Reconnaissance et classification d'images de documents

Olivier Augereau

Thèse

présentée à

L'UNIVERSITÉ BORDEAUX 1

École Doctorale Mathématique et Informatique

par **Olivier AUGEREAU**

POUR OBTENIR LE GRADE DE DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Reconnaissance et classification d'images de documents

---

dirigée par Jean-Philippe DOMENGER

Soutenue le 14 février 2013

Devant la commission d'examen formée de :

M. Rémy Mullot	Université de la Rochelle	rapporteur
M. Antoine Tabbone	Université de Lorraine	rapporteur
M. Nicolas Ragot	Université de Tours	examinateur
Mme. Maylis Delest	Université de Bordeaux	présidente du jury
M. Jean-Philippe Domenger	Université de Bordeaux	directeur
M. Nicholas Journet	Université de Bordeaux	examinateur

LaBRI - Laboratoire Bordelais de Recherche en Informatique
Univ. Bordeaux, UMR 5800
F-33400 Talence, France.

Remerciements

Mes remerciement vont en premier lieu à Jean-Philippe Domenger mon directeur de thèse au LaBRI et Jean-Marc Nahon, le responsable des études informatiques de la société Gestform, pour avoir tout deux cru en mes capacités en me confiant ces travaux de recherche.

La réalisation de cette thèse n'aurait tout simplement pas été possible sans le soutient de Nicholas Journet, mon encadrant. Son attention, son aide et sa bonne humeur inaltérable m'ont permis d'avancer dans la bonne direction, du premier au dernier jour. Grâce à lui ces trois années de doctorat ont été à la fois intéressantes et agréables, je suis heureux de pouvoir profiter de ces quelques lignes pour lui exprimer toute mon estime et ma gratitude.

Je remercie également les membres du jury d'avoir porté leur attention sur mes travaux de thèse. Leurs conseils et leur remarques tant sur la forme que le fond ont été bénéfiques à l'amélioration de mes travaux, je leur en suis reconnaissant.

Je salue l'ensemble de l'équipe Image et Son du LaBRI et du service EMI de Gestorm avec qui j'ai pu travailler dans une bonne ambiance pendant mon doctorat. Je tiens à remercier tout particulièrement Benjamin Martin, ex-doctorant du LaBRI ainsi que Paul Fournier développeur à Gestform avec qui j'ai eu la chance de travailler et de faire plus ample connaissance tout au long de ces 3 années.

J'en profite également pour saluer le soutient moral apporté par mes plus proches connaissances : Sophie Bastard, Christophe Calmettes, Pierre-Marie Leguay et Asma Maachou.

Une pensée va pour mes grands-parents, grâce à qui j'ai eu la chance de pouvoir continuer mes études jusqu'au plus haut stade.

Enfin, je tiens à remercier profondément ma proche famille, mes frères et surtout mes parents dont la bienveillance et l'appui n'ont cesse de m'encourager depuis mon enfance dans chaque chose que j'entreprends. Ils m'ont permis de surmonter toutes les épreuves, même les plus difficiles. Je souhaite dédier ces trois années de travail à ma mère et mon père, Patricia et Bernard mes bien-aimés parents.

Résumé

Résumé

Ces travaux de recherche ont pour ambition de contribuer à la problématique de la classification d'images de documents. Plus précisément, ces travaux tendent à répondre aux problèmes rencontrés par des sociétés de numérisation dont l'objectif est de mettre à disposition de leurs clients une version numérique des documents papiers accompagnés d'informations qui leurs sont relatives. Face à la diversité des documents à numériser, l'extraction d'informations peut s'avérer parfois complexe. C'est pourquoi la classification et l'indexation des documents sont très souvent réalisées manuellement. Ces travaux de recherche ont permis de fournir différentes solutions en fonction des connaissances relatives aux images que possède l'utilisateur ayant en charge l'annotation des documents.

Le premier apport de cette thèse est la mise en place d'une méthode permettant, de manière interactive, à un utilisateur de classer des images de documents dont la nature est inconnue.

Le second apport de ces travaux est la proposition d'une technique de recherche d'images de documents par l'exemple basée sur l'extraction et la mise en correspondance de points d'intérêts.

Le dernier apport de cette thèse est l'élaboration d'une méthode de classification d'images de documents utilisant les techniques de sacs de mots visuels.

Mots-clefs

Classification d'images de documents ; recherche de documents ; application industrielle ; extraction de points d'intérêts ; recherche par l'exemple ; sacs de mots visuels

---

Document image retrieval and classification

Abstract

The aim of this research is to contribute to the document image classification problem. More specifically, these studies address digitizing company issues which objective is to provide the digital version of paper document with information relating to them. Given

the diversity of documents, information extraction can be complex. This is why the classification and the indexing of documents are often performed manually. This research provides several solutions based on knowledge of the images that the user has.

The first contribution of this thesis is a method for classifying interactively document images, where the content of documents and classes are unknown.

The second contribution of this work is a new technique for document image retrieval by giving one example of researched document. This technique is based on the extraction and matching of interest points.

The last contribution of this thesis is a method for classifying document images by using bags of visual words techniques.

Keywords

Document image clustering; document retrieval; industrial application; interest point extraction; query by example; bags of visual words

Table des matières

Introduction

La dématérialisation consiste à transformer un objet physique tel qu'un document papier en une image numérique. Les entreprises de dématérialisation souhaitent produire une valeur ajoutée à cette dématérialisation. La problématique globale est donc d'extraire des informations sur le contenu des documents dématérialisés afin d'en faciliter la recherche et l'accès au contenu. Par exemple, en plus de créer l'image d'un document, il est possible d'extraire le texte, les illustrations, les photos ou encore les tableaux qui le composent. De manière générale, face à la diversité des images de documents à traiter, il reste complexe d'extraire une information précise d'un document si l'on ne sait pas quel est le type de document que l'on est en train d'analyser. Par exemple, l'extraction du nom d'une personne sur une carte d'identité ne sera pas réalisée de la même manière que l'extraction du nom d'une personne sur une facture. Les travaux de recherche relatifs à cette thèse tendent à explorer de nouvelles techniques permettant de regrouper des documents par similarité de contenu. Notre ambition est donc de mettre en place, selon différents cas de figures, un ensemble de méthodes permettant de reconnaitre en amont les images des documents afin que chaque classe soit traitée de manière adéquate.

Le fil conducteur de cette thèse s'appuie sur un double postulat. Le premier est qu'il n'est pas envisageable de mettre en place un système de classification d'images de documents qui soit totalement automatique. Nous pensons donc qu'il est indispensable de proposer des solutions intégrant un utilisateur dans ce processus de classification. Le deuxième postulat est qu'il n'existe pas une tâche unique de classification d'images de documents mais plusieurs. Nous pensons donc qu'il faut aborder cette problématique en mettant en perspective les informations dont dispose l'utilisateur pour réaliser cette classification. En effet, les connaissances de l'utilisateur peuvent être de différents ordres. Est-ce que le nombre de classes composant la base d'images à traiter est connu ? Est-ce que des exemples de documents sont disponibles pour chacune des classes ? Si oui, en quelle quantité ? Est-ce que les documents aux seins des classes sont très variés ? Quelles sont les informations permettant de différentier deux images se ressemblant mais appartenant à des classes différentes ? Nous montrerons que selon le degré de connaissance d'un utilisateur sur les documents qu'il doit classer, il est important de mettre en place des chaînes de traitements, d'analyses et de comparaisons différentes.

Alors que la majorité des travaux relatifs à la classification d'images de documents utilisent des techniques basées sur l'extraction et l'analyse de texte [Seb02], nous souhaitons dans ces travaux de thèse étudier l'apport de techniques d'analyse d'images.

Dans le chapitre 1, les concepts généraux liés à la numérisation de documents seront présentés. Nous montrerons ensuite un ensemble d'applications afin de mettre en avant la grande diversité des objectifs de la dématérialisation. Une chaine classique de numérisation sera présentée étape par étape, ainsi que les principaux outils utilisés par les industries de dématérialisation. L'observation des différentes tâches exécutées lors de ces étapes nous permettra de montrer les principales difficultés que peut rencontrer une entreprise de

dématérialisation.

L'ambition de cette thèse est d'explorer plusieurs pistes permettant d'automatiser en partie (et donc d'accélérer) un processus de reconnaissance et de classification d'images de documents qui à l'heure actuelle est quasi-exclusivement réalisée manuellement. Les trois chapitres suivants détaillent une contribution différente et originale à la classification d'images de documents en fonction des informations que possède un utilisateur sur le corpus d'images à trier.

Dans le second chapitre nous nous plaçons dans le cas de figure où un utilisateur ne dispose d'aucune connaissance sur le corpus d'images à classer. Nous présenterons les caractéristiques habituellement utilisées pour décrire des images de documents ainsi que des techniques classiques de classification non-supervisées. Ceci permettra d'introduire notre proposition de méthode exploratoire permettant de présenter à l'utilisateur des suggestions de regroupement d'images. Le système ne prend aucune décision automatique, car sans avoir de connaissance il est presque certain qu'un grand nombre de documents seraient mal labellisés. L'objectif est donc simplement d'aider l'utilisateur à labelliser de nombreuses images de documents de manière plus rapide que si elles avaient été labellisées manuellement.

Dans le troisième chapitre, l'utilisateur a cette fois-ci une connaissance partielle du corpus d'images puisque nous considérons qu'il est en mesure de fournir un "document exemple" dont il recherche des exemplaires de même type dans une base composée d'un grand nombre de documents et de classes différentes. Nous proposons une méthodologie capable de résoudre cette problématique en utilisant un unique exemple du type de document qu'il souhaite retrouver. Afin de proposer une technique précise et robuste, le domaine d'application est limité aux documents semi-structurés. Pour une classe donnée, une partie significative des informations des documents doit se trouver à la même position d'un document à l'autre. Les techniques de recherche d'images par l'exemple ont déjà fait l'objet de nombreuses recherches dans le cadre applicatif des images naturelles [YSST07], [PAK10], [BL07], [Low04]. Malheureusement, comme l'explique les auteurs de [US09], l'application directe de ces techniques ne se sont pas montrées concluantes pour les images de documents. La principale contribution de ce chapitre consiste en la mise en place d'une adaptation aux images de documents, d'une technique initialement dédiée à la recherche d'images naturelles.

Enfin, dans le dernier chapitre nous présentons des techniques utilisables lorsqu'un utilisateur connait plusieurs classes dans la base qu'il souhaite trier. Il est également nécessaire que l'utilisateur puisse donner plusieurs exemples d'images de documents pour chaque classe qu'il souhaite retrouver. Dans un premier temps, nous présentons une technique de classification générique, basée sur un apprentissage supervisé. Cet apprentissage permet d'exploiter l'ensemble des exemples fournis par l'utilisateur. La première contribution de ce chapitre est d'appliquer les sacs de mots visuels [CDF+04] à la classification d'images de documents. Nous présenterons ensuite une variante de cette technique, inspirée des travaux de [RL09]. Cette technique est utilisée pour la reconnaissance de logos dans les images de documents, nous l'étendons à la classification d'images "complètes" de documents. Cette méthode permet d'obtenir une nette amélioration des performances de classification en termes de précision vis-à-vis des techniques de sacs de mots visuels.

Les travaux de cette thèse ont été effectués dans le cadre d'une convention CIFRE (Conventions Industrielles de Formation par la REcherche) entre le LaBRI (Laboratoire Bordelais de Recherche en Informatique) et Gestform, une entreprise de dématérialisation. L'ensemble des tests réalisés dans ce manuscrit ont été réalisés à partir de productions venant de la société Gestform. La technique d'aide à l'exploration dans une base d'images

de documents issue du chapitre 2 est couramment utilisée depuis plus d'un an en production. La recherche de documents semi-structurés (chapitre 2) a permis de répondre à une problématique qui n'avait jusqu'alors pas de solution industrielle : l'identification et l'extraction de papiers d'identité. Enfin, la technique du chapitre 4 a permis de constater que la classification d'images de documents automatique par apprentissage est possible avec une excellente précision pour certains types d'images, que cela soit réalisé par une analyse du texte ou de l'image.

Chapitre 1

La dématérialisation de documents en industrie

Dans ce chapitre, nous présentons le contexte d'application des travaux de cette thèse. Dans un premier temps nous introduirons les concepts généraux liés à la dématérialisation. Ensuite, nous présenterons plusieurs exemples d'applications afin de montrer la diversité des applications relatives à l'analyse d'images de document. Dans une troisième partie seront détaillées plus précisément les différentes étapes d'une chaîne de traitements telle qu'on peut en rencontrer dans un contexte industriel. Ceci permettra de mettre en évidence les tâches pouvant être améliorées grâce à l'analyse d'images de documents. En effet, comme nous le verrons, les besoins et problématiques autour de la numérisation de documents sont nombreux. Les travaux de cette thèse portent essentiellement sur la problématique de la reconnaissance et de la classification de documents. La solution industrielle commune à ces verrous scientifiques est d'analyser le contenu textuel. Malheureusement les techniques de reconnaissance de caractères ne sont pas toujours très performantes. Nous détaillerons principalement comment la présence de bruits divers, la variété des caractères utilisés, la complexités de mises en pages sont à l'origine de nombreuses erreurs de la part de logiciels d'OCR (Optical Character Recognition).

Nous conclurons enfin sur l'intérêt d'utiliser des techniques d'analyse d'images dans une tâche de classification d'images de documents (contrairement à la majorité des solutions industrielles utilisant exclusivement du texte) .

1.1 Quelques définitions

1.1.1 Dématérialisation

La dématérialisation peut désigner deux choses distinctes : l'action de dématérialiser ou le fait d'être dématérialisé. Dans l'ensemble du manuscrit nous utiliserons le mot dématérialisation au sens premier du terme : l'action de dématérialiser, c'est-à-dire le fait de transformer un objet physique en un ou plusieurs fichiers numériques.

Une fois le document physique transformé en document numérique, il devient beaucoup plus simple de le consulter, de l'envoyer ou de le partager, de le modifier, le supprimer, le dupliquer ou encore de l'archiver. Il devient également possible d'effectuer n'importe quel traitement que peut subir une donnée informatique, comme par exemple le versionnage, l'horodatage ou la signature numérique. L'horodatage permet d'associer une date de référence à des données. La signature numérique permet de certifier qu'un fichier est une copie conforme et qu'il n'a subit aucune modification. L'horodatage et la signature

numérique sont utilisés conjointement afin de certifier qu'un document est bien le même que le document d'origine à une date précise.

Pour une entreprise, un fichier numérique présente donc de nombreux avantages vis-à-vis d'une version physique. Par exemple, les données prennent moins de place : des millions de documents peuvent être stockés sur un serveur. Grâce à Internet et aux réseaux informatiques, une entreprise peut alors factoriser l'ensemble de ses documents et permettre aux personnes autorisées d'accéder, de partager, d'ajouter ou encore de modifier facilement des documents ainsi que des données qui y sont liées en accédant à une base de données, et cela depuis n'importe où et à n'importe quel moment.

La dématérialisation est généralement décomposée en 3 étapes : la numérisation, le stockage et la diffusion des documents. L'étape de numérisation est détaillée dans la sous-section suivante. Le stockage numérique consiste à entreposer les données pour pouvoir les conserver. Pour cela il existe de nombreux supports matériels tels que des CD-Rom, DVD, carte mémoire, clé USB, disque dur ou encore des serveurs. Le choix du support de stockage dépendra de la durée de vie nécessaire à la conservation (par exemple le CD-Rom a une durée variant de 5 ans à quelques dizaines d'années). Cela est suffisant par exemple pour stocker les documents des employés congédiés, car leur durée légale de conservation est de 3 ans. La fréquence d'utilisation des documents est également à prendre en compte. Si un accès aux documents est nécessaire fréquemment, on privilégiera par exemple, un stockage sur serveur. Enfin, la diffusion consiste à donner un accès aux données à l'utilisateur en lui distribuant une copie du support de stockage, en lui envoyant les fichiers par mail ou par FTP (File Transfer Protocol) ou encore en lui donnant un accès aux serveurs.

1.1.2 La numérisation de documents

Le processus

La numérisation consiste à transformer un signal analogique, par exemple du son ou un document papier, en signal numérique tel qu'un fichier MP3 ou une image TIFF. La numérisation de documents est faite à l'aide de scanners ou d'appareils photo. Un document papier est alors transformé en une image numérique composée elle même d'un ensemble de pixels. Deux procédés sont utilisés pour transformer le signal analogique en signal numérique : l'échantillonnage et la quantification.

L'échantillonnage correspond à un découpage temporel ou spatial du signal. Dans le cas d'une image réelle (un signal 2D), il existe un nombre infini de coordonnées auxquelles la couleur puisse être prélevée. Cet échantillonnage correspond à la résolution de l'image numérique : plus la résolution est élevée, plus le nombre de pixels utilisés pour décrire l'image sera grand. Et inversement, plus la résolution est basse, plus le nombre de pixels utilisés sera faible. Par exemple, pour un document de format A4, soit : 210 x 297 mm, si la résolution de la numérisation est fixée à 300 PPP (Pixels Par Pouce, ou DPI - Dots Per Inch), l'image sera composée d'environ 2480 x 3508 pixels. Si la résolution est divisée par deux, il y aura deux fois moins de pixels utilisés pour décrire chacune des deux dimensions du signal, soit 4 fois moins de pixels au total.

La quantification consiste à sélectionner un ensemble fini de valeurs que peut prendre le signal. Une image réelle peut contenir une infinité de variations de couleurs, cependant les pixels de l'image numérique eux, ne pourront contenir qu'un ensemble fini de valeurs. Par exemple, une quantification utilisant 24 bits permet de stocker : $2^{24} = 16777216$, soit environ 16 millions de couleurs. Pour stocker une image en niveaux de gris, on utilise généralement un codage sur 8 bits, permettant de stocker 256 nuances de gris. Pour les images en noir et blanc, 2 bits sont suffisants, car seules deux couleurs sont utilisées.

Les scanners

Les outils utilisés pour numériser les documents sont des scanners. Les scanners les plus couramment utilisés par des particuliers sont des scanners à plat, composés d'une vitre. Ces scanners sont rarement utilisés en industrie car le temps de manipulation est long. Il existe également des scanners verticaux, assimilables à des appareils photos. Ces scanners sont utilisés pour des documents de très grandes tailles ou des documents fragiles. Les scanners les plus couramment utilisés pour la dématérialisation de documents industriels sont les scanners à défilement. Il existe de nombreuses marques de scanners, chacune ayant ces avantages et inconvénients. On peut citer par exemple Canon. Le modèle X10C permet une numérisation "multistream", l'image peut être numérisée simultanément en noir et blanc en niveaux de gris et en couleur. Les modèles 9050 et 9080 sont utilisés pour numériser tout types de documents A3 ou A4. En effet, certains documents dont l'épaisseur est très faible, tels que les copies carbones, sont complexes à numériser. On peut également citer les scanners Kodak I1420 et Fujitsu 6670A, qui sont utilisés pour numériser des documents de tailles diverses. Au lieu de poser une pile de documents, les rouleaux du scanner tournent en permanence et les documents sont insérés les uns après les autres. Ceci est nécessaire pour des prestations telles que la numérisation de notes de frais car les documents ont des formats très variables : tickets de métro, d'avion, de train, de restaurant, etc.

Les fonctionnalités fournies par la plupart des scanners sont les suivantes : détection automatique du format du papier, détection de doubles feuilles par capteur ultrason, réglage de la luminosité et du contraste manuel ou automatique, numérisation en recto-verso, redressement de l'image (de la page, pas du contenu). À noter que le réglage de la luminosité et du contraste sont généralement fait en manuel plutôt qu'en automatique. Les scanners tels que le Fujitsu 6670A sont de plus en plus utilisés car, tout comme les scanners Bell & Howell de la série Spectrum, ils sont utilisables avec une option VRS ("Virtual ReScan") embarqué. Le logiciel VRS est un outil développé par la société Kofax, permettant d'améliorer la qualité des images de documents. Il permet par exemple le redressement de la page mais également du contenu, le recadrage et le remplissage des bordures, le remplissage de trous, la détection du sens de lecture, la suppression de pages blanches, le réglage automatique du contraste et de la luminosité, le lissage ou la suppression de la couleur de fond et la création de tramage pour la reproduction fictive de niveaux de gris. La suppression de pages blanches n'est que rarement utilisée car des documents risquent d'être supprimés à tord (beaucoup de fausses détections). La détection automatique du contraste et de la luminosité sont par contre plus performantes que celles proposées par les scanners classiques. À noter également que le VRS propose une fonction d'apprentissage, permettant d'apprendre que certains réglages sont à appliquer sur certains documents. Lorsque le logiciel reconnait le document, les mêmes réglages que ceux appris sur l'image utilisée pour l'apprentissage sont utilisés. La figure 1.1 illustre l'amélioration apportée par l'utilisation de VRS.

1.1.3 Gestion et analyse du contenu des documents

La figure 1.2 présente un exemple classique de GED (Gestion Électronique de Documents) composée de 3 étapes (numérisation, stockage et diffusion). Deux étapes optionnelles peuvent être ajoutées : le traitement et l'indexation des images. Le traitement des images peut permettre, par exemple, d'éliminer les pages que l'on ne souhaite pas stocker (par exemple les pages blanches). Cette étape peut également servir à faire un certain nombre de prétraitements visant à améliorer la qualité des images telles que le redres-

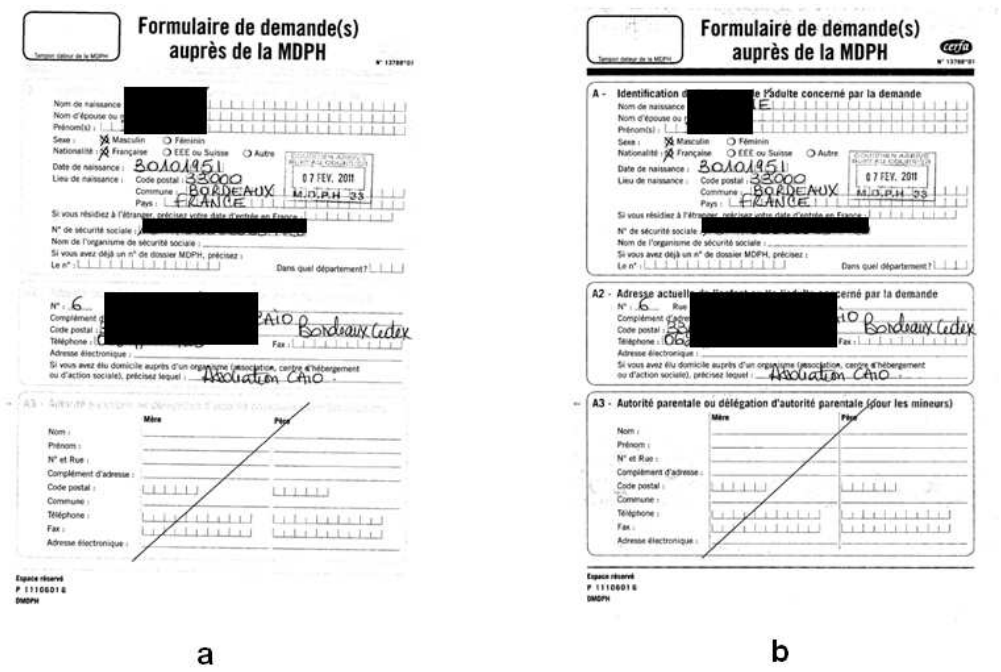


FIGURE 1.1 – Amélioration de la qualité d'image grâce au VRS. L'image a) est la sortie après numérisation, sans utilisation du VRS. L'image b) a été traitée par le VRS. On remarque que le contenu de l'image a été redressé et que la binarisation est sensiblement meilleure.

sement ou le débruitage. L'indexation permet d'identifier le document en lui associant des informations qui permettront de le retrouver plus facilement, ces informations sont également appelées métadonnées. Ces deux étapes peuvent bien évidemment venir toutes les deux avant ou après le stockage et être effectuées manuellement, automatiquement ou semi-automatiquement.

L'OCR est une technique permettant de retranscrire le texte typographié contenu dans une image. Les logiciels d'OCR les plus utilisés en industrie sont FineReader de la société ABBYY et Omnipage de la société Nuance Communications. Dans la partie 1.5 est présenté un ensemble de limites de ces systèmes d'OCR. Sur des documents de bonne qualité et comportant principalement du texte typographié tels que des livres, les performances de l'OCR sont très bonnes. Certains OCR incluent des traitements tels que le redressement des images, la détection de la langue, le nettoyage de l'image, la segmentation de doubles pages, etc.

La retranscription de texte manuscrit, se fait avec des logiciels d'ICR (Intelligent Character Recognition). Les entreprises tels que ABBYY et Omnipage proposent également des solutions logicielles d'ICR. Cependant les performances de l'ICR (situées entre 80% et 90% de reconnaissance) sont bien moins bonnes que celles de l'OCR (située entre 95% et 99%) car le problème de détection de manuscrit est plus complexe. De plus, les résultats des logiciels d'ICR sont exploitables, pour le moment, uniquement pour des écritures avec caractères détachés.

L'objectif de la LAD (Lecture Automatique de Documents) est de lire une portion d'image précise afin d'en extraire une ou plusieurs informations. Par exemple, on souhaite extraire le mot figurant à droite de la chaîne de caractère "NOM :" car on sait que figurera à cet endroit le nom de la personne. Un outil de LAD performant permet de définir des zones

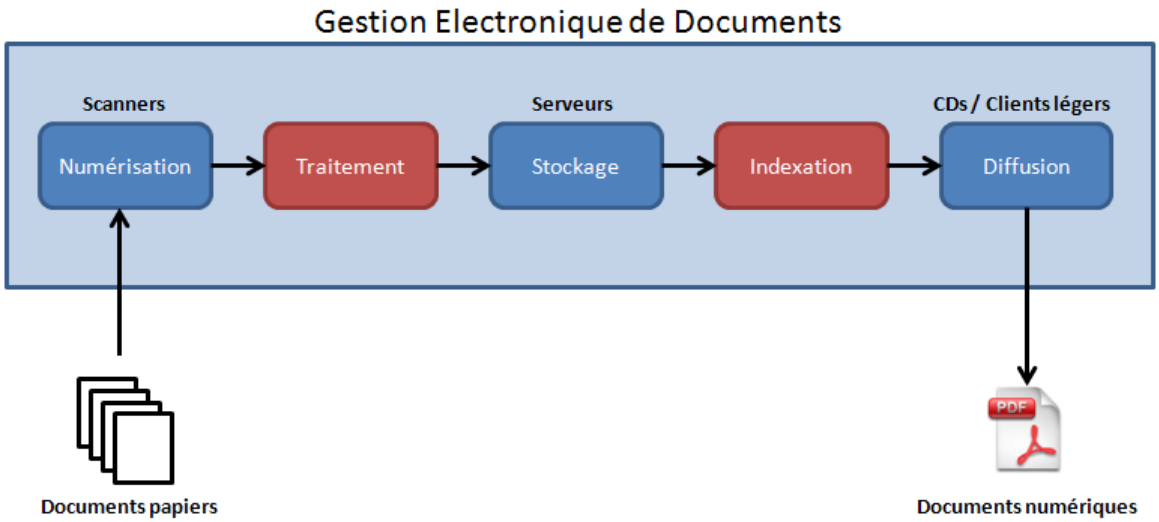


Figure 1.2 – Une chaine de traitement pour la gestion électronique de documents. En bleu figurent les 3 opérations basiques de la dématérialisation et, en rouge, figurent 2 opérations optionnelles pour améliorer la qualité des documents et extraire des informations.

de manière absolue ou relative par rapport à d'autres zones. Un moteur de LAD utilise un OCR pour lire les informations de type textuel. La RAD (Reconnaissance Automatique de Document) permet de différencier différents types de documents. La RAD est généralement combinée avec la LAD, car suivant le type de document que l'on a reconnu, on ne lira pas les mêmes informations aux mêmes endroits et inversement, selon les information lues, le type de document peut être reconnu. ABBYY propose une solution de LAD/RAD pour les formulaires. Un expert créé des modèles de formulaires à l'aide du logiciel FlexiLayout 8.0. Chaque modèle est composé d'une ensemble de blocs correspondants aux informations à lire. Un exemple est illustré sur la figure 1.3. Les images à analyser sont ensuite traitées par le logiciel FlexiCapture 8.0 qui teste une liste de modèles et attribue à chaque image un modèle en donnant un niveau de confiance de reconnaissance. Un exemple d'application est détaillé sur la figure 1.4.

1.1.4 Terminologie : image, document, feuille, page

Dans le cadre de ce manuscrit, nous considérons qu'un document est composé d'un ensemble de feuilles et que chaque feuille possède deux pages (recto et verso). Une image correspond alors à une page d'un document. La plupart du temps (et sauf indication contraire dans le manuscrit), seule la 1ère page du document est analysée pour faire la classification ou la reconnaissance du document.

Cependant, il arrive également dans certains cas qu'une image contienne plusieurs documents. C'est le cas par exemple pour les notes de frais où les employés disposent plusieurs tickets sur une page, ou encore sur les documents d'identité où les utilisateurs peuvent parfois disposer plusieurs documents d'identité sur une même page avant de les numériser.

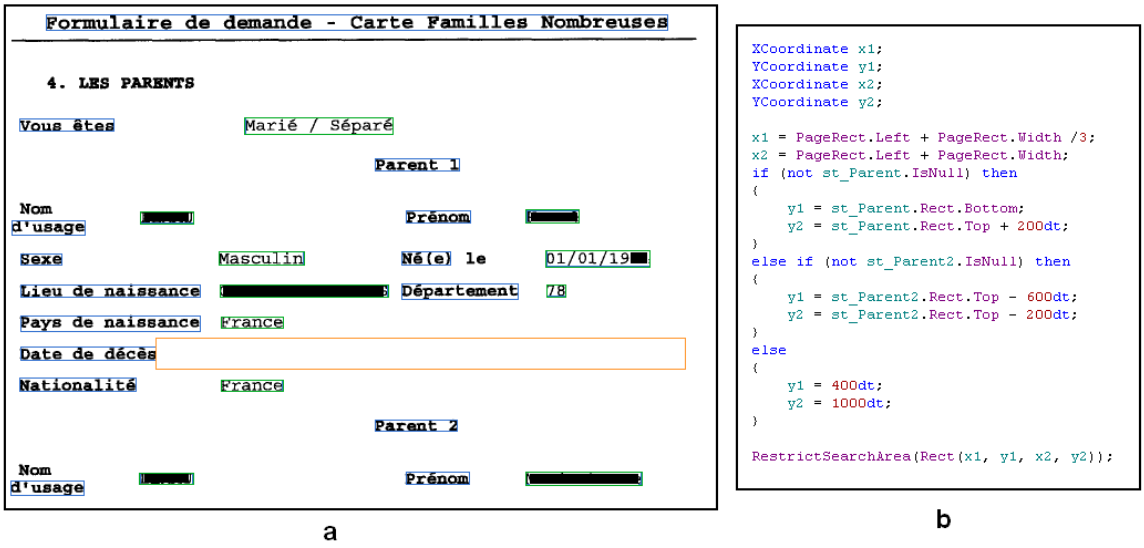


FIGURE 1.3 – Création d'un modèle de LAD avec FlexiLayout 8.0. a) En bleu et en vert figurent respectivement des blocs de textes statiques et dynamiques. Ces blocs sont définis par le modèle. En orange figure un bloc de texte dynamique qui n'a pas été identifié. b) La position de certains blocs peut être déterminée relativement par rapport à un ou plusieurs blocs par le code.

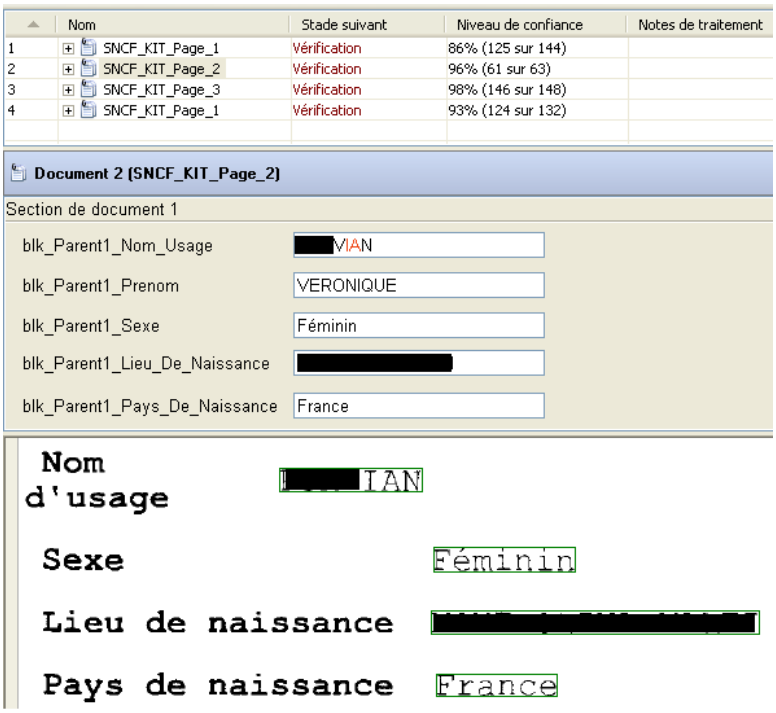


FIGURE 1.4 – Application de la RAD avec FlexiCapture 8.0. Pour chaque image, le logiciel compare l'image aux modèles de sa base de connaissances et choisit celui qui le correspond le plus. On peut voir l'ensemble des blocs qui ont été détectés. Le résultat de l'OCR est également affiché avec en rouge, les caractères dont la transcription est jugée incertaine.

1.2 Exemples d'applications à dimension industrielle

Les applications liées à la dématérialisation d'images de documents sont diverses. Nous allons passer en revue un ensemble non exhaustif d'exemples dont les objectifs sont variés.

Le traitement de courriers

Les travaux de la thèse de Gaceb [Gac09] se situent dans le cadre du tri automatique de documents et de courriers postaux. Une des applications est de reconnaitre automatiquement une adresse sur une enveloppe afin de trier automatiquement le courrier postal.

C'est une problématique complexe parce que l'adresse peut être écrite manuellement ou de manière typographiée et à des positions différentes sur le document. Une partie de ces recherches porte sur la reconnaissance de courriers postaux et plus précisément sur la segmentation et la localisation de zones d'intérêts telles que le bloc d'adresse. La figure 1.5 illustre cette problématique.

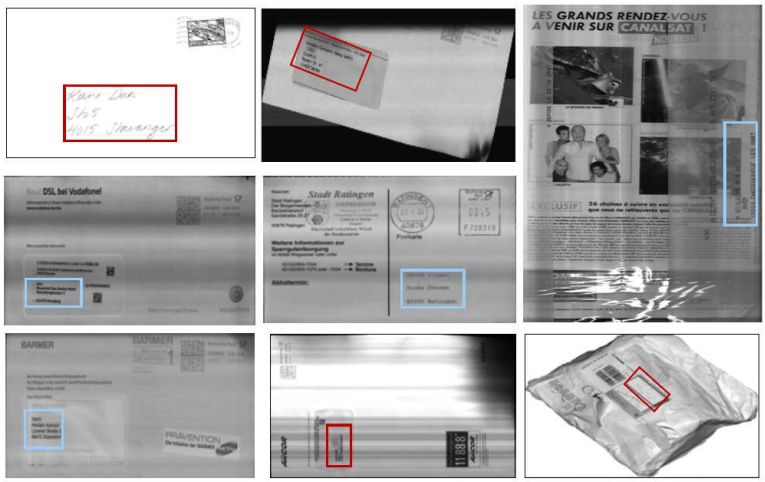


Figure 1.5 – Localisation du bloc d'adresse dans les courriers postaux. La complexité de la tâche vient de la variation de la position des enveloppes et de la représentation de la zone contenant l'adresse. L'illustration est issue de [Gac09].

La classification de factures

L'objectif des auteurs de [DFG03] est de classer un ensemble de documents parmi 9 modèles de factures connues. L'illustration 1.6 présente un exemple de chaque classe. La classification des factures est une étape importante dans la dématérialisation, car chaque classe de factures est ensuite analysée sur la base d'un modèle qui lui est propre. On effectue ici la reconnaissance du type de document pour savoir où sont les informations à lire.

Cette problématique est plus simple que la précédente, car les factures sont des documents très structurés, c'est-à-dire que la plupart des informations sont présentes à des positions qui ne varient pas d'une image à l'autre.

Les auteurs de [HBBC08a] se sont également intéressés à la classification de factures. Les auteurs de [HBBC08b] proposent un système de classification où le contenu des documents sont modélisés avec des graphes. Les nœuds sont des mots-clés définis à l'avance et les arrêtes représentent la distance entre ces nœuds. La base de documents est divisée

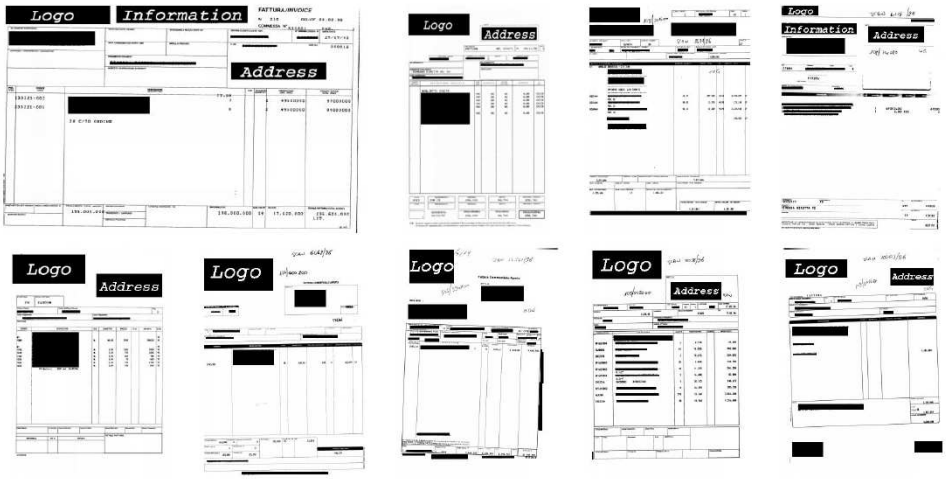


FIGURE 1.6 – Classification de formulaires. Un exemple de chacun des 9 types de formulaires sont représentés. Illustration issue de [DFG03].

en deux parties : 324 documents sont utilisés pour l'apprentissage et 169 documents pour les tests, le tout réparti en 8 classes de factures. En faisant varier différents paramètres internes à la technique, les auteurs obtiennent un pourcentage de reconnaissance moyen compris entre 97,63% et 99,40%.

Le traitement de chèques de banque

La société A2IA propose un outil de lecture automatique de chèques bancaires [GAA+01]. Pour les entreprises, les chèques bancaires sont complexes à traiter automatiquement car leur visuel peut être très différent d'une banque à l'autre. La figure 1.7 montre des exemples de chèques traités par les auteurs. De plus, selon leur provenance, les chèques peuvent avoir été remplis de manière manuscrite ou typographiée. Il est souvent complexe de lire le montant, l'émetteur, le bénéficiaire, etc. Il peut également être intéressant de reconnaitre la position de la signature et de la contrôler.

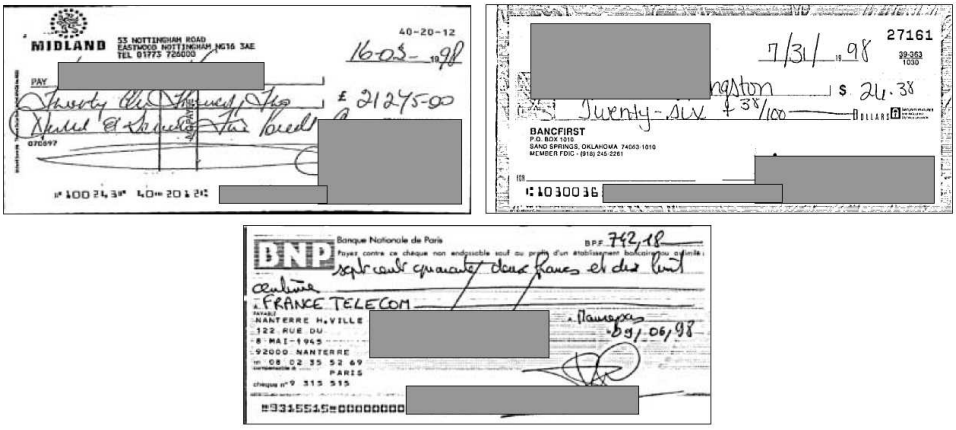


FIGURE 1.7 – Analyse de chèques bancaires, exemples de 3 chèques. Illustration issue de [GAA+01].

Dans cette publication est présenté un algorithme permettant la lecture du montant des chèques, ces montants pouvant être typographiés ou manuscrits. La première étape

consiste à trouver la zone contenant le montant du chèque. Ensuite, il faut détecter si ce montant est écrit de manière manuscrite ou typographiée. Selon les auteurs, cette opération est exécutée de manière fiable sur 95% des chèques. Dans tous les cas, la zone contenant le montant est nettoyée puis segmentée en caractères. Quatre OCR indépendants sont ensuite combinés pour reconnaitre le texte.

L'identification de formulaires

Shin*et al.* [SDR01] proposent une technique de recherche d'images de documents par similarité visuelle. Leur but est de construire un classifieur permettant de labelliser les documents en fonction de leur type. Selon les auteurs, classifier les documents selon leur type est un moyen efficace permettant d'améliorer la recherche de documents.

Selon une étude basée sur la perception des utilisateurs détaillée dans l'article, l'utilisation du contenu visuel de l'image permet de faire une classification des images telle que le ferait des humains.

Les auteurs de [APCU09] se sont intéressés à l'identification de formulaires très similaires. La complexité de cette tâche vient du fait qu'il faut faire la différence entre des images qui sont très similaires mais appartenant à des classes différentes. La figure 1.8 illustre un ensemble formulaires visuellement très similaires.

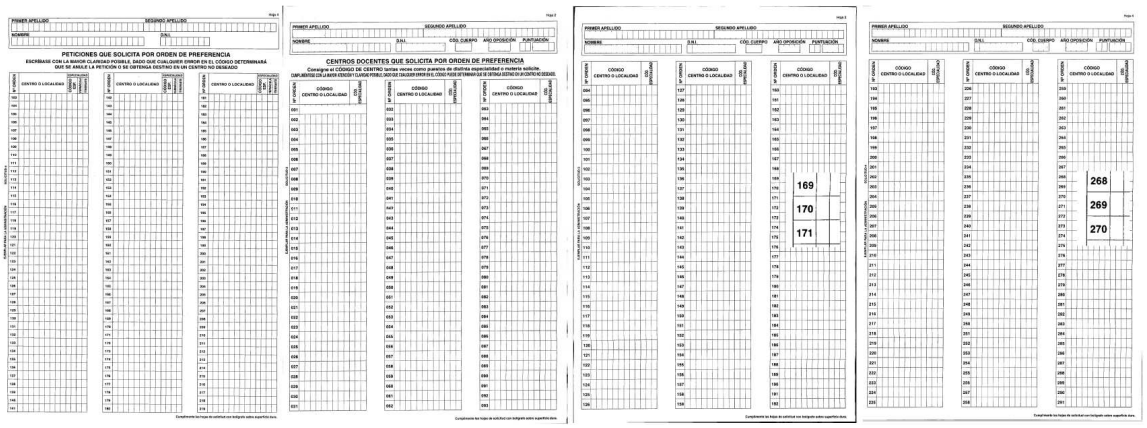


Figure 1.8 – Classification de formulaires très similaires, exemples de 4 formulaires. Illustration issue de [APCU09].

Analyse des documents techniques

Adam *et al.* [AOC+01] proposent une technique permettant la lecture de caractères dans des plans de réseaux de l'opérateur de télécommunications France Télécom tel que celui présenté sur la figure 1.9.

La complexité de l'application vient du fait que les caractères qu'il faut lire peuvent avoir une orientation et une échelle quelconque et qu'ils peuvent également toucher des éléments graphiques. Les auteurs de [CK11] tentent de lire du texte curviligne ayant une orientation et une taille variable sur des cartes géographiques.

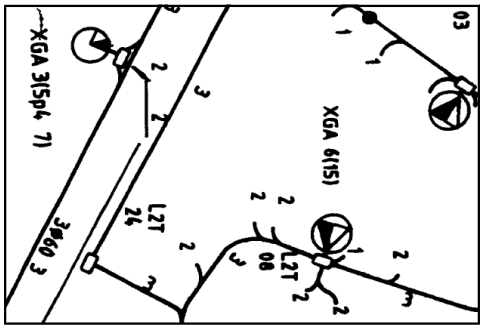


FIGURE 1.9 – Analyse des plans de réseaux de France Télécom. Illustration issue de [AOC+01].

Indexation d'images de documents patrimoniaux

Il existe également des applications dans le domaine de la culture permettant d'aboutir à une dématérialisation de documents anciens. Le CESR [1] (Centre d'Études Supérieures de la Renaissance) effectue des recherches dans des domaines tels que l'histoire de l'art et des sciences, la littérature française et européenne, la musicologie et la philosophie. Des chercheurs en analyse d'image du laboratoire d'informatique de Tours ont collaboré avec le CESR pour fournir des outils d'accès au contenu des documents et mettre en place les "Bibliothèques Virtuelles Humanistes" [RLDB07]. Les auteurs proposent une interface graphique évoluée permettant à un utilisateur non expert en traitement d'images de manipuler des descripteurs simples (taille, surface, position, ...) afin d'exprimer ses connaissances métiers sur les images de documents numérisés. Les interactions entre l'utilisateur et le logiciel aboutissent à l'édition de scénarios permettant de segmenter et labelliser le contenu d'images de documents.

La figure 1.10 présente des exemples de documents patrimoniaux provenant du CESR.

FIGURE 1.10 – Quelques exemples d'images de documents historiques, provenant de [RLDB07].

1. http ://cesr.univ-tours.fr

L'analyse de registres de naissances, de mariages et d'état civil

Les auteurs de [CL03] proposent une méthode générique pour analyser la mise en page de documents. Cette technique est appliquée à différents types de documents et notamment aux formulaires d'incorporations militaires du XIXe siècle stockés aux archives de la Mayenne et des Yvelines. L'objectif est de reconnaitre les différentes parties du document car une zone confidentielle de l'image doit être supprimée afin de mettre en ligne ces images.

L'intérêt de la méthode réside dans sa généricité car elle peut être utilisée pour décrire une grande variété de structures de documents. En effet la méthode a été utilisée pour l'analyse de partitions musicales [CC94] ou encore la reconnaissance de formules mathématiques (MatRead) [Coü01]. Cependant les documents traités par DMOS doivent posséder une structure forte, stable et descriptible par un ensemble de règles définies par un utilisateur expert. Cette méthode a été utilisée pour reconnaitre la structure de 88 745 pages des archives départementales de la Mayenne et des Yvelines [Coü06]. Sur l'ensemble de ces documents 1,18% ont été rejetés car ils étaient trop dégradés et 98,82% ont été correctement détectés. Aucune fausse détection n'a été faite. L'auteur considère que la structure a été correctement trouvée si elle est à 1mm près de la structure réelle.

La segmentation de bande dessinée

Les auteurs de [RTBO12] proposent une méthode pour extraire des cases et des bulles de dialogues des images de bandes dessinées. L'objectif est de permettre à un utilisateur d'effectuer des recherches sur des informations autres que les simples métadonnées correspondant au nom de l'auteur ou de l'ouvrage.

La figure 1.11 présente un exemple d'image de document sur laquelle travaillent les auteurs.

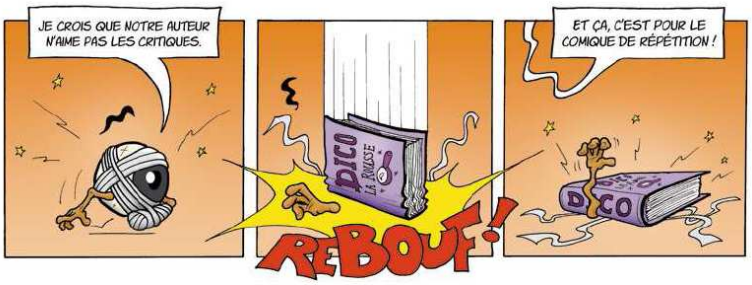


Figure 1.11 – Un exemple de bande dessinée, provenant de [RTBO12].

Reconnaissance de mots-clés d'un texte manuscrit

Les auteurs de [TCTH12] proposent une méthode permettant de reconnaitre si un mot-clé prédéfini est contenu dans un document manuscrit. Les courriers manuscrits reçus par les entreprises sont parfois nombreux et de natures diverses : des demandes de remboursement, des lettres de changement d'adresse, etc. La reconnaissance de mots-clés peut permettre ainsi d'identifier la nature de la demande et d'orienter le courrier vers la bonne personne voir même de traiter automatiquement la demande.

L'analyse de manuscrit est une tâche complexe due au fait de la grande diversité d'écriture, à la non régularité de l'écriture, mais également au fait que certains caractères se touchent et que l'écriture est souvent penchée.

1.3 Présentation du contexte industriel de la thèse

Le domaine d'application de cette thèse est lié à la nature même des documents (que nous appellerons documents industriels) envoyés aux sociétés de numérisation. Dans les faits, la majorité des catégories des documents traités dans la littérature sont des documents anciens, manuscrits, etc. Or, il se trouve que ces catégories de documents sont numérisés par des sociétés spécialisés. La majorité des sociétés de numérisation concentrent leur activité sur la dématérialisation de documents industriels. Sans être exhaustifs, nous pouvons citer comme types de documents : les documents de ressources humaines (des bulletins de salaire, des contrats d'embauche, des fiches de médecine du travail, etc.), les notes de frais (des tickets de restaurants, des factures d'hôtel, des reçus de taxi, etc.), les documents techniques, les factures, etc. Quelques exemples d'images sont visibles sur la figure 1.12.

FIGURE 1.12 – Quelques exemples d'images de documents à traiter. La couleur du cadre indique la classe des documents.

Une entreprise de dématérialisation reçoit des documents papiers de sociétés souhaitant transformer des documents papiers en fichiers numériques. Cette tâche peut avoir un degré de complexité variable car les demandes de prestations industrielles sont de différentes natures. La quantité d'information à extraire des documents est dépendante de la tâche. Nous allons présenter trois exemples de prestations différentes.

1. Dans le cas le plus simple, la demande est de faire une simple numérisation des documents. Dans ce cas, aucune information n'est à extraire. C'est le cas des documents d'archive numérisés à titre de conservation.
2. D'autres prestations sont composées de documents ayant tous la même mise en page. Par exemple des formulaires d'enquêtes de satisfaction. L'objectif sera de lire

les informations présentes à des positions prédéfinies dans l'image. On utilise pour faire cela un logiciel de LAD qu'il faut configurer pour chaque type de document.

3. Enfin, un cas complexe se présente lorsque la prestation est composée de nombreux documents aux visuels variés. Dans ce cas, il faudra dans un premier temps extraire des informations telles que la mise en page, la présence de logos, analyser le champ lexical et toutes autres informations nécessaire à l'identification du type de documents. Puis, dans un second temps, en fonction de son type, il faudra extraire les informations définies par le client. Pour cela on utilise un logiciel de RAD combiné à un logiciel de LAD.

La figure 1.13 présente en détail un exemple de prestation complexe (cas 3) d'une société de dématérialisation. Elle se compose de 12 étapes principales. Les chiffres sont donnés à titre indicatif et sont des moyennes calculées sur un ensemble de prestations gérées par la société Gestform.

1. Réception. Les documents sont livrés par camion à une fréquence de 1 camions tout les 1,5 mois. Chaque camion contient environ 170 cartons, contenant eux-mêmes 1200 boites représentant un total de 150000 documents. Ces chiffres sont relativement variables.
2. Préparation. Les documents sont sortis des boites. Un document est composé de plusieurs feuilles reliées entre elles par une agrafe. Pour pouvoir numériser les documents, il est nécessaire d'enlever les agrafes. Un opérateur dégrafe environ 800 pages par heure. Pour gagner du temps, les documents ne sont pas numérisés un à un. Les feuilles n'étant plus reliées par des agrafes, les documents sont superposés les uns sur les autres en intégrant des feuilles au visuel particulier (appelées feuilles "patchs") afin de pouvoir identifier le début et la fin de chaque document numérisé. La figure 1.14 représente deux exemples de feuilles "patchs" qui sont de simples feuilles avec un symbole visuel reconnaissable facilement numériquement. Ces feuilles permettent de séparer les documents.
3. Numérisation. Toutes les pages de tous les documents sont numérisées. En règle général les clients demandent une numérisation noir et blanc à 300 dpi. Certaines prestations sont numérisées en couleur ou encore à 200 dpi. Les scanners performants numérisent 1400 pages par heure. Le résultat numérique de chaque image est contrôlé afin de garantir que l'image ait été correctement numérisée. Selon la demande du client ce contrôle est fait de manière plus ou moins précise.
4. Contrôle du nombre d'images. Cette étape consiste à vérifier qu'aucune page de document n'est manquante ou n'est en double après la numérisation. Cela peut arriver à cause d'un problème matériel où plusieurs pages ont été numérisées ensemble ou simplement par inadvertance. Il peut également arriver que certaines feuilles soient numérisées en deux fois ou qu'un paquet complet de feuilles soit numérisé deux fois.
5. Suppression des pages "blanches". Les pages ne contenant pas d'information ne sont généralement pas conservées (pour diminuer le coût du stockage). Cette opération n'est pas forcément simple, car une page sans information est rarement toute blanche. Il y a généralement du bruit, des tâches, du texte visible par transparence, du texte entièrement rayé, etc. Des exemples de pages "blanches" sont présentées sur la figure 1.15. Ce traitement est fait manuellement.
6. Rotation. Les documents sont tournés de manière à être dans le sens de lecture (portrait ou paysage). Ici sont effectuées des rotations qui sont multiples de 90 degrés. Cette rotation est faite soit manuellement soit automatiquement en effectuant l'OCR

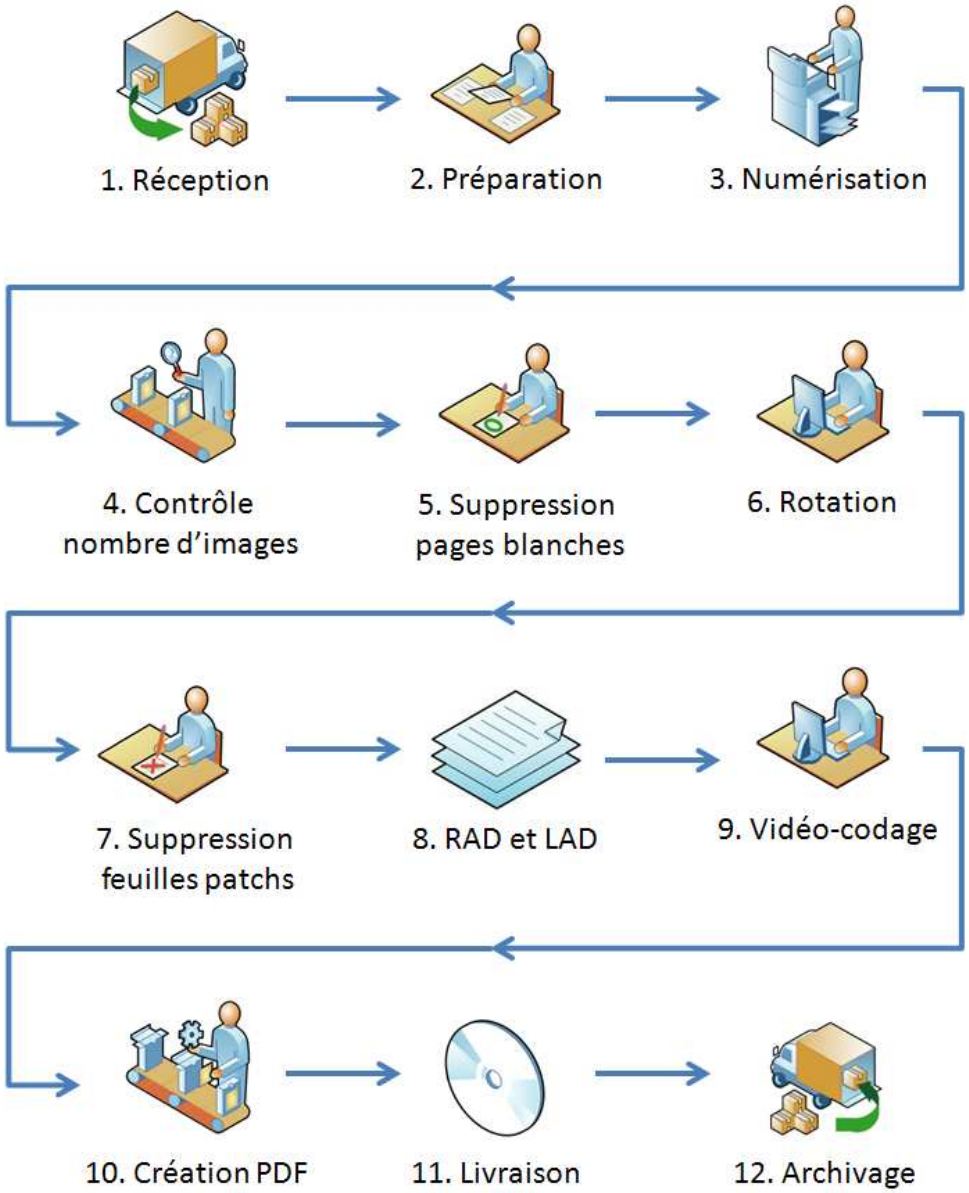


FIGURE 1.13 – Exemple d'une prestation de dématérialisation décomposée en 12 étapes principales.

selon les 4 orientations et en gardant l'orientation ayant le plus de mots reconnus (orientation qui correspondra alors au sens de lecture).

7. Suppression des feuilles patchs. Les feuilles introduites lors de la préparation sont détectées automatiquement par un logiciel *ad hoc*. Les pages sont alors regroupées pour reformer chacun des documents originaux.

8. RAD et LAD. Dans cette prestation, les documents peuvent avoir des visuels différents. En fonction du type de document, l'information à extraire ne sera donc pas présente au même endroit. Une fois le type de document reconnu grâce au moteur de RAD, l'information peut être lue grâce à la LAD. La LAD est utilisée pour lire des informations telles que le nom de la personne, le montant de la facture, etc., dans des zones spécifiques du document.

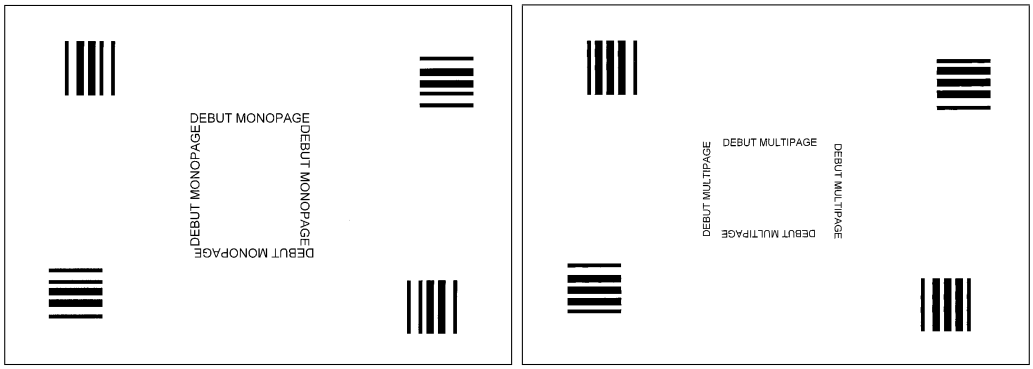


Figure 1.14 – Deux exemples de feuilles "patchs".

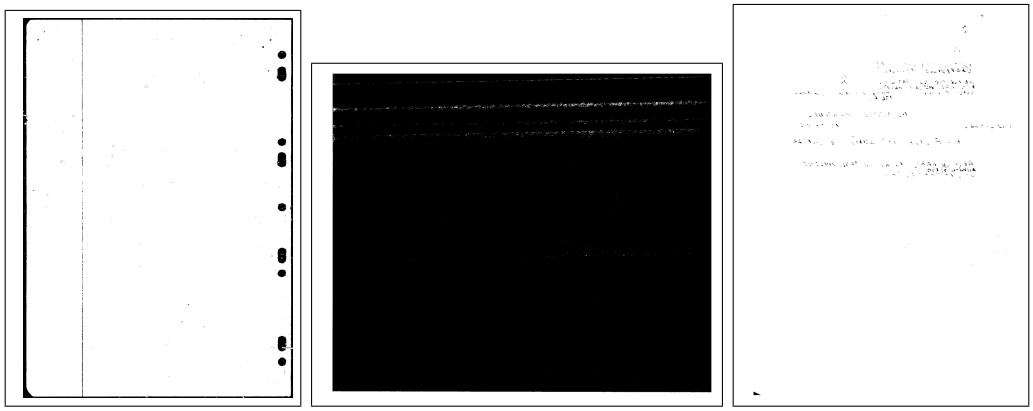


Figure 1.15 – Trois exemples de feuilles "blanches" ne contenant aucune information.

9. Vidéo-codage. Si la lecture automatique du document échoue ou n'est pas possible pour certaines parties, la zone est lue par un opérateur et saisie manuellement. Le contrôle est lui aussi fait de manière manuelle et les corrections sont faites par vidéocodage (retranscription manuelle des informations).
10. Création PDF. À partir des images et du texte, un PDF est créé pour chaque document.
11. Livraison. Les fichiers PDFs sont livrés au client soit par CDs soit par FTP.
12. Archivage. La plupart du temps, les documents papiers sont stockés pour valeur juridique. Ils sont soit stockés par l'entreprise de dématérialisation, dans des entrepôts, soit par une autre entreprise spécialisée.

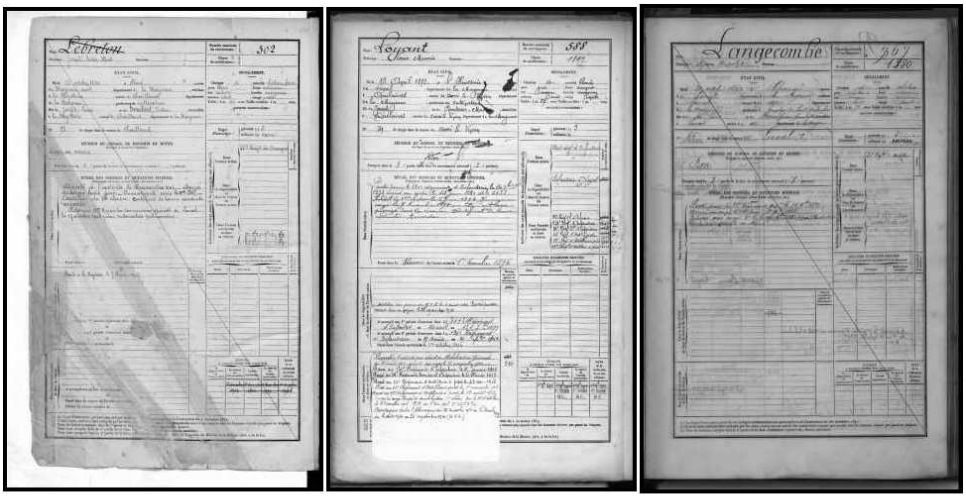
1.4 Besoins et problématiques autour de la numérisation

Dans l'exemple de chaîne de dématérialisation présentée ci-dessus, de nombreuses tâches peuvent être automatisées ou semi automatisées grâce à l'analyse et au traitement d'image. Nous les présenterons dans deux sous-sections :

- Le contrôle du résultat de la numérisation (étape 3) et du nombre d'images (étape 4), la correction de certains défauts tels que l'inclinaison des images (étape 6).
- La reconnaissance de page sans informations importantes (étape 5) et du contenu des documents, qu'elle soit automatique (étape 8) ou manuelle (étape 9).

Contrôle de la numérisation et corrections

L'étape de numérisation est source de nombreux problèmes. Tout d'abord se présentent des problèmes liés à la variété des supports physiques des documents. Les plans cadastraux sont de grandes tailles et le traitement de données volumineuses (des images de plusieurs gigaoctets) peut s'avérer complexe. L'acquisition d'un document doit se faire en plusieurs fois, se pose alors le problème de la reconstruction de l'image complète à partir de plusieurs images. La numérisation de livres est également délicate à cause de la présence de la reliure. Les documents anciens présentent de nombreuses difficultés dues à leur âge : les feuilles sont tachées, brulées, froissées, etc. On peut également citer des prestations particulières comme la numérisation de registres militaires qui comportent de multiples retombes et collages, voir figure 1.16.



a b c

FIGURE 1.16 – Registres militaires : a. le document a été déchiré, b. le document a des tâches et une retombe, c. l'encre a pénétré le papier. Images issues de [Coü06].

Une grande problématique liée à la numérisation de documents est de contrôler la qualité des images après numérisation. Comme on l'a vu précédemment il peut apparaitre de nombreux défauts : pliure, transparence, page manquante, page en double, défaut de luminosité, présence de tâches, ratures, image tournée, tronquée, etc. Il est important de détecter la présence de ces défauts soit pour prévenir que l'analyse automatique risque d'être compromise à cause de la mauvaise qualité du document, soit, lorsque ceci est possible, pour corriger ces défauts.

Il arrive que certaines tâches fonctionnant bien pour la plupart des documents échouent dans certains cas particuliers. La binarisation est généralement exécutée automatiquement par le scanner, en embarqué. Cependant, les documents avec de nombreuses couleurs, des dégradés, ou avec des couleurs inversées seront mal binarisés. Un autre exemple est le redressement automatique des images. Certains algorithmes sont fonctionnels sur des documents contenant une majorité de contenu textuel, mais ils ne fonctionnent pas ou mal sur des images contenant des photos ou des dessins comme les documentations techniques. Pour chaque cas d'application particulier seront alors utilisés des algorithmes *ad hoc*.

Le contrôle du nombre d'images consiste à détecter que certaines images ont été numérisées en double. Pour cela il faut faire une détection de doublon. On peut utiliser des techniques de recherche d'image ("image retrievial") pour effectuer cette tâche, comme par exemple celle de [Hul98] ou [NKI06]. La complexité réside dans le fait que les images dou-

blons ne sont pas strictement identiques : il peut y avoir des différences dans la position des blocs, l'orientation ou le contraste global de la page, etc. car les images ont été numérisées deux fois mais pas de manière totalement identique. Pour détecter qu'il n'y a pas de page manquante, une solution consiste, lorsque c'est possible, à analyser la pagination des images comme le propose les auteurs de [DM08].

Analyse et reconnaissance du contenu des documents

L'analyse et la reconnaissance des images de documents peut avoir plusieurs objectifs. Nous distinguerons trois objectifs distincts : la recherche, la classification et l'indexation de documents.

- la recherche de documents consiste à rechercher un ou plusieurs documents dans une base répondant à des critères donnés par l'utilisateur. On se focalise alors sur un type en particulier d'images. L'utilisateur s'attend alors à ce que le système lui renvoie un ensemble de documents pertinents.
- L'indexation des images, permet d'associer à chaque image un ou plusieurs mots-clés en fonction de leur contenu.
- La classification consiste à former des groupes de documents, ou à attribuer des documents à des groupes existants.

Dans tous les cas, pour pouvoir atteindre un de ces objectifs, l'utilisateur doit donner des informations. Dans le cas de la recherche de documents, il peut par exemple donner un ou plusieurs mots clés et le système doit renvoyer les documents susceptibles d'intéresser l'utilisateur, comme le fait un moteur de recherche classique. Pour les documents contenant essentiellement du texte, les techniques les plus classiques sont celles reposant sur les n-grammes [TSYX00] et tf-idf [LBH+09]. Une autre possibilité est que l'utilisateur donne un exemple du genre d'image qu'il souhaite retrouver. Dans ce cas on utilise des techniques de recherche par le contenu CBIR (Content Based Image Retrieval). La figure 1.17 illustre ce principe. De nombreux système de CBIR existent pour les images naturelles [VC09], [MGW07], [DLW05] mais très peu pour les images de document [AAF+07]. Enfin, il est également possible de laisser l'utilisateur définir un ensemble de règles afin de retrouver les images respectant ces règles [Coü06], [RBD06].

Figure 1.17 – Recherche d'images similaires avec Google Image.

La plupart des techniques utilisées pour la recherche d'information par les entreprises de dématérialisation restent généralement basées sur le texte et donc sur le résultat des OCR. L'objectif des travaux de cette thèse et de privilégier les techniques basées sur l'image afin de fournir des outils pouvant être utilisés en plus des outils basés sur le texte ou à leurs places lorsque ceux-ci sont mis en défaut. Dans la section suivante nous présenterons des exemples de cas dans lesquels l'OCR peut rencontrer des difficultés. Les techniques

proposées dans cette thèse pourront alors être appliquées sur de telles images car nos techniques sont indépendantes du résultat de l'OCR.

1.5 Limites de l'OCR

L'OCR est la solution la plus couramment utilisée en entreprise pour analyser le contenu des documents. En effet, cette technique est l'une des premières a avoir été conçue spécialement pour les images de documents. De plus les outils d'OCR les plus aboutis tels que FineReader ou Tesseract, permettent de faire de nombreux prétraitements tels que le débruitage, le redressement, la binarisation et permettent de fournir le contenu textuel mais également une segmentation en blocs du document. Les performances de ces outils sur des documents créés numériquement et contenant du texte sont généralement plutôt bonnes. Par exemple, les auteurs de [YM11] ont montré que le taux de reconnaissance de FineReader 8.0 sur une base de 160 livres dont 100 en anglais, 20 en français, 20 en allemand et 20 en espagnol est en moyenne de 96,7 %.

Malheureusement, l'OCR ne fonctionne de manière optimale que sous certaines conditions. Les tests réalisés par [YM11] sont bons essentiellement car la base est constituée de livres contenant exclusivement du texte. De plus, la qualité d'impression du texte est bonne (les romans ne sont pas imprimés avec des imprimantes personnelles). Dans le contexte de l'analyse de dessins techniques, les auteurs de [LNGT01] présentent un ensemble de 24 causes d'erreurs d'OCR réparties en 4 classes : défaut d'image (marques isolées, bruit, etc.) , typographie (italique, souligné, etc.), écriture manuscrite ou au pochoir (espacement, inclinaison, etc.) ou encore de la mise en page (lignes verticales, caractères qui se touchent, etc.). Pour plus de détails sur les performances des OCR, les articles [Bre08], [CS06] et [RJN96] peuvent être consultés.

Dans les sous-sections suivantes, un ensemble non exhaustif de paramètres pouvant influer sur la qualité du résultat de l'OCR est présenté. Les OCR Tesseract 3.0 et FineReader 10.0 ont été utilisés pour extraire le texte des images de documents.

La résolution de l'image

Comme on peut le voir sur la figure 1.18, si de bons résultats sont obtenus à 300 dpi, ils se dégradent à 200 dpi et sont généralement inutilisables à 100 dpi.

Résolution	300 dpi	200 dpi	100 dpi
Image	Dans une optique de documentaire et de la	Dans une optique de documentaire et de la	Dans une optique de documentaire et de la
Tesseract	Dans une optique de documentaire et de la	Dans une optique de tiottuttterttaire et " la	Vsnsunetrmtqw.g. ¢=<..,.....-. nan |.
FineReader	Dans une optique de documentaire et de la	Dans une optique de documentaire et de (a	Dans une opOQM* 4» documsMMNMAli

FIGURE 1.18 – Impact de la résolution sur l'OCR.

L'orientation de l'image

L'image du document n'est pas toujours droite, cela est principalement dû au processus de numérisation. Un écart de quelques degrés est toléré par la plupart des OCR (voir figure

1.19). Des prétraitements de redressement sont souvent utilisés, mais ils ne s'adaptent pas à tout type de document. Par contre, l'orientation du texte est très souvent déterminée à 180 degrés près.

Rotation	0°	5°	10°
Image	Dans une optique de documentaire et de la	Dans une optique de documentaire et de la	Dans une optique de documentaire et de la
Tesseract	Dans une optique de documentaire et de la	Dans une optique de documentaire et de la	Dans une optique de '"ruar7tl'i'J, et de la
FineReader	Dans une optique de documentaire et de la	Dans une optique de documentaire et de la	Dans une optique de documentaire et de ia

Figure 1.19 – Impact de la rotation sur l'OCR, au-delà de 5 degrés, l'OCR est très perturbé. Une étape de redressement des documents est alors obligatoire préalablement si l'on souhaite appliquer l'OCR.

Images couleur et luminosité

Ces deux problèmes sont liés à l'étape de binarisation des OCR. En effet, la plupart des OCRs travaillent sur des images binaires avec le texte noir et le fond blanc. Sur des images de magazines, il est fréquent d'avoir de nombreux changements de couleur dans le texte et/ou dans le fond, ce qui rend la binarisation et donc la reconnaissance du texte plus complexe. Sur la figure 1.20 on observe que la qualité de la binarisation d'une image couleur impacte directement la qualité de la reconnaissance de l'OCR.

Image originale

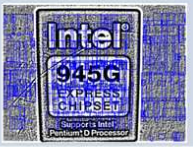
Binarisation	Niblack	Sauvola	Kasar
Image			
Tesseract	_"i'ES" , iamiiiir%' E B8grf,; '.'~ w vi? r iWi'd 'il 1ittlfrat! ':iWEi) vt,itlt(" ttIli)),?.],,??,:',,, _:'i'"iiifji, I P,' y ' "d4ra1Ti7'sgic"(g ae w (,'ii,ivrii' 'ii,Cili'_,if,,',l.f,"ivt),1 s	i o,ui, 'r.."':',:):'.,",',',"",',"".')'.:"'"	0 I nte l 94 5 G E X P R E S S C H I P S E T Supports Intel Pennum D Processor
FineReader	iW 945G « almdanhwi'. m •*£**& »-Rt (exrressj chiçsetl •>arvmi' «sy»»* » Supports Intel! Pentium* D Processof H « Wi	945G EXPRESS CHIPSET	Intel* 945G EXPRESS CHIPSET Supports Intel Pentium DProcessor

Figure 1.20 – Impact de la binarisation sur l'OCR, images issues de [KKR07]. Les performances de l'OCR sont optimales lorsque les caractères sont noirs, de taille suffisamment grande et sans bruit.

Police d'écriture

Comme on peut l'observer sur la figure 1.21, des caractères trop gros ou trop petits ne seront pas reconnus par l'OCR. L'utilisation de polices peu courantes peut également rendre la tâche de l'OCR plus complexe.

Image	
Tesseract	ocd,,,ghijurnnopqrst UVW xy it
FineReader	abcde fghijklmnopqrstUV wxyZ

FIGURE 1.21 – Impact de la taille de la police d'écriture sur l'OCR. Une police d'écriture trop grande ou trop petite entrainera parfois des erreurs d'OCR.

Trame, fax et journaux

La plupart des OCR ne reconnaissent plus les caractères lorsqu'ils sont composés d'une trame de points. Un exemple de l'effet produit par une trame est présenté sur la figure 1.22.

FIGURE 1.22 – Exemple de trame. Les documents imprimés en trame sont complexes pour l'OCR, car les caractères ne sont plus qu'un ensemble de points. Les points des figures vont également perturber le résultat de l'OCR.

Présence de textes manuscrits, graphiques, photos ou schémas

La présence de manuscrit sur un document peut perturber grandement les performances de l'OCR, car ce dernier essaie de retranscrire des symboles alors qu'il n'est pas capable de le faire (voir figure 1.23). L'OCR peut, par exemple, localiser des caractères à des endroits où il n'y en a pas.

Langue

Certaines langues sont plus complexes à analyser que d'autres en raison de la complexité de la forme des caractères. De plus, l'OCR s'auto-corrige en utilisant des dictionnaires, les résultats sur des noms propres peuvent être très variables. Il peut également y avoir des documents multi-langues, qui rendent complexe l'utilisation de dictionnaires La plupart des OCR nécessitent d'ailleurs que soit donnée en entrée la langue du document.

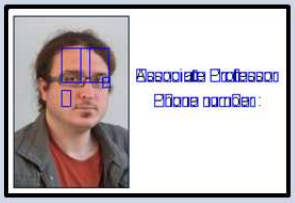
	Photo	Manuscrit
Image		
Tesseract	ka A Associate Professor I Phone number :	"aio-r, / Re, ' MIL. 1joic, Q,;,.,.,p1.v_ qvova
FineReader	Associate Professor Phone number :	2>&o^>i*r ^ \JiDKi' Cc>rr> n^- çpn K2/7&C A Rifi>

Figure 1.23 – L'OCR tente de lire du texte dans la photo puis dans le texte manuscrit alors que dans un cas il n'y a pas de texte et dans l'autre cas il n'est pas censé pouvoir reconnaitre l'écriture manuscrite.

Multi-documents

Une image peut contenir plusieurs documents. Ces documents peuvent avoir des orientations différentes, des langues différentes. Il est nécessaire de préalablement segmenter les sous-images de documents avant de pouvoir exécuter l'OCR. La figure 3.20 présente des exemples d'image contenant plusieurs documents.

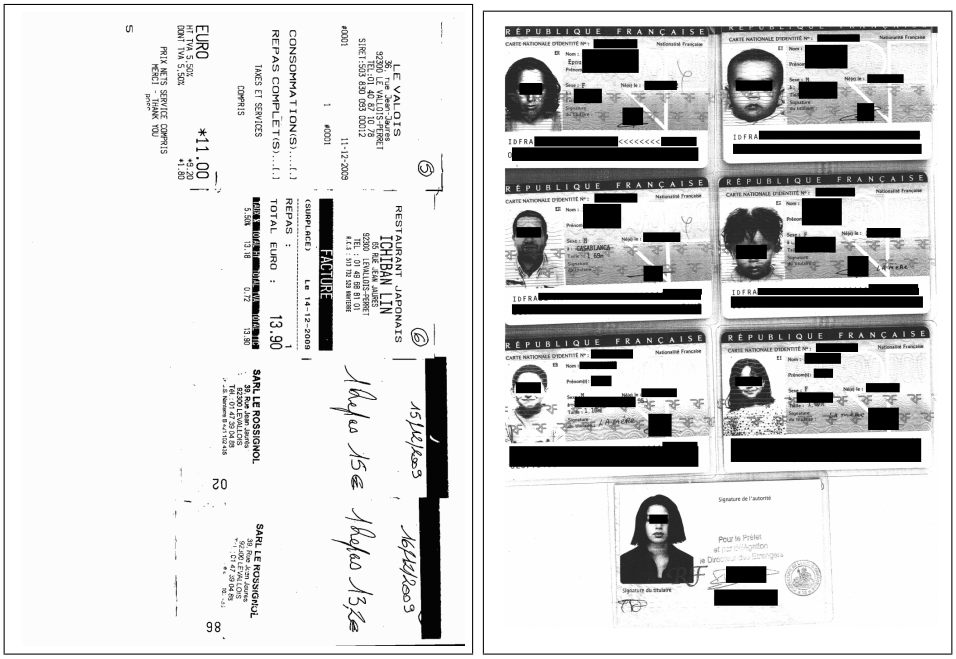


Figure 1.24 – Un document peut être composé de plusieurs pages et donc de plusieurs images. Un document peut également être constitué d'une unique page et donc d'une unique image. Comme le montre cette figure, une image peut également contenir plusieurs documents.

1.6 Verrous scientifiques

La complexité de l'analyse des images de documents vient du fait que, une fois numérisé, le document n'est qu'une image, c'est-à-dire un ensemble de pixels. Or l'utilisateur souhaite pouvoir faire des recherches sur des informations sémantiques de plus hauts-niveaux qui sont contenues dans le document comme le texte, la mise en page, la nature du document, etc. On parle alors de "semantic gap". Ce sont les techniques d'analyse et de traitement d'image qui peuvent permettre d'extraire ces informations.

Des chercheurs se sont récemment intéressés à différentes problématiques que nous avons citées précédemment. On peut notamment citer des travaux sur l'étude de la transparence dans les documents anciens [RJD+11] , sur la binarisation indépendante de la couleur de fond et d'écriture [KKR07] ou encore sur la détection rapide et robuste de l'angle d'inclinaison [NJ11].

Depuis une petite dizaine d'années, plusieurs méthodes génériques permettant de faire de la reconnaissance d'image de documents ont vu le jour. On peut citer par exemple DMOS [Coü06] où la description de la structure du document se fait à l'aide du formalisme haut-niveau baptisé EPF (Enhanced Position Formalism). On peut également citer Agora [RLDB07] qui est un outil d'analyse de mise en page interactif spécialisé dans la rétro-conversion des documents anciens.

Il existe également des entreprises qui proposent des solutions complètes comme A2IA pour la lecture automatique de chèques [GAA+01]. Pour l'OCR, les logiciels FineReader de ABBYY et Omnipage de Nuance Communications sont très utilisés en entreprise. Mais il existe également des outils open source tels que l'OCR Tesseract, développé par Google.

Bien que des solutions scientifiques soient proposées pour certaines problématiques, de nombreux verrous sont encore clairement identifiés par la communauté. La classification et la reconnaissance d'images de documents font parties de ceux là. C'est pourquoi, l'objectif de cette thèse est de mettre en avant des techniques d'analyse, de traitement, de reconnaissance permettant d'aboutir à une classification d'images de documents qui soit générique afin d'être utilisable dans différents contextes industriels. Afin que nos techniques soient utilisables lorsque l'OCR est mis en défaut, nous nous sommes focalisés pour notre étude sur des techniques n'utilisant pas le résultat du texte issu de l'OCR. Ainsi, nos méthodes pourront également être utilisées en parallèle de celles utilisant le texte afin de combiner les avantages des deux techniques.

La problématique de recherche et classification d'images dans une base de documents peut être décomposée en différentes sous-problématiques chacune dépendante d'un contexte particulier. Si l'ensemble des propositions faites dans ces travaux de recherche incluent systématiquement un utilisateur, nous modulons nos propositions sur le constat réalisé en production qu'un utilisateur, dont la tâche est de réaliser un classement d'une base d'images, ne possède pas systématiquement l'ensemble des connaissances nécessaires à la réalisation de cette tâche.

L'un des aspects originaux de nos travaux est donc de ne pas proposer des méthodes de classification exclusivement `ad hoc` à une catégorie d'images mais également d'intégrer la quantité de connaissance qu'à un utilisateur de la base d'images.

La figure 1.25 illustre la position des chapitres de la thèse en fonction des informations possédées par l'utilisateur. Nous allons présenter trois problématiques différentes de classification où la quantité d'information connue par l'utilisateur constituera l'axe directeur de chaque chapitre.

Tout d'abord, nous présenterons une méthode permettant d'aider un utilisateur à labelliser rapidement une base de documents sans qu'il n'ait besoin d'avoir en sa possession

une quelconque information préalable sur cette base. L'objectif est de créer une méthode permettant d'explorer une base d'images de documents et de regrouper les images en fonction de leur visuel. Si le résultat est satisfaisant pour l'utilisateur il pourra s'en servir pour labelliser rapidement un grand nombre de documents. Puisque l'utilisateur ne donne aucune information, nous privilégierons dans ce chapitre l'utilisation de techniques de classification non supervisées. Sur la figure 1.25, ce chapitre se positionne au minimum de l'axe représentant la connaissance utilisateur.

Ensuite, nous avons considéré le cas ou l'utilisateur veut retrouver une classe de documents particuliers dans une base de données contenant des documents divers. Afin de développer une méthode qui soit précise et robuste, nous nous sommes intéressés uniquement à des documents semi-structurés. Les documents d'une même classe doivent ainsi avoir une partie de leur image qui soit fixe. Cette contrainte ne restreint cependant que peu le champ d'action de notre algorithme car la plupart des documents industriels sont semi-structurés. On peux citer comme exemples de documents semi-structurés : des factures, des fiches administratives, des documents d'identité, des tickets de transport, etc. Sur ces documents, certaines informations changent (position et contenu) d'un document à l'autre mais une autre partie des informations reste fixe. C'est en comparant ces informations qui varient peu d'un document à un autre qu'il nous est possible de retrouver des documents tels que des cartes d'identités, des passeports, des tickets de métro, des factures de taxi dans une base de taille conséquente. Sur la figure 1.25, ce chapitre se positionne au milieu de l'axe représentant la connaissance de l'utilisateur.

Enfin, dans le dernier chapitre nous présenterons des techniques qui se veulent plus génériques. La technique du chapitre précédent permet d'avoir d'excellents résultats sur des documents semi-structurés. Cependant elle présente plusieurs inconvénients. Les documents appartenant à des classes différentes mais présentant des similarités visuelles fortes peuvent être confondus. De même, des documents d'une même classe peuvent présenter certaines différences visuelles notables rendant ainsi complexe l'étape de classification. L'objectif de ce chapitre est de mettre en place un ensemble de méthodes permettant de réaliser une classification générique en s'appuyant sur une quantité d'information donnée par l'utilisateur plus importante que dans les deux chapitres précédents. Nous étudierons les avantages et inconvénients qu'il y a à utiliser une technique supervisée basée sur l'image par rapport à une technique basée sur l'utilisation du texte. Ainsi, nous montrerons que les techniques basées sur l'image offrent des performances relativement proches de celles basées sur le texte. Nous proposerons également une adaptation des sacs de mots visuels permettant d'améliorer nettement la précision des résultats. Sur la figure 1.25, ce chapitre se positionne au maximum de l'axe représentant la connaissance de l'utilisateur.

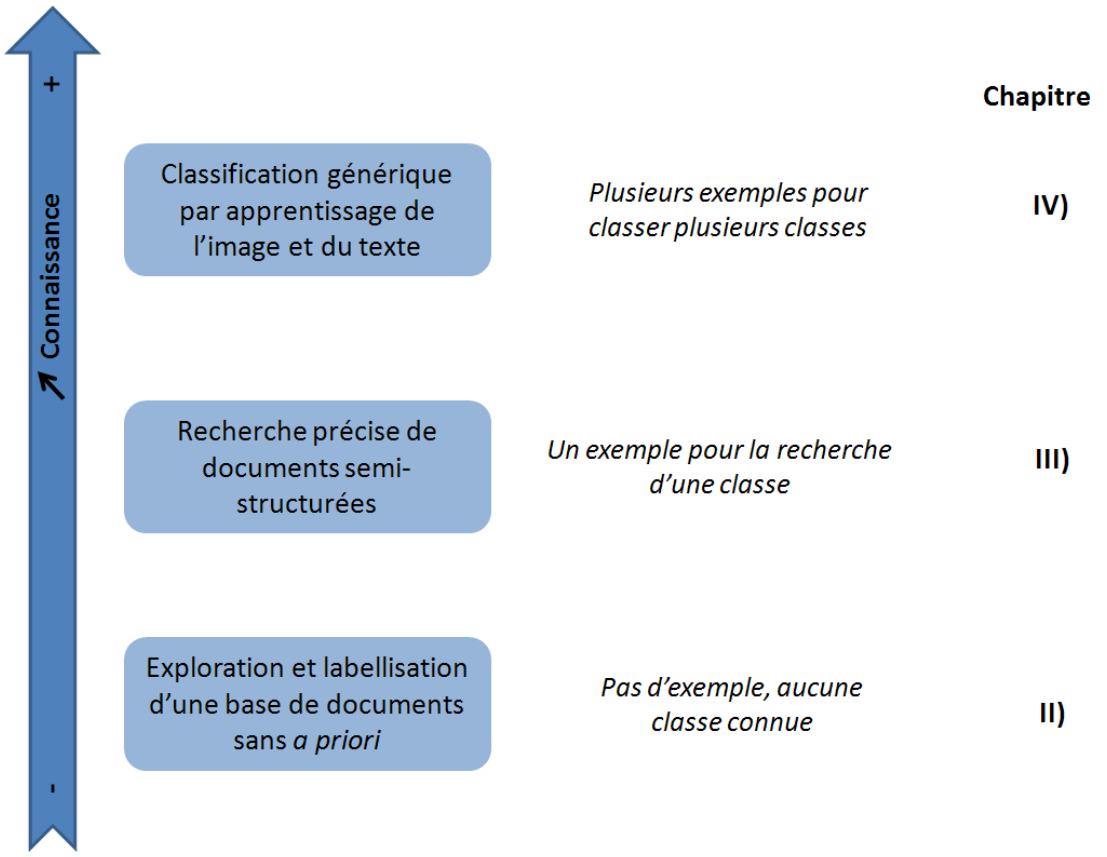


FIGURE 1.25 – Organisation des chapitres de la thèse autour des connaissances utilisateur.

Chapitre 2

Exploration de bases de documents sans *a priori* sur leurs contenus

Dans cette section, nous abordons une problématique particulière qui est la classification d'une base de documents sans connaissance *a priori* sur le contenu. Le contenu des images, le nombre de classes et la répartition des images dans les classes sont inconnus. De plus aucune image n'est pré-labellisée. Cette opération se rapproche d'une tâche d'exploration. En effet, nous considérons ici que l'utilisateur n'a pas de connaissance préalable sur le contenu des documents qu'il doit classer. Il arrive effectivement que le client de la prestation lui-même ne connaisse le contenu des documents qu'il souhaite numériser, notamment dans le cas de dématérialisation de vieux stock de documents pour leur archivage. Une telle tâche est actuellement généralement effectuée entièrement manuellement et semble difficilement automatisable. Nous proposons dans ce chapitre une méthode permettant d'aider un utilisateur à explorer une base afin d'en classer rapidement son contenu.

Notre ambition n'est pas de proposer une méthode automatique de classement d'images de documents. En effet, si l'inconvénient qu'il y a à réaliser un classement manuel est essentiellement le temps que cela prend, ce tri manuel possède néanmoins l'avantage que la décision est prise par un opérateur humain. Nous souhaitons donc, dans le cas de figure où l'on ne possède pas de connaissance sur la base à trier, proposer une méthode où l'opérateur humain reste au cœur du système. Nous nous intéressons donc à la définition d'une méthodologie permettant de simplifier et d'accélérer le processus de classification manuel de documents. Comme souligné récemment dans [Sau11], cette problématique représente un enjeu économique important pour les sociétés de numérisation.

Cette problématique s'inscrit dans le cadre de l'analyse et la classification d'images de documents. Dans un premier temps, ce chapitre s'attardera sur la description d'un ensemble de solutions proposées dans l'état de l'art de l'analyse et la classification d'images de documents. Nous détaillerons ensuite notre proposition d'aide à la classification d'images de documents. Nous montrerons, entre autre, que notre proposition permet de diviser le temps de labellisation manuel par un facteur 3 par rapport à un processus de tri manuel séquentiel et exhaustif d'une base.

2.1 État de l'art sur les techniques de classification d'images de documents

Il existe de nombreuses références détaillant comment classer un ensemble d'images de documents. Il est possible de les regrouper en fonction de la technique d'apprentissage utilisée ou encore selon la nature des caractéristiques utilisées pour cette tâche de classification.

Les techniques d'apprentissage se distinguent en trois principales catégories : l'apprentissage supervisé, l'apprentissage semi-supervisé et l'apprentissage non supervisé. La différence porte principalement sur les informations connues d'un utilisateur sur les éléments à classer. Si on dispose de nombreux éléments déjà labellisés, on parle d'apprentissage supervisé. Si on dispose de quelques éléments labellisés, on parle d'apprentissage semi-supervisé. Enfin si on ne dispose d'aucun élément labellisé on parle d'apprentissage non supervisé. La lecture de l'état de l'art de Chen et Blostein [CB07] des techniques de classification d'images de documents montre que les techniques les plus répandues s'appuient sur une classification supervisée. L'inconvénient des techniques supervisées est qu'il est nécessaire de connaître à la fois le nombre de classes d'images composant la base mais aussi de disposer d'une vérité terrain représentative de la distribution de cette base. Il faut également que la base d'apprentissage soit de taille suffisamment importante afin de permettre un apprentissage performant. Ces deux conditions sont d'autant plus difficiles à satisfaire que le volume d'images à indexer est important. Il faut donc que de nombreux documents soient préalablement classés.

Nous allons présenter différentes solutions existantes pour la classification d'images de documents en les regroupant suivant la nature des descripteurs utilisés pour faire cette classification. Tout comme les auteurs de [CB07], nous distinguons trois principales familles de descripteurs :

- le texte (généralement extrait avec un logiciel d'OCR.)
- l'image (couleurs, textures, ...)
- la structure (description de la mise en page des documents)

Dans les sous-sections suivantes, nous présentons des exemples d'utilisation de chacune de ces caractéristiques.

2.1.1 Caractéristiques textuelles

De nombreux travaux se basent sur l'utilisation du texte. Ceci s'explique principalement par les progrès réalisés ces dernières années par les logiciels d'OCR. Parmi ces travaux on peux notamment citer ceux de [Seb02] , [LBH+09] ou encore [HL09]. Ces travaux sont principalement basés sur l'utilisation des fréquences d'apparition de mots ou de groupes de lettres tels que "tf-idf" ou les "n-grammes". Les auteurs de [LBH+09] proposent une technique de classification basée sur les n-grammes. Les n-grammes représentent des séquences de n-éléments. Les auteurs proposent de calculer des n-grammes de mots et des n-grammes de caractères. Par exemple, la chaîne de caractères "structured document image matching" comporte deux 3-grammes : "structured document image" et "document image matching". La chaîne de caractère "image" contient trois 3-grammes de caractères : "ima", "mag" et "age". Les documents sont caractérisés puis classés grâce aux fréquences d'apparitions de ces n-grammes. Leur base de test est constituée de 2000 documents divisés en 24 catégories. De nombreux tests sont effectués, les meilleurs résultats sont obtenus avec l'utilisation de n-grammes à 3 caractères et l'utilisation des SVM. Une précision moyenne de 96,8% est obtenue. Selon les auteurs, les performances de leurs techniques se détériorent

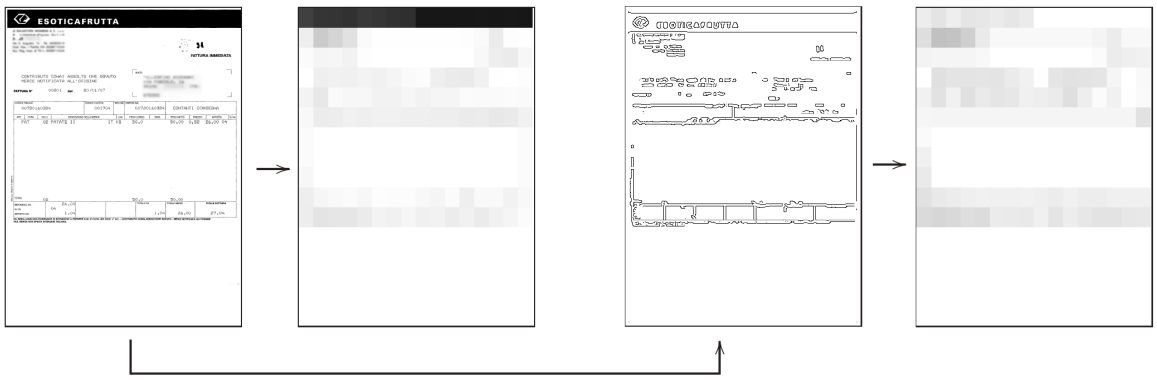


Figure 2.1 – Les images sont découpées en 16 lignes et 16 colonnes. La densité de pixels est calculée sur l'image originale et sur les contours de cette image. L'illustration est extraite de [BDMS10].

lorsque le nombre de mots par document diminue. De nombreux autres articles utilisent les n-grammes pour la classification d'images de documents : [TWL02], [BA04], [SH07], [PN07].

2.1.2 Descripteurs images

Pour ce qui est de la segmentation, l'analyse ou la classification d'images naturelles, il est courant d'utiliser des descripteurs tels que la couleur, la texture et la forme [DLW05]. Les auteurs de [KSK11] proposent d'utiliser des moments de couleurs, des transformées en ondelettes pour caractériser respectivement la couleur et la texture des images. Pour la caractérisation de formes, les auteurs indiquent que les moments invariants et les descripteurs de Fourrier sont les outils les plus couramment utilisés. Cependant, pour les images de documents, les caractéristiques extraites sont différentes.

Les auteurs de [BDMS10] présentent une technique de classification d'images de documents basée sur 2 caractéristiques principales : la densité des pixels noirs des images et la densité des pixels noirs du contour des images. La détection des contours est appliquée après avoir réduit la résolution de l'image originale. Pour calculer ces densités, les images sont découpées en grilles de n lignes et n colonnes. Chaque image est alors décrite par un vecteur de $2 * n * n$ dimensions (qui est la concaténation des $n * n$ valeurs de densité de l'image originale et des $n * n$ valeurs de densité des contours). La figure 2.1 présente un exemple des caractéristiques extraites. Les auteurs appliquent une ACP (Analyse en Composantes Principales) afin de simplifier l'espace de description. L'ensemble des dimensions sélectionnées sont celles permettant d'obtenir une variance supérieur à 95%. Ensuite la classification est effectuée à l'aide des SVM. En fixant $n = 16$ et en utilisant une image d'apprentissage par classe, les auteurs obtiennent un taux de classification correct de 97,17 %.

Les auteurs de [HDRT98] présentent plusieurs techniques permettant la classification de formulaires. L'une de ces technique est basée sur l'utilisation de caractéristiques visuelles. Les auteurs découpent les images à caractériser en différents niveaux n. Chaque niveau n correspond à une taille de grille $n * n$. Pour chaque niveau est calculée la densité de pixels noirs pour chaque case obtenue par le découpage de la grille. La figure 2.2

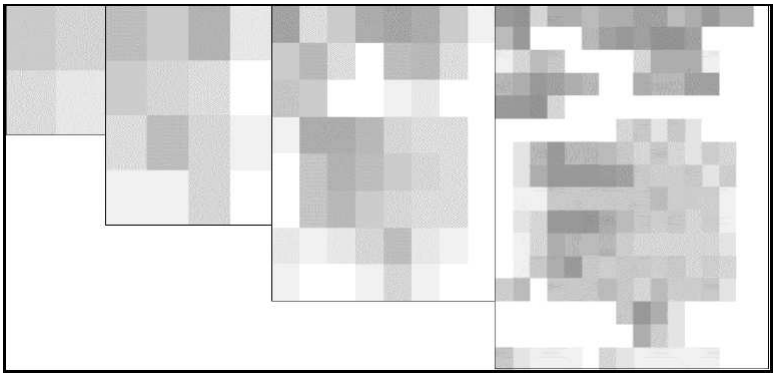


FIGURE 2.2 – Décomposition pyramidale de l'image en 4 niveaux différents [HDRT98].

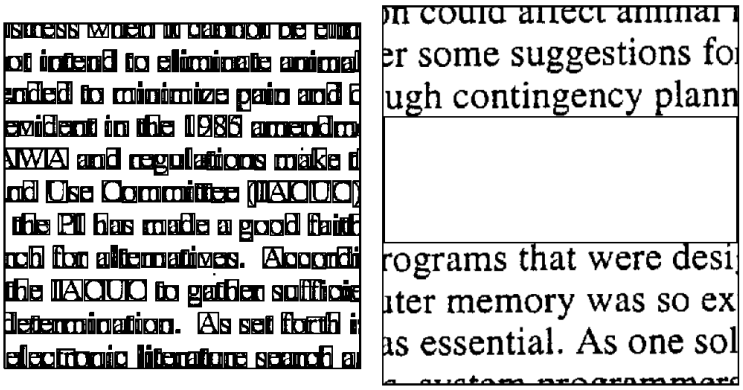


FIGURE 2.3 – Exemples de caractéristiques. Sur l'image de gauche sont extraites les composantes connexes afin de calculer des statistiques sur leur hauteur, largeur, périmètre, etc. Sur l'image de droite est caractérisé l'espace blanc entre les paragraphes. Les images sont extraites de [SDR01].

illustre cette décomposition pyramidale de l'image. Les méthodes de classification MLP (Multi Layer Perceptron) et k-PPV (k Plus Proches Voisins) sont utilisés. des test ont été effectués sur une base de 570 formulaires comprenant 27 classes. La moitié de la base a été utilisée pour l'apprentissage. En combinant les 4 premiers niveaux, un taux de reconnaissance de 100% est obtenu avec les k-PPV (en fixant k=1) et de 99,65% en utilisant les MLP (utilisant une couche cachée à 27 neurones).

Pour classer des images de documents, Shin *et al.* [SDR01] proposent une approche basée sur les arbres de décision. Les auteurs calculent un grand nombre de descripteurs tels que des statistiques sur la taille, la densité, le périmètre des composantes connexes, le nombre de lignes horizontales, verticales, la présence d'espaces blancs, etc. Des exemples de caractéristiques sont présentés sur la figure 2.3. Un arbre de décision est construit à partir de la vérité terrain extraite d'un échantillon représentatif de la base d'images. Cet arbre est ensuite utilisé pour classer le reste de la base. Sur une base de formulaires de 5590 images composées de 20 classes, en utilisant une vérité terrain de 2000 images, les auteurs obtiennent une précision de 99.7% sur la classification de 2000 autres images de la base.

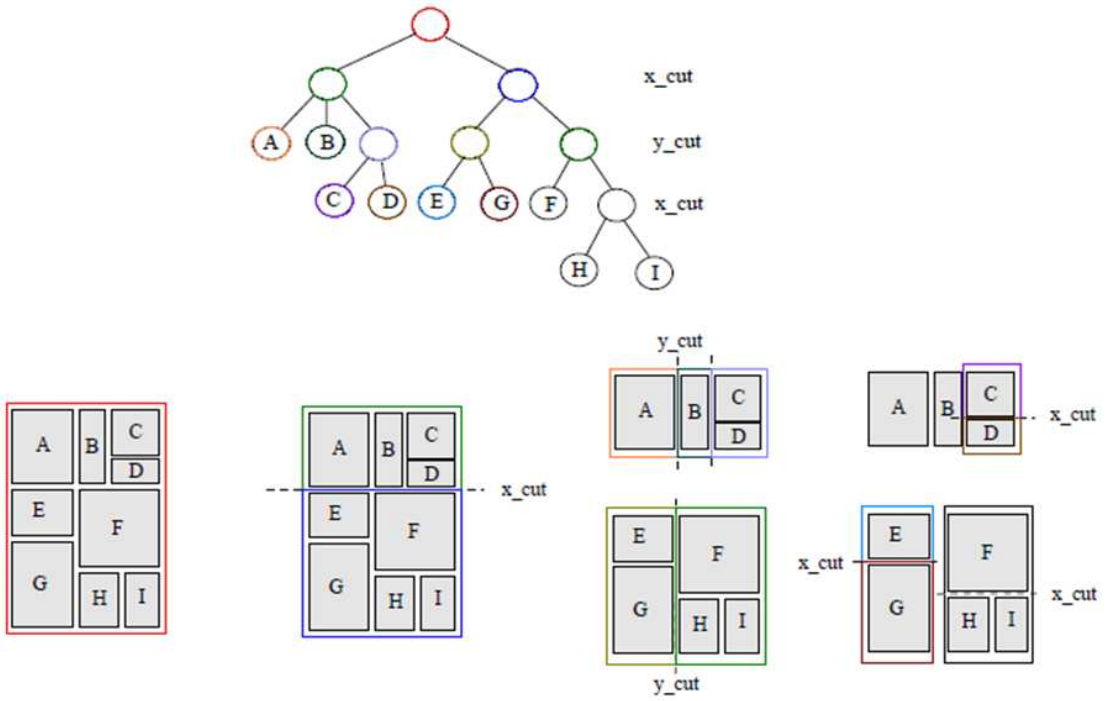


FIGURE 2.4 – Une image de document segmenté récursivement en x et en y et sont arbre MXY correspondant. Schéma inspiré de [CLMS01] et [CGMS99].

2.1.3 Caractéristiques structurelles

Pour caractériser la structure d'un document, il faut caractériser sa mise en page. La principale problématique est alors de segmenter le document en différentes parties (également appelées blocs) puis de représenter cette structure, par exemple sous forme de graphe. Comparer des images de documents reviendra alors à comparer leurs mises en pages. Si l'on souhaite effectuer une tache de classification, ceci induit que des documents avec des mises en page similaires appartiendrons à la même classe. Les auteurs de [CCMM98] présentent un état de l'art et une discussion sur les techniques d'analyse de la mise en page d'images de documents.

Dans [CLMS01], la structure est extraite sous forme d'arbre MXY qui segmente récursivement le document horizontalement et verticalement. La segmentation horizontale et verticale se fait soit le long d'espaces blancs (entre des paragraphes, par exemple) soit le long de lignes séparatrices. Les documents sont alors représentés par des graphes. Les nœuds internes peuvent prendre 4 valeurs : hs (horizontal space), vs (vertical space), hl (horizontal line) et vl (vertical line). La valeur d'un nœud représente la manière dont ses enfants sont segmentés. Les feuilles peuvent prendre 4 valeurs : HL (horizontal line), VL (vertical line), T (texte) et I (image). Ces valeurs représentent le contenu des blocs. La figure 2.4 illustre un exemple d'arbre MXY. La structure de chaque document est ensuite encodée dans un vecteur de caractéristiques. Les auteurs proposent de classer les images de documents en utilisant des réseaux de neurones. La base de test est composée de 600 formulaires appartenant à 5 classes. La vérité terrain est composée de 305 images. Une précision de 92% est obtenue.

Les auteurs de [BW03] utilisent des graphes gaussiens du premier ordre pour classer des documents relativement à leur structure (qui est préalablement extraite par l'OCR Scansoft TextBridge). La base est composée de 857 documents répartis en 5 classes. Un taux

de reconnaissance d'environ 99% est obtenu en utilisant 50 images pour l'apprentissage.

Les travaux de [HBBC08a] reposent sur une variante des réseaux de neurones appelée "Incremental Growing Neural Gas". Les auteurs extraient la structure physique et logique des documents en se basant uniquement sur les mots extraits par l'OCR. Les graphes sont ensuite mis en correspondance par "graph probing". Leurs travaux sont testés sur une base de documents industriels composés de 324 documents utilisés pour l'apprentissage et 169 éléments pour tester les performances de leur méthode. Les documents sont répartis en 8 classes. Des résultats sont exposés en fonction de la variation de plusieurs paramètres internes au système, le meilleur résultat obtenu donne un taux de reconnaissance de 99.40%.

Gordo et Valveny [GV09] proposent de faire une comparaison de mise en page des documents à l'aide d'un descripteur invariant en rotation. Afin de calculer des performances qui soient indépendantes de celles de la segmentation de la structure du document, les auteurs utilisent une segmentation faite manuellement reflétant la vérité terrain. Leur base est composée de 823 documents répartis en 8 classes. Les auteurs appliquent une validation croisée de type "leave-one-out cross validation", c'est à dire que 822 documents sont utilisées pour l'apprentissage puis utilisés pour déterminer l'image restante. Ceci est fait pour chaque image. Une précision moyenne de l'ordre de $65,9\%$ est obtenue.

Dans la section suivante sont détaillées différentes techniques permettant d'extraire la structure des documents.

2.2 Analyse d'images de documents

Pour pouvoir étudier le contenu d'une image de document, la première étape consiste à extraire des informations de cette image. Ces informations sont appelées caractéristiques ou encore descripteurs, car elles permettent de caractériser l'image et de décrire son contenu. Il est courant, avant d'analyser les caractéristiques, de prétraiter les images des documents afin que les descripteurs ne soient pas perturbés par du bruit ou une mauvaise orientation de l'image. Nous ne présenterons pas les techniques de binarisation car les documents que nous souhaitons traiter sont binarisés automatiquement à la numérisation.

2.2.1 Pré-traitement

Cette étape est obligatoire lorsque les outils utilisés pour extraire les descripteurs ne sont pas robustes au bruit ou à la rotation. Elle ne donne pas une information supplémentaire sur le contenu de l'image, mais est faite pour aider la tâche d'extraction de caractéristiques.

Traitement du bruit

Le bruit est très souvent présent sur les images de documents, car il peut apparaitre à différents endroits de la chaîne de numérisation : à l'impression, pendant la vie du document et à la numérisation.

Des traitements classiques tels que les filtres médians [ZL11] ou l'algorithme kFIll [AKTS08] sont utilisés pour filtrer des pixels isolés sur l'image (bruit impulsionnel, également appelé bruit "sel et poivre"). Les opérateurs de morphologie mathématique tels que la dilatation ou l'érosion et leurs compositions (ouverture et fermeture) sont également très souvent utilisés [OK95], [ZL11] pour corriger de légers défauts sur les images comme "recoller" des morceaux de caractères (voir figure 2.5).

Figure 2.5 – Fermeture morphologique appliquée à des caractères dégradés. Ceci permet de fusionner deux composantes connexes qui faisaient parties du même caractère originellement.

Redressement des pages

Un autre problème lié à la numérisation d'un document est qu'il peut se retrouver être légèrement tourné par rapport à l'axe horizontal. Cela peut venir de l'impression, car la feuille est rarement exactement perpendiculaire à l'axe d'impression. Lors de la numérisation également il est fréquent que la feuille ne soit pas exactement positionnée comme il faut. Si l'angle est inférieur à 2 degrés, cela a généralement peu d'impact sur les traitements suivants, au-delà il faut redresser l'image si l'analyse n'est pas robuste aux rotations.

Pour cela, il existe plusieurs techniques. On peut citer celles basées sur les profils de projection [LSS07], la détection de lignes (Hough) [AF00], l'algorithme RAST [Bre02a], les chaînes de plus proches voisins [LLT03], l'ACP (Analyse en Composantes Principales) [SIR99] ou plus récemment par recouvrement de parallélogrammes [NJ11].

2.2.2 Segmentation de la structure d'un document

L'extraction de la structure des documents consiste à identifier le contenu de différentes parties d'un document. Le but est d'arriver à localiser les zones d'information : textes typographiés ou manuscrits, les tableaux, les figures, les code barres, les photos, les signatures, les logos, les tampons... Il n'est pas question ici de lire ou d'interpréter le contenu de ces zones, mais " simplement " de détecter leur présence.

Comme le souligne Cattoni *et al.* dans l'état de l'art [CCMM98], une grande quantité de publications propose des solutions utilisables uniquement dans des contextes particuliers. Nous allons donc nous concentrer sur des techniques relativement génériques.

La plupart des techniques fournissent une segmentation par bloc (ou par zone) mais certaines d'entre elles permettent une segmentation au niveau pixel ou encore au niveau composante connexe.

RLSA

Le RLSA (Run Length Smoothing Algorithm) [WCW82] consiste à relier les pixels noirs d'un document entre eux si leurs distances est inférieure à un certain seuil. Suivant le seuil choisi, cela permet de segmenter une lettre, un mot, une ligne ou un paragraphe. La figure 2.6 illustre ce principe.

Cet algorithme n'est pas robuste à des rotations de grand angle, mais il marche tout de même très bien pour des rotations de quelques degrés. On peut également souligner que le seuil utilisé pour regrouper les pixels peut être déterminé par une pré-analyse du document (statistiques sur les tailles des composantes connexes). Il existe de nombreuses variantes du RLSA, notamment celle de Shi [SG04], basée sur un lissage directionnel flou

qui permet de localiser et segmenter le texte. Sun [Sun06] propose une version modifiée de RLSA permettant de segmenter l'image en blocs de formes quelconques.

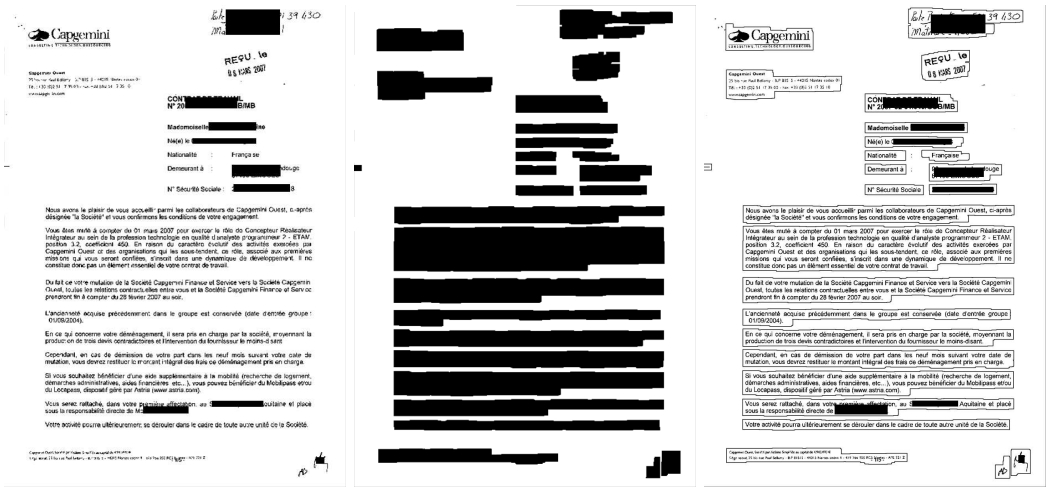


FIGURE 2.6 – Application de RLSA pour la segmentation physique de document. Dans notre cas, RLSA a été paramétré pour faire une segmentation en paragraphes.

Profil de projection

Cette méthode consiste à projeter horizontalement et verticalement les pixels d'une image. L'utilisation la plus connue est le "R-XY cut" [NS84] qui consiste à découper le document récursivement à partir des espaces blancs du profil et à créer un arbre X-Y représentant la structure du document. On obtient ainsi une représentation hiérarchique de la structure [DA02]. Cesarini [CLMS01] propose un arbre XY modifié tenant compte non seulement des découpages des espaces blancs, mais aussi le long des lignes. Han [HKYJ07] reprend l'idée de Cesarini mais utilise en plus un réseau de neurones pour rejeter le bruit.

Comme on peut le voir sur la figure 2.7, le principal inconvénient de cette méthode est qu'elle ne permet pas de segmenter des documents ayant une structure particulière comme des zones concaves, convexes ou inclues les unes dans les autres.

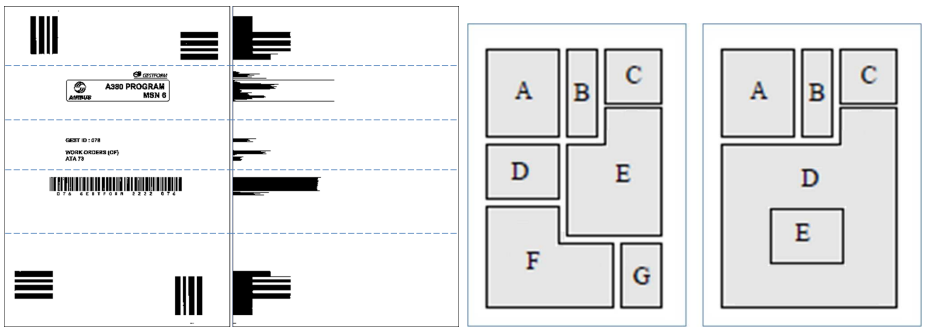


FIGURE 2.7 – Application du RXY cut pour la segmentation physique de document. De gauche à droite : une image de document, son profil de projection horizontal et deux exemples de structures ne pouvant pas être segmentées par profil de projection. Les schémas sont inspirés de [MMCS04].

Espaces blancs

Baird [Bai92] et Antonacopoulos [Ant98] proposent chacun une technique de segmentation basée sur l'analyse du fond du document. En effet, les espaces blancs dans un document sont des délimiteurs universels. Le but de leurs approches est de trouver respectivement les plus grands rectangles blancs et les espaces blancs entre les zones d'intérêts afin de mettre en évidence la structure du document.

Breuel [Bre02b] utilise une technique similaire, il recherche les plus grands espaces blancs rectangulaires et les évalue afin de décider s'ils sont des délimiteurs ou non.

Texture

Plusieurs approches basées sur les textures ont vu le jour. On peut noter principalement celles de Jain *et al.* [JB92] qui considèrent que les zones de texte ont des textures différentes de celles des zones de fond, de graphique, etc., et les identifie en partitionnant l'ensemble des pixels à l'aide de filtres de Gabor. L'intérêt de leur méthode est qu'elle est robuste aux rotations et détecte le texte typographié comme le texte manuscrit. Sur le même principe Jain et Zhong [JZ96] utilisent un réseau de neurones pour entrainer un ensemble discriminant de masques de textures. Les caractéristiques des textures sont obtenues en calculant le produit de convolution du masque obtenue par l'image d'entrée. Journet *et al.* [JME+08] proposent une technique basée sur l'analyse multirésolution pour la classification des pixels d'images de document. Ces techniques donnent une segmentation de chaque pixel composant l'image.

Voronoï

Kise[KSI98]a popularisé l'utilisation de l'algorithme de Voronoï pour segmenter les documents. Beusekom [vBKSB06] présente une extension de cette méthode. Une illustration de la segmentation par Voronoï est présentée sur la figure 2.8.

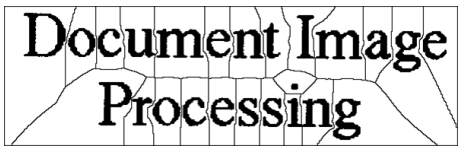


Figure 2.8 – Segmentation en caractères en utilisant Voronoï, d'après [KSI98]

Tab stop

Au lieu d'étudier les espaces blancs, Smith [Smi09] utilise les "tab-stop" c'est à dire les alignements des caractères en début et fin de lignes. En effet, si le texte est écrit en justifié, toutes les lignes d'un paragraphe (sauf la première et la dernière) sont alignées à la fois à droite et à gauche. Grâce à cette information, Smith détermine la position des colonnes de texte dans les images, comme l'illustre la figure 2.9.

Méthodes hybrides

Les méthodes présentées ayant chacune leurs avantages et leurs inconvénients, certains auteurs ont conçu de nouvelles méthodes à partir de plusieurs existantes. Wang et al. [WS89] comparent et utilisent RLSA et XY cut. Fisher et al. [FHD90] utilisent RLSA puis

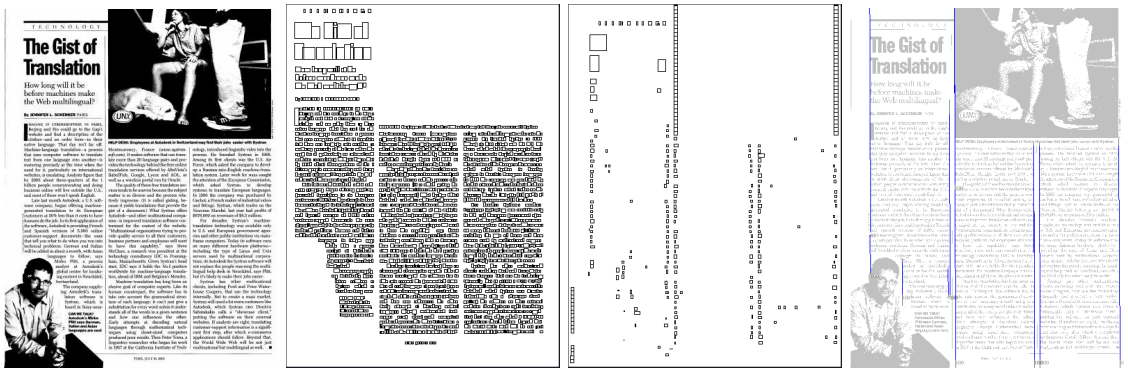


Figure 2.9 – Application du tab-stop pour segmenter les colonnes de texte, d'après [Smi09]. De gauche à droite : image originale, image avec composantes connexes filtrées par taille, image avec les composantes "tab-stop" et image de la détection des colonnes après filtrage par position.

des règles sur les CC. Benjlaiel [BKA06] propose une méthode utilisant RLSA, le profil de projection et effectue un traitement à part pour les tableaux à l'aide de la transformée de Radon. Agrawal *et al.* ont mis au point Voronoi++ qui est un mélange de Voronoi et Docstrum [AD09].

Extraction de la structure par OCR

Initialement, le but d'un OCR est de lire le texte présent dans une image. Mais pour arriver à lire le texte sur des documents dont la structure est complexe, la plupart des logiciels d'OCR procèdent préalablement à une analyse de la structure de document avant d'aller lire le texte. Sur la figure 2.10 figure le résultat de l'extraction de structure faite par FineReader 8. Le logiciel arrive à reconnaitre le texte typographié, les tableaux, les images et les code-barres, mais il est perturbé par la présence d'autres informations telles que le texte manuscrit, les signatures et les tampons. FineReader et Tesseract ont des performances équivalentes aux techniques de l'état de l'art d'après le concours de segmentation organisé pour la conférence ICDAR 2009 [APBP09].

2.2.3 Description des blocs segmentés

Une fois la structure de l'image extraite, l'objectif est de caractériser plus précisément le contenu de chacun des blocs. Cette caractérisation se déroule en deux étapes pendant lesquelles chacun des blocs sont étudiés un à un. D'abord, un certain nombre de descripteurs est extrait puis, ces valeurs sont analysées afin de déterminer la nature du bloc. Les caractéristiques issues de chaque bloc peuvent être liées à l'organisation des pixels dans le bloc, aux composantes connexes au sein d'un même bloc ou encore à l'analyse des formes générales. Un ensemble non exhaustif est présenté ici.

Les auteurs de [KSB07] proposent un ensemble de descripteurs simples permettant de classer le contenu de chaque bloc en huit catégories : les formules mathématiques, les logos, le texte, les tableaux, les dessins, les images, les lignes et le bruit. Pour cela un ensemble de 9 caractéristiques sont utilisées. Les trois premières on fait leurs preuves dans le domaine du CBIR (Content Based Image Retrieval) [DKN08] : l'histogramme des textures de Tamura, les histogrammes d'invariance relationnelle et des images redimensionnées en 32 x 32 pixels. La quatrième est utilisée pour faire la différence entre les dessins et le texte.

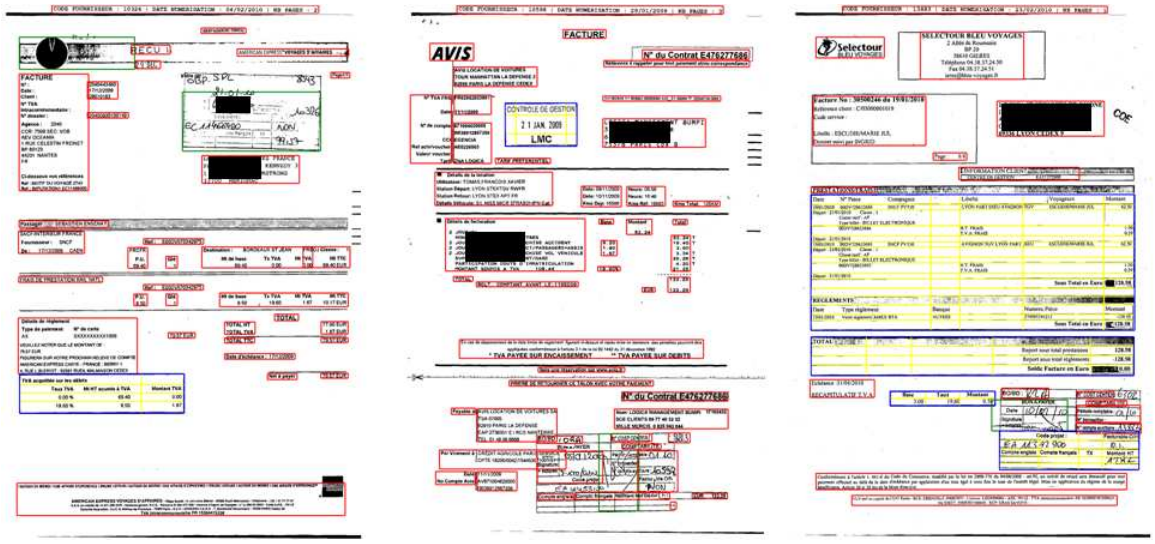


Figure 2.10 – Résultat de l'extraction de la structure des documents faite avec FineReader 8. Les zones de texte sont encadrées en rouge ; les tableaux en bleu, les ligne et colonnes des tableaux en jaune et les images en vert.

Baird *et al.* [BMAC07] caractérisent chacun des pixels de l'image en utilisant le canal de luminosité de l'image. Les auteurs ont sélectionné 26 caractéristiques parmi plus de 60. Parmi ces 26 caractéristiques, 4 correspondent à la luminosité moyenne des pixels voisins du pixel cible, avec un masque de taille N x N (pour N = 1,3,9,27). Seize autres caractéristiques sont calculées à partir des 4 masques. Les masques sont découpés horizontalement, verticalement et selon les deux diagonales, puis les différences entre chacune des moitiés sont ensuite calculées.

Bukhari *et al.* [BAA+10] proposent une méthode basée sur l'apprentissage de la forme des composantes connexes, mais également sur la forme des composantes voisines. Les auteurs se basent sur l'hypothèse que la taille des composantes connexes représentant le texte est généralement plus petite que les composantes représentant une information autre que le texte. Ils supposent également que les formes des composantes représentant du texte sont plus régulières que celles des composantes qui ne représentent pas du texte. Cette méthode donne une segmentation d'un point de vue des composantes connexes.

Dans l'article [SDR01] Shin et Doermann proposent une grande quantité de descripteurs permettant de caractériser visuellement le contenu d'une image de document. Ces descripteurs sont calculés sur des fenêtres. Une fenêtre correspond soit à une sous-partie quelconque de l'image (mais de taille suffisamment grande pour contenir une information haut-niveau telle que le nombre de mots présents dans la fenêtre) ; soit à une bande c'est-à-dire à un ensemble horizontal ou vertical de fenêtres ; soit à l'image entière. Sur un document binarisé, l'ensemble des composantes connexes (CC) de l'arrière-plan et de l'avant-plan sont extraites. Les boites englobantes (BE) de chacune des CC sont calculées. Un ensemble de règles basées sur la hauteur, la largeur des CC et sur la densité des BE permettent de différencier 4 types de CC différentes : le texte, les lignes horizontales, les lignes verticales et un ensemble de lignes horizontales et verticales (tel qu'un cadre ou un tableau). Enfin sont calculées 21 types de caractéristiques comme le rapport entre le nombre de pixels noirs et la taille de la fenêtre, le ratio entre le nombre de pixels noirs et l'aire des BE, le ratio entre l'aire de BE et l'aire de la fenêtre. Sont également calculées un ensemble de statistiques telles que la somme, la moyenne, le cardinal, le minimum, le

maximum, l'écart-type, la valeur médiane et le mode des propriétés des CC et des BE telles que leurs hauteur, largeur, aire, périmètre, circularité, etc.

2.3 Les méthodes de classification non supervisées

Pour pouvoir classer des données, il faut dans un premier temps extraire des caractéristiques qui sont ensuite utilisées par un algorithme de classification. Maintenant que nous avons présenté un ensemble de caractéristiques, nous allons étudier des techniques de classification. Dans le cadre de ce chapitre, nous ne disposons pas d'éléments labellisés pour classer les images d'une base, nous concentrerons donc notre étude sur les techniques de classification non supervisées.

Le but de la classification non supervisée est de créer des regroupements de données (des classes) en utilisant uniquement des informations propres aux données (ici se seront les caractéristiques calculées sur des images) et sans utiliser de données étiquetées, c'est-à-dire des données dont on connaît déjà la classe. Pour cela, on se base généralement sur la similarité des éléments en supposant que des éléments proches les uns des autres font partie d'une même classe. Certaines techniques sont orientées pour répondre à des problématiques précises telles que les données discrètes [GRS99], les grandes bases de données [EKSX96], [GRS98], [HK98], [ZRL96] et les donnés de grandes dimensions qui feront apparaitre les techniques basées sur les projections et les sous-espaces.

La tâche de classification peut se décomposer en 5 grandes sous-étapes [JMF99] :

1. la sélection et l'extraction de caractéristiques (représentation des données)
2. la définition d'une mesure de proximité/similarité (comparaison des données)
3. le regroupement de données à proprement dit (partitionnement ou "clustering")
4. la caractérisation des groupes (description des groupes)
5. la mesure de qualité (évaluation des groupes)

La sélection de paramètres s'adosse au problème d'identification de sous-ensemble et de sélection de dimensions. Lors de l'extraction des paramètres, il est possible de leur faire subir certaines transformations notamment pour normaliser les données. L'étape 4 consiste à trouver une description compacte des groupes (abstraction). Cela consiste notamment à chercher des notions représentatives d'un groupe comme le centroïde ou le diamètre par exemple. Notre étude portera sur l'étape 3.

Habituellement, deux principales catégories d'algorithmes de clustering sont distinguées ([EKSX96], [GRS98], [GRS99], [JMF99]) : les algorithmes de partitionnement (tel que les K-means) et les algorithmes hiérarchiques (par exemple la Classification Ascendante Hiérarchique). Il existe de nombreux autres algorithmes basés sur la théorie des graphes, sur la densité, les grilles, les réseaux de neurones, etc. Les techniques basées sur la densité ont connu un réel succès depuis DBSCAN [EKSX96].

L'article de Jain [JMF99] et la thèse de Candillier [Can06] concernant la classification non supervisée peuvent être consultés pour avoir un état de l'art plus détaillé. La figure 2.11 présente une synthèse non exhaustive présentée dans ces articles. Nous allons par la suite présenter uniquement les méthodes hiérarchiques et les méthodes de partitionnement.

Méthodes hiérarchiques

Les méthodes hiérarchiques permettent de classer les objets de proche en proche et donc sans connaitre le nombre de classes à l'avance. Il existe deux types de méthodes hiérarchiques : la classification ascendante (CAH) qui consiste en un ensemble de fusions

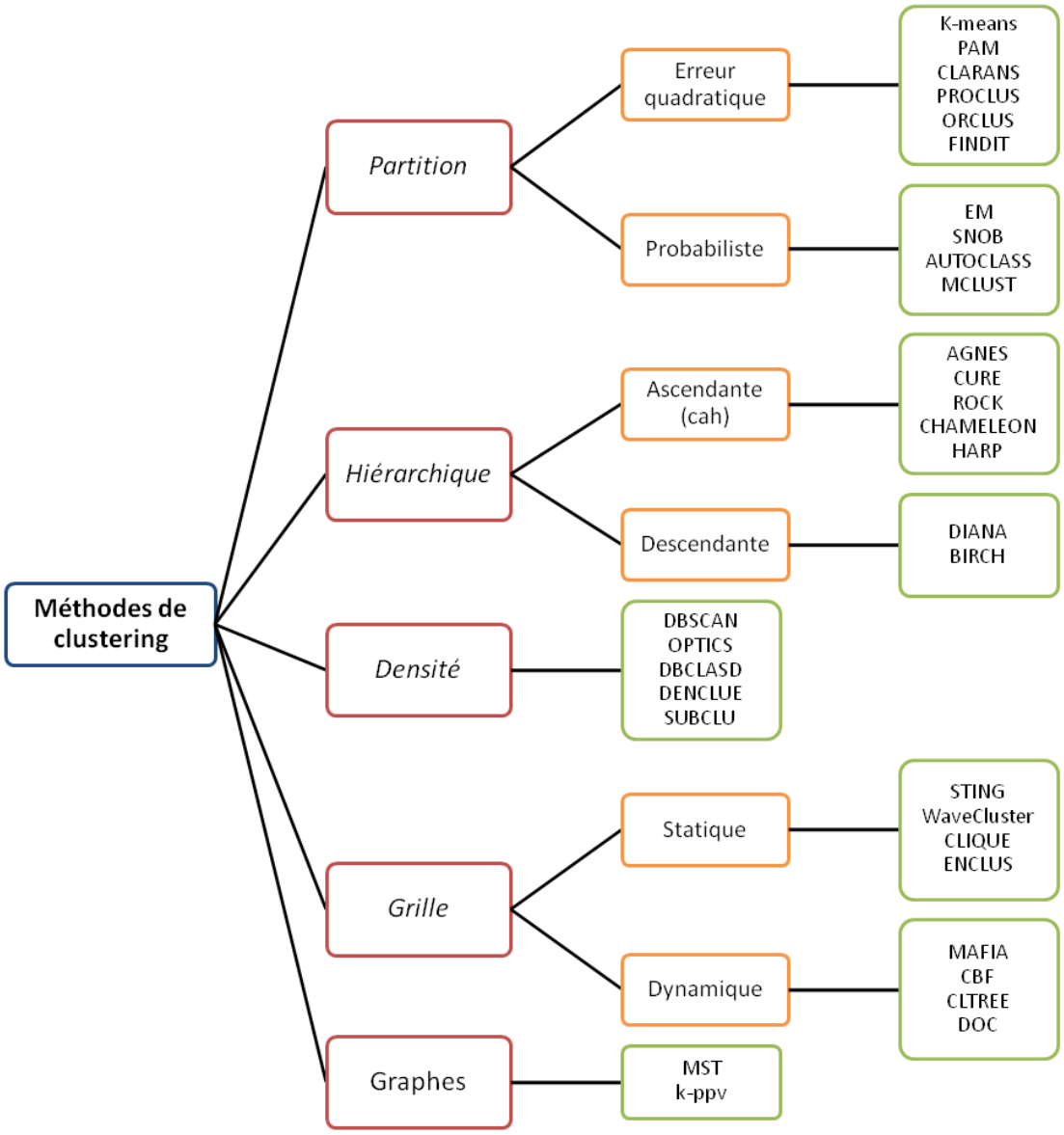


Figure 2.11 – Ensemble non-exhaustif de méthodes de classification non supervisées.

successives des éléments en classes (voir figure 2.12) et la classification descendante qui découpe successivement les regroupements d'objets. Les éléments sont représentés sous forme d'arbre. A chaque nœud correspond un ensemble de données qui a été obtenu par une fusion ou une division (suivant que la méthode soit ascendante ou descendante). Une fois l'arbre construit, il reste encore à le découper pour obtenir la formation des classes. Pour cela on peut faire un simple seuillage horizontal mais d'autres méthodes de coupures adaptatives donnent généralement de bons résultats [Rib98].

Il existe plusieurs manières de regrouper les éléments. Les trois plus connues sont les méthodes "single-link" , "complete-link" et "average link". Ce qui les différencie est la manière dont la similarité entre deux groupes est calculée. Pour le "single-link" la similarité de deux groupes correspond à la distance la plus faible entre toutes les paires de données entre les deux clusters, pour le "complete-link" elle correspond à la distance la plus grande et pour le "average link" elle correspond à la moyenne des distances. La première méthode a tendance à créer des clusters allongés et se trouve être sensible à l'effet de chaîne alors

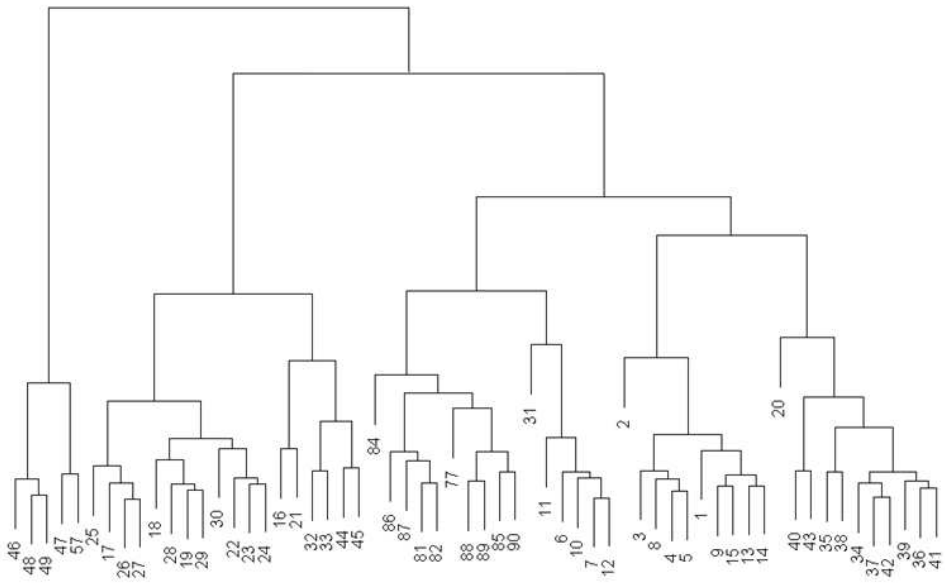


FIGURE 2.12 – Dendrogramme issu d'une classification hiérarchique ascendante.

que la seconde a tendance à créer des clusters très compacts.

DIANA [KR90] et AGNES [KR90] sont respectivement les méthodes de classification hiérarchique descendante et ascendante standards. Bisecting K-means [SKK00], et HARP [YCN04] sont des extensions de ces méthodes hiérarchiques. Bisecting K-means combine les k-means avec la classification descendante hiérarchique. Les k-means avec $k = 2$ sont exécutés récursivement pour subdiviser des groupes dont les caractéristiques intrinsèques ne sont pas suffisamment bonnes. HARP est basé sur un algorithme standard de CAH. L'algorithme utilise un ensemble de paramètres permettant de choisir les groupes à fusionner. Un processus de validation basé sur l'algorithme de Kolmogorov-Smirnov est utilisé afin de supprimer les dimensions dont la distribution est presque uniforme, car elles sont jugées inutiles.

Partitionnement

Ces techniques de classification créent directement un découpage de l'espace de telle sorte que la partition formée optimise un critère donné. Le principal avantage de cette méthode est d'avoir une complexité linéaire, ce qui rend ces méthodes bien plus rapides que les méthodes hiérarchiques.

Le critère le plus intuitif et le plus utilisé est le critère minimisant l'erreur quadratique moyenne (MSE). Pour utiliser cette méthode, il faut tout d'abord définir le nombre K de classes. Ensuite, l'algorithme se déroule en trois étapes illustrées par la figure 2.13 :

1. K éléments sont choisis aléatoirement, ils formeront les noyaux (le cœur des classes) initiaux.
2. Chaque autre élément est associé à la classe du noyau le plus proche
3. Le noyau est redéfini. Puis on retourne à l'étape 2 tant qu'au moins un élément change de classe.

La méthode du K-means définit le noyau comme étant le barycentre de tous les éléments d'une même classe alors que la méthode des K-médoïdes définit le noyau comme étant l'élément médian d'une classe.

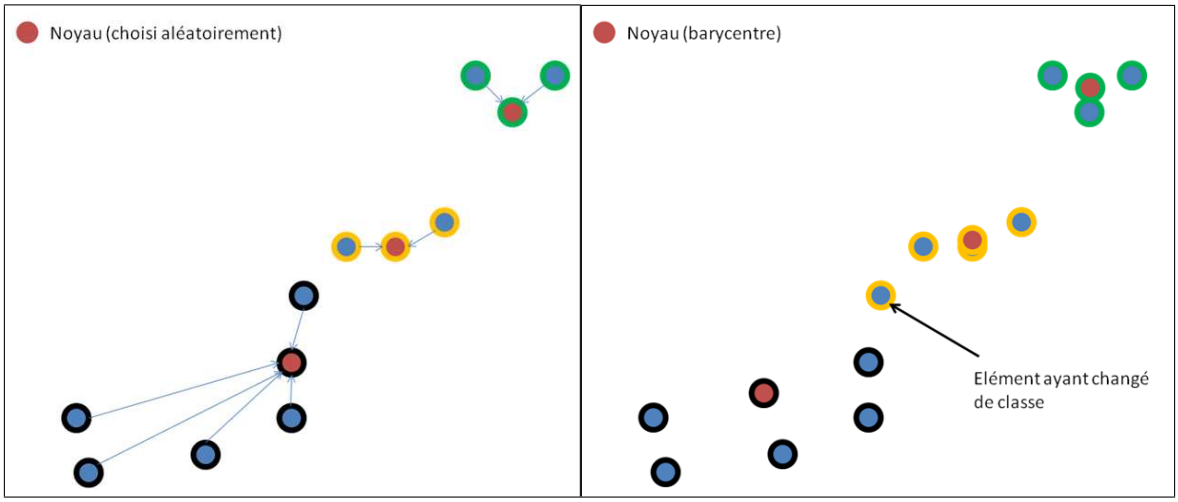


Figure 2.13 – Étapes 1 et 2 du K-means (à gauche) et 3 (à droite)

Le principal inconvénient vient du fait que les classes créées sont de formes hypersphériques. Il faut alors fusionner ou diviser des classes lorsque leur variance est trop faible ou trop forte. CLARANS [NH94], K-Harmonic Means [ZHD99], PROCLUS [AWY+99], ORCLUS [AY00], FINDIT [WLKL04], SPARCL [CHSZ08], sont des évolutions de la technique basée sur l'erreur quadratique moyenne.

Les méthodes probabilistes consistent à considérer que les éléments ont été générés par un ensemble de distributions de probabilité. En observant les données, on en détermine les paramètres des différentes distributions et on en déduit la probabilité qu'a un élément d'appartenir à une classe. On peut également construire un modèle statistique puis calculer la distance pondérée entre chaque élément [RTZK99]. La méthode la plus connue est l'approche Espérance Maximisation (Expectation Maximization - EM) [DLR77]. Cet algorithme comporte deux phases :

1. évaluation de l'espérance (E), où l'on calcule l'espérance de la vraisemblance en tenant compte des dernières variables observées. Ceci revient à trouver les classes des éléments en fonction des paramètres courants du modèle probabiliste.
2. maximisation (M), où l'on estime le maximum de vraisemblance des paramètres en maximisant la vraisemblance trouvée à l'étape E. Ceci revient à calculer les nouveaux paramètres du modèle en fonction des nouvelles classes.

MCLUST [FR06], P3C [MSE06] et STATPC [MS08] sont des techniques de clustering récentes basées sur EM.

2.4 Contribution à l'exploration sans connaissance *a priori* sur le contenu

L'objectif de la méthode présentée est de permettre à un utilisateur de labelliser rapidement un ensemble de documents. La plupart des techniques de classification sont basées sur un apprentissage supervisé, ce qui les rend difficilement utilisables dans notre contexte. En effet, nous ne disposons pas d'éléments déjà labellisés et le nombre de classes constituant la base n'est pas connu. C'est la raison pour laquelle nous avons privilégié une méthode basée sur une classification non supervisée des données.

Il n'existe pas à notre connaissance de proposition permettant de classer des images de documents dans ce contexte particulier. Pour répondre à la problématique de ce chapitre, nous proposons dans [AJD11a] une solution, étendue ensuite dans [AJD11b].

La méthodologie proposée est résumée en cinq étapes sur la figure 2.14 :

1. Des caractéristiques sont extraites des images de document.
2. Le nombre de classes k composant la base de données est estimé.
3. Un partitionnement des données en k classes est effectué.
4. Les documents sont classés à l'aide d'une technique de type CBIR avec retour de pertinence.
5. S'il reste des images à labelliser, on retourne en 2).

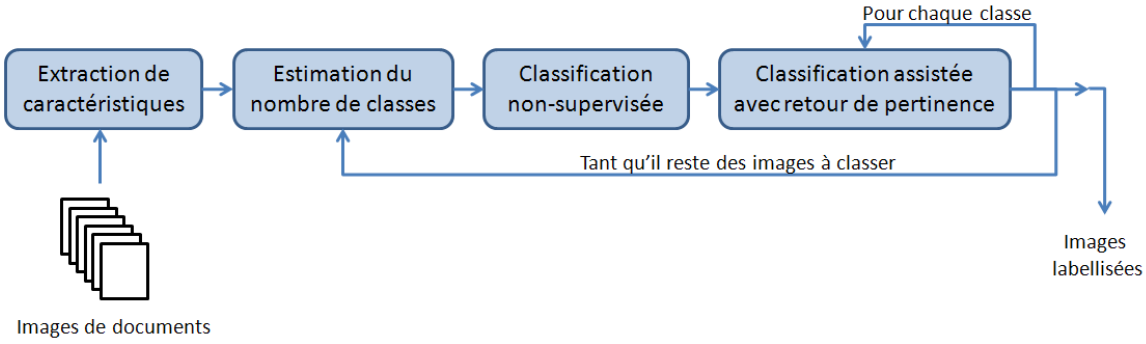


FIGURE 2.14 – Module d'aide à la classification.

Dans un premier temps, des descripteurs globaux sont extraits. Nous avons choisi d'implémenter les caractéristiques visuelles de [SDR01]et de [HDRT98] décrites dans la partie 2.1.2. Les caractéristiques de [SDR01] nécessitent que la structure des documents soit extraite. Dans notre cas, la segmentation de l'image est faite à l'aide de FineReader 9.0, car c'est une des rares solutions logicielles disponibles pour faire cette opération et donnant une segmentation en blocs de qualité, malgré une grande diversité des documents.

Sur la base des descripteurs extraits sur les images, le nombre de classes k composant un jeu de données est estimé. Cette estimation est faite car elle est nécessaire pour l'étape de partitionnement des données. Pour choisir le meilleur k, une des solutions courantes consiste à créer différents partitionnement et à les comparer afin de garder le partitionnement de meilleur qualité. Les auteurs de [HVB00] présentent des mesures de qualité de partitionnement. L'homogénéité (la distance intraclasse) ainsi que la séparation (la distance interclasse) sont les mesures de qualité les plus couramment utilisées. Le critère de la silhouette moyenne décrit dans [Rou87] et [PVDL02] permet de prendre en compte à la fois l'homogénéité et la séparation des différentes classes. Ceci correspond exactement à notre objectif : nous souhaitons regrouper au sein d'une même classe les documents les plus similaires tout en essayant de faire en sorte que des classes différentes soient éloignées les unes des autres. Nous avons donc choisi d'estimer le nombre de classes à l'aide du critère de la silhouette moyenne.

Une fois k estimé, un algorithme de classification non-supervisée permet d'obtenir un partitionnement en k classes avec pour chaque classe, un représentant pertinent qui sera appelé *image de référence*. Différentes méthodes de classification ont été testées. Le constat est le même que pour la segmentation de structure, il n'existe pas de méthode parfaite permettant d'obtenir les partitionnements optimaux. De plus, la densité des groupes et la

taille des grilles sont totalement inconnues puisque nous travaillons dans un contexte sans *a priori*. C'est pour cette raison que nous avons choisi de nous intéresser aux méthodes de partitionnement. Parmi ces méthodes nous avons privilégié PAM (Partitioning Around Medoids) par rapport à K-means pour deux raisons principales. La première est que PAM est moins sensible aux valeurs aberrantes, car basée sur les valeurs médianes plutôt que les valeurs moyennes. La seconde raison est que PAM renvoie un partitionnement avec un élément qui est le centre de chaque partition. Ce centre est appelé medoïde.

Enfin, un aspect innovant de notre méthode est la mise en place d'un module d'aide à la classification manuelle basée sur les techniques de CBIR ("Content Based Image Retrieval"). Les images présentées à l'utilisateur sont les représentants extraits automatiquement par l'étape précédente (*images de référence*). Notre proposition permet de regrouper puis de montrer à l'utilisateur des images possédant de fortes similarités visuelles. Le tri manuel est ainsi rendu plus simple et plus rapide, permettant à l'utilisateur de labelliser en une seule fois un ensemble d'images ayant un aspect visuel similaire. Afin d'améliorer la mesure de similarité au sein d'une même classe et d'affiner les descripteurs utilisés, il est possible d'utiliser des méthodes de retour de pertinence [ZH03].

La figure 2.15 résume les deux principales étapes de notre module d'aide à l'exploration d'une base inconnue.

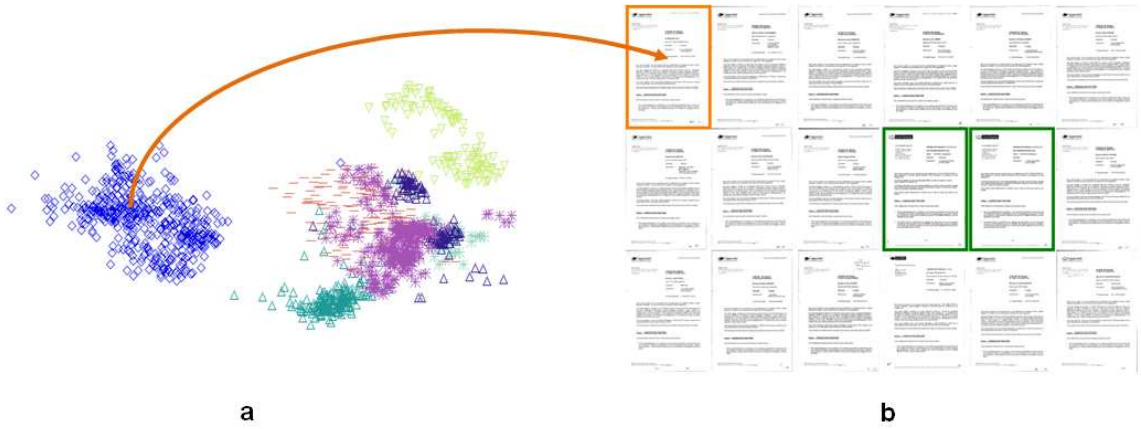


Figure 2.15 – Résumé de la méthodologie de classification assistée. a) Les caractéristiques des images sont extraites, le nombre de classes est estimé, l'ensemble des données sont partitionnées. b) Chaque medoïde devient une image référence utilisée pour rechercher les images similaires (image encadrée en orange). Les images les plus similaires sont montrées à l'utilisateur, ce dernier sélectionne les mauvaises images (encadrée en vert). Après cette interaction, une sélection de caractéristiques est appliquée pour améliorer la recherche d'images similaires. Le processus complet est de nouveaux appliqué sur les images non labellisées.

2.4.1 Présentation des bases

Cinq bases de documents issues de productions industrielles ont été labellisées manuellement. Les bases $DB1$ et $DB3$ sont composées uniquement de factures. Elles contiennent respectivement 1509 et 2574 documents réparties en 7 et 33 classes. Des exemples d'images sont présentés sur la figure 2.16.

Les bases $DB2$, $DB4$ et $DB5$ sont composées de documents de ressources humaines divers tels que des contrats de travail, des conventions de stage, des déclarations d'embauche, des fiches d'entretien personnel, de renseignement administratif, etc. Ces bases

FIGURE 2.16 – Exemples d'images des bases $BD1$ et $BD3$. Deux images de trois classes différentes sont présentées.

contiennent respectivement 883, 3352 et 5962 documents répartis en 19, 30 et 48 classes. Des exemples d'images sont présentés sur la figure 2.17.

FIGURE 2.17 – Exemples d'images des bases $BD2$, $BD4$ et $BD5$. Deux images de trois classes différentes sont présentées.

2.4.2 Estimation du nombre de classes k

Lors de la première création d'une base d'images, le nombre de classes n'est pas nécessairement connu. Une première étape consiste donc à estimer le nombre de classes présentes dans la base.

La silhouette d'un élément x (dans notre cas, une image de document représentée par un vecteur de caractéristiques) est calculée à partir de la moyenne des distances entre l'élément x et le reste des éléments $a(x)$ de sa classe C_x. Le minimum des distances euclidiennes moyennes entre l'élément x et les autres centres de classes $b(x)$ est ensuite calculé. La silhouette de l'élément x se calcule de la manière suivante :

$$silh(x) = \frac{b(x) - a(x)}{\max(a(x), b(x))}$$

Une fois la silhouette calculée sur chaque élément, il est possible de calculer la moyenne des silhouettes pour une classe :

$$S_{C_i} = \frac{\sum_{j \in C_i} silh(j)}{Card(C_i)}$$

Finalement, la moyenne des silhouettes des classes est calculée

$$GS = \frac{\sum_{i=1}^{k} S_{C_i}}{k}$$

Plus la valeur de GS est proche de 1 plus la partition est de bonne qualité, car elle traduit une forte variabilité inter-classes et faible variabilité intraclasse. Afin de sélectionner le

nombre de classes qui optimise le partitionnement, GS est calculée pour chaque valeur de k allant de 3 à K où K est spécifié par l'utilisateur (pour les tests, K a été fixé à $\sqrt{(tailleBase/2)}$). Le k retenu est celui qui maximise GS.

Des tests ont été effectués sur les bases présentées dans la section 2.4.1. Le tableau 2.1 illustre la pertinence du critère de la silhouette pour estimer automatiquement le nombre de classes composant une base d'images. On remarque que pour les bases $DB1$ et $DB3$, le DBk estimé est très proche du k réel parce que le visuel des factures d'une même entreprise présente très peu de variation, donc les distances intra-classes sont faibles. Pour les bases $DB2$,$DB4$ et $DB5$, k est sous estimé, car des documents des classes différentes se ressemblent. La figure 2.18 montre l'évolution de la silhouette en fonction de k pour les bases $DB1$ et $DB5$.

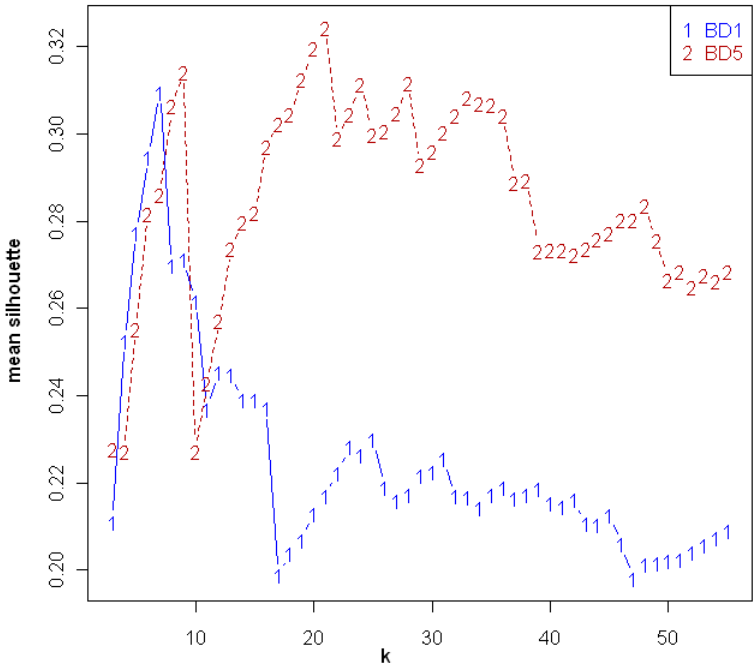


Figure 2.18 – Courbe de la silhouette en fonction du nombre de classes k pour la base $DB1$ (1509 documents, 7 classes) et la base $DB5$ (5962 documents, 48 classes). Le nombre de classes estimé est déterminé par le maximum de la silhouette moyenne, dans notre cas : $k = 7$ pour $DB1$ et $k = 21$ pour $DB5$. On remarque que la silhouette moyenne de la base $DB5$ a un maximum local pour $k = 48$, qui est le "vrai" nombre de classes pour cette base.

Le F_1score est une métrique permettant de mesurer les performances du partitionnement, elle est basée sur le rappel r et la precision p tel que : $F_1score = 2 \cdot \frac{p*r}{p+r}$. Le tableau 2.1 montre que le F_1score est similaire entre le k estimé et le k réel. Ceci peut être expliqué par le fait que les techniques de classification telles que PAM ou K-means ont tendance à séparer les classes possédant de nombreux éléments et a fusionner les classes qui en ont peu. Enfin, on remarque également que dans certains cas de figure, le k estimé donne un meilleur partitionnement qui si on avait utilisé le k réel, ce qui confirme que trouver le nombre exacte de classe n'est pas la chose la plus importante, mieux vaut créer un bon partitionnement des données au sens de la silhouette.

Une fois que le nombre de classes est estimé, l'ensemble des données est partitionné à l'aide d'un algorithme de classification non supervisé. Pour cela nous avons choisi la méthode de partitionnement PAM. À la fin de cette première étape, nous avons donc

TABLE 2.1 – Synthèse des tests réalisés sur l'estimation du nombre de classes à l'aide de la mesure de silhouette. Pour les bases $DB1$ et $DB3$, le k estimé est très proche du k réel. Pour les bases $DB2$ le k est sous-évalué. Pour les bases $DB4$ et $DB5$, le nombre de classes est également sous-évalué mais un meilleur F_1score est obtenu. Une étude approfondie de ces deux bases montre que les documents composant certaines classes sont visuellement similaires et que certaines des classes sont marginales, car elles contiennent peu d'éléments.

		Vrai k		k estimé	
Base	nb images	k	PAM F_1score	k	PAM F_1score
DB1	1509	7	0,7469	7	0,7469
DB2	883	19	0,6746	11	0,6668
DB3	2574	33	0,5778	35	0,5864
DB4	3352	30	0,4371	16	0,5660
DB5	5962	48	0,5823	21	0,5914

un classement des images, les distances relatives entre chaque image sont connues. Ces regroupements ne sont pas parfaits mais il y a tout de même de fortes chances pour que les images proches des centres des classes soient relativement similaires entres elles d'un point de vue visuel. De plus, une partie des erreurs vont être corrigées dans la partie suivante, grâce à l'utilisation d'un retour de pertinence.

2.4.3 Classification assistée avec retour de pertinence

L'estimation automatique du nombre de classes k permet, à partir des médoïdes de chacune des k classes, d'extraire k images de référence représentant au mieux chacune des classes. En nous inspirant des techniques de CBIR telles que [DLW05] ou [JED+05], nous proposons de trier les images par ressemblance décroissante. La classification assistée est réalisée en montrant successivement à l'utilisateur une des images de référence accompagnée des 50 images qui lui sont le plus similaires. L'utilisateur indique les images qui n'appartiennent pas à la même classe que l'image présentée, une nouvelle série de 50 images est alors proposée pour la même classe. Durant le traitement d'une classe, si 5 images ou plus sont indiquées comme n'appartenant pas à la classe, un algorithme de sélection de caractéristiques est exécuté. Si 19 images ou plus sont sélectionnées par l'utilisateur, on passe à la classe suivante. La figure 2.19 illustre ce principe.

À partir des indications de l'utilisateur, un retour de pertinence peut être utilisé afin d'améliorer la mesure de similarité entre les documents. Ce retour de pertinence permet de réaliser une sélection de caractéristiques. Cette étape est importante, puisque la nature des documents n'est pas connue à l'avance il est nécessaire d'extraire de nombreuses caractéristiques différentes, la sélection de caractéristiques permet de ne garder ou de pondérer uniquement celles qui sont pertinentes pour une classe donnée. L'algorithme de sélection de caractéristiques Fealect [1] permet de pondérer les dimensions du vecteur de descripteurs des images. Pour affecter ces poids, cet algorithme de sélection commence par générer plusieurs sous-ensembles d'éléments. L'algorithme détermine ensuite l'apport de chaque caractéristique dans le processus de classification à l'aide d'une régression des moindres angles (Least Angle Regression [EHJT04]). Notre objectif ici n'est pas d'analyser précisément le comportement de l'outil permettant de faire de la sélection de caractéristique

1. http ://cran.r-project.org/web/packages/FeaLect/FeaLect.pdf

Figure 2.19 – Classification assistée avec retour de pertinence. Les documents sont triés par similarité décroissante. La première image encadrée en gris est l' *image de référence*. Les deux images encadrées en noir ont été sélectionnées par l'utilisateur car elles n'appartiennent pas à la même classe. Ces images vont être utilisées pour le retour de pertinence.

mais simplement de montrer l'intérêt d'utiliser un tel outil dans une chaîne de classification d'images de documents. D'autres méthodes de sélection de caractéristiques sont disponibles dans la littérature [KR10], [ZWL10], [BMD09], [NPBT07].

2.4.4 Tests et résultats

Les tableaux 2.2 et 2.3 synthétisent les résultats obtenus pour la classification des 5 bases de documents de type "ressources humaines". Les descripteurs utilisés consistent en de nombreuses statistiques (somme, moyenne, médiane, écart-type, minimum, maximum) de nombreux paramètres tels que l'air et le périmètre des composantes connexes, des boites englobantes des composantes connexes, des blocs de texte, des tableaux et des images, du nombre de lignes horizontales et verticales. Ils sont décrit dans [SDR01]. L'extraction de la mise en page est faite avec FineReader 9.0. Même si l'extraction des blocs n'est pas parfaite, cela n'impacte pas trop les statistiques que l'on calcul avec car les positions relatives des blocs ne sont pas utilisées.

Comme on peut le voir sur la figure 2.20, la sélection de caractéristiques est utile pour les classes qui ont beaucoup d'images. Cette figure illustre précisément ce qu'il se passe pendant la labellisation d'une *image de référence* avec et sans la sélection de caractéristiques. L'axe des x représente le nombre d'itérations *i.e.* le nombre de fois que n_{Im} images ont été affichées. Quatre valeurs sont affichées. La première représente le nombre d'images restantes de la classe à labelliser. La seconde est la troisième représentent respectivement le nombre d'images similaires et dissimilaires montrées à l'utilisateur. Le nombre de mauvaises images est égal au nombre d'images sélectionnées par l'utilisateur. Si n_{Wrong} images sont sélectionnées, le processus est arrêté. Le processus est également arrêté si toutes les images de la classe sont labellisées. Lorsqu'il y a plus de n_{FS} images dissimilaires la sé-

TABLE 2.2 – Résultats de la classification assistée avec et sans sélection de caractéristiques. Un gain moyen de $9,2760\%$ est obtenu.

		Pourcentage de base labellisée	
base	images	avec FS	sans FS
DB1	1509	57,3227	85,6858
DB2	883	78,2559	81,0872
DB3	2574	76,8453	80,9634
DB4	3352	73,4486	79,8031
DB5	5962	60,2314	64,9446

lection de caractéristiques est lancée et les distances sont recalculées pour la prochaine itération. S'il n'y a pas suffisamment d'images dissimilaires, la dernière sélection de caractéristiques est gardée. Quand la sélection de caractéristiques est activée, une croix est dessinée sur les courbes du graphique. Par exemple, à la seconde itération il y a 46 documents de la même classe et 4 documents d'une autre classe. Parce qu'il y avait eu aussi 1 document sélectionné à la première itération, la sélection de caractéristiques est exécutée. Dans ce cas, 9 caractéristiques parmi les 99 sont sélectionnées. Les images sélectionnées ont peu de différences parce que les contrats d'embauche ont de légères différences en fonction des années où ils ont été produits. Par exemple, une des caractéristiques sélectionnée est le nombre de paragraphes. On remarque que ce dernier est plus grand dans l'image sélectionnée que dans les images qui appartiennent à la classe de l' *image de référence*.

Afin de montrer l'intérêt de l'utilisation de l'algorithme de sélection de caractéristiques, nous avons comparé le processus de labellisation avec et sans activation de la sélection de caractéristiques. Les résultats du tableau 2.2 montrent que la sélection de caractéristiques augmente systématiquement le pourcentage de documents labellisés. Pour chaque base, deux valeurs sont calculées. La première valeur *without FS* représente le pourcentage de documents labellisés par notre méthode de classification assistée sans utiliser la sélection de caractéristiques. La seconde valeur *with FS* représente le pourcentage de documents labellisés par le module de classification assisté avec l'utilisation de la sélection de caractéristiques. Grâce à l'étape de sélection de caractéristiques, le taux de classification est augmenté en moyenne de plus de 9%.

Pour les entreprises, la conséquence pratique de disposer d'un tel outil en production est que la classification assistée de documents permet de labelliser plus rapidement les documents qu'un triage séquentiel "classique". Les tests réalisés en production sur plusieurs milliers d'images et avec plusieurs personnes ont montré qu'il faut en moyenne 8 secondes pour un opérateur professionnel pour labelliser une image prise au hasard. Avec la classification assistée, il faut en moyenne 25 secondes pour sélectionner les images correspondant à la requête parmi les 50 images qui sont proposées. Le tableau 2.3 montre que l'utilisation de la classification assistée divise le temps de labellisation par plus de 3 pour un bouclage complet de la chaîne.

Après cette étape de classification assistée, environ 20% des documents ne sont pas labellisés. On souhaite labelliser ces documents en bouclant sur la chaîne complète. k est estimé une fois de plus, mais uniquement avec les documents non labellisés à l'étape précédente. On boucle alors sur la chaîne complète jusqu'à que tous les documents soient labellisés.

La figure 2.21 montre l'évolution du pourcentage de documents labellisés en fonction

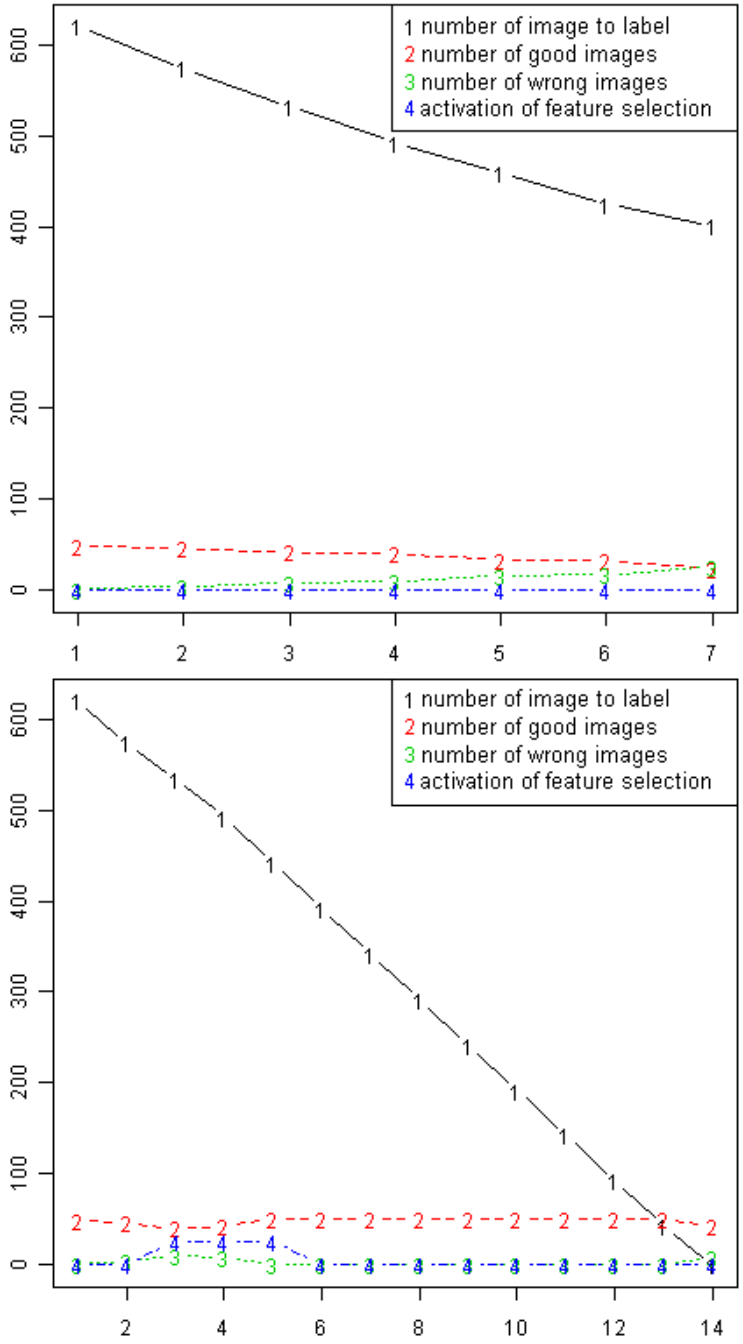


Figure 2.20 – Utilité de la sélection de caractéristiques. Les courbes tracées correspondent à la labellisation du quatrième médoïde de la base $DB4$. Sur la première courbe, la sélection de caractéristiques n'est pas utilisée et 401 documents ne sont pas labellisés. Sur la seconde,la sélection de caractéristiques est activée. Tous les documents de la classe ont été labellisés. La distance entre les documents est recalculée dans les itérations 3,4 et 5. Après la 5ème itération la sélection de caractéristiques n'est plus appelée, les distances restent alors les mêmes.

du nombre de bouclages. Toutes les bases sont labellisées en deux ou trois boucles. Par exemple 60% de la base $DB5$ est labellisée dans la première boucle, 91% dans la seconde

TABLE 2.3 – Temps utilisé pour labelliser une base. La classification assistée avec sélection de caractéristiques permet de diviser le temps de labellisation par $3,4$ en moyenne.

		Temps de labellisation (minutes)	
base	images	classification manuelle	classification assistée
DB1	1509	201,2	42,1
DB2	883	117,7	38,5
DB3	2574	343,2	101,1
DB4	3352	447,0	131,1
DB5	5962	794,9	338,7

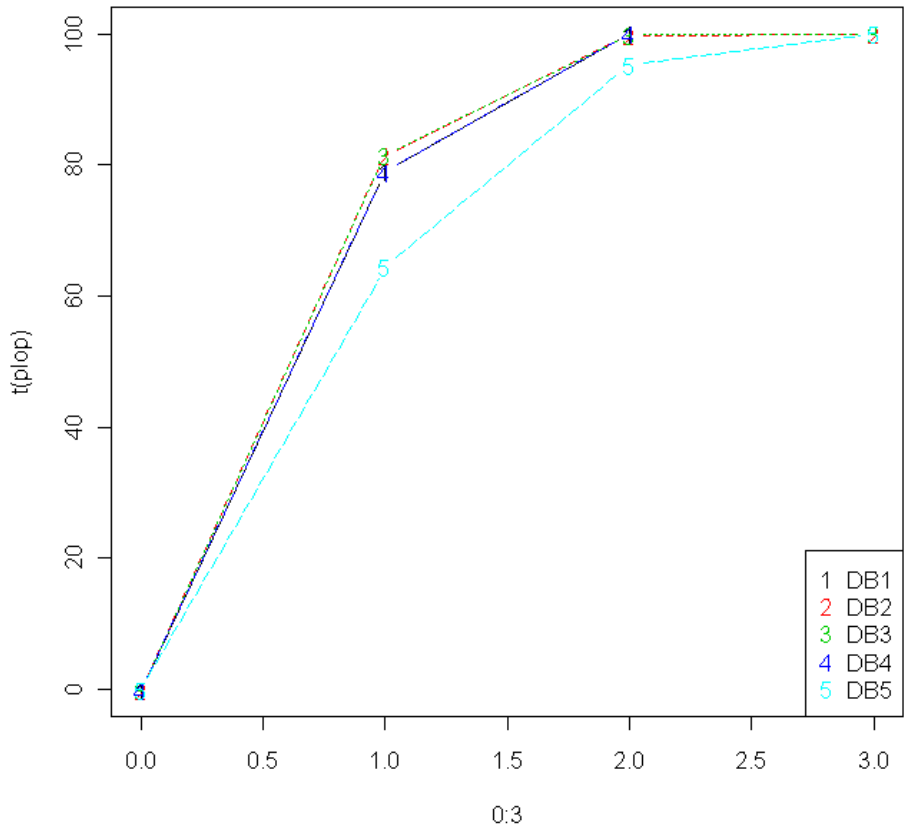


FIGURE 2.21 – Pourcentage de base labellisé en fonction de nombre de bouclages. Les bases sont entièrement labellisées en 3 boucles, même $DB5$, la base la plus grande.

boucle puis la totalité dans la dernière.

Il est à noter que chaque image est associée à une image de référence correspondant au même type. Cependant, en cas de sur segmentation (estimation du nombre de classes trop grand) il peut y avoir plusieurs images de référence représentant le même type de document. Il faudra manuellement indiquer que ces références font partie d'un même type afin de leur affecter au final le même label. La sous-segmentation (estimation du nombre

de classes trop faible) impliquera qu'au moins une des classes ne sera pas représentée. Au moins un bouclage sera alors nécessaire.

2.5 Conclusion et perspectives

Ce chapitre présente de nouvelles méthodologies de classification d'images de documents de type industriels dans une optique d'exploration lorsque qu'aucune connaissance n'est mise à disposition du système. Le premier point important de notre proposition consiste en l'estimation du nombre de catégories de documents composant un jeu de données, là où les méthodes de l'état de l'art considèrent que cette information est donnée par l'utilisateur. La grande nouveauté de notre proposition réside dans la mise en place d'un système de classification adapté aux images de documents et basé sur la "requête par l'exemple" de style CBIR. Notre proposition permet de garder l'opérateur humain au cœur du système, garantissant que les images soient correctement labellisées. Au fur et à mesure que l'utilisateur classe les images de la base, notre système se spécialise permettant ainsi à l'opérateur humain de labelliser encore plus rapidement de grandes quantités d'images. Les tests effectués mettent en avant un gain de temps du processus d'indexation de l'ordre d'un facteur 3. Cette méthodologie est mise en place depuis deux ans en production et à fait ses preuves sur de nombreuses prestations.

Les principales perspectives portent sur l'amélioration de l'étape de l'estimation du nombre k de classes présentes dans la base. Pour cela, nous envisageons de combiner notre estimateur actuel avec d'autres estimateurs de qualité de partitionnement tels que le BIC (Bayesian Information Criterion) [ZHF08]. Il serait également intéressant de comparer les différences entre les techniques de partitionnement de l'état de l'art (K-means, classification ascendante hiérarchique, etc.) afin de choisir une méthode de classification optimale. Il est également possible de combiner plusieurs techniques de partitionnement afin de trouver un partitionnement optimal et de trouver le nombre de classes tel que cela est fait dans les méthodes de "consensus clustering" [SAJ10].

Enfin, une dernière piste d'amélioration des performances de la classification des images de documents réside dans la réduction du décalage sémantique (semantic gap) qu'il y a entre les attentes de l'utilisateur et les vecteurs de caractéristiques utilisés. Pour cela nous travaillons actuellement sur une adaptation de la méthodologie permettant à l'utilisateur de sélectionner des éléments de contenu (un logo, un tableau, etc.).

Chapitre 3

Recherche d'images de documents semi-structurés

Dans ce chapitre, nous nous plaçons dans la situation où les entreprises de dématérialisation souhaitent déterminer si un type de document particulier est présent dans un ensemble d'images. Concrètement, la technique proposée dans ce chapitre peut s'appliquer à des documents tels que des pièces d'identité (cartes d'identité, passeports, titre de séjour, etc.), des notes de frais (ticket de train, reçu d'hôtel, facture de taxi, etc.) et n'importe quelles catégories de documents, du moment qu'ils soient semi-structurés. Dans ce chapitre, nous proposons donc de traiter des documents dont les éléments d'une même classe possèdent des informations fixes d'un exemplaire à un autre (mise en page commune).

Quelques exemples d'images à analyser sont présentés sur la figure 3.2. Si un document est trouvé sur l'image, il doit pouvoir être extrait précisément afin d'être transmis à un logiciel de LAD (Lecture Automatique de Document) qui effectuera la lecture de certaines informations. L'objectif de ce chapitre est de mettre en place une méthode de recherche "non-exacte" d'images de documents à partir d'une image exemple fournie par un utilisateur. Cette recherche est dite "non-exacte" puisque les documents à retrouver possèdent des différences (sur le contenu, la position de certains blocs, etc.) par rapport à l'image requête. Par exemple, on souhaite chercher toutes les cartes d'identité françaises dans une base et non pas la carte d'identité d'une personne en particulier. Les éléments recherchés possèdent donc une forte similarité avec l'élément requête, mais possèdent tout de même quelques différences. À noter également que les méthodes présentées dans ce chapitre se concentrent de la recherche d'une classe d'images à la fois.

Certain documents présentent des particularités qui leurs sont propres. Les pièces d'identité contiennent généralement une photo d'identité, des logos, un fond texturé et du texte. Elle sont également protégées contre les contrefaçons, certaines informations (comme des hologrammes) disparaissent ou sont déformées lors de la numérisation. Les notes de frais ont la particularité d'être généralement de mauvaise qualité. Il est fréquent que l'encre soit partiellement effacée ou encore que le papier soit déchiré, plié ou froissé. Des exemples de notes de frais de mauvaise qualité sont présentés sur la figure 3.1.

La localisation et l'extraction des documents d'identité ou des notes de frais est complexe car les documents peuvent être disposés d'une manière quelconque sur l'image. Pour les cartes d'identité, cela est dû au fait que l'utilisateur peut photocopier ou numériser puis imprimer sa carte lui-même. Dans le cas des notes de frais cela est dû au fait que certaines entreprises demandent à leurs employés de coller leur notes de frais sur une feuille. Il peut également arriver que l'image du document ne soit pas à l'échelle standard et que certaines parties soient coupées ou occultées par d'autres informations. De plus il peut

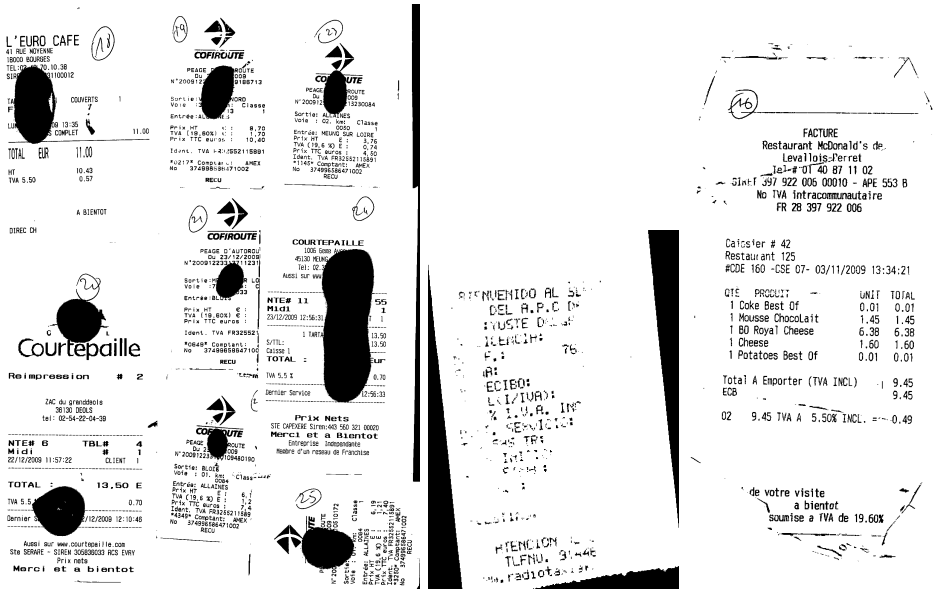


FIGURE 3.1 – Exemples de notes de frais de mauvaise qualité. Les documents de l'image de gauche sont collés sur une feuille. La colle utilisée a détériorée les images. Une partie de l'encre de l'image du milieu est effacée. Le document de droite a été froissé, les coins sont pliés et déchirés.

y avoir des documents de différents types sur une même image. Notre problématique se rapproche donc de la recherche de sous-image (*sub-image retrieval*) [DPGM04].

Notre approche permet *in fine* de reconnaitre et d'extraire automatiquement des images d'une base de documents pour faciliter leur classement et leur indexation. Nous évaluons notre proposition en grande partie en fonction du nombre de fausses détections et du nombre de documents oubliés. Selon le contexte industriel, il peut être préférable soit de minimiser le nombre de fausses détections quitte à devoir labelliser manuellement des documents oubliés par le système ; soit de minimiser le nombre de documents oubliés quitte à avoir de nombreuses fausses détections.

Les contraintes industrielles nous pousseront à choisir une méthode privilégiant donc la précision au rappel : en effet il est préférable que les décisions prises par l'algorithme de reconnaissance soient correctes le plus souvent possible. Mieux vaut ne pas détecter certains documents présents dans l'image plutôt que faire de mauvaises détections. En effet si l'utilisateur ne peux pas avoir confiance en la décision du programme, il devra contrôler manuellement l'ensemble des résultats.

3.1 Recherche de sous-images

Notre problématique consiste à reconnaitre et à extraire des portions d'images. Cette problématique a été abordée dans le cadre de la recherche dans des bases d'images naturelles. Des propositions ont été faites pour permettre d'aligner des images les unes par rapport aux autres [YSST07], pour rechercher des logos [RL09], [PAK10] ou encore pour créer des panoramas [BL07]. La technique de reconnaissance d'objets actuellement la plus utilisée est celle proposée par Lowe [Low04]. Toutes ces techniques sont basées sur la détection de points d'intérêts tels que SIFT [Low04] ou SURF [BETVG08]. Ces techniques sont très utilisées car robustes au changement d'échelle, à la rotation, au bruit, au changement de luminosité et aux occlusions.

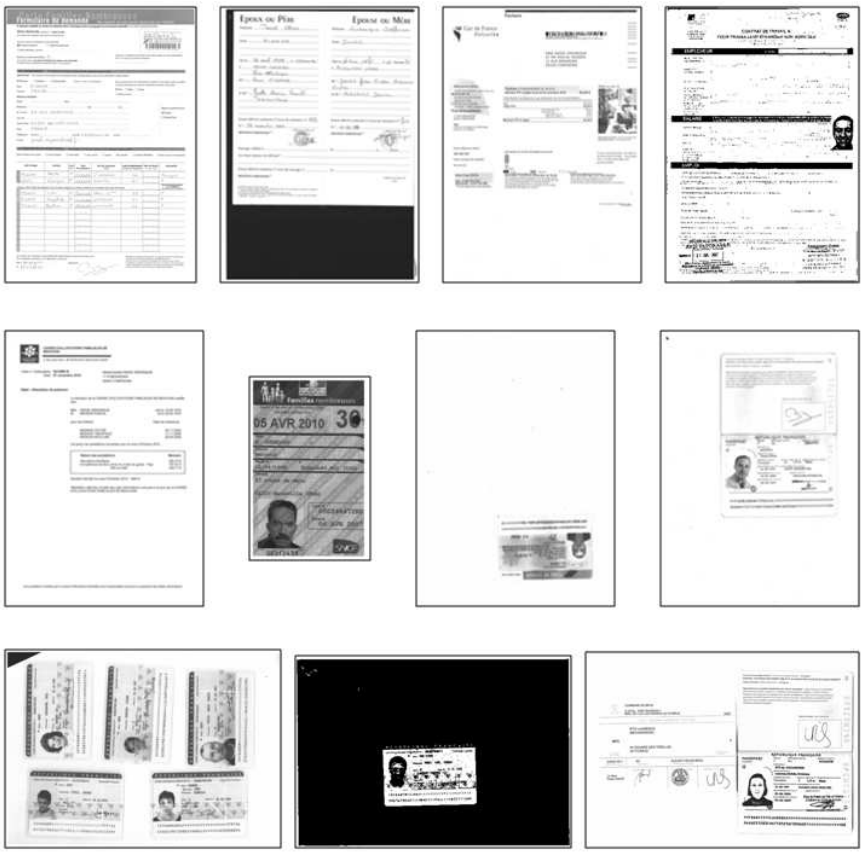


Figure 3.2 – Exemples de documents présents dans la base (de gauche à droite et de haut en bas) : un formulaire, un acte de mariage, une facture, un contrat, une attestation, une carte d'abonnement, un titre de séjour, un passeport français, cinq cartes d'identité françaises, une carte d'identité française et un passeport français accompagné d'un autre document.

Ces techniques n'ont a ce jour jamais été utilisées pour la recherche de documents semi-structurés. Comme le souligne les auteurs de [US09] les techniques basées sur l'utilisation de points d'intérêt tels que SIFT ou SURF ne sont pas du tout adaptées aux images de documents car elles échouent sur des images faiblement texturées ou qui ont des patterns répétitifs (tels que le texte, pour les images de documents).

Agam *et. al* [AAF+07] traitent le problème de la reconnaissance d'images basée sur le contenu (CBIR) appliquée à des images de documents complexes. Leur contexte est le même que le notre. Les documents sont imprimés et parfois annotés manuellement. Les documents peuvent contenir des graphiques, des tableaux et d'autres éléments non textuels. Habituellement, seul l'OCR est utilisé pour la recherche de documents et les auteurs suggèrent d'utiliser également des informations structurelles et non textuelles. Pour faire ceci, de nombreux prétraitements sont appliqués, notamment le SDK de ABBYY [abb] mais également d'autres bibliothèques (DocLib, le système de reconnaissance de signature de CEDAR et Text Solutions). Parmi les prétraitements appliqués on peut citer : la suppression de bruit, la détermination de l'angle d'inclinaison du document dans l'image et la segmentation de la mise en page. Le principal inconvénient de cette méthode est que chaque étape de prétraitement est susceptible de générer des erreurs qui s'accumuleront au fur et à mesure. Par exemple, la segmentation de la mise en page d'un document est

très complexe et le SDK de Abbyy est bien souvent mis en défaut. Des études ont montré que le SDK éprouvait des difficultés sur les documents multi-orientés et de tailles multiples [CK11], sur la reconnaissance d'images et de graphiques [CPA11] et sur des documents complexes tels que les documents historiques [ACPP11].

Les auteurs de [TKI11] présentent une méthode permettant de retrouver un document identique à la requête, ceci en temps réel et dans une base de données d'un million de documents. Les lettres sont regroupées par mot en appliquant une gaussienne afin de flouter l'image. Les paramètres de la gaussienne sont déterminés à partir de la taille estimée des caractères (la racine carrée du mode des aires des composantes connexes). Les points d'intérêt extraits sont les centres des mots. Chaque point est ensuite caractérisé par un ensemble de descripteurs présentés sur la figure 3.3. Une base de documents est créée sous la forme d'une de table de hashage. Cette base contient l'ensemble des descripteurs d'un millions de documents. Pour reconnaitre une image requête, chacun de ses points d'intérêt est mis en correspondance avec un des points d'intérêt stocké dans la base. Un vote majoritaire élit l'image de la base dont les points d'intérêt ont été le plus utilisés pour la mise en correspondance comme étant l'image cherchée. Cette technique est très performante mais l'inconvénient est qu'elle est entièrement basée sur le texte et permet de faire de la recherche de document exact uniquement. Il faut donc que le texte soit segmenté proprement et les lettres regroupées efficacement. Selon les auteurs, cette méthode donne de moins bons résultats lorsque des illustrations sont présentes sur les documents. Or, comme nous l'avons présenté dans la problématique, une partie des documents à reconnaitre contiennent des images, photos, logos, etc. De plus nous souhaitons faire une mise en correspondance "non-exacte", cette technique ne pourra donc pas être appliquée dans notre cas.

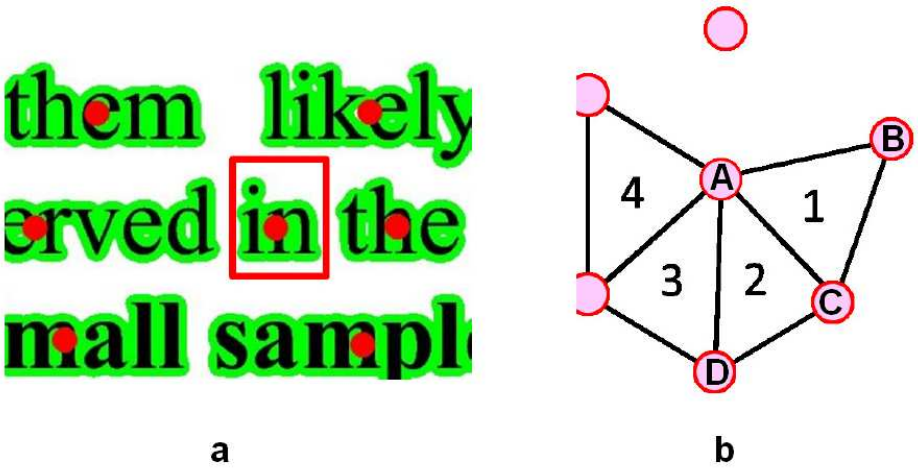


FIGURE 3.3 – Localisation et descriptions des points d'intérêt de [TKI11]. a) Localisation des points d'intérêt. b) Description d'un point. Les k plus proches points d'intérêts (avec $k = 6$) du point à décrire sont recherchés. Un "invariant" est calculé comme étant le ratio des aires des triangles ACD et ABC, où A est le points d'intérêt à caractériser et B, C et D sont 4 des 6 PPV. Toutes les combinaisons d'invariants sont calculées (soit un total de 15 invariants). À ce vecteur sont ajoutées 6 autres valeurs basées également sur les PPV du point d'intérêt.

Hull *et al.* [HEG+07] présentent une méthode de recherche d'images de documents basée sur les N-grammes. Le nombre de caractères de chaque mot est calculé. Par exemple, les 3-grammes de “structured document image matching" sera : “10 - 8 - 5" et “8 - 5 -

8", car le nombre de lettres de “structured", “document", “image" et “matching" sont respectivement 10, 8, 5 et 8. Des N-grammes horizontaux et verticaux sont combinés. Tout comme la technique de Takeda *et. al*, l'inconvénient de cette méthode est qu'elle est utilisée uniquement pour la recherche d'image de document contenant principalement du texte.

Les méthodes usuellement appliquées aux images de documents ne peuvent pas être utilisées sur nos images pour deux raisons principales. D'une part le contenu des documents industriels est trop varié (présence de tableaux, photos, schémas, logos, tampons, etc.). D'autre part, nous souhaitons effectuer une recherche non-exacte contrairement à ce qui est proposé par [TKI11] et [HEG+07]. Nous proposons une adaptation des techniques issues de l'état de l'art des méthodes de reconnaissance d'objets dans les images naturelles basées sur SIFT [Low04] ou SURF [BETVG08].

Dans la suite de cette section, nous allons présenter dans un premier temps une chaîne standard de reconnaissance d'objet habituellement utilisée pour la reconnaissance d'objet dans les images naturelles. Grâce à un test sur base réelle nous observerons, tout comme les auteurs de [US09], que cette technique donne de mauvais résultats sur les images de documents. Nous proposerons alors plusieurs améliorations et adaptations qui permettront d'obtenir de bien meilleurs performances, rendant utilisable la technique dans un contexte industriel. La robustesse au bruit de Kanungo [KHP93] sera également étudiée. En effet, la déformation des caractères est un défaut fréquemment rencontré dans les bases d'images numérisées et nous souhaitons évaluer la robustesse de notre méthode face à ce bruit typique.

3.2 Reconnaissance d'objet standard

L'algorithme que nous proposons pour la reconnaissance et l'extraction de documents semi-structurés est décrit dans la figure 3.4. Notre méthode s'inspire d'une des méthodes les plus classiques : celle de Lowe [Low04]. La première étape consiste à sélectionner une image requête. Cette image est simplement un exemple du type d'image que l'on souhaite reconnaitre. Les points d'intérêt des différentes images requêtes et des images à analyser de la base de données sont extraits et décrits à l'aide de la méthode SURF [BETVG08]. Ensuite les points d'intérêt d'une image requête sont mis en correspondance avec les points d'intérêt de chacune des images de la base. Pour cela, l'algorithme de recherche rapide et approximative de plus proches voisins FLANN [ML09] est utilisé. Enfin, la transformation (modèle à 4 paramètres) permettant de passer de l'image requête à l'image analysée est estimée avec RANSAC [FB81]. Cette opération permet de localiser très précisément le modèle dans l'image requête. Selon Lowe, si au moins 3 mises en correspondance valident la transformation géométrique, l'objet (dans notre cas, le document) est considéré comme étant présent sur l'image. La matrice de transformation géométrique est utilisée pour localiser le document uniquement si ce dernier a été reconnu, *i.e.* si 3 mises en correspondance valident la transformation.

Dans les sous-sections suivantes nous allons présenter chacune des étapes de la chaîne de reconnaissance d'objets standards.

3.2.1 Détection et description de points d'intérêt

Cette étape consiste à détecter des points d'intérêt dans des images à comparer. Les points sont caractérisés par la configuration de leur voisinage. Le descripteur associé au

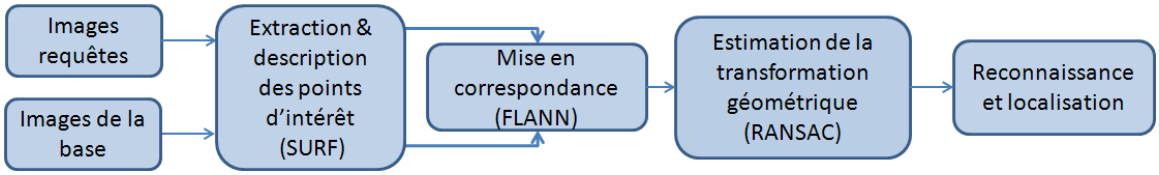


FIGURE 3.4 – Chaîne de traitement pour la mise en correspondance d'images. Les points d'intérêt de l'image requête et des images de la base à analyser sont extraits et caractérisés avec SURF puis mis en correspondance avec FLANN. La transformation géométrique est ensuite estimée à l'aide de RANSAC.

point doit être le plus robuste possible afin d'être reconnu même si la position et l'orientation du document diffèrent entre les deux images à comparer.

La plupart des détecteurs existants sont invariants en translation, comme par exemple le détecteur de Harris [HS88]. Les évolutions telles que Harris-Laplace ou les méthodes basées sur les DoG (Difference of Gaussians) sont quant à elles invariantes en rotation et en changement d'échelle. Les techniques telles que MSER [MCUP04] et LLD [Cao08] ont été mises au point dans l'objectif d'être invariantes aux transformations affines. Cependant, SIFT reste la référence en matière de détection de points d'intérêt. Il combine les DoG qui sont invariants en translation, rotation et mise à l'échelle avec un descripteur basé sur les distributions d'orientations de gradient qui, de plus, est robuste aux changements d'illumination et de points de vues. Depuis, quelques variantes et extensions de SIFT telles que PCA-SIFT [KS04], ASIFT [MY09] et SURF ont été mises au point. Les différences entre ces techniques sont minimes et portent principalement sur la rapidité d'exécution et leur robustesse au changement d'échelle, à la rotation, au flou gaussien, au changement de luminosité et aux transformations affines. Les auteurs de [JG09] présentent une étude comparative de SIFT, PCA-SIFT et SURF.

L'une des principale qualité du descripteur SURF réside dans sa rapidité de calcul. L'étude comparative [JG09] démontre la supériorité du descripteur SURF par rapport à SIFT et PCA-SIFT en terme de temps d'exécution et de sa robustesse aux changements d'illumination. L'algorithme SURF est composé de deux étapes principales. La première consiste à détecter des points d'intérêt sur l'image et la seconde consiste à décrire ces points d'intérêt à l'aide d'un vecteur de 64 caractéristiques.

Détection des points

Afin de gagner en temps de calcul, l'image à analyser est transformée en image intégrale [VJ01]. Les images intégrales permettent de faire beaucoup plus rapidement les calculs de convolution et d'aires rectangulaires. Soit i, notre image de départ, $i(x, y)$ représente la valeur d'un pixel de l'image aux coordonnées x et y. L'image intégrale, notée $ii(x, y)$, est une image de même taille que l'image d'origine, calculée à partir de celle-ci. Chaque pixel de l'image intégrale contient la somme des pixels situés au dessus et à gauche de ce pixel dans l'image initiale. La valeur d'un pixel de l'image intégrale ii est définie à partir de l'image i par l'équation suivante :

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

Une fois l'image intégrale calculée, la somme des intensités S_i d'un rectangle $ABCD$ de l'image d'origine peut être évaluée en accédant seulement à la valeur des quatre sommets

alors qu'il faudrait accéder à toutes les valeurs des pixels du rectangle sans image intégrale :

$$S_i = \sum_{\substack{x_A \leq x' \leq x_B \\ y_B \leq y' \leq y_C}} i(x', y') = ii(C) - ii(B) - ii(D) + ii(A)$$

Les zones de fort changement d'intensité des pixels sont recherchées dans l'image. La matrice hessienne, basée sur le calcul des dérivées partielles d'ordre deux, est utilisée pour cela. Pour une fonction à deux variables $f(x, y)$, la matrice hessienne est définie comme suit :

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

Si le déterminant de la matrice hessienne est positif, alors les valeurs propres de la matrice sont toutes les deux positives ou toutes les deux négatives, ce qui signifie qu'un extremum est présent. Les points d'intérêt seront donc localisés là où le déterminant de la matrice hessienne est maximal. Concrètement, les dérivées partielles du signal sont calculées par un produit de convolution avec des gaussiennes. Afin de gagner en rapidité, ces gaussiennes sont approximées par une fonction à paliers appelée *box filter* (voir figure 3.5).

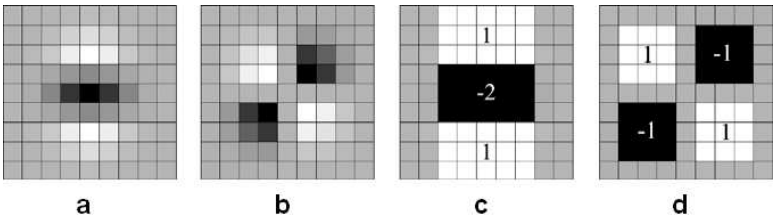


Figure 3.5 – Les gaussiennes et leur approximations par "box filter". L'image c) est l'approximation de la gaussienne affichée dans l'image a) et l'image d) est l'approximation de la gaussienne affichée dans l'image b). Les illustrations sont issues de [BETVG08].

Une représentation à des niveaux d'échelles plus bas est obtenue en augmentant la taille des filtres gaussiens. Au final, les points d'intérêt dont le déterminant de la matrice hessienne est positif et qui sont maximum locaux dans un voisinage 3 x 3 x 3 (abscisse x ordonnée x échelle) sont conservés.

Description des points

Une fois les points d'intérêt extraits, la seconde étape de SURF consiste à calculer le descripteur correspondant. Le descripteur SURF décrit l'intensité des pixels dans un voisinage autour de chaque point d'intérêt. La réponse en x et en y des ondelettes de Haar est calculée dans un voisinage de $6s$ où s est l'échelle à laquelle le point d'intérêt a été trouvé. A partir de ces valeurs, l'orientation dominante de chaque point d'intérêt est calculée en faisant glisser une fenêtre d'orientation.

Pour calculer le descripteur, un carré de taille $20s$ orienté selon l'orientation dominante est extrait. Cette zone est subdivisée en 4×4 carrés. Pour chacune de ces sous-régions, les ondelettes de Haar sont calculées sur 5×5 points. Soit d_x et d_y la réponse à l'ondelette de Haar, 4 valeurs sont calculées pour chacune des sous-régions : $\sum d_x$, $\sum d_y$, $\sum |d_x|$ et $\sum |d_y|$. Au final, chacun des points extraits à l'étape précédente est décrit par un vecteur composé de $4 \times 4 \times 4$ valeurs soit 64 dimensions.

3.2.2 Mise en correspondance des points

Dans cette étape, les points d'intérêt d'une image sont mis en correspondance avec les points d'intérêt d'une autre image afin d'estimer le degré de similitude entre ces deux images. Chaque point d'intérêt de l'image modèle est associé aux deux points d'intérêt de l'image requête qui lui sont les plus proches en terme de distance euclidienne dans l'espace des 64 dimensions (le second plus proche sera utilisé pour l'étape de filtrage que nous détaillons un peu plus loin). Pour trouver les 2-PPV (deux plus proches voisins) la distance euclidienne entre les descripteurs à 64 dimensions est utilisée.

La recherche de PPV peut s'avérer longue si elle est faite de manière exhaustive. Les arbres KD permettent de structurer l'espace de recherche afin d'accélérer la comparaison d'un élément avec les autres. La figure 3.6 illustre la construction et la recherche de plus proches voisins avec un arbre KD. Plus le nombre de dimensions est grand, plus les performances de recherche d'un arbre KD se rapprochent de celles d'une recherche linéaire. Un arbre KD n'est pas performant lorsque le nombre de dimensions est élevé. Cette technique ne serait donc pas optimale pour SURF puisque les descripteurs de ce dernier comportent 64 dimensions. L'algorithme FLANN décrit dans [ML09] utilise le principe d'arbres KD aléatoires proposé récemment par [SAH08]. Dans ce cas, n arbres KD sont utilisés en parallèle, chacun utilisant uniquement 5 dimensions tirées aléatoirement.

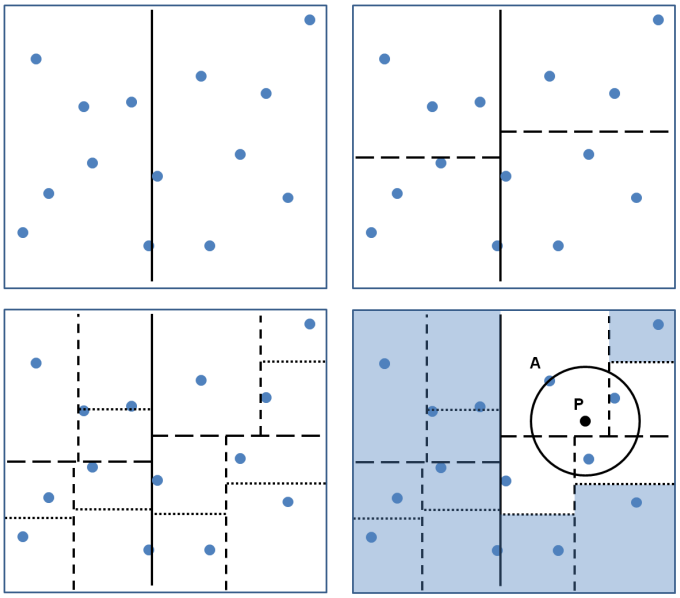


FIGURE 3.6 – Fonctionnement de la recherche de plus proches voisins par arbre KD. L'espace est récursivement découpé en hyperplans, à hauteur de la médiane selon une dimension (x) puis selon la dimension suivante (y), etc. L'arbre KD est l'arbre binaire correspondant au découpage. La première partie de la recherche consiste à déterminer dans quelle case de l'arbre est le point P dont le PPV est cherché. Le point A présent dans cette case est marqué comme étant l'actuel PPV. Toutes les cases qui sont plus loin que l'hypersphère de centre P et de rayon PA sont éliminées. Il ne reste plus qu'à trouver si un point dans les cases restantes est plus près de P que A.

Après la mise en correspondance, chacun des points du modèle est associé aux deux points les plus ressemblants dans l'image requête. On souhaite éliminer le plus de fausses mises en correspondance possibles afin de faciliter l'estimation de la transformation. Le premier filtre consiste à supprimer les mises en correspondance dont les 2-ppv dans l'espace

de dimension 64 sont trop proches l'un de l'autre. C'est le filtrage par unicité, il permet d'éliminer les mises en correspondance ambiguës. Ensuite les mises en correspondance sont filtrées en fonction de l'échelle et l'orientation. Le rapport de l'échelle et la différence d'orientation des mises en correspondance sont calculés. L'espace des angles est découpé par tranche de 20 degrés et l'espace des échelles par facteurs de $1,5$. Les correspondances dont l'échelle et la rotation ne correspondent pas au vote de l'échelle et l'angle majoritaire, sont éliminées. La figure 3.8 met en évidence l'intérêt de ces filtrages qui permettent d'éliminer facilement un grand nombre de mauvaises mises en correspondance (passage de l'image a) à l'image b)).

3.2.3 Estimation de la transformation géométrique

Dans notre cas, les documents sont numérisés à plat, il n'y a pas de distorsion ni de rotation autre que celle dans le plan. Le modèle recherché comporte quatre inconnues : l'angle θ de rotation dans le plan, la translation T_x selon l'axe x, la translation T_y selon l'axe y et la mise à l'échelle α (uniforme en x et y). La matrice de transformation recherchée M_t est de la forme suivante :

$$M_t.\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha.cos(\theta) & -sin(\theta) & T_x \\ sin(\theta) & \alpha.cos(\theta) & T_y \\ 0 & 0 & 1 \end{bmatrix}.\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

La difficulté de cette partie est de trouver une transformation géométrique parmi les différentes mises en correspondance alors qu'il reste des mises en correspondance erronées et que le document recherché peut ne pas être présent sur l'image.

Pour ces deux raisons, la méthode des moindres carrés ne peut pas être utilisée car elle serait perturbée par les mises en correspondance erronées (valeurs aberrantes) et trouverait une transformation même quand le modèle n'est pas présent. L'estimation de la transformation géométrique M_t doit donc être faite à l'aide d'un algorithme capable d'estimer un modèle sans prendre en compte les valeurs aberrantes (aussi appelées outliers). C'est la raison pour laquelle l'algorithme RANSAC est très souvent utilisé pour l'estimation de transformation géométrique [SNB11].

L'algorithme RANSAC s'articule en deux étapes principales. 1) Le sous ensemble le plus petit possible permettant d'estimer le modèle géométrique est sélectionné aléatoirement. 2) On cherche d'autres éléments validant le modèle. Ces éléments sont appelés "*inliers*". S'il n'y a pas suffisamment d'inliers, l'algorithme retourne en 1), sinon le modèle est validé. Si aucun modèle n'est validé après MAX_ITER essais, l'algorithme s'arrête. MAX_ITER est fixé empiriquement à 200.
En ce qui concerne les paramètres, si MAX_ITER est trop bas le meilleur ensemble de mises en correspondance peut être manqué. Mais si sa valeur est trop grande, le temps d'exécution sera allongé. Un autre seuil est fixé dans RANSAC : $DIST\_VALID$. C'est la distance maximale entre la position attendue de la mise en correspondance donnée par la transformation et la position réelle de la mise en correspondance. Cette valeur représente la tolérance du processus de mise en correspondance. Puisque la transformation géométrique est supposée être linéaire (on considère qu'il peut y avoir uniquement une translation, une rotation dans le plan et un changement d'échelle) la valeur de ce seuil ne doit pas être trop haute. Cependant, nous travaillons avec des images qui sont relativement larges, les documents A4 documents numérisés à 200 dpi font environ 1654 pixels sur 2339 pixels. Nous avons expérimentalement déterminé $DIST\_VALID$ à 10 pixels. Cette valeur permet d'être tolérant à de légères déformations. Si cette valeur est trop élevée, cela risque

d'augmenter le nombre de mauvaises mises en correspondance. La figure 3.7 illustre un exemple d'utilisation de RANSAC.

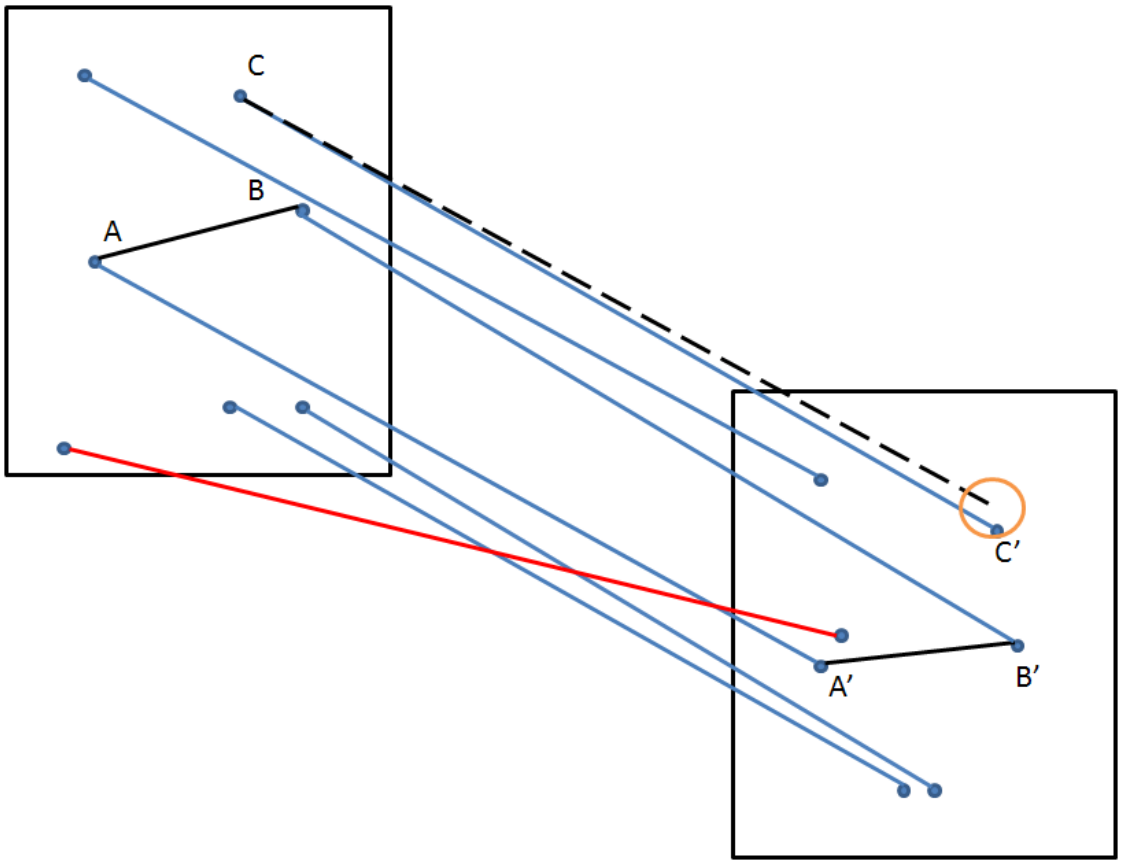


FIGURE 3.7 – Utilisation de RANSAC. La matrice de transformation M_t est celle qui transforme le vecteur $\overrightarrow{AB}$ en $\overrightarrow{A'B'}$. Si le point C' est à une distance inférieur à $DIST\_VALID$ de la transformation de C pat M_t, alors le point est validé et fait parti des "inliers". Cette distance est symbolisée par le cercle orange sur la figure. Sinon il n'est pas validé et fait parti des "outliers" (symbolisé par un trait rouge).

La figure 3.8 montre l'impact de RANSAC sur les mises en correspondance (passage de l'image b) à l'image c)).

Enfin, si au moins t mises en correspondances sont validées par le modèle géométrique, le document est considéré comme étant présent dans l'image. Nous avons choisi $t = 3$, selon les recommandations de Lowe [Low04].

3.2.4 Performance de la méthode standard appliquée à des images de documents semi-structurés

La technique standard a été appliquée à une base de données d'images de documents réelle. Nous appellerons cette première base BD_NB. Les images de cette base ont été numérisées et binarisées à la volée par les scanners afin d'être stockées dans des fichiers de format TIF groupe 4. Dans cette base, la plupart des images sont au format A4 et toutes ont une résolution de 200 dpi. Chaque image ne contient qu'un seul document. Cette base contient 2155 images de documents. Parmi ces documents nous nous intéresserons à 7 types de documents en particulier. Ces documents sont les suivants : 483 cartes d'identité

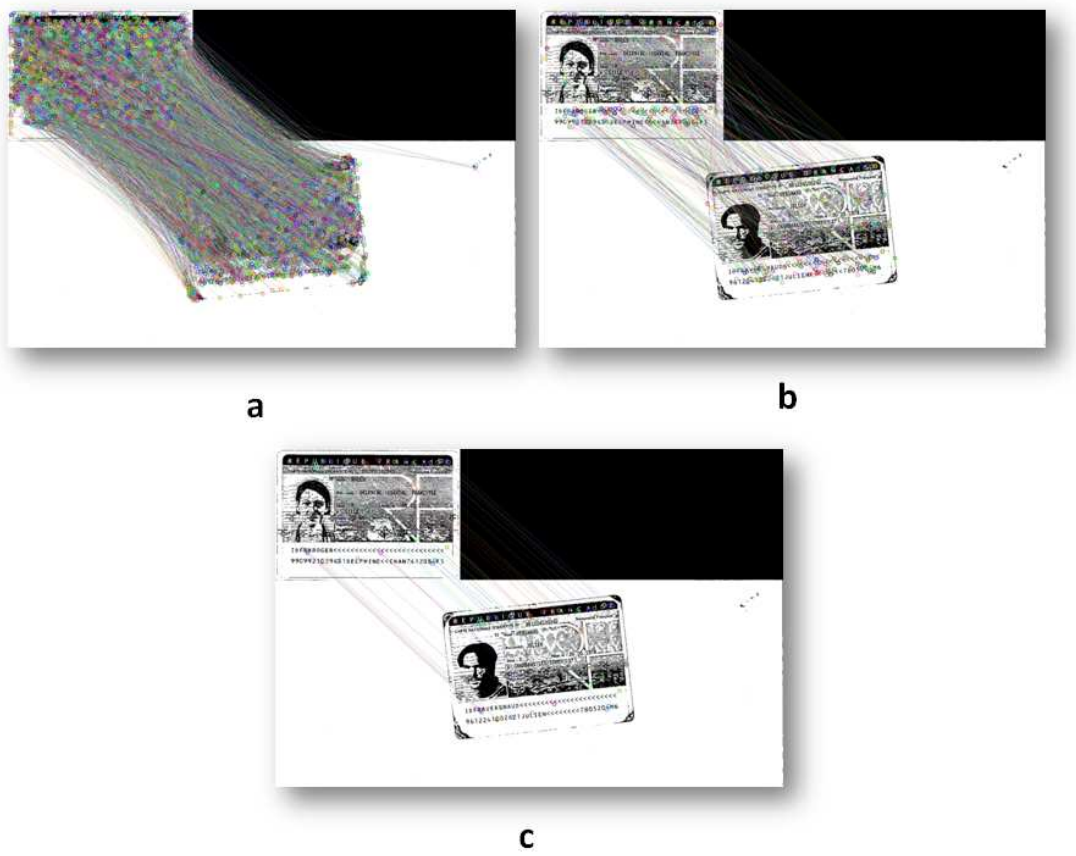


Figure 3.8 – Mise en correspondance de documents. a) Il y a 7687 mises en correspondance. b) Après filtrage, il y a 235 mises en correspondance. c) Après RANSAC il reste plus que 40 mises en correspondance. La transformation trouvée a les paramètres suivant : rotation = 5.665 degrés, mise à l'échelle = 1.003, translation horizontale = 801.7 pixels et translation verticale = 108.3.

françaises, 89 passeports français, 35 tickets de train S.N.C.F., 228 bordereaux d'envoi, 58 tickets de restaurant "Challenger", 58 facture "Orange" et 41 reçu "American Express". Les 1257 images de documents restantes correspondent à 281 types de documents. Le nombre moyen de points d'intérêt est de 7098 pour les cartes d'identité, 16158 pour les passeports et 13391 pour les autres documents. La carte d'identité et le passeport utilisés comme modèles possèdent respectivement 7687 et 10059 points d'intérêt.

Pour une classe donnée, l'objectif est de retrouver l'ensemble des documents dans la base en donnant un exemple d'image. Cette image requête, préalablement choisie par l'utilisateur, est comparée à chaque autre image de la base. Si on arrive à mettre en correspondance une image de la base avec l'image requête alors le document est détecté comme étant présent puis est extrait. Il est donc naturellement préférable de choisir une image requête dont la qualité ne soit pas de trop mauvaise qualité. Les documents utilisés comme requête sont présentés sur la figure 3.9.

Tous les tests de ce chapitre ont été effectués sur un ordinateur Intel Core 2 Duo, 2 GHz. Il est à noter que les tests de recherche d'une classe donnée sont appliqués indépendamment des autres classes. Le tableau 3.1 présente les résultats de la recherche de 7 types d'images dans la base BD_NB. La recherche de chaque classe est indépendante des autres. Ce tableau montre clairement les limites d'une simple transposition d'une mé-

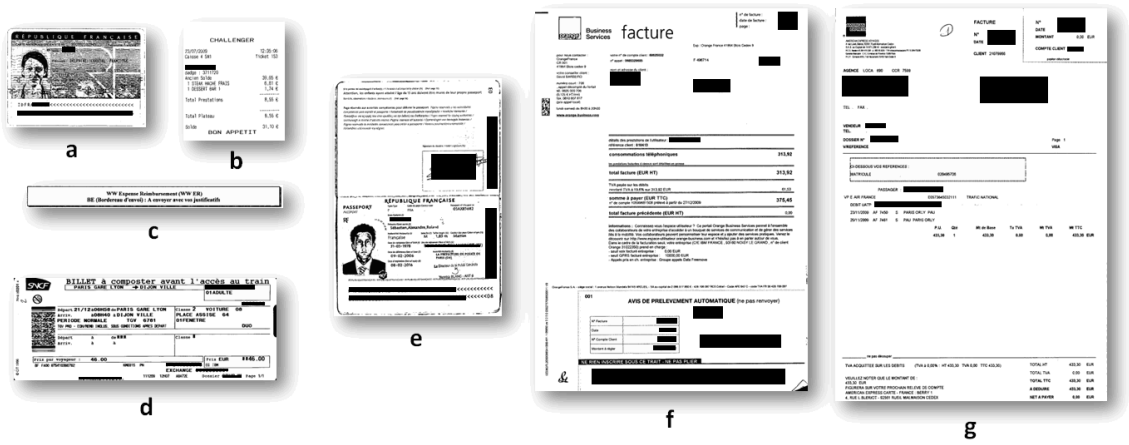


FIGURE 3.9 – Images utilisées comme requête pour la détection des 7 types de documents. a) Carte d'identité "ID", b) ticket de restaurant "Challenger", c) un résumé de note de frais "Bordereau", d) un ticket de train "SNCF", e) un passeport français, f) une facture "Orange" et g) un reçu de paiement "American". Les zones noires ont été ajoutées pour des raisons de confidentialité.

thode de reconnaissance d'objets à des images de documents. Les performances en terme de précision sont extrêmement mauvaises, une moyenne de 17,0% est obtenue. La grande majorité des documents détectés sont des "faux positifs", c'est à dire des documents détectés à tord. Ceci s'explique à cause de la grande quantité de points extraits sur chaque image ce qui engendre une difficulté de la mise en correspondance des points. Les mauvaises mises en correspondance sont tellement nombreuses que le système arrive à former un consensus de 3 mises en correspondance sur la plupart des images. On peut également, grâce à ce tableau, établir un lien entre le nombre de points des images requêtes et la précision de la mise en correspondance.

TABLE 3.1 – Rappel, précision et moyenne du temps d'exécution par image en utilisant la technique standard sur la base d'image de documents BD_NB pour la détection de 7 types de documents. De très nombreuses mauvaises détections sont faites, la précision est extrêmement basse.

Type	Nb images	Nb points	Rappel	Précision	Tps / image
Cartes Id	483	7687	0,996	0,419	2,9 s
Passeport	89	10059	1,00	0,117	3,9 s
SNCF	35	4934	1,00	0,0502	2,4 s
Bordereau	229	751	0,991	0,464	1,3 s
Challenger	58	1573	1,00	0,137	1,6 s
Orange	58	14741	1,00	0,0366	5,0 s
American	41	8714	0,951	0,0266	2,8 s

Les images de documents contiennent de nombreux points d'intérêt principalement à cause du texte contenu dans les documents. De plus les images sont relativement grande : à 200 dpi, un document A4 mesure environ 1654 pixels de largeur et 2339 pixels de hauteur. Cette grande quantité d'information rend l'opération de mise en correspondance des points

complexe car le risque de faire de mauvaises mises en correspondance est plus élevé.

La figure 3.10 présente les 2 cartes d'identité qui n'ont pas été retrouvées en utilisant cette approche. Comme énoncé précédemment, nous avons une contrainte forte visant à cherche une précision le plus proche possible de 100%. De ce fait, nous nous sommes concentrés sur l'amélioration de la précision.

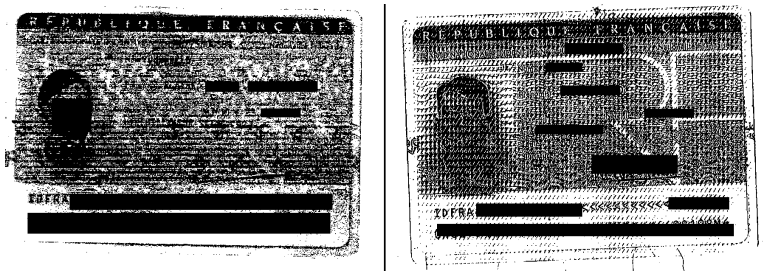


Figure 3.10 – Exemple d'images non détectées. Ces deux images ont un bruit très marqué et ne sont pas détectées.

Comme le disent les auteurs de [US09], les techniques basées sur les points d'intérêt tels que SIFT ou SURF ne sont pas directement transposables aux images de documents car des patterns répétitifs (comme les lettres des mots) apparaissent fréquemment dans les images de document. Concrètement, ces patterns sont des petites portions d'images qui se répètent. Si une image présente de nombreux patterns répétitifs, cela peut engendrer un problème d'auto-similarité, car ces portions d'images qui se répètent risquent d'être confondues les unes avec les autres. De plus, de nombreux points sont extraits sur les images de documents, ce qui engendre un très grand nombre de mauvaises mises en correspondance. Le problème de l'application de la technique classique de reconnaissance d'image vient de la complexité de la mise en correspondance et de la validation d'une transformation parmi un ensemble très vaste. Si aucune précaution n'est prise, de nombreuses images seront alors détectées comme étant identiques à l'image requête.

Nous allons proposer dans la section suivante plusieurs solutions permettant d'améliorer à la fois la précision de la mise en correspondance d'images de documents mais aussi le temps d'exécution.

3.3 Reconnaissance d'objets adaptée aux images de documents

Dans cette section, nous présentons des améliorations pour adapter la technique de reconnaissance d'objets aux images de documents. Dans un premier temps, nous nous sommes consacrés à l'amélioration de la précision de la technique standard. En effet, comme nous l'avons vu dans la partie précédente de nombreuses images sont détectées à tord (faux positifs). Pour remédier à cela, nous proposons une version adaptée de RANSAC. Ensuite, nous proposons une étape supplémentaire permettant de sélectionner des points d'intérêt pertinents de l'image requête et ainsi de diminuer le temps d'exécution mais également de diminuer la probabilité de faire de mauvaises mises en correspondance. Nous montrons également quel est l'impact de t (le nombre minimal de mises en correspondance pour valider une transformation) sur le rappel et la précision du système.

La nouvelle chaîne de traitements est présentée sur la figure 3.11.

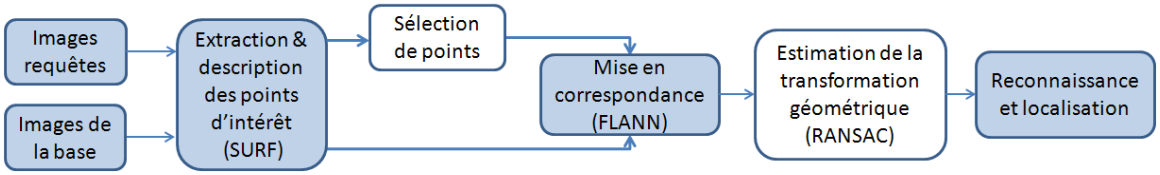


FIGURE 3.11 – Chaîne de traitement adaptée pour la mise en correspondance d'images de documents. Toutes les étapes sont les mêmes que la méthode standard sauf ceux sur fond blanc. Une étape supplémentaire permet de sélectionner certains des points d'intérêt de l'image requête. La transformation géométrique est estimée à l'aide d'une version adaptée de RANSAC qui permet d'augmenter les contraintes géométriques.

3.3.1 Adaptation de RANSAC pour une reconnaissance précise des images de documents

Afin d'augmenter la précision de RANSAC nous avons ajouté des contraintes géométriques au sein de l'algorithme. Notre version adaptée de RANSAC est détaillée dans l'algorithme 1. Les entrées de l'algorithme sont S_Q et S_D qui représentent l'ensemble des points d'intérêt de l'image requête S_Q mis en correspondances avec les points de l'image de la base à analyser S_D. En sortie de l'algorithme on obtient I, un ensemble d'inliers, c'est à dire un ensemble de mises en correspondance validant une même transformation géométrique M.

Dans cette version adaptée apparaissent deux nouveaux seuils : MIN_NORME et $MIN\_DVALIDE$. On utilise MIN_NORME pour ne pas calculer la matrice de transformation a partir de points qui sont trop proches car cela risquerait d'induire un calcul approximatif (par analogie, il est préférable d'estimer le coefficient directeur d'une droite en prenant deux points éloignés). Le second seuil $MIN\_DVALIDE$ est utilisé pour ne pas valider des inliers qui sont trop proches. Ces deux seuils permettent d'éviter des mauvaises validation comme celles de la figure 3.13. MIN_NORME et $MIN\_DVALIDE$ ont été fixés à 5 pixels. Ces seuils ont été validés par des tests effectués sur des documents de différentes tailles. La figure 3.12 illustre l'utilisation des deux nouveaux seuils.

Les seuils ne sont pas très sensibles, mais il faut les fixer à une valeur minimale afin d'éviter des problèmes tels que ceux de la figure 3.13.

Enfin, pour augmenter encore plus la contrainte géométrique nous avons choisi d'augmenter le nombre minimum d'inlier t pour valider la transformation. Lowe le fixait à 3 dans les images naturelles nous avons choisi de le fixer à 8 pour nos images de document. On rappel qu'après l'application de RANSAC, la transformation n'est validée que si : $Card(I) >= t$. RANSAC adapté permet de grandement augmenter la précision grâce aux contraintes géométriques supplémentaires et à légèrement diminuer le temps d'exécution.

3.3.2 Sélection automatique de points d'intérêt

Nous proposons une autre amélioration visant à diminuer le temps d'exécution et également d'augmenter la précision. L'objectif est de limiter le nombre de points d'intérêt utilisés pour caractériser les images requêtes. En effet, grâce aux tests effectués dans la sous-section 3.2.4, nous avons vu que l'augmentation du nombre de points d'intérêt décrivant les requêtes a tendance à augmenter le temps d'exécution et à diminuer la précision. Ceci s'explique par le fait que l'augmentation du nombre de points d'intérêt induit une augmentation des possibles mises en correspondance et donc l'augmentation du risque de faire de mauvaises mises en correspondance.

Algorithm 1 Adapted RANSAC

INPUT S_Q, S_D
OUTPUT I, M
$S_I \leftarrow \emptyset, iter \leftarrow 0$
while $iter < MAX\_ITER$ **do**
  Soient P_{Q1} & P_{Q2} 2 points aléatoirement choisis dans S_Q.
  Soit P_{D1} & P_{D2} la mise en correspondance de P_{Q1} & P_{Q2} dans S_D
  if $(||\overrightarrow{P_{Q1}P_{Q2}}|| < MIN\_NORM)||(||\overrightarrow{P_{D1}P_{D2}}|| < MIN\_NORM)$ **then**
    $iter \leftarrow iter + 1$
    continue
  Soit M_t la matrice de transformation courante
  Soit I_t l'ensemble d'inliers courant
  $M_t \leftarrow transfo(\overrightarrow{P_{Q1}P_{Q2}}, \overrightarrow{P_{D1}P_{D2}})$
  $I_t \leftarrow \emptyset$
  for each $P_{Qi} \in S_Q \backslash \{P_{Q1}, P_{Q2}\}$ **do**
    Soit P_{Di} la mise en correspondance de P_{Qi} dans S_D
    $I_t \leftarrow$ testAjoutInlier$(I_t, P_{Qj}, P_{Qi}, P_{Dj}, P_{Di})$
  if $Card(I_t) > Card(I)$ **then**
    $I \leftarrow I_t, M \leftarrow M_t$
  $iter \leftarrow iter + 1$
return I, M
function testAjoutInlier$(I_t, P_{Qj}, P_{Qi}, P_{Dj}, P_{Di})$
  for each $I_j \in I_t$ **do**
    if $(||\overrightarrow{P_{Qj}P_{Qi}}|| < MIN\_DVALID)||(||\overrightarrow{P_{Dj}P_{Di}}|| < MIN\_DVALID)$ **then**
      return I_t
  $P'_{Qi} \leftarrow M_t[P_{Qi}]$
  if $||\overrightarrow{P_{Di}P'_{Qi}}|| < MAX\_DIST$ **then**
    $I_t \leftarrow I_t \cup \overrightarrow{P_{Di}P'_{Qi}}$
  return I_t

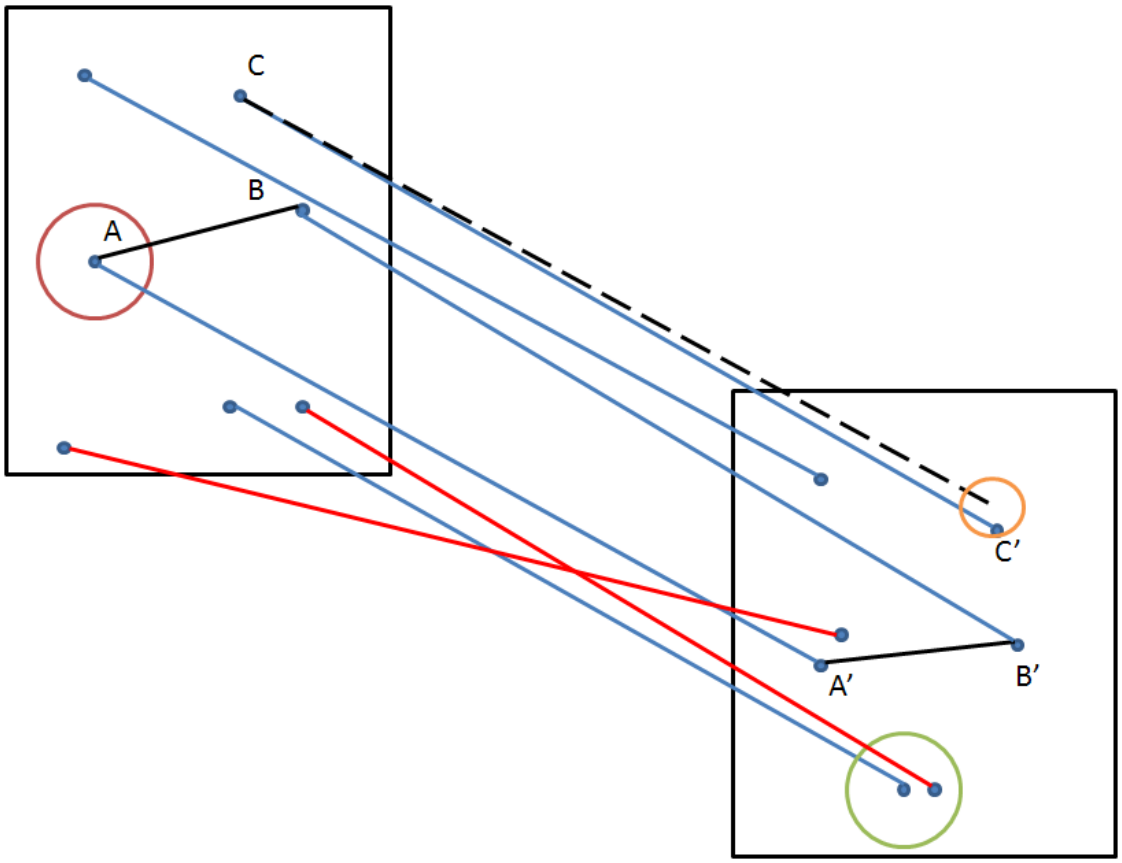


FIGURE 3.12 – Utilisation de RANSAC adapté. Le fonctionnement global est le même que celui de la figure 3.7. La norme minimale pour calculer la matrice de transformation MIN_NORME est symbolisée par le cercle rouge. La distance minimale pour pouvoir valider un nouveau point $MIN\_DVALIDE$ est symbolisée par le cercle vert.

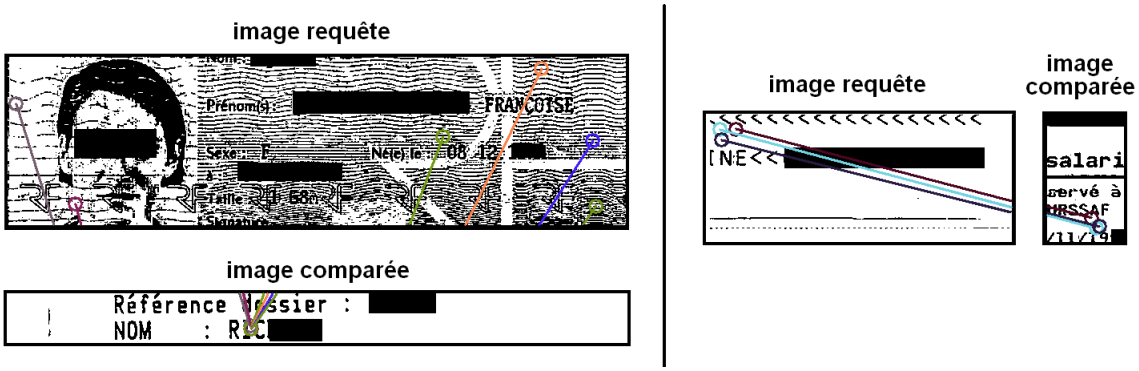


FIGURE 3.13 – Exemple de mauvaises détections évitées grâce à la version de RANSAC adaptée. a) 6 points éloignés sont mis en correspondance avec deux points proche. b) 3 points proches sont mis en correspondance avec 3 autres points proches. Dans les deux cas il y a trop de mises en correspondance faites dans un espace trop réduit de l'image. Ransac adapté interdit de telles mises en correspondance et réduit ainsi le nombre de mauvaises mises en correspondance.

Dans une première version de ces travaux [AJD12], nous avons proposé une solution consistant à tracer un rectangle blanc sur les régions d'images requêtes afin de n'extraire

les points d'intérêt que sur les zones les plus pertinentes de l'image requête. L'inconvénient est qu'il peut s'avérer complexe de définir manuellement de telles zones et particulièrement pour un opérateur qui n'est pas spécialiste en analyse d'images. Nous proposons ici une évolution permettant d'éviter cette sélection.

La solution que nous proposons consiste à fournir 5 exemples d'images au lieu d'une. Un ensemble de points d'intérêt de l'image requête utilisés pour la mise en correspondance en comparant l'image requête à 4 autres images du même type est alors sélectionné. La première image requête est mise en correspondance avec chacune des 4 autres images en utilisant RANSAC adapté. Si une transformation est validée, l'ensemble des points d'intérêts mis en correspondance sont alors sélectionnés. En itérant ainsi sur les 4 nouvelles images, on conserve finalement l'union des points d'intérêt utilisés pour l'ensemble des mises en correspondance faites. Ceci permet donc de garder uniquement les points d'intérêt utiles à la détection de documents du même type.

La figure 3.14 montre l'impact de la sélection de points d'intérêt sur 5 modèles. La sélection permet en moyenne de diviser le nombre de points d'intérêt par image requête d'un facteur 4,11.

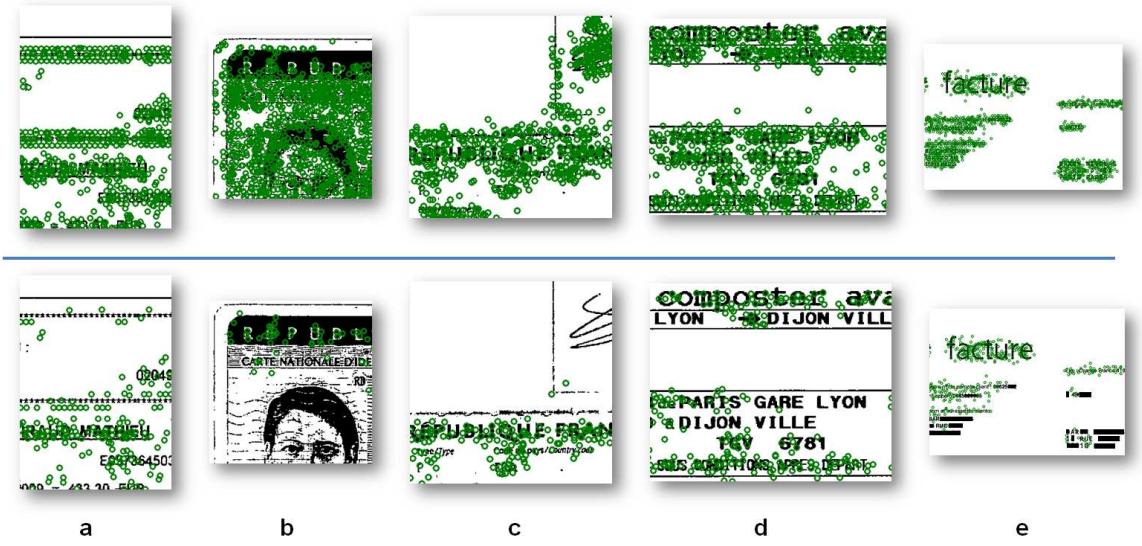


Figure 3.14 – Sélection de points d'intérêt. Les cercles verts dessinés sur chaque image correspondent à des points d'intérêt. Sur la ligne du dessus figure les images requêtes avant la sélection de points. De nombreux points sont présents. En dessous, sont présenté les mêmes images requêtes après sélection des points d'intérêt. Le nombre de points a largement diminué. On observe que les points d'intérêt situés sur des parties variables de l'image ne sont pas sélectionnés. On peut citer par exemple : b) la photographie de la carte d'identité, c) la signature du passeport, d) le nom des villes sur un ticket de train et d) l'adresse de destination sur une facture. Les nombres de points d'intérêt avant et après sélection sont détaillés dans le tableau 3.2.

3.4 Tests sur bases réelles avec la méthode adaptée

3.4.1 Application à une base de documents noir et blanc

Nous avons testé notre méthode intégrant les modifications sur RANSAC et la sélection optimisée de points d'intérêt sur l'image requête sur la même base d'images que dans la partie précédente (BD_NB). Le tableau 3.2 présente les résultats obtenus. On remarque qu'il y a une très légère diminution du rappel mais en contrepartie, une forte hausse de la précision est réalisée. Dans le test de détection standard, le rappel était très haut car un très grand nombre d'images étaient mal détectées. En moyenne, la précision est multipliée par un facteur 5,7 et le temps d'exécution est divisé par 2,0. Les gains maximaux sont obtenus pour les passeports et les factures "Orange" qui sont les documents ayant le plus de points d'intérêts et qui étaient les plus mal détectés par la méthode d'origine.

TABLE 3.2 – Performances de détection de 7 types de documents en utilisant la technique avec RANSAC adapté sur la base BD_NB. On remarque une large diminution du nombre de mauvaises détections.

Type	Nb images	Nb points	Nb pts sélec	Rappel	Précision	Tps / image
Cartes Id	483	7687	306	0,919	1,00	1,01 s
Passeport	89	10059	1284	0,978	1,00	1,14 s
SNCF	35	4934	1846	0,971	1,00	1,58 s
Bordereau	229	751	281	0,900	1,00	1,06 s
Challenger	58	1573	626	1,00	0,879	1,15 s
Orange	58	14741	4444	1,00	0,951	2,30 s
American	41	8714	3017	0,853	0,972	1,49 s
Moyenne				0,928	0,989	

La précision est très bonne mais n'est pas totalement parfaite pour certaines classes d'images. Des mauvaises détections apparaissent parce que des documents de classes différentes présentent des similarités locales. La figure 3.15 montre des exemples de mauvaises détections. La classe "Challenger" contient des tickets de restaurants, les 6 mauvaises détections se produisent avec d'autres tickets de restaurant qui ont une mise en page et quelques mots clés similaires. Les factures "Orange" sont confondues avec 3 tickets "Air France" parce qu'un mot écrit avec une grande police est présent sur les images. Un autre exemple de mauvaises détections est observé avec le logo des reçus "American" qui se retrouve être identique avec celui utilisé sur des factures "American Express".

3.4.2 Retour sur l'utilisation de cette méthode en production

Dans un cas d'application en production de cette méthode, tout ce qui touche à la définition des paramètres doit être mis en perspective des objectifs souhaités. L'amélioration du rappel ou de la précision est possible en diminuant ou augmentant certains des seuils. Nous proposons dans cette sous-section de discuter de la manipulation des paramètres lors de tests réalisés en production.

Si l'utilisateur souhaite obtenir une précision encore plus forte que celle obtenue dans le tableau 3.2, les contraintes géométriques globales doivent être renforcées. Au lieu de prendre seulement $t = 8$ inliers minimum pour la validation de la transformation, t peut être augmenté. L'inconvénient est que le rappel chutera si t est trop élevé. De plus ceci

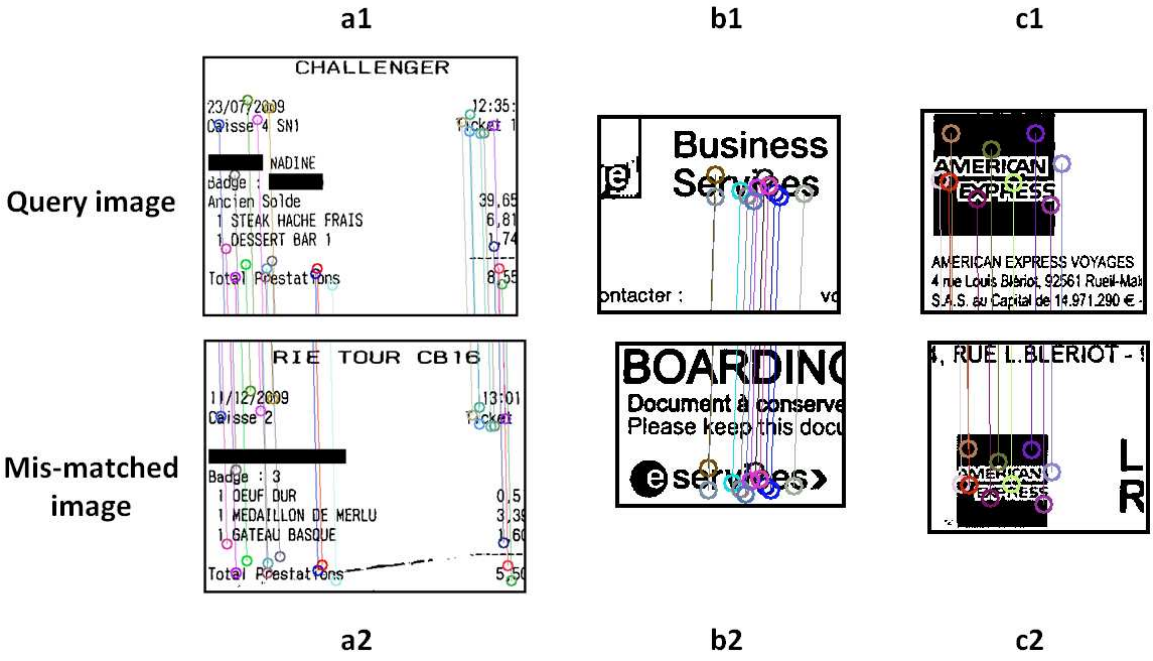


Figure 3.15 – Exemples de mauvaises détections. Le document requête"Challenger" a1) est mis en correspondance avec l'autre ticket de restaurant a2). La facture "Orange" b1) est mis en correspondance avec le ticket "Air France" b2). Le reçu "American" est mise en correspondance avec une facture "American express" c2).

n'empêchera pas la mise en correspondance locale d'inliers. Pour faire cela, nous conseillons d'augmenter le seuil $MIN\_DVALIDE$. Dans ce cas, la distance minimale pour valider un nouveaux inlier sera augmentée. Mais ici encore, si MIN_DVALID est trop augmenté, le rappel risque de diminuer, notamment parce que les documents bruités ne seront plus détectés. La figure 3.16 montre des exemples de bonnes détections pour chacun des 7 types de documents.

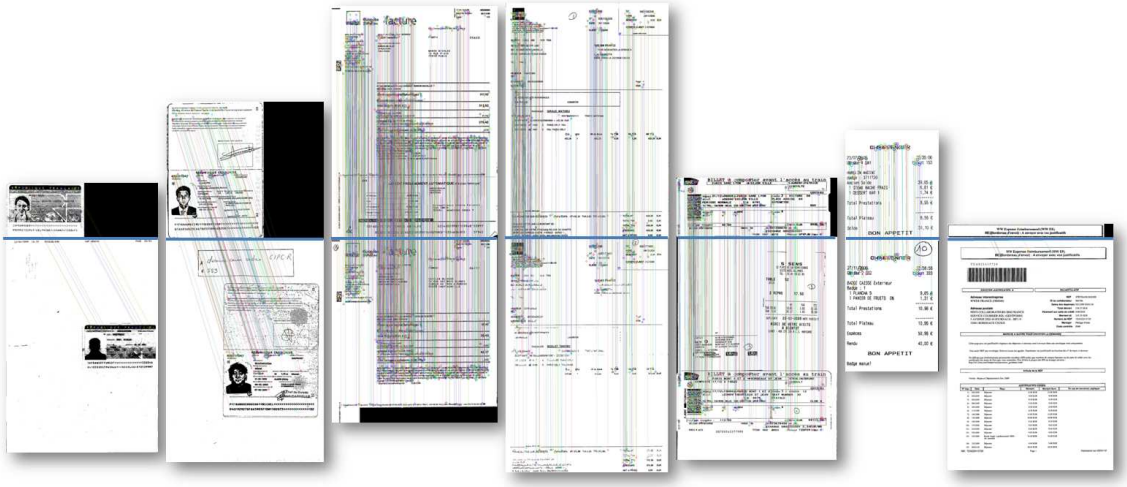


Figure 3.16 – Exemples de bonnes détections pour chacune des 7 images requêtes.

Afin de montrer l'impact du seuil t (si plus de t inliers sont trouvés, la transformation

est validée), le rappel et la précision ont été calculés pour trois types d'images en faisant varier t. On observe sur la figure 3.17, les courbes correspondantes aux résultats. On en déduit qu'une valeur de t faible induira une précision faible et un rappel élevé tandis qu'une valeur de t élevée impliquera un rappel faible et une précision élevée. Cependant le rappel chute beaucoup moins rapidement que la précision augmente. Nous avons choisi $t = 8$ afin d'obtenir le meilleur compromis. Donc, si l'utilisateur souhaite obtenir les meilleurs performances possibles pour chacune des classes de documents, l'idéal est de fixer une valeur de t différente pour chaque classe.

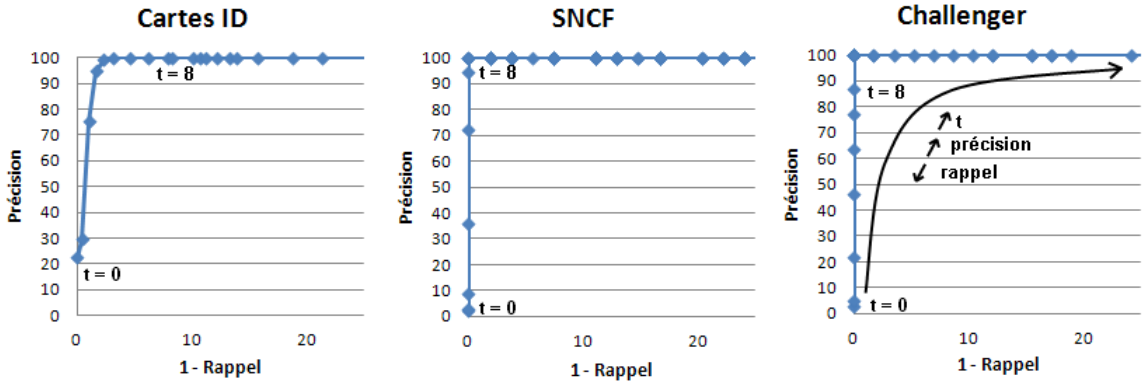


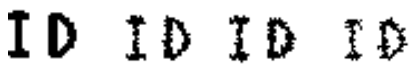
FIGURE 3.17 – Impact de t sur le rappel et la précision. La précision augmente avec t tandis que le rappel diminue. Une valeur trop faible ou trop élevé de t diminue fortement respectivement le rappel et la précision, ce qui justifie notre choix de $t = 8$. Attention à la lecture du graphique : plusieurs points sont superposés, l'abscisse ne représente pas le rappel mais (1-rappel).

3.4.3 Application sur une base semi-synthétique bruitée

Des tests sur la robustesse de SURF au changement d'échelle, à la rotation, au flou, au changement d'illumination et aux transformations affines ont déjà été réalisés par Juan & Gwun [JG09]. La spécificité de nos travaux étant l'utilisation de SURF sur les images de documents, nous avons décidé de tester la robustesse de ces points d'intérêt sur des images contenant des déformations situées aux niveaux des caractères. Ce bruit propre aux images de document provient du processus de numérisation. Ce bruit peut être généré synthétiquement grâce au modèle de Kanungo [KHP93]. Ce bruit se produit lorsque la résolutions de numérisation est basse ou que la qualité des documents est mauvaise. Dans ces cas, l'OCR a généralement de mauvais résultats.

La même base de données BD_NB est utilisée, mais toutes les images ont été dégradées avec un bruit de Kanungo aléatoire variant d'un facteur 0 (pas de dégradation) à un facteur 9 (fortes dégradations). La figure 3.18 illustre quelques exemples de caractères bruités en fonction du facteur de bruit de Kanungo.

Le tableau 3.3 montre que même avec des documents bruités, notre système reste performant. On peut noter que la précision augmente pour les 2 types de documents "Challenger" et "Orange". L'explication est que certaines mauvaises détections étaient faites de "justesse" c'est-à-dire avec le minimum d'inliers nécessaires ($t = 8$). A cause du bruit, ces documents ne sont plus détectés par le système ce qui fait que la précision augmente dans ces cas particuliers. La précision du système reste en moyenne la même et le rappel diminue très légèrement car quelques documents sont trouvés en moins. Par le biais de



a b c d

Figure 3.18 – Exemples de bruits de Kanungo selon différents facteurs. a) Facteur 0 : pas de dégradation. b) Facteur 3 : peu de dégradations. c) Facteur 6 : fortes dégradations. d) Facteur 9 : très fortes dégradations.

la robustesse de SURF à la dégradation des caractères, notre méthode se trouve donc également robuste à ce bruit couramment rencontré.

Table 3.3 – Performances de détection de 7 types de document en utilisant RANSAC adapté sur la base BD_NB bruité par le modèle de déformation de Kanungo. On remarque une très légère diminution du nombre de mauvaises détections. Les informations entre parenthèses représentent l'évolution des performances par rapport au même test fait sans bruit.

Type	Nb images	Nb points	Nb pt. sélec	Rappel	Précision
Cartes Id	483	7687	306	0,830 (-0,089)	1,00
Passeport	89	10059	1284	0,978	1,00
SNCF	35	4934	1846	0,971	1,00
Bordereau	229	751	281	0,721 (-0,179)	1,00
Challenger	58	1573	626	1,00	0,983 (+0,104)
Orange	58	14741	4444	0,983 (-0,017)	1 (+0,049)
American	41	8714	3017	0,659 (-0,09)	0,964 (-0,003)

Afin de montrer plus précisément l'impact du bruit de Kanungo sur la détection, nous avons appliqué du bruit de Kanungo sur une image requête en faisant varier la force du bruit de Kanungo de 0 à 9. Nous avons ensuite mis en correspondance chacune de ces 10 images avec l'image requête originale. Lorsque l'image est mise en correspondance avec elle même, le nombre d'inliers est alors de 100%, chaque point d'intérêt est mis en correspondance avec lui-même. Quand le bruit augmente, le nombre d'inliers diminue. Nous avons effectué ce test sur 3 types de documents : les cartes d'identité, les passeports et le tickets Challenger. On rappelle que le nombre de points d'intérêt de chaque image est respectivement : 306, 1284 et 626. La figure 3.19 présente les résultats obtenus. Même avec un bruit très élevé, les documents sont retrouvés.

3.4.4 Recherche d'images en niveaux de gris

La plupart des images de documents sont numérisées puis binarisées à la volé pour des raisons de coût de stockage. Cependant, quelques prestations sont numérisées en niveaux de gris ou en couleur. Nous avons donc choisi de tester le fonctionnement de l'algorithme dans le cas où les images n'ont pas été binarisées. L'objectif de cette section est de montrer qu'il est préférable d'utiliser un modèle qui soit de la même nature (noir et blanc ou niveau de gris) que celle des documents à analyser.

Les images de la seconde base ont été numérisées en niveaux de gris, à une résolution de 300 dpi. Les images sont différentes de celles de la première base. Chaque image est composée également d'un seul document. BD_NdG contient 91 cartes d'identité fran-

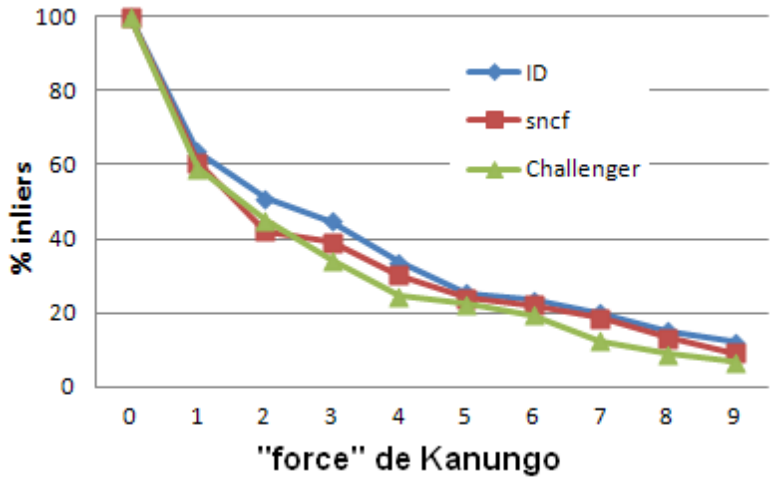


FIGURE 3.19 – Robustesse au bruit de Kanungo en faisant varier la "force" du bruit de 0 à 9 sur trois images requêtes. Le pourcentage d'inliers diminue lorsque le bruit augmente. Cependant les images restent détectables tant que plus de 8 inliers sont présents, soit 2.6 % pour les cartes d'identité, 0.62% pour les passeports et 1.27% pour les tickets Challengers.

çaises, 17 passeports français, et 449 images de documents de types quelconques. Soit un total de 557 images. Le nombre moyen de points d'intérêt est de 9591 pour les cartes d'identité, 18641 pour les passeports et 37543 pour les documents divers. Le modèle de carte d'identité possède 2438 points d'intérêt et celui du passeport en possède 2062. Le tableau 3.4 montre qu'il est important d'avoir une image requête de même nature que les images constituant la base de données. C'est-a-dire que l'image requête doit être en noir et blanc si les images de la base sont en noir et blanc ou en niveau de gris si les images de la base sont en niveau de gris.

TABLE 3.4 – Performances de détection de pièces d'identité, sur la base d'images de documents en niveaux de gris (NdG) BD_NdG avec des images modèles en niveaux de gris. Aucune mauvaise détection n'est faite et au moins 94% des cartes d'identité ou des passeports sont retrouvés. Avec un modèle en noir et blanc (N&B) le nombre de documents détectés baisse.

Modèles	Rappel	Précision
Carte Id (NdG)	0,97	1
Passeport (NdG)	0,94	1
Carte Id (N&B)	0,92	1
Passeport (N&B)	0,59	1

Les tests montrent que la précision de notre technique est très bonne quelque soit la nature du modèle utilisé. Nous conseillons cependant de conserver les images en niveaux de gris pour l'étape de reconnaissance car cela permet d'améliorer la détection des documents.

3.4.5 Gestion de la multi-détection

La plupart des documents sont composés d'une ou plusieurs pages et donc d'une ou plusieurs images. Cependant il arrive également qu'une seule image puisse contenir plu-

sieurs documents. Le fait que plusieurs documents d'un même type soient présents sur une même image peut complexifier la mise en correspondance. En effet puisque le même type de document est présent plusieurs fois sur une même image, la mise en correspondance des points d'intérêt de l'image requête risquent d'être dispersées entre les différents exemplaires du document. Cependant puisque le nombre de points d'intérêt est grand, la mise en correspondance reste possible. La méthodologie pour faire de la multi détection est la suivante. On essaye de mettre en correspondance l'image requête avec les images de la base. Si le document est détecté, il est localisé grâce à la matrice de transformation et la zone correspondante au document est supprimée de l'image (remplacée par un rectangle blanc). Ensuite, l'image est traitée de nouveau jusqu'à qu'aucun document ne soit détecté. La figure 3.20 illustre ce principe.

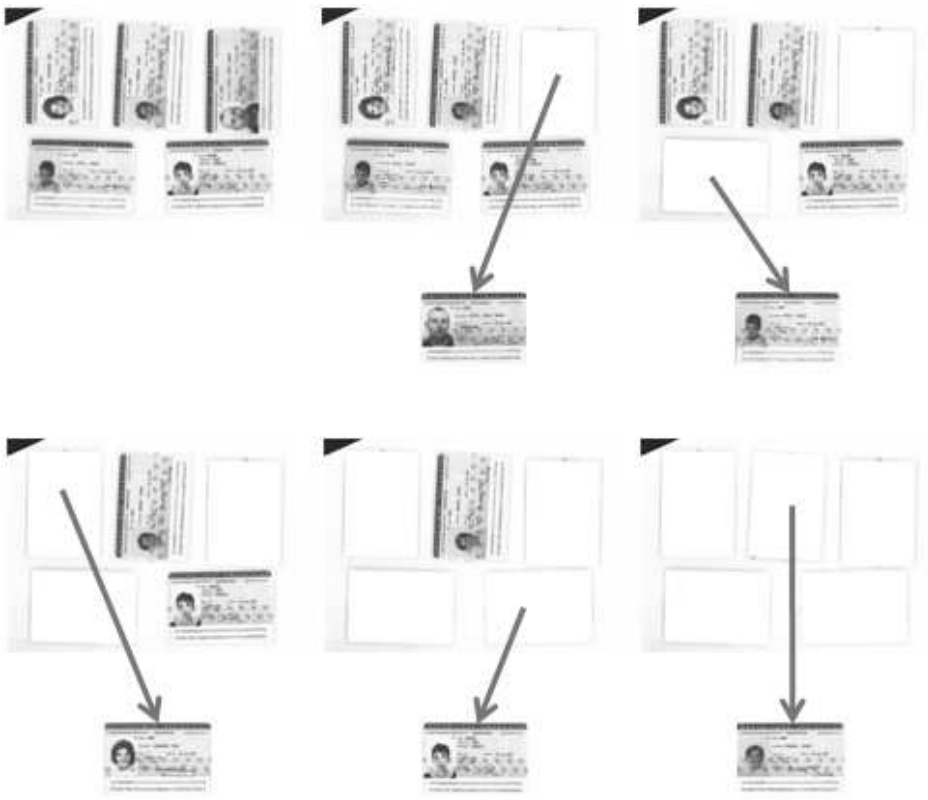


Figure 3.20 – Multi-détection. On cherche un modèle dans chaque image. Si un modèle est trouvé, il est supprimé de l'image puis on cherche à nouveau si un modèle est présent sur l'image. On réitère jusqu'à qu'on ne trouve plus de correspondance.

Nous avons testé ce principe sur une base où chaque image contient plusieurs documents de même type. La nature des images de la base BD_Multi est la même que celle de la base BD_NdG. La base est composée d'un total de 32 images contenant 89 pièces d'identité. Le tableau 3.5 montre que la multi-détection fonctionne très bien grâce à notre méthodologie.

La figure 3.21 montre des exemples de détection de cartes d'identité multiple dans une même image.

La figure 3.22 illustre que notre proposition est tout à fait adaptée à d'autres modèles de documents.

TABLE 3.5 – Les documents multiples dans les images sont trouvés par l'algorithme. Il peut tout de même manquer des documents pour les mêmes raisons que précédemment : le document manquant est un document très bruité.

Modèles	Nb d'im	Nb de doc	Nb de doc trouvés
Carte Id (NdG)	29	83	82
Passeport (NdG)	3	6	6

FIGURE 3.21 – Exemples de multi-détection de cartes d'identité. L'ensemble des cartes sont retrouvées.

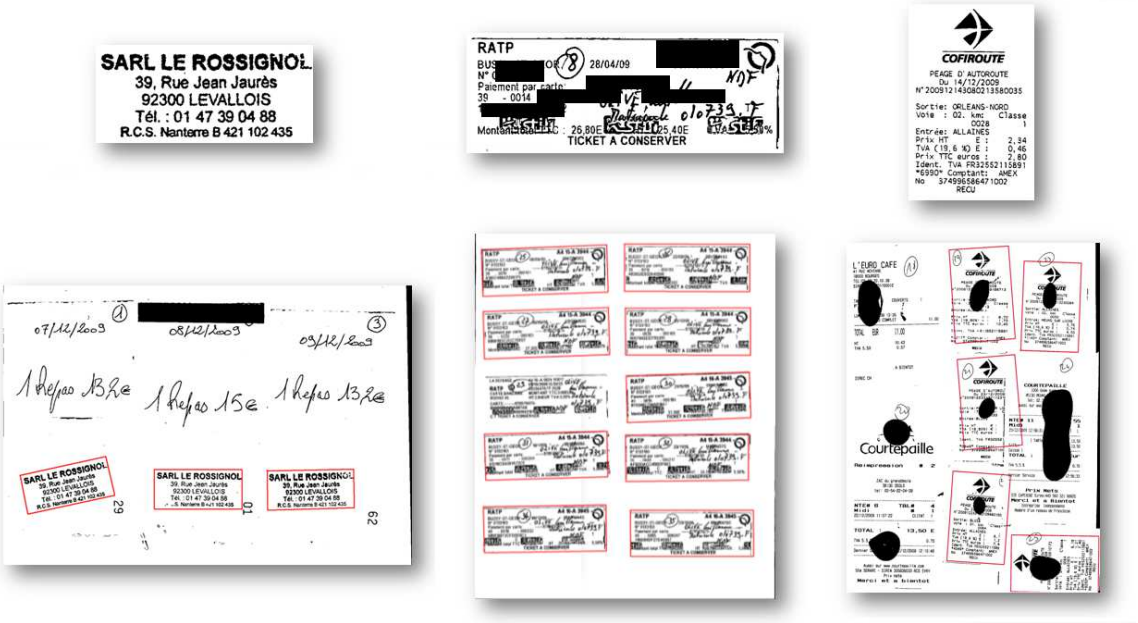


FIGURE 3.22 – Exemples de multi-détection de documents divers. Détection d'un tampon, d'un ticket de métro et d'un ticket d'autoroute. Le ticket de métro non retrouvé n'a pas la même mise en page que les autres tickets. Les tickets d'autoroute sont retrouvés mais 2 tickets ne sont pas très bien positionnés car ils n'ont pas exactement la même mise en page.

3.5 Conclusion

L'identification de sous-parties d'images de documents peut se faire très précisément à l'aide de la méthode exposée dans ce chapitre. Le principal intérêt de cette méthode est de pouvoir garantir une excellente précision pour l'identification d'images de documents. De plus, la méthode peut être étendue très simplement à la recherche d'autres types de sous-images, il suffit de fournir une image requête. Les limites de la méthode sont celles liées au champ d'application, qui se trouve être restreint aux documents semi-structurés. En effet si une classe est composée de documents dont la structure change d'un document à l'autre on ne pourra reconnaitre les documents. L'autre limite est liée au problème de l'auto-similarité. Dans le cas où des images contiennent beaucoup de texte, la répétition de tels motifs peut engendrer de mauvaises mises en correspondance et aboutir à une mauvaise détection. Cela implique également que si des documents fortement similaires sont présents dans des classes différentes, la méthode présentée ne pourra les différencier.

Une des principales perspectives est liée à l'optimisation du temps de traitement. L'extraction des points d'intérêt est rendue plus rapide grâce à l'utilisation de SURF plutôt que SIFT. Néanmoins l'extraction des points peut prendre de 3 à 10 secondes par images. Pour cette raison, l'ensemble des points d'intérêt des documents des bases à analyser sont extraits par un traitement hors-ligne. Ceci permet d'économiser en moyenne 5 secondes par image et de passer à un temps de traitement d'environ 3 secondes par image.

Les traitements seraient également plus rapides à basse résolution. Si l'image est mise en correspondance, il n'y a alors pas besoin de faire le traitement à la résolution originale de l'image. Des premiers tests ont montré que la plupart des images A4 à 200 dpi redimensionnées à 60% de leur taille originale sont toujours détectées par le programme.

Le traitement pourrait également être accéléré en utilisant des descripteurs plus légers comme ceux utilisés dans [TKI11], en extrayant moins de points, en quantifiant les vecteurs de caractéristiques ou encore, en développant la sélection de points d'intérêt telle que nous l'avons présenté dans la partie 3.3.2.

L'utilisation des sacs de mots visuels ("bags of visual words") [YJHN07] permettrait de détecter si potentiellement l'image est présente sur le document avant de chercher à la mettre en correspondance précisément. De plus, l'usage des sacs de mots visuels permettraient d'avoir une mesure plus "souple" car ils ne prennent pas en compte l'information de position des points d'intérêt. L'information est obtenue de manière plus rapide puisqu'il n'y a pas de transformation géométrique à valider. La détection est alors faite en deux étapes successives : les sacs de mots visuels sont utilisés pour la détection, puis l'extraction de l'image est faite par la méthode présentée dans ce chapitre.

Les techniques de sacs de mots visuels sont développées dans le chapitre suivant.

Chapitre 4

Classification par apprentissage de l'image et du texte

Dans ce chapitre, des techniques de classification d'images de documents par apprentissage supervisé sont présentées. L'utilisation d'images d'apprentissage permet de prendre en compte la diversité des documents au sein d'une base de documents complexes. Depuis quelques années, les techniques basées sur les points d'intérêt et notamment les sacs de mots visuels sont de plus en plus utilisées pour la recherche et la classification d'images naturelles [YJHN07]. L'objectif principal de ce chapitre est d'étendre ces techniques à la classification d'images de documents.

Dans la suite de ce chapitre, nous parlons tout d'abord de la problématique de la performance d'algorithmes de traitement et d'analyse d'images de documents dont le contenu est très varié. Ensuite, la technique des sacs de mots ("Bags of Words" BoW) est présentée. C'est une des techniques les plus répandues et efficace pour la classification de texte ([Seb02], [WPS06], [LBH+09]). Une variante de cette technique basée uniquement sur l'analyse du texte est celle des sacs de mots visuels ("Bags of Visual Words" BoVW ou "Bags of Features" BoF) qui consiste en une analyse statistique de l'apparition de certains motifs dans l'image. Si les BoF sont très répandus pour l'analyse d'images naturelles, ils sont rarement utilisés dans le cadre de l'analyse d'images de documents. Nous présentons une nouvelle base de documents et utilisons les BoW et les BoF pour la classification de cette base de documents. Le premier apport de ce chapitre est ainsi d'appliquer et d'étudier les performances des BoF pour la classification d'images de documents.

Dans la seconde moitié de ce chapitre, de nouvelles techniques basées sur l'adaptation des sacs de mots visuels "standard" sont présentées. Ces adaptations sont inspirées de l'article de Boiman *et al* [BSI08], qui montre que l'utilisation des SVM (Machine à vecteurs de Support, également appelés Séparateur à Vaste Marge) conjointement avec le partitionnement des descripteurs a tendance à réduire les performances de l'étape de reconnaissance des BoF. Les auteurs suggèrent d'utiliser les k-PPV (Plus Proches Voisin) à la place des SVM et de ne plus quantifier les données. De nouvelles techniques liant l'extraction de points d'intérêts et l'apprentissage par k-PPV pour la classification d'images sont détaillées à la fin du chapitre. Notre objectif ici est d'adapter et d'étendre à la classification complète d'images de documents ce qui jusqu'ici a été testé uniquement à la reconnaissance de logos dans les images. Par rapport aux BoF "standard", une nette amélioration des performances de classification est alors observée.

4.1 La problématique de la diversité des documents

Deux principaux types de documents peuvent poser problème si l'on essaie de les classer avec l'une des deux techniques présentées dans les chapitres précédents. Les documents qui ont une faible variabilité inter-classe rendent difficile la différenciation entre une ou plusieurs classes. Les documents qui ont une forte variabilité intra-classe rendent difficile la reconnaissance des éléments d'une même classe. L'utilisation de techniques d'apprentissage permet de prendre en compte ces diversités.

Des exemples d'images avec une faible variabilité inter-classe sont présentées sur la figure 4.1. Sur cette figure sont représentés des formulaires de satisfaction provenant de trois classes différentes. Les images sont quasi-identiques, le principal changement entre les classes est la langue : la première est en anglais, la seconde en français et la troisième en espagnole. La mise en page de ces documents est également très similaire entre ces trois classes. De plus, sont présents les même logos, cadres et cases à cocher.

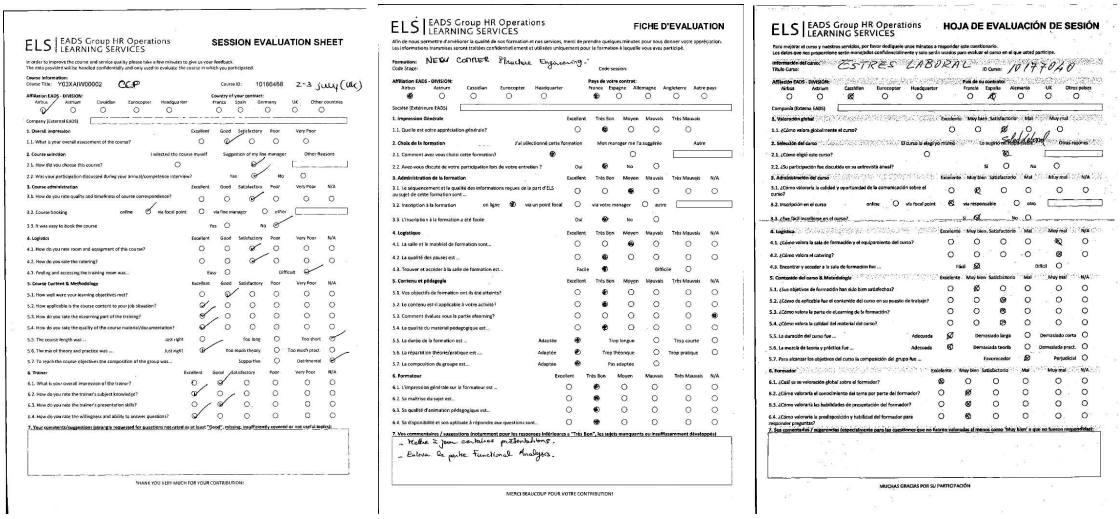
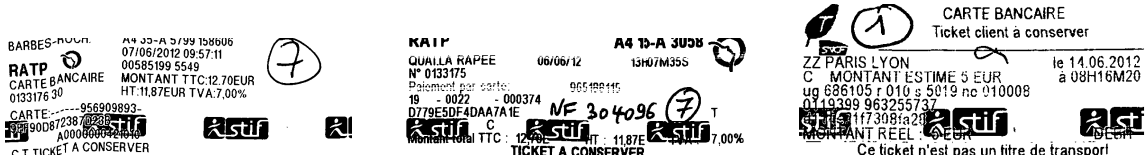


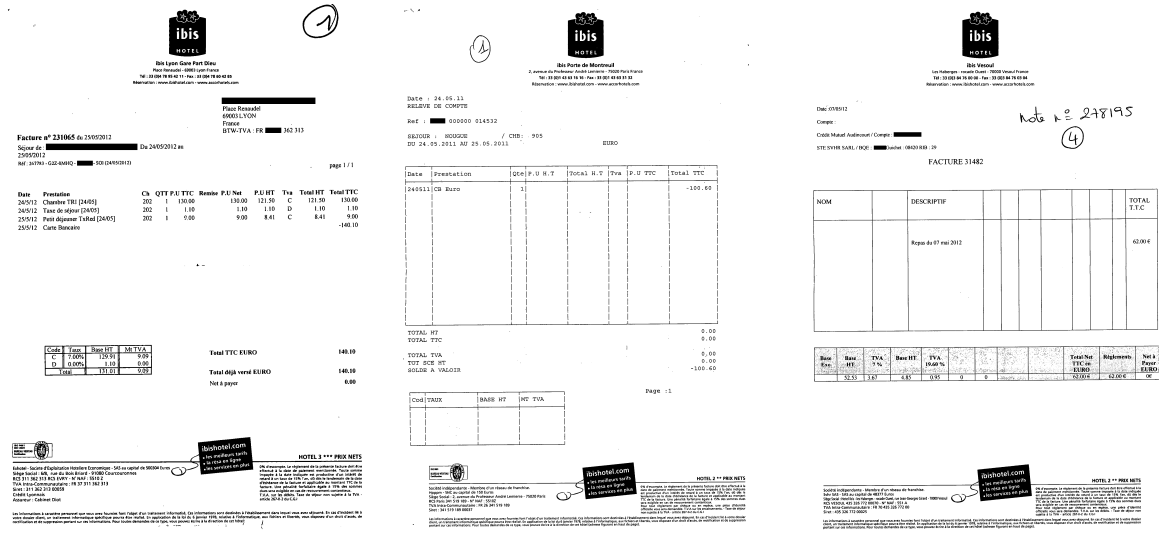
FIGURE 4.1 – Exemples de trois documents appartenant à trois classes différentes. Ces classes ont une faible variabilité inter-classe, c'est-à-dire que les documents provenant des trois classes présentent de fortes similarités visuelles.

Un ensemble d'images réparties sur deux classes sont présentées dans la figure 4.2. Chaque classe a la particularité de posséder une forte variabilité intra-classe. Ainsi, les éléments composant la classe "tickets de métro" et "factures IBIS" ont un visuel significativement différent d'une image à l'autre. Sur ces images, la mise en page diffère beaucoup d'un document à l'autre. Cependant une portion des informations reste similaire (par exemple le logo "stif" sur les tickets RATP et le logo IBIS ainsi que le pied de page des factures des hôtels IBIS) , ce qui laisse supposer qu'il est tout de même possible de regrouper ces documents malgré ces grandes variations.

Pour pouvoir différencier des images trop similaires ou regrouper des images dissimilaires, il est nécessaire d'utiliser plus de connaissances. Dans ce chapitre, les images de documents sont classées à l'aide d'algorithmes d'apprentissages supervisés. L'apport de connaissances se fait alors *via* les images d'exemples données pour chaque classe à reconnaitre. À noter également que contrairement au chapitre précédent, le cadre d'application de ce chapitre ne se limite pas aux documents semi-structurés.



a) Tickets RATP



b) Factures IBIS

Figure 4.2 – Exemples de deux classes de documents ("RATP" et "IBIS") dont les différents documents présentent de grandes variations visuelles. Ces classes ont une forte variabilité intra-classe. Une portion des documents reste tout de même similaire d'un exemplaire à un autre.

4.2 Les sacs de mots (BoW)

Comme nous l'avons vu dans le chapitre précédent, la mise en correspondance de points d'intérêt est une technique précise et robuste. Malheureusement dans certaines configurations elle échoue :

- Si des documents d'une même classe sont trop différents les uns des autres et n'ont pas de parties communes (les documents ne sont pas semi-structurés).
- Si des documents de différentes classes sont trop similaires. Il se produit le problème d'auto-similarité, la technique met en correspondance les documents qui sont alors confondus.

Ces deux problèmes viennent du fait qu'en plus d'extraire des points d'intérêt, on essaye de les mettre en correspondance pour valider une transformation géométrique. Dans le cas où aucune portion des documents n'est commune, aucune transformation n'est trouvée. Dans le cas opposé, quand les images (bien que différentes) ont de grandes parties communes, une transformation est trouvée.

Les techniques basées sur les sacs de mots et sacs de mots visuels permettent de passer outre ces problèmes car la position des points d'intérêt n'est pas utilisée. Chaque document est vu comme un sac contenant des mots (ou des mots visuels), le document est alors décrit par la fréquence d'apparition des mots (ou des mots visuels) le composant.

4.2.1 Utilisation du texte pour la classification d'image

Les méthodes les plus répandues pour comparer et classifier des images de documents consistent à comparer leur contenu textuel [Seb02], [WPS06], [LBH+09] ou encore [HL09]. L'utilisation du texte est rendue possible grâce aux logiciels d'OCR qui permettent de faire la reconnaissance des caractères contenues dans une image.

Les techniques de sacs de mots ("Bags of Words" - BoW) reposent sur le principe que le type d'un document peut être déterminé à partir des mots qu'il contient. Pour mettre en place la technique des BoW, la première chose à faire est de constituer un dictionnaire qui représente le vocabulaire (*i.e.* l'ensemble des mots), que l'on prend en compte pour l'étude. Par défaut, le dictionnaire est constitué à partir de tous les mots différents présents dans l'ensemble des documents à traiter. Cependant, certains mots sont exclus du dictionnaire car, étant tellement courant, ils n'apportent pas de sens au texte (on parle de mots vides ou "stop words"). Dans la langue française, on exclue par exemple les articles "le", "la", "du", etc. Les mots vides sont généralement les prépositions, les articles et les pronoms. On enlève également les mots qui sont trop courts (moins de 3 lettres). Un autre filtrage est fait sur les mots les moins fréquents. En effet, un mot qui n'est utilisé que dans un seul document n'apporte aucune information car il n'apparait pas dans les documents du même type. De plus, ceci permet généralement d'enlever des mots qui avaient été mals reconnus par l'OCR. Une autre étape permettant de réduire le dictionnaire est la lemmatisation. L'objectif est de remplacer des mots d'une même famille par un seul mot commun. Par exemple "directeur" et "directrice" ou encore des formes conjuguées "payé", "payer", "payez", etc.

Une fois le dictionnaire créé, chaque document est décrit par un vecteur (ou un histogramme) représentant le nombre d'occurrences de chacun des mots du dictionnaire dans le document. La comparaison entre deux documents se fait par la comparaison de leur vecteur respectif. Les vecteurs sont généralement de grandes dimensions, puisque le nombre de dimensions est égal au nombre de mots du dictionnaire. La similarité entre deux vecteurs est calculée à l'aide du cosinus [Seb02], [LBH+09], [HL09] :

$$Similar(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|},$$

où θ est l'angle entre les vecteur $\boldsymbol{A}$ et $\boldsymbol{B}$, $\cdot$ est le symbole du produit scalaire et $\|\|$ représente la norme au sens mathématique.

Au lieu de calculer le nombre d'occurrences de chacun des mots du dictionnaire dans le document, on peut également calculer pour chaque mot une valeur appelée "tf-idf" (Term Frequency - Inverse Document Frequency) représentant à la fois la fréquence d'apparition d'un mot dans le document (tf) et le nombre de documents dans lequel ce mot apparait (idf) [Seb02], [LBH+09], [HL09]. Les termes tf et idf sont définis comme suit :

$$TF(i) = \frac{N(i)}{M}, et\ IDF(i) = log\frac{D}{T_k(i)},$$

où $N(i)$ correspond au nombre d'occurrences du mot i dans le document, M est le nombre de mot total dans le document, D correspond au nombre total de documents dans le corpus et $T_k(i)$ est le nombre de documents dans lequel le terme i apparait ($T_k(i) \neq 0$). On obtient alors pour chaque mot la valeur tf-idf :

$$TF\_IDF(i) = TF(i) * IDF(i).$$

La figure 4.3 illustre une chaîne de traitements classique basée sur les sacs de mots.

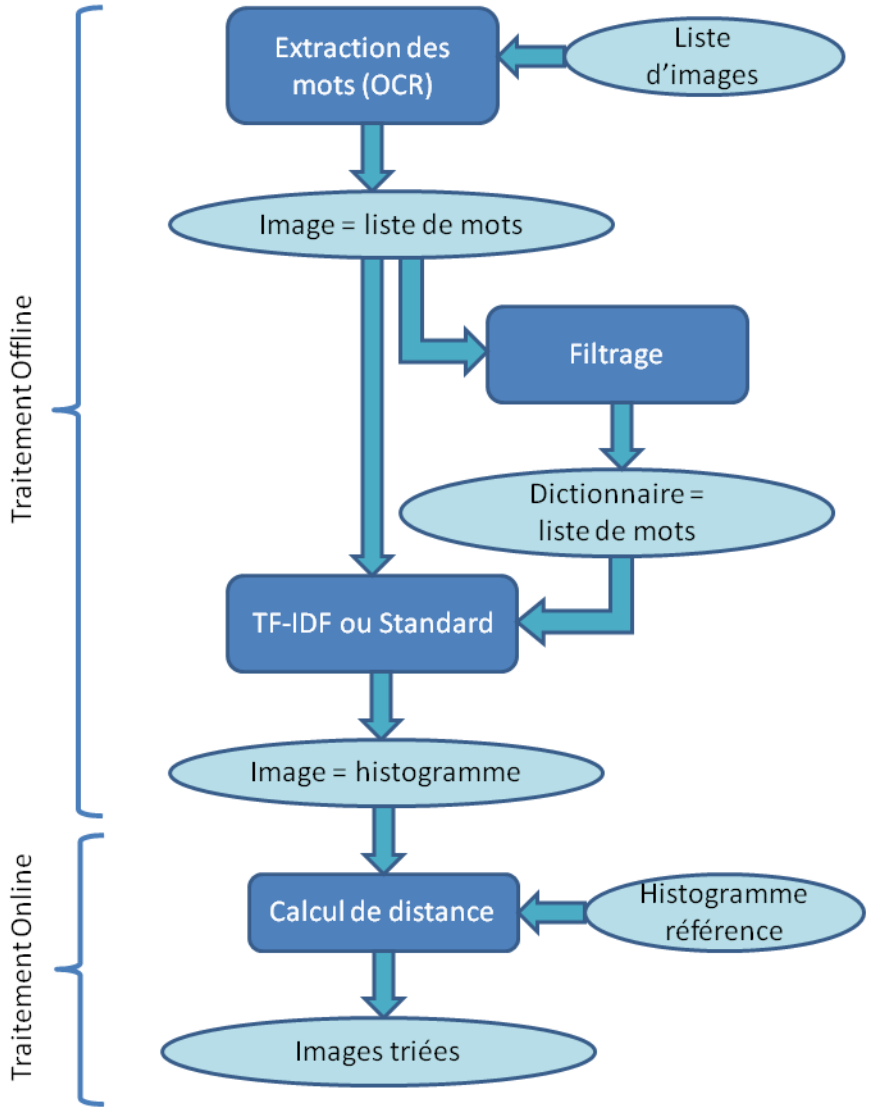


Figure 4.3 – Principe de fonctionnement des sacs de mots appliqué à la comparaison d'images.

4.2.2 Les applications des BoW

Joachims pose en 1998 les bases de la classification de textes en combinant la caractérisation des documents par leur fréquence d'apparition (tf-idf) et les SVM [Joa98]. La base de documents testée est : Reuters-21578 [1]. 9603 documents sont utilisés pour l'apprentissage et 3299 pour les tests. Sur les 135 classes, seules 90 sont utilisées. La base Reuters-21578 est connue pour avoir une correspondance directe entre les mots et les classes. Par exemple, pour la classe "blé", le nombre de mots "blé" utilisés dans le document est une caractéristique pertinente. L'auteur utilise le "Precision-recall breakeven point" pour mesurer les performances du système de classification. Ce point est obtenu lorsque le rappel est égal à la précision. Une micro-moyenne (ceci correspond à la moyenne des mesures de chaque classe) de 86,4 % est obtenue.

Les auteurs de [LBH+09] classent également des documents en fonction de leur contenu

1. http ://www.daviddlewis.com/resources/testcollections/reuters21578/

textuel. Les techniques de sacs de mots sont testées sur une base de documents provenant de la société ITESOFT. La base est constituée de 2000 fichiers répartis dans 24 catégories. Le texte provient du résultat de logiciels d'OCR. Les documents présents dans la base sont de type : bulletins de salaires, certificats, avis d'impôt, etc. La base est caractérisée par la variété des documents, leur faible contenu textuel (130 mots en moyenne par document) et la répartition très hétérogène des différentes classes. L'utilisation des sacs de mots avec pondération "tf-idf" et apprentissage par SVM permet d'obtenir une précision moyenne de 95,8 % sur la base.

Quelques uns des derniers travaux sur les BoW consistent à trouver les meilleurs facteurs de pondération en prenant en compte des informations apportées par chaque classe [Ko12], ou encore à remplacer l'utilisation de dictionnaires par des tables de hachage ce qui permet de réduire grandement la consommation en mémoire [WDL+09].

4.3 Les sacs de mots visuels (BoF)

Les BoF sont directement inspirés du principe des sacs de mots ("Bags of Words" BoW) décrit dans la section précédente.

4.3.1 Principe de la méthode

La principale différence entre les BoW et les BoF est que les BoW reposent sur les mots alors que les BoF reposent sur les "mots visuels" que l'ont peut vulgariser à des "petits bouts d'image". Dans la grande majorité des applications, les "mots visuels" sont des points d'intérêt. L'algorithme des BoF est présenté sur la figure 4.4. Son fonctionnement peut être résumé en 4 principales étapes :

1. Les points d'intérêts sont extraits sur l'ensemble des images que l'on souhaite classifier.
2. Un partitionnement en k groupes de ces points d'intérêt est créé. Ces k groupes représenterons les k dimensions de l'histogramme qui sert à décrire chaque image.
3. Pour caractériser une image, on regarde à quel groupe appartient chaque point d'intérêt extrait de cette image et l'on incrémente en conséquence le nombre d'occurrences des groupes correspondants. L'image est caractérisée par l'histogramme ainsi créé.
4. Les images sont classifiées en utilisant un algorithme de classification supervisé.

Habituellement, les points d'intérêts sont extraits avec SIFT ou SURF ; le partitionnement des descripteurs est effectué avec l'algorithme k-means et l'apprentissage supervisé est fait par SVM.

Techniquement, les BoF sont calculés en hors-ligne : il n'y a besoin d'aucune interaction avec un utilisateur pour créer le partitionnement ni pour calculer les histogrammes des images. Le paramètre principal à fixer dans cette technique est le nombre de groupes k créés par les k-means. Il n'est pas nécessaire d'augmenter grandement la valeur de k car le partitionnement créé par les k-means contient alors de nombreuses classes vides. Par contre, si k est trop faible cela peut diminuer les performances de détection car le dictionnaire n'est plus assez varié pour décrire les images.

Afin de classifier des documents nous utilisons les SVM comme cela est fait dans la majorité des articles utilisant les BoW ou les BoF [NJT06], [YYGH09], [HJS09] et [WYY+10]. Le principe des SVM consiste à déterminer les hyperplans permettant de séparer chacune des classes en maximisant l'écart entre le séparateur et les points de la base d'apprentissage. Dans le cas où les données ne sont pas linéairement séparables, les

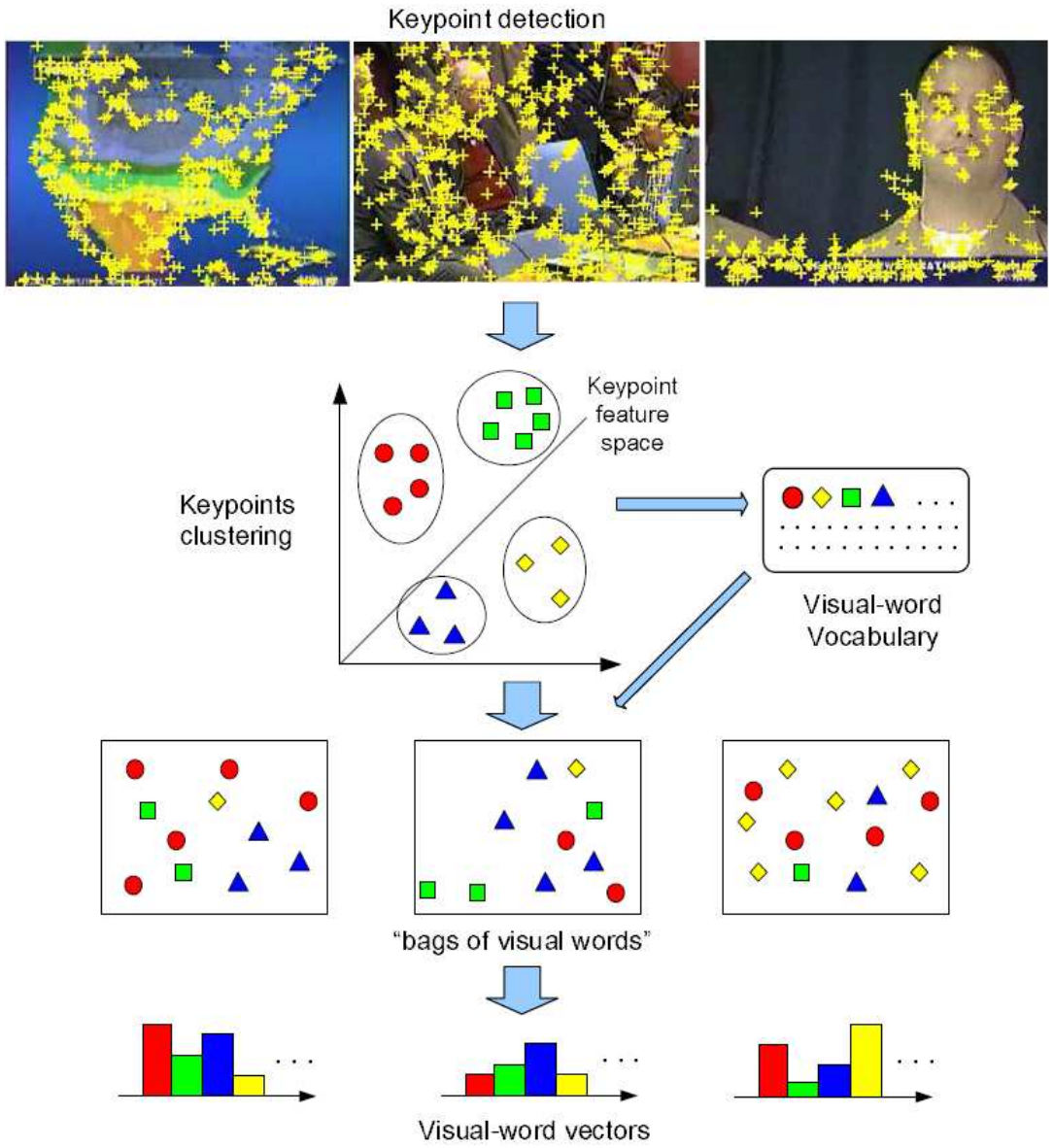


Figure 4.4 – Principe de fonctionnement des BoF issu de [YJHN07]. 1) Les points d'intérêt sont extraits sur chaque image. 2) On crée un partitionnement à k classes de l'ensemble des points d'intérêt. 3) A chaque point d'intérêt est associée une des k classes grâce au partitionnement. 4) Chaque image est décrite par un histogramme représentant le nombre d'occurrences de chacune des k classes.

SVM transforment l'espace de représentation des données en augmentant leur dimension. Dans cet espace de dimension plus grand, il est probable qu'une séparation linéaire soit possible. On appelle cette technique : l'astuce du noyau ("kernel trick") ; cette idée est illustrée sur la figure 4.5. Pour plus de détails, le lecteur peut se reporter aux nombreux articles traitant des SVM [Cor02], [NJT06], [YYGH09], [HJS09] et [WYY+10].

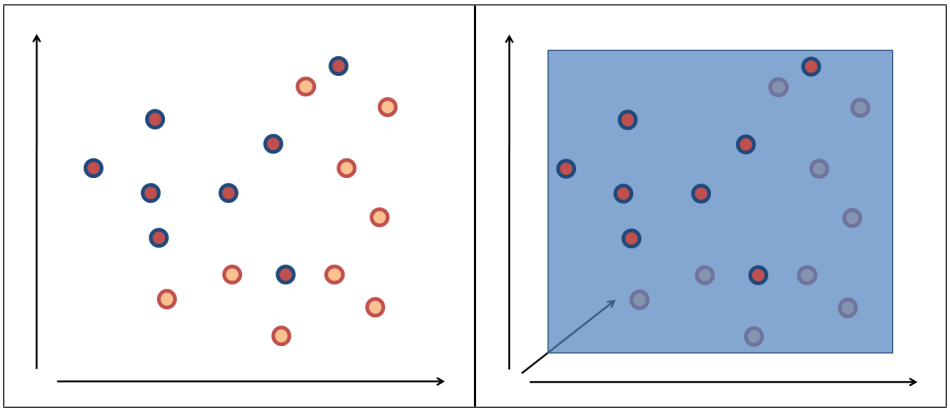


FIGURE 4.5 – "Kernel trick" utilisée par les SVM. Le nombre de dimensions est fictivement augmenté ce qui permet de passer d'un cas non linéairement séparable (image de gauche) à un cas linéairement séparable (image de droite).

4.3.2 Exemples d'applications

Plébiscité par la communauté "computer vision", les BoF sont utilisés pour la recherche d'images naturelles et principalement pour des tâches de CBIR [VC09], de reconnaissance d'objets [Low04] et de classification d'images [YJHN07], [CDF+04].

Les auteurs de [CDF+04] utilisent les BoF pour classifier des images naturelles. Deux bases de données sont testées. La première contient 1776 images séparées en 7 classes : 792 visages, 150 bâtiments, 150 arbres, 201 voitures, 216 téléphones, 125 vélos et 142 livres. La seconde contient 3721 images séparées en 5 classes : 450 visages, 1074 avions vus de profil, 651 voitures vues de derrière, 720 voitures vues de profil et 826 motos vues de profil. Les auteurs testent deux méthodes d'apprentissage supervisées : la classification naïve bayésienne et les SVM . L'utilisation des SVM leur permet de réduire le taux d'erreur de la première base à 15%, alors qu'il était de 28% pour la classification naïve bayésienne. Un taux d'erreur de 4% est obtenu pour la seconde base, en utilisant les SVM.

Les auteurs de [YJHN07] utilisent les BoF pour classifier des images de TRECVID 2005 et PASCAL 2005. TRECVID contient 29252 images extraites de vidéos que les auteurs souhaitent classifier en 20 classes, sachant qu'une image peut avoir aucun, un ou plusieurs labels. PASCAL contient 1578 images labellisées en 4 classes : motos, vélos, personnes et voitures. Ici encore, les SVM sont utilisés pour faire l'apprentissage. Les auteurs calculent pour chaque base la MAP ("Mean Average Precision"). Pour la base TRECVID, qui est très complexe, les auteurs obtiennent au mieux : $MAP = 0,278$. Pour la base PASCAL, qui est plus simple, les auteurs obtiennent au mieux : $MAP = 0,777$.

Pour ce qui est de l'application aux images de documents, les BoF ont été appliqués pour faire de la reconnaissance de logos [RL09] (la reconnaissance du logo permettant ensuite de classifier le document) et de lettrines [NCO11] (dans ce cas est appliquée une variante appelée "bags of strokes"). Les BoF sont également utilisés pour faire de la recherche de mots ("word spotting") [RATL11] et de caractères manuscrits [SUL11]. Les BoF ont donc été appliqués a plusieurs reprises pour la recherche de sous parties de document telles que les logos, lettrines, et mots. Les auteurs de [RL09] classifient des images de documents en fonction du logo reconnu sur les images. Leur base est constituée de 1000 images de documents qui ont été faxées puis scannées, représentant des factures, des lettres, des reçus, etc. Les documents peuvent contenir du texte manuscrit, typographié,

des logos, des tampons, des tableaux, etc. La base contient 18 logos apparaissant dans 180 images, les autres images n'ont pas de logo et devrons être rejetées. Les auteurs utilisent des métriques basées sur le nombre d'éléments correctement labellisés, composés des "true positif" TP et "true negative" TN et sur le nombre d'éléments mal labellisés "false positif" FP et "false negative" FN. Sont alors calculés le "taux de vrai positif" TPR et le "taux de faux positifs" FPR tels que :

$$TPR = \frac{TP}{TP + FN} \; et \; FPR = \frac{FP}{FP + TN}.$$

Les performances obtenues sont les suivantes : $TPR = 92,2\%$ et $FPR = 1\%$.

Notre objectif est d'étendre l'utilisation des BoF à la recherche et à la classification de documents entiers.

4.4 Application des BoW et BoF aux images de documents

Afin de tester les techniques de BoW et BoF, une nouvelles base de documents est présentée dans cette section. Nous détaillons ensuite les protocoles d'application des deux techniques avant de présenter et comparer les résultats obtenus sur la nouvelle base de documents.

4.4.1 Base de documents

Dans ce chapitre une nouvelle base de documents est utilisée : BD_VAR. Chaque document ne contient qu'une page. La base est constituée de 5978 documents. Parmi toutes les classes de la base, seules 12 classes sont connues. Ces 12 classes représentent 1982 documents, soit environ 33% de la base. Cette base a été choisie car certaines classes ont une forte variabilité intra-classe et d'autres ont une faible variabilité inter-classe comme cela a été présentée dans la section introductive de ce chapitre.

Sur la figure 4.6 est présentée un exemple de chaque image de chaque classe. Les tickets de train "SNCF" ont tous le même format. Il y a quelques variations dans la mise en page mais les tickets sont très similaires les uns par rapport aux autres. Les formulaires de cases à cocher "ElsUk", "ElsEs" et "ElsFr" représentent trois classes différentes mais de mises en pages très similaires. Se sont des documents A4 dont la plupart sont de bonne qualité (les conditions sont favorables à l'utilisation d'un OCR). Les factures "IBIS" présentent de nombreuses variations car la plupart ont été émises d'hôtels différents. Les feuilles "Lot" sont des feuilles A4 contenant généralement très peu d'informations parmi lesquelles : un code barre et une ligne contenant un lien URL. Les tickets d'autoroute "ASF" et "SANEF" et les tickets de métro "RATP" ont une taille fixe contrairement aux tickets de restaurant "Buffalo", "Flunch" et "Quick" dont la longueur est variable. La qualité de ces documents est très variable car ils sont imprimés sur des papiers de moins bonne qualité et dont l'encre a parfois disparu ou bavé. Enfin les tickets de métro "SNCF" présentent la particularité d'être des documents de petite taille dont la plupart sont tordus ou pliés.

Le tableau 4.1 résume la répartition des documents dans la base. La majorité des documents sont de type "autre", c'est-à-dire des documents à rejeter. Les feuilles "Lot", les formulaires "ElsUk" et "ElsEs", les tickets "SNCF" et "RATP" représentent une grande proportion des documents à reconnaitre tandis que les documents de types "ASF", "IBIS", "Flunch", "BUffalo", "QUick" et "SANEF" sont très minoritaires.

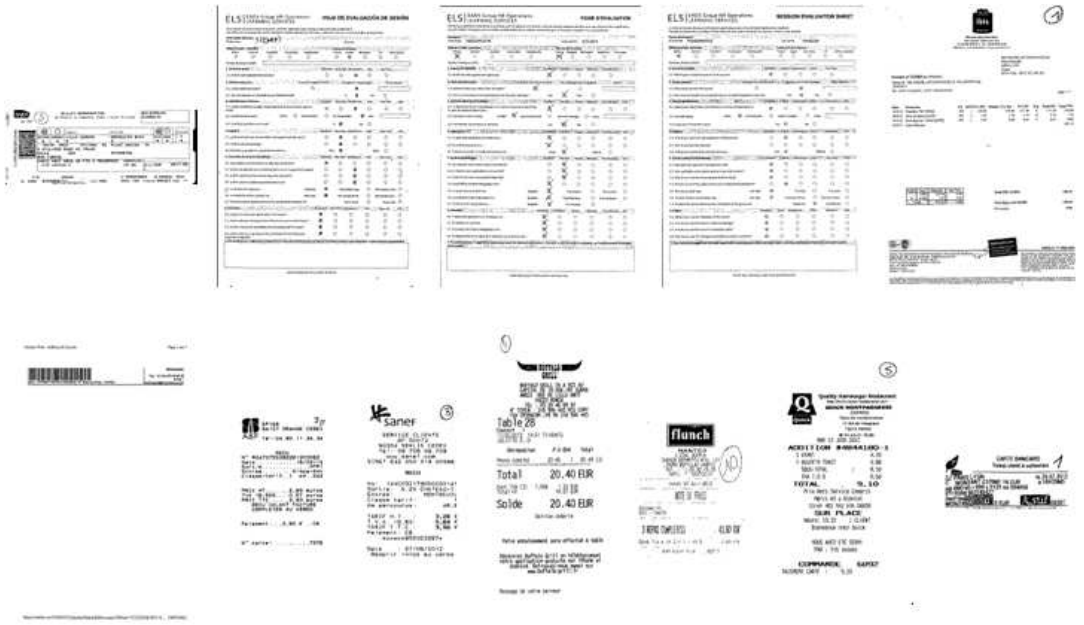


FIGURE 4.6 – Exemple de chacune des 12 classes présentes dans la base d'image. De gauche à droite et haut en bas figurent les classes suivantes : "SNCF", "ElsUk", "ElsEs", "ElsFr", "IBIS", "Lot", "ASF", "SANEF", "Buffalo", "Flunch", "Quick" et "RATP".

TABLE 4.1 – Répartition des documents dans la base.

Nom de la classe	Nombre de documents
Lot	675
ASF	34
IBIS	46
SNCF	199
Flunch	9
RATP	115
Buffalo	18
Quick	17
SANEF	26
ElsUk	435
ElsEs	304
ElsFr	104
Rejet / Autre	3996

4.4.2 Tests et résultats

Pour la mise en application des BoW, nous avons choisi de ne pas pondérer les mots par la méthode "tf-idf" alors que la pondération des termes selon leurs fréquences d'apparition est "théoriquement" censée améliorer les résultats de la classification. Cette décision a été prise car le nombre de mots dans les documents d'une même classe peut grandement varier, ce qui relativise l'intérêt qu'il y aurait à utiliser "tf-idf" sur notre base de documents. Nous avons également choisi de créer un dictionnaire contenant les 2000 mots les plus utilisés

dans l'ensemble des documents de la base. Pour l'application des BoF, SURF a été utilisé pour extraire et caractériser les points d'intérêt. On rappel que la description de SURF peut être trouvée dans la section 3.2.1 du chapitre 3.

Les BoW et les BoF avec apprentissage par SVM ont été testés sur la base BD_VAR. Cinq images par classe sont utilisées pour l'apprentissage des SVM. La base étant constituée d'autres documents que ceux présents dans les 12 classes étudiées, une classe de rejet a été créée en plus. Les documents de "type" rejet représentent une grande partie de la base, 149 images ont été utilisées pour l'apprentissage de cette classe rejet.

Les résultats sont présentés dans le tableau 4.2. Le nombre de documents par classe correspond au nombre de documents à reconnaitre, sachant que 5 exemples sont utilisés pour l'apprentissage par classe (sauf pour la classe de rejet qui en utilise 149). Tous les prétraitements nécessaires à l'application des BoF et BoW ont été réalisés hors-ligne (OCR, extraction des points, ...). Il faut environ 10 millisecondes aux SVM pour affecter chaque image à une classe.

Table 4.2 – Performances des BoF et BoW avec SVM. La moyenne est calculée uniquement sur les 12 classes (sans le rejet). Afin de ne pas privilégier les classes ayant plus de documents que les autres, la moyenne calculée n'est pas pondérée.

		BoF		BoW	
Modèles	Nb docs	Rappel	Précision	Rappel	Précision
Lot	670	0,9955	0,9794	0,9895	0,9822
ASF	29	0,8966	0,9286	1	0,3924
IBIS	41	0,8537	0,3182	0,7804	0,5614
SNCF	194	0,9794	0,9005	0,9948	0,9897
Flunch	4	1	0,3333	1	0,0555
RATP	110	1	0,4089	0,6181	0,2753
Buffalo	13	1	0,1970	1	0,8125
Quick	12	1	0,0833	1	0,6667
SANEF	21	1	0,4118	1	0,6363
ElsUk	430	0,9907	0,9552	1	1
ElsEs	299	1	0,9492	1	1
ElsFr	99	0,6566	0,9559	1	1
Rejet	3847	0,8724	0,9964	0,9087	0,9835
Moy		0,9477	0,6184	0,9485	0,6977

Globalement les performances entre les BoW (basés sur le texte) et les BoF (basés sur l'image) sont du même ordre de grandeur. Ce premier résultat est intéressant car il démontre que l'utilisation de l'image peut permettre aussi bien de classer les documents qu'avec une technique utilisant exclusivement le contenu textuel. On peux également constater que les BoW fonctionnent mieux sur certains documents tandis que les BoF fonctionnent mieux sur d'autres documents, ce qui laisse envisager une certaine complémentarité des deux techniques. Par exemple, les BoW sont très performants sur des formulaires "ElsUk", "ElsEs" et "ElsFr" qui ont exactement le même visuel mais qui utilisent des langues différentes. Sur la classe "ElsFr", les BoF n'obtiennent qu'un rappel de $0,6566$ car ces documents sont confondus avec les classes "ElsUk" et "ElsEs". Au contraire, pour la classe "RATP" qui est constituée de petits tickets dont la qualité n'est pas tou-

jours très bonne, les caractéristiques visuelles marchent bien mieux que le texte. En effet, le nombre de mots est faible et les erreurs d'OCR nombreuses sur ces documents.

Le principal intérêt des BoF et BoW réside dans leur grande généricité ainsi que la rapidité de prise de décision des SVM. Les performances sur une base complexe telle que BD_VAR sont bonnes, mais de nombreuses classes ont une précision qui est moyenne : "IBIS", "Flunch", "RATP", "SANEF" ; voire très médiocre : "Buffalo" et "Quick". La précision de ces classes est très basse car des documents qui auraient du être rejetés ont été confondus avec ces classes. La variété des documents est tellement grande que les BoF ne sont pas assez précis pour faire la différence entre des images comportant une certaine similarité. La figure 4.7 montre un exemple de documents confondus.

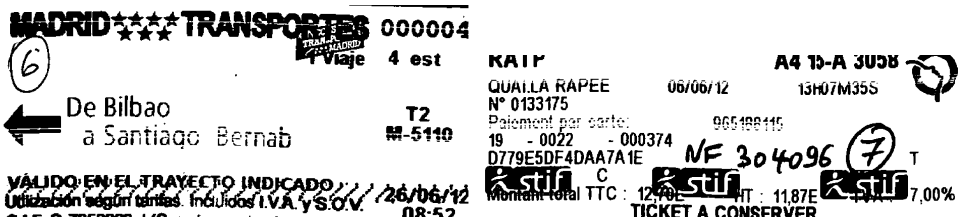


FIGURE 4.7 – Exemple de confusion entre un document de la classe "Rejet" (à gauche) qui a été labellisé à tort comme un document de la classe "RATP" (à droite).

Boisman *et. al* [BSI08] expliquent que l'étape de partitionnement qui est appliquée dans les BoF dégrade le pouvoir discriminant des descripteurs (cette étape de partitionnement correspondant à une quantification des descripteurs). Le manque de précision des BoF pour la classification d'images de documents est donc certainement lié à cet étape de partitionnement. Cependant le partitionnement est nécessaire pour pouvoir appliquer les SVM. Si l'étape de quantification est supprimée, il faut alors utiliser une autre technique d'apprentissage. Les auteurs conseillent d'utiliser les plus proches voisins à la place des SVM et de ne plus faire de partitionnement.

Dans la section suivante, des variantes des BoF sont présentées en accord avec l'idée défendue par Boisman *et. al*. Nous observons alors une nette amélioration des performances du système.

4.5 Variantes des BoF sans partitionnement et avec apprentissage par k-PPV

Pour l'utilisation des BoF, le partitionnement des points d'intérêt est nécessaire à l'utilisation des SVM. Cela permet de créer un dictionnaire avec un nombre de dimensions relativement faible et ne pas entrainer de sur-apprentissage. Cependant cela n'est pas nécessaire lors de l'utilisation des k-PPV.

4.5.1 Application aux images naturelles

Boisman *et. al* introduisent dans leur article [BSI08], une méthode appelée NBNN (Naive-Bayes Nearest-Neighbor) permettant de classifier des images en comparant les points d'intérêt d'une image à classifier avec les points d'intérêt d'images d'apprentissages.

La technique est résumée dans l'algorithme 2. Le but est de déterminer la classe d'un document requête. Soit $d_1, ..., d_n$ les descripteurs extraits de l'image requête. Les données d'apprentissage sont constituées de l'ensemble des points d'intérêt extrais des images des

différentes classes. Le plus proche voisin d'un descripteur d_i dans la classe C est noté $NN_C(d_i)$.

Algorithm 2 NBNN [BSI08]

INPUT Un index de recherche des PPV pour chaque classe C, utilisé par $NN_C()$.
INPUT Une image requête Q et ces descripteurs d_i.
for each $d_i \in Q$ **do**
  for each classes C **do**
    $tot[C] \leftarrow tot[C] + ||d_i - NN_C(d_i)||^2$
return $argmin_C tot[C]$

Pour classifier une image requête, la distance entre chaque point d'intérêt et son plus proche voisin dans chacune des classes est calculée. Pour chaque classe, la somme de ces distances est ensuite calculée. L'image est alors associée à la classe dont la distance lui est la plus faible. La figure 4.8 illustre ce principe.

McCann et Lowe ont récemment proposé une adaptation de cette méthode appelée LNBNN (Local Naive-Bayes Nearest-Neighbor) [ML12]. L'amélioration consiste à ne pas regarder les plus proches voisins de chacune des classes mais simplement de prendre les k-PPV "globaux" (toutes classes confondues) et d'ajouter à chaque classe la distance du plus proche point trouvé. Si pour certaines classes aucun point ne figure parmi les k-PPV alors on leur ajoute la distance du k+1 ième. La technique est décrite dans l'algorithme 3.

Algorithm 3 LNBNN [ML12]

INPUT Un index de recherche des PPV pour chaque descripteur : $NN(descripteur, \#voisin)$.
INPUT Une table de correspondance $Classe(descripteur)$ permettant de retrouver la classe d'un descripteur.
for each $d_i \in Q$ **do**
  $\{p_1, p_2, ..., p_{k+1}\} \leftarrow NN(d_i, k+1)$
  $dist_B \leftarrow ||d_i - p_{k+1}||^2$
  for each classes C trouvées dans les k-PPV **do**
    $dist_C = min_{\{p_j | Classe(p_j)=C\}} ||d_i - p_j||^2$
    $tot[C] \leftarrow tot[C] + dist_C - dist_B$
return $argmin_C tot[C]$

Les auteurs ont fixé expérimentalement $k = 10$. La figure 4.9 illustre le principe du LNBNN. La principale différence avec le NBNN est que le LNBNN ne cherche pas forcément à trouver un plus proche voisin pour chaque classe. La technique est alors beaucoup plus rapide (et d'autant plus qu'il y a de nombreuses classes) mais sans dégrader les performances en terme de précision. McCann et Lowe ont testé les techniques NBNN et LNBNN sur les bases d'images naturelles "Caltech 101" [FFFP07] et "Caltech 256" [GHP07]. Les images sont redimensionnées de manière à ce que le coté le plus long mesure 300 pixels. Une précision moyenne de 66,1 % est obtenue pour l'utilisation des LNBNN en utilisant 15 images d'apprentissage sur la base "Caltech 101", tandis qu'une précision de 63,2 % est obtenue avec le NBNN.

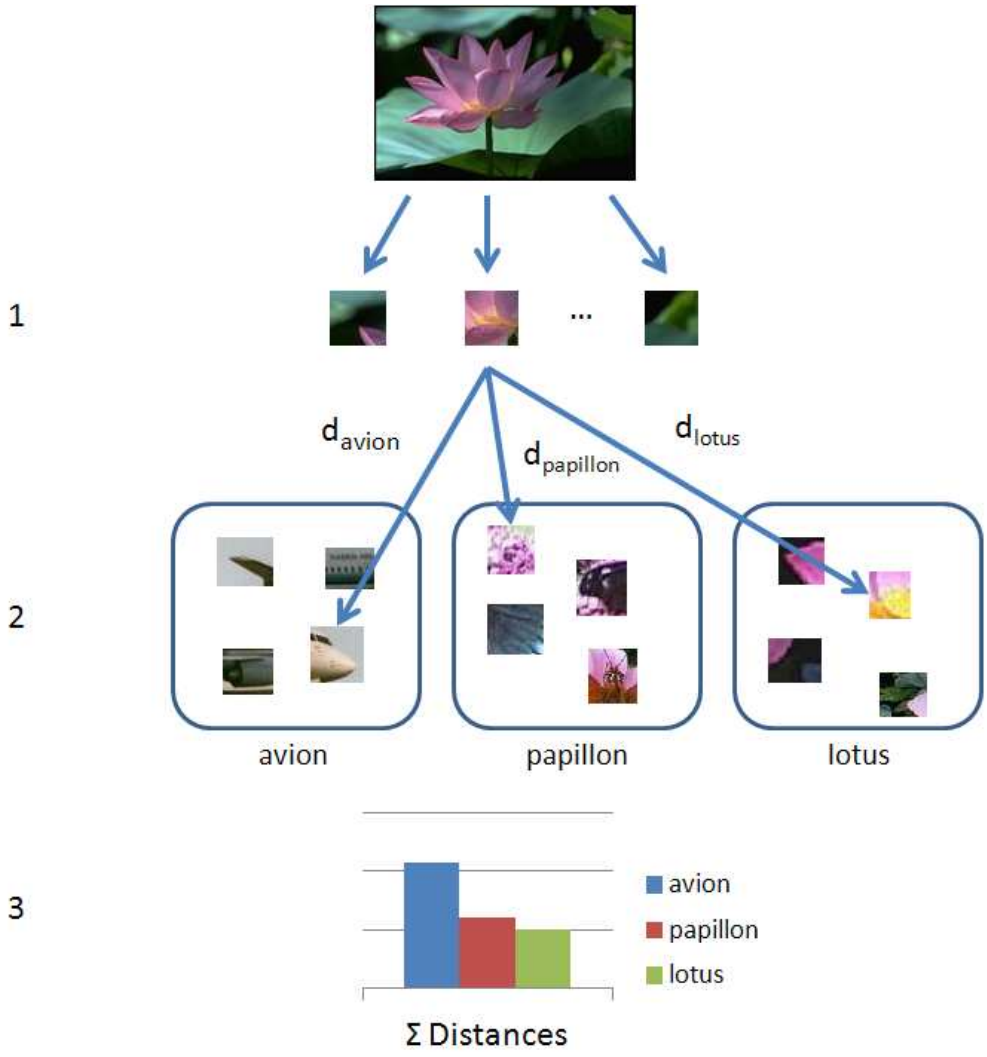


FIGURE 4.8 – Principe du NBNN. 1) Pour chaque mot visuel d'une image, 2) on calcule la distance du mot visuel le plus similaire dans chaque classe d'apprentissage. 3) La somme des distances de chaque classe est calculée afin d'associer l'image à la classe dont la distance lui est la plus faible.

4.5.2 Application aux images de documents

Dans le domaine des images de document, les auteurs de [RL09] effectuent la reconnaissance de logo afin de classifier des images de document en utilisant une technique similaire au LNBNN. La figure 4.10 illustre leur méthodologie. Tout comme pour le NBNN et le LNBNN aucun partitionnement n'est créé. Un exemple de chaque logo à reconnaitre est utilisé pour constituer la base d'apprentissage. L'ensemble des points d'intérêt de ces logos sont extraits et stockés dans un dictionnaire. Ensuite, pour savoir quel logo est contenu dans une image à classifier, chacun des points d'intérêt de cette image est mis en correspondance avec le point d'intérêt qui lui est le plus ressemblant parmi tous les points d'intérêt du dictionnaire. Ceci est fait à l'aide d'une recherche de k-PPV (Plus Proches Voisins). Un filtre est appliqué afin de ne garder que les mises en correspondances de points d'intérêts discriminants. Pour cela les auteurs calculent le ratio de la distance entre le 1-PPV et le 2-PPV et ne gardent que les mises en correspondances dont le ratio est supérieur à un

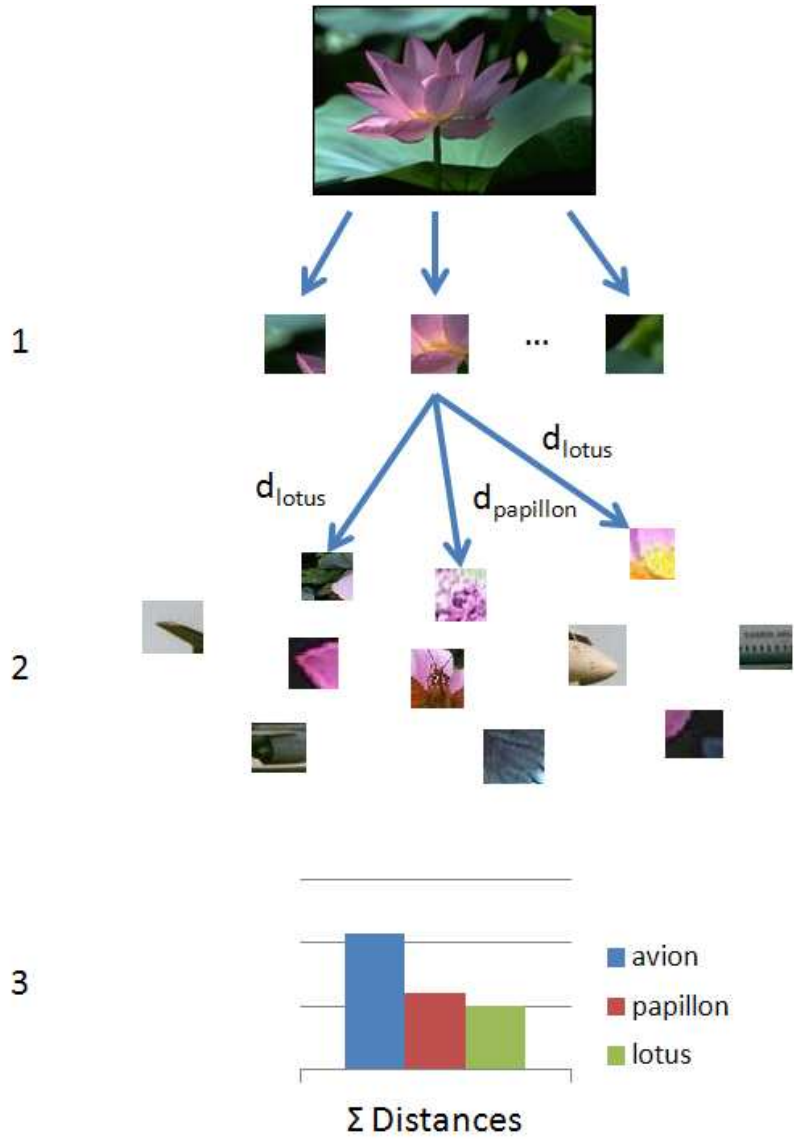


Figure 4.9 – Principe du LNBNN. 1) Pour chaque mot visuel d'une image, 2) on calcule la distance des k mots visuels les plus similaires dans l'ensemble d'apprentissage. 3) Pour chaque classe est ajoutée la distance du PPV trouvé parmi les k-PPV. Si pour certaines classes aucun élément n'a été trouvé on prend la distance du $k+1$ième élément. Ces distances sont ensuite sommées, l'image est alors associée à la classe dont la distance est la plus faible.

seuil fixé. Si la mise en correspondance est valide, la classe à laquelle appartient le point mis en correspondance est enregistré. On procède ainsi pour tous les points de l'image à analyser et on somme le nombre d'occurrences de chaque label. Cette somme est ensuite normalisée par le nombre de points d'intérêt constituant chaque classe (on rappel qu'une classe correspond à un logo). Si la valeur finale est supérieur à un seuil fixé alors les auteurs considèrent que l'image contient le logo ayant le plus d'occurrences.

Nous avons choisi d'adapter la méthode de Rusinol et Lladós car leur champ d'application concerne également les images de documents. De plus, les auteurs font face à deux problématiques qui nous sont familières et auxquelles ne se sont pas confrontés les auteurs

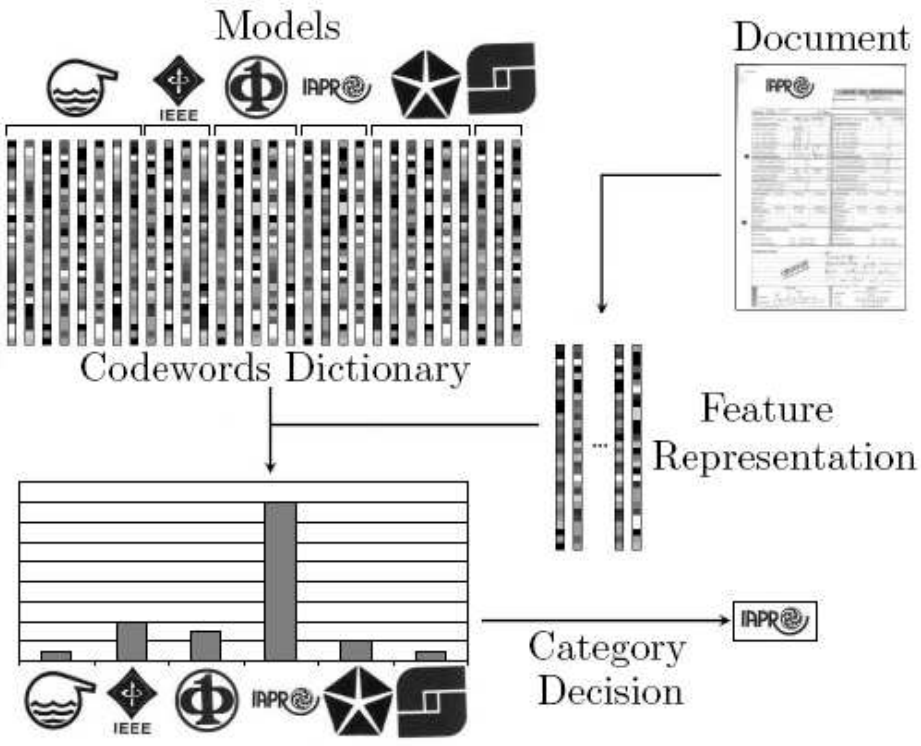


FIGURE 4.10 – La variante des sacs de mots visuels issue de [RL09] se compose de 4 étapes principales. 1) Le dictionnaire est créé à partir de l'ensemble des points d'intérêt constituant chacun des logos à reconnaitre. 2) Chaque point d'intérêt du document à reconnaitre est extrait et mis en correspondance avec le point le plus similaire du dictionnaire. 3) On note alors le label du point d'intérêt trouvé (le nom du logo associé). Chaque point d'intérêt vote pour un label. 4) Le logo ayant la majorité est associé au document.

des techniques NBNN et LNBNN : le rejet de certains documents et la normalisation du score en fonction du nombre de points d'intérêt des classes. En effet, le nombre de points d'intérêt présents sur les images de documents est très variable d'une classe à l'autre. Les documents présentant de nombreux points d'intérêt ont alors tendance à être détectés plus souvent que les documents qui ont peu de points d'intérêt.

4.5.3 Application à une base complexe de documents

L'objectif de cette application est de tester une technique semblable à celle exposée par Rusiñol et Lladós [RL09] mais cette fois pour décrire et comparer la totalité des images des documents et non plus seulement les logos. Nous allons donc adapter et tester la méthode sur la base de documents BD_VAR. Les adaptations nécessaires à notre application sont les suivantes :

- Au lieu d'utiliser uniquement des logos pour l'apprentissage, nous avons utilisé les images complètes des documents.
- Afin de couvrir au mieux la variété des documents présents dans la base, 5 exemples de documents par classe sont utilisés pour l'apprentissage.

Chaque point d'intérêt de l'image à classifier doit être mis en correspondance avec les deux points d'intérêt du dictionnaire les plus similaires (2-PPV). Pour effectuer cette opération nous avons utilisé FLANN. Nous rappelons que FLANN est présentée dans la partie

3.2.2. Cette technique est à la fois non exhaustive et approximative, son principal intérêt est d'être beaucoup plus rapide qu'une méthode exacte. Les mises en correspondance dont les *ratio* sont supérieurs à $t = 0.6$ ne sont pas pris en compte. Le vote de chaque classe est divisé par le nombre moyen de points d'intérêt des documents d'apprentissage de la classe. Le vote final n'est conservé que s'il est supérieur à T (fixé empiriquement), sinon l'image est rejetée. Dans la technique originale, il n'est donc pas nécessaire d'utiliser des images d'apprentissage pour le rejet car celui-ci est fait via le seuil T. Cependant l'utilisation d'images d'apprentissage pour le rejet permet d'améliorer légèrement les résultats de la classification. De plus ces images sont utilisées pour les BoF, nous les utiliserons donc afin de comparer les deux techniques dans des conditions similaires.

Les résultats sont présentés dans le tableau 4.3.

Table 4.3 – Performances de l'adaptation "Adapt" des BoF comparées aux performances des BoF standards "Std". La différence entre l'adaptation et la technique standard est calculée dans la colonne "Diff". La moyenne calculée ne tient pas compte du rejet et n'est pas pondérée par le nombre de documents contenus dans chaque classe.

		Rappel			Précision		
Modèles	Nb docs	Adapt	Std	Diff	Adapt	Std	Diff
Lot	670	0,9985	0,9955	+0,0029	0,9780	0,9794	-0,0013
ASF	29	0,9354	0,8966	+0,0389	1	0,9286	+0,0714
IBIS	41	0,8536	0,8537	+0,0000	0,9722	0,3182	+0,6540
SNCF	194	0,9948	0,9794	+0,0154	1	0,9005	+0,0995
Flunch	4	1	1	+0,0000	0,8000	0,3333	+0,4666
RATP	110	0,4727	1	-0,5272	1	0,4089	+0,5910
Buffalo	13	1	1	+0,0000	1	0,1970	+0,8030
Quick	12	0,8333	1	-0,1666	1	0,0833	+0,9166
SANEF	21	0,8636	1	-0,1363	0,8260	0,4118	+0,4143
ElsUk	430	0,9813	0,9907	-0,0093	1	0,9552	+0,0448
ElsEs	299	1	1	+0,0000	0,9739	0,9492	+0,0247
ElsFr	99	1	0,6566	+0,3434	1	0,9559	+0,0441
Rejet	3847	0,9946	0,8724	+0,1222	0,9815	0,9964	-0,0149
Moyenne		0,8981	0,9477	-0,0496	0,9789	0,6184	+0,3605

Les résultats de cette méthode sont très encourageants car une très forte augmentation de la précision par rapport à l'utilisation des BoF est observée. En effet, une précision moyenne de 0,9789 est obtenue sur l'ensemble des 12 classes. Contrairement à l'application des BoF standards, l'ensemble des classes ont une bonne précision, il y a peu de confusion entre la classe de rejet et les autres classes. Cependant, une telle hausse de la précision implique une très légère diminution du rappel pour certaines classes. L'amélioration de la précision est obtenue car les points d'intérêts ne sont plus partitionnés, les caractéristiques sont alors bien plus sélectives. Mais puisqu'elles sont moins approximatives, ceci tend à faire baisser le rappel. La classe la plus touchée par ce "phénomène" est la classe "RATP", dont le rappel a largement chuté. Ceci vient du fait que cette classe contient les images les plus petites et les plus dégradées de la base. Un grand nombre de tickets "RATP" ne sont plus trouvés mais l'ensemble des documents labellisés "SNCF" font effectivement parti de cette classe.

4.6 Conclusion et perspective

Dans ce chapitre, nous avons montré que les BoF habituellement utilisés dans le domaine des images naturelles peuvent être également utilisés pour les images de documents. Les performances de la technique "classique" avec partitionnement et apprentissage par SVM donnent des résultats corrects. Sur notre base de documents, les résultats obtenus sont du même ordre de grandeur que ceux obtenus en utilisant les BoW.

Cependant, la variante des BoF sans partitionnement et avec une apprentissage par PPV donne des performances en terme de précision qui sont bien supérieurs au BoF standards. Les résultats obtenus sont très encourageants pour effectuer la classification d'images de documents.

À noter tout de même que si les documents sont semi-structurés et que le système doit avoir la meilleur précision possible, la méthode du chapitre 3 permet d'obtenir de meilleurs résultats car elle est optimisée pour ce genre de documents.

Dans ce chapitre, nous avons montré que les performances obtenues par analyse du texte et de l'image sont du même ordre de grandeur, mais chaque méthode est plus ou moins efficace selon les types de documents. C'est un résultat important car il laisse envisager que la combinaison intelligente du texte et de l'image permettrait d'améliorer encore plus les performances de classification d'images de documents. En effet, lorsque les documents contiennent une majorité de texte et que la qualité du document est bonne (afin que les performance de l'OCR soit optimales), alors les techniques telles que les BoW, basées sur l'analyse du texte peuvent être privilégiées. Les techniques basées sur l'image quant-à-elles peuvent être utilisées dans la plupart des cas, car les méthodes basées sur les points d'intérêt sont robustes au bruit, au changement d'échelle, à la rotation, etc. ce qui permet de garantir un bon résultat même lorsque la qualité des images de documents est mauvaise. Cependant si les images sont trop différentes ou trop similaires, il risque d'y avoir rejet ou confusion. La principale perspective réside dans la combinaison des descripteurs texte et image pour l'amélioration de la classification de documents.

Conclusion et perspectives

Conclusion

Les travaux de recherche détaillés dans ce manuscrit ont permis de mettre en évidence qu'il n'existe pas une unique façon de classer et reconnaitre des documents, mais plusieurs. Nous proposons ainsi d'explorer trois approches distinctes pour lesquelles l'utilisateur est à chaque fois au cœur du système mais, selon les cas de figure, ce dernier dispose d'informations plus ou moins variées et plus ou moins précises des documents qu'il doit classer.

Notre première proposition permet à un utilisateur n'ayant aucune information préalable sur la nature des documents (nombre de classes, qualité de la numérisation, variabilité des éléments au sein d'une même classe, ...) d'explorer le contenu puis de trier rapidement l'ensemble d'une base d'images de documents. Là où la majorité des méthodes proposées dans la littérature cherchent à trier automatiquement des images de documents sur la base d'une phase d'apprentissage très lourde, supposant qu'un utilisateur dispose de connaissances métiers précises, nous proposons une méthode originale reposant sur sur un ensemble d'interactions entre le système et l'utilisateur. Ces interactions permettent d'accélérer la tâche manuelle de classification. Le système présenté ne prend donc pas de décision à la place de l'utilisateur mais lui propose des regroupements d'images supposées être de la même classe. Ces regroupements sont réalisés en étudiant les similarités visuelles des images de documents. Plus précisément, puisque le nombre de classes n'est pas connu, la première étape consiste à faire une estimation de ce nombre. Pour cela nous effectuons plusieurs partitionnement puis comparons leurs qualités en utilisant le critère de silhouette moyenne. Nous souhaitons privilégier ce critère, car il favorise un partitionnement ou les distances intra-classe sont faibles et les distances inter-classes sont grandes. Une fois le nombre de classe estimé et le partitionnement fait, chaque classe est représentée par son médoïde (son image centrale). Le système se comporte alors comme un système de requête par l'exemple où les requêtes sont les médoïdes. On présente à l'utilisateur le médoïde et les 50 images qui lui sont le plus similaire. Un retour de pertinence est effectué en fonction des interactions de l'utilisateur avec le système. Ce dernier peut sélectionner des images si elles ne semblent pas correspondre à la même classe. Si plusieurs images sont sélectionnées (ce qui signifie qu'un nombre significatif d'images non pertinentes lui ont été montrées), une sélection de caractéristiques est appliquée à l'ensemble des images, les distances sont alors mises à jours, l'ordre des images change et les nouvelles 50 images les plus similaires sont affichées. Si trop d'images sont sélectionnées comme étant mauvaises, cela signifie que le système n'est plus capable de regrouper de manière pertinente les images qu'il reste à trier (ou que toutes ont été triées). On passe alors à la classification d'une autre classe en utilisant le médoïde suivant. Une fois que toutes les classes ont été traitées, s'il reste des documents non classés, on réitère l'ensemble du processus (estimation du nombre de classes, partitionnement, requête par l'exemple et classification assistée.) Grâce à cette méthode, le temps de classification manuelle des images de document est divisé par 3,4.

Notre seconde proposition, concerne la recherche d'image de documents. Ceci répond au cas de figure où l'utilisateur souhaite retrouver des exemplaires d'un document précis. Il est donc capable de fournir un exemple de l'image de document qu'il souhaite retrouver. Pour répondre à ce besoin, nous avons choisi de nous inspirer des techniques de reconnaissance d'objets habituellement utilisées dans le domaine des images naturelles. L'utilisation de techniques de recherche non-exactes basées sur l'utilisation de points d'intérêt permet de retrouver les images avec une très bonne précision. La méthode proposée se base sur l'extraction et la description de points d'intérêts ayant la propriété d'être robustes aux rotations, translations, changement d'échelle et de luminosité. Ceci nous permet non seulement de pouvoir mettre en place une méthode de comparaison "non exacte" d'images de documents, mais aussi de pouvoir analyser et comparer des images contenant des bruits divers liés à leur cycle de vie (impression, dégradation physique, re-numérisation, ...). Notre proposition s'articule sur les 3 grandes étapes suivantes. Tout d'abord, une extraction de points d'intérêt sur l'image d'exemple (fournie par l'utilisateur) ainsi que sur l'ensemble de la base d'images est réalisée. Ces points d'intérêt de l'image d'exemple sont mis en correspondance avec ceux de l'image à analyser. On cherche à valider une transformation géométrique rigide dans le plan afin de valider ou non la présence du document exemple sur l'image à analyser. Ces deux dernières étapes sont réitérées sur l'ensemble des images de la base. En moyenne, une précision de 0,989 et un rappel de 0,928 sont obtenus sur une base de 2155 documents, pour la recherche de 7 classes de documents. La précision n'est pas tout à fait optimale dans les cas où une portion d'image commune avec le document d'exemple est présente sur un document d'une classe différente.

Enfin, nous proposons une méthode correspondant à un cas de figure où l'utilisateur peut donner en entrée du système différents éléments de plusieurs classes qu'ils cherche à labelliser. Dans ce cas, nous avons privilégié l'utilisation de techniques d'apprentissage supervisée. La majorité des méthodes permettant de retrouver un document dans une base repose sur une analyse et une comparaison des textes contenus dans les images de documents (méthodes à base de sacs de mots). Récemment, les sacs de mots visuels (inspirés directement des techniques de sacs de mots) ont été utilisés avec succès pour la classification et la reconnaissance d'images naturelles. Nous montrons dans le chapitre de ce manuscrit que ces techniques peuvent également être utilisées pour la classification d'images de documents et que les performances obtenues sur notre base sont du même ordre de grandeur que celles obtenues en utilisant uniquement le texte. Les évolutions les plus récentes suppriment le partitionnement des points d'intérêt et sont basées sur les k-PPV plutôt que les SVM. Nous avons notamment testé et adapté une technique originalement utilisée pour la reconnaissance de logos dans les images de documents afin de reconnaitre des images des documents "complètes" (et plus seulement une portion). Cette technique permet d'améliorer les résultats obtenus par les sacs de mots visuels. Une précision moyenne de 0,9789 et un rappel moyen de 0,8981 est obtenue pour la classification de 12 classes représentant 1922 documents contenus dans une base de 5769 documents, en utilisant 5 exemples d'apprentissage par classe et 149 images d'apprentissage pour le rejet.

Perspectives

Dans cette thèse, nous avons proposé différentes méthodes de classification d'images de documents relatives à la quantité d'informations qu'est en mesure de donner un utilisateur (aucune information, un exemple par classe, plusieurs exemples par classe). Si les méthodes détaillées dans ce manuscrit couvrent avec efficacité un large spectre de documents diffé-

rents récupérés en production, nous avons cependant observé que pour certaines classes particulières de documents et certains objectifs spécifiques de l'utilisateur, les résultats de classification peuvent être insuffisants.

Plus précisément, la qualité des résultats chute dès lors que l'utilisateur souhaite retrouver un document dont lui seul est capable d'exprimer les règles. Par exemple, si un utilisateur souhaite reconnaitre des documents dont seul une partie spécifique est importante, il faut être en mesure de lui permettre d'exprimer comment réaliser un tel objectif. Si l'on souhaite que l'apprentissage soit guidé par l'utilisateur, il faut fournir à ce dernier des outils afin qu'il puisse définir des règles ou transmettre ses connaissances sous diverses formes au système de recherche.

Il existe dans la littérature des systèmes permettant à un utilisateur d'exprimer ses connaissances métiers. Nous allons présenter deux de ces outils afin de positionner notre perspective plus précisément. Ces deux systèmes sont : DMOS et Agora.

DMOS est une méthode générique de reconnaissance de documents. Cette méthode a été appliquée à l'analyse de la structure de tableaux [Coü06], à la reconnaissance de partitions musicales [CC94] ou de formules mathématiques [Coü01]. La méthode DMOS (Description and Modification of Segmentation) est composée d'un langage appelé EPF (Enhanced Position Formalism) permettant de formuler la description de documents. L'avantage principal de la méthode est quelle permet de formuler des descriptions plus ou moins spécifiques en fonction de la tâche à exécuter. Cependant, l'inconvénient est la prise en main et la complexité du langage EPF qui est proche de celle d'un langage de développement et qui n'est pas facilement abordable pour un utilisateur non expert en informatique.

Agora [RBD06] est un logiciel permettant de définir des scénarios de recherche de manière interactive. Un ensemble d'interfaces permet à l'utilisateur de construire des scénarios d'analyse. Un scénario peut être composé de règles de détection de blocs (texte, graphique, etc.) et de seuils permettant la fusion ou la séparation de blocs. Les actions de l'utilisateur avec le logiciel sont appliquées en temps réel ce qui lui permet de visualiser les résultats et de valider son scénario. L'utilisateur peut a tout moment supprimer, ajouter ou modifier des règles afin d'affiner les résultats. Une fois que le scénario est considéré comme bon, il peut être sauvegardé afin de l'appliquer sur une base de documents plus conséquente. Le principale avantage d'Agora est que le logiciel peut être utilisé par des non-informaticiens. Nous pensons également qu'il est important de pouvoir fournir à l'utilisateur des interfaces lui permettant d'exprimer simplement ses besoins et ses connaissances.

Nous proposons une perspective allant dans le sens d'Agora. Cependant nous souhaitons pouvoir mettre à la disposition de l'utilisateur un ensemble d'outils génériques permettant la classification ou la reconnaissance d'images de documents en fonction des connaissances données par l'utilisateur. La chaîne de traitements proposée sur la figure 4.11 pour la classification d'images de documents repose sur 3 parties principales :

- L'introduction de connaissances par l'utilisateur sous différentes formes.
- L'analyse et l'extraction de caractéristiques.
- Un moteur d'interprétation permettant de prendre des décisions.

L'analyse et l'extraction de caractéristiques sont partiellement guidées par les connaissances de l'utilisateur dans la chaîne de traitements. En effet, le chapitre 2 et 3 ont montré la complexité d'avoir un système à la fois automatique et performant. Il semble à la fois utopique et dangereux d'imaginer qu'un système soit capable de prendre des décisions sans avoir aucune connaissance sur la tâche à effectuer et sur la nature des documents. De plus, les informations présentes sur un document sont très nombreuses. Il est donc complexe de sélectionner les caractéristiques pertinentes parmi la masse d'informations contenue dans

le document. Pour classifier un document, il faudra parfois regarder la simple présence d'un logo, d'autres fois ça sera la mise en page du document, tandis que d'autres fois encore, la présence d'un mot ou d'une expression régulière fera la différence, etc. On peut également imaginer que pour des prestations différentes, les mêmes documents puissent appartenir à des classes différentes, ce que la machine ne peux deviner. Ces règles sont connues de l'utilisateur, il faut donc qu'elles soient communiquées d'une quelconque manière au système. Comme évoqué par [Coü12], nous envisageons dans notre système de pouvoir introduire 3 formes de connaissances différentes : les connaissances *a priori*, externes et internes. Elles sont définies comme suit :

- Les connaissances *a priori* sont liées à un type de document ou à une prestation comprenant certains genres de documents. Le format des documents (taille, couleur,...) ou la langue du texte sont très souvent des connaissances *a priori*.
- Les connaissances internes à la page sont des informations présentes dans l'image que l'utilisateur regarde afin de différencier deux documents. La présence d'un tableau, d'un logo ou d'un mot particulier sont des exemples de connaissances internes.
- Les connaissances externes à la page sont des connaissances qui ne sont pas contenues dans l'image elle-même. Il est possible que des informations contenue dans les images précédentes ou suivantes aide à reconnaitre l'image courante.

Nous pensons donc que l'ensemble de ces connaissances permettra de guider les algorithmes afin de savoir quelles informations il faut analyser dans le document afin de le classifier.

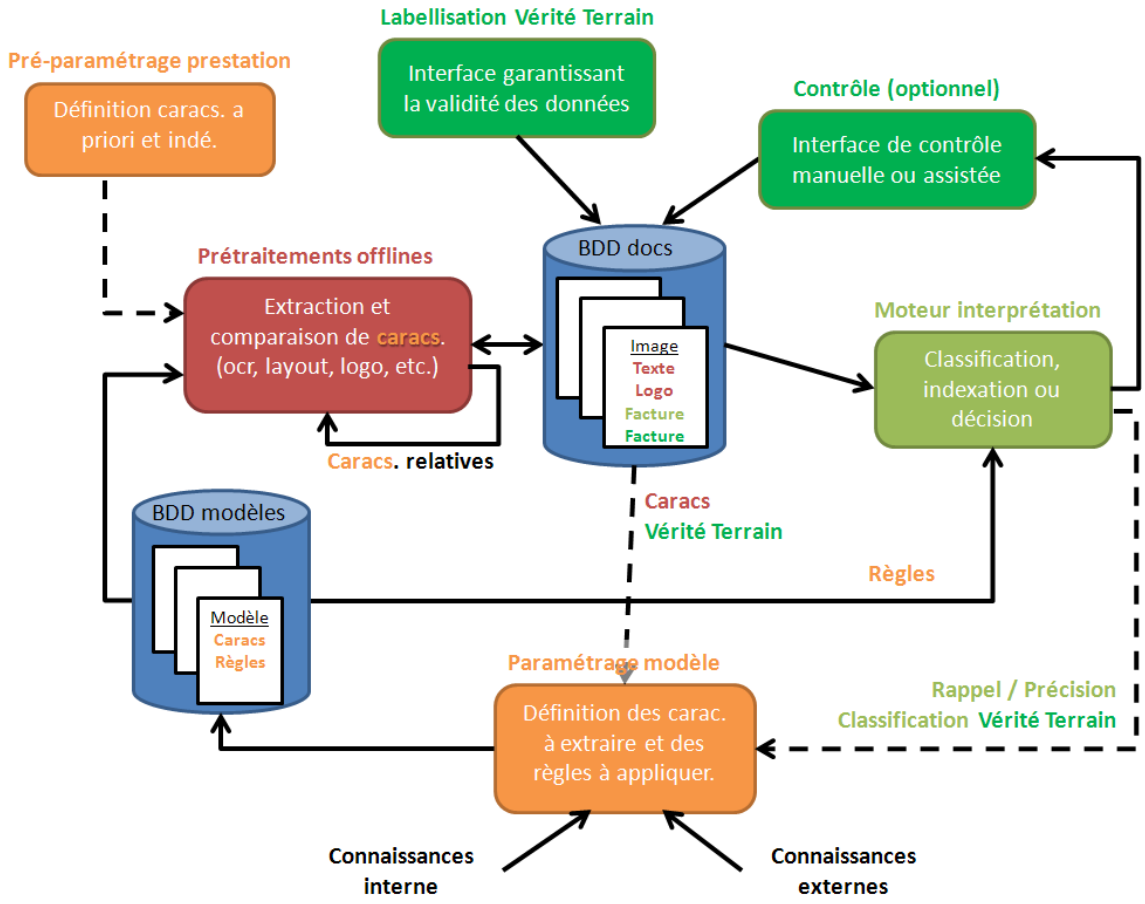


FIGURE 4.11 – Chaîne de traitement pour la classification.

Nous avons développé un prototype dans lequel il est possible pour un utilisateur de définir tout un ensemble de règles lui permettant d'exprimer au système ce qu'il cherche à extraire. Voici, dans les grandes lignes, ce que nous mettons à disposition de l'utilisateur. Tout d'abord, le programme demande à l'utilisateur s'il peut exprimer des connaissances *a priori* sur les documents qu'il souhaite reconnaitre. Par exemple, il peut renseigner si les documents ont des signes distinctifs "évidents" tels que : le format (A4, A5, etc.) , le sens de lecture (portrait / paysage), la langue du texte (français, anglais, etc.). La liste grandira en fonction des retours de l'utilisation du programme.

Ensuite, lors d'une phase de pré-traitements hors ligne, un grand nombre de caractéristiques sont extraites des images. Sur chaque image de document, 3 grands types de descripteurs sont extraits : le texte, la mise en page et l'image. Pour le texte, l'ensemble des mots, de leur position, police d'écriture et toutes les informations typographiques sont stockés. Pour la mise en page, la position et la nature (texte, tableau, image) des blocs sont sauvegardées. Pour l'image, les points d'intérêt et les densités de pixels sont extraits pour 3 résolutions (image originale, image redimensionnée à 60% et à 20% de sa taille originale). Ces 3 niveaux permettent de capturer l'information de manière plus ou moins détaillée. Les différents niveaux seront utilisés en fonction de la demande de l'utilisateur.

Enfin, le logiciel propose des caractéristiques plus haut-niveau que celles extraites lors de la phase de pré-traitement. Ces caractéristiques sont distribuées en 3 familles :

- texte : recherche de mot sur les zones textuelles contenues dans une image, recherche du contenu de paragraphes similaires, d'expressions régulières, de typographies particulières, etc.
- mise en page : recherche de la présence d'un tableau, image ou texte à une position fixe ou relative et de taille fixe ou variable, etc.
- image : comparaison globale de l'image, détection de logo, word-spotting, etc.

L'utilisateur peux également utiliser les outils présentés dans cette thèse, lui permettant de faire une recherche précise par l'image ou une classification supervisée par le texte et l'image en fournissant plusieurs exemples. Les règles qui lui sont proposées sont, pour l'instant, un simple ensemble de "ET", "OU" et "NON" logiques entre les caractéristiques.

Le prototype décrit dans ces perspectives est actuellement en cours de refactorisation afin de simplifier sa maintenance et de le rendre plus générique. Il sera dans les prochains mois testé en production et représente donc la perspective principale des travaux présentés dans ce manuscrit.

Glossaire

ACP	Analyse en Composantes Principales, 39, 43
BE	Boites Englobantes, 47
BoF	Bag of Features, 89
BoW	Bag of Words, 89
CAH	Classification Ascendante Hiérarchique, 48, 50
CBIR	Content Based Image Retrieval, 29, 46, 52, 53, 56, 61, 65
CC	Composantes Connexes, 45, 47
CESR	Centre d'Études Supérieures de la Renaissance, 21
DoG	Difference of Gaussians, 68
DPI	Dots Per Inch (points par pouces), 14
EM	Expectation Maximization, 51
EPF	Enhanced Position Formalism, 34, 109
FLANN	Fast Library for Approximate Nearest Neighbors, 67
FTP	File Transfer Protocol, 14
GED	Gestion Electronique de Documents, 15
ICR	Intelligent Character Recognition, 16
LAD	Lecture Automatique de Document, 16, 24, 26, 63
MLP	Multi Layer Perceptron, 39
MSE	Mean Squared Error, 50
OCR	Optical Character Recognition, 13, 16, 20, 25, 29, 34, 38, 41, 46, 65, 92
PAM	Partitioning Around Medoids, 52, 55
PPV	Plus Proches Voisins, 39, 66, 69, 89, 108

RAD	Reconnaissance Automatique de Document, 16, 25, 26
RANSAC	RANdom SAmple Consensus, 67, 71
RLSA	Run Length Smoothing Algorithm, 43
SDK	Software development kit, 65
SIFT	Scale-invariant feature transform, 64, 94
SURF	Speeded Up Robust Features, 64, 94
SVM	Support vector machine, 38, 39, 89, 108
VRS	Virtual ReScan, 15

Bibliographie

[AAF+07] G. Agam, S. Argamon, O. Frieder, D. Grossman et D. Lewis : Content-based document image retrieval in complex document collections. *In Document Recognition and Retrieval XIV (Part of the IS&T/SPIE Electronic Imaging Symposium)*, volume 6500, pages 65000S1–65000S12. SPIE, 2007.

[abb] Ocr, icr, omr and linguistic software. http://www.abbyy.com.

[ACPP11] A. Antonacopoulos, C. Clausner, C. Papadopoulos et S. Pletschacher : Historical document layout analysis competition. *In Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1516–1520. IEEE, 2011.

[AD09] M. Agrawal et D. Doermann : Voronoi++ : A Dynamic Page Segmentation approach based on Voronoi and Docstrum features. *In Document Analysis and Recognition (ICDAR), 2009 International Conference on*, pages 1011–1015. IEEE, 2009.

[AF00] A. Amin et S. Fischer : A document skew detection method using the hough transform. *Pattern Analysis & Applications*, 3(3):243–253, 2000.

[AJD11a] O. Augereau, N. Journet et J.-P. Domenger : Classification d'images de documents avec retour de pertinence : Application aux documents de type ressources humaines. *In 23ème Colloque sur le traitement du signal et des images*. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 2011.

[AJD11b] O. Augereau, N. Journet et J.-P. Domenger : Document images indexing with relevance feedback : an application to industrial context. *In Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1190–1194. IEEE, 2011.

[AJD12] O. Augereau, N. Journet et J.-P. Domenger : Reconnaissance et extraction de pièces d'identité. *In Actes du Douzième Colloque International Francophone sur l'Écrit et le Document (CIFED)*, pages 179–194, 2012.

[AKTS08] H. Al-Khaffaf, A.Z. Talib et R.A. Salam : Removing salt-and-pepper noise from binary images of engineering drawings. *In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[Ant98] A. Antonacopoulos : Page Segmentation Using the Description of the Background. *Computer Vision and Image Understanding*, 70(3):350–369, 1998.

[AOC+01] S. Adam, JM Ogier, C. Cariou, R. Mullot, J. Gardes et Y. Lecourtier : Utilisation de la transformée de fourier-mellin pour la reconnaissance de formes multi-orientées et multi-échelles : application à l'analyse automatique de documents techniques. *Traitement du signal*, 18(1):17, 2001.

[APBP09] A. ANTONACOPOULOS, S. PLETSCHACHER, D. BRIDSON et C. PAPADOPOULOS : ICDAR 2009 page segmentation competition. *In Document Analysis and Recognition (ICDAR), 2009 International Conference on*, pages 1370–1374. IEEE, 2009.

[APCU09] J. ARLANDIS, J.C. PEREZ-CORTES et E. UNGRIA : Identification of very similar filled-in forms with a reject option. *In Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, pages 246–250. IEEE, 2009.

[AWY+99] C.C. AGGARWAL, J.L. WOLF, P.S. YU, C. PROCOPIUC et J.S. PARK : Fast algorithms for projected clustering. *ACM SIGMOD Record*, 28(2):61–72, 1999.

[AY00] C.C. AGGARWAL et P.S. YU : Finding generalized projected clusters in high dimensional spaces. *ACM SIGMOD Record*, 29(2):81, 2000.

[BA04] R. BEKKERMAN et J. ALLAN : Using bigrams in text categorization. *Department of Computer Science, University of Massachusetts, Amherst*, 1003, 2004.

[BAA+10] S.S. BUKHARI, A. AZAWI, M.I. ALI, F. SHAFAIT et T.M. BREUEL : Document image segmentation using discriminative learning over connected components. *In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 183–190. ACM, 2010.

[Bai92] H.S. BAIRD : Background structure in document images. *In Advances in Structural and Syntactic Pattern Recognition*, pages 253–269, 1992.

[BDMS10] A. BARTOLI, G. DAVANZO, E. MEDVET et E. SORIO : Improving features extraction for supervised invoice classification. *In Artificial Intelligence and Applications*. ACTA Press, 2010.

[BETVG08] H. BAY, A. ESS, T. TUYTELAARS et L. VAN GOOL : Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[BKA06] M. BENJLAIEL, S. KANOUN et A. ALIMI : Une méthode de segmentation d'Images de Documents Composites. *In Actes du Neuvième Colloque International Francophone sur l'Écrit et le Document (CIFED)*, pages 121–126, 2006.

[BL07] M. BROWN et D.G. LOWE : Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[BMAC07] H.S. BAIRD, M.A. MOLL, C. AN et M.R. CASEY : Document image content inventories. *In Document Recognition and Retrieval XIV (Part of the IS&T/SPIE Electronic Imaging Symposium)*, pages 65000X1 – 65000X12. SPIE, 2007.

[BMD09] C. BOUTSIDIS, M.W. MAHONEY et P. DRINEAS : Unsupervised feature selection for the k-means clustering problem. *Advances in Neural Information Processing Systems*, 22:153–161, 2009.

[Bre02a] T.M. BREUEL : Robust least square baseline finding using a branch and bound algorithm. *In Document Recognition and Retrieval VIII (Part of the IS&T/SPIE Electronic Imaging Symposium)*, volume 4670, pages 20–27. SPIE, 2002.

[Bre02b] T.M. BREUEL : Two geometric algorithms for layout analysis. *Lecture Notes in Computer Science*, pages 188–199, 2002.

[Bre08] T.M. BREUEL : The ocropus open source ocr system. *In Document Recognition and Retrieval XV (Part of the IS&T/SPIE Electronic Imaging Symposium)*, volume 6815, pages 68150F1–68150F15. SPIE, 2008.

[BSI08] O. BOIMAN, E. SHECHTMAN et M. IRANI : In defense of nearest-neighbor based image classification. *In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[BW03] A.D. BAGDANOV et M. WORRING : First order Gaussian graphs for efficient structure classification. *Pattern Recognition*, 36(6):1311–1324, 2003.

[Can06] L. CANDILLIER : *Contextualisation, visualisation et évaluation en apprentissage non supervisé*. Thèse de doctorat, Université Charles de Gaulle, Lille 3. Chinchor, N.-A.(1997). Overview of MUC-7/MET-2., 2006.

[Cao08] F. CAO : *A theory of shape identification*, volume 1948. Springer Verlag, 2008.

[CB07] N. CHEN et D. BLOSTEIN : A survey of document image classification : problem statement, classifier architecture and performance evaluation. *International Journal on Document Analysis and Recognition*, 10(1):1–16, 2007.

[CC94] B. COÜASNON et J. CAMILLERAPP : Using grammars to segment and recognize music scores. *In International Association for Pattern Recognition Workshop on Document Analysis Systems*, pages 15–27, 1994.

[CCMM98] R. CATTONI, T. COIANIZ, S. MESSELODI et CM MODENA : Geometric layout analysis techniques for document image understanding : a review. *ITC-IRST Technical Report*, 9703(09), 1998.

[CDF+04] G. CSURKA, C. DANCE, L. FAN, J. WILLAMOWSKI et C. BRAY : Visual categorization with bags of keypoints. *In Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22, 2004.

[CGMS99] F. CESARINI, M. GORI, S. MARINAI et G. SODA : Structured document segmentation and representation by the modified XY tree. *In Document Analysis and Recognition (ICDAR), 1999 International Conference on*, pages 563–566, 1999.

[CHSZ08] V. CHAOJI, M.A. HASAN, S. SALEM et M.J. ZAKI : Sparcl : Efficient and effective shape-based clustering. *In Eighth IEEE International Conference on Data Mining, 2008. ICDM'08*, pages 93–102, 2008.

[CK11] Y.Y. CHIANG et C.A. KNOBLOCK : Recognition of multi-oriented, multi-sized, and curved text. *In Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1399–1403. IEEE, 2011.

[CL03] B. COÜASNON et I. LEPLUMEY : A generic recognition system for making archives documents accessible to public. *In Document Analysis and Recognition (ICDAR), 2003 International Conference on*, pages 228–232. IEEE, 2003.

[CLMS01] F. CESARINI, M. LASTRI, S. MARINAI et G. SODA : Encoding of modified XY trees for document classification. *In Document Analysis and Recognition (ICDAR), 2001 International Conference on*, page 1131. Published by the IEEE Computer Society, 2001.

[Cor02] A. CORNUÉJOLS : Une nouvelle méthode d'apprentissage : Les SVM. Séparateurs à vaste marge. *Bulletin de l'AFIA*, 51:14–23, 2002.

[Coü01] B. COÜASNON : Dmos : A generic document recognition method, application to an automatic generator of musical scores, mathematical formulae and table structures recognition systems. *In Document Analysis and Recognition (ICDAR), 2001 International Conference on*, pages 215–220. IEEE, 2001.

[Coü06] B. COÜASNON : Dmos, a generic document recognition method : application to table structure analysis in a general and in a specific way. *International Journal on Document Analysis and Recognition*, 8(2):111–122, 2006.

[Coü12] B. COÜASNON : Fusion des connaissances en analyse de documents. *In Actes du Douzième Colloque International Francophone sur l'Écrit et le Document (CIFED)*, pages 3–3, 2012.

[CPA11] C. CLAUSNER, S. PLETSCHACHER et A. ANTONACOPOULOS : Scenario driven in-depth performance evaluation of document layout analysis methods. *In Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1404–1408. IEEE, 2011.

[CS06] E.C.J. CORRY et E.A. SWAIN : Optical character recognition (ocr) testing : British geological survey report ir/06/066. 2006.

[DA02] P. DUYGULU et V. ATALAY : A hierarchical representation of form documents for identification and retrieval. *International Journal on Document Analysis and Recognition*, 5(1):17–27, 2002.

[DFG03] M. DILIGENTI, P. FRASCONI et M. GORI : Hidden tree Markov models for document image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:519–523, 2003.

[DKN08] T. DESELAERS, D. KEYSERS et H. NEY : Features for image retrieval : an experimental comparison. *Information Retrieval*, 11(2):77–107, 2008.

[DLR77] A.P. DEMPSTER, N.M. LAIRD et D.B. RUBIN : Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[DLW05] R. DATTA, J. LI et J.Z. WANG : Content-based image retrieval : approaches and trends of the new age. *In Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval, November*, pages 10–11. ACM, 2005.

[DM08] H. DÉJEAN et J.L. MEUNIER : Versatile page numbering analysis. *In Document Recognition and Retrieval XV (Part of the IS&T/SPIE Electronic Imaging Symposium)*, volume 6815, pages 68150K1–68150K9. SPIE, 2008.

[DPGM04] C. DOMENICONI, D. PAPADOPOULOS, D. GUNOPULOS et S. MA : Subspace clustering of high dimensional data. *In Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 517–521. SIAM, 2004.

[EHJT04] B. EFRON, T. HASTIE, I. JOHNSTONE et R. TIBSHIRANI : Least angle regression. *Annals of statistics*, 32(2):407–451, 2004.

[EKSX96] M. ESTER, H.P. KRIEGEL, J. SANDER et X. XU : A density-based algorithm for discovering clusters in large spatial databases with noise. *In Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, volume 96, pages 226–231. AAAI Press, 1996.

[FB81] M.A. FISCHLER et R.C. BOLLES : Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FFFP07] L. FEI-FEI, R. FERGUS et P. PERONA : Learning generative visual models from few training examples : An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[FHD90] JL FISHER, SC HINDS et DP D'AMATO : A rule-based system for document image segmentation. *In Pattern Recognition (ICPR), 1990 International Conference on*, volume 1, pages 567–572, 1990.

[FR06] C. FRALEY et A.E. RAFTERY : MCLUST version 3 for R : Normal mixture modeling and model-based clustering. Rapport technique, Department of Statistics, University of Washington, 2006.

[GAA+01] N. GORSKI, V. ANISIMOV, E. AUGUSTIN, O. BARET et S. MAXIMOV : Industrial bank check processing : the a2ia checkreader tm. *International Journal on Document Analysis and Recognition*, 3(4):196–206, 2001.

[Gac09] D. GACEB : *Contributions au tri automatique de documents et de courrier d'entreprises*. Thèse de doctorat en informatique, Institut National de Sciences Appliquées de Lyon, octobre 2009.

[GHP07] G. GRIFFIN, A. HOLUB et P. PERONA : Caltech-256 object category dataset. 2007.

[GRS98] S. GUHA, R. RASTOGI et K. SHIM : Cure : an efficient clustering algorithm for large databases. *In SIGMOD '98 : Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 73–84. ACM, 1998.

[GRS99] S. GUHA, R. RASTOGI et K. SHIM : Rock : A robust clustering algorithm for categorical attributes. *In International Conference on Data Engineering (ICDE)*, pages 512–521. IEEE, 1999.

[GV09] A. GORDO et E. VALVENY : A rotation invariant page layout descriptor for document classification and retrieval. *In Document Analysis and Recognition (ICDAR), 2009 International Conference on*, pages 481–485. IEEE Computer Society, 2009.

[HBBC08a] H. HAMZA, Y. BELAÏD, A. BELAÏD et B.B. CHAUDHURI : An end-to-end administrative document analysis system. *In Document Analysis Systems (DAS), 2008 The Eighth IAPR International Workshop on*, pages 175–182. IEEE, 2008.

[HBBC08b] H. HAMZA, Y. BELAÏD, A. BELAÏD et B.B. CHAUDHURI : Incremental classification of invoice documents. *In Pattern Recognition (ICPR), 2008 19th International Conference on*, pages 1–4. IEEE, 2008.

[HDRT98] P. HÉROUX, S. DIANA, A. RIBERT et E. TRUPIN : Classification method study for automatic form class identification. *In Pattern Recognition, 1998 Fourteenth International Conference on*, volume 1, 1998.

[HEG+07] J.J. HULL, B. EROL, J. GRAHAM, Q. KE, H. KISHI, J. MORALEDA et D.G. VAN OLST : Paper-based augmented reality. *In Artificial Reality and Telexistence, 17th International Conference on*, pages 205–209. Ieee, 2007.

[HJS09] H. HARZALLAH, F. JURIE et C. SCHMID : Combining efficient object localization and image classification. *In Computer Vision, 2009 IEEE 12th International Conference on*, pages 237–244. IEEE, 2009.

[HK98] A. HINNEBURG et D.A. KEIM : An efficient approach to clustering in large multimedia databases with noise. *Knowledge Discovery and Data Mining*, 5865, 1998.

[HKYJ07] E. HAN, K. KIM, H.K. YANG et K. JUNG : Frame Segmentation Used MLP-Based XY Recursive for Mobile Cartoon Content. *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, pages 872–881, 2007.

[HL09] RM HAMOU et A. LEHIRECHE : La classification non supervisée (clustering) de documents textuels par les automates cellulaires. *In International Conference on Information Technology and its Applications (CIIA-09), Saïda/Algeria*, pages 1613–0073, 2009.

[HS88] C. HARRIS et M. STEPHENS : A combined corner and edge detector. *In Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[Hul98] J.J. HULL : Document image similarity and equivalence detection. *International Journal on Document Analysis and Recognition*, 1(1):37–42, 1998.

[HVB00] M. HALKIDI, M. VAZIRGIANNIS et Y. BATISTAKIS : Quality scheme assessment in the clustering process. *Principles of Data Mining and Knowledge Discovery*, pages 265–276, 2000.

[JB92] A.K. JAIN et S. BHATTACHARJEE : Text segmentation using Gabor filters for automatic document processing. *Machine Vision and Applications*, 5(3):169–184, 1992.

[JED+05] W. JIANG, G. ER, Q. DAI, L. ZHONG et Y. HOU : Relevance feedback learning with feature selection in region-based image retrieval. *In Proc. of IEEE Int. Conf. Acoustics, Speech and Signal Processing*, volume 2, pages 509–512, 2005.

[JG09] L. JUAN et O. GWUN : A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.

[JME+08] N. JOURNET, R. MULLOT, V. EGLIN, J.Y. RAMEL *et al.* : Analyse d'images de documents anciens : une approche texture. *traitement du signal*, 24(6):461–479, 2008.

[JMF99] A.K. JAIN, M.N. MURTY et P.J. FLYNN : Data clustering : a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[Joa98] T. JOACHIMS : Text categorization with support vector machines : Learning with many relevant features. *Machine learning : ECML-98*, pages 137–142, 1998.

[JZ96] A.K. JAIN et Y. ZHONG : Page segmentation using texture analysis. *Pattern Recognition*, 29(5):743–770, 1996.

[KHP93] T. KANUNGO, R.M. HARALICK et I. PHILLIPS : Global and local document degradation models. *In Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 730–734. IEEE, 1993.

[KKR07] T. KASAR, J. KUMAR et AG RAMAKRISHNAN : Font and background color independent text binarization. *In Second International Workshop on Camera-Based Document Analysis and Recognition*, pages 3–9, 2007.

[Ko12] Y. KO : A study of term weighting schemes using class information for text classification. *In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1029–1030. ACM, 2012.

[KR90] L. KAUFMAN et PJ ROUSSEEUW : Finding groups in data : an introduction to cluster analysis. *NY John Wiley & Sons*, 1990.

[KR10] M.B. KURSA et W.R. RUDNICKI : Feature selection with the boruta package. *Journal of Statistical Software*, 36(11):1–13, 2010.

[KS04] Y. KE et R. SUKTHANKAR : Pca-sift : A more distinctive representation for local image descriptors. 2004.

[KSB07] D. KEYSERS, F. SHAFAIT et T.M. BREUEL : Document image zone classification - a simple high-performance approach. *In 2nd Int. Conf. on Computer Vision Theory and Applications*, pages 44–51, 2007.

[KSI98] K. KISE, A. SATO et M. IWATA : Segmentation of page images using the area Voronoi diagram. *Computer Vision and Image Understanding*, 70(3):370–382, 1998.

[KSK11] M.D.K. KUMAR, K. SUNEERA et P.V.C. KUMAR : Content Based Image Retrieval-Extraction by Objects of User Interest. *International Journal on Computer Science and Engineering*, 3(3):1068–1074, 2011.

[LBH+09] S. LAROUM, N. BÉCHET, H. HAMZA, M. ROCHE *et al.* : Classification automatique de documents bruités à faible contenu textuel. *RNTI : Revue des Nouvelles Technologies de l'Information*, 1:25, 2009.

[LLT03] Y. LU et C. LIM TAN : A nearest-neighbor chain based approach to skew estimation in document images. *Pattern Recognition Letters*, 24(14):2315–2323, 2003.

[LNGT01] JC LECOQ, L. NAJMAN, O. GIBOT et E. TRUPIN : Benchmarking commercial ocr engines for technical drawings indexing. *In Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 138–142. IEEE, 2001.

[Low04] D.G. LOWE : Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[LSS07] S. LI, Q. SHEN et J. SUN : Skew detection using wavelet decomposition and projection profile analysis. *Pattern recognition letters*, 28(5):555–562, 2007.

[MCUP04] J. MATAS, O. CHUM, M. URBAN et T. PAJDLA : Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[MGW07] R. MARÉE, P. GEURTS et L. WEHENKEL : Content-based image retrieval by indexing random subwindows with randomized trees. *In Proceedings of the 8th Asian conference on Computer vision-Volume Part II*, pages 611–620. Springer-Verlag, 2007.

[ML09] M. MUJA et D.G. LOWE : Fast approximate nearest neighbors with automatic algorithm configuration. *In International Conference on Computer Vision Theory and Applications (VISSAPP 09)*, volume 340, pages 331–340, 2009.

[ML12] S. MCCANN et D.G. LOWE : Local naive bayes nearest neighbor for image classification. *In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3650–3656. IEEE, 2012.

[MMCS04] S. MARINAI, E. MARINO, F. CESARINI et G. SODA : A general system for the retrieval of document images from digital libraries. pages 150–173, 2004.

[MS08] G. MOISE et J. SANDER : Finding non-redundant, statistically significant regions in high dimensional data : a novel approach to projected and subspace clustering. *In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 533–541. ACM, 2008.

[MSE06] G. MOISE, J. SANDER et M. ESTER : P3c : A robust projected clustering algorithm. *In Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 414–425. IEEE, 2006.

[MY09] J.M. MOREL et G. YU : Asift : A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[NCO11] T.T.H. NGUYEN, M. COUSTATY et J.M. OGIER : Bags of strokes based approach for classification and indexing of drop caps. *In Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 349–353. IEEE, 2011.

[NH94] R.T. NG et J. HAN : Efficient and effective clustering methods for spatial data mining. *In Proceedings of the International Conference on Very Large Data Bases*, pages 144–155, 1994.

[NJ11] S. NA et P. JINXIAO : Fast and robust skew detection for scanned documents. *In Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on*, volume 8, pages 4170–4173. IEEE, 2011.

[NJT06] E. NOWAK, F. JURIE et B. TRIGGS : Sampling strategies for bag-of-features image classification. *Computer Vision–ECCV 2006*, pages 490–503, 2006.

[NKI06] T. NAKAI, K. KISE et M. IWAMURA : Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. *Document Analysis Systems VII*, pages 541–552, 2006.

[NPBT07] R. NILSSON, J.M. PEÑA, J. BJORKEGREN et J. TEGNÉR : Consistent feature selection for pattern recognition in polynomial time. *The Journal of Machine Learning Research*, 8:589–612, 2007.

[NS84] G. NAGY et S. SETH : Hierarchical representation of optically scanned documents. *In Seventh International Conference on Pattern Recognition : Montreal, Canada, July 30-August 2, 1984 : proceedings*, page 347. IEEE Computer Society Press, 1984.

[OK95] L. O'GORMAN et R. KASTURI : *Document image analysis*. IEEE Computer Society Press London, 1995.

[PAK10] A.P. PSYLLOS, C.N.E. ANAGNOSTOPOULOS et E. KAYAFAS : Vehicle logo recognition using a sift-based enhanced matching scheme. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):322–328, 2010.

[PN07] F. PARADIS et J.Y. NIE : Contextual feature selection for text classification. *Information processing & management*, 43(2):344–352, 2007.

[PVDL02] K. POLLARD et M.J. VAN DER LAAN : A method to identify significant clusters in gene expression data. *Invited Proceedings of Sci2002*, 2:318–325, 2002.

[RATL11] M. RUSINOL, D. ALDAVERT, R. TOLEDO et J. LLADÓS : Browsing heterogeneous document collections by a segmentation-free word spotting method. *In Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 63–67. IEEE, 2011.

[RBD06] JY RAMEL, S. BUSSON et ML DEMONET : Agora : the interactive document image analysis tool of the bvh project. *In Second International Conference on Document Image Analysis for Libraries DIAL06*, pages 145–155. Ieee, 2006.

[Rib98] A. RIBERT : Structuration évolutive de données : Application à la construction de classifieurs distribués. *These de doctorat, Université de Rouen*, 1998.

[RJD+11] Vincent RABEUX, Nicholas JOURNET, Jean-Philippe DOMENGER *et al.* : Ancient documents bleed-through evaluation and its application for predicting ocr error rates. 2011.

[RJN96] S.V. RICE, F.R. JENKINS et T.A. NARTKER : *The fifth annual test of OCR accuracy*. Information Science Research Institute, 1996.

[RL09] M. RUSINOL et J. LLADÓS : Logo spotting by a bag-of-words approach for document categorization. *In 2009 10th International Conference on Document Analysis and Recognition*, pages 111–115. IEEE, 2009.

[RLDB07] J.Y. RAMEL, S. LERICHE, ML DEMONET et S. BUSSON : User-driven page layout analysis of historical printed books. *International Journal on Document Analysis and Recognition*, 9(2):243–261, 2007.

[Rou87] P.J. ROUSSEEUW : Silhouettes : a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[RTBO12] C. RIGAUD, N. TSOPZE, J.C. BURIE et J.M. OGIER : Extraction robuste des cases et du texte de bandes dessinées. pages 349–360, March 2012.

[RTZK99] S. RAMDANE, B. TACONET, A. ZAHOUR et S. KEBAIRI : Apprentissage et reconnaissance automatique de types de formulaires par une méthode statistique. *In 17ème Colloque sur le traitement du signal et des images, FRA, 1999*. GRETSI, Groupe d'Études du Traitement du Signal et des Images, 1999.

[SAH08] C. SILPA-ANAN et R. HARTLEY : Optimised kd-trees for fast image descriptor matching. *Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[SAJ10] T.I. SIMPSON, J.D. ARMSTRONG et A. JARMAN : Merged consensus clustering to assess and improve class discovery with microarray data. *BMC bioinformatics*, 11(1):590, 2010.

[Sau11] E. SAUND : Scientific Challenges Underlying Production Document Processing. *Proceedings of Document Recognition and Retrieval XVIII*, 7874:787402, 2011.

[SDR01] C. SHIN, D. DOERMANN et A. ROSENFELD : Classification of document pages using structure-based features. *International Journal on Document Analysis and Recognition*, 3(4):232–247, 2001.

[Seb02] F. SEBASTIANI : Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[SG04] Z. SHI et V. GOVINDARAJU : Line separation for complex document images using fuzzy runlength. *In Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on*, pages 306–312, 2004.

[SH07] M. SUZUKI et S. HIRASAWA : Text categorization based on the ratio of word frequency in each categories. *In Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 3535–3540. IEEE, 2007.

[SIR99] T. STEINHERZ, N. INTRATOR et E. RIVLIN : Skew detection via principal components analysis. *In Document Analysis and Recognition (ICDAR), 1999 International Conference on*, pages 153–156. IEEE, 1999.

[SKK00] M. STEINBACH, G. KARYPIS et V. KUMAR : A comparison of document clustering techniques. *In KDD, International Conference on Knowledge Discovery in Data*, volume 400, pages 525–526. ACM, 2000.

[Smi09] R. SMITH : Hybrid Page Layout Analysis via Tab-Stop Detection. pages 241–245, 2009.

[SNB11] F. SUR, N. NOURY et M.-O. BERGER : Image point correspondences and repeated patterns. Research Report 7693, INRIA, July 2011.

[SUL11] W. SONG, S. UCHIDA et M. LIWICKI : Look inside the world of parts of handwritten characters. *In Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 784–788. IEEE, 2011.

[Sun06] H.M. SUN : Enhanced constrained run-length algorithm for complex layout document processing. *International Journal of Applied Science and Engineering*, 4(3):297–309, 2006.

[TCTH12] S. THOMAS, C. CHATELAIN, P. THIERRY et L. HEUTTE : Combinaison architecture profonde/hmm pour l'extraction de séquences dans des documents manuscrits. pages 7–22, March 2012.

[TKI11] K. TAKEDA, K. KISE et M. IWAMURA : Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved llah. pages 1054–1058, 2011.

[TSYX00] C.L. TAN, S.Y. SUNG, Z. YU et Y. XU : Text retrieval from document images based on n-gram algorithm. *In Text and Web Mining Workshop, 6th Pacific Rim International Conference on Artificial Intelligence*, 2000.

[TWL02] C.M. TAN, Y.F. WANG et C.D. LEE : The use of bigrams to enhance text categorization. *Information processing & management*, 38(4):529–546, 2002.

[US09] H. UCHIYAMA et H. SAITO : Augmenting text document by on-line learning of local arrangement of keypoints. *In Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pages 95–98. IEEE, 2009.

[vBKSB06] J. van BEUSEKOM, D. KEYSERS, F. SHAFAIT et TM BREUEL : Distance measures for layout-based document image retrieval. *In Document Image Analysis for Libraries, 2006. DIAL'06. Second International Conference on*, pages 11–pp, 2006.

[VC09] E. VALLE et M. CORD : Advanced Techniques in CBIR : Local Descriptors, Visual Dictionaries and Bags of Features. *In Tutorials of the XXII Brazilian Symposium on Computer Graphics and Image Processing*, pages 72–78. IEEE, 2009.

[VJ01] P. VIOLA et M. JONES : Rapid object detection using a boosted cascade of simple features. *In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[WCW82] K.Y. WONG, R.G. CASEY et F.M. WAHL : Document analysis system. *IBM journal of research and development*, 26(6):647–656, 1982.

[WDL+09] K. WEINBERGER, A. DASGUPTA, J. LANGFORD, A. SMOLA et J. ATTENBERG : Feature hashing for large scale multitask learning. *In Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM, 2009.

[WLKL04] K.G. WOO, J.H. LEE, M.H. KIM et Y.J. LEE : FINDIT : a fast and intelligent subspace clustering algorithm using dimension voting. *Information and Software Technology*, 46(4):255–271, 2004.

[WPS06] F. WOLF, T. POGGIO et P. SINHA : Human document classification using bags of words. *Computer Science and Artificial Intelligence Laboratory, MIT*, 2006.

[WS89] D. WANG et S.N. SRIHARI : Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics, and Image Processing*, 47(3): 327–352, 1989.

[WYY+10] J. WANG, J. YANG, K. YU, F. LV, T. HUANG et Y. GONG : Locality-constrained linear coding for image classification. *In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[YCN04] K.Y. YIP, D.W. CHEUNG et M.K. NG : Harp : A practical projected clustering algorithm. *IEEE Transactions on knowledge and data engineering*, 16(11): 1387–1397, 2004.

[YJHN07] J. YANG, Y.G. JIANG, A.G. HAUPTMANN et C.W. NGO : Evaluating bag-of-visual-words representations in scene classification. *In Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206. ACM, 2007.

[YM11] I.Z. YALNIZ et R. MANMATHA : A fast alignment scheme for automatic ocr evaluation of books. *In Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 754–758. IEEE, 2011.

[YSST07] G. YANG, C.V. STEWART, M. SOFKA et C.L. TSAI : Alignment of challenging image pairs : Refinement and region growing starting from a single keypoint correspondence. *IEEE Trans. Pattern Anal. Machine Intell*, 23(11):1973–1989, 2007.

[YYGH09] J. YANG, K. YU, Y. GONG et T. HUANG : Linear spatial pyramid matching using sparse coding for image classification. *In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. Ieee, 2009.

[ZH03] X.S. ZHOU et T.S. HUANG : Relevance feedback in image retrieval : A comprehensive review. *Multimedia systems*, 8(6):536–544, 2003.

[ZHD99] B. ZHANG, M. HSU et U. DAYAL : K-harmonic means-a data clustering algorithm. *Hewllet-Packard Research Laboratory Technical Report HPL-1999-124*, 1999.

[ZHF08] Q. ZHAO, V. HAUTAMAKI et P. FRANTI : Knee Point Detection in BIC for Detecting the Number of Clusters. *In Advanced Concepts for Intelligent Vision Systems*, pages 664–673. Springer, 2008.

[ZL11] Y. ZHANG et WU LENAN : A fast document image denoising method based on packed binary format and source word accumulation. *Journal of Convergence Information Technology*, 6(2):131–137, 2011.

[ZRL96] T. ZHANG, R. RAMAKRISHNAN et M. LIVNY : BIRCH : an efficient data clustering method for very large databases. *ACM SIGMOD Record*, 25(2): 103–114, 1996.

[ZWL10] Z. ZHAO, L. WANG et H. LIU : Efficient spectral feature selection with minimum redundancy. *In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 2010.