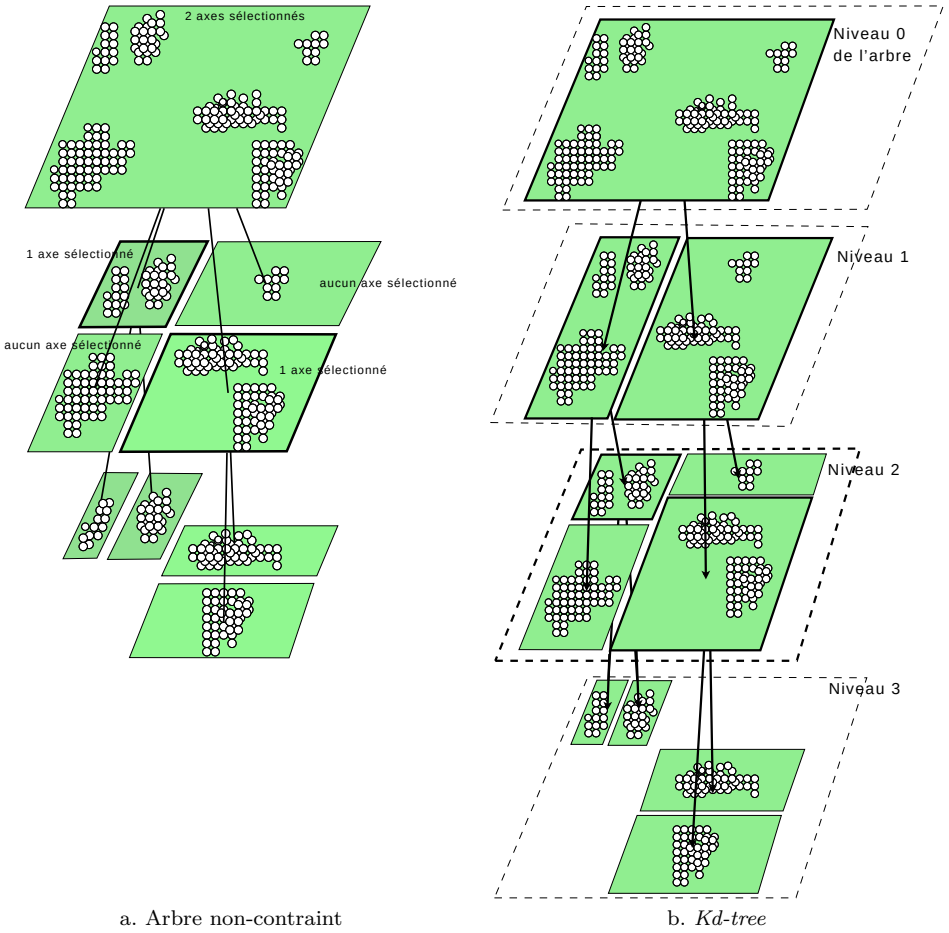
Autrement dit, si $\{A_i\}_i$ est l'ensemble d'axes sélectionnés pour la séparation, un point de partitionnement $\mathcal{P}_i$ est associé à chacun des axes A_i. Ainsi, $\mathcal{P}_i$ est le point d'intersection entre A_i et l'hyperplan qui lui est associé. Un individu $x = (x_1, ..., x_d)$ de l'échantillon est assigné à un nœud fils différent selon que $x_i \leq \mathcal{P}_i$ ou $x_i > \mathcal{P}_i$. Ainsi d_s comparaisons sont

1. d_s est le nombre d'axes sélectionnés par le partitionneur ou prédéfini par l'utilisateur. On a $0 \leq d_s \leq d$. d_s est donc le nombre d'axes pris en compte pour subdiviser un nœud donné.

effectuées pour chaque individu du nœud ; le nombre de sous-classes s'élève donc à 2^{d_s}. Nous associons un nœud fils à chaque sous-classe non vide.

Ce processus de projection / division est récursivement appliqué à tous les nœuds de l'arbre jusqu'aux feuilles caractérisées par $d_s = 0$.

La figure III.7.a illustre ce processus dans un espace de dimension 2 (dans ce cas l'arbre est appelé arbre quaternaire connu sous le nom de *quadtree*).



a. Arbre non-contraint b. *Kd-tree*

FIGURE III.7 – Deux modèles d'arbre dans un espace de dimension 2 (sans projecteur)

3.2.2 Arbre binaire

En employant un arbre binaire d_s peut prendre deux valeurs possibles : 1 ou 0 si le nœud est une feuille. Dans le cas où le nœud n'est pas terminal, le partitionneur calcule, pour l'axe sélectionné A_s, un point de partitionnement $\mathcal{P}_s$. Pour tout individu $x = (x_1, ..., x_d)$ du nœud en cours, si $x_s \leq \mathcal{P}_s$ x est assigné au nœud fils droit, sinon il est affecté au fils gauche.

Un nœud est une feuille si aucun axe n'est sélectionné par le partitionneur [2].

La figure III.7.b représente les données utilisées dans l'arbre quaternaire (figure III.7.a) dans un *Kd-tree*.

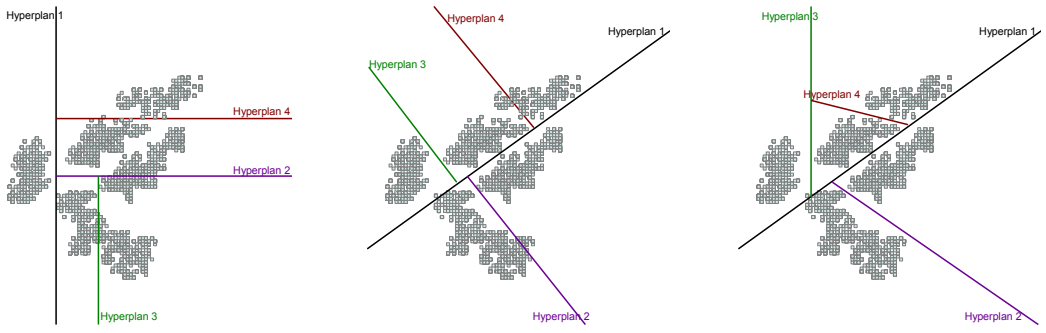
3.3 Analyseurs-projecteurs

Les descripteurs de données issus d'une simple extraction de caractéristiques ne sont pas tous aussi discriminants les uns que les autres et présentent souvent une certaine redondance. Nous pouvons interpréter ces observations en associant un poids proportionnel à son pouvoir discriminant à chaque axe de l'espace et en détectant toutes les combinaisons de descripteurs corrélés.

L'Analyse en Composantes Principales (ACP), a l'avantage de créer de nouveau axes triés par ordre de pouvoir discriminant. Qui plus est, la dimension de l'espace se voit réduite grâce à ce projecteur.

L'analyse discriminante (LDA) permet une analyse supervisée des données. Cette analyse combine linéairement les données de manière à maximiser la discrimination entre les classes.

En employant un arbre non-contraint ou binaire, il est possible d'effectuer une analyse des données et de réduire ainsi la dimension une seule fois sur l'ensemble des données avant de construire l'arbre (figure III.8.b) ou alors récursivement au niveau de chaque nœud de l'arbre (figure III.8.c).



a. Partitionnement sans projecteur b. Projecteur appliqué au préalable c. Projecteur appliqué en cascade

FIGURE III.8 – Différentes applications d'un analyseur-projecteur dans le cas d'un *Kd-tree*

2. Pour éviter toute ambiguïté, nous rappelons que c'est l'analyseur-projecteur qui définit l'ensemble d'axes possibles (m axes) et le partitionneur sélectionne, parmi cet ensemble, d_s axes pris en compte lors des bi-partitionnements.

3.3.1 Méthode ACP

L'Analyse en Composantes Principales (*PCA* en anglais pour *Principal Component Analysis*) permet de projeter les données dans un espace de plus faible dimension sans perte significative d'information.

Du point de vue géométrique, l'ACP est une projection qui transforme l'espace original des caractéristiques dans un nouveau repère qui maximise la répartition des points le long des axes créés. Mathématiquement parlant, il s'agit d'un ensemble de transformations linéaires agissant sur les descripteurs d'origine et dirigées par la variance. Une description détaillée de cette approche est pourvue en annexe C.

3.3.2 Méthode LDA

À la différence de l'ACP, l'analyse discriminante n'est applicable que dans le cadre d'une supervision assumée, c'est-à-dire que les étiquettes des données (leurs classes respectives) doivent être connues *a priori*.

La LDA (*Linear Discriminant Analysis*) est basée sur le critère de la maximisation de l'écart entre les classes et la minimisation de la dispersion autour de la moyenne de chaque classe. Cette analyse combine linéairement les données de manière à maximiser la discrimination entre les classes.

Le nombre d'axes générés par la LDA est inférieur ou égal au nombre d'étiquettes différentes.

Contrairement à l'analyse factorielle qui maximise la variance des N observations en fonction des d variables, l'analyse discriminante maximise la répartition des N observations dans leurs classes respectives.

Cet approche est développée dans l'annexe D. Nous y faisons appel dans le cadre d'une application supervisée présentée dans la section 5.

3.3.3 Analyse en composantes indépendantes

L'Analyse en Composantes Indépendantes (ACI) [22] consiste en la recherche d'une transformation linéaire permettant l'obtention d'un résumé (exhaustif ou comprimé) des données sous forme de composantes statistiquement indépendantes. Ce concept peut être vu comme une extension de l'Analyse en Composantes Principales (ACP) par le recours aux statistiques d'ordre supérieur à deux.

Une variante assez populaire de ce type de transformation est connue sous le nom Fast-ICA [50]. Cet algorithme dépend d'un certain nombre de paramètres 'abstraits'. C'est pour cette raison que nous n'avons pas pu mener des tests conséquents en employant cette technique.

3.4 Partitionneurs

Chaque nœud d'un arbre non-contraint (resp. *kd-tree*) est subdivisé en 0 à 2^{d_s} (*resp.* 2) sous-classes en effectuant de 0 à d_s (d_s vaut 1 dans le cas d'un arbre binaire) subdivisions successives de l'espace.

Le nombre d_s de directions de l'espace concernés par la subdivision et les hyperplans associés peuvent être estimés par différentes méthodes de partitionnement. Nous avons choisi, à cet effet, d'adapter les méthodes EM et AP respectivement.

Nous associons à chaque partitionneur un critère d'arrêt basé sur une mesure d'évaluation interne, donc calculée automatiquement à partir des résultats d'analyse des données. Pour le partitionneur EM, par exemple, cette mesure correspond à la vraisemblance logarithmique. Cela limite le besoin de connaissance *a priori* sur les données et réduit l'intervention de l'utilisateur au maximum.

Nous rappelons que Chavent *et al.* [15] découpent systématiquement tous les nœuds de l'arbre jusqu'à ce que le nombre de classes à un niveau donné atteigne une valeur prédéfinie K. Une fois le nœud à découper choisi, la subdivision se fait en évaluant la variance intra-classes engendrée par chacun des seuils possibles (*cf.* section 2.1.2). Nous n'avons pas opté pour ce type de partitionnement en raison de sa complexité élevée (le temps d'exécution requis).

En employant un arbre non-contraint, l'algorithme de partitionnement est séquentiellement appliqué suivant tous les axes de l'espace. En revanche dans le cadre d'un arbre binaire, il est appliqué suivant un seul axe A_s.

Soit A_i la direction de l'espace considérée à une itération donnée ; $i = 1, .., d$ dans le cas d'un arbre non-contraint et $i = s$ si l'arbre est de type binaire. Nous considérons la projection des données $\{x_i^j\}_{j=1,..,n}$ du nœud selon l'axe A_i. Ainsi, chaque partitionnement s'effectue dans un espace mono-dimensionnel.

Initialement, $d_s \leftarrow d$ dans le cas où l'arbre utilisé est de type 'non-contraint'.

3.4.1 Algorithme EM et mélange de Gaussiennes

EM est une approche statistique. Les calculs ne prennent donc pas de sens si l'échantillon est de taille trop réduite. Nous veillerons donc à ne pas partitionner les nœuds composés de peu d'éléments.

Déroulement La méthode EM (présentée dans la section 2.2) est appliquée deux fois successives :

- une première fois en supposant que les données (les valeurs des projections selon A_i) sont issues du mélange de deux Gaussiennes,

• et la deuxième fois en supposant que ces données suivent une loi normale.

L'algorithme EM permet de caractériser les lois de distributions, dans les deux cas, et de déterminer leurs vraisemblances logarithmiques respectives $\mathcal{V}_1$ et $\mathcal{V}_2$.

Nous rappelons qu'il s'agit de la vraisemblance du produit

$$\prod_{j=1}^{n} \sum_{k=1}^{K} \alpha_k \mathcal{N}(x^j; \mu_k, \sigma_k); \tag{III.8}$$

K étant le nombre de Gaussiennes mélangées : K vaut 2 et respectivement 1 dans notre cadre d'utilisation. Les points x^j, $j \in [\![1..n]\!]$, constituent l'ensemble d'individus dans le nœud.

La vraisemblance logarithmiques d'un mélange de K Gaussiennes vaut :

$$\mathcal{V}_K \;=\; \sum_{j=1}^{n} \log\left(\sum_{k=1}^{K} \alpha_k \, \mathcal{N}(x_i^j; \mu_k, \sigma_k)\right) \;=\; \sum_{j=1}^{n} \log\left(\sum_{k=1}^{K} \frac{\alpha_k}{\sigma_k\sqrt{2\pi}} \, e^{-\frac{1}{2}\left(\frac{x_i^j - \mu_k}{\sigma_k}\right)^2}\right); \tag{III.9}$$

α_k, μ_k et σ_k sont respectivement définis dans les équations A.8, A.9 et A.10 ; x_i^j est la projection du $j^{ème}$ individu selon l'axe A_i.

La vraisemblance en fonction du nombre de Gaussiennes K est une application croissante. Ainsi, $\mathcal{V}_K$ atteint son maximum lorsque $K = N$ (chaque classe compte un seul individu) et nous avons toujours $\mathcal{V}_2 > \mathcal{V}_1$.

Il existe, dans la littérature, certaines approches permettant de calculer le nombre optimal de classes pour une population donnée. Or, la majorité de ces méthodes nécessite la réalisation de nombreux partitionnements, avec un nombre K différent pour chaque test, afin de trouver ce nombre optimal.

Cependant, dans notre cas de figure, nous avons besoin de comparer la pertinence de classification entre deux valeurs de K uniquement : 1 et 2. Cela permet de décider si un axe A_i est sélectionné pour prendre part au partitionnement ou pas.

Le critère d'information Bayesien BIC (*Bayesian Information Criterion*), dit critère de Schwarz aussi, permet d'équilibrer la vraisemblance logarithmique de telle façon que cette fonction n'est plus croissante. Il a été prouvé que ce critère est le plus performant pour évaluer la pertinence de modélisation de mélanges Gaussiens [41]. Nous avons donc sélectionné ce critère pour évaluer la pertinence de nos modèles.

Le critère BIC appliqué à un mélange de K Gaussiennes s'écrit :

$$\mathrm{BIC}(\mathcal{V}_K) \;= -2\mathcal{V}_K + K\log(n), \tag{III.10}$$

n étant le nombre d'individus de l'échantillon (du nœud).

- Si BIC($\mathcal{V}_2$) > BIC($\mathcal{V}_1$) alors la distribution est mieux représentée par un mélange de deux Gaussiennes, c'est-à-dire à l'aide de deux classes distinctes ; A_i est donc ajouté à la liste des axes sélectionnés ; $\mathcal{P}_i$ correspond au point d'intersection des deux Gaussiennes.
- Sinon (si BIC($\mathcal{V}_1$) $\leq$ BIC($\mathcal{V}_2$)), les données issues de la projection des individus de l'échantillon selon A_i sont considérées suffisamment homogènes. Ainsi, aucun hyperplan orthogonal à A_i n'est utilisé pour partitionner l'espace. Dans le cas d'un arbre non-contraint, $d_s \leftarrow d_s - 1$.

3.4.2 Adaptation de la méthode AP

L'application de l'algorithme AP (*cf.* section 2.3.2) permet de partager les données en deux ensembles disjoints $\mathcal{E}$ et S.

Nous appliquons la méthode AP aux données projetées selon A_i, donc dans un espace de dimension 1 à chaque fois.

De même que dans la section 3.4.1, il nous importe de savoir si le partitionnement selon un axe A_i donné est pertinent ou pas. Nous réutilisons donc le critère BIC pour évaluer la cohésion du modèle avant et après le bi-partitionnement selon A_i.

Pour tous les partitionneurs basés sur le calcul des centroïdes, comme par exemple AP ou K-means, nous utilisons la variance pour le calcul de la fonction de vraisemblance (au lieu de la vraisemblance logarithmique dans le cas de EM). Cette fonction intervient dans le calcul du critère BIC.

La variance σ^2 est donnée par $\sigma^2 = V_{intra} + V_{inter}$. K étant le nombre de classes, $\overline{c}$ le barycentre de la population, n_i le nombre d'éléments dans la classe i, c_i son centroïde et σ_i^2 sa variance intra-classe, on a [3] :

$$V_{intra} = \sum_{i=1}^{K} \frac{n_i}{n} \sigma_i^2 \quad \text{et} \quad V_{inter} = \sum_{i=1}^{K} \frac{n_i}{n} (c_i - \overline{c})^2. \tag{III.11}$$

Ainsi, pour une partition de K classes, le critère BIC est donné par :

$$\text{BIC} = 2\log(\sigma^2) + K\log(n). \tag{III.12}$$

Ce critère permet donc de décider si un axe A_i intervient dans le processus de bi-partitionnement.

Notons que la variance est une mesure statique dont les calculs ont peu de sens sur un échantillon réduit. Comme le nombre minimum d'individus pour que la variance soit significative n'a jamais été défini, nous introduisons un paramètre N_{min} représentant le

3. Nous pouvons constater que, pour $K = 1$, on $\sigma^2 = V_{intra}$.

nombre minimum d'individus par classes. Ce paramètre est donné de façon approximative et est à prédéfinir par l'utilisateur. Ainsi, si cardinal(Ω) $< N_{min}$ ou cardinal($\mathcal{S}$) $< N_{min}$ alors A_i n'est pas sélectionné pour contribuer à la subdivision du nœud. Notons bien que ce paramètre est optionnel. De façon générale, nous lui affectons une valeur empirique de $N_{min} = 30$, et ce quelque soit la base de données.

3.4.3 Choix de A_s dans le cadre d'un arbre binaire

Dans un arbre binaire, si A_i n'est pas sélectionné, autrement dit, si BIC($\mathcal{V}_1$) $\geq$ BIC($\mathcal{V}_2$), le processus de partitionnement et comparaison est répété en projetant les données selon l'axe A_{i+1}. Si aucun axe n'est sélectionné, le nœud n'est pas découpé : il s'agit d'une feuille.

3.5 Exemples

Nous détaillons dans cette section l'une des variantes de notre méthode de classification : appliquer une ACP, au préalable, sur l'ensemble des données qui sont ensuite partitionnées en employant un *Kd-tree* et l'algorithme EM.

Algorithme Il s'agit d'appliquer les étapes suivantes à chaque nœud de l'arbre.

1. Projeter tous les individus du nœud dans l'espace engendré par l'ACP. Soit d la dimension de cet espace. Initialement, $i \leftarrow 1$.
2. Projeter les individus selon l'axe A_i. Les données sont ainsi représentées dans un espace de dimension 1.
3. L'algorithme EM est appliqué avec, respectivement 1 et 2 Gaussiennes. Leurs vraisemblances respectives sont comparées.
   - Si les données sont mieux représentées avec une Gaussienne, alors :
     - $i \leftarrow i + 1$;
     - si $i \leq d$ alors reprendre l'algorithme en 2 ; sinon le nœud courant est une feuille, reprendre alors l'algorithme en 1 sur un autre nœud du même niveau de l'arbre.
   - Sinon, si les individus sont mieux représentés avec 2 Gaussiennes alors :
     - un nouveau point de partitionnement $\mathcal{P}_i$ relatif à A_i est calculé ;
     - le nœud courant est partitionné en deux nœuds fils.
4. Appliquer récursivement le processus aux nœuds fils.

L'algorithme ci-dessus est illustré par la figure III.9. Nous y présentons les courbes de projections des données au niveau de quelques nœuds. L'algorithme EM détermine, à partir de chacune de ces courbes, le nombre optimal de Gaussiennes selon lesquelles les

données sont réparties ; la décision de découpage selon un axe donné étant intrinsèque à ce nombre.

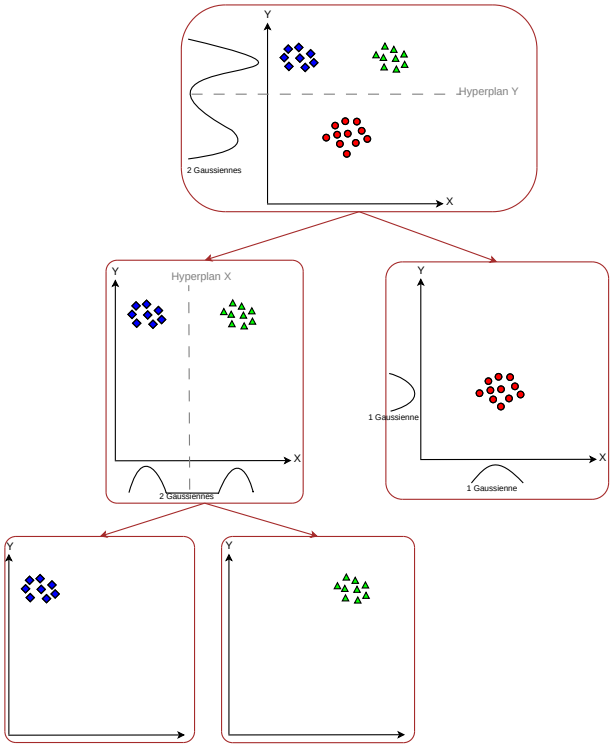


FIGURE III.9 – Illustration de l'algorithme 1 dans un espace de dimension 2 (ACP appliquée au préalable)

Le même procédé en remplaçant le *Kd-tree* par un arbre non-contraint est illustré par la figure III.10.

3.6 Conclusion

Nous avons présenté dans cette section une méthode de classification hiérarchique inspirée de plusieurs approches préexistantes comme les techniques par partitionnement de la principale direction (PDDP), les méthodes probabilistes, *etc.*

Notre approche repose sur trois concepts complémentaires : un arbre permettant de partitionner les données de façon hiérarchique, un projecteur qui génère un nouvel espace permettant de représenter les données de manière optimale et un partitionneur qui sélectionne les axes les plus discriminant vis-à-vis de la séparation des données et qui calcule les hyperplans qui leur sont associés.

Les principaux avantages de l'approche (ACPP) proposée sont sa généricité et sa rapidité. Par généricité, nous entendons indépendance de tout paramètre abstrait comme

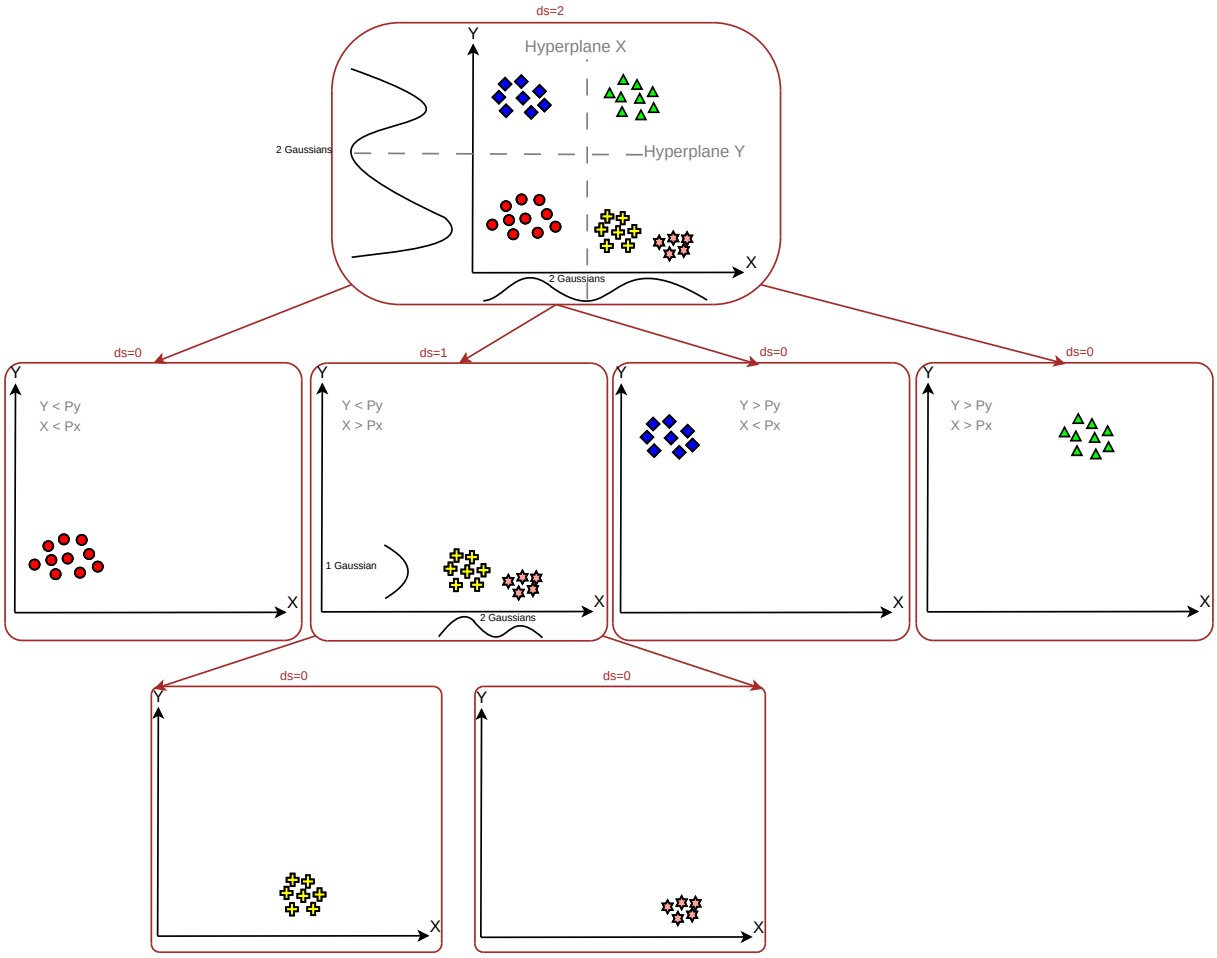


FIGURE III.10 – Illustration de l'algorithme 1, en employant un arbre non-contraint, $d = 2$ (sans projecteur)

la forme des classes ou leurs rayons respectifs. En effet, il est possible d'appliquer cette méthode sans aucune intervention de l'utilisateur.

La méthodologie ACPP inclut plusieurs variantes dont le point de divergence porte sur la nature de l'arbre employé, le type de l'analyseur-projecteur ou le choix du partitionneur. Les expérimentations présentées dans la section 4 permettent d'évaluer les performances de chacune de ces variantes.

4 Résultats de classification

Comme nous l'avons mentionné précédemment, la classification est utile dans différents cadres d'applications sur une infinité de bases de données. Nous en avons sélectionné quelques unes des plus proches de notre domaine d'activité à savoir le traitement des images de documents imprimés et la quantification de ces images.

La base BLidm0 de Baird [7] est composée d'un ensemble d'images de caractères nu-

mériques affectés par différents types de déformations. Étant donné que cette base est étiquetée, nous l'avons sélectionnée afin de pouvoir évaluer les résultats quantitativement. Nous utiliserons donc ce corpus d'images pour la validation de notre moteur de classification.

Les résultats obtenus seront comparés à ceux relatifs aux approches de partitionnement les plus liées à notre méthode comme k-means et Mean-shift.

Afin de faciliter la comparaison de nos résultats par rapport à d'autres approches, nous avons testé notre moteur de classification sur un ensemble varié de bases disponibles en lignes (*UCI Machine Learning Repository*). Les performances atteintes *via* différentes variantes de ACPP sont présentées et discutées ci-après.

4.1 Nomenclature

Selon que l'ACP est appliquée au préalable (globalement sur l'ensemble des données une seule fois) ou en cascade (au niveau de tous les nœuds), que l'arbre adopté est de type *Kd-tree* ou 'non-contraint' et selon la méthode de partitionnement appliquée au niveau des nœuds, nous distinguons plusieurs variantes de notre approche. Le tableau III.1 présente les différentes appellations que nous emploierons pour désigner ces différentes variantes.

Appellation	Type de l'arbre	Application de l'ACP	Méthode de partitionnement
KdGlobal	*Kd-tree*	au préalable	
KdGlobalAP	*Kd-tree*	au préalable	AP
KdGlobalEM	*Kd-tree*	au préalable	EM
KdGlobal2means	*Kd-tree*	au préalable	2-means
KdCascade	*Kd-tree*	en cascade	
KdCascadeAP	*Kd-tree*	en cascade	AP
KdCascadeEM	*Kd-tree*	en cascade	EM
KdCascade2means	*Kd-tree*	en cascade	2-means
NcGlobal	non-contraint	au préalable	
NcGlobalAP	non-contraint	au préalable	AP
NcGlobalEM	non-contraint	au préalable	EM
NcGlobal2means	non-contraint	au préalable	2-means
NcCascade	non-contraint	en cascade	
NcCascadeAP	non-contraint	en cascade	AP
NcCascadeEM	non-contraint	en cascade	EM
NcCascade2means	non-contraint	en cascade	2-means

TABLE III.1 – Nomenclature des variantes de ACPP

4.2 Mesures d'évaluation

Nombre de mesures ont été élaborées afin de mesurer la qualité d'une partition donnée. Ces métriques permettent de comparer la qualité de la partition créée par un algorithme de classification donné par rapport à une partition modèle connue *a priori*[4]. Une telle évaluation est dite 'évaluation externe'.

La pureté, l'entropie, la mesure-F et NMI (*Normalized Mutual Information*) sont les critères d'évaluation les plus couramment utilisées dans les références bibliographiques. Ces mesures sont définies ci-dessous.

Soient $\Omega = \{\omega_1, \omega_2, ..., \omega_K\}$ la partition modèle (étiquetée), $\mathcal{C} = \{C_1, C_2, ..., C_m\}$ la partition à évaluer et N le nombre d'individus de la population.

4.2.1 Pureté

La pureté est une mesure simple et transparente qui reflète la précision de la classification. Sa formule est donnée par :

$$\text{Pureté}(\mathcal{C}) = \frac{1}{N} \sum_i \max_j [\text{card}(\omega_j \cap C_i)]. \tag{III.13}$$

Dans l'exemple de la figure III.11 la pureté vaut : $\frac{1}{21}(4 + 4 + 1 + 5)$.

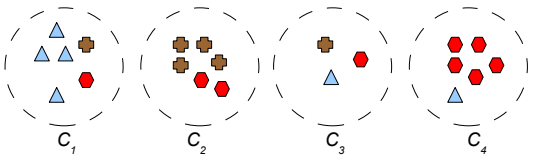


FIGURE III.11 – Exemple de partition à évaluer

Nous considérons que cette métrique n'est pas objective dans la mesure où elle favorise les partitions où le nombre de classes m est élevé. En l'occurrence, la pureté atteint la valeur maximale de 1 sur une partition de N classes, comptant un seul individu chacune.

4. Les mesures utilisées pour évaluer un algorithme de classement comme la matrice de confusion, la précision et le rappel ne sont pas exploitables dans le cadre d'une classification car certains critères comme le nombre de classes produites, leurs étiquettes respectives,...) ne peuvent pas être prédéterminés

4.2.2 Entropie

De même que la pureté, l'entropie permet d'évaluer l'homogénéité des classes formant une partition :

$$\text{Entropie}(\mathcal{C}) = 1 + \frac{1}{N}\sum_{i=1}^{m}\sum_{j}^{K} \text{card}(\omega_j \cap C_i)\ \log_K\left(\frac{\text{card}(\omega_j \cap C_i)}{\text{card}(C_i)}\right). \tag{III.14}$$

Dans l'exemple de la figure précédente l'entropie vaut : $1 + \frac{1}{21}[(4\log_3(\frac{4}{6}) + 1\log_3(\frac{1}{6}) + 1\log_3(\frac{1}{6})) + (4\log_3(\frac{4}{6}) + 2\log_3(\frac{2}{6})) + (1\log_3(\frac{1}{3}) + 1\log_3(\frac{1}{3}) + 1\log_3(\frac{1}{3})) + (5\log_3(\frac{5}{6}) + 1\log_3(\frac{1}{6}))]$.

Cette mesure présente également l'inconvénient de favoriser les partitions formées par de nombreuses classes.

4.2.3 Mesure-F

Les mesures-F sont des métriques basées sur la combinaison de la précision et du rappel. Parmi cette catégorie de critères, la mesure-F1 associe des poids égaux à la précision et au rappel :

$$\text{F1}(\mathcal{C}) = \frac{1}{N}\sum_{j}^{K} \text{card}(\omega_j)\ \max_i \frac{2\ \text{card}(\omega_j \cap C_i)}{\text{card}(\omega_j) + \text{card}(\mathcal{C}_i)}. \tag{III.15}$$

Contrairement aux deux mesures précédentes, F1 n'est pas biaisée par les classes de taille réduite.

4.2.4 NMI

L'information mutuelle (NMI) est reconnue comme étant la mesure la mieux fondée sur le plan théorique.

$$\text{NMI}(\mathcal{C}) = \frac{2}{N}\sum_{i=1}^{m}\sum_{j}^{K} \text{card}(\omega_j \cap C_i)\ \log_{K\times m}\left(\frac{\text{card}(\omega_j \cap C_i) \times N}{\text{card}(C_i) \times \text{card}(\omega_j)}\right). \tag{III.16}$$

Les valeurs données par les quatre mesures définies ci-dessus sont dans l'intervalle $[0, 1]$. Ces valeurs sont proportionnelles à la qualité de la classification.

Nous emploierons ces différentes mesures pour évaluer la qualité de chacune des partitions induites par nos expérimentations.

4.3 Validation par la base BLidm0

Nous présentons, dans cette section, les résultats de classification sur la base BLidm0 obtenus avec les différentes variantes de notre système de classification. Ces résultats sont

évalués en employant des critères complémentaires définis dans la section précédente. Nous comparerons ensuite ces performances avec les méthodes *k-means*, *I-k-means* et *Mean-Shift* respectivement.

4.3.1 Présentation de la base de données

La base BLidm0 (*Bell Labs image defect model database, version 0*) est composée de 8 565 750 images binaires. Ces imagettes, de taille variant autour de 28 × 28 pixels, représentent des caractères isolés affectés par différents modèles de déformations comme par exemple le lissage, de redimensionnement, la rotation, *etc.*

La figure III.12 présente un échantillon représentatif des caractères numériques de cette base.

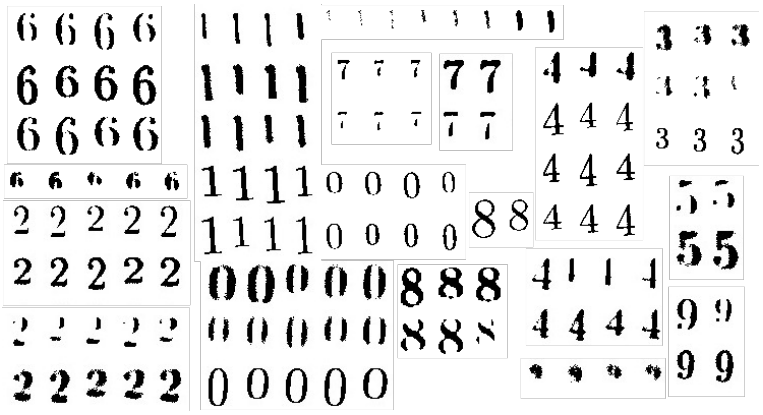


FIGURE III.12 – Échantillon de la base BLidm0

Cette base a été conçue pour mesurer les performances des classificateurs, caractériser la qualité des images de document et l'élaboration d'un classificateur de haute performance.

Nous avons sélectionné cette base pour sa maniabilité : cet ensemble d'images peut être partitionné selon différents critères comme la nature de la déformation, la valeur des paramètres associés à chaque déformation, le code Ascii du caractère représenté, la police de caractères employée, *etc.* Un partitionnement hiérarchique est le mieux placé pour prendre en compte le maximum de critères.

Comme il n'est pas facile d'évaluer la classification en tenant compte de tous ces critères conjointement, nous avons choisi de mettre en avant le critère de séparation selon le code Ascii lors de l'analyse des résultats.

Pour que les tests restent simples mais représentatifs, nous avons restreint nos expérimentations aux caractères numériques de cette base. Par ailleurs, certains modèles de déformation réduisent le caractère à une forme ponctuelle non reconnaissable même à

l'œil nu. Nous avons donc éliminé ce type d'images de notre base de test. En définitive, notre base de validation compte 71 484 imagettes de caractères numériques (de 0 à 9).

Notre base est donc composée d'images de chiffres affectées par des rotations de différents angles (ne dépassant pas 30°), des translations, des homothéties, des étirements et des flous Gaussien d'écart type variant entre 0 et 1. Les caractères de notre base sont écrits avec la même police de caractères.

4.3.2 Résultats et comparaisons

Pour toutes les expérimentations présentées dans cette section, tout individu est représenté par une matrice de pixels de taille 16×16 correspondant aux valeurs de luminosité (entre 0 et 255).

ACP en cascade

Arbre binaire Le tableau III.2 présente les résultats de classification obtenus en employant un arbre de type *Kd-tree* et en appliquant une ACP au niveau de chaque nœud pour choisir l'axe A_s. Le partitionnement est effectué en employant 3 approches différentes : EM, AP et 2-means respectivement. Pour toute expérimentation, la partition évaluée correspond à celle formée par les feuilles de l'arbre.

Méthode	NMI	F1	Entropie	Pureté	Nombre de classes
KdCascadeEM	0.896	0.901	0.914	0.923	10
KdCascadeAP	0.978	0.989	0.978	0.989	10
KdCascade2means	**0.985**	**0.993**	**0.985**	**0.993**	11

TABLE III.2 – Résultats de 3 variantes de KdCascade

Arbre non contraint Le tableau III.3 affiche les résultats de classification obtenus en employant un arbre non-contraint dont chaque nœud peut être découpé selon 2 axes au plus. Il s'agit des axes de poids le plus fort d'après les résultats de l'ACP qui est a été récursivement appliquée au niveau de tous les nœuds.

Nous n'avons pas utilisé tous les axes pour le partitionnement des nœuds, conformément à la définition d'un arbre non-contraint, afin d'éviter l'explosion du nombre de classes. En effet, à partir de $d_s = 3$ nous avons $2^3 = 8$ classes au premier niveau de l'arbre !

Comme l'attestent les valeurs du tableau ci-dessus, même avec $d_s = 2$ le nombre de classes est trop grand (entre 14 et 16) d'où la baisse des valeurs de NMI et F1. En revanche, l'emploi de ce type d'arbre permet, généralement d'atteindre des valeurs de pureté plus élevées.

Méthode	NMI	F1	Entropie	Pureté	Nombre de classes
NcCascadeEM	0.888	0.943	0.979	0.988	14
NcCascadeAP	0.881	**0.952**	0.946	0.952	15
NcCascade2means	**0.899**	0.927	**0.991**	**0.996**	16

TABLE III.3 – Résultats de classification de 3 variantes de NcCascade

Par ailleurs, cette sur-segmentation est également due à la forme des classes dans l'espace de caractéristiques. Elle peut être corrigée par une phase de fusion des classes adjacentes par des méthodes de connectivités par linkage des classes proches.

ACP préalable La figure III.13 présente les résultats de classification de quelques variantes de l'algorithme KdGlobal et ce dans différents espaces de représentation. La dimension de l'espace est directement liée aux taux d'information retenue à l'issue de l'ACP. Cet espace est commun à tous les nœuds de l'arbre. Autrement dit, l'ACP est appliqué au préalable de la classification sur l'ensemble de la population.

Chacune des courbes de cette figure correspond aux résultats d'évaluation des partitions, formées par les feuilles, en se basant sur un critère différent.

L'observation des courbes ci-dessus permet de constater que les 3 méthodes de partitionnement mènent à des résultats assez similaire. Plus précisément, l'emploi de 2-means permet d'atteindre de meilleures performances. Cette dernière affirmation n'est toutefois pas généralisable : d'autres tests en employant des caractéristiques différentes révèlent que la méthode AP ou EM permet d'atteindre de meilleurs résultats.

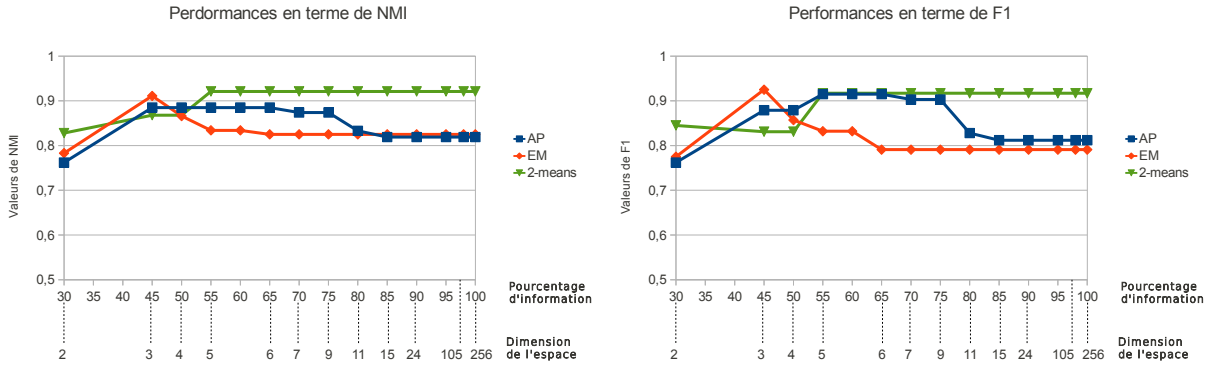
Le PDDP [9] original est adapté au clustering de données dans des espaces de grandes dimensions. Notre proposition qui généralise le PDDP est aussi efficace sur les données en grande dimension comme le montre la figure III.13.

L'application de l'ACP de façon globale et unique donne lieu à un algorithme de classification extrêmement rapide tout en assurant de bonnes performances. Cependant, l'application de l'ACP en cascade est presque toujours plus performante que son application au préalable. Ainsi, l'utilisateur peut choisir d'appliquer l'ACP en cascade ou de façon globale en fonction de ses exigences.

Comparaisons Nous comparons dans ce paragraphe les résultats de classification relatifs à ACPP à ceux assurés par les algorithmes de classification les plus populaires et les plus proches de notre proposition.

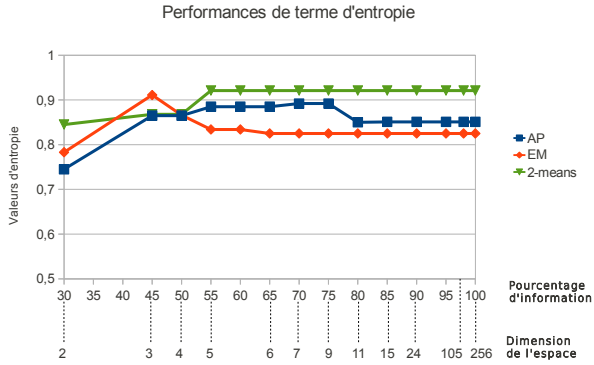
Pour chacune de ces méthodes, nous avons tâché de sélectionner les paramètres optimaux qui procurent les meilleures performances.

- **k-means** Sachant que le nombre optimal de classes est 10, nous avons fixé k à

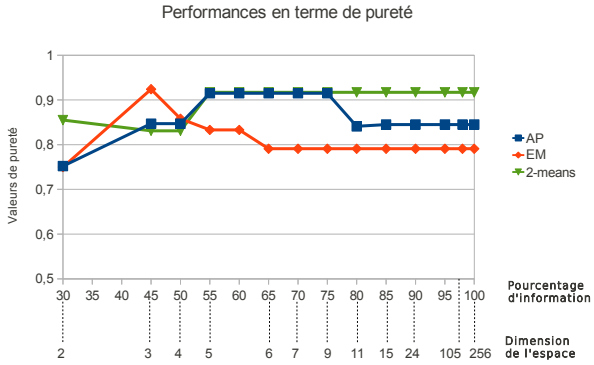


(a) NMI

(b) F1



(c) Entropie



(d) Pureté

FIGURE III.13 – Résultats de classification de KdGlobal dans des espaces de dimensions différentes

10 pour tous les tests relatifs à cet algorithme. Nous avons également initialisé l'algorithme avec des individus représentatifs de leurs classes respectives : les 10 prototypes initiaux présentent 10 étiquettes différentes.

- **k-means Intelligent** Cette variante de k-means présente un point fort très attractif et si rare : l'indépendance de tout paramètre. Cet algorithme ne requiert donc aucune connaissance *a priori* ; il est complètement automatisée.
- **Mean-shift** Nous rappelons que cette approche dépend de 2 paramètres : σ pour le noyau et le facteur d'échelle. N'ayant aucune idée sur les valeurs optimales de ces paramètres, nous avons procédé par leur recherche par dichotomie dans une base réduite. Suite à de nombreux tests, nous avons sélectionné les valeurs de 100 000 et 100 000 pour σ et h respectivement dans l'espace d'origine (de dimension 256). Dans les espaces où une ACP a été appliquée σ et h valent respectivement 5 et 0.1 !

KdCascade Le tableau III.4 permet de comparer les performances de 4 méthodes de classification en s'appuyant sur 2 critères complémentaire [5]. Pour ce lot de tests, chacune des méthodes concurrentes a été testée sur les données d'origine (sans ACP). Nous comparons ces approches à la variante KdCascade2means de notre méthode (présentée dans le tableau III.2).

Critère	10-means	k-means intelligent	Mean-shift	KdCascade2means
NMI	0.937	0.685	0.712	**0.985**
Pureté	0.901	0.804	**0.999**	0.993
Nombre de classes	10	22	59	11

TABLE III.4 – Comparaison de KdCascade2means à 3 autres méthodes de classification appliquées dans un espace de dimension 256 non modifié par l'ACP

Ce dernier tableau montre que l'approche proposée permet d'atteindre les meilleurs résultats globalement. Mean-shift produit des classes plus pures encore mais au détriment des valeurs de NMI. En effet, cet algorithme donne lieu à une soixantaine de classes.

En termes de temps d'exécution, Mean-shift est l'algorithme le plus lent suivi de notre approche (en raison des calculs récursifs de l'ACP) suivi de k-means Intelligent puis k-means.

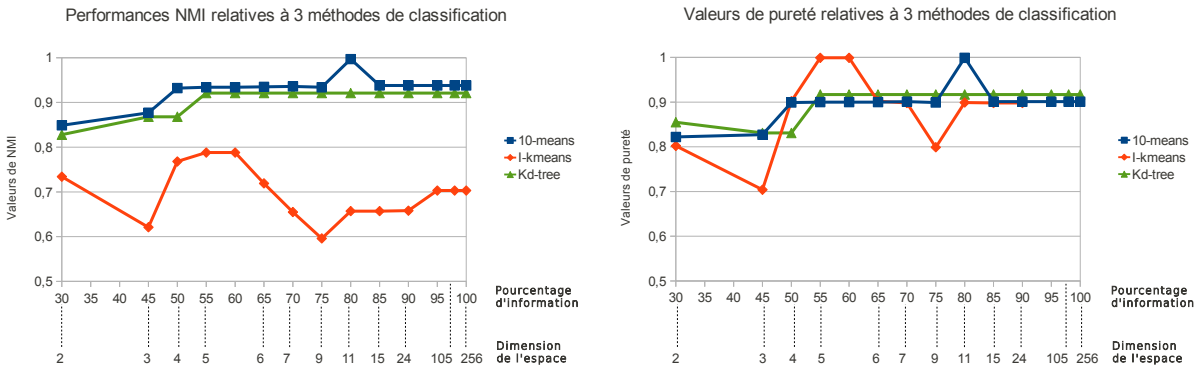
KdGlobal Les courbes III.14 ci-après illustrent la variation de performance des 3 approches concurrentes en fonction de la dimension de l'espace (définie par le taux d'information retenue suite à l'exécution d'une ACP). Nous pouvons remarquer que notre approche ainsi que les deux variantes de la méthode k-means sont robustes vis-à-vis de ces variations, sur cette base de données.

Les variantes KdGlobal de ACPP testées dans ces dernières expérimentations ne requiert pas des temps de calcul considérables. En l'occurrence, notre approche est la plus rapide. Qui plus est, elle permet de concilier performance et rapidité.

Les imagettes de la figure III.15 correspondent au 9 premiers vecteurs propres calculés par l'ACP (calculée au préalable sur l'ensemble des données). Les valeurs propres, normalisées, correspondantes varient de $\lambda_1 = 0.253$ à $\lambda_9 = 0.026$; $\sum_{i=1}^{9} \lambda_i \simeq 0.76$, c'est-à-dire qu'en se restreignant à 76% de l'information, nous obtenons un espace de dimension 9. Cela signifie que toute l'information est portée par les premiers axes fournis par l'ACP !

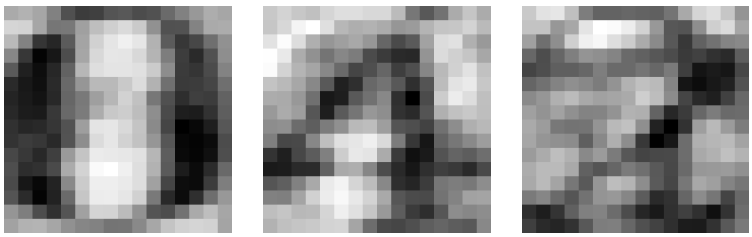
Dans cette figure, les points les plus sombres correspondent aux poids les plus forts ; Les pixels blancs de ces images n'apportent donc aucune information.

5. Désormais nous nous contenterons de présenter les valeurs de NMI et de pureté car ces dernières présentent une redondance avec F1 et l'entropie respectivement.

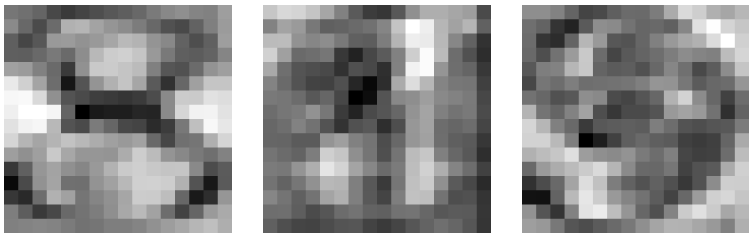


(a) NMI (b) Pureté

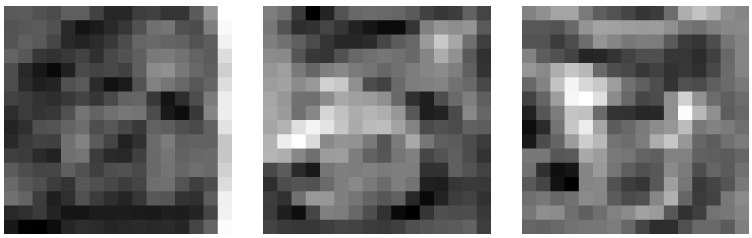
FIGURE III.14 – Résultats de classification de 4 approches en fonction de la dimension de l'espace



(a) $\lambda_1 = 0.253$ (b) $\lambda_2 = 0.123$ (c) $\lambda_3 = 0.094$



(d) $\lambda_4 = 0.076$ (e) $\lambda_5 = 0.069$ (f) $\lambda_6 = 0.043$



(g) $\lambda_7 = 0.041$ (h) $\lambda_8 = 0.039$ (i) $\lambda_9 = 0.026$

FIGURE III.15 – Les 9 premiers vecteurs propres

4.3.3 Conclusion

Nous avons présenté dans cette section des résultats quantitatifs permettant d'évaluer la méthode de classification proposée ainsi que de la comparer à d'autres approches.

Les expérimentations montrent que notre approche permet d'assurer le meilleur compromis entre la qualité de la classification et le temps d'exécution. Cette approche est, de

surcroît, indépendante de tout paramètre abstrait et capable de traiter des données dans des espaces de grandes dimensions.

4.4 Validation par des bases UCI

La plupart des bases de données du répertoire UCI [31] sont conçues pour tester des méthodes de classement. Il existe 9 bases prévues pour les applications de classification. Cependant, ces ensembles ne sont pas toujours étiquetés et le type de données utilisées n'est pas approprié à notre approche (attributs catégoriques, données séquentielles, *etc.*).

Ainsi, nous avons choisi des bases de la catégorie classement pour évaluer notre moteur de classement. Celles-ci ont été sélectionnées pour leur leurs tailles (les plus grandes) et le nombre de classes qu'elles comptent, les partitions formées d'un nombre réduit de classes ou d'individus n'étant pas appropriées à notre méthode.

4.4.1 Présentation des bases

Les bases sélectionnées relèvent de différents domaines d'application.

- *Iris* est une base particulièrement populaire composée de 150 observations qui correspondent à 3 types de fleurs.
- *Image Segmentation Data Set* est composée de 2310 images naturelles segmentée manuellement en 7 types de régions.
- *Statlog Landsat Satellite Data Set* est un ensemble de 6435 images satellite générées par la NASA. Les caractéristiques qui lui sont associées visent à différentier 7 types de sol.
- *ISOLET* est un ensemble de 7797 lettres prononcées par 150 personnes différentes. Cette base est caractérisée par un grand nombre de descripteurs : 617.
- *Letter Recognition Data Set* est une base de 20000 images de caractères alphabétiques écrits en majuscule et caractérisées par des descripteurs géométriques.

Le tableau III.5 affiche les principales propriétés de ces bases.

Base de donnée	Nombre d'individus	Nombre de descripteurs	Nombre de classes
Iris	150	4	3
Image Segmentation	2310	19	7
Landsat Satellite	6435	36	6
ISOLET	7797	617	26
Letter Recognition	20000	16	26

TABLE III.5 – Échantillon de bases de données du répertoire UCI

4.4.2 Résultats et comparaisons

Pour des raisons de lisibilité, nous n'encombrons pas cette section avec les résultats relatifs à toutes les variantes de ACPP ; nous nous focalisons sur l'une de ces variantes dans chaque paragraphe.

Par ailleurs, nous ne présenterons plus les résultats relatifs à l'algorithme Mean-shift en raison de la complexité inhérente à l'estimation de ses paramètres.

Approche	NMI	Pureté	Nombre de classes
Ik-means	0.715	0.880	4
3-means	0.751	0.893	3
KdCascadeAP	**0.798**	0.920	3
PDDP [9]	0.740	**0.933**	4

TABLE III.6 – Résultats obtenus sur la base *Iris*

Base *Iris* Les résultats présentés dans le tableau III.6 révèlent que les classes de la base *Iris* sont linéairement séparables. En effet, nous obtenons des résultats très satisfaisants en l'absence de toute supervision.

Par ailleurs, en appliquant une ACP au préalable et en gardant 90% de l'information, la dimension de l'espace passe à 1. Dans cet espace, nous obtenons exactement les mêmes résultats avec KdGlobalAP que KdCascadeAP, c'est-à-dire NMI=0.798 et une pureté de 92% ! Ainsi, une simple projection au préalable suffit dans de tels cas de figure.

Exceptionnellement pour la base *Iris*, nous avons pu comparer notre approche par rapport à PDDP (les résultats sont accessibles en ligne) [6]. Cette approche permet de former des classes plus pures puisque, disposant d'un critère d'arrêt prédéfini par l'utilisateur, à chaque niveau de l'arbre, le nœud à partitionner est sélectionné en comparant une mesure d'homogénéité interne entre tous les nœuds du niveau ; mais cela peut être parfois coûteux.

Base *Image Segmentation* Le tableau III.7 présente les résultats obtenus avec quelques variantes de ACPP et permet de les comparer à k-means et k-means intelligent respectivement. Nous pouvons en déduire que, sur cette base, certaines variantes de ACPP assurent le meilleur compromis entre le nombre de classes et leur pureté.

Notons, par ailleurs, que l'emploi d'un arbre non-contraint permet d'améliorer la pureté au détriment du nombre de classes. En effet, le tableau III.8 confirme que, indépen-

6. Nous rappelons que ACPP est une généralisation de l'algorithme PDDP

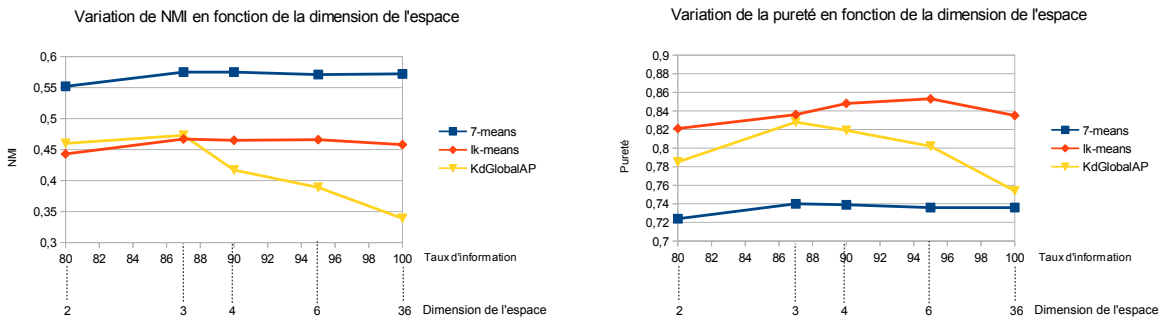
Approche	NMI	Pureté	Nombre de classes
Ik-means	0.444	0.739	32
7-means	0.455	0.568	7
KdCascadeAP	**0.548**	0.734	18
NcCascade2means	0.510	**0.828**	69
KdGlobalAP	0.503	0.753	32

TABLE III.7 – Résultats obtenus sur la base *Image Segmentation*

damment du partitionneur employé, il est possible d'améliorer la pureté avec un nombre[7] d'axes $d_s > 1$.

Partitionneur \ $d_s \leq$		1	2	3
AP	NMI	**0.548**	0.518	0.447
	Pureté	0.734	0.809	0.768
2-means	NMI	0.510	0.510	0.495
	Pureté	0.732	0.828	**0.870**

TABLE III.8 – Variation des performances de NcCascade en fonctions des valeurs maximales de d_s autorisées dans un arbre non-contraint (base : *Image Segmentation*)



(a) NMI (b) Pureté

FIGURE III.16 – Résultats de classification en fonction de la dimension de l'espace, sur la base *Landsat*

Base *Landsat Satellite* Les résultats d'expérimentations présentés dans la figure III.16 correspondent aux variation de performances de classification en fonction du taux d'information retenue après l'ACP.

7. Nous considérons qu'à partir de $d_s = 5$, le nombre de fils dans l'arbre explose (32 fils au premier niveau de l'arbre, jusqu'à 1024 au second, *etc.*). Nous évitons ainsi les tests au delà de cette valeur.

Dimension	2	3	4	6	36
Nombre de classes obtenues avec Ik-means	44	46	48	48	48
Nombre de classes obtenues avec KdGlobalAP	92	77	64	34	26

TABLE III.9 – Nombre de classes produites en fonction de la dimension de l'espace, sur la base *Landsat*

Nous pouvons constater, à partir de la figure III.16 et du tableau III.9 que les performances de l'algorithmes k-means intelligent quasi-insensibles vis-à-vis des variations de dimension. En revanche, notre méthode est sensiblement plus efficace lorsque $N >> d$. En effet, cela permet une analyse plus profonde selon tous les axes tandis que, lorsque la dimension est relativement grande et que l'ACP n'est pas appliquée en cascade, la profondeur de l'arbre peut être inférieure à d ; dans ce cas, certains axes ne sont jamais pris en compte durant tout le processus de bi-partitionnement !

Approche	NMI	Pureté	Nombre de classes
Ik-means	0.681	0.484	19
26-means	**0.715**	0.573	26
KdCascadeAP	0.596	0.590	57
NcCascadeAP	0.580	**0.688**	184
KdCascade2means	0.601	0.591	56
NcCascade2means	0.577	0.675	175

TABLE III.10 – Résultats obtenus sur la base *ISOLET*

Base *ISOLET* La base *ISOLET* est notamment caractérisée par une dimension trop élevée par rapport au nombre d'individus qu'elle compte. Les résultats présentés dans le tableau III.10 confirment que notre approche est moins performante dans de tels cas de figure. Les résultats restent toutefois satisfaisants en comparaison par rapport aux autres approches.

Approche	NMI	Pureté	Nombre de classes
Ik-means	0.329	0.251	23
26-means	0.351	0.282	26
KdCascadeAP, $N_{min} = 200$	0.328	0.302	63
KdCascadeFisher	0.457	0.549	273
NcCascadeFisher	**0.464**	**0.605**	460

TABLE III.11 – Résultats obtenus sur la base *Letter Recognition*

Base *Letter Recognition* Le tableau III.11 montre des résultats de classification médiocres, et ce pour toutes les approches testées. Ceci est prévisible dans la mesure où cette base, ainsi que celles étudiées dans les paragraphes précédents, sont conçues pour des applications de reconnaissance. Une base de connaissance (apprentissage) est donc nécessaire pour atteindre de hautes performances dans ce cas.

4.4.3 Conclusion

Dans cette section, nous avons évalué notre moteur de classification sur quatre bases de données du dossier UCI. Cela nous a permis de souligner les points forts et les points faible du ACPP.

Ces dernières expérimentations ont été élaborées sur des bases conçues pour des applications de reconnaissance. En revanche, des utilisations plus conformes au principe de classification seront présentées dans le prochain chapitre.

4.5 Conclusion et perspectives

Nous avons présenté dans cette section une nouvelle méthode de classification dont les principaux avantages sont :

- l'indépendance de tout paramètre abstrait (comme par exemple un noyau),
- la rapidité, notamment si l'ACP est appliquée en amont de la classification,
- l'efficacité qui fut prouvée via les expérimentations effectuées sur différents types de données.

Pour l'ensemble des expérimentations effectuées, nous avons employé l'ACP comme analyseur-projecteur. Ce dernier est certes incapable de calculer l'espace de représentation optimal pour tous les jeux de données. Nous envisageons ainsi de mesurer l'impact des projecteurs non linéaires (exemple : Kernel-PCA) sur les performances de notre approche et de mettre en place de nouvelles techniques d'analyse plus intelligentes basées sur une projection sélective.

Nous présenterons, dans la prochaine section, une nouvelle façon d'exploiter notre approche de classification. La principale nouveauté sera d'assurer une supervision plus assumée, ce qui nous permettra de remplacer l'ACP par la LDA.

5 Extension du moteur de classification : moteur de classement

Partant du constat que les feuilles de notre arbre de classification sont souvent pures, nous avons présumé qu'il suffirait d'étiqueter ces éléments pour superviser la classification de nouveaux candidats.

À partir de là, nous avons adapté les différentes composantes de notre algorithme de classification afin de l'appliquer en mode classement de façon idoine.

5.1 Méthodologie

Nous nous plaçons désormais dans le cadre d'une supervision assumée. En l'occurrence, le nombre de classes cibles ainsi que leurs étiquettes respectives sont connues *a priori*. Dans ce cas, nous ne parlons plus de classification mais de classement, voire de reconnaissance.

Une profusion de références bibliographiques, décrivant des approches de classement et d'apprentissage atteignant des performances élevées, existe la littérature. Ainsi, nous ne visons pas à en inventer une nouvelle approche mais à étendre le champ d'application de nos algorithmes de classification.

Nous proposons, en effet, une méthode de classement en cascade basée sur le même principe que l'approche de classification que nous avons adoptée. Le classement est basé sur un arbre (*Kd-tree* ou non-contraint) dit 'arbre de connaissances'. Il s'agit d'un arbre étiqueté qui peut être élaboré à partir de l'une des variantes de notre approche de classification.

L'insertion des individus de la base de test dans cet arbre permet de déterminer leurs étiquettes respectives.

5.1.1 Arbre de connaissances

L'arbre de connaissance, qui peut être un *Kd-tree* ou arbre non-contraint, est obtenu en classifiant des données étiquetées de la base de connaissance. La différence par rapport à l'approche de classification présentée précédemment réside dans cette connaissance des étiquettes qui va permettre de guider le partitionnement d'une part et de mieux choisir l'espace de représentation d'autre part.

Partitionnement Nous rappelons que notre approche consiste, entre autres, à partitionner chaque nœud (non terminal) de l'arbre en associant un point de partitionnement à un axe donné de l'espace. Dans le cadre d'une supervision candide, ce point est calculé par une méthode de partitionnement séparant les individus du nœud en deux sous-ensembles.

Dans le cadre d'un classement (supervision assumée), nous choisissons, simplement, le point qui maximise l'homogénéité de la partition résultante ; ce critère étant représenté par des mesures comme la pureté ou l'entropie. Pour ce faire, si n est le nombre d'individus dans un nœud donné, $n-1$ seuils permettent de séparer la population en deux classes. Nous calculons alors l'entropie de chacune des $n-1$ partitions et nous retenons celle dont l'entropie [8] est maximale.

Espace de représentation Disposant d'un ensemble de données étiquetées, nous avons la possibilité d'appliquer une LDA qui assure une représentation des données dans un espace optimal. Nous remplacerons donc l'ACP par la LDA que nous appliquerons en cascade (au niveau de tous les nœuds non-terminaux de l'arbre).

5.1.2 Classement

Un nœud de l'arbre de connaissance est considéré terminal si sa pureté (ou son entropie) atteint une valeur assez élevée. En l'occurrence, nous avons fixé cette valeur à 90% car un critère plus strict (une pureté de 100%) engendre des nœuds composés d'un ou deux éléments, ce qui entraîne un sur-apprentissage. Ainsi, à chaque feuille de l'arbre est associée l'étiquette la plus représentée parmi ses membres.

Pour classer un nouvel individu (dont l'étiquette est inconnue *a priori*), il suffit de l'insérer dans l'arbre de connaissance. Son étiquette estimée est celle représentée par la feuille où il aboutit.

5.2 Conclusion

Dans le cas où l'on dispose d'un ensemble de données étiquetées, il devient plus facile de superviser la classification. Dans cette optique, nous avons adapté notre algorithme de classification de manière à pouvoir l'exécuter en mode classement.

Des expérimentations permettant de valider ce mode de supervision seront présentées dans le prochain chapitre.

6 Conclusion

Nous avons présenté, dans ce chapitre, un moteur de classification opérationnel avec un degré de supervision flexible. Ce critère varie, en effet, de la supervision candide pour les

8. Nous avons choisi d'employer l'entropie pour évaluer la qualité des bi-partitionnement. Notons, toutefois, qu'il est possible de remplacer cette mesure par la pureté, la mesure-F, *etc.* , les résultats étant très similaires.

variantes de classification sans aucun critère d'arrêt à la supervision assumée la déclinaison de notre algorithme dédiée au classement.

L'approche proposée est de type hiérarchique, par partitionnements selon la principale direction ; chaque partitionnement étant effectué dans un espace de représentation optimal calculé à l'aide d'un projecteur.

Nous avons validé notre moteur de classification sur une base étiquetée de caractère numériques imprimés d'une part et sur des bases du repertoire UBCI d'autre. Les résultats sont généralement très satisfaisants et dépassent certains algorithmes très populaires comme Mean-Shift et k-means. De nouvelles applications, comme la classification de composantes connexes dans des images de manuscrits ou la quantification dans des photos naturelles seront présentées dans le prochain chapitre.

Ces résultats prometteurs nous ont incité à appliquer cette même approche, tout en l'adaptant, dans le cadre d'un apprentissage supervisé.

Pour toutes les expérimentations effectuées, nous avons employé des projecteurs linéaires (ACP ou LDA). Nous envisageons d'en tester de nouveaux : non-linéaires ou de nouvelles approches d'analyse-projection.

Nous présenterons, dans le prochain chapitre, de nouvelles applications à notre méthode de classification/classement. Ces fonctionnalités permettront d'alimenter la chaîne de traitement répondant au projet Mediabox.

Chapitre IV

Applications du ACPP

Table des matières

1 Introduction

La méthodologie ACPP offre un champ d'applications étendu, aussi bien en mode non-supervisé qu'en mode reconnaissance. Ainsi, le choix des applications à développer se doit d'être judicieux.

Nous avons pu évaluer notre méthode de classification quantitativement en l'appliquant à des bases étiquetées. Néanmoins, la classification est bien plus avantageuse lorsque les classes modèles ne sont pas connues a priori. Par exemple, il serait coûteux de prédéfinir les classes colorimétriques en amont d'une quantification d'une image naturelle. Par ailleurs, nous avons souvent besoin d'accomplir cette tâche (la quantification colorimétrique) de façon rapide. Pour toutes ces raisons, une approche semble appropriée pour cette application.

De même, il est difficile de disposer de connaissances *a priori* sur les caractères d'un manuscrit et les méthodes de reconnaissances de caractères les plus performantes échouent face à de tels corpus. Ainsi, afin de faciliter et accélérer leur transcription, nous proposons une classification de composantes connexes obtenus à partir d'une simple binarisation de l'image.

Dans le cadre d'une supervision assumée, nous évaluerons ACPP sur une base conséquente de caractères manuscrits (base MNIST) et nous comparerons les résultats à d'autres approches non-linéaires.

Nous réévaluerons également le classement de blocs de presse (présenté dans le chapitre II) en employant ACPP cette fois.

L'accumulation de preuves est l'un des principaux usages de ACPP. Cette application sera illustrée par l'exemple de reconnaissance de polices de caractères dans les images de presse. Cela se déroulera avec un niveau de supervision particulièrement faible.

2 Applications à la classification

Une page de document peut contenir des caractères alphanumériques, des figures, des tableaux, *etc.*. Il est difficile de prédéfinir les classes en amont de la classification. Les résultats seront donc présentés à titre illustratif.

De la même façon, l'évaluation des résultats d'une quantification colorimétrique d'une photo naturelle est très subjective dans la mesure où il difficile de définir un résultat modèle. Dans ce cas, nous n'évaluerons donc pas les performances de notre approche quantitativement mais afficherons quelques images issues de la quantification pour donner un aperçu des résultats.

2.1 Illustration avec la classification de connexités

2.1.1 Aide à la transcription des manuscrits

La transcription assistée par ordinateur utilise une classication des formes similaires de caractères pour réduire le temps de saisie manuelle à un seul exemple de chaque classe. Avec un taux de redondance de 90% de redondance de formes similaires, un ouvrage peut être saisi en quelques heures seulement.

C'est une application très exigeante sur les performances du classement des formes de caractères similaires. Le classement doit absolument produire des classes pures pour ne pas faire d'erreurs en propageant le même code ASCII à tous les caractères situés dans les mauvaises classes. Il ne doit pas créer un trop grand nombre de classes pour éviter la surcharge de travail de la saisie manuelle.

La classification de connexité présentée dans cette section facilite cette phase et la rend plus efficace en offrant au transcripteur les différentes classes de formes.

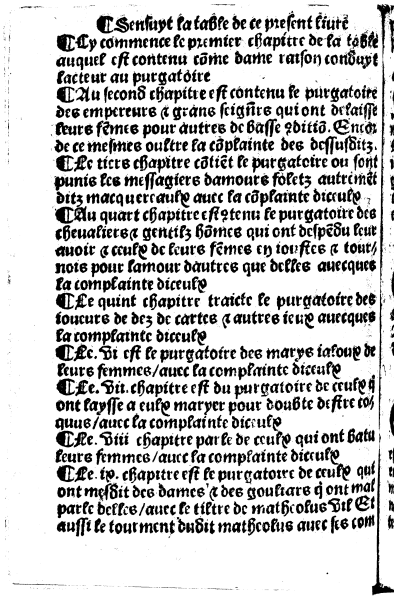
La figure IV.1 présente les résultats de classification sur une page d'incunable datant de la renaissance.

Après avoir éliminé les connexités de tailles trop grandes (resp. trop petites), 1074 composantes connexes sont candidates à la classification. Ces dernières sont représentées par un vecteur de caractéristiques composé par : les 4 profils (gauche, droit, haut, bas) ainsi que les deux histogrammes de projections verticales et horizontale. Chacun de ces six histogrammes est redimensionné à une taille fixe : 20 ; ce choix est empirique [1]. Ainsi

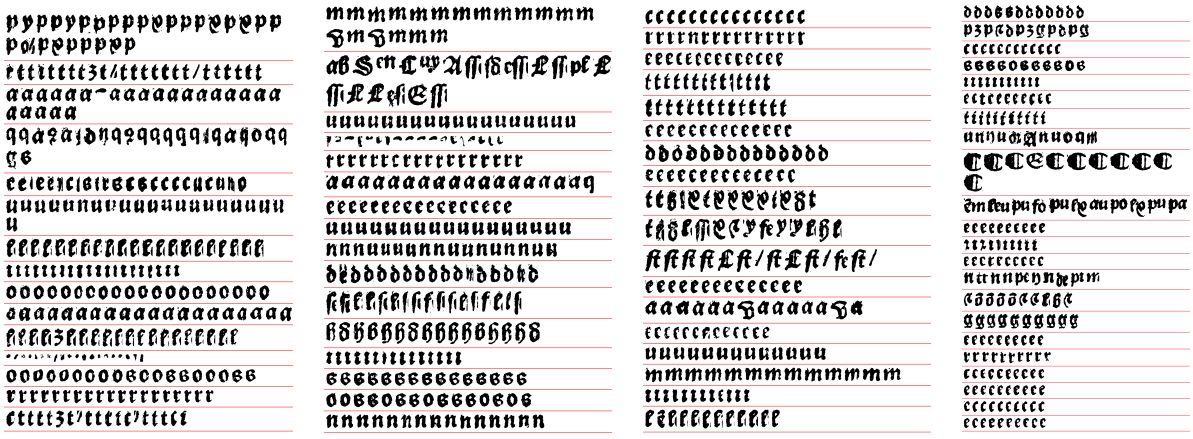
1. D'autres tailles d'histogramme ont été testée (16, 32, 36, *etc.* et cela n'affecte pas les résultats.

chaque individu est représenté par un descripteur de dimension 120.

La méthode KdCascade2means donne lieu à 71 classes dont 38 pures. Ainsi, il suffit de



(a) Image en entrée



(b) Résultats de classification : 71 classes dont 38 de pureté 100%

TABLE IV.1 – Résultats de classification sur un manuscrit médiéval en employant la méthode KdCascade2means

transcrire 38 caractères pour reconnaître 53% du texte de la page au moins. Notons que nous ne présentons pas un système de reconnaissance de caractères manuscrits élaborés mais une simple illustration de notre classificateur. En effet, nous n'avons pas sélectionné les descripteurs les mieux appropriés pour représenter ce type d'objets. Par ailleurs, nous n'avons pas effectué une segmentation en caractères mais une simple extraction de composantes connexes à partir de l'image binarisée.

De surcroît, ce procédé est extrêmement rapide et ne nécessite aucun apprentissage ni même une base de connaissance.

2.1.2 Application à la base Squid

La base Squid est composée de 1100 images binaires représentant la silhouette de différentes espèces de poissons de tailles et d'orientations variables. Cette base n'est cependant pas étiquetée. Ainsi, nous ne pouvons pas pas évaluer les résultats de façon objective dans cette section.

Pour classifier ces images nous avons opté pour des caractéristiques robustes par rapport à la rotation et à l'échelle, à savoir les 12 moments de Zernike [97].

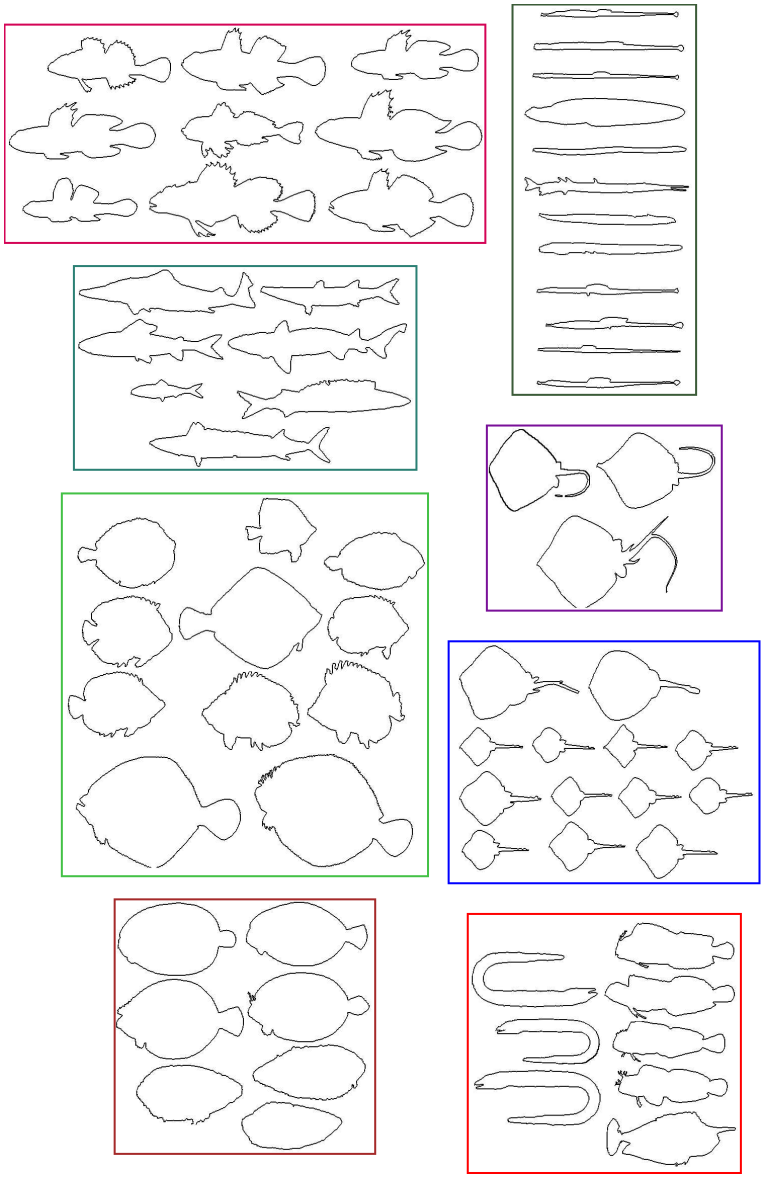


FIGURE IV.1 – 8 classes parmi les 132 produites par NCCascadeAP sur la base Squid.

La méthode NCCascadeAP, avec $A_s = 2$, appliquée à cette base donne lieu à 132 classes. Faute d'espace, nous ne présentons qu'un extrait (aléatoire) de 8 classes des résultats obtenus (*cf.* figure IV.1).

Nous avons introduit un critère d'arrêt : une classe composée de moins de 20 éléments ne doit pas être subdivisée. De ce fait, certaines formes faiblement présentées se retrouvent mélangées à d'autres (telles que les anguilles dans la classe encadrée en rouge) puisque le critère d'arrêt empêche la subdivision de la classe qui les contient.

En faisant abstraction de ces quelques cas de sous-segmentation, nous pouvons considérer que les résultats de classification de silhouettes sont très satisfaisants.

2.2 Illustration avec la quantification colorimétrique

Nous présentons dans cette section les résultats de classification de pixels RVB de quelques images connues et disponibles sur le Web.

Les vecteurs de caractéristiques, de dimension 3, sont donc directement lus à partir des images brutes.

2.2.1 Quantification dans l'espace RVB

De même que dans la section précédente, nous ne donnerons pas de résultats quantitatifs de la classification mais les images résultant de la quantification. La valeur RVB des pixels d'une classe donnée correspond à la moyenne de ses membres.

Nous avons choisi de présenter les résultats relatifs aux variantes de notre approche produisant peu de classes. En effet, les images quantifiées avec beaucoup (à partir de 300 environ) de couleurs rendent difficile l'appréciation visuelle des résultats puisque l'œil humain ne perçoit pas la différence avec l'image d'origine.

Les tableau IV.2 présente des résultats de classification de la variante KdCascadeAP de notre algorithme sur un échantillon de 5 images. Pour chacune des images de test, nous renseignons le nombre de classes résultats ainsi que le nombre de candidats à la classification dans ce même tableau.

Dans ce tableau, les partitions résultats correspondent toujours aux feuilles des arbres respectifs.

Le tableau IV.3 montre les résultats de classification des 5 images précédentes (tableau IV.2.(a)) en employant la variante NCGlobalAP cette fois. Chaque case de la colonne (c) présente la partition formée par les feuilles de l'arbre tandis que la colonne (d) correspond au niveau 2.

Le tableau IV.4 montre les résultats de classification des 5 mêmes images (présentées dans dans la première colonne du tableau IV.2) en employant la variante NCGlobal2means de notre approche. Chaque case de la colonne (e) présente la partition formée par les nœuds du premier niveau de l'arbre tandis que la colonne (f) correspond au deuxième niveau.

(a) Image de 262 144 pixels dont 109 219 pixels différents

(b) Résultat : 51 classes

(a) Image de 262 144 pixels dont 230 427 pixels différents

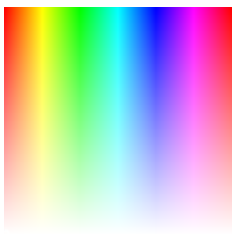
(b) Résultat : 95 classes

(a) Image de 173 280 pixels dont 53 240 pixels différents

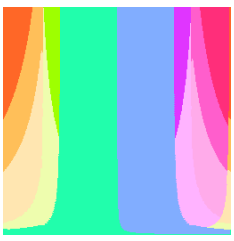
(b) Résultat : 29 classes

(a) Image de 262 144 pixels dont 148 279 pixels différents

(b) Résultat : 52 classes



(a) Image de 65 536 pixels dont 59 550 pixels différents



(b) Résultat : 12 classes

TABLE IV.2 – Résultats de quantification sur 5 images par la méthode KdCascadeAP

(c) Résultat : 216 feuilles

(d) Niveau 2 : 35 nœuds

(c) Résultat : 236 feuilles

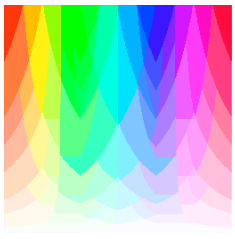
(d) Niveau 2 : 35 nœuds

(c) Résultat : 148 feuilles

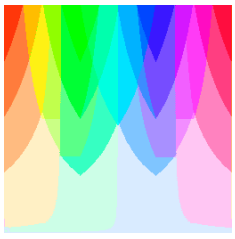
(d) Niveau 2 : 35 nœuds

(c) Résultat : 254 feuilles

(d) Niveau 2 : 35 nœuds



(c) Résultat : 125 feuilles



(d) Niveau 2 : 35 nœuds

TABLE IV.3 – Résultats de quantification sur 5 images par la méthode NCGlobalAP

(e) Niveau 1 : 8 nœuds

(f) Niveau 2 : 50 nœuds

(e) Niveau 1 : 8 nœuds

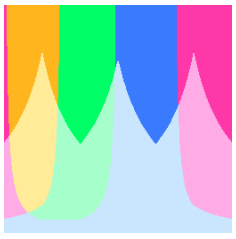
(f) Niveau 2 : 64 nœuds

(e) Niveau 1 : 8 nœuds

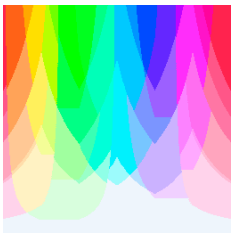
(f) Niveau 2 : 57 nœuds

(e) Niveau 1 : 8 nœuds

(f) Niveau 2 : 64 nœuds



(e) Niveau 1 : 8 nœuds



(f) Niveau 2 : 64 nœuds

TABLE IV.4 – Résultats de quantification sur 5 images par la méthode NCGlobal2means

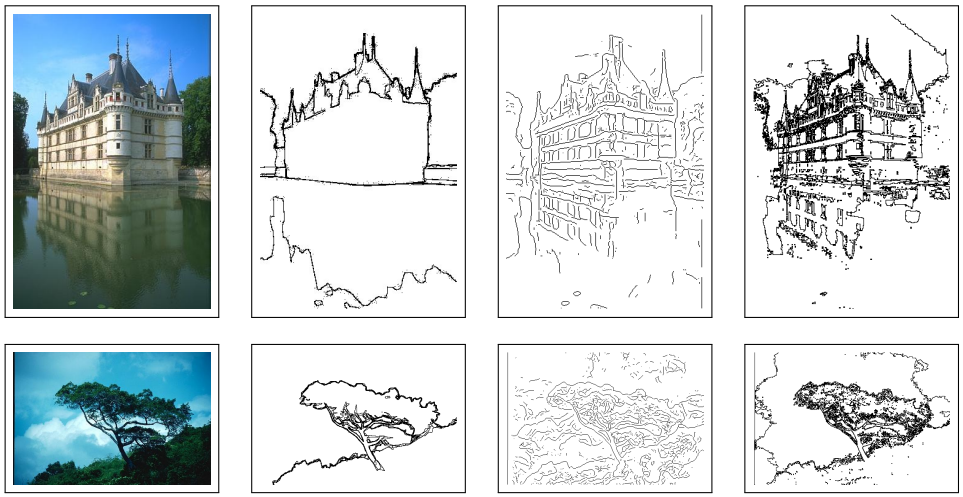
Les résultats présentés dans ces trois derniers tableaux montrent que notre approche peut donner lieu à une méthode de quantification faiblement paramétrée pour les images naturelles. Nous pouvons remarquer que les différentes variantes produisent des résultats légèrement différents. Le choix de la meilleure qualité de quantification est une question subjective.

2.2.2 Quantification dans l'espace L*a*b*

La base Berkeley [76] est un ensemble d'images accessibles en lignes pour évaluer des algorithmes de segmentation.

Nous avons testé notre moteur de classification sur quelques images de cette base mais, faute de temps, nous n'avons pas pu mesurer ses performances quantitativement. Par ailleurs, pour l'ensemble des expérimentations réalisées à cet effet, nous avons utilisé des caractéristiques très basiques, à savoir les valeurs des pixels L*a*b*.

Afin de pourvoir comparer nos résultats par rapport à la vérité terrain (établie manuellement), nous n'affichons pas des images couleur quantifiées mais les frontières entre chaque paire de pixels appartenant à des classes colorimétriques différentes. L'extrait pré-



a. Images d'origine | b. Vérité terrain | c. Approche la mieux classée [75] | d. KdCascadeAP

TABLE IV.5 – Résultats de segmentation d'images naturelles

senté dans le tableau IV.5 montre des résultats très encourageants et comparables avec les approches les plus compétitives, entre autres celle qui détient le meilleur score de segmentation [75].

Nous tâcherons d'approfondir nos recherches sur cette application en élaborant des caractéristiques plus appropriées et nous mesurerons ensuite ses performances quantitativement.

3 Validation du classement par la base MNIST

N'ayant pas évalué notre moteur de classement dans le précédent chapitre, nous présentons dans cette section ses résultats de reconnaissance sur une base conséquente de caractères manuscrits choisie pour sa popularité d'une part et pour la dimensionnalité élevée qu'elle implique d'autre.

3.1 Présentation de la base

Le corpus *Modified NIST* ou MNIST a été mis au point par LeCun [2] *et al.*. Il s'agit d'une base de données annotée décrivant une collection de chiffres manuscrits. Chaque chiffre est associé à une image en niveaux de gris sur 256 valeurs et de taille 28×28 (exemples en figure IV.2).

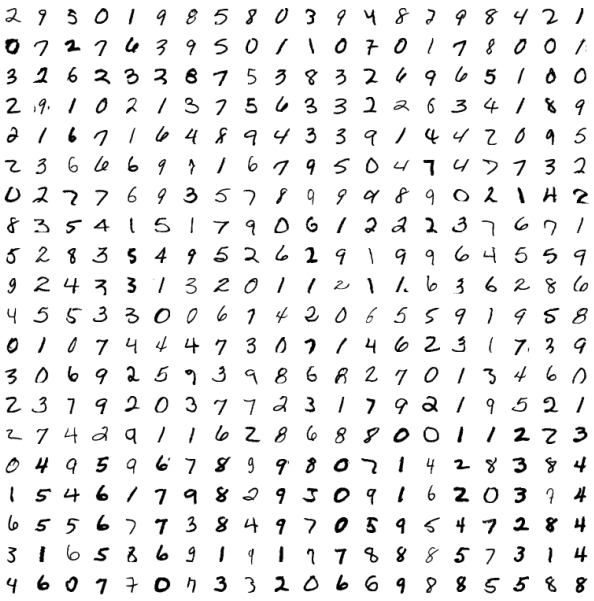


FIGURE IV.2 – Échantillon de la base MNIST

Les caractères manuscrits composant cette base ont été collectés respectivement parmi une population de lycéens et un groupe d'employés de bureaux du services des recensements. D'où une différence significative dans la qualité des écritures (environ 500 scripteurs différents ont participé à l'élaboration de cette base).

Ces imagettes sont réparties en deux sous-ensembles : une base d'apprentissage comptant 60 000 éléments et une base de test composée de 10 000 individus. Le tableau IV.6 décrit la répartition des classes de base (les chiffres de 0 à 9) dans chacun de ces deux ensembles.

2. http ://yann.lecun.com/exdb/mnist/index.html

Classe	0	1	2	3	4	5	6	7	8	9
Apprentissage	5 923	6 742	5 958	6 131	5 842	5 421	5 918	6 265	5 851	5 949
Test	980	1 135	1 032	1 010	982	892	958	1 028	974	1 009

TABLE IV.6 – Distribution des individus par classes dans la collection MNIST

3.2 Mesures d'évaluation

Comme nous l'avons mentionné précédemment, notre base de test compte 10 000 images de caractères numériques manuscrits tandis que la base de connaissance en compte 60 000. C'est ce dernier ensemble qui alimentera l'arbre de connaissance.

Les résultats de classement seront présentés en termes de matrices de confusion (*cf.* section 3.3.1) ou taux de reconnaissance qui est égal à la précision P et au rappel R moyens (on a toujours la valeur moyenne de P égale à la valeur moyenne de R). Les valeurs moyennes de P et R correspondent aux moyennes pondérés des 10 classes à reconnaître.

3.3 Résultats

3.3.1 Valeurs de luminosité des pixels comme descripteurs

Dans un premier temps, nous avons utilisé les descripteurs les plus triviaux pour classer les caractères de la base Mnist, c'est-à-dire la valeurs de luminosité de la matrice de pixels de taille 28×28. La dimension de l'espace de représentation s'élève donc à 784.

La matrice de confusion IV.1 est obtenu avec un arbre non-contraint dont d_s varie entre 0 et 4. Au niveau de chaque nœud, une LDA est appliquée afin de sélectionner les axes de projections. Nous pouvons calculer à partir de cette matrices le taux de reconnaissance qui vaut 86.5%.

$$\begin{pmatrix} 6 & 945 & 4 & 0 & 4 & 4 & 2 & 8 & 5 & 2 \\ 3 & 0 & 4 & 1086 & 1 & 6 & 12 & 3 & 2 & 18 \\ 7 & 10 & 15 & 7 & 7 & 906 & 16 & 32 & 17 & 15 \\ 46 & 8 & 10 & 7 & 22 & 28 & 834 & 0 & 17 & 38 \\ 8 & 6 & 833 & 6 & 53 & 9 & 10 & 21 & 11 & 25 \\ 711 & 13 & 12 & 4 & 17 & 10 & 60 & 15 & 11 & 39 \\ 28 & 12 & 30 & 5 & 1 & 16 & 3 & 848 & 0 & 15 \\ 9 & 6 & 12 & 11 & 34 & 26 & 23 & 1 & 901 & 5 \\ 45 & 10 & 26 & 24 & 27 & 20 & 36 & 22 & 10 & 754 \\ 11 & 9 & 61 & 5 & 830 & 8 & 27 & 3 & 29 & 26 \end{pmatrix} \quad \text{(IV.1)}$$

La matrice IV.2 présente les résultats de classement en employant la méthode du plus

proche voisin (1-NN), en employant la distance L1. Le taux de reconnaissance correspondant est de 95.6%.

$$\begin{pmatrix} 3 & 970 & 0 & 1 & 0 & 1 & 0 & 4 & 1 & 0 \\ 1 & 0 & 0 & 1129 & 0 & 3 & 1 & 1 & 0 & 0 \\ 0 & 11 & 1 & 11 & 0 & 980 & 6 & 2 & 16 & 5 \\ 19 & 1 & 0 & 2 & 4 & 4 & 963 & 0 & 9 & 8 \\ 0 & 0 & 927 & 9 & 36 & 0 & 0 & 4 & 4 & 2 \\ 844 & 3 & 0 & 1 & 3 & 0 & 25 & 6 & 4 & 6 \\ 4 & 5 & 3 & 4 & 0 & 1 & 0 & 939 & 0 & 2 \\ 0 & 0 & 4 & 23 & 17 & 5 & 2 & 0 & 977 & 0 \\ 23 & 5 & 6 & 5 & 5 & 7 & 23 & 2 & 7 & 891 \\ 6 & 2 & 21 & 7 & 943 & 1 & 7 & 1 & 19 & 2 \end{pmatrix} \quad \text{(IV.2)}$$

En dépit de sa rapidité, notre méthode de classement n'est pas satisfaisante. En effet, nous perdons 9% de précision et de rappel par rapport à 1-NN.

Ces résultats moyens sont dûs à l'incapacité du projecteur employé à gérer des espaces de dimension trop élevée. De surcroît, nous rappelons que notre approche est plus efficace lorsque $N >> d$. Nous envisageons de mettre au point de nouveaux projecteurs afin de mieux gérer ce type de données.

Espace de dimension réduite par l'ACP Pour remédier à ce problème, nous avons appliqué une ACP au préalable sur l'ensemble des données de la base de connaissance afin de réduire la dimentionnalité. En retenant 77% de l'information, notre espace est désormais de dimension 97. Les éléments de la base de test sont projetés dans cette nouvelle base avant d'être insérés dans l'arbre de connaissance.

Les résultats de classement obtenus avec le même type d'arbre que l'expérimentation précédente (arbre non-contraint dont $d_s \leq 4$ et LDA appliquée à chaque nœud) sont présentés en IV.7.(a). Cette matrice de confusion engendre un taux de reconnaissance de $R = 90.6\%$.

$$\begin{pmatrix} 5 & 953 & 2 & 0 & 2 & 2 & 1 & 7 & 2 & 6 \\ 0 & 0 & 4 & 1095 & 2 & 8 & 5 & 6 & 3 & 12 \\ 6 & 12 & 5 & 8 & 5 & 911 & 21 & 14 & 15 & 35 \\ 43 & 1 & 2 & 7 & 21 & 23 & 883 & 1 & 8 & 21 \\ 5 & 2 & 886 & 3 & 45 & 7 & 0 & 8 & 17 & 9 \\ 773 & 8 & 2 & 4 & 13 & 10 & 38 & 14 & 3 & 27 \\ 21 & 14 & 11 & 2 & 0 & 5 & 4 & 898 & 0 & 3 \\ 4 & 4 & 2 & 11 & 24 & 23 & 14 & 2 & 930 & 14 \\ 34 & 13 & 8 & 17 & 16 & 10 & 30 & 5 & 12 & 829 \\ 17 & 4 & 37 & 8 & 902 & 2 & 15 & 3 & 12 & 9 \end{pmatrix}$$

(a) Notre approche

$$\begin{pmatrix} 4 & 966 & 0 & 0 & 1 & 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 1129 & 0 & 3 & 1 & 2 & 0 & 0 \\ 2 & 8 & 3 & 4 & 2 & 974 & 14 & 4 & 9 & 12 \\ 19 & 5 & 1 & 1 & 4 & 5 & 945 & 1 & 8 & 21 \\ 3 & 1 & 928 & 1 & 28 & 4 & 2 & 7 & 6 & 2 \\ 824 & 3 & 1 & 2 & 7 & 1 & 23 & 11 & 2 & 18 \\ 8 & 8 & 5 & 4 & 0 & 3 & 1 & 925 & 0 & 4 \\ 0 & 0 & 6 & 15 & 29 & 12 & 3 & 0 & 960 & 3 \\ 17 & 1 & 7 & 1 & 5 & 7 & 19 & 4 & 6 & 907 \\ 7 & 3 & 22 & 4 & 934 & 3 & 6 & 1 & 22 & 7 \end{pmatrix}$$

(b) 1-NN

TABLE IV.7 – Résultats dans un espace, réduit par l'ACP, de dimension 97

Comme le montre le tableau IV.7, les résultats atteint par notre approche restent inférieurs, en terme de précision et rappel, à ceux induits par 1-NN pour lequel $R = 94.9$. Notons toutefois que notre méthode est beaucoup plus rapide.

Quelques comparaisons Duong *et al.* [28] appliquent la méthode 1-NN en cascade en appliquant une ACP visant à séparer les classes les plus corrélées à chaque niveau. Cette méthode de classement, qui présente certains points de similarité avec la notre, permet d'atteindre un taux de reconnaissance d'environ 96% sur cette base de données.

Par ailleurs, les meilleurs résultats, un taux de reconnaissance de 99.2%, sur la base Mnist sont atteints par les machines à vecteurs de support (SVM) à noyau polynomial de degré 9. Cependant, il faut avoir une connaissance *a priori* profonde de la base pour savoir que le noyau optimal est un polynôme de degré 9 ! De plus, dans l'absolu, les approches non-linéaires permettent toujours d'obtenir de meilleurs résultats que les approches linéaires quand les paramètres sont bien choisis.

3.3.2 Projections et profils comme descripteurs

Les descripteurs triviaux engendrant une dimension très élevée n'étant pas adéquats à notre approche, nous les avons remplacés par des caractéristiques légèrement plus pertinentes sans être très élaborées pour autant. Il s'agit des 6 histogrammes de projection et profils de taille 16 chacun que nous avons déjà utilisés auparavant (pour la classification des caractères manuscrits). La dimension de l'espace est donc de 96 pour les prochaines expérimentations.

La matrice de confusion IV.3 correspond au résultat de classement en employant un arbre non-contrait avec $d_s \leq 3$. Une LDA est appliquée au niveau de chaque nœud de cet arbre. Nous déduisons de cette matrice la valeur $R = 94.3\%$

$$\begin{pmatrix} 0 & 942 & 2 & 1 & 0 & 8 & 1 & 7 & 3 & 16 \\ 1 & 0 & 6 & 1116 & 0 & 5 & 0 & 3 & 1 & 3 \\ 2 & 4 & 1 & 0 & 1 & 985 & 10 & 1 & 10 & 18 \\ 8 & 1 & 0 & 2 & 8 & 20 & 944 & 0 & 7 & 20 \\ 1 & 1 & 934 & 3 & 15 & 10 & 0 & 6 & 1 & 11 \\ 837 & 2 & 3 & 5 & 7 & 2 & 17 & 8 & 0 & 11 \\ 14 & 8 & 6 & 4 & 0 & 5 & 1 & 910 & 0 & 10 \\ 4 & 1 & 6 & 7 & 16 & 8 & 10 & 0 & 971 & 5 \\ 11 & 11 & 8 & 5 & 6 & 12 & 14 & 9 & 6 & 892 \\ 1 & 2 & 35 & 9 & 901 & 7 & 8 & 0 & 29 & 17 \end{pmatrix} \quad \text{(IV.3)}$$

Les résultats de classement relatifs à 1-NN (distance L1) [3] sont présentés par la matrice IV.4. Nous en déduisons : $R = 92.5$.

$$\begin{pmatrix} 1 & 958 & 1 & 1 & 1 & 5 & 0 & 4 & 0 & 9 \\ 0 & 0 & 0 & 1123 & 0 & 4 & 1 & 4 & 0 & 3 \\ 0 & 14 & 1 & 1 & 1 & 955 & 25 & 5 & 8 & 22 \\ 27 & 3 & 0 & 2 & 5 & 13 & 914 & 2 & 13 & 31 \\ 0 & 1 & 902 & 3 & 52 & 2 & 0 & 11 & 9 & 2 \\ 778 & 6 & 2 & 1 & 8 & 1 & 49 & 10 & 2 & 35 \\ 3 & 16 & 1 & 4 & 0 & 0 & 1 & 929 & 0 & 4 \\ 0 & 0 & 15 & 7 & 74 & 10 & 2 & 0 & 915 & 5 \\ 15 & 56 & 3 & 4 & 9 & 8 & 12 & 6 & 11 & 850 \\ 2 & 6 & 19 & 5 & 922 & 2 & 10 & 0 & 32 & 11 \end{pmatrix} \quad \text{(IV.4)}$$

Le tableau IV.8 présente quelques résultats de classements de la méthode k-NN, avec différente valeurs de k. Ces derniers sont en terme de taux de reconnaissance.

Méthode	1-NN	3-NN	5-NN	7-NN	10-NN	**Notre approche**
$R = P$ (en %)	92.5	92.8	92.7	92.7	92.8	**94.3**

TABLE IV.8 – Résultats de classement dans un espace de dimension 96

3.4 Conclusion

Dans le cas où l'on dispose d'un ensemble de données étiquetées, il devient plus facile de superviser la classification. Dans cette optique, nous avons adapté notre algorithme de classification de manière à pouvoir l'exécuter en mode classement.

Les expérimentations montrent que, certes, nous n'avons pas surpassé les performances des algorithmes de reconnaissance les plus réputés mais les résultats obtenus sont très satisfaisants. Par ailleurs, notre approche est extrêmement rapide (complexité en $\mathcal{O}(\log(N))$).

De même que la plupart des approches existantes, notre méthode est moins performante lorsque la dimension de l'espace est trop élevée par rapport à la quantité des données.

3. Nous avons choisi de toujours appliquer la méthode k-NN avec la distance L1 car cette dernière permet d'atteindre des performances souvent meilleures par rapport à la distance classique L2 (distance Euclidienne). À titre d'exemple, pour cette expérimentation, 1-NN em employant L2 atteint $P = R = 91.6$.

4 Classement d'articles de presse et détection de publicités

Dans cette section, nous présenterons les résultats de classement de blocs de presse et de détection de publicités en employant notre classificateur. Les résultats obtenus seront comparées à ceux données par k-NN et AdaBoost (*cf.* chapitre II)

4.1 Présentation

Nous avons présenté, dans la section 3.3, les résultats de classement d'articles de presse, provenant de la segmentation physique de 5 documents différents, en employant les méthodes k-NN et AdaBoost respectivement.

Nous rappelons que :

- les descripteurs utilisés sont calculés à partir d'un ensemble de statistiques sur l'aspect colorimétrique et textuel des blocs ;
- l'espace de représentation est de dimension 21 ;
- le classement vise à répartir les candidats dans 4 classes respectivement étiquetées : *Texte*, *Texte&graphique*, *Publicité* et *Graphique* ;
- pour chaque expérimentation, les candidats sont répartis entre la base de connaissance (ou d'apprentissage) et la base de test selon un quota prédéfini. Ce dernier est fixé à **50%** pour toutes les expérimentations présentées dans cette section. Une centaine d'expérimentations sont effectuées en partageant les données entre les deux ensemble de façon aléatoire et en respectant le quota de répartition. Chaque paire de 'Précision, Rappel' présentée correspond donc aux valeurs moyennes des résultats inférés par cette centaine de tests.

Nous avons adopté ce même protocole expérimental pour classer ces mêmes données à l'aide de notre approche (présenté dans le chapitre précédent). Pour toutes nos expérimentations, nous avons utilisé un arbre non-contraint dans lequel une LDA est appliquée en cascade. La valeur maximale de d_s est choisie aléatoirement mais vérifie toujours : $1 \leq d_s \leq 3$. Toutes les bases de connaissances utilisées sont équilibrées.

Le tableau IV.9 présente les valeurs moyennes de résultats de classement, par notre approche, sur une totalité de 3458 blocs provenant de 5 journaux et magazines différents. Pour construire l'arbre de connaissance, le critère d'arrêt suivant a été appliqué : un nœud n'est pas subdivisé si la pureté de ses membres dépasse 95%.

		Texte	*Texte&graphique*	*Publicité*	*Graphique*	**Global**
document 1	*R*	69.2	91.7	**90.1**	48.8	**85.4**
	P	25.0	97.5	**93.8**	58.3	**85.4**
document 2	*R*	95.6	40.0	**96.6**	63.6	**92.0**
	P	98.2	40.0	**95.4**	58.3	**92.0**
document 3	*R*	98.6	42.9	**93.8**	61.8	**90.3**
	P	98.9	26.8	**93.5**	76.4	**90.3**
document 4	*R*	98.9	72.4	**59.7**	90.2	**90.8**
	P	99.7	31.8	**66.2**	97.0	**90.8**
document 5	*R*	96.1	87.5	**52.4**	62.5	**84.2**
	P	97.4	100	**50.0**	50.0	**84.2**
Moyenne	*R*	95.5	60.8	**78.9**	69.4	**89.5**
	P	92.3	43.8	**80.7**	76.6	**89.5**

TABLE IV.9 – Résultats de classement d'articles de presse en employant notre approche

4.2 Comparaisons

Nous ne pouvons juger de la qualité d'un résultat donné sans le comparer à d'autres références. Nous évaluerons donc les résultats de classement d'articles de presse, par notre méthode, en les comparant à deux approches qui se sont avérées complémentaires : k-NN et AdaBoost.

4.2.1 Notre approche *vs.* 3-NN

Les valeurs de P/R présentées en IV.10 sont extraites des tableaux IV.9 et II.3 [4].

		Notre approche	3-NN
Classement de publicités	*R*	78.9	69.8
	P	80.7	54.6
Classement global	*R*	89.5	86.4
	P	89.5	88.1

TABLE IV.10 – Comparaison des résultats de notre approche avec la méthode 3-NN

Nous pouvons constater que notre approche permet d'atteindre de meilleures performances, notamment en matière de détection et reconnaissance de publicités. Les résultats de classement globaux sont toutefois proches en raison de la prépondérance de la classe *Texte* qui est un peu mieux reconnue par k-NN.

4. À partir du tableau II.3, nous avons calculé les valeurs moyennes de précision et rappel entre les bases de connaissance équilibrées et celles qui ne le sont pas

En effet, la construction de l'arbre de connaissance est généralement satisfaisante : les feuilles sont souvent toutes pures à 100% ; la grande majorité des blocs sont répartis entre 4 nœuds représentant chacun une classe différente ; les nœuds restants sont composés de 1 à 3 éléments chacun. Ces derniers représentent les cas isolés, qui prêtent à confusion.

4.2.2 Notre approche *vs.* AdaBoost

Étant binaire, l'approche AdaBoost (la version de base) ne permet pas de séparer 4 classes. Nous avons donc choisi de reconnaître les publicités et de mettre les 3 autres classes dans un même lot. Il en découle une base formée de 8% d'échantillons positifs (publicités) et 92% de négatifs, en moyenne (entre les 5 documents).

Ces éléments sont répartis entre la base de connaissances et celle de test avec un taux de partage de 50%, en veillant à ce que la base de connaissance soit toujours équilibrée.

		Notre approche	AdaBoost
document 1	R	76.2	77.9
	P	99.7	32.9
document 2	R	90.9	72.0
	P	15.6	39.7
document 3	R	91.8	84.0
	P	37.2	46.3
document 4	R	96.6	85.9
	P	24.1	44.7
document 5	R	87.5	81.2
	P	31.8	33.2
Moyenne	R	91.3	82.8
	P	36.6	42.6

TABLE IV.11 – Comparaison des résultats de notre approche avec la méthode AdaBoost sur un corpus de 5 journaux / magazines

Le tableau IV.11 permet de comparer les performances de notre approche par rapport à la méthode populaire AdaBoost [5].

Nous pouvons constater, à partir de ces résultats, que notre méthode permet d'atteindre de meilleures performances globalement en haussant les valeurs de rappel. Cependant les deux approches comparées obtiennent des précisions insuffisantes.

En effet, l'observation des arbres de connaissance révèle que le découpage des nœuds s'arrête souvent au premier niveau de l'arbre, c'est-à-dire qu'un unique hyperplan permet de séparer les données de la base de connaissance en 2 classes parfaitement pures ! Or, les

5. Les résultats obtenus par AdaBoost sont présentés et discutés dans la section 3.3.3

valeurs de précision montrent que ce simple découpage ne permet pas de bien reconnaître les candidats de l'ensemble de test. Nous estimons donc qu'il s'agit d'un problème de sur-apprentissage : La taille de la base de connaissance est beaucoup trop réduite (en raison de l'équilibrage) ; elle est donc facilement séparable contrairement aux données de l'ensemble de test.

La méthode Adaboost est affectée par le même problème. Cette dernière présente, en effet, des points de similarité avec la notre : un seuil est calculé à partir d'une combinaison linéaire des axes de l'espace, de même que la LDA appliquée au niveaux des nœuds.

Nous pouvons en conclure qu'un unique seuil (hyperplan) n'est pas suffisant dans ce cas de figure.

Cependant, comme l'intervention humaine reste nécessaire tant que la précision (et le rappel) n'atteint pas 100%, nous pouvons considérer que ces résultats sont acceptables vu le nombre réduit de publicités dans un journal, en absolu.

4.3 Conclusion

Les résultats présentés ci-dessus montrent que notre méthode de classement permet d'atteindre des performances très satisfaisantes en terme de classement de blocs de presse en notamment en matière de détection et filtrage de publicités. En effet, cette application offre des conditions propices comme la dimension de l'espace (pas trop élevée ni trop faible). Cependant, la proportion réduite des publicités dans le corpus rend la précision de certaines variantes de cette application assez faible. L'algorithme de classement réputé AdaBoost est également affecté par ce même problème.

Nous présenterons, dans la prochaine section, une nouvelle application à notre méthode de classement.

5 Accumulation de preuves : application à la reconnaissance de polices

5.1 Méthodologie

5.1.1 Présentation

L'accumulation de preuves est applicable sur des images ou tout objet divisible en sous-parties. Il s'agit de classer, individuellement, les sous-ensembles formés et de déduire ensuite le type de l'objet global en considérant 'les votes' de ses parties.

Si le candidat au classement est une image, par exemple, nous proposons de la subdiviser en un ensemble de tuiles que l'on classe chacune indépendamment des autres. Le

type de l'image globale est celui le plus représenté par les imagettes formées (tuiles).

5.1.2 Motivations

Certaines images (ou objets quelconques) sont composées de zones d'aspects variables. Le classement de tels objets au niveau global peut être biaisé par la variabilité de ses composantes. Un traitement en local peut donc être une alternative efficace à cette problématique.

Pour ce faire, il est préférable d'effectuer une segmentation (logique, colorimétrique, *etc.*) produisant des éléments homogènes, mais si nous ne disposons pas des acquis nécessaires, il suffit de subdiviser l'image en tuiles de taille fixe.

5.2 Application à la reconnaissance de polices en utilisant la cooccurrence

Une page de document, même un article d'un journal, est rarement écrit avec des caractères partageant tous la même police et le même style (taille, mise en forme...). Pour cela, nous appliquons le concept d'accumulation de preuves pour classer les polices dans les images d'articles de presse.

5.2.1 Base de connaissances

Ne disposant pas de la charte graphique des journaux et magazines que nous traitons et ignorant ainsi les polices et styles utilisés dans ces documents, nous avons construit notre base de connaissance à partir d'un ensemble d'images 'de synthèse'. Il s'agit de pages du faux texte *lorem ipsum* écrit en l'une des 3 polices les plus fréquentes Times, Arial et Courrier. La taille du texte est fixe pour chaque image et est de 11, 12, 14 ou 18. Le format de la police (gras, italique, *etc.*) est également commun à tous les caractères d'une page de cette base. Le texte peut être dépourvu de toute mise en forme, gras ou italique. Notre base de connaissance compte ainsi 36 images.

Afin de se conformer aux candidats de la base de test, chaque image de synthèse est subdivisée en un ensemble de tuiles, de taille 256×256 chacune, avant d'être inséré dans l'arbre de connaissances.

5.2.2 Base de tests

Les candidats à la classification sont, de même que pour la détection des publicités, des blocs de journaux et magazines issus de la segmentation logique présentée dans le chapitre II.

Chaque bloc est subdivisé en un ensemble de tuiles de taille 256×256. Le nombre de représentants par candidat varie donc en fonction de sa taille.

5.2.3 Descripteurs

La cooccurrence [79] des niveaux de gris, connue sous le nom de SGLD (*Spatial Grey Level Dependence*), est une mesure statistique d'ordre 2. C'est la mesure de cooccurrence la plus utilisée dans la littérature. Soit I l'image sur laquelle nous calculons la SGLD. $I(x, y)$ est la valeur d'intensité observée en (x, y) et $I(x+u, y+v)$ correspond à la valeur d'intensité observée suite à une translation (u, v) des coordonnées. SGLD(u, v, i, j) compte le nombre de fois que $I(x, y)$ prend la valeur d'intensité i et que $I(x+u, y+v)$ la valeur d'intensité j. En termes de statistiques, la cooccurrence permet de calculer la loi conjointe d'observer simultanément les événements $(I(x, y) = i)$ et $(I(x+u, y+v) = j)$ pour tous les pixels (x, y) de l'image.

$$\text{SGLD}(u, v, i, j) = P(I(x, y) = i, I(x+u, y+v) = j). \tag{IV.5}$$

En passant au système de coordonnées polaires, la SGLD s'exprime par :

$$\text{SGLD}(\rho, \theta, i, j) = P(I(x, y) = i, I(x+\rho\cos(\theta), y+\rho\sin(\theta)) = j). \tag{IV.6}$$

Nous binarisons tous les blocs à classer avant d'y calculer les caractéristiques puisque les images de la base de connaissance sont en noir et blanc. Ainsi, i et j peuvent prendre deux valeurs possibles : 0 ou 255.

Pour chaque direction θ et chaque déplacement ρ, nous avons une matrice de cooccurrence de taille 2×2 (chaque cellule la matrice correspond à une valeur du couplet i, j). Si nous avons un déplacement ρ limité à ρ_{max} et un nombre θ_{max} de directions possibles, la SGLD est donc une matrice de $\rho_{max} \times \theta_{max}$ matrices de taille 2×2 chacune.

La taille de la fenêtre d'analyse ($\rho_{max} \times \theta_{max}$) doit être choisie en fonction de la taille des caractères en termes de pixels. Si on choisit une échelle d'analyse fine avec des déplacements infinitésimaux de quelques pixels, la cooccurrence mesure alors les formes des caractères. Avec un déplacement ρ deux fois plus grand que la taille d'un caractère, la cooccurrence mesure alors la différence entre les lettres adjacentes. Dans ce cas elle sera sensible à la langue utilisée et à la fréquence des lettres isolées ou à l'agencement des lettres successives qui dépendent de la langue et du script. Si le déplacement dépasse la taille de plusieurs lignes de texte, alors elle mesure la mise en page et l'organisation du texte.

Pour mesurer uniquement la forme des caractères, nous choisissons un nombre réduit de déplacement : $\rho_{max} = 8$; et pour simplifier la représentation, nous fixons le nombre

maximum d'orientations θ_{max} à 8 également. Nous appliquons ensuite une ACP sur les 64 matrices résultantes pour n'en garder que les 2 les plus discriminantes. Notre espace de représentation est ainsi de dimension 8 ($2 \times 2 \times 2$).

Il va de soi qu'il est inutile de calculer la cooccurrence sur les zones vides (blanches) de l'image. De même les zones plates (uniformes), où le gradient de magnitude nulle, n'apportent aucune information pertinente. Ainsi, chaque matrice de taille 2×2 correspond à la valeur moyenne des probabilités conjointes calculées sur les contours de l'image.

5.2.4 Résultats préliminaires

Comme nous l'avons mentionné précédemment, nous ne connaissons pas les polices définissant les chartes graphiques des journaux et magazines que nous traitons. Nous ne pourrons donc pas évaluer les résultats de reconnaissance de polices quantitativement. Nous présenterons, toutefois, quelques échantillons de résultats sous forme d'illustrations.

Pour l'ensemble des tests effectués, nous avons utilisé un *Kd-tree* et nous avons appliqué la LDA en cascade au niveau de tous les nœuds. Comme notre arbre de connaissance est construit à partir d'images de synthèses, nous pouvons l'utiliser pour tous les journaux ; c'est-à-dire qu'il n'y a pas besoin de prévoir une base de connaissance différente pour chaque base de test. Cela rend le classement particulièrement rapide et générique.

Échantillon de résultats Après avoir construit l'arbre de connaissances, nous y avons inséré l'ensemble des 3458 blocs provenant de 5 documents utilisés lors des expérimentations décrites dans la section précédente.

Les résultats de classement révèlent que les deux classes Arial 11 et Times 11, dont des échantillons sont présentés par les figures IV.3 et IV.4 respectivement, sont les plus peuplées. Ce résultat est prévisible puisque la plupart des articles de presse sont composés de texte de mise en forme simple et de taille réduite. Par ailleurs, ces polices sont souvent bien reconnues bien que nos images de presse soient numérisées sous différentes résolutions et présentent donc des aspects différents.

Nous pouvons constater, également, que le découpage de tuiles a souvent permis de classer correctement les blocs les plus mélangés. Considérons, par exemple, le premier bloc présenté dans la figure IV.4. Cet article compte trois polices différentes et une zone graphique. La subdivision en tuiles donne lieu à 41 imagettes non vides dont 14 de type Times 11, 3 de type Times 18, 3 de type Arial 14, 5 de type Courrier 14, 5 de type Arial 18, *etc.* Nous pouvons donc constater que le système de votes a permis d'affecter ce bloc à la bonne classe, les zones 'bruit' étant dispersées entre de nombreuses classes.

La figure IV.5 montre des blocs classés selon le type de leurs titres respectifs. Cela se produit quand le titre occupe une plus grande superficie que le reste du texte. C'est,

FIGURE IV.3 – Exemple de résultats satisfaisants : extrait de la classe reconnue comme Arial 11

encore une fois, une illustration de la démocratie qui donne raison à la majorité relative et qui fait en sorte que les résultats sont satisfaisants pour certains et décevants pour d'autres :-)

Les éléments assignés à la classe 'Courrier italique 18' (figure IV.6) partagent deux points commun : la mise en forme en italique et la présence de nombreux caractères écrits en majuscule. Nous estimons que ce sont ces critères qui ont guidé le classement. Or, nous pouvons constater, visuellement, que la police utilisée pour ces quatre articles n'est pas du

Courrier. Nous en concluons que, faute d'échantillon représentatif de ces candidats dans la base de connaissance, ils ont été assignés à la classe de police la plus similaire, 'Courrier', caractérisée, entre autres, par la présence de grands espacements entre les mots. Nous pouvons remarquer ce même effet dans les blocs de la figure IV.6.

5.3 Conclusion

La reconnaissance de polices à partir d'une base d'images de synthèse est une application très intéressante dans la mesure où il n'y a pas besoin de prévoir un système d'apprentissage supplémentaire pour chaque nouveau document.

Dans un premier temps, nous avons subdivisé les blocs en un ensemble de tuiles qui sont classées individuellement ; le type de l'article étant déduit des votes de ses composants. Le classement au niveau local permet de mieux classer les blocs de contenu hétérogène.

Les résultats montrent que la décomposition en tuiles est efficace sur les blocs dont le contenu représentatif (le plus intéressant) couvre une superficie suffisante mais pas sur les articles comptant peu de texte écrit en différentes polices. Ainsi, il serait intéressant de remplacer les tuiles par une segmentation plus appropriée en tenant compte des propriétés du texte.

Nous prévoyons, également, d'enrichir la base de connaissances en tachant de récupérer la charte graphique de certains journaux et magazines auprès de notre partenaire commercial. Cela permettra d'atteindre de meilleurs résultats et de les évaluer numériquement.

6 Conclusion

Nous avons présenté dans ce chapitre différents cas d'utilisation de notre moteur de classification/classement en commençant par deux principales applications qui se déroulent sans supervision, c'est-à-dire sans aucune intervention de l'utilisateur. Il s'agit notamment d'un système d'aide à la transcription de manuscrits et d'une quantification. Dans la mesure où les caractéristiques utilisées sont basiques (afin d'évaluer le classificateur, non pas les caractéristiques), les résultats obtenus sont très prometteurs : l'emploi de descripteurs plus sophistiqués permettrait d'atteindre de hautes performances.

Nous avons ensuite appliqué notre approche en mode reconnaissance. C'est la base de caractères manuscrits Mnist qui a été utilisée pour valider cette fonctionnalité. Les expérimentations révèlent que notre approche est performante lorsque les données dont représentées par des caractéristiques non-triviales (sans être forcément très élaborées pour autant), comme par exemple les projections et profils. Toutefois, l'usage de descripteurs

basiques, comme les pixels, impliquant une dimensionnalité trop élevée sans être informatifs, n'est pas approprié pour notre approche.

Les projecteurs utilisés restent linéaires et certaines données ont besoin d'une classification non linéaire. Pour traiter les données non linéairement séparables, nous devons utiliser un noyau (Kernel) pour l'analyseur-projecteur et donc appliquer un KPCA pour la classification ou un KLDA pour le classement. Mais le choix du noyau et son degré restent un véritable problème puisque ce choix dépend des données et des formes des classes.

Nous avons également appliqué notre approche pour classer les blocs de presse et détecter les publicités. De la segmentation colorimétrique au moteur de classification en passant par l'extraction de texte, le classement des blocs englobe quasiment tous les travaux réalisés dans le cadre de notre thèse.

Nous avons finalement mis en place une application du principe d'accumulation de preuves en mode supervisé. C'est un système de reconnaissance de polices de caractères à partir d'images de synthèse. Nous estimons que cette application est particulièrement intéressante pour sa généricité et la réutilisabilité de la base de connaissances.

Times de taille 11

financiers, le caricaturiste Siné, dit Bob, et sa femme jettent l'épo après un an et demi de résistance

«Siné Hebdo» se fait hara-kiri

«*Une rumeur est arrivée dans mes petites oreilles. Siné signerait la fin du seul journal mal élevé sur la place de France. Bob, pique du blé à Bedos, à Onfray, fais un casse.*» Depuis hier matin, les lettres de soutien affluent dans la boîte mail du courrier des lecteurs. La vente de tee-shirts connaît un regain (avec rupture de stock en XL de «*Ça va péter*»). Les chèques ne sont plus encaissés. Eh oui, *Siné Hebdo* ferme boutique. Puisque c'est Bob qui vous le dit. Ce jour même dans son édito sinique, dans «sa zone». «*On touche le fond de nos fouilles et, même avec toute votre générosité [...], le drapeau noir flotte désespérément sur notre marmite !*»
Un bandana noir à tête de mort lui ceint le front. Dans les locaux des éditions de l'Enragé, avenue de la Résistance à Montreuil, le patron octogénaire s'amuse encore et contemple avec satisfaction la une du numéro tout chaud de la semaine : un pape qui a la trique... en forme de biberon. Le traitement dessins du sujet en page 9 est gratiné.
«*Chier dans la colle et les bégonias*» ne fait pas assez vendre. Le célèbre caricaturiste de *Charlie Hebdo*, viré avec fracas par Philippe Val après une chronique sur une supposée conversion au judaïsme de Jean Sarkozy, avait bénéficié d'une vague de sympathie. Son éviction avait fait grand bruit, plus de 15 000 signataires avaient soutenu la pétition. Siné et sa femme, Catherine Sinet, lançaient alors dans la foulée leur propre canard sans publicité avec un capital de 2 400 euros. Les ventes de l'hebdomadaire, né le 10 septembre 2008, flirtaient à ses débuts avec les 100 000 exemplaires. Après une période de bricolage et de volontariat dans leur maison de Noisy-le-Sec, l'équipe avait pu emménager dans ces locaux à Montreuil. Et même embaucher – onze salariés – et payer des pigistes. Le journal semblait tenir la barre, tenait même la dragée haute à son ennemi juré. «*Qu'est-ce qui a foiré ?*, se demande Siné. *On n'aura pas réussi à la déclencher, la révolution...*»

Le ministre irlandais des Finances, Brian Lenihan, a dévoilé hier un plan de sauvetage bancaire qui verra l'Etat apporter de l'argent à l'établissement nationalisé Anglo Irish Bank, aider à se recapitaliser Allied Irish Banks (AIB) et Bank of Ireland (BofI) et nationaliser deux banques mutualistes, de taille plus modeste, l'Irish Nationwide et l'EBS. Lenihan a martelé devant les parlementaires que la nationalisation totale ou partielle de cinq des plus grandes banques du pays, à laquelle ce plan devrait aboutir, était «*la moins pire des solutions*». Dans le détail, l'Etat va injecter 8,3 milliards d'euros de capitaux dans l'Anglo Irish Bank, établissement qu'il avait nationalisé début 2009. Le ministre a ajouté qu'elle aura peut-être encore besoin d'une dizaine de milliards d'euros de capitaux pour mener à bien sa restructuration. L'Etat va également aider à se recapitaliser les groupes bancaires cotés AIB et BofI, après avoir déjà apporté à chacun 3,5 milliards d'euros. Le ministre a souligné que ces décisions seraient soumises à l'accord de la Commission européenne.

ETAT CIVIL Marie-José Pérec a donné hier naissance à son premier enfant. Le père est Sébastien Foucras, vice-champion olympique de ski acrobatique en 1998.

FOOT Sven-Göran Eriksson va conduire la Côte-d'Ivoire au Mondial. Le Suédois, ex-coach de l'Angleterre et du Mexique, remplace Vahid Halilhodzic, remercié après l'élimination des Eléphants en quart de finale de la Coupe d'Afrique des nations.

FOOT Raymond Domenech annoncera le 11 mai la liste des 23 joueurs qu'il emmènera au Mondial, et non le 2, comme prévu. Le 11 mai, c'est la Sainte-Estelle, mais ça n'a rien à voir.

Avec 100 millions d'euros par an, Mitterrand veut mettre fin à *«la lente érosion»* de la lecture.

sion des horaires d'ouverture»

...bitants, «*dont les deux tiers ne disposent toujours pas d'un accès internet*», seront informatisées. Le ministre souhaite aussi «*favoriser l'exten-*

...édiées ...e dou- ...ête du ...itulée ...se dé- roulera du 27 au 30 mai prochains et vise à inciter à lire dans l'espace public. «Premières pages», dispositif qui consiste à offrir à chaque nouveau-né un livre et un guide de lecture, sera étendu à la moitié des départements d'ici à 2015.
Enfin, le ministre préconise

FIGURE IV.4 – Exemple de résultats satisfaisants : extrait de la classe reconnue comme Times 11

FIGURE IV.5 – Exemple de résultats où le type du bloc correspond à son titre

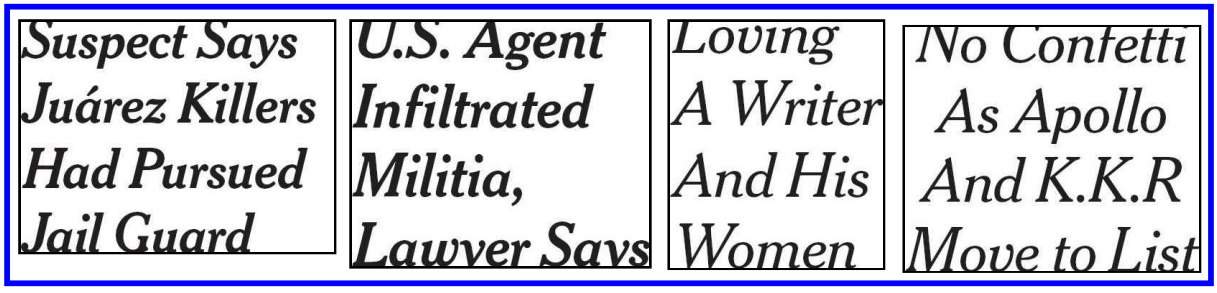


FIGURE IV.6 – Exemple de résultats médiocres

Conclusion

Table des matières

1 Application de synthèse

Il est consubstantiel, dans le cadre d'une thèse CIFRE, d'élaborer une application concrète donnant lieu à des résultats profitables dans un contexte industriel.

Le projet MediaBox a pour objectif de créer une chaîne de traitements complète et automatisée pour les images de presse de différentes provenances. Il s'agit, en l'occurrence, de filtrer les bruits de numérisation, de segmenter chaque journal / magazines en articles, de reconnaître les différentes composantes d'une page (texte, figure, filet, *etc.*), de fournir les moyens d'accéder au contenu sémantique des articles et de détecter les éventuelles zones publicitaires.

Nous avons présenté, dans les précédents chapitres, des applications permettant de préparer et de faciliter ces traitements. Nous proposons donc de réutiliser le travail effectué et de compléter les points non-abordés afin de concrétiser notre chaîne globale.

En entrée, nous avons une image de page (ou double-page) de presse quelconque sur laquelle nous ne disposons d'aucune information.

1. Le premier maillon de la chaîne de traitement consiste à effectuer une segmentation colorimétrique. Ceci permettra de déterminer le type d'opérations appropriées pour chaque zone de l'image et de filtrer certains bruits, par la même occasion. La séparation colorimétrique compte trois étapes clés : l'identification des éventuelles régions colorées suivie de la binarisation sélective et la séparation des couleurs qui peuvent s'exécuter parallèlement.
2. Les masques colorimétriques obtenus forment un support idéal pour appliquer la

technologie de compression MRC. Cet acquis permet, en effet, un taux de compression élevé et une qualité d'image irréprochable.

3. Une extraction de texte est ensuite effectuée.
4. Les deux premières étapes rendent la décomposition en blocs immédiate. Cette opération est donc réalisée pour donner lieu à des blocs de texte, des zones de graphiques et des blocs composites.
5. Nous pouvons appliquer un OCR sur l'image de texte générée par l'étape 2 ou alors sur chacun des masques binaires issus de la segmentation colorimétrique.
6. Nous calculons, pour chaque bloc, un vecteur de caractéristiques directement déduites des résultats de la décomposition colorimétrique et physique. Ces descripteurs permettent d'alimenter notre moteur de classement (ACPP) qui sépare les blocs selon leur contenu et détecte les éventuelles zones publicitaires.
7. Le suivi des articles devient possible à partir des résultats de l'OCR de la caractérisation issue de l'étape précédente. Il s'agit d'identifier les blocs de texte formant un même article, ce qui facilitera leur formatage et leur mise en ligne par la suite. Par ailleurs, cela permettra à certaines entreprises et aux particuliers d'accéder directement aux articles et publicités les concernant.
8. Jusqu'à maintenant, nous nous sommes intéressés à l'aspect structurel et sémantique du texte mais pas à son format. Or, dans le cadre de notre projet, il importe de disposer d'un système de caractérisation d'articles complet. En l'occurrence, chaque bloc doit être défini par son contenu (texte, graphique, *etc.*), le nombre de lignes le composant et leur aspect géométrique, la police de caractère utilisée, *etc.*

   Les articles de presse sont écrits en différents polices, sous différentes formes. Or, la charte graphique des journaux et magazines ne nous est pas fournie. Nous proposons donc de classer les styles de texte rencontrés à partir d'une base de connaissance composée d'images de synthèse.

L'ordre des étapes mentionnées précédemment est flexible : certains traitements sont intervertibles ou parallélisables tandis que d'autres sont facultatifs comme, par exemple, la compression MRC.

La figure IV.7 résume les principales composantes de la chaîne proposée. Ainsi, à partir des applications présentées dans les chapitres précédents, nous avons établi une chaîne de traitement complète dont les étapes ont déjà été développées ou préparées. Nous avons ainsi mis en œuvre la décomposition colorimétrique, la segmentation physique des images de presse, notre système de classification ou de classement, *etc.* pour nourrir les différents maillons de notre chaîne globale.

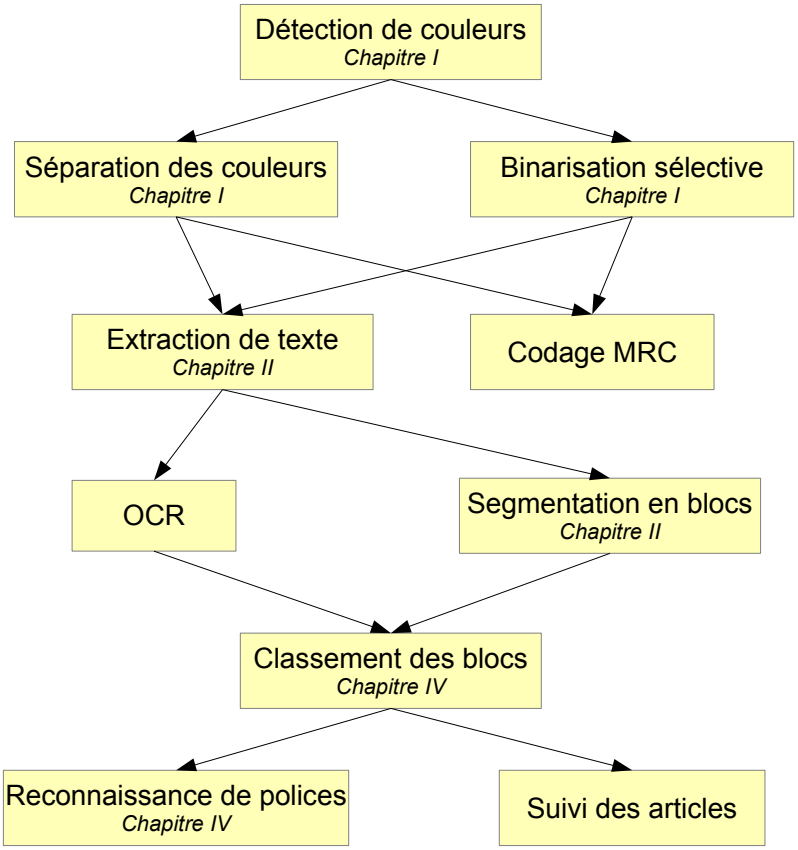


FIGURE IV.7 – Chaîne de traitement

2 Bilan

La première partie de cette thèse a été consacrée à la caractérisation, notamment colorimétrique, des images de documents. En l'occurrence, nous avons montré qu'il était possible de réserver à chaque image, ou zone d'image, un traitement idoine qui s'adapte à son contenu et prend en considération ses particularités locales, et ce de manière complètement automatisée. Ainsi, l'analyse et traitements proposés se détachent des processus traditionnels qui nécessitent une intervention humaine afin d'adapter les traitements aux images.

Pour ce faire, nous avons présenté un système de segmentation colorimétrique particulièrement efficace sur les images bruitées de par son aptitude à réparer les distorsions introduites par la chaîne de numérisation et de compression. Cette segmentation prépare, entre autres, une quantification (ou binarisation) conditionnelle sans perte d'information.

Tous les paramètres intervenant dans ce système sont évalués automatiquement grâce à une nouvelle mesure permettant d'estimer l'épaisseur des traits dans une image conte-

nant du texte. Par ailleurs, la séparation colorimétrique proposée est non-supervisée et générique, donc applicable à tout type de document.

Ce système comporte trois phases complémentaires et indépendantes : chacune est réutilisable indépendamment des autres. La séparation chromatique / achromatique se base, notamment, sur une nouvelle formulation de la saturation ainsi qu'un ensemble de filtres éliminant le bruit de saturation. Ainsi, les expérimentations révèlent que la détection des couleurs atteint une précision de 99.9%. À la suite de cette phase, les segmentations chromatique et achromatique sont lancées parallèlement. Ces dernières reposent sur une double validation utilisant des histogrammes locaux et globaux pour le classement des couleurs.

Nous avons mesuré l'apport de cette caractérisation sur deux applications fondamentales. La plus immédiate consiste en une méthode d'extraction de texte qui s'adapte aux propriétés locales de chaque image : une approche structurelle à base de regroupement de connexités est appliquée dans les zones monochromatiques tandis qu'une approche à base de gradients cumulés est appliquée dans les régions polytonales.

L'extraction des lignes de texte a permis d'améliorer le rappel de l'OCR Abby FineReader de 10% environ. Par ailleurs, le regroupement de ces lignes en paragraphes constitue une phase élémentaire dans le processus de segmentation physique que nous effectuons en amont d'une phase de classement des blocs de presse.

Cette dernière application est basée sur des descripteurs intuitifs calculés à partir de l'information colorimétrique et textuelle cumulées antérieurement. Le classement des blocs permet, notamment, de détecter les publicités qui s'infiltrent à l'intérieur de certains articles de journaux et magazines.

Nos descripteurs ont permis d'alimenter deux approches de classement dont les résultats se sont avérés complémentaires : Adaboost est plus pertinent pour les applications à objectif de pointage tandis que k-NN est plus approprié aux fonctionnalités de filtrage.

Il est, bien entendu, possible d'employer d'autres classifieurs pour classer les blocs de presse et détecter ainsi les zones publicitaires. Ainsi, nous avons présenté, dans le chapitre III une nouvelle méthode de classification / classement particulièrement rapide et facile à utiliser par un utilisateur quelconque grâce à son indépendance de tout paramètre abstrait. En l'occurrence, nous avons obtenu des taux de détection de publicité plus élevés en employant cet approche (ACPP) qu'avec les deux précédents classificateurs.

Il s'agit d'une approche hiérarchique et linéaire permettant de représenter les données dans un espace optimal à chaque niveau de partitionnement.

ACPP a été appliqué avec différents degrés de supervision. Dans le cadre d'une classification (supervision candide ou sous-jacente), cette approche a permis de catégoriser différents types d'objets, comme les composantes connexes dans des images de manus-

crits ou les couleurs dans des photos. Les résultats sont généralement très satisfaisants et dépassent certains algorithmes très populaires comme Mean-Shift et k-means.

Dans le cadre d'une classification assumée (un classement), nous avons employé ACPP (tout en l'adaptant) pour construire une base de connaissance et reconnaître les caractères numériques de la base Mnist. Les expérimentations liées à cette application révèlent un taux de reconnaissance très satisfaisant, en comparaison avec d'autres méthodes linéaires, et un temps d'exécution particulièrement court. Nous avons, par ailleurs, appliqué cette même méthode de classement, dans le cadre d'une accumulation de preuves, afin de reconnaître les polices de caractère dans les images de presse à partir d'un ensemble d'images de synthèse.

3 Perspectives

Notre système d'analyse colorimétrique permet de séparer une images de documents en plans chromatiques/achromatiques puis respectivement en plans multi-chromatiques non quantifiable / mono-chromatiques quantifiables pour la couche chromatique et en image quasi noir&blanc binarisable / niveaux de gris non binarisable pour la couche achromatique. Pour chaque plan, nous avons développé un test automatique pour savoir si celui-ci est binarisable. En revanche, faute de temps, nous n'avons pas sélectionné la méthode de binarisation ou de segmentation couleur la mieux adaptée pour chaque zone. Le LIRIS dispose de méthodes robustes de binarisation que nous envisageons d'incorporer dans notre système afin de l'enrichir d'une part et de mettre au point une méthode de compression MRC innovante d'autre part.

Dans le cadre de la reconnaissance de texte par OCR, le redimensionnement des zones de texte dont la mise de forme n'est pas gérée par FineReader permettrait d'améliorer les valeurs de rappel de façon considérable.

Il est également envisagé d'employer cette information sémantique acquise pour enrichir le vecteur de caractéristiques intervenant dans le processus de classement des blocs de presse. Cela permettra, par ailleurs, le suivi des articles de texte réguliers.

Notre moteur de classification effectue des partitionnements successifs des données en veillant à ce que ces dernières soient représentées dans un espace optimal à chaque niveau. Cet espace de représentation est déterminé à l'aide d'une ACP ou d'une LDA dans le cas d'une supervision assumée. L'inconvénient majeur de la LDA est de réduire la dimension de façon parfois drastique (la dimension maximale étant égale au nombre d'étiquettes différentes présentées par les données). Nous tâcherons donc de concevoir une nouvelle approche permettant de décorréler les axes de l'espace lorsque les étiquettes sont

connues *a priori*. Cela permettrait d'augmenter les taux de reconnaissance de notre arbre de connaissance.

Nous envisageons également de mettre en place un nouveau analyseur-projecteur non-linéaire afin de mieux représenter les données dans le cadre d'une classification (*clustering*) en utilisant des noyaux.

Dans le cas où ACPP emploie un partitionneur qui modélise les données par un mélange de Gaussiennes, il est envisagé d'utiliser ces modèles pour générer des données par projections inverses.

Annexe A

Algorithme EM (*Expectation Maximisation*)

Cet algorithme est conçu pour estimer les variables cachées. Il est notamment employé pour l'estimation des modèles de Markov cachées (HMM) ainsi que la résolution du problème des mélanges de Gaussiennes.

Soit $\mathcal{P} = \{C_1, ..., C_K\}$ la partition de l'espace recherchée et α_i le quota d'individus appartenant à une classe C_i. Nous avons donc :

$$\sum_{i=1}^{K} \alpha_i \ = 1. \tag{A.1}$$

P_i, $i = 1, .., K$, exprimant la probabilité d'appartenir à la classe C_i, la règle de Bayes appliquée à un échantillon x s'écrit :

$$P(x) = \sum_{i=1}^{K} \alpha_i P(x \backslash C_i). \tag{A.2}$$

Supposons que les individus de chaque classe i suivent une loi normale $\mathcal{N}(x; \mu_i, \sigma_i)$ (Gaussienne) définie par sa moyenne μ_i et son écart type σ_i. $P(x)$ vaut donc :

$$P(x) = \sum_{i=1}^{K} \alpha_i \mathcal{N}(x; \mu_i, \sigma_i). \tag{A.3}$$

C'est cette dernière somme qui définit le terme largement répandu de "mélange de Gaussiennes".

L'algorithme EM permet d'estimer α_k, μ_k et σ_k pour $k = 1, ..., K$ en employant un processus itératif composé des deux principales étapes 'E' et 'M' après l'initialisation de

$\vec{v}$ (c'est-à-dire l'estimation de $\vec{v}^{(0)}$).

$$\vec{v}^{(i)} = \begin{pmatrix} \alpha_1 & \mu_1 & \sigma_1 \\ \alpha_2 & \mu_2 & \sigma_2 \\ \vdots & \vdots & \vdots \\ \alpha_K & \mu_K & \sigma_K \end{pmatrix} \text{ à la } i^{ème} \text{ itération.} \tag{A.4}$$

Le nombre de variables cachées à estimer s'élève donc à $3K$ à chaque itération.

1 Étape 'E'

Il s'agit d'estimer, à l'itération i, les valeurs $h(n,k)^{(i)}$ avec $n = 1,..,N$ (N est le nombre d'individus) et $k = 1,..,K$ (K est le nombre de Gaussiennes) ; $h(n,k)$ exprime la probabilité qu'un individu x_n suit la $k^{ième}$ loi normale.

$$h(n,k)^{(i)} = \frac{\alpha_k^{(i)} \mathcal{N}(x_n; \mu_k^{(i)}, \sigma_k^{(i)})}{\sum_{j=1}^{K} \alpha_j^{(i)} \mathcal{N}(x_n; \mu_j^{(i)}, \sigma_j^{(i)})}. \tag{A.5}$$

2 Étape 'M'

Cette étape vise à maximiser la vraisemblance de l'ensemble des évènements x_j ($j = \{1, ..., N\}$) inhérents au mélange de Gaussiennes. Ces évènements sont supposés indépendants. Il s'agit donc de maximiser le produit :

$$\prod_{j=1}^{N} P(x_j/\vec{v}) = \prod_{j=1}^{N} \sum_{i=1}^{K} \alpha_i P(x_j \backslash C_i). \tag{A.6}$$

Pour ce faire, nous avons besoin de déterminer le vecteur $\hat{v}$ qui permet de maximiser ce produit.

$$\hat{v} = \text{argmax}_{j=1}^{K} \prod_{j=1}^{N} P(x_j/\vec{v}). \tag{A.7}$$

Pour résoudre cette équation, nous recalculons $\vec{v}^{(i+1)}$ à l'itération $i+1$ à partir des données construites à l'itération i.

Les résultats de calculs donnent :

$$\alpha_k^{(i+1)} = \frac{1}{N} S_k^{(i+1)} = \frac{1}{N} \sum_{j=1}^{N} h(j,k)^{(i)}, \tag{A.8}$$

$$\mu_k^{(i+1)} = \frac{1}{S_k^{(i+1)}} \sum_{j=1}^{N} h(j,k)^{(i)} x_j, \tag{A.9}$$

et

$$\sigma_k^{(i+1)^2} = \frac{1}{S_k^{(i+1)}} \sum_{j=1}^{N} h(j,k)^{(i)} (x_j - \mu_k^{(i+1)})^2; \tag{A.10}$$

sachant que

$$S_k^{(i+1)} = \sum_{j=1}^{N} h(j,k)^{(i)}. \tag{A.11}$$

Les itérations s'arrêtent quand la suite $(\vec{v}^{(}i))_i$ converge ; c'est-à-dire quand :

$$\| \vec{v}^{(i+1)} - \vec{v}^{(i)} \| \leq \epsilon. \tag{A.12}$$

Annexe B

Une classification par partionnements récursifs

Nous décrivons dans cette partie une méthode de classification hiérarchique descendante où les caractéristiques sont utilisées séquentiellement.

En partant du sommet de l'arbre vers les feuilles, à chaque niveau, une classe $\mathcal{C}$ (l'un des nœuds de l'arbre) est subdivisée en deux classes $\mathcal{C}_1$ et $\mathcal{C}_2$ selon un critère d'inertie intra-classes défini ci-dessous.

1 Critère d'inertie

Soit N le nombre d'individus définis dans l'espace Ω de dimension p. À chaque individu $x_i \in \mathbb{R}^p$ est associé un poids ω_i (généralement égal à $1/N$) ; $i = 1, ...N$.

L'inertie I d'une classe C_k est une mesure d'homogénéité égale à :

$$I(C_k) = \sum_{x_i \in C_k} \omega_i d^2(x_i, \bar{x_k}); \tag{B.1}$$

d étant la distance Euclidienne et $\bar{x_k}$ le centre de gravité de C_k.

Si K est le nombre de classes à un niveau donné, le critère d'inertie intra-classes W de la partition correspondante vaut :

$$W = \sum_{k=1}^{K} I(C_k). \tag{B.2}$$

2 Subdivision de $\mathcal{C}$

Soit n le nombre d'individus dans $\mathcal{C}$.

Selon chaque axe i, $i = 1, ..., p$, il existe $(n-1)$ bipartitions différentes de telle façon que $\mathcal{C}^1$ et $\mathcal{C}^2$ sont toutes les deux non vides. Le nombre total de bipartitions possibles issues de la subdivision de $\mathcal{C}$ s'élève donc à $p(n-1)$.

La bipartition choisie est celle qui maximise le critère W.

3 Choix de la classe $\mathcal{C}$

Le nœud à découper, à un niveau donné du dendrogramme, est choisi conformément au même critère d'inertie. Il s'agit de la classe $\mathcal{C}$ maximisant la différence d'inertie $\Delta(C_k)$ parmi toutes les classes C_k, $k = 1, ..K$, de bipartitions respectives C_k^1 et C_k^2.

$$\mathcal{C} = \text{argmax}_{k=1}^{K} \Delta(C_k) = I(C_k) - I(C_k^1) - I(C_k^2). \tag{B.3}$$

La détermination de la classe $\mathcal{C}$ qui sera effectivement subdivisée à un niveau donné nécessite donc de rechercher la subdivision optimale (en C_k^1 et C_k^2) relative à chacun des nœuds de l'arbre, $k = 1, ..K$. Ces calculs (notamment au voisinage des feuilles de l'arbre) font que cet algorithme assez complexe.

Annexe C

Analyse en Composantes Principales

L'analyse en composantes principales (ACP) est une méthode factorielle classique en traitement des données numériques. Nous rappelons ici les principales étapes calculatoires de l'ACP. Les démonstrations algébriques peuvent être retrouvées dans les nombreux ouvrages consacrés au méthodes multidimensionnelles [].

1 Principe

Soit $\mathcal{X}$ une famille de N vecteurs de formes de dimension d. Munissons l'espace de représentation des formes $\mathbb{R}^d$ d'un repère orthonormé d'origine O. Le centre de gravité de $\mathcal{X}$ sera

$$g = \begin{pmatrix} g_1 \\ \vdots \\ g_N \end{pmatrix} \tag{C.1}$$

avec

$$g_i = \frac{1}{N} \sum_{k=1}^{N} X_{k,i} \quad \forall i \in [\![1..d]\!], \tag{C.2}$$

X_k représentant le $k^{ième}$ élément de $\mathcal{X}$.

En guise de préliminaire, nous centrons l'ensemble des individus $\mathcal{X}$ par rapport à O. C'est-à-dire que nous retirons g à chaque vecteur X_k pour $k \in [\![1..N]\!]$. Ce qui revient encore à ramener le centre de gravité du nuage $\mathcal{X}$ sur O. Nous pouvons les consigner les individus ainsi translatés dans un tableau $N \times d$. Notons $\mathcal{X}$ cette matrice.

Nous calculons $Cov(X) = {}^tXX$, la matrice de covariance associée à X. Il s'agit d'une matrice carrée de taille $d \times d$. Si nous diagonalisons $Cov(X)$, nous obtenons un ensemble de d valeurs propres positives que nous noterons $\{\lambda_1 ... \lambda_d\}$. Soit v_i le vecteur propre associé à la valeur propre λ_i $\forall i \in [\![1..d]\!]$. Les vecteurs propres de la matrice de covariance sont aussi appelés composantes principales de l'échantillon statistique $\mathcal{X}$.

2 Commentaires

La famille $\{v_i\}_{i\in[\![1..d]\!]}$ constitue une base orthogonale de $\mathbb{R}^d$ et les vecteurs de $\mathcal{X}$ présentent une dispersion maximale dans le repère $\Big(O, \{v_i\}_{i\in[\![1..d]\!]}\Big)$.

Ces vecteurs propres permettront de projeter dans $\mathbb{R}^{d'}$, $d' \leq d$ les données $\mathcal{X}$.

$\forall v_i$, $i \in [\![1..d]\!]$, nous définissons une quantité

$$c_i = \frac{\lambda_i}{\sum_{k=1}^{d} \lambda_k} \tag{C.3}$$

qui donne la contribution de la composante λ_i dans la dispersion des données : les coefficients de v_i décrivent une combinaison linéaire des variables d'origine qui donnent une nouvelle caractéristique expliquant $(c_i \times 100)\%$ de la variance globale de l'échantillon statistique.

Annexe D

Analyse Linéaire Discriminante

Contrairement à l'analyse factorielle qui maximise la variance des N observations en fonction des d axes, l'analyse discriminante maximise la répartition des N observations dans leurs classes respectives. Il existe de nombreuses techniques de discrimination. Nous retiendrons la discrimination de Fisher qui suppose que les classes sont approximativement Gaussiennes et qu'elles peuvent être séparées linéairement. Si nous obtenons une bonne discrimination par une analyse discriminante linéaire alors la séparabilité des classes sera démontrée et il sera inutile d'utiliser des méthodes plus complexes. L'analyse discriminante ou *Linear Discriminant Analysis (LDA)* est aussi appelée *Fisher Linear Discriminant (FLD)* dans d'autres domaines de recherche.

1 Présentation

Soit $\mathcal{X}$ un échantillon de N points de dimension d (chaque point est donc décrit par d descripteurs) et répartis dans c classes *a priori* connues.

On définit $x_{i,j}$ comme le $j^{\text{ème}}$ individu de la classe i. Soit N_i son effectif. La matrice de dispersion intra-classe (*Within-class scatter matrix*) notée S_w décrit la dispersion des N points autour des barycentres des classes m_i pour chaque classe i de 1 à c.

$$S_w = \sum_{i=1}^{c} \sum_{j=1}^{N_i} {}^t(x_{i,j} - m_i)\,(x_{i,j} - m_i) \quad \text{avec} \quad m_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{i,j}. \tag{D.1}$$

La matrice de dispersion inter-classe (*Between-class scatter matrix*), notée S_b, décrit la dispersion des c classes entre elles autour du barycentre m de toutes les classes.

$$S_b = \sum_{i=1}^{c} {}^t(m_i - m)\,(m_i - m) \quad \text{avec} \quad m = \frac{1}{N} \sum_{i=1}^{c} N_i m_i. \tag{D.2}$$

Soit W un hyperplan de projection. On cherche W^* l'hyperplan qui assure la plus forte

discrimination entre les classes (voir figure D.1) [27]. W^* doit maximiser la dispersion inter-classes et minimiser la dispersion intra-classe. Si $\psi_b(W)$ et $\psi_w(W)$ sont respectivement les matrices de dispersion inter et intra classe dans l'espace projeté par W, alors la meilleur façon de définir W^* consiste à faire le rapport entre le déterminant de ψ_b et celui de ψ_w. S étant la matrice de dispersion et ψ la fonction de projection, on a $\psi(W) = {}^tWSW$. Par conséquent, W^* est l'argument pour lequel le rapport entre le déterminant de ${}^tWS_bW$ et celui de ${}^tWS_wW$ atteint un maximum. Ainsi, on cherche le vecteur de projections qui maximise la dispersion S_b des points entre les classes et qui minimise, simultanément, la dispersion S_w des points à l'intérieur des classes.

$$W^* = \text{argmax}_W \frac{|\psi_b(W)|}{|\psi_w(W)|} = \text{argmax}_W \frac{|{}^tWS_bW|}{|{}^tWS_wW|} \tag{D.3}$$

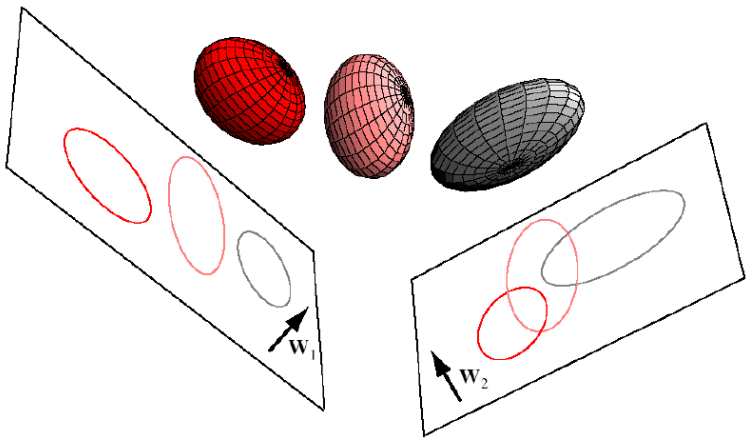


FIGURE D.1 – Contrairement à W_2, l'axe W_1 projette les données sur un plan de façon à discriminer les classes efficacement

Cette équation peut être résolue comme une optimisation de $\psi_b(W)$ sous contrainte que la dispersion projetée intra-classe est faible, c'est-à-dire que $\psi_w(W) = \epsilon$. Il faut donc maximiser $J(W) = \psi_b(W) - \lambda(\psi_w(W) - \epsilon)$, avec λ le multiplicateur de Lagrange. En annulant la dérivée de $J(W)$ par rapport à W on constate que les vecteurs de projection W qui maximisent la discrimination entre les classes sont les vecteurs propres de l'inverse de la matrice de dispersion intra-classe S_w multipliée par la matrice de dispersion inter-classe S_b. C'est-à-dire que W^* est formé par les vecteurs propres de $(S_w)^{-1}S_b$ associés aux plus grandes valeurs propres λ.

$$\begin{aligned} &\frac{\partial J(W)}{\partial W} = 0 \Rightarrow \frac{\partial\left({}^tWS_bW - \lambda\left({}^tWS_wW - \epsilon\right)\right)}{\partial W} = 0 \\ &\Rightarrow S_bW - \lambda S_wW = 0 \Rightarrow S_w^{-1}S_bW = \lambda W \end{aligned} \tag{D.4}$$

Il suffit donc, théoriquement, de diagonaliser $S_w^{-1}S_b$ pour obtenir les vecteurs de pro-

jection dans un nouvel espace où les classes sont les plus discriminées. Comme la matrice de dispersion inter-classe S_b est de rang $c-1$ (c est le nombre de classes), on ne peut trouver que $c-1$ vecteurs propres discriminants. Par conséquent l'analyse discriminante linéaire de Fisher peut projeter les N observations dans un espace de dimension $c-1$ au plus. La figure D.1 montre que les 3 classes peuvent être projetées au mieux sur un plan.

2 Commentaires

2.1 Difficultés

En pratique, la mise en œuvre de la LDA pose plusieurs difficultés :

- Le premier vecteur discriminant W_1 associé à la plus grande valeur propre projette les points de façon à discriminer au mieux les classes. Cependant, les autres vecteurs discriminants ne sont que des vecteurs orthogonaux à W_1 et qui ne sont pas toujours des vecteurs qui séparent les classes (voir figure D.1).
- Pour calculer $(S_w)^{-1}S_b$, il faut que S_w soit inversible, ce qui n'est pas toujours le cas si le nombre d'observations est trop faible en comparaison avec le nombre de caractéristiques d ou du nombre de classes c.
- $(S_w)^{-1}S_b$ n'est pas une matrice symétrique. Or, la plupart des algorithmes de diagonalisation fonctionnent sur des matrices symétriques. Il est donc nécessaire d'effectuer une décomposition de Cholesky de $S_w = L^tL$, où L est une matrice triangulaire inférieure, et de diagonaliser $L^{-1}{S_b}^t(L-1)$.
- La discrimination de Fisher sous-entend que les classes sont linéairement séparables et compactes. Par conséquent, les classes ne peuvent pas être divisées et éparpillées ou imbriquées dans tout l'espace.
- Pour pouvoir interpréter correctement les vecteurs discriminants W_k, $1 \leq k < c$, il est nécessaire de les normaliser en corrigeant les effets de Sw^{-1} qui joue le rôle d'une distance de Mahalanobis et qui réévalue certains facteurs au détriment d'autres. Pour normaliser les vecteurs W_k de façon à les interpréter, il suffit de multiplier $W_{k,r}$ par s_r le $r^{ème}$ élément de la diagonale de S_w.
- Pour que l'analyse discriminante fonctionne numériquement il est recommandé que $N > d > c$ ce qui pose problème dans certains cas d'utilisation.

2.2 Analyse discriminante non-linéaire

Le *Kernel-LDA*, KLDA, (comme le *Kernel-PCA*) revient à appliquer l'analyse discriminante sur des données transformées par un noyau K. Le KLDA (appelé aussi KFLD) permet donc de séparer des données non linéairement séparables ou imbriquées si on

connaît *a priori* le noyau K qui rend les données plus linéairement séparables. Yang *et al.* [123] montrent que le KPCA suivi d'une LDA est la meilleure méthode pour l'extraction de caractéristiques et la reconnaissance des formes pour des applications qui ont un très grand nombre de descripteurs comme celui de la reconnaissance d'images de visage ou d'objets par une analyse directe des pixels.

Mes publications

Revues internationales avec comité de lecture

- **Towards an omnilingual word retrieval system for ancient manuscripts.** Y. Leydier, A. Ouji, F Lebourgeois, H. Emptoz. *Pattern Recognition* 42(9) :2089-2105, Elsevier Science. 2009.

Conférences internationales avec comité de lecture et actes

- **Comprehensive color segmentation system for noisy digitized documents to enhance text extraction.** A. Ouji, Y. Leydier, F Lebourgeois. Dans *Document Recognition and Retrieval, SPIE* ed. Burlingame, California United States. 2012. (à paraître)
- **Chromatic / achromatic separation in noisy document images.** A. Ouji, Y. Leydier, F Lebourgeois. Dans *IEEE International Conference on Document Analysis and Recognition (ICDAR 2011)*, Beijing, China.
- **Advertisement detection in digitized press images.** A. Ouji, Y. Leydier, F Lebourgeois. Dans *IEEE International Conference on Multimedia&Expo*, IEEE ed. Barcelonne, Espagne. 2011.

Conférences nationales avec comité de lecture et actes

- **Extraction de texte à base de segmentation colorimétrique dans les images de presse.** A. Ouji, Y. Leydier, F Lebourgeois. Dans *CIFED* Bordeau, France. 2012. (à paraître)

Bibliographie

[1] S. Abirami and D. Manjula. Text region extraction from quality degraded document images. In *Proceedings of the 2nd international conference on Pattern recognition and machine intelligence*, PReMI'07, pages 519–527, Berlin, Heidelberg, 2007. Springer-Verlag.

[2] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, SIGMOD '98, pages 94–105, New York, NY, USA, 1998. ACM.

[3] Enrico Appiani, Francesca Cesarini, Anna Maria Colla, Michelangelo Diligenti, Marco Gori, Simone Marinai, and Giovanni Soda. Automatic document classification and indexing in high-volume applications. *IJDAR*, 4(2) :69–83, 2001.

[4] P.B. ; Ramakrishnan A.G. ; Indian Inst. of Sci. Bangalore Arvind, K.R. ; Pati. Automatic text block separation in document images. In *Intelligent Sensing and Information Processing. ICISIP 2006. Fourth International Conference on*, 2006.

[5] A. Atsalakis, N. Papamarkos, N. Kroupis, D. Soudris, and A. Thanailakis. Colour quantisation technique based on image decomposition and its embedded system implementation. *VISP*, 151(6) :511–524, December 2004.

[6] Henry S. Baird. Document image defect models. In Lawrence O'Gorman and Rangachar Kasturi, editors, *Document image analysis*, chapter Document image defect models, pages 315–325. IEEE Computer Society Press, Los Alamitos, CA, USA, 1995.

[7] H.S. Baird. Document image defect models and their uses. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 62 –67, oct 1993.

[8] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18 :509–517, September 1975.

[9] Daniel Boley. Principal direction divisive partitioning. *Data Min. Knowl. Discov.*, 2 :325–344, December 1998.

[10] Léon Bottou, Patrick Haffner, Paul G. Howard, Patrice Simard, Yoshua Bengio, and Yann Lecun. High quality document image compression with djvu. *Journal of Electronic Imaging*, 7 :410–425, 1998.

[11] J.P. Braquelaire and Luc BRUN. Comparison and optimization of methods of color image quantization. *IEEE Trans. on image processing*, 6 :1048–1051, 1997.

[12] S Bres, J.-M. Jolion, and F. Le Bourgeois. *Traitement et analyse des images numériques*. Hermes Science Publications, 2003.

[13] R. Cattoni, T. Coianiz, S. Messelodi, C. M. Modena, and Itc irst Via Sommarive. Geometric layout analysis techniques for document image understanding : a review. Technical report, 1998.

[14] G. CELEUX and E. DIDAY. *Classification automatique des données*. Dunod Informatique, 1989.

[15] Marie Chavent. A monothetic clustering method. *Pattern Recognition Letters*, 19(11) :989 – 996, 1998.

[16] Nawei Chen and Dorothea Blostein. A survey of document image classification : problem statement, classifier architecture and performance evaluation. *Int. J. Doc. Anal. Recognit.*, 10(1) :1–16, 2007.

[17] Qiang Chen, Quan-sen Sun, Pheng Ann Heng, and De-shen Xia. A double-threshold image binarization method based on edge detector. *Pattern Recogn.*, 41(4) :1254–1267, 2008.

[18] Yen-Lin Chen and Bing-Fei Wu. A multi-plane approach for text segmentation of complex document images. *Pattern Recogn.*, 42 :1419–1444, July 2009.

[19] Mark Ming-Tso Chiang and Boris Mirkin. Experiments for the number of clusters in k-means. In *Proceedings of the aritficial intelligence 13th Portuguese conference on Progress in artificial intelligence*, EPIA'07, pages 395–405, Berlin, Heidelberg, 2007. Springer-Verlag.

[20] S. Chowdhury, S. Mandal, A. Das, and B. Chanda. Segmentation of text and graphics from document images. In *ICDAR07*, pages 619–623, 2007.

[21] Byung Tae Chun, Younglae Bae, and Tai-Yun Kim. Automatic text extraction in digital videos using fft and neural network. In *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE International*, 1999.

[22] P. COMON. Analyse en composantes indépendantes et identification aveugle. *Traitement du Signal*, 07 :5, 1990.

[23] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20 :273–297, 1995. 10.1007/BF00994018.

[24] E. Deday. Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques. *Revue de Statistique Appliquée*, 19 :19–33, 1971.

[25] Fadoua Drira, Frank Lebourgeois, and Hubert Emptoz. A coupled mean shift-anisotropic diffusion approach for document image segmentation and restoration. In IEEE, editor, *ICDAR*, pages 814–818, September 2007.

[26] Fadoua Drira, Frank Lebourgeois, and Hubert Emptoz. OCR Accuracy Improvement Through a PDE-based Approach. In IEEE, editor, *The 9th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1068–1072, September 2007.

[27] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[28] Jean Duong and Hubert Emptoz. Cascade Classifier : Design and Application to Digit Recognition. In *8th International Conference on Document Analysis and Recognition (ICDAR2005)*, August 2005.

[29] R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4 :1–9, 1974. 10.1007/BF00288933.

[30] Lloyd A. Fletcher and Rangachar Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10 :910–918, November 1988.

[31] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[32] Ana L.N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 :835–850, 2005.

[33] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the Second European Conference on Computational Learning Theory*, pages 23–37, London, UK, 1995. Springer-Verlag.

[34] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1) :32 – 40, jan 1975.

[35] Utpal Garain, Thierry Paquet, and Laurent Heutte. On foreground-background separation in low quality document images. *Int. J. Doc. Anal. Recognit.*, 8 :47–63, March 2006.

[36] B. Gatos, I.E. Pratikakis, and S.J. Perantonis. Adaptive degraded document image binarization. *Pattern Recognition*, 39(3) :317–327, March 2006.

[37] Michael Gervautz and Werner Purgathofer. Graphics gems. chapter A simple method for color quantization : octree quantization, pages 287–293. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[38] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[39] Changsheng Gong and Fuxi Zhu. On detection of contextual advertisements. In *CAR'10 : Proceedings of the 2nd international Asia conference on Informatics in control, automation and robotics*, pages 29–32, Piscataway, NJ, USA, 2010. IEEE Press.

[40] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure : an efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, SIGMOD '98, pages 73–84, New York, NY, USA, 1998. ACM.

[41] Ujjwal Das Gupta, Vinay Menon, and Uday Babbar. Detecting the number of clusters during expectation-maximization clustering using information criterion. In *Machine Learning and Computing, International Conference on*, 2010.

[42] P. Haffner, L. Bottou, Y. Lecun, and L. Vincent. A general segmentation scheme for djvu document compression. In *ISMM'02, International Symposium on Mathematical Morphology*. Publications, 2002.

[43] Hiroyuki Hase, Masaaki Yoneda, Shogo Tokai, Jien Kato, and Y. Suen. Color segmentation for text extraction. *Int. J. Doc. Anal. Recognit.*, 6 :271–284, April 2003.

[44] Lifeng He, Yuyan Chao, Kenji Suzuki, and Kesheng Wu. Fast connected-component labeling. *Pattern Recogn.*, 42 :1977–1987, September 2009.

[45] Paul Heckbert. Color image quantization for frame buffer display. *SIGGRAPH Comput. Graph.*, 16(3) :297–307, 1982.

[46] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the theory of neural computation*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1991.

[47] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8) :832–844, August 1998.

[48] Thai V. Hoang and Salvatore Tabbone. Text extraction from graphical document images using sparse representation. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 143–150, 2010.

[49] J. J. Hopfield. *Neural networks and physical systems with emergent collective computational abilities*, pages 457–464. MIT Press, Cambridge, MA, USA, 1988.

[50] A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks, IEEE Transactions on*, 3 :626–634, 1999.

[51] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : a review. *ACM Comput. Surv.*, 31 :264–323, September 1999.

[52] A.K. Jain and Bin Yu. Automatic text location in images and video frames. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 2, pages 1497 –1499 vol.2, August 1998.

[53] Anil K. Jain. Data clustering : 50 years beyond k-means. *Pattern Recognition Letters*, 31(8) :651 – 666, 2010. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR), 19th International Conference in Pattern Recognition (ICPR).

[54] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[55] Anil K. Jain and Bin Yu. Document representation and its application to page decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3) :294–308, 1998.

[56] J.H. Jang and K.S. Hong. Binarization of noisy gray-scale character images by thin line modeling. *Pattern Recognition*, 32(5) :743–752, May 1999.

[57] Jung, Kim, and Jain. Text information extraction in images and video : a survey. *Pattern Recognition*, 37(5) :977–997, May 2004.

[58] Karin Kailing, Hans P. Kriegel, and Peer Kroger. Density-Connected Subspace Clustering for High-Dimensional Data. In *Proc. 4th SIAM International Conference on Data Mining*, April 2004.

[59] Tapas Kanungo and Song Mao. Stochastic language models for style-directed layout analysis of document images. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 12 :583–596, 2003.

[60] D. Karatzas and A. Antonacopoulos. Colour text segmentation in web images based on human perception. *Image Vision Comput.*, 25(5) :564–577, 2007.

[61] Swapnil Khedekar, Vemulapati Ramanaprasad, Srirangaraj Setlur, and Venugopal Govindaraju. Text - image separation in devanagari documents. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, ICDAR '03, pages 1265–, Washington, DC, USA, 2003. IEEE Computer Society.

[62] Jae Hyup Kim, Do Kyung Shin, and Young Shik Moon. Color transfer in images based on separation of chromatic and achromatic colors. In *MIRAGE '09 : Procee-*

dings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques, pages 285–296, Berlin, Heidelberg, 2009. Springer-Verlag.

[63] Charles ; Teboulle Marc Kogan, Jacob ; Nicholas. *Grouping Multidimensional Data : Recent Advances in Clustering*. Springer, 2006.

[64] F. Le Bourgeois. Content based image retrieval using gradient color fields. In *Pattern Recognition, International Conference on*, 2000.

[65] F. Le Bourgeois and H. Emptoz. Debora : Digital access to books of the renaissance. *International Journal on Document Analysis and Recognition*, 9(2) :193–221, 2007.

[66] F. LeBourgeois and H. Emptoz. Document analysis in gray level and typography extraction using character pattern redundancies. In *ICDAR '99*, pages 177–, Washington, DC, USA, 1999. IEEE Computer Society.

[67] Yann Leydier, Frank Lebourgeois, and Hubert Emptoz. Serialized k-means for adaptative color image segmentation : application to document images and others. In *DAS2004*, Lecture Notes in Computer Science, pages 252–263. Springer, 2004.

[68] Yann Leydier, Frank Lebourgeois, and Hubert Emptoz. Serialized unsupervised classifer for adaptative color image segmentation : Application to digitized ancient manuscripts. In *ICPR 2004*, pages 494–497, 2004.

[69] Dongfang Li, Bin Wang, Zhiwei Li, Nenghai Yu, and Mingjing Li. On detection of advertising images. In *ICME*, pages 1758 –1761, jul. 2007.

[70] Yanhong Li, Daniel Lopresti, George Nagy, and Andrew Tomkins. Validation of image defect models for optical character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18 :99–108, 1996.

[71] Young-Kyu Lim, Song-Ha Choi, and Seong-Whan Lee. Text extraction in mpeg compressed video for content-based indexing. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 409 –412 vol.4, 2000.

[72] Y. Liu and S.N. Srihari. Document image binarization based on texture features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5) :540–544, May 1997.

[73] Zongyi Liu, Hanning Zhou, and Ning Yang. Semi-supervised learning for text-line detection. *Pattern Recogn. Lett.*, 31 :1260–1273, August 2010.

[74] Poh Kok Loo and Chew Lim Tan. Adaptive region growing color segmentation for text using irregular pyramid. In Simone Marinai and Andreas Dengel, editors, *Document Analysis Systems VI*, volume 3163 of *Lecture Notes in Computer Science*, pages 103–106. Springer Berlin / Heidelberg, 2004.

[75] Michael Maire, Pablo Arbelaez, Charless Fowlkes, and Jitendra Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008.

[76] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[77] B Mirkin. *Clustering for Data Mining : A Data Recovery Approach*. Chapman & Hall/CRC, 2005.

[78] S. Mo and V.J. Mathews. Adaptive, quadratic preprocessing of document images for binarization. *IEEE Transactions on Image Processing*, 7(7) :992–999, July 1998.

[79] Ikram Moalla Koubaa. *Caractérisation des écritures médiévales par des méthodes statistiques basées sur la cooccurrence*. PhD thesis, LIRIS, INSA de Lyon, 2009.

[80] A. Moghaddamzadeh and N. Bourbakis. A fuzzy region growing approach for segmentation of color images. *PR*, 30(6) :867–881, June 1997.

[81] Frank Moosmann, Eric Nowak, and Frédéric Jurie. Randomized Clustering Forests for Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9) :1632–1646, September 2008.

[82] P. Nagabhushan and S. Nirmala. Text extraction in complex color document images for enhanced readability. *Intelligent Information Management*, 2 :120–133, 2010.

[83] G Nagy and S Seth. Hierarchical representation of optically scanned documents. In *Proceedings of the 7th International Conference on Pattern Recognition Montreal Canada*, pages 347–349. IEEE Computer Society, 1984.

[84] Nikos Nikolaou and Nikos Papamarkos. Color reduction for complex document images. *Int. J. Imaging Syst. Technol.*, 19(1) :14–26, 2009.

[85] S. Nirmala and P. Nagabhushan. Foreground text extraction in color document images for enhanced readability. In *Proceedings of the 3rd International Conference on Pattern Recognition and Machine Intelligence*, PReMI '09, pages 387–392, Berlin, Heidelberg, 2009. Springer-Verlag.

[86] L. O'GORMAN. Subsampling text images. In *ICDAR*, 1991.

[87] H.H. Oh, K.T. Lim, and S.I. Chien. An improved binarization algorithm based on a water flow model for document image with inhomogeneous backgrounds. *Pattern Recognition*, 38(12) :2612–2625, December 2005.

[88] J. Ohya, A. Shio, and S. Akamatsu. Recognizing characters in scene images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16 :214–220, February 1994.

[89] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1) :62–66, January 1979.

[90] Asma Ouji, Yann Leydier, and Frank LeBourgeois. Chromatic / achromatic separation in noisy document images. In *IEEE International Conference on Document Analysis and Recognition*, 2011.

[91] Nikos Papamarkos, Antonis E. Atsalakis, Charalampos P., and Charalampos P. Strouthopoulos. Adaptive color reduction. *IEEE Systems Man and Cybernetic - Part B*, 32 :44–56, 2002.

[92] T. PAVLIDIS and J. ZHOU. Page segmentation by white streams. In *ICDAR*, 1991.

[93] T. Perroud, K. Sobottka, and H. Bunke. Text extraction from color documents-clustering approaches in three and four dimensions. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 937 –941, 2001.

[94] Alain Pujol and Liming Chen. Color quantization for image processing using self information. In *international Conference on Information Communications and Signal Processing (ICICS)*, December 2007.

[95] Alain Pujol and Liming Chen. Coarse adaptive color image segmentation for visual object classification. In *15th International Conference on Systems, Signals and Image Processing*, 2008.

[96] S. Raju, P. Pati, and A. Ramakrishnan. Text localization and extraction from complex color images. In George Bebis, Richard Boyle, Darko Koracin, and Bahram Parvin, editors, *Advances in Visual Computing*, volume 3804 of *Lecture Notes in Computer Science*, pages 486–493. Springer Berlin / Heidelberg, 2005.

[97] Jérôme Revaud, Guillaume Lavoué, and Atilla Baskurt. Improving zernike moments comparison for optimal similarity and rotation angle retrieval. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 31(4) :627–636, April 2009.

[98] Neil C. Rowe, Jim Coffman, Yilmaz Degirmenci, Scott Hall, Shong Lee, and Clifton Williams. Automatic removal of advertising from web-page display. In *JCDL '02 : Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 406–406, New York, NY, USA, 2002. ACM.

[99] David A. Sadlier, Seán Marlow, Noel E. O'Connor, and Noel Murphy. Automatic tv advertisement detection from mpeg bitstream. In *PRIS '01*, pages 14–25. ICEIS Press, 2001.

[100] S. Salvador and P. Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 576 – 584, 2004.

[101] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases : The algorithm gdbscan and its applications. *Data Min. Knowl. Discov.*, 2 :169–194, June 1998.

[102] I. Sarafis, A.M.S. Zalzala, and P.W. Trinder. A genetic rule-based data clustering toolkit. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, volume 2, pages 1238 –1243, 2002.

[103] P. Scheunders. A comparison of clustering algorithms applied to color image quantization. *Pattern Recogn. Lett.*, 18(11-13) :1379–1384, 1997.

[104] Jean Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA, 1983.

[105] Tominaga Shoji. *A color classification algorithm for color images*, volume 301 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1988.

[106] E. Smigiel, A. Belaid, and H. Hamza. Self-organizing maps and ancient documents. In *Document Analysis Systems,*, pages 125–134, 2004.

[107] Karin Sobottka, Heino Kronenberg, T. Perroud, and Horst Bunke. Text extraction from colored book and journal covers. *International Journal on Document Analysis and Recognition*, 2 :163–176, 2000.

[108] C. Strouthopoulos, N. Papamarkos, and A. E. Atsalakis. Text extraction in complex color documents. *PATTERN RECOGNITION*, 35 :1743–1758, 2002.

[109] S. K. Tasoulis, D. K. Tasoulis, and V. P. Plagianakos. Enhancing principal direction divisive clustering. *Pattern Recogn.*, 43(10) :3391–3411, October 2010.

[110] Taku A. Tokuyasu and Philip A. Chou. Turbo recognition : A statistical approach to layout analysis, 2001.

[111] Oivind Due Trier and Torfinn Taxt. Evaluation of binarization methods for document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17 :312–315, 1995.

[112] A. Trémeau, C. Fernandez-Maloigne, and P. Bonton. *Image numérique couleur*. Dunod, 2004.

[113] S Tsujimoto and H Asada. Major components of a complete text reading system. *IEEE PAMI*, 80(7) :1133–1149, 1992.

[114] Toshio Uchiyama and Michael A. Arbib. Color image segmentation using competitive learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16 :1197–1206, December 1994.

[115] Jinqiao Wang, Lingyu Duan, Qingshan Liu, Hanqing Lu, and Jess S. Jin2. Robust commercial retrieval in video streams. In *ICME*, 2007.

[116] Alok Watve and Shamik Sural. Soccer video processing for the detection of advertisement billboards. *Pattern Recognition Letters*, 29(7) :994 – 1006, 2008.

[117] A. Weeks and G. Hague. Color segmentation in the hsi color space using the k-means algorithm. *SPIE*, 3026 :143–154, February 1997.

[118] C. Wolf and J.-M. Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal on Document Analysis and Recognition*, 8(4) :280–296, 2006.

[119] C. Wolf, J.-M. Jolion, and F. Chassaing. Text Localization, Enhancement and Binarization in Multimedia Documents. In *Proceedings of the International Conference on Pattern Recognition*, volume 2, pages 1037–1040, 2002.

[120] Xiaolin Wu. Color quantization by dynamic programming and principal analysis. *ACM Trans. Graph.*, 11(4) :348–372, 1992.

[121] Chunxia Xiao and Meng Liu. Efficient mean-shift clustering using gaussian kd-tree. *Computer Graphics Forum*, 29(7) :2065–2073, 2010.

[122] Hua Yang, Norikazu Onda, Masaaki Kashimura, and Shinji Ozawa. Extraction of bibliography information based on image of book cover. In *Proceedings of the 10th International Conference on Image Analysis and Processing*, pages 921–, 1999.

[123] Jian Yang, A.F. Frangi, Jing-Yu Yang, David Zhang, and Zhong Jin. Kpca plus lda : a complete kernel fisher discriminant framework for feature extraction and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(2) :230 –244, feb. 2005.

[124] Jian Yi, Yuxin Peng, and Jianguo Xiao. Color-based clustering for text detection and extraction in image. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 847–850, New York, NY, USA, 2007. ACM.

[125] Liang Zhang, Zhenfeng Zhu, and Yao Zhao. Robust commercial detection system. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 587 –590, jul. 2007.

[126] Yu Zhong, Kalle Karu, and Anil K. Jain. Locating text in complex color images. *Pattern Recognition*, 28(10) :1523 – 1535, 1995.

FOLIO ADMINISTRATIF

THÈSE SOUTENUE DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON

Nom : OUJI **Date de soutenance** : 01 juin 2012
Prénoms : Asma

Titre : Segmentation et classification dans les images de documents numérisés

Nature : Doctorat **Numéro d'ordre** : 2012-ISAL0044
École doctorale : InfoMaths
Spécialité : Informatique

Résumé :
Les travaux de cette thèse ont été effectués dans le cadre de l'analyse et du traitement d'images de documents imprimés afin d'automatiser la création de revues de presse. Les images en sortie du scanner sont traitées sans aucune information a priori ou intervention humaine. Ainsi, pour les caractériser, nous présentons un système d'analyse de documents composites couleur qui réalise une segmentation en zones colorimétriquement homogènes et qui adapte les algorithmes d'extraction de textes aux caractéristiques locales de chaque zone. Les informations colorimétriques et textuelles fournies par ce système alimentent une méthode de segmentation physique des pages de presse numérisée. Les blocs issus de cette décomposition font l'objet d'une classification permettant, entre autres, de détecter les zones publicitaires. Dans la continuité et l'expansion des travaux de classification effectués dans la première partie, nous présentons un nouveau moteur de classification et de classement générique, rapide et facile à utiliser. Cette approche se distingue de la grande majorité des méthodes existantes qui reposent sur des connaissances *a priori* sur les données et dépendent de paramètres abstraits et difficiles à déterminer par l'utilisateur. De la caractérisation colorimétrique au suivi des articles en passant par la détection des publicités, l'ensemble des approches présentées ont été combinées afin de mettre au point une application permettant la classification des documents de presse numérisée par le contenu.

Mots-clés : images scannées bruitées, analyse colorimétrique, segmentation physique, classification, classement.

Laboratoire de recherche : Laboratoire d'InfoRmatique en Images et Systèmes d'information (LIRIS)
Président de jury :
Composition du jury : Pr. Jean-Marc OGIER, Pr. Christian VIARD-GAUDIN, Pr. Patrick LAMBERT, Pr. Atilla BASKURT, Frank LEBOURGEOIS, Pierre-François BESSON