Segmentation d'images de documents manuscrits composites : application aux documents de chimie

Nabil Ghanmi

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Segmentation d'images de documents manuscrits composites : application aux documents de chimie

THÈSE

présentée et soutenue publiquement le 30 Septembre 2016

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Nabil Ghanmi

Composition du jury

Président :	Rolf Ingold	Professeur, Universitté de Fribourg
Rapporteurs :	Laurent Wendling	Professeur, Université Paris Descartes
	Bertrand Coüasnon	MC-HDR, INSA de Rennes
Examinateur :	Bart Lamiroy	MC-HDR, Université de Lorraine
Invité :	Frank Hoonakker	Président d'eNovalys
Directeur de thèse :	Abdel Belaïd	Professeur, Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Remerciements

Je remercie tout d'abord Dieu tout puissant de m'avoir donné le courage, la force et la patience d'achever ce modeste travail.

Je remercie mon directeur de thèse Abdel Belaïd pour sa disponibilité et ses efforts tout au long des années de ma thèse. Je remercie également Frank Hoonakker pour la collaboration industrielle.

Je tiens à remercier aussi l'ensemble des membres de mon jury : Rolf Ingold, Laurent Wendling, Bertrand Coüasnon et Bart Lamiroy. Je les remercie d'avoir accepté d'évaluer ce travail. La qualité de leur lecture, leurs remarques et suggestions me permettent d'envisager de plus riches perspectives.

Mes remerciements vont également aux membres de l'équipe READ du LORIA, ainsi qu'à tous mes amis, spécialement Abdessalem, Chaouki et Issam. Je les remercie pour leurs amitiés et leur soutien moral.

Je tiens à adresser mes plus vifs remerciements à ma merveilleuse fiancée pour l'amour et le soutien qu'elle m'a gratifiés tout au long de la thèse.

Enfin, je tiens à remercier très vivement ma chère mère, mes frères et mes sœurs. Leur affection et leur soutien m'ont été d'un grand secours tout au long de ma vie professionnelle et personnelle.

Je dédie cette thèse
à ma chère mère
à l'âme de mon cher père
à toute ma famille

Sommaire

Chapitre 1
Introduction générale

Chapitre 2
Description de la base de documents de chimie

2.1 Motivation . . . 13
2.2 Description des documents de chimie . . . 14
2.2.1 Caractéristiques générales . . . 14
2.2.2 Structure physique du document . . . 14
2.3 Création de la base . . . 17
2.3.1 Collecte et acquisition des documents . . . 17
2.3.2 Génération de la vérité terrain . . . 17
2.4 Conclusion . . . 20

Chapitre 3
Etat de l'art

3.1 Introduction . . . 25
3.2 Séparation texte/graphique . . . 26
3.2.1 Les méthodes basées sur les composantes connexes . . . 26
3.2.2 Les méthodes de zonage . . . 27
3.3 Segmentation en lignes . . . 28
3.3.1 Méthodes de projection . . . 29
3.3.2 Transformée de Hough . . . 30
3.3.3 Méthodes de lissage . . . 31
3.3.4 Méthodes de regroupement . . . 32
3.4 Extraction de tableaux . . . 33
3.4.1 Détection de tableaux . . . 34
3.4.1.1 Méthodes basées sur les lignes graphiques . . . 34
3.4.1.2 Méthodes basées sur les espaces . . . 35

3.4.1.3 Méthodes basées sur l'alignement vertical 36
3.4.2 Reconnaissance de tableaux 38
3.4.2.1 Méthodes basées sur les séparateurs graphiques 39
3.4.2.2 Méthodes basées sur les espaces 39
3.4.2.3 Alignement de texte 41
3.4.2.4 Autres approches 43
3.5 Extraction de champs numériques 43
3.5.1 Localisation de champs numériques 43
3.5.2 Reconnaissance de chiffres connectés 47
3.5.2.1 Stratégie de segmentation puis reconnaissance 48
3.5.2.2 Stratégie de segmentation-reconnaissance 50
3.6 Conclusion 51

Chapitre 4
Extraction de formules chimiques

4.1 Introduction 53
4.2 Difficultés 53
4.3 Prétraitement 54
4.4 Extraction de la formule chimique 56
4.4.1 Segmentation 56
4.4.1.1 Granularité de la segmentation : les structures linéaires 56
4.4.1.2 Segmentation de la page en structures linéaires 57
4.4.2 Caractérisation des structures linéaires 60
4.4.3 Classification des structures linéaires 62
4.4.3.1 Choix du classifieur 62
4.4.3.2 Problème de données déséquilibrées 62
4.4.3.3 Règle de sélection 64
4.5 Expérimentation et résultats 66
4.5.1 Évaluation de la segmentation 66
4.5.2 Apprentissage du classifieur 67
4.5.3 Évaluation de l'extraction de la formule chimique 67
4.6 Conclusion 69

Chapitre 5
Détection de tableaux

5.1 Introduction 71
5.2 Étiquetage de séquences de données 73

5.2.1 Les modèles génératifs . . . 73
5.2.2 Les modèles discriminants . . . 74
5.3 Les champs aléatoires conditionnels . . . 74
5.3.1 Formalisme des CACs . . . 74
5.3.2 Quelques applications de CACs dans l'ananlyse des images de documents . . . 76
5.4 Proposition d'un modèle CAC pour la détection de tableaux . . . 79
5.4.1 Modélisation du problème . . . 79
5.4.2 Extraction des descripteurs . . . 80
5.4.2.1 Segmentation en lignes . . . 80
5.4.2.2 Segmentation en mots . . . 81
5.4.2.3 Extraction de descripteurs de lignes . . . 84
5.4.3 Apprentissage du modèle . . . 85
5.4.4 Décodage . . . 87
5.5 Expérimentation et résultats . . . 88
5.5.1 Évaluation de la segmentation . . . 88
5.5.2 Évaluation de la détection de tableau . . . 89
5.6 Conclusion . . . 91

Chapitre 6
Extraction de la structure de tableaux

6.1 Introduction . . . 95
6.2 Système proposé . . . 95
6.3 Niveau structurel . . . 98
6.3.1 Détection des lignes graphiques . . . 98
6.3.2 Analyse de la grille graphique . . . 99
6.3.3 Projection des boîtes englobantes . . . 100
6.4 Niveau contenu . . . 100
6.4.1 Extraction et filtrage des CCs . . . 101
6.4.2 Discrimination chiffre/non chiffre . . . 102
6.4.2.1 Détection de chiffres isolés . . . 103
6.4.2.2 Détection et segmentation de chiffres connectés . . . 104
6.4.3 Correction de la segmentation par l'analyse du contenu . . . 110
6.5 Expérimentation et résultats . . . 110
6.6 Conclusion . . . 113

Chapitre 7
Conclusion et perspectives

Annexe A
Liste de publications

Bibliographie **119**

Résumé

Cette thèse traite de la segmentation structurelle de documents issus de cahiers de chimie. Ce travail est utile pour les chimistes en vue de prendre connaissance des conditions des expériences réalisées. Les documents traités sont manuscrits, hétérogènes et multi-scripteurs. Bien que leur structure physique soit relativement simple, une succession de trois régions représentant : la formule chimique de l'expérience, le tableau des produits utilisés et un ou plusieurs paragraphes textuels décrivant le déroulement de l'expérience, les lignes limitrophes des régions portent souvent à confusion, ajouté à cela des irrégularités dans la disposition des cellules du tableau, rendant le travail de séparation un vrai défi.

La méthodologie proposée tient compte de ces difficultés en opérant une segmentation à plusieurs niveaux de granularité, et en traitant la segmentation comme un problème de classification. D'abord, l'image du document est segmentée en structures linéaires à l'aide d'un lissage horizontal approprié. Le seuil horizontal combiné avec une tolérance verticale avantage le regroupement des éléments fragmentés de la formule sans trop fusionner le texte. Ces structures linéaires sont classées en Texte ou Graphique en s'appuyant sur des descripteurs structurels spécifiques, caractéristiques des deux classes. Ensuite, la segmentation est poursuivie sur les lignes textuelles pour séparer les lignes du tableau de celles de la description. Nous avons proposé pour cette classification un modèle CAC qui permet de déterminer la séquence optimale d'étiquettes associées à la séquence des lignes d'un document. Le choix de ce type de modèle a été motivé par sa capacité à absorber la variabilité des lignes et à exploiter les informations contextuelles.

Enfin, pour le problème de la segmentation de tableaux en cellules, nous avons proposé une méthode hybride qui fait coopérer deux niveaux d'analyse : structurel et syntaxique. Le premier s'appuie sur la présence des lignes graphiques et de l'alignement de texte et d'espaces ; et le deuxième tend à exploiter la cohérence de la syntaxe très réglementée du contenu des cellules. Nous avons proposé, dans ce cadre, une approche contextuelle pour localiser les champs numériques dans le tableau, avec reconnaissance des chiffres isolés et connectés.

La thèse étant effectuée dans le cadre d'une convention CIFRE, en collaboration avec la société eNovalys, nous avons implémenté et testé les différentes étapes du système sur une base conséquente de documents de chimie.

Mots-clés: document de chimie, structures linéaires, séparation texte/graphique, classification, extraction de tableaux, champs aléatoires conditionnels, extraction de numériques.

Abstract

This thesis deals with chemistry document segmentation and structure analysis. This work aims to help chemists by providing the information on the experiments which have already been carried out. The documents are handwritten, heterogeneous and multi-writers. Although their physical structure is relatively simple, since it consists of a succession of three regions representing : the chemical formula of the experiment, a table of the used products and one or more text blocks describing the experimental procedure, several difficulties are encountered. In fact, the lines located at the region boundaries and the imperfections of the table layout make the separation task a real challenge.

The proposed methodology takes into account these difficulties by performing segmentation at several levels and treating the region separation as a classification problem. First, the document image is

segmented into linear structures using an appropriate horizontal smoothing. The horizontal threshold combined with a vertical overlapping tolerance favor the consolidation of fragmented elements of the formula without too merge the text. These linear structures are classified in text or graphic based on discriminant structural features. Then, the segmentation is continued on text lines to separate the rows of the table from the lines of the raw text locks. We proposed for this classification, a CRF model for determining the optimal labelling of the line sequence. The choice of this kind of model has been motivated by its ability to absorb the variability of lines and to exploit contextual information.

For the segmentation of table into cells, we proposed a hybrid method that includes two levels of analysis : structural and syntactic. The first relies on the presence of graphic lines and the alignment of both text and spaces. The second tends to exploit the coherence of the cell content syntax. We proposed, in this context, a Recognition-based approach using contextual knowledge to detect the numeric fields present in the table. The thesis was carried out in the framework of CIFRE, in collaboration with the eNovalys campany. We have implemented and tested all the steps of the proposed system on a consequent dataset of chemistry documents.

Keywords: Chemistry document, Linear structure, Text/Graphic separation, Classification, Table extraction, Conditional Random Fields, numeric extraction.

Chapitre 1

Introduction générale

Le travail de cette thèse est une contribution à la mise en place d'un système de reconnaissance des images de documents issues de la numérisation des cahiers de laboratoires de recherche en chimie. Ce sujet est important pour les chimistes car il leur permet d'exploiter les informations contenues dans ces documents et d'améliorer l'efficacité de leur travail expérimental.

Dans la chaîne de découverte des médicaments, la synthèse de molécules bioactives est un travail répétitif qui nécessite plusieurs étapes d'élaboration et d'expérimentation avant de découvrir la molécule désirée. Actuellement, le taux de réactions infructueuses est très élevé (entre 50% et 70%), ce qui induit une productivité faible, un gâchis de matière première et une perte financière importante (de 45 à 63 milliards d'euros investis à perte). Ces échecs sont dus à l'utilisation de moteurs de recherche dont la technologie, basée sur la recherche structurale, date des années 80, d'une part. D'autre part, les bases de données utilisées sont constituées par l'indexation de quelques publications scientifiques et d'une infime partie des expériences de chimie mondialement réalisées.

Pour combler ce manque, la société eNovalys a développé un moteur de recherche innovant qui permet de calculer des similarités réactionnelles. Ce moteur permet de présenter au chimiste les meilleures expériences réalisées et de lui extraire de manière intelligente des informations pertinentes. Cet avantage considérable est cependant atténué par la nature des données indexées dans les bases actuelles qui ne permettent pas de fournir directement le mode opératoire et les conditions réactionnelles nécessaires à la synthèse des molécules. C'est pour cette raison que la société eNovalys a sollicité l'équipe READ, dans le cadre d'un contrat CIFRE, pour développer un environnement numérique de travail. Cet environnement travaillera directement à partir des cahiers de chimie. Il s'occupera de toutes les tâches depuis la numérisation des cahiers jusqu'à l'extraction d'information nécessaire aux chimistes. Ce système permettra à terme de rassembler le maximum d'expériences chimiques effectuées, avec une lecture la plus fidèle possible de son contenu. L'objectif de la société est donc de numériser le patrimoine du savoir en chimie existant dans les cahiers de laboratoires.

Comme le montre l'exemple de page illustré dans la figure 1.1, les documents sont bien structurés selon une séquence qui est toujours identique et qui contient un schéma réactionnel (appelé aussi schéma de formule chimique) illustrant l'expérience réalisée, un tableau contenant les produits utilisés, un ou plusieurs paragraphes de texte décrivant le déroulement de l'expérience. En plus de ces trois principales régions, nous notons parfois la présence d'un entête (date, numéro de page ou de manipulation, etc.) et des images de petites tailles correspondant aux plaques employées dans la CCM (chromatographie sur couche mince) lors de l'expérience.

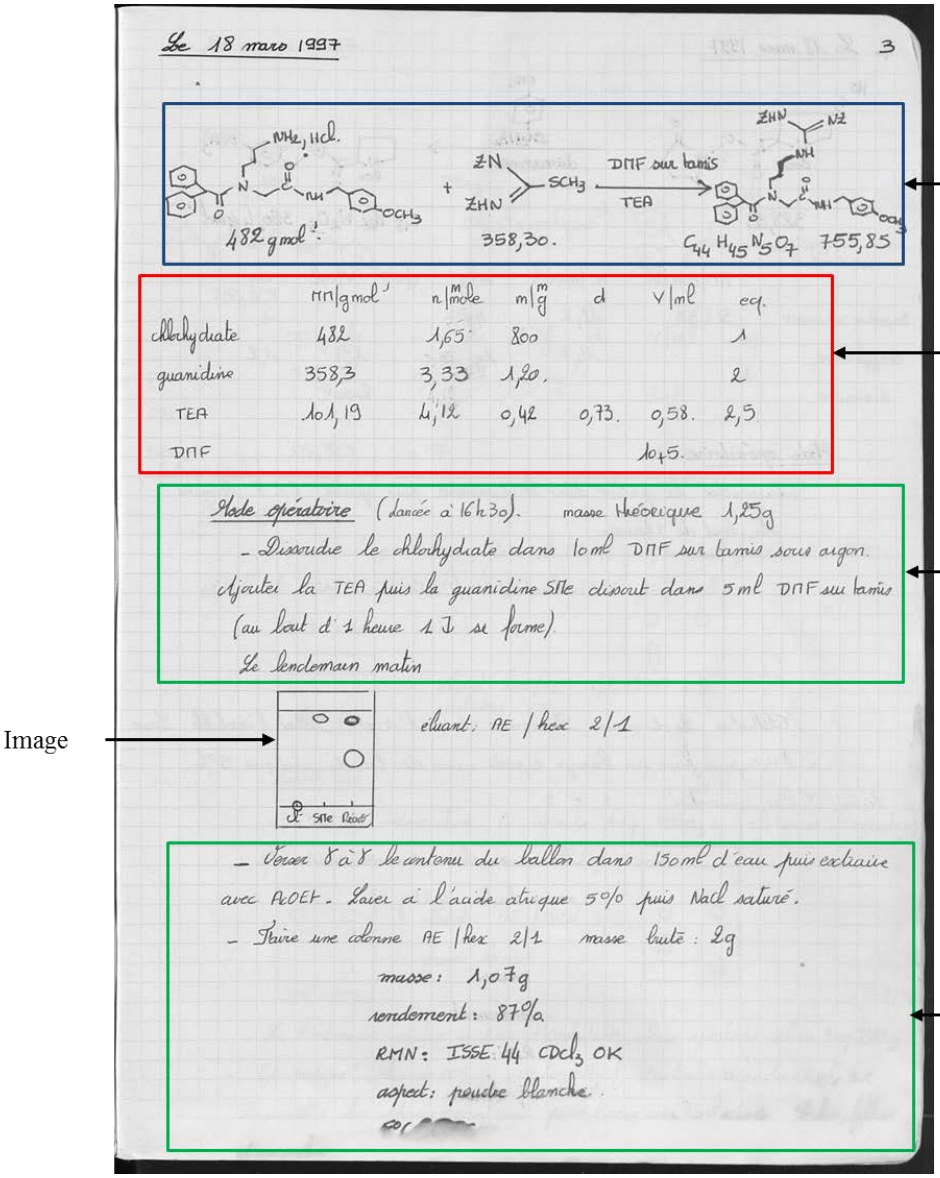


FIGURE 1.1 – Exemple d'un document de chimie avec ses principales zones d'informations.

Le travail proposé consiste à segmenter le document en ses différentes régions en vue de les faire reconnaître par des systèmes spécifiques. Aussi, notre tâche a consisté à trouver les méthodes de segmentation les plus adéquates pour ces documents.

La bibliographie en analyse de documents n'étant pas forcément concentrée sur ce type de problème et ces types de documents, nous nous sommes alors tournés vers les techniques de segmentation spécialisées de type : séparation texte/graphique pour l'extraction du schéma de la formule chimique, classification de lignes pour séparer les lignes de tableau de celles du texte, et enfin extraction de cellules de tableau par localisation et reconnaissance des numériques.

Le mémoire est organisé comme suit. Dans le **chapitre 2**, nous décrivons les documents de chimie et nous présentons la base ChemistryDB que nous avons constituée. Il s'agit d'une base annotée des docu-

ments extraits des cahiers de laboratoires de chimie qui nous a servi pour l'apprentissage et l'évaluation des méthodes développées.

Dans le **chapitre 3**, nous présentons une étude bibliographique sur différents traitements situés à différents niveaux de la chaîne d'analyse des images de documents.

Dans le **chapitre 4**, nous présentons la méthode de séparation texte/graphique pour séparer la formule chimique du reste du document. Nous avons traité ce problème comme une opération de classification binaire (Texte ou Graphique) de segments homogènes issus de la segmentation d'une page à un niveau de granularité choisi adéquatement pour cette classification.

Dans le **chapitre 5**, nous décrivons l'approche de détection du tableau dans le restant de la page de laquelle a été enlevée la formule chimique. Ce problème est abordé de la même manière que précédemment par classification de lignes de texte. Un modèle basé sur les Champs Aléatoires Conditionnels (CACs) permettant de tenir compte des caractéristiques individuelles d'une ligne ainsi que de ses caractéristiques contextuelles (dans son voisinage) est utilisé pour la classification.

Dans le **chapitre 6**, nous présentons la méthode de segmentation du tableau en cellules. C'est une méthode hybride qui s'appuie à la fois sur la structure physique/géométrique du tableau et sur la syntaxe du contenu des cellules. L'analyse de la syntaxe n'était pas possible sans avoir reconnu quelques éléments textuels dans le tableau ; c'est pour cela que nous avons proposé une méthode d'extraction de champs numériques qui constituent des champs révélateurs de la syntaxe des tableaux.

Nous terminerons ce mémoire par une discussion sur les travaux accomplis et présenterons quelques perspectives dans le **chapitre 7**.

Chapitre 2

Description de la base de documents de chimie

Sommaire

2.1 Motivation 13
2.2 Description des documents de chimie 14
2.2.1 Caractéristiques générales 14
2.2.2 Structure physique du document 14
2.3 Création de la base 17
2.3.1 Collecte et acquisition des documents 17
2.3.2 Génération de la vérité terrain 17
2.4 Conclusion 20

2.1 Motivation

La richesse que constituent les documents de chimie justifie le besoin d'un système d'analyse et de reconnaissance de ces documents. Étant les premiers (à notre connaissance) à travailler sur de tels documents et en l'absence d'une vérité terrain, la construction d'une base de documents annotés nous a paru nécessaire. Nous avons collecté et annoté 500 documents extraits de cahiers de laboratoires de recherche en chimie. Plusieurs annotations ont été effectuées sur les images de ces documents. Outre notre besoin personnel de ces annotations, celles-ci peuvent servir de références pour différents travaux, notamment en matière de :

— analyse de structures de documents et extraction de régions d'intérêts, plus précisément la séparation texte/graphique et la localisation de tableaux ;
— extraction de lignes dans les régions textuelles ;
— segmentation en mots ;
— analyse et interprétation de tableaux ;
— localisation de champs numériques.

2.2 Description des documents de chimie

2.2.1 Caractéristiques générales

Un document décrit une expérience chimique. Il contient toute l'information essentielle à la reproduction d'une manipulation réalisée. Ces documents sont :

— manuscrits, en langue française ;
— multi-scripteurs : la base appartient à plusieurs scripteurs, mais un même document est écrit par un seul scripteur ;
— hétérogènes : contenant différents types de régions : graphique, texte, tableau.
— non-contraints : les écritures ont été faites sans aucune contrainte de position ou de taille ;
— mono-pages et mono-colonnes.

Par ailleurs, il convient de souligner que ces documents disposent de deux caractéristiques avantageuses d'un point de vue de la lecture automatique des documents :

— étant apposée sur des cahiers pré-tracés, l'écriture est le plus souvent horizontale et ne nécessite aucun traitement de correction d'inclinaison ;
— la structure physique des documents, c'est-à-dire leur organisation en régions homogènes, est à peu-près identique.

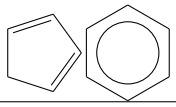
2.2.2 Structure physique du document

Un exemple de document a été montré dans la Figure 1.1 Nous allons donner une description plus détaillée des trois principales régions composant ce type de document.

Schéma de la réaction chimique. Nous l'appelons formule chimique dans la suite de ce manuscrit. Cette région est composée de deux couches : une couche texte et une autre graphique (voir Figure 2.1). Les principaux symboles composant une formule chimique sont illustrés dans le Tableau 2.1.

Tableau. Il a pour vocation de stocker les valeurs de différentes grandeurs pour différents réactifs et produits chimiques utilisés dans l'expérience décrite dans le document. Le tableau a une structure 2D simple, c'est-à-dire une matrice 2D de cellules avec des lignes et des colonnes simples dont les entêtes ont un seul niveau hiérarchique. Une structure typique complète est celle présentée dans la figure 2.2, où la première colonne représente la colonne entête (les noms des produits chimiques), la première ligne représente la ligne entête (des grandeurs) et les cellules de données, que nous noterons $DC(i, j)$, $1 \leq i \leq nbLignes$ et $1 \leq j \leq nbColonnes$, contiennent les valeurs correspondantes de l'entête de

TABLE 2.1 – Les symboles composant une formule chimique.

Couche	Symboles	Alphabet
Texte	chiffres	1 2 3 4 5 6 7 8 9
	lettres	a..z et A..Z
	opérateurs	+ - = ( ) ≡
Graphique	liaisons chimiques	
	opérateurs	→ ↔
	cycles	

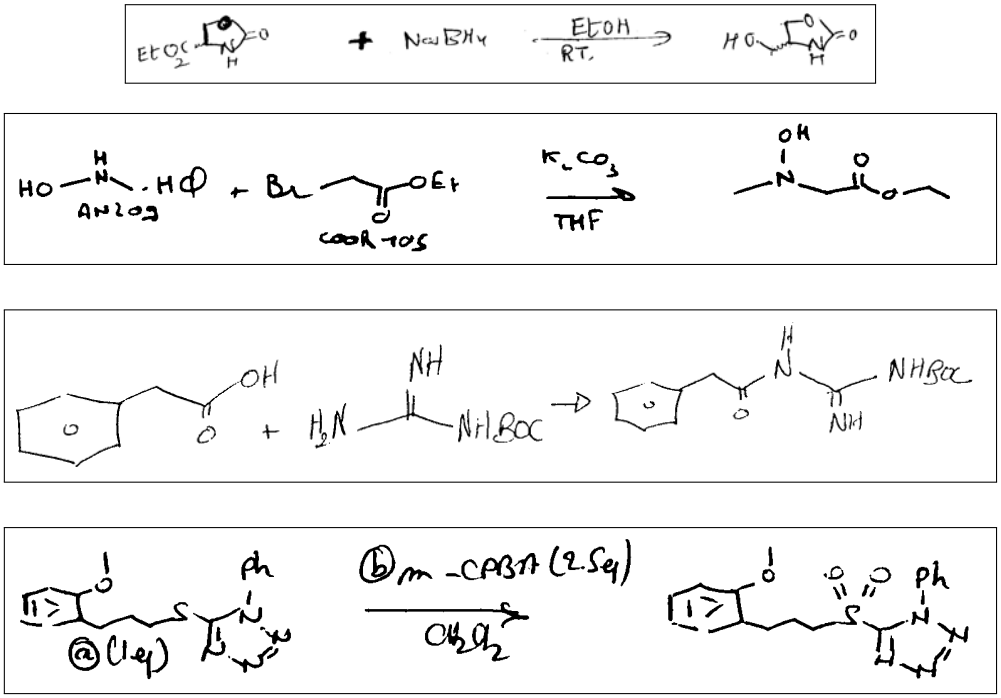


FIGURE 2.1 – Exemples de formules chimiques

colonne j pour l'entête de ligne i. Il est cependant important de noter que la ligne entête et/ou la colonne entête peuvent être absentes (voir Figure 2.3).

Région de texte. Elle contient une description du déroulement du processus expérimental sous forme d'une séquence de lignes. Le contenu de cette région est constitué d'un vocabulaire ouvert. Cependant, ce sont des mots ou des expressions spécifiques du domaine de la chimie qui constituent les principales informations dans cette région. Par exemple, le champ "rendement", couramment utilisé chez les chimistes pour exprimer le résultat d'une expérience chimique, est une information clé dans un document. De même, les champs numériques sont en général des informations importantes.

Visuellement, nous notons une large variabilité de l'écriture d'un scripteur à un autre (voir Figure 2.4) :

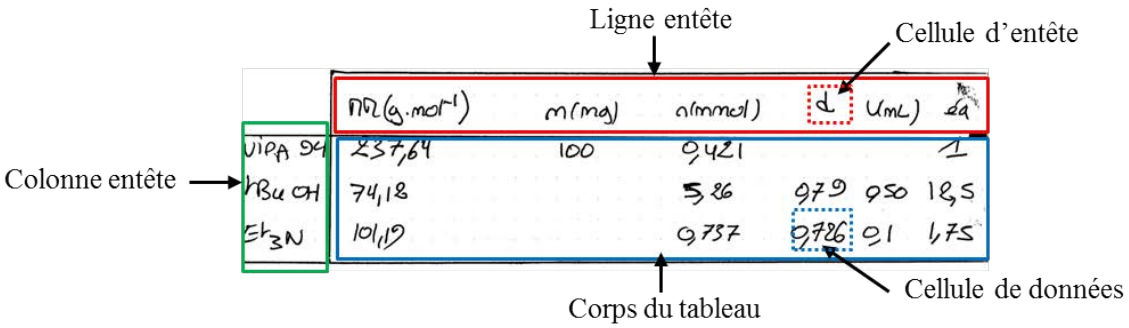
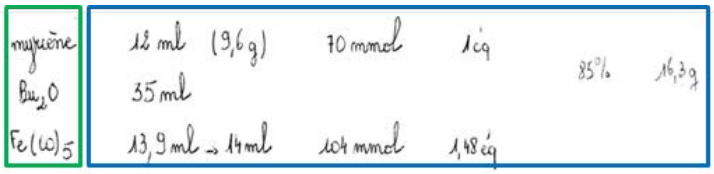
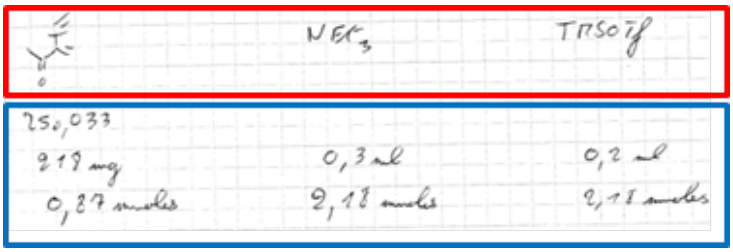


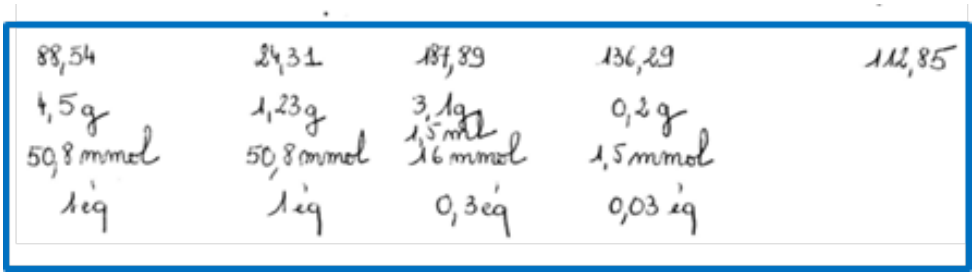
FIGURE 2.2 – Exemple d'une structure typique complète d'un tableau



(a)



(b)



(c)

FIGURE 2.3 – Exemples de tableaux incomplets, (a) sans ligne entête, (b) sans colonne entête et (c) sans les deux entêtes.

— au niveau du style : selon les habitudes de chaque scripteur, l'écriture peut être scripte (en bâtonnet), cursive ou mixte ;
— au niveau de la taille ;
— au niveau de la lisibilité.

2.3 Création de la base

2.3.1 Collecte et acquisition des documents

La base des documents de chimie (nous l'appelons ChemistryDB) est composée de 500 documents collectés auprès de trois laboratoires de recherche en chimie dans la région Strasbourgeoise. La numérisation de ces documents est effectuée en utilisant un scanner à plat et une résolution de 300 ppp. Lors de cette étape, un traitement/réglage a été appliqué afin d'éliminer le quadrillage des pages qui est perturbant pour les outils d'analyse. Les images obtenues sont stockées en niveaux de gris, au format TIF.

2.3.2 Génération de la vérité terrain

La mise à disposition des données décrivant la vérité terrain associée à une base de documents rend cette dernière plus utile et plus attractive. Mais, la production de telles données est une tâche difficile et fastidieuse car elle nécessite une saisie manuelle des annotations, ce qui induit un coût et un délai élevé.

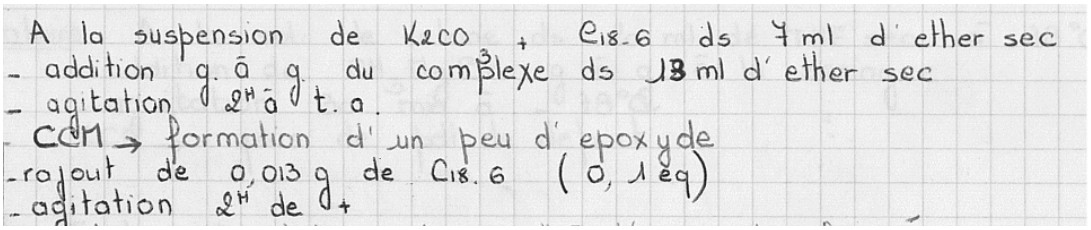
Pour pallier ce problème, nous avons généré les données de vérité associées aux documents de chimie de manière semi-automatique en utilisant l'outil GEDI [1]. Les données de vérité sont présentées dans un fichier XML décrivant une image comme un ensemble de zones ayant plusieurs attributs. D'abord, une annotation automatique a été effectuée en appliquant nos algorithmes sur les images des documents. Le résultat de cette annotation est enregistré dans un fichier XML ayant un format spécifique et un nom identique à celui du document image pour qu'il puisse être édité avec GEDI. Ensuite, des corrections/ajouts manuels ont été effectués sur ces annotations. Dans cette étape, l'image d'origine et le XML associé sont ouverts par GEDI et une inspection humaine est effectuée pour corriger/ajouter quelques zones ou leurs attributs.

Annotation de pages. L'annotation d'une page consiste à étiqueter ses principales zones à l'aide d'un ensemble d'attributs. Le niveau de détails présentés dans ces attributs dépend du niveau de l'analyse à effectuer sur les images. Comme nous nous proposons d'extraire les trois principales régions qui composent un document (la formule chimique, le tableau et la région de texte), nous avons opté pour une annotation qui décrit la position (les attributs col et row) et les dimensions (les attributs $width$ et $height$) de chacune de ces régions. Un exemple de fichier de vérité terrain est présenté dans la figure 2.5.

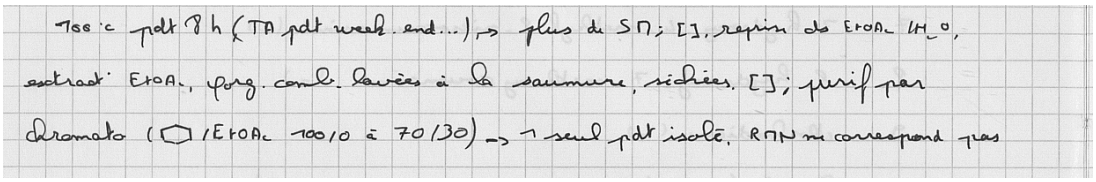
A partir de cette annotation, les imagettes correspondantes aux différentes régions sont isolées afin d'y effectuer les annotations nécessaires pour les traitements qui seront réalisés dessus. Le tableau 2.2 décrit l'organisation de la base *ChemistryDB* sous forme de quatre sous-bases.

Il convient de signaler ici que nous n'effectuons aucun traitement sur les formules chimiques et que nous nous contentons uniquement de les extraire. Pour cela, nous n'avons pas annoté le contenu des imagettes correspondantes. En revanche, les régions de texte et de tableau seront analysées davantage dans le cadre du présent travail. Les annotations de ces régions sont décrites en détail dans les paragraphes qui suivent.

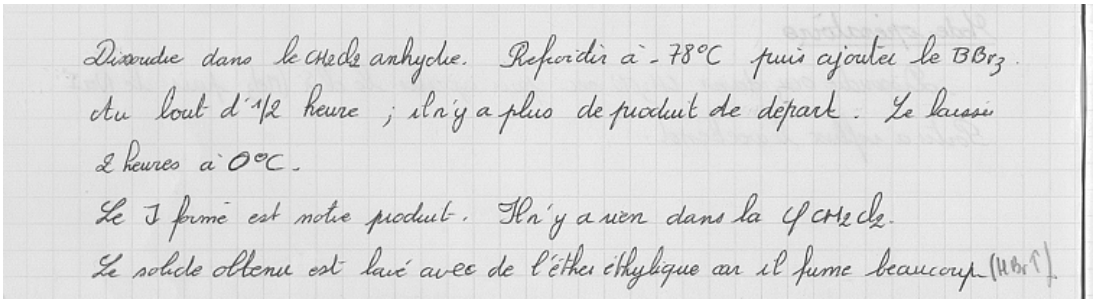
1. GEDI : **G**roundtruthing **E**nvironment for **D**ocument **I**mages : est un outil d'annotation générique d'images. Il est disponible gratuitement à la communauté de recherche et continue d'être développé.



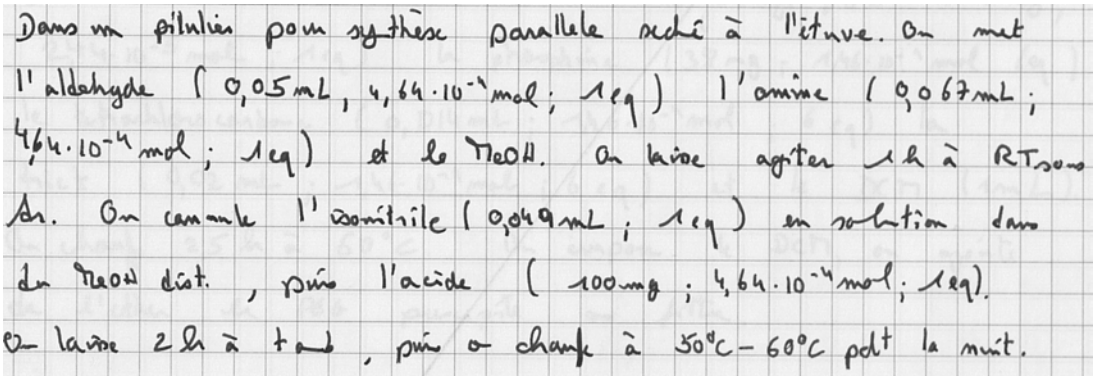
(a)



(b)

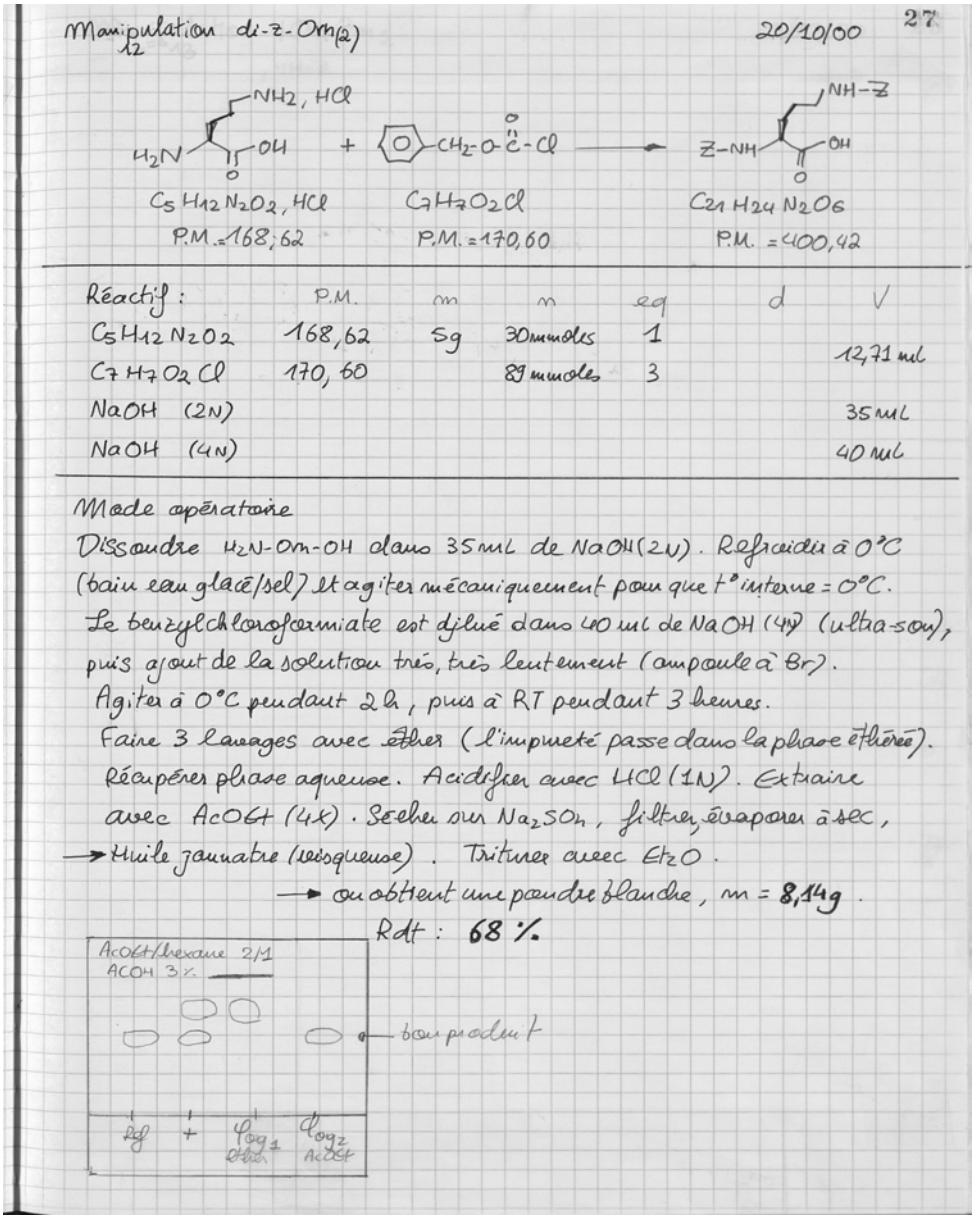


(c)



(d)

FIGURE 2.4 – Exemples illustrant la variabilité des écritures, (a) écriture en bâtonnet, (b) et (c) deux écritures cursives de tailles différentes et (d) écriture mixte.



(a)

```
<GEDI xmlns="http://lamp.cfar.umd.edu/media/projects/GEDI/" GEDI_version="2.4" GEDI_date="07/29/2013">
    <USER name="nghanmi" date="4/13/2016 16:49" dateFormat="mm/dd/yyyy hh:mm">  </USER>
    <DL_DOCUMENT src="11413.tif" NrOfPages="1" docTag="xml">
        <DL_PAGE gedi_type="DL_PAGE" src="11413.tif" pageID="1" width="2535" height="3816">
            <DL_ZONE gedi_type="Header" id="78" col="148" row="16" width="2304" height="148"> </DL_ZONE>
            <DL_ZONE gedi_type="Formula" id="79" col="324" row="200" width="2016" height="604"> </DL_ZONE>
            <DL_ZONE gedi_type="Table" id="80" col="204" row="860" width="2216" height="608" GraphicLines="false"> </DL_ZONE>
            <DL_ZONE gedi_type="Text" id="81" col="140" row="1540" width="2368" height="1432"> </DL_ZONE>
        </DL_PAGE>
    </DL_DOCUMENT>
</GEDI>
```

(b)

FIGURE 2.5 – Exemple d'annotation d'un document, (a) l'image du document, (b) le fichier d'annotation correspondant.

TABLE 2.2 – Organisation de la base ChemistryDB

ChemicalPage	contient 500 documents mono-pages.
ChemicalFormula	contient 500 imagettes des régions de formules chimiques extraites des documents originaux de la base ChemicalPage.
ChemicalText	contient 500 imagettes des régions textuelles (texte brute et tableau) extraites des documents originaux de la base ChemicalPage.
ChemicalTable	contient 500 imagettes des tableaux extraits des documents originaux de la base ChemicalPage.

Annotation de régions de texte. Nous considérons les deux régions de tableau et de description de l'expérience comme une région de texte composée de lignes de tableau (TableRow) et de lignes de texte (TextLine). L'annotation effectuée sur ces régions consiste à distinguer ces deux types de lignes. Ainsi, les documents de vérité correspondant contiennent les informations suivantes :

— les boîtes englobantes de toutes les lignes ainsi que leurs types : TableRow ou TextLine.
— les boîtes englobantes de tous les mots dans chacune des lignes.

La figure 2.6 illustre un exemple d'annotation d'une image extraite de la base ChemicalText.

Annotation de tableaux. Vu l'importance de la région du tableau et la profondeur de l'analyse que nous allons y effectuer, nous avons opté pour une annotation "complète" de cette région. Ainsi toutes les informations nécessaires pour la reproduction du tableau sont fournies en décrivant plusieurs niveaux de zones à l'aide de plusieurs attributs :

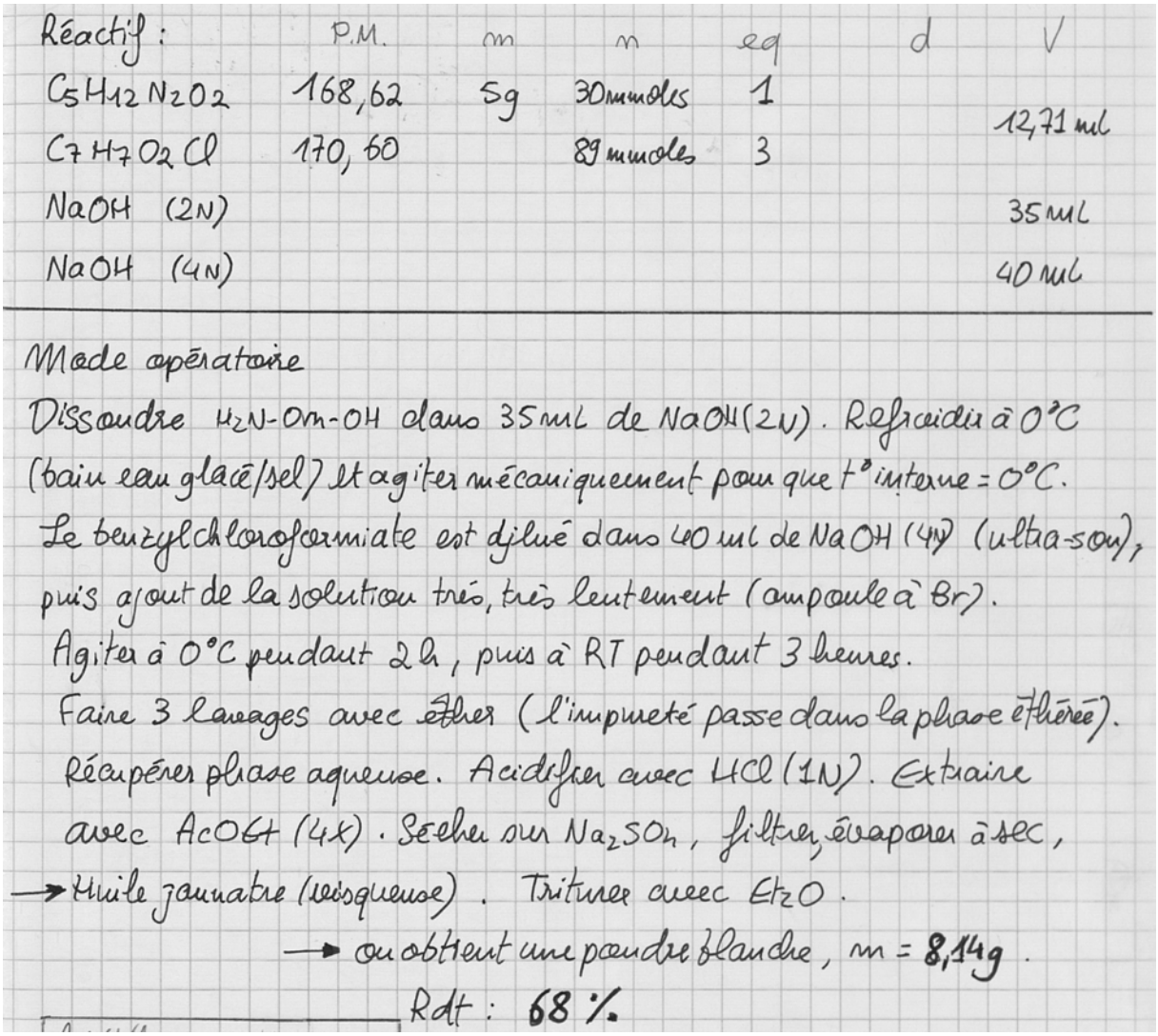
— niveau ligne : chaque ligne dans un tableau est décrite par sa boîte englobante ;
— niveau cellule : chaque cellule d'une ligne est décrite par sa boîte englobante, son numéro de colonne, son type et son contenu
— niveau mot : chaque mot dans une cellule est défini par son contenu et son type ;
— niveau composante connexe (CC) : chaque CC est défini par sa boîte englobante, son contenu et son effet. L'attribut effet prend sa valeur parmi la liste des valeurs suivantes "subscript" (indice), "superscript" (exposant), "normal" ou "crossed" (barré). Cet attribut est important dans un contexte de chimie ou les noms de produits chimiques contiennent généralement des indices et les montants peuvent contenir des exposants.

La figure 2.7 montre un exemple de fichier d'annotation d'un tableau.

A partir de l'annotation des composantes connexes, nous avons collecté celles qui sont des chiffres pour constituer une base annotée de chiffres isolés. Cette base est créée pour servir dans des tâches de reconnaissance ultérieure. Plus précisément, nous l'avons utilisé pour entraîner les différents classifieurs utilisés lors de l'étape d'extraction de chaînes numériques dans les tableaux. Elle est composée des images de chiffres stockés dans des fichiers de format tif dont les noms sont de la forme ID_NUM où ID est un numéro d'ordre et NUM correspond à la valeur du chiffre image (voir Figure 2.8).

2.4 Conclusion

Dans ce chapitre, nous avons présenté dans un premier temps les caractéristiques et la structure des documents de chimie. Cette description permet d'expliciter les différents aspects à considérer dans le système à développer et de guider le choix des méthodes. Ensuite, nous avons décrit la base ChemistryDB composée d'un corpus de documents représentatifs avec la vérité terrain associée. Elle contient toutes les données de vérité dont nous avons besoin pour l'apprentissage et l'évaluation des différentes phases de notre système.



Réactif : P.M. m n eq d V

$C_5H_{12}N_2O_2$ 168,62 5g 30 mmoles 1 12,71 mL

$C_7H_7O_2Cl$ 170,60 89 mmoles 3

NaOH (2N) 35 mL

NaOH (4N) 40 mL

Mode opératoire

Dissoudre H_2N-Orn-OH dans 35 mL de NaOH (2N). Refroidir à 0°C (bain eau glacé/sel) et agiter mécaniquement pour que t° interne = 0°C.

Le benzylchloroformiate est dilué dans 40 mL de NaOH (4N) (ultra-son), puis ajout de la solution très, très lentement (ampoule à Br).

Agiter à 0°C pendant 2h, puis à RT pendant 3 heures.

Faire 3 lavages avec éther (l'impureté passe dans la phase éthérée).

Récupérer phase aqueuse. Acidifier avec HCl (1N). Extraire avec AcOEt (4x). Sécher sur Na_2SO_4, filtrer, évaporer à sec,

→ Huile jaunatre (visqueuse). Triturer avec Et_2O.

→ on obtient une poudre blanche, m = 8,14g.

Rdt : 68 %

(a)

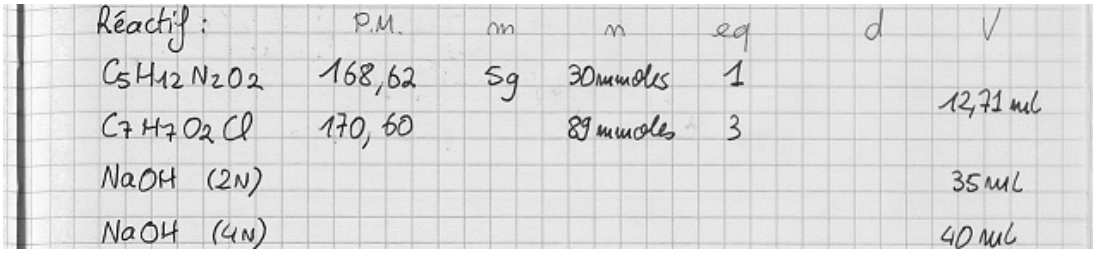
```
<GEDI xmlns="http://lamp.cfar.umd.edu/media/projects/GEDI/" GEDI_version="2.4" GEDI_date="07/29/2013">
    <USER name="nghanmi" date="4/13/2016 16:49" dateFormat="mm/dd/yyyy hh:mm">  </USER>
    <DL_DOCUMENT src="11413.tif" NrOfPages="1" docTag="xml">
        <DL_PAGE gedi_type="DL_PAGE" src="11413.tif" pageID="1" width="2368" height="2132">
            <DL_ZONE gedi_type="TableRow" id="19" col="81" row="20" width="2100" height="111" segmentation="word"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="129" col="81" row="20" width="213" height="111"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="130" col="677" row="38" width="92" height="48"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="128" col="988" row="60" width="62" height="29"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="126" col="1260" row="66" width="47" height="28"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="127" col="1510" row="62" width="88" height="66"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="131" col="1866" row="28" width="42" height="74"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="132" col="2140" row="27" width="41" height="75"> </DL_ZONE>

                                        ⋮

            <DL_ZONE gedi_type="TextLine" id="2" col="840" row="2021" width="433" height="83" segmentation="word"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="24" col="840" row="2022" width="127" height="82"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="25" col="1083" row="2025" width="86" height="77"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="26" col="1199" row="2021" width="74" height="77"> </DL_ZONE>
        </DL_PAGE>
    </DL_DOCUMENT>
</GEDI>
```

(b)

FIGURE 2.6 – Exemple d'annotation d'une imagette contenant du texte, (a) l'image du texte, (b) le fichier d'annotation correspondant à la première ligne et la dernière ligne.



Réactif :	P.M.	m	n	eq	d	V
$C_5H_{12}N_2O_2$	168,62	5g	30mmoles	1		12,71 mL
$C_7H_7O_2Cl$	170,60		89 mmoles	3		
NaOH (2N)						35 mL
NaOH (4N)						40 mL

(a)

```
<DL_DOCUMENT src="imgTable.tif" NrOfPages="1" docTag="xml">
    <DL_PAGE gedi_type="DL_PAGE" src="imgTable.tif" pageID="1" width="2535" height="1000">

        <DL_ZONE gedi_type="TableRow" id="1" col="221" row="0" width="2100" height="115" > </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="2" col="221" row="0" width="213" height="113" Tcol="1" Content="Reactif"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="3" col="814" row="18" width="95" height="51" Tcol="2" Content="P.M"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="4" col="1124" row="41" width="64" height="29" Tcol="3" Content="m"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="5" col="1397" row="48" width="48" height="29" Tcol="4" Content="n"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="6" col="1650" row="42" width="86" height="68" Tcol="5" Content="eq"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="7" col="2002" row="9" width="44" height="75" Tcol="6" Content="d"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="8" col="2277" row="6" width="43" height="78" Tcol="7" Content="v"> </DL_ZONE>


        <DL_ZONE gedi_type="TableRow" id="9" col="233" row="124" width="1491" height="78" > </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="10" col="233" row="125" width="373" height="76" Tcol="1" Content="C5H12N2O2"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="11" col="732" row="125" width="226" height="77" Tcol="2" Content="168,62"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="12" col="1120" row="142" width="78" height="60" Tcol="3"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="13" col="1120" row="142" width="42" height="48" contents="5"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="14" col="1169" row="156" width="28" height="45" contents="g"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="15" col="1312" row="138" width="236" height="59" Tcol="4"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="16" col="1313" row="138" width="64" height="52" contents="30"> </DL_ZONE>
                <DL_ZONE gedi_type="Patch" id="17" col="1378" row="138" width="170" height="59" ="mmoles"> </DL_ZONE>
                    <DL_ZONE gedi_type="CC" id="18" col="1377" row="160" width="41" height="31" contents="m" effect="normal"> </DL_ZONE>
                    <DL_ZONE gedi_type="CC" id="19" col="1422" row="163" width="35" height="26" contents="m" effect="normal"> </DL_ZONE>
                    <DL_ZONE gedi_type="CC" id="20" col="1462" row="137" width="87" height="62" contents="oles" effect="normal"> </DL_ZONE>
            <DL_ZONE gedi_type="Cell" id="53" col="1672" row="130" width="52" height="60" Tcol="5" Content="1"> </DL_ZONE>

                                    ⋮

    </DL_PAGE>
</DL_DOCUMENT>
```

(b)

FIGURE 2.7 – Exemple d'annotation d'un tableau, (a) l'image du tableau, (b) le fichier d'annotation correspondant aux deux premières lignes du tableau.

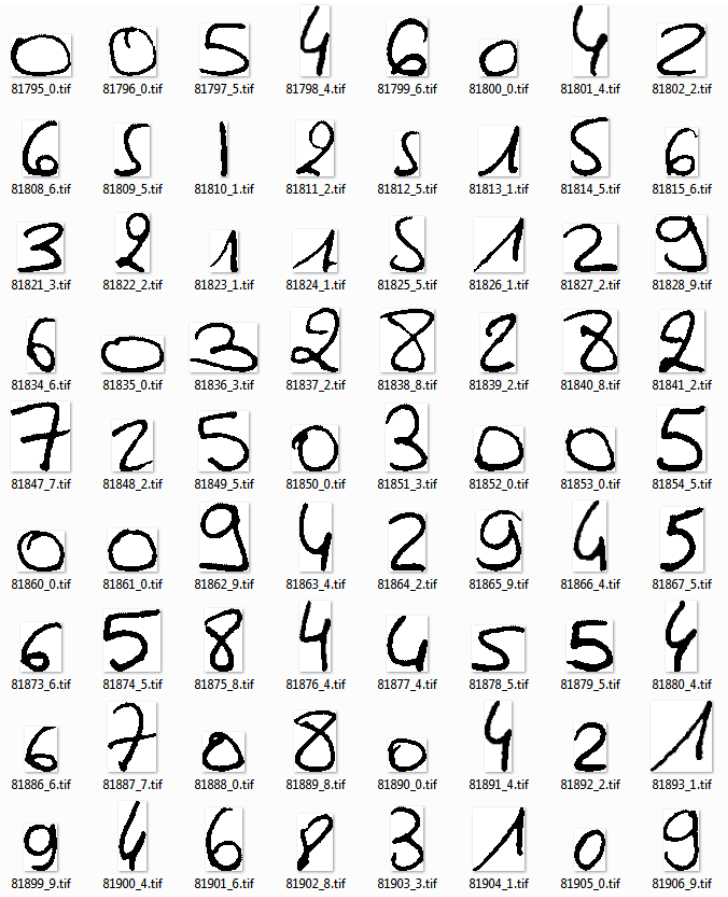


FIGURE 2.8 – Exemples de chiffres extraits des documents de chimie

Chapitre 3

Etat de l'art

Sommaire

3.1 Introduction 25
3.2 Séparation texte/graphique 26
3.2.1 Les méthodes basées sur les composantes connexes 26
3.2.2 Les méthodes de zonage 27
3.3 Segmentation en lignes 28
3.3.1 Méthodes de projection 29
3.3.2 Transformée de Hough 30
3.3.3 Méthodes de lissage 31
3.3.4 Méthodes de regroupement 32
3.4 Extraction de tableaux 33
3.4.1 Détection de tableaux 34
3.4.2 Reconnaissance de tableaux 38
3.5 Extraction de champs numériques 43
3.5.1 Localisation de champs numériques 43
3.5.2 Reconnaissance de chiffres connectés 47
3.6 Conclusion 51

3.1 Introduction

La chaîne d'analyse de documents images en vue de leur conversion en format électronique inclut différents niveaux de traitements : pré-traitement, segmentation, analyse et reconnaissance. Ces traitements permettent respectivement de restaurer ou nettoyer l'image, d'extraire les différentes structures qui la composent, telles que les illustrations (images, courbes, diagrammes, etc.), les tableaux, les lignes de texte, les mots, etc. et de reconnaître l'ensemble (ou une partie) du contenu de ces régions. Dans ce chapitre, nous décrivons les principaux travaux existants dans la littérature qui portent sur les traitements suivants :

1. la séparation texte/graphique ;
2. l'extraction de tableaux ;
3. la segmentation en lignes ;
4. l'extraction de chaînes numériques.

3.2 Séparation texte/graphique

La séparation texte/graphique consiste à séparer le contenu d'une image d'un document en deux couches : le graphique et le texte. La couche graphique est composée de différents éléments tels que des lignes ou toute autre forme géométrique (cercle, polygone, etc.), les diagrammes, les courbes, etc. Quant à la couche textuelle, elle contient principalement des lettres et des chiffres regroupés en chaînes de caractères. De nombreuses méthodes ont été proposées pour résoudre le problème de séparation texte/graphique dans les images de documents. Ces méthodes consistent à analyser des éléments de l'image et à affecter à chacun un label texte ou graphique. La granularité de l'élément analysé dépend de la nature des documents (manuscrit ou imprimé) et de leurs contenus. Nous distinguons alors deux types de méthodes : le premier regroupe les méthodes basées sur les composantes connexes et le deuxième regroupe les méthodes basées sur des zones (de taille fixe ou variable) de l'image.

3.2.1 Les méthodes basées sur les composantes connexes

L'idée sous-jacente à ces méthodes est que les dimensions et la morphologie des composantes connexes graphiques sont différentes de celles textuelles.

Une des principales méthodes dans ce groupe est celle de Fletcher et al. [Fletcher1988] qui se base sur le ratio des dimensions des composantes connexes pour filtrer les composantes de grosses tailles qui sont probablement des graphiques. Ensuite, on utilise une transformée de Hough en considérant comme points votants les centres des boîtes englobantes des composantes connexes, pour détecter les composantes colinéaires et les regrouper en chaînes de caractères. Les composantes isolées sont considérées comme graphiques. Cette méthode présente l'avantage de pouvoir séparer le texte du graphique même dans des documents complexes contenant du texte de différentes orientations englobées dans des graphiques de différentes formes. Cependant, comme indiqué dans [Tombre2002], cette méthode présente des limites quant à la séparation des composantes textuelles collées aux graphiques.

Quelques améliorations ont été proposées dans [Tombre2002, Roy2010] pour pallier cette limite. En se basant sur l'hypothèse que le texte est généralement présent sous forme de chaînes et non pas de caractères isolés et que la chaîne n'est pas entièrement collée au graphique, Tombre et al. [Tombre2002] déterminent l'orientation de la chaîne à partir des boîtes englobantes des caractères déjà retrouvés pour définir une zone de recherche d'éventuels caractères collés au graphique. Les caractères appartenant à cette zone, sont ensuite détachés du graphique en segmentant le squelette et en reconstituant chaque partie à part. Récemment, Roy et al. [Roy2010] ont développé une méthode basée sur les descripteurs invariants SIFT pour détecter les caractères touchant le graphique. L'idée consiste à labéliser les caractères isolés, détectés par une analyse des composantes connexes [Fletcher1988], en utilisant SIFT. Le système apprend au fur et à mesure les différentes formes de chaque caractère. Ensuite, les images de ces caractères sont utilisées comme requêtes pour extraire des caractères similaires collés au graphique.

Une autre méthode basée sur la classification des composantes connexes en vue de la séparation de texte/non-texte dans les documents composites est présentée dans [Barlas2014]. Cette classification est réalisée avec un perceptron multicouches (PMC) en utilisant un ensemble de descripteurs géométriques extraits de la composante connexe. En plus des descripteurs individuels, d'autres caractéristiques contextuelles telles que la taille et le positionnement relatif des composantes voisines sont aussi utilisées. Les tests effectués sur des documents extraits de la base de la première campagne MAURDOR ont donné un taux avoisinant 83% de composantes connexes correctement classées.

Dans [Mollah2009], les auteurs proposent une classification à base de règles afin de séparer les composantes textuelles des autres composantes (logo, image, texture, graphique). Des descripteurs structurels tels que la hauteur, la largeur, le ratio hauteur/largeur, la densité de pixels, le nombre de segments horizontaux et verticaux et le nombre de transitions sont utilisés pour caractériser les composantes connexes.

En se basant sur plusieurs heuristiques, des seuils adaptatifs sont estimés pour la conception des règles de classification. Ces règles décrivent les principales propriétés des composantes connexes textuelles telles que la régularité des dimensions et la faible densité en pixels par rapport aux graphiques. La méthode a été testée sur 100 images de cartes de visite sous différentes résolutions. Les meilleurs résultats obtenus sont de $98,5\%$ de bonne séparation de composantes connexes.

Il convient de noter que toutes ces méthodes ont été testées sur des documents imprimés. Elles ne peuvent pas être appliquées sur les documents manuscrits puisque les caractéristiques géométriques et morphologiques des composantes graphiques et textuelles ne sont pas nécessairement différentes.

3.2.2 Les méthodes de zonage

Dans le cas où des composantes graphiques et textuelles se touchent fréquemment, ce qui peut générer des composantes hétérogènes, il est intéressant d'étiqueter des zones au lieu des composantes connexes. Par exemple, dans [Jang2005], les auteurs proposent une méthode pour isoler les informations texte de celles qui sont non-texte dans les images des cartes de visite. D'abord, une séparation entre le fond et l'avant-plan de l'image est effectuée en classifiant des blocs de tailles fixes (8×8). Les coefficients de la transformée en cosinus discrète à basse résolution et la densité de pixels sont utilisées pour cette classification. Ensuite, une deuxième classification en texte ou non-texte, en utilisant le même principe, est effectuée sur les blocs de l'avant-plan.

Dans [Journet2005], une approche basée sur l'estimation des orientations caractéristiques dans des zones de l'image est proposée pour extraire du graphique dans des documents anciens. L'image du document est parcourue en utilisant une fenêtre glissante. Sur chaque zone délimitée par la fenêtre, l'orientation caractéristique de l'image est extraite en utilisant une fonction d'auto-corrélation. Cette fonction consiste à combiner l'image avec elle-même mais translatée d'un vecteur (i, j) afin de détecter les périodicités et les orientations dans sa texture. Les pixels de l'image ayant la même direction vont être situés sur la même ligne permettant ainsi de saillir les directions importantes. Par rapport à un bloc graphique, un bloc de texte est caractérisé par une grande régularité et une direction horizontale privilégiée. Ainsi, une zone est considérée comme graphique si la direction horizontale n'a pas été détectée en tant que l'une de ses directions privilégiées.

Dans [Sarkar2011], les auteurs proposent une méthode basée sur la classification des composantes connexes obtenues en appliquant sur l'image binaire une version modifiée de la méthode Run Length Smoothing Algorithm (RLSA) appelé RLSA spiral. A la différence d'un RLSA classique, qui permet de connecter les pixels proches en balayant l'image horizontalement puis verticalement, le RLSA spiral effectue un balayage en spiral pour connecter les pixels voisins. L'utilisation de cette version est justifiée par le fait que le RLSA classique présente des limites sur les documents manuscrits du fait qu'ils ont une densité de pixels faible. De plus, l'application d'un RLSA classique échoue, dans certaines configurations, à connecter deux pixels bien qu'ils soient proches (voir Figure 5.1). Les composantes connexes de l'image résultat du RLSA spiral (voir Figure 3.2) sont ensuite classées en utilisant un SVM à deux classes : texte ou graphique. Pour ce faire, des descripteurs géométriques de dimensions et de distribution de pixels ont été extraits sur ces composantes.

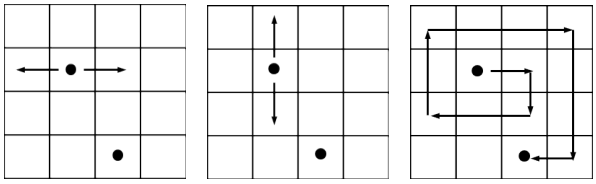


FIGURE 3.1 – (a) RLSA horizontal, (b) RLSA vertical et (c) RLSA spiral.

Cette méthode est sensible au bruit de numérisation surtout que les auteurs n'ont pas ignoré le bruit (dont ils n'ont pas réussi à supprimer lors de l'étape de pré-traitement) dans les documents d'apprentissage mais ils l'ont considéré comme du graphique. Ceci a conduit à des erreurs de classification des ponctuations et des fragments de mots ou de caractères qui peuvent ressembler à du bruit. De plus, le balayage en spiral permet de regrouper le maximum des pixels voisins dans tous les sens. Ce traitement peut donner de bons résultats si on fait l'hypothèse que deux composantes de types différents sont suffisamment espacées pour qu'elles ne deviennent pas connectées avec l'algorithme RLSA spiral. Cependant, dans les documents manuscrits non contraints, cette hypothèse n'est pas toujours vérifiée et on peut même avoir deux composantes de types différents qui sont connectés. Dans la figure 5.1, même si les deux pixels noirs appartenaient à deux composantes de types différents, ils seront connectés en appliquant un RLSA spiral (Figure 3.1(c)) et la composante résultante (composée de graphique et de texte) aura un seul label texte ou graphique. Néanmoins, avec un RLSA classique, les deux composantes restent séparées (Figures 3.1(a) et 3.1(b)).

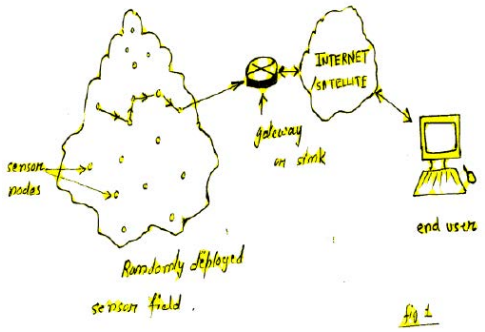


FIGURE 3.2 – Résultat de l'application d'un RLSA spiral sur une image binaire. En noir, les pixels de l'image initiale et en jaune, les pixels ajoutés par le RLSA spiral pour connecter les pixels proches [Sarkar2011].

3.3 Segmentation en lignes

La segmentation en lignes est vue comme une étape de pré-traitement pour différentes tâches d'analyse et de reconnaissance dans les images de documents textuels. Elle constitue une étape cruciale qui conditionne les résultats des étapes ultérieures. De nombreuses méthodes de segmentation des documents imprimés et manuscrits ont été proposées dans [Ouwayed2010, Arivazhagan2007, Papavassiliou2010, Tripathy2004, Li2006].

Dans cette section, nous allons présenter les principales d'entre elles. Nous distinguons quatre types de méthodes : les méthodes de projection, les méthodes utilisant la transformée de Hough, les méthodes de lissage et les méthodes de regroupement.

3.3.1 Méthodes de projection

Cette méthode permet de détecter les lignes de texte en recherchant les pics du profil de projection sur l'axe orthogonal à l'orientation du texte dans le document. Généralement, un maximum dans le profil de projection correspond à une ligne et les minima qui l'entourent correspondent aux espaces inter-lignes (en-dessus et en-dessous). Ainsi, les composantes connexes situées entre ces deux minima sont regroupées dans la même ligne. Selon le type de lignes, droites ou fluctuantes, la projection est effectuée globalement ou par morceaux. Par exemple, dans [Manmatha2005], les auteurs utilisent une projection globale, appliquée sur une image en niveaux de gris, pour extraire les lignes dans les manuscrits de Georges Washington. Dans ces documents, les lignes sont droites et horizontales, régulièrement espacées et présentent peu de chevauchement et de connexions. L'utilisation d'une projection globale s'avère alors efficace. Le profil obtenu est ensuite lissé en utilisant un filtre passe-bas (gaussien) afin d'éliminer les faux pics et de réduire sa sensibilité au bruit. Enfin, les maxima locaux sont obtenus en déterminant les points ou la dérivée du profil s'annule. Ces étapes sont illustrées dans la figure 3.3 extraite de [Manmatha2005].

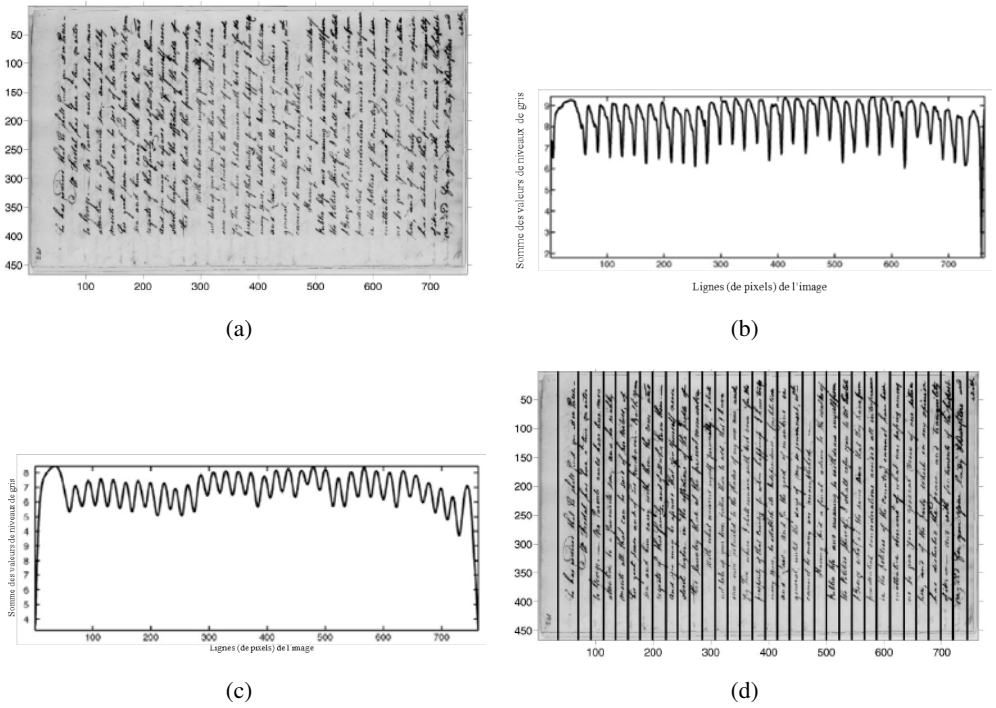
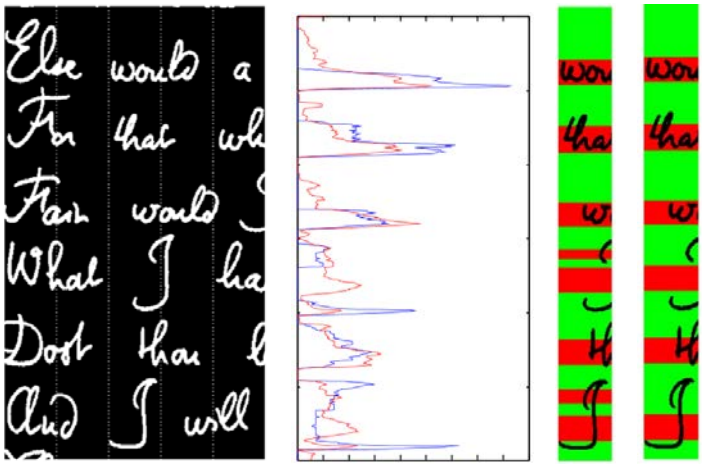


FIGURE 3.3 – Les étapes de segmentation en lignes en utilisant une projection globale. (a) Image du document, (b) Profil de projection, (c) profil de projection lissé et (d) détection de pics et segmentation en lignes.

Quand l'écriture fluctue par rapport à l'horizontale, d'autres variantes de la technique de projection peuvent être utilisées. Dans [Papavassiliou2010], les documents sont d'abord décomposés en bandes verticales suffisamment étroites pour que l'effet d'inclinaison soit négligé mais aussi suffisamment large pour contenir une quantité adéquate de texte (Figure 3.4(a)). Vu que les bandes situées dans les marges gauche et droite du document contiennent généralement peu de texte, elles sont ignorées si leur densité en pixels est inférieure à un certain seuil. Les profils de projection sont ensuite établis pour les autres bandes. Dans certains cas, même les bandes au milieu de la page peuvent ne pas contenir suffisamment de texte dans chaque ligne à cause de la présence de larges espaces inter-mots (voir la troisième ligne

dans la troisième bande de la Figure 3.4(a)). Afin de diminuer l'influence de ces occurrences, le profil de chaque bande est lissé en considérant la moyenne des profils de bandes voisines, à gauche et à droite (Figure 3.4(b)). Les maxima et les minima de chaque profil sont extraits pour délimiter grossièrement les lignes (voir Figure 3.4(c)). Pour raffiner cette détection, les auteurs utilisent un modèle de Markov caché (MMC) pour modéliser la succession de textes et d'espaces inter-lignes. L'algorithme de Viterbi est utilisé pour trouver la meilleure succession. Une approche similaire est présentée dans [Arivazhagan2007] où 20 bandes verticales de largeur fixe (5% de la largeur de la page) sont utilisées. Les profils de chaque bande sont lissés en utilisant un filtre moyenneur de longueur 5. Un ensemble de règles est utilisé pour construire les bords des lignes en connectant les vallées des profils de bandes voisines. Les fragments de texte intersectés par ces bords sont affectés à la ligne au-dessus ou en-dessous en utilisant deux méthodes probabilistes ; une basée sur la densité des pixels dans les lignes et l'autre, sur les distances.

Une autre méthode de projection [Marti2001] utilise les transitions noir-blanc au lieu des pixels. Pour chaque ligne de l'image, le nombre de transitions est compté et l'histogramme correspondant est construit. Il est ensuite lissé en utilisant un filtre médian. Les bords (haut et bas) des lignes de texte correspondant aux minima de l'histogramme, sont déterminés. Selon la position de leur centre de gravité, les composantes connexes intersectées par les bords des lignes sont affectées à la ligne au-dessus ou en-dessous.



(a) (b) (c) (d)

FIGURE 3.4 – (a) Image du document découpée en 5 bandes verticales , (b) en bleu, le profil de projection de la troisième bande et en rouge le profil lissé correspondant, (c) détection des lignes de texte et d'espaces inter-lignes en se basant sur le profil, et (d) détection raffinée en utilisant un MMC.

3.3.2 Transformée de Hough

La transformée de Hough est une technique permettant de détecter des formes géométriques bien précises dans une image. L'application la plus simple consiste à détecter les lignes. Un ensemble de points (x, y) appartenant à une droite d'équation $y = ax + b$ seront associés au même couple (a, b) dans l'espace des paramètres, ce qui crée une zone d'accumulation. Un point (x, y) dans l'espace réel est exprimé dans l'espace des paramètres à l'aide de l'équation paramétrique suivante :

$$\rho = x\cos(\theta) + y\sin(\theta) \tag{3.1}$$

où θ dénote l'angle d'inclinaison par rapport à l'axe des abscisses et ρ, la distance de la droite par rapport à l'origine du système de coordonnées (x, y). Ainsi, la forme associée à un point est une sinusoïde dans

l'espace des paramètres. Des points alignés dans l'espace réel se traduisent, dans l'espace des paramètres, par l'intersection de leurs sinusoïdes respectives.

Cette méthode a été utilisée par Likforman-Sulem et al. [Likforman-S1995] pour extraire les lignes de texte dans des documents manuscrits. Pour étudier l'alignement des composantes connexes, leurs centres de gravité sont projetés dans l'espace des paramètres. Dans cet espace, chaque zone d'accumulation correspond à des composantes connexes appartenant à une même ligne. Les lignes ainsi extraites sont validées en analysant les distances entre les composantes connexes voisines situées à l'intérieur et à l'extérieur de chaque ligne.

Dans [Louloudis2009], les auteurs ont également utilisé une méthode basée sur la transformée de Hough pour extraire les lignes dans les documents manuscrits grecs. D'abord, les composantes connexes utilisées pour étudier les alignements sont sélectionnées selon un critère de taille. Celles qui sont de petite taille (bruit, accents, etc.) ou de grande hauteur, sont écartées, vu qu'elles perturbent l'alignement au sein d'une ligne de texte. L'alignement des autres composantes est étudié à l'aide de la transformée de Hough. Chaque composante est partitionnée en plusieurs blocs de largeur fixe (égale à la largeur moyenne d'un caractère, estimée sur le document entier) et le centre de chaque bloc est utilisé comme point votant. Ainsi, une composante connexe est représentée par un nombre de points votants proportionnel à sa largeur. Pour décider qu'une composante appartienne à une ligne de texte, au moins la moitié des points représentant ses blocs doivent avoir voté pour cette ligne. En post-traitement, les fragments d'une même ligne sont regroupés en examinant leur intersection. Les composantes écartées dans l'étape précédente sont assignées à la plus proche ligne de texte. Cette méthode a été testée sur un ensemble de 20 documents grecs contenant 450 lignes et a donné un taux de bonne détection est de 96%.

La transformée de Hough a été également utilisée pour extraire les lignes dans des documents manuscrits de différents types (lettres, notes, etc.)[Malleron2009]. D'abord, les composantes connexes sont extraites et le voisinage de chacune (dans 18 directions) est examiné afin de déterminer celles qui sont situées sur l'un des 4 bords du document (haut, bas, gauche et droite). Ensuite, le contour de chaque composante connexe est déterminé à l'aide du filtre de Canny. La transformée de Hough est appliquée sur ce contour afin de déterminer l'orientation locale de la composante connexe correspondante. A l'issue de cette étape, une carte d'orientation des segments représentant les contours est construite. Cette carte est balayée de droite à gauche afin de regrouper les contours qui ont la même orientation tout en tolérant de petites variations. Les composantes connexes dont les contours appartiennent à un même groupe sont affectées à une même ligne de texte. La méthode a été testée sur un ensemble de 280 documents. Les performances obtenues varient entre 70% et 96% de rappel et entre 78% et 96% de précision selon la complexité de la structure des documents.

Une approche similaire est proposée dans [Saha2010] en utilisant un filtre de sobel pour la détection des contours des composantes connexes. La transformée de Hough est appliquée sur l'image binarisée de la carte de contours. La méthode a été expérimentée sur des images de documents manuscrits et imprimés et sur des images de carte de visite. Les auteurs illustrent quelques exemples de résultats sans pour autant donner les taux de segmentation obtenus sur l'ensemble des documents.

3.3.3 Méthodes de lissage

Plusieurs travaux basés sur la technique de lissage [Li2006, Shi2004] ont été menés pour la segmentation en lignes des documents manuscrits. Pour faire ressortir la structure des lignes, Li et al. [Li2006] utilisent une méthode de lissage gaussien qui convertit une image binaire en une image en niveaux de gris. Une fenêtre rectangulaire est adoptée pour effectuer ce lissage. La hauteur de la fenêtre est plus petite que la valeur moyenne de l'espace inter-lignes pour que ces dernières restent séparées. Sa largeur est relativement grande de sorte que les espaces inter-mots et inter-caractères soient comblés. Ainsi, dans l'image obtenue, les pixels ont une intensité faible sur les lignes de texte et une intensité élevée dans

les espaces verticaux. Ce traitement permet de brouiller les détails des traits manuscrits et d'améliorer les structures des lignes de texte, ce qui les rend indépendant du scripteur. Une première estimation des lignes est effectuée en binarisant l'image lissée. Néanmoins, cette estimation peut conduire à des fragments de lignes. Pour constituer des lignes plus complètes, une méthode de croissance de régions, basée sur la technique des surfaces de niveaux [Osher2003], a été utilisée. A la fin du traitement, les orientations et les longueurs des lignes détectées sont étudiées afin de fusionner celles qui se rapportent à une même ligne. La difficulté inhérente dans cette méthode réside dans le choix des seuils de lissage horizontal et vertical.

Comme solution à ce problème, un RLSA flou a été proposé dans [Shi2004]. Partant d'une image binaire, une image en niveaux de gris de même taille, est construite. La valeur d'un pixel dans cette image correspond à la distance qui le sépare du $n$ème pixel noir dans la direction horizontale ou verticale, où n est un paramètre de l'algorithme. L'image obtenue est ensuite binarisée par seuillage global. En utilisant des règles basées sur la hauteur, la longueur et la densité des pixels des composantes connexes de l'image binaire, les lignes sont identifiées. La technique de lissage fonctionne bien sur les documents où les espaces et les tailles des caractères présentent une certaine régularité. Dans le cas contraire, d'autres versions de RLSA peuvent être utilisées. Par exemple, dans [Nikolaou2010], les auteurs ont proposé une version modifiée du RLSA horizontal, appelé RLSA adaptatif (ARLSA) afin d'empêcher le regroupement dans la même ligne des caractères de tailles nettement différentes. L'idée principale d'un ARLSA est de connecter deux composantes connexes si elles vérifient un ensemble de contraintes géométriques relatives à leurs dimensions et à leurs positions. Plus précisément, ces contraintes portent sur le ratio de leurs tailles, la distance horizontale qui les sépare et le chevauchement vertical de leurs boîtes englobantes. Grâce à ces conditions, les fragments de lignes inclinées qui peuvent être proches, ne sont pas rassemblés dans la même ligne. Dans [Zhao2010], les auteurs adoptent une méthode similaire mais en effectuant un ARLSA horizontal et un ARLSA vertical. Sur chacune des images résultats, des opérations morphologiques d'ouverture sont effectuées afin de supprimer les traits isolés. Ensuite, les profils de projections des images lissées sont analysés et les lignes sont extraites.

3.3.4 Méthodes de regroupement

Les méthodes de regroupement consistent à rassembler les composantes connexes en lignes en se basant sur des heuristiques comme dans [Likforman-Sulem1994, Saabni2011], ou des mesures de distances comme dans [Simon1997] ou des méthodes issues des travaux de l'intelligence artificielle [Nicolas2004].

Dans [Likforman-Sulem1994], une méthode basée sur la physiologie de la vision humaine est proposée pour extraire les lignes dans des images de documents manuscrits de différents types. Pour cela, l'orientation locale de chaque composante connexe est estimée en appliquant 4 masques (correspondant aux 4 directions principales). Si la réponse de la composante à un masque est élevée (90%), cette première est considérée comme un germe d'une ligne ayant la direction de ce masque. En se référant aux principes de regroupement du système visuel qui se base sur les critères de proximité et de continuité d'orientation, les composantes connexes proches d'un germe et ayant son orientation lui sont liés. Ceci peut conduire à des conflits dans les orientations de quelques composantes puisqu'elles peuvent être liées à plusieurs germes. Un ensemble de règles basées sur la qualité de l'alignement, la configuration spatiale de ses éléments et la continuité de l'orientation est utilisé pour fusionner, scinder, étendre ou supprimer des alignements. Les auteurs ne fournissent pas une évaluation quantitative mais se contentent de comparer qualitativement les résultats qu'ils ont obtenus avec d'autres techniques telles que la transformée de Hough et la technique de lissage.

Récemment, Saabni et El-Sana[Saabni2011] ont proposé une méthode basée sur le regroupement des composantes connexes en déterminant les chemins de plus basses énergies traversant le document de gauche à droite. La procédure commence par déterminer la carte d'énergie correspondante à l'image

binaire en utilisant la technique de transformée en distance signée (TDS). Avec une TDS, les pixels situés à l'intérieur des composantes sont négatifs et ceux à l'extérieur sont positifs. Ainsi, suivre les minima locaux dans la carte d'énergie obtenue par TDS conduit à un chemin qui traverse les composantes connexes de la même ligne. Un algorithme de programmation dynamique est adopté pour effectuer ce suivi. Les composantes connexes regroupées dans une même ligne sont utilisées pour estimer la moyenne et l'écart type de la hauteur de la ligne. Les grosses composantes connexes -leurs hauteurs étant nettement supérieures à la hauteur moyenne- sont considérées comme deux composantes collées et sont alors segmentées au milieu. Les composantes connexes n'ayant pas été traversées par aucun chemin, sont affectées aux lignes les plus proches. Les lignes ainsi formées sont marquées et les composantes qui y appartiennent ne sont pas considérées dans les prochaines itérations. Ce traitement est répété jusqu'à ce qu'il ne reste plus aucune composante connexe. L'approche a été testée sur 50 documents manuscrits de différentes langues (anglais, français, allemand et grec) extraits de la base ICDAR2007 (utilisée dans la compétition de segmentation de l'écriture manuscrite) et un taux de bonne segmentation de 98% a été obtenu. Une autre base privée contenant 50 documents manuscrits arabes et chinois a été aussi utilisée et les résultats sont aux alentours de 99%.

Dans [Simon1997], les auteurs proposent une méthode ascendante pour l'extraction complète de la structure physique des documents imprimés. Cette méthode consiste à déterminer l'arbre couvrant minimal du graphe connectant toutes les composantes connexes dans une page. Les composantes connexes étant extraites, un graphe connexe et non-orienté dont les arcs sont pondérés par la distance séparant les noeuds qu'il connecte est construit. L'algorithme de Kruskal [Aho1983] est utilisé pour générer l'arbre couvrant minimal à partir de ce graphe. Il s'agit d'un algorithme glouton qui construit une forêt dont tous les éléments finissent par fusionner pour former un arbre couvrant. L'utilisation de cet algorithme est justifiée par le fait qu'il existe une correspondance un-à-un entre les éléments physiques de la page (caractère, mot, ligne, bloc de texte, etc.) et les sous-arbres générés. Les éléments obtenus sont classés au niveau mot, en texte ou graphique. Ensuite, les mots de texte sont regroupés en une même ligne s'ils présentent un chevauchement vertical important. Les tests ont été effectués sur 98 documents extraits de livres, rapports et journaux. Les résultats obtenus reflètent une bonne performance avec un taux d'erreur de 1% sur l'ensemble de blocs (environ 2000 blocs).

Nicolas et al. [Nicolas2004] ont adopté une méthode basée sur les systèmes de production pour regrouper les composantes connexes en lignes. Dans cette méthode, la base de faits est représentée sous forme d'un graphe dont les nœuds représentent les composantes connexes ou les lignes et les arcs représentent la distance euclidienne entre une paire de nœuds adjacents. A partir de ce graphe, l'état initial est généré aléatoirement en sélectionnant une composante connexe comme germe d'une ligne de texte. L'état final, correspondant à la fin de la segmentation, est décrit intuitivement comme un état où les alignements sont stabilisés, c'est-à-dire toute composante connexe est affectée à exactement une seule ligne. Les lignes sont construites incrémentalement en considérant les composantes connexes adjacentes à une ligne et en appliquant une des deux opérations "fusionner" ou "ne pas fusionner". Une probabilité d'application de chaque opération est définie en fonction de la distance euclidienne entre la composante connexe et la ligne étudiée. En cas de non-fusion, la composante connexe constitue le germe d'une nouvelle ligne. Le processus de recherche est répété jusqu'à aucune modification ne soit effectuée dans la configuration des composantes connexes. Les auteurs se sont concentrés sur l'aspect conceptuel de la méthode afin de prouver sa généricité et son efficacité sans accorder beaucoup d'intérêt à son implémentation et son évaluation. Ils affirment alors que les premiers tests sont effectués sur un petit ensemble de documents sans décrire quantitativement la base utilisée et les résultats obtenus. Les limites de cette méthode sont principalement dues à l'insuffisance de la distance euclidienne pour modéliser un alignement.

3.4 Extraction de tableaux

L'extraction d'un tableau dans un document image peut-être divisée en deux tâches : la détection de tableau et la reconnaissance de tableau. La première consiste à vérifier l'existence d'un tableau dans un document image et déterminer sa position et ses frontières. Quant à la deuxième, elle traite un tableau déjà identifié dans l'image d'un document et cherche à déterminer sa structure physique et logique et à reconnaitre son contenu. Un état de l'art récent des méthodes d'analyse de tableaux et de formulaires dans différents types de documents (images ou ASCII) peut être trouvé dans [Couasnon2014].

Dans cette section, nous présentons les principales méthodes existantes dans la littérature traitant de la détection et de la reconnaissance de tableaux.

3.4.1 Détection de tableaux

Les méthodes existantes dans la littérature peuvent être réparties en trois classes :
— Méthodes basées sur les lignes graphiques.
— Méthodes basées sur les espaces.
— Méthodes basées sur l'alignement vertical.

3.4.1.1 Méthodes basées sur les lignes graphiques

Quand les lignes séparatrices des cellules d'un tableau sont présentes, elles sont souvent utilisées comme révélateurs de la région de tableau [Gatos2005, Kasar2013, Chen2013]. Le problème revient alors à détecter et à analyser des lignes graphiques.

Une des premières méthodes basées sur les lignes graphiques pour la détection de tableaux est celle présentée dans [Hirayama1995]. D'abord, l'image est lissée à l'aide de l'algorithme RLSA et les composantes connexes résultantes sont extraites. Celles-ci sont classées selon les hauteurs de leurs boîtes englobantes en chaînes de caractères, lignes horizontales et verticales ou autres objets. L'histogramme des hauteurs de chaînes de caractères ainsi que l'histogramme des distances entre leurs lignes de base sont établis en vue d'un regroupement des chaînes en blocs de texte. Ainsi, l'image est séparée en deux couches texte et non-texte. Dans la couche non-texte, une classification tableau/image est effectuée. Pour cela, les lignes graphiques dans cette couche sont regroupées en se basant sur un ensemble de règles. La zone occupée par chaque groupe de lignes est considérée comme tableau si l'intersection de ces lignes engendre un ensemble de rectangles qui occupent plus que la moitié de la zone. Pour tester cette méthode, les auteurs ont utilisé 65 pages composites extraites de différents documents japonais (rapports techniques, magazines et livres). Sur les 34 tableaux contenus dans l'ensemble des documents, un rappel de 100% et une précision de 90% sont obtenues. Bien que cette méthode ait montré une efficacité sur ces documents, elle présente une fragilité au niveau de l'étape de classification des composantes connexes. En effet, nous ne pensons pas que la hauteur de la boîte englobante seule permet de décider de la nature d'une composante connexe (ligne, chaîne de caractères ou autre).

Gatos et al. [Gatos2005] proposent une approche en trois étapes : pré-traitement de l'image, détection des lignes horizontales et verticales et enfin, détection des tableaux dans une variété de documents tels que des formulaires, des journaux/magazines, des billets, des chèques de banques, des documents manuscrits, etc. Le pré-traitement inclut la binarisation, l'amélioration de la qualité visuelle de l'image, la correction de l'inclinaison et la suppression du bruit de bords. Pour la détection des lignes, d'abord, un ensemble d'opérations morphologiques, en utilisant des éléments structurants, est appliqué sur l'image afin de connecter les petites coupures. Ensuite, une technique basée sur la détection des séquences de pixels noirs (black runs) ainsi que sur l'estimation des zones de textes et d'images est utilisée pour localiser les lignes graphiques. La détection des tableaux est effectuée en analysant les points d'intersections

des lignes extraites. Ces points sont regroupés horizontalement et verticalement pour être réalignés au niveau de la position moyenne de chaque groupe. Les lignes ainsi reconstruites correspondent aux lignes du tableau. La méthode proposée a été testée sur 102 images de documents. L'évaluation a été effectuée au niveau lignes et le résultat obtenu est aux alentours de 80% de F-mesure.

Dans [Chen2013], les auteurs utilisent une variante probabiliste de la transformée de Hough pour détecter les lignes graphiques dans un document. Ils se basent sur le fait que les lignes d'un tableau sont parallèles ou orthogonales pour exclure celles qui n'appartiennent pas à un tableau. En outre, les lignes appartenant au texte sont ignorées après avoir localisé la zone de texte par un filtrage des composantes connexes basé sur le ratio hauteur/largeur. Les points d'intersection des lignes sélectionnées sont déterminés et analysés afin de délimiter le tableau recherché en se basant sur un tableau modèle. Pour cela, une procédure d'optimisation est employée pour sélectionner le sous-ensemble le plus probable des points d'intersection qui forment la structure du tableau.

Dans [Kasar2013], les auteurs utilisent un algorithme basé sur le run-length pour extraire les lignes droites horizontales et verticales. Les lignes de longueur inférieure à un seuil fixé par rapport à la largeur de l'image $(1/20)$ sont ignorées. L'intersection de 3 (ou plus) lignes est utilisée comme une première indication sur la présence d'un tableau. Cette intersection se présente sous forme d'une grille qui est classée en "tableau" ou "non-tableau" en utilisant un SVM. Pour effectuer cette classification, différentes caractéristiques décrivant la régularité des longueurs et des espacements entre les lignes ainsi que la taille de la grille et le nombre de cases sont extraites et fournies au SVM.

L'inconvénient de ces méthodes est qu'elles ne peuvent pas être appliquées pour détecter des tableaux ne contenant pas de lignes séparatrices.

3.4.1.2 Méthodes basées sur les espaces

Shafait et al. [Shafait2010] ont proposé un algorithme pour la détection de tableaux dans une grande variété de documents ayant différentes structures (rapports d'entreprise, articles de journaux, pages de magazines, etc). Une implémentation de l'algorithme a été intégrée dans le système de reconnaissance Tesseract [2]. Cet algorithme est principalement conçu pour détecter les tableaux dans des documents multi-colonnes. Tout d'abord, les régions de texte sont localisées (par filtrage des composantes connexes) et partitionnées en colonnes de pages en se basant sur les taquets de tabulations. Ensuite, dans chaque colonne, les lignes sont extraites en utilisant une méthode de regroupement. Une ligne qui contient au moins un espace horizontal supérieur à l'espace inter-mots dans un texte normal est considérée comme une ligne de tableau. Enfin, le nombre de colonnes d'un tableau est estimé en examinant le profil de projection vertical de l'ensemble de ses lignes. Si le nombre de colonnes est supérieur à 1, la présence d'un tableau est confirmée. La méthode a été testée sur 214 documents extraits de la base UNLV [3], contenant 268 tableaux. Une mesure d'évaluation basée sur le taux de chevauchement entre les boîtes englobantes des tableaux de la vérité terrain et celles détectées par le système a été adoptée. Les résultats obtenus sont aux alentours de 50% de tableaux correctement détectés, 25% de tableaux partiellement détectés. Les erreurs sont réparties entre des tableaux manqués (environ 17%) et des fausses détections (environ 5%). L'analyse de ces erreurs montre la dépendance de la méthode à la présence de larges espaces, ce qui engendre par exemple des fausses détections au niveau du texte annotant un graphique.

Mandal et al. [Mandal2006] décrivent une méthode simple et efficace de détection de tableaux dans les images de documents imprimés. L'idée principale de cette méthode repose sur le fait que, dans une

2. Tesseract est un logiciel de reconnaissance optique de caractères conçu par les ingénieurs de Hewlett Packard de 1985 à 1995. En 2005, les sources du logiciel sont libérées sous licence Apache et le logiciel est actuellement développé par Google. Initialement limité aux caractères ASCII, il supporte parfaitement les caractères UTF-8 et reconnait maintenant 40 langues.

3. **UNLV** est une base d'images de documents produite par Informations Science Research Institute de l'**U**niversité de **N**évada, **L**as **V**egas , États-Unis.

ligne d'un tableau, l'espace inter-colonnes est supérieur à l'espace inter-mots. Même pour les tableaux avec des lignes graphiques, ces dernières n'ont pas été utilisées dans l'algorithme proposé. En se basant sur les dimensions des composantes connexes, les lignes graphiques sont déterminées et supprimées de l'image. Les autres composantes connexes sont regroupées en lignes de texte en se basant sur leurs chevauchements verticaux. L'histogramme des distances entre les paires de composantes connexes consécutives d'une même ligne est établi afin de déterminer la distance inter-mots. Une fermeture morphologique par un élément structurant "allongé" de longueur égale à l'espace inter-mot permet de construire des "blobs". Tous les mots d'une ligne de texte forment un seul blob alors que dans une ligne d'un tableau plusieurs blobs sont formés. Ainsi, toute ligne contenant plus qu'un blob est sélectionnée candidate pour être une ligne d'un tableau. A cette sélection, est ajoutée toute ligne contenant un seul blob et dont ses deux lignes voisines (en dessus et en dessous) ont été déjà sélectionnées. Chaque ensemble contigu de lignes candidates est considéré comme un tableau. Pour valider cette détection, le profil de projection horizontale de la région de tableau est observé. La présence de plusieurs pics et plusieurs vallées confirme la présence du tableau. La méthode a été testée sur un ensemble de 300 documents extraits des bases publiques UW-I et UW-II [4] . A peu près 48% de ces documents contiennent des tableaux. Malgré la simplicité de l'idée, cette méthode a permis d'obtenir des résultats encourageants avec un taux d'environ 93% de bonne détection de tableaux. Cependant, cette méthode reste sensible à l'irrégularité de la taille du texte et des espaces inter-caractères, inter-mots et inter-lignes ainsi qu'à l'homogénéité du contenu d'un document.

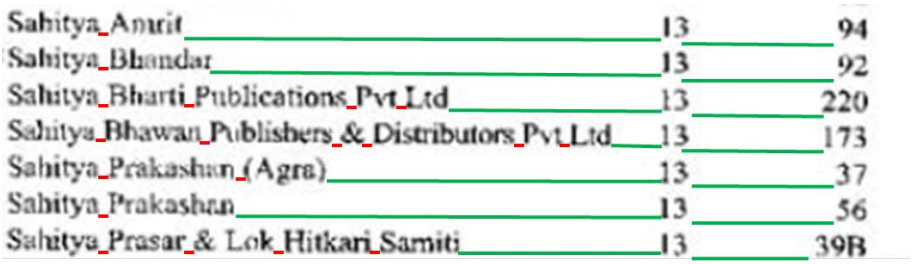


FIGURE 3.5 – Illustration de la différence entre les espaces inter-colonnes (en vert) et les espaces inter-mots (en rouge) dans les lignes d'un tableau.

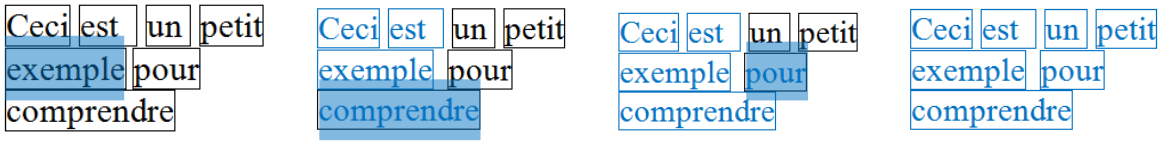
3.4.1.3 Méthodes basées sur l'alignement vertical

Kieninger propose une méthode d'extraction de tableaux basée sur la segmentation du document en blocs [Kieninger1998]. Cette méthode prend en entrée les mots du document et les regroupe en blocs en se basant sur le critère de voisinage vertical : à partir d'un mot arbitraire d'amorce (appelé aussi germe) de bloc, l'algorithme étend de manière récursive ce bloc à tous les mots verticalement voisins de chacun des mots de ce bloc. Cette méthode peut être décrite par les étapes suivantes :

1. Initialiser un nouveau bloc B_i en choisissant arbitrairement un mot ne faisant pas partie d'un autre bloc (ce mot est appelé germe)
2. Déterminer tous les mots M_j qui se chevauchent verticalement dans la ligne précédente et suivante et les ajouter au bloc
3. Refaire l'étape 2 pour tous les mots M_j
4. Si aucun chevauchement n'est trouvé, incrémenter i et aller à l'étape 1.
5. Arrêt si tous les mots font partie de blocs.

4. **UW**-I et **UW**-II sont la première et la deuxième base d'une série de bases publiques d'images de documents produites par le Laboratoire de systèmes intelligents, à l'**U**niversité de **W**ashington, Seattle, Washington, États-Unis

La Figure 3.6 illustre la construction de blocs en suivant les étapes décrites ci-dessus. Admettant l'existence d'un espacement horizontal entre les colonnes du tableau, cette segmentation permet d'identifier et d'isoler ces colonnes. L'ensemble des colonnes définit un tableau. Les auteurs affirment que cette approche fonctionne bien pour des fichiers ASCII et peut être appliquée sur des documents numérisés. Cependant, elle peut générer de fausses détections de tableaux sur les paragraphes dont les lignes présentent des espaces à la même position horizontale. Le même type d'erreur peut aussi avoir lieu sur une ligne isolée : chaque mot sera considéré comme une colonne. De plus, une ligne de texte située en-dessus ou au-dessous d'un tableau peut être à l'origine de non détection du tableau en regroupant toutes les colonnes en un seul bloc.

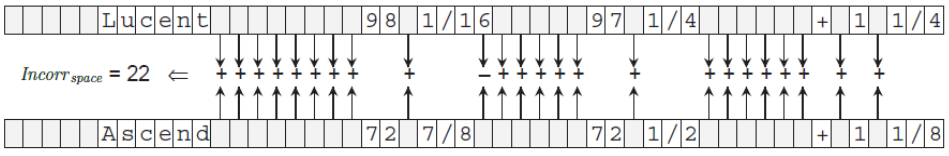


(a) (b) (c) (d)

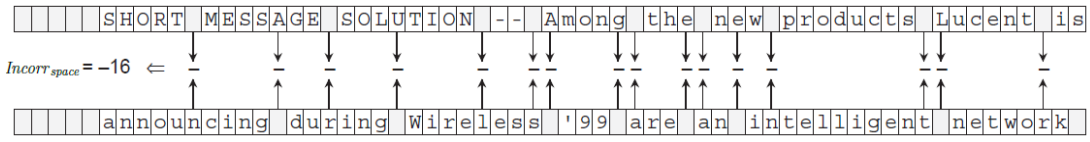
FIGURE 3.6 – Exemple illustrant les étapes de la méthode proposée. (a) Initialisation du bloc au mot "exemple" et recherche de ses voisins verticaux. (b), (c) Extension du bloc à tous les voisins verticaux et recherche des voisins des mots nouvellement ajoutés. (d) Résultat : la colonne entière constitue un bloc unique

La méthode la plus générique pour la détection de tableaux est proposée dans [Hu2000]. Elle est basée simultanément sur l'analyse des espaces horizontaux et sur l'alignement de texte. Le problème de détection de tableaux est décrit comme un problème de mesure de corrélation entre les lignes. Cette corrélation est mesurée en fonction de l'alignement vertical des espaces et du texte dans des lignes adjacentes.

Corrélation basée sur les espaces : Plus il y a des espaces qui sont situés à la même position horizontale dans deux lignes adjacentes, plus ces deux lignes sont corrélées (voir Figure 3.7). Les espaces aux extrémités gauche et droite des deux lignes ne sont pas considérés dans cette mesure.



(a)



(b)

FIGURE 3.7 – Corrélation entre deux lignes basée sur les espaces : (a) forte corrélation et (b) faible corrélation.

Corrélation basée sur l'alignement vertical du texte : La mesure de cette corrélation est basée sur le même principe de répartition en blocs verticaux décrit précédemment [Kieninger1998]. En se basant sur le chevauchement horizontal des mots dans deux lignes adjacentes, les lignes sont transformées en un ensemble de blocs verticaux. Dans la Figure 3.8, les mots ayant un chevauchement significatif (supérieur à un seuil) constituent un même bloc vertical ; ils ont alors le même label. Plus le nombre de blocs traversant deux lignes adjacentes (les deux en même temps) est élevé, plus les deux lignes sont corrélées. Par contre, la présence d'un bloc dans une seule ligne atténue la corrélation entre les deux lignes.

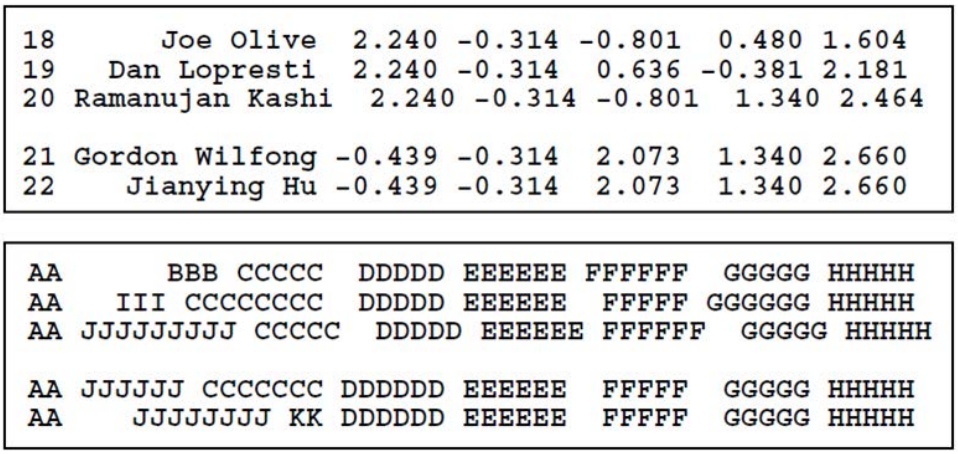


FIGURE 3.8 – Illustration de l'alignement vertical de blobs : en haut, des lignes de texte et en bas, des lignes contenant les labels des blocs verticaux correspondants.

La corrélation globale entre deux lignes est obtenue par une combinaison linéaire des deux mesures décrites ci-dessus. La détection du tableau est ensuite effectuée en adoptant une technique similaire à la croissance de régions [Kamdi2011], en utilisant la mesure de corrélation globale.

Cette méthode a été testée sur des documents images issus de la numérisation de pages de journaux. Si des lignes graphiques sont présentes dans les documents, elles sont détectées lors de l'étape de prétraitement et ne sont pas utilisées lors de la détection des tableaux. Les performances obtenues sont de 81% de rappel et 91% de précision. Ces performances s'expliquent par le fait que dans les documents imprimés, les lignes de tableaux présentent une corrélation élevée. En revanche, dans les documents manuscrits, la corrélation interligne peut être atténuée par les imperfections altérant la disposition des champs dans le tableau ; et donc les performances peuvent chuter.

3.4.2 Reconnaissance de tableaux

La reconnaissance de tableaux inclut les tâches suivantes :

— extraction des lignes et des colonnes ;
— délimitation des cellules ;
— discrimination des cellules d'entêtes/cellules de données et association des cellules de données aux cellules d'entêtes correspondantes ;
— reconnaissance du texte à l'intérieur du tableau.

Notre étude est limitée aux deux premières. Bien que ces deux tâches semblent être relativement faciles dans les documents imprimés, elles constituent un défi dans les documents manuscrits. En effet, il s'agit de trouver les frontières délimitant les différents éléments dans un tableau : lignes, colonnes et cellules dans des documents où les séparateurs graphiques peuvent être absents ou incomplets et la dis-

position physique des éléments peut être altérée. Comme pour la détection de tableaux, les méthodes de reconnaissance de tableaux peuvent être réparties en trois classes selon qu'elles utilisent les séparateurs graphiques, les espaces ou l'alignement de texte.

3.4.2.1 Méthodes basées sur les séparateurs graphiques

Quand les cellules d'un tableau sont délimitées par des séparateurs graphiques (principalement des lignes), l'extraction de ces séparateurs s'avère la méthode la plus appropriée pour extraire les cellules. La méthode proposée dans [Neves2006] consiste à chercher les intersections des lignes de tableaux en appliquant des dilatations morphologiques binaires basées sur 9 éléments structurants. Chacun de ces éléments correspond à un type d'intersection comme illustré dans la Figure 3.9. Une étape de vérification et de correction d'erreurs de détection d'intersections est ensuite effectuée. Elle consiste à examiner les intersections situées dans le 8-voisinage de chaque intersection et à détecter celles qui sont incompatibles (par exemple les intersections 2 et 4 représentent deux voisins horizontalement incompatibles). La présence de quelques artefacts dans les tableaux est la cause principale des erreurs dans la détection des points d'intersection. Ces artefacts, présents généralement sous forme de composantes connexes très compactes, sont filtrées en se basant sur leur facteur de compacité. A partir de l'ensemble des points d'intersection, les cellules sont identifiées comme l'ensemble des rectangles formés par ces points. 305 formulaires contenant des tableaux pré-tracées remplis à la main sont utilisés pour tester cette méthode. Un taux de bonne segmentation de 85% est obtenu.

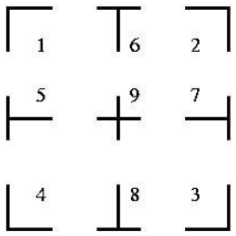


FIGURE 3.9 – Représentation des 9 types d'intersections de lignes dans un tableau.

Dans [Tsuruoka2001], les auteurs proposent une méthode basée sur la détection des lignes graphiques pour segmenter un tableau en cellules. D'abord une segmentation verticale est effectuée en estimant les positions des lignes verticales à partir du profil de projection. Elles correspondent aux positions horizontales des pics dépassant un certain seuil (fixé expérimentalement). Ensuite, une segmentation horizontale est effectuée séparément sur les colonnes délimitées par les lignes verticales estimées dans l'étape précédente. Les positions des lignes horizontales sont estimées à partir du profil de projection horizontale dans chaque colonne. La méthode a été conçue pour extraire les tableaux d'horaires de télé extraits des journaux japonais. Pour cela, des connaissances a priori sur la structure et le contenu de ces tableaux sont exploitées dans une phase de post-traitement pour raffiner la délimitation des cellules.

Concernant les résultats obtenus par cette méthode, les auteurs fournissent seulement quelques exemples de tableaux segmentés sans donner de mesures de performances sur l'ensemble des documents. La faiblesse de cette méthode réside dans l'utilisation de la technique de projection pour la détection des lignes graphiques. En effet, cette technique échoue à détecter les lignes inclinées et nécessite l'utilisation d'un seuil pour identifier les pics qui correspondent à des lignes.

3.4.2.2 Méthodes basées sur les espaces

Dans [Chen2012], les auteurs proposent une méthode de détection et de segmentation simultanées des tableaux. La méthode est basée sur la détection de points clés : intersections des bandes de "blancs" horizontales (espaces inter-lignes) et verticales (espaces inter-mots) (voir Figure 3.10.)

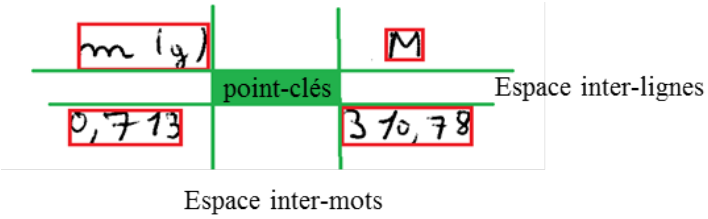


FIGURE 3.10 – Analyse des points d'intersections des flux vertical et horizontal des espaces

D'abord, les points clés sont identifiés et corrigés en se basant sur les lignes graphiques qui sont présentes dans le tableau. Ensuite, ces points sont complétés en ajoutant des points imaginaires de manière à obtenir une grille 2D régulière. Enfin, cette grille est validée en utilisant un modèle de champ aléatoire conditionnel qui permet de distinguer les points d'intersection corrects de ceux qui sont incorrects. Pour cela, des caractéristiques structurelles décrivant la distance verticale entre deux points consécutifs d'une même colonne ainsi que le texte situé entre deux points consécutifs d'une même ligne, sont extraites. L'algorithme min-cut/max-flow est utilisé pour la détermination de la configuration la plus probable des points constituant la grille.

La méthode a été testée sur 22 tableaux manuscrits contenant 584 cellules et a permis d'obtenir une précision de 100% et un rappel de 93.3%.

Un autre travail basé sur les espaces dans le texte contenu dans un tableau est décrit dans [Hu2001]. L'idée consiste à effectuer une classification hiérarchique de tous les mots de la région du tableau afin de générer tous les groupements possibles. De tels groupements sont représentés à l'aide d'un arbre binaire, où les feuilles représentent les mots, la racine représente le tableau entier et les nœuds intermédiaires représentent des groupes imbriqués à différents niveaux. Cet arbre est construit de manière ascendante. Initialement, les feuilles sont générées où chaque mot constitue une classe. Ensuite, des opérations de fusion sont effectuées de manière itérative en regroupant à chaque fois les deux classes les plus proches. Ces deux classes deviennent les fils de la classe nouvellement créée qui sera représentée par un nœud intérieur dans l'arbre. Cette opération est répétée récursivement jusqu'à obtenir un seul groupe, qui est représenté par le nœud racine. Pour mettre en œuvre cette classification, la distance entre deux mots (les feuilles) ainsi que la distance entre deux groupes de mots (nœuds internes) doivent être définies. Pour la première, la distance horizontale entre les boîtes englobantes des mots est utilisée. Pour la deuxième, la moyenne des distances entre toutes les paires de mots inter-classes est adoptée. L'arbre ainsi formé représente la structure hiérarchique du tableau en termes de regroupement vertical de mots. Il s'agit ensuite de partitionner l'arbre précédemment construit de manière à ce que chaque partition corresponde à une colonne. Pour ce faire, l'arbre est parcouru en largeur et une décision sur la scission au niveau du nœud visité est prise en utilisant des heuristiques sur les distances entre les mots. Les partitions obtenues à l'issue de ce parcours correspondent aux colonnes du tableau. Pour délimiter les cellules du tableau, les lignes sont aussi détectées en considérant toute bande d'espace horizontale comme un séparateur entre deux lignes.

La méthode a été testée sur deux corpus de tableaux au format ASCII extraits respectivement de 26 articles du quotidien américain de Wall Street et de 16 messages électroniques. Les performances obtenues sont respectivement 82% et 73% de bonne segmentation. Comme l'affirment les auteurs, cette

méthode peut être utilisée sur des documents images, qui doivent être, préalablement, segmentés en mots.

	FIRST QUARTER	SECOND QUARTER
Norman Robertson Mellon Bank	4.0	2.3
Neal Soss First Boston	3.1	2.3
Lawrence Kudlow Bear Stearns	3.0	3.5
Donald Ratajczak Georgia State Univ.	3.2	2.8

FIGURE 3.11 – Exemple de détection de lignes d'un tableau.

3.4.2.3 Alignement de texte

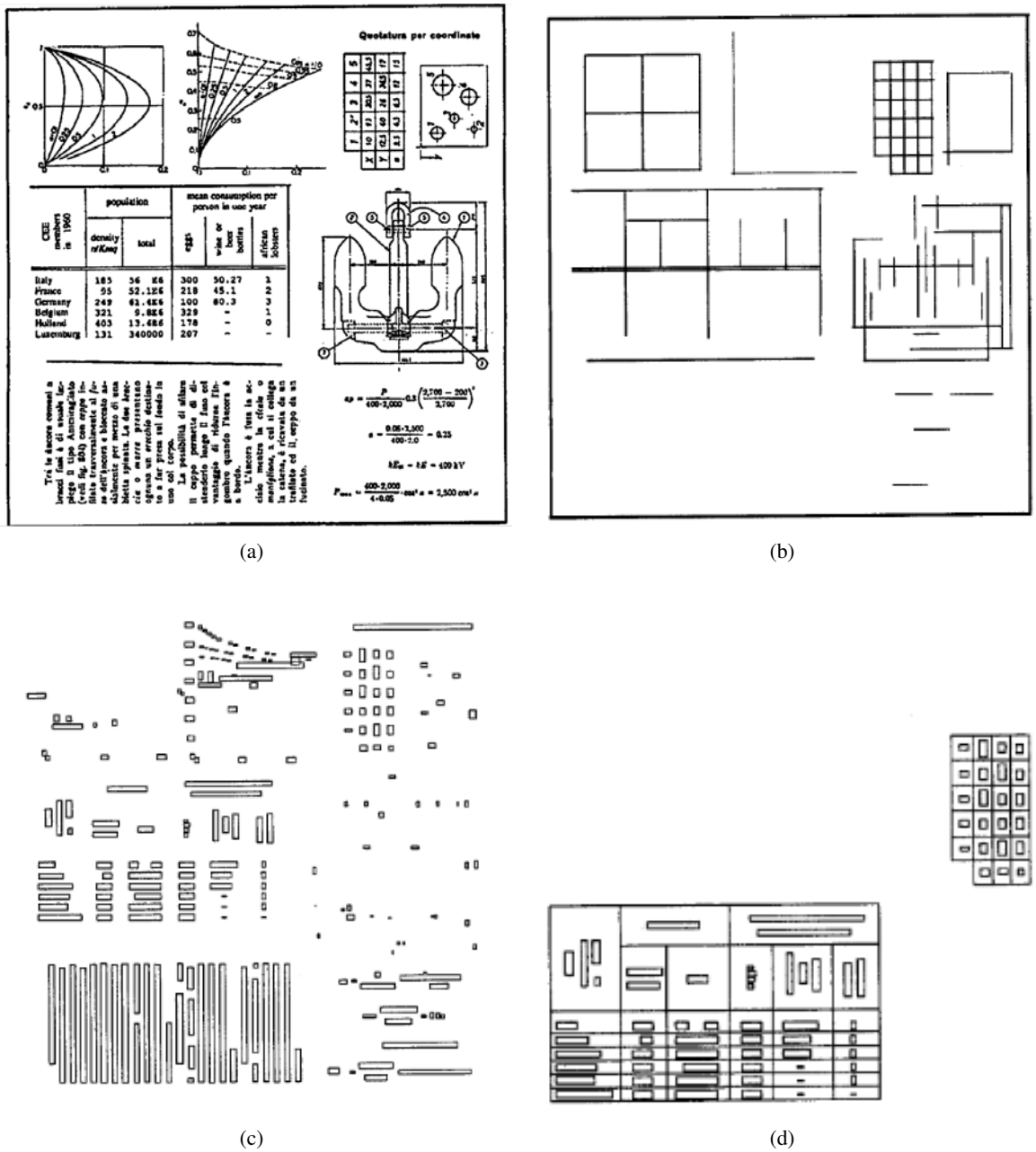
Dans [Laurentini1992], une méthode combinant la disposition de texte avec les lignes graphiques est proposée pour extraire les tableaux dans des images de documents imprimés composites (contenant des graphiques, du texte, des tableaux, des diagrammes, etc.). Les lignes graphiques sont extraites en cherchant les séquences de pixels dont la longueur dépasse un certain seuil tout en tolérant de petites coupures. Dans cette recherche, seules les directions horizontale et verticale sont considérées (voir Figure 3.12(b)). D'un autre côté, le texte est identifié en adoptant une approche ascendante. D'abord, les composantes connexes sont extraites et regroupées, selon un critère de colinéarité, en mots et les mots sont regroupés en chaînes. Seuls des regroupements horizontaux ou verticaux sont autorisés et cela sans regrouper des éléments séparés par une des lignes déjà extraites (voir Figure 3.12(c)). Pour compléter d'éventuelles lignes manquantes dans la structure du tableau, les chaînes de mots à l'intérieur de chaque cellule sont projetées horizontalement et verticalement afin de se prononcer sur la régularité de la disposition du texte. Le cas échéant, des lignes verticales et horizontales sont ajoutées au niveau des vallées dans les profils de projection (voir Figure 3.12(d)).

Les auteurs ont testé leur méthode sur 20 tableaux sans fournir des mesures exactes des performances obtenues.

L'approche proposée dans [Zuyev1997] consiste à segmenter un tableau en déterminant les séparateurs de lignes et de colonnes. Dans le cas d'absence de séparateurs explicites, les positions de séparation sont détectées à partir de l'alignement du texte. Pour ce faire, la technique de projection a été utilisée. D'abord, les composantes connexes contenues dans le tableau sont projetées verticalement en vue de la détection des espaces inter-colonnes. Les vallées du profil obtenu sont classées, à l'aide de l'algorithme K-moyennes ($K = 2$), pour identifier celles qui correspondent à de vrais espaces inter-colonnes. Les descripteurs utilisés pour cette classification sont la largeur et la profondeur normalisée de chaque vallée. Ensuite, la séparation des lignes du tableau est effectuée par une analyse du profil de projection horizontale. L'histogramme des largeurs des vallées de ce profil est utilisé pour identifier celles qui correspondent à des espaces entre deux lignes de tableaux.

La méthode a été implémentée et intégrée dans le système de reconnaissance FineReader[5]. Les auteurs affirment son expérimentation sur une grande variété de tableaux sans pour autant reporter les résultats obtenus. La méthode semble être efficace sur les documents imprimés où une certaine homogénéité typographique est garantie. Mais, elle est sensible à la qualité de numérisation et à la variabilité (au sein du même document) de l'écriture au niveau du style, de la taille et de l'espacement.

5. FineReader est un logiciel OCR développé par la multinationale Russe spécilaiste dans les logiciels de reconnaissance optique de caractères (FineReader), de capture de documents et des logiciels d'enseignement assisté par ordinateur pour micro-ordinateurs et appareils mobiles.



(a) (b)

(c) (d)

FIGURE 3.12 – Image extraite de [Laurentini1992] illustrant (a) un exemple de document composite contenant deux tableaux, (b) le résultat de l'extraction des lignes graphiques, (c) le résultat de l'extraction du texte et son regroupement en chaînes et (d) le résultat de l'extraction des tableaux.

3.4.2.4 Autres approches

En plus des deux séparateurs classiques (lignes graphiques et espaces), l'analyse des caractéristiques suivantes peut aider à détecter les frontières des différents blocs (lignes, colonnes, cellules, etc.) dans un tableau.

— des séparateurs spéciaux comme par exemple des séquences des symboles -, +, *, :, !.
— la taille et/ou le type de police ;
— la justification ;
— le contenu ;
— la syntaxe de contenu ;

Les trois premières caractéristiques peuvent être utilisées uniquement pour la segmentation des tableaux imprimés ou ASCII [Pinto2003] alors que les deux autres peuvent être exploitées pour segmenter même des tableaux manuscrits, à condition de pouvoir reconnaître leur contenu.

Dans [Peterman1997], les auteurs proposent une méthode basée sur l'analyse syntaxique du contenu des cellules pour extraire la structure d'un tableau. Ils se sont basés sur le fait que les données dans un tableau, vérifient une syntaxe précise, le long de lignes ou de colonnes. Afin de déterminer la syntaxe des champs d'un tableau, une méthode basée sur les expressions régulières réduites a été utilisée. Cette technique consiste à remplacer chaque caractère par un symbole spécial représentant le groupe de caractères auquel il appartient. Par exemple, toutes les lettres minuscules sont remplacées par un même caractère spécial ainsi que pour les lettres majuscules et les chiffres. Les chaînes ainsi obtenues sont comparées en utilisant la distance d'édition. Celles qui sont horizontalement ou verticalement voisines et ayant une distance d'édition inférieure à un certain seuil sont regroupées afin de former des lignes ou des colonnes. Au sein d'une même ligne ou colonne, les limites des cellules sont définies par les limites de la chaîne vérifiant la syntaxe correspondante. Cette méthode a été expérimentée sur un corpus de 100 tableaux imprimés ayant des structures variées. En plus des images des tableaux, le système prend en entrée le contenu reconnu par un OCR (ayant un taux d'erreur de 1%). Les tableaux contiennent environ 4500 cellules. Les résultats obtenus sont de 99% de précision et de $99,5\%$ de rappel.

3.5 Extraction de champs numériques

Dans cette section, nous allons étudier les principales méthodes d'extraction de champs numériques. Ce problème peut être subdivisé en deux tâches : la localisation et la reconnaissance. La première cherche à déterminer la position des chaînes numériques dans les images de documents. Quant à la deuxième, elle prend en entrée des champs numériques isolés et vise à déterminer leur valeur. Les deux tâches peuvent être effectuées séparément ou simultanément.

3.5.1 Localisation de champs numériques

Les champs numériques constituent généralement des éléments d'information pertinents. Ils ont fait l'objet de plusieurs travaux comme l'extraction des montants [Anisimov1995] et des dates [Mandal2012, Lam2002] dans les chèques de banques, l'extraction des codes postaux et des numéros de téléphones [Koch2005, Chatelain2006a] et des codes clients et d'autres champs numériques [Haji2011] dans les courriers manuscrits.

Dans [Anisimov1995], les auteurs ont développé un système de lecture des montants dans les chèques de banques françaises. Après avoir estimé l'orientation des contours et l'inclinaison moyenne de l'écriture sur la partie droite supérieure du chèque (partie contenant le montant dans les chèques français), les histogrammes de projection horizontal et vertical sont utilisés pour estimer la position exacte du montant

numérique. L'imagette contenant ce montant est ensuite filtrée afin de supprimer les lignes horizontales et verticales (si elles existent) ainsi que le bruit résiduel. Étant extraites, les composantes connexes de cette imagette sont analysées afin de détecter celles qui représentent des caractères isolés. Pour cela, une probabilité basée sur plusieurs caractéristiques géométriques de chaque composante est calculée. Si cette probabilité est supérieure à un certain seuil, la composante connexe est considérée comme un caractère isolé. De la même manière, chaque paire de composantes adjacentes est étudiée afin de décider si leur fusion forme un caractère isolé. Les grosses composantes, quant à elles, sont segmentées à l'aide de lignes droites positionnées selon plusieurs critères telles que l'intersection de ces lignes avec la composante segmentée, le profil supérieur et inférieur de cette dernière, etc. Ainsi, plusieurs hypothèses de segmentation sont générées. Pour chaque hypothèse, les probabilités que les fragments résultants soient des caractères isolés sont calculées. En considérant seulement les 16 meilleures hypothèses, la séquence de caractères obtenue est fournie à un système de reconnaissance de caractères formé par la combinaison de quatre classifieurs à 20 classes : 10 chiffres, 3 séparateurs (',', '.', '-'), 6 lettres indiquant les unités monétaires telles que 'F', 'f', etc. et 1 classe de rejet. La sortie de cette reconnaissance est une liste de montants candidats. Parmi ces derniers, ceux qui sont syntaxiquement invalides sont rejetés et ceux qui sont sémantiquement équivalents sont davantage pris en compte. Ce système a été développé et intégré dans un logiciel industriel développé par la société A2iA [6]. Il a été testé sur une base de chèques bancaires réels. Un taux de bonne reconnaissance de 65% a été obtenu. Vu l'intolérance d'un tel système à tous les types d'erreur, une option de rejet a été adoptée, permettant de réduire les erreurs à moins de 0.1%.

Dans [Mandal2012], les auteurs cherchent à extraire les champs de date dans des documents manuscrits. Ce travail traite plusieurs aspects liés aux différents formats de dates mais nous allons décrire uniquement la méthode adoptée pour l'extraction des parties numériques. Un document est d'abord segmenté en lignes en utilisant une méthode de regroupement perceptif des composantes connexes, basée sur des critères de proximité, de similarité et de continuité de direction [Likforman-Sulem1994]. Les composantes connexes de chaque ligne sont ensuite classées en 3 classes : ponctuation, chiffre ou lettre. Un vecteur de 400 descripteurs basés sur les histogrammes de gradient et un SVM à trois classes sont utilisés pour cette classification. Afin de prendre en compte la présence de caractères connectés, les composantes dont la probabilité de classification est inférieure à un seuil (fixé expérimentalement à 0.4) sont envoyées à un système de segmentation de chaînes de caractères. Dans cette étape, un ensemble de chemins de segmentation potentiels est généré en se basant sur le concept de réservoirs d'eau [Pal2002]. La segmentation optimale est ensuite déterminée en utilisant la technique de programmation dynamique [Roy2012]. Ainsi, toutes les composantes connexes sont classées et les chiffres sont localisés.

Les résultats du système global montrent une bonne performance quant à l'extraction des dates sous format complètement numérique par rapport à celles qui sont sous format semi-numérique, ce qui peut traduire l'efficacité de la méthode d'extraction de chiffres. Cette méthode présente l'inconvénient de classer toutes les composantes connexes alors qu'une simple étape de filtrage (basée sur les dimensions et la position par exemple) pourrait être utile pour retenir uniquement celles qui sont probablement des caractères isolés. De plus, la méthode de segmentation basée sur les réservoirs d'eau a été principalement conçue pour la segmentation de chaînes numériques [Pal2002] et nous pensons qu'elle n'est pas très adaptée pour la segmentation de chaînes alphabétiques ou mixtes.

Un autre travail situé dans ce même contexte, a été proposé dans [Lam2002] où les auteurs ont mené une étude détaillée sur la combinaison de classifieurs et la sélection des descripteurs les plus efficaces pour la discrimination des données numériques de celles alphabétiques. Cette étude a été motivé par

6. **A2iA** (**A**rtificial **I**ntelligence and **I**mage **A**nalysis) : est une société spécialisée dans le domaine de l'extraction de contenus, la classification de documents et la reconnaissance d'écriture manuscrite et imprimée.

une application réelle visant la différenciation entre les dates écrites en format chiffre et celles écrites en format mixte afin de les traiter adéquatement. Plusieurs types de descripteurs (gradients, distances, histogrammes, etc.) ainsi que plusieurs combinaisons de classifieurs (neuronaux de type PMC) ont été testés sur la base de chèques CENPARMI_IRIS [7]. Les résultats obtenus ont permis aux auteurs de choisir la meilleure façon de combiner un sous-ensemble parmi tous les classifieurs conçus. Pour ce travail, les auteurs ne détaillent pas le traitement des documents pour l'extraction des éléments à classer et ne donnent pas d'idée sur leurs natures (des composantes connexes, des mots ou des champs de date entiers).

Les trois travaux cités précédemment présentent la particularité de traiter des documents contraints où la position des champs numériques ou semi-numériques étudiés est connue à l'avance. En outre, l'extraction des dates et des montants dans les chèques peut être vérifiée et corrigée par différents moyens de vérification, comme le montant écrit en lettres et la syntaxe des champs de date.

Dans le cas des documents non-contraints, l'extraction de champs numériques est plus difficile et cette difficulté s'accentue en l'absence des moyens de vérification. Des exemples de méthodes qui traitent de ce problème sont présentés dans [Koch2005, Chatelain2006b, Chatelain2006c, Haji2011].

Les travaux de Koch et Chatelain sont effectués dans le même cadre et portent sur l'extraction de trois types de champs numériques (numéros de téléphone, codes clients et codes postaux) dans des courriers manuscrits. Les méthodes présentées fonctionnent sur les lignes de texte, qui sont extraites par regroupement de composantes connexes en se basant sur des critères d'alignement et de distance. Elles sont fondées autour d'une idée simple qui consiste à détecter et à reconnaître les chiffres dans un document pour y extraire les champs numériques recherchés en se basant sur leurs syntaxes. Pour ce faire, Koch et al. [Koch2005] ont recours à une classification morphologique des composantes connexes de chaque ligne selon qu'elles appartiennent à un champ numérique (chiffre isolé, chiffre-double, séparateur) ou non (rejet), sans effectuer aucune reconnaissance. Cette classification est effectuée par l'algorithme KPPV en se basant sur un vecteur de 7 caractéristiques décrivant la régularité d'une séquence numérique en terme de hauteur, de largeur et d'espacement. Afin de corriger d'éventuelles erreurs de labellisation, un modèle de Markov représentant la syntaxe de chaque champ numérique à extraire, est utilisé. Il permet d'interpréter la séquence des composantes connexes d'une ligne afin de déterminer la meilleure séquence de labels qu'elles peuvent avoir. Pour l'évaluation de cette approche, un corpus de 585 courriers manuscrits obtenus d'un grand cabinet a été utilisé, dont 292 ont été utilisés pour l'apprentissage des classifieurs et des modèles de Markov et 293 documents ont été utilisés pour les tests. Un taux d'extraction moyen sur tous les champs (724 champs) d'environ 55% est obtenu. Il convient de noter que plus un champ est syntaxiquement contraint, plus son taux d'extraction est élevé.

Le précédent travail a été étendu dans les travaux de thèse de Chatelain [Chatelain2006a] en intégrant une étape de reconnaissance. Ceci a été réalisé de deux façons différentes, conduisant à deux approches : la première [Chatelain2006b] est basée sur une stratégie de segmentation-reconnaissance-rejet et la deuxième [Chatelain2006c], inspirée des méthodes d'extraction d'information, réalise indépendamment la localisation et la reconnaissance.

Dans la première méthode, chaque composante connexe dans une ligne est successivement vue comme étant un chiffre isolé, un chiffre-double et un chiffre-triple. La production de l'hypothèse chiffre isolé est effectuée par le biais d'un classifieur chiffre. Pour produire les autres hypothèses, la composante connexe est soumise à un système de segmentation-reconnaissance. Ce système utilise l'algorithme "drop-fall" [Congedo1995] qui génère quatre chemins de segmentation. La production de l'hypothèse

7. CENPARMI_IRIS : une base de documents composée de chèques français et anglais disponible pour tout usage à des fins de recherche

chiffre-double est effectuée en choisissant le meilleur chemin de segmentation en fonction des scores de confiance des deux premières propositions. La production de l'hypothèse chiffre triple est effectué en réitérant le processus de segmentation sur le fragment de la composante ayant le plus faible score. Le classifieur utilisé pour la reconnaissance des fragments résultant d'une segmentation est constitué de la combinaison (à l'aide d'une règle de type produit) de deux PMCs dont les entrées sont respectivement 128 descripteurs de chain codes et 117 descripteurs statistiques/structurels. Une classe rejet (composante non-chiffre) a été aussi prévue et sa probabilité est estimée à partir du score de la première proposition du classifieur. A l'issue de cette étape, un treillis d'hypothèses de reconnaissance à trois niveaux est obtenu. La recherche des meilleures solutions valides au sens des contraintes syntaxiques est effectuée à l'aide de modèles de lignes de texte. Ces modèles intègrent les contraintes syntaxiques exprimées sous formes de transitions chiffre/rejet dans une ligne. Testé sur 293 documents, le système a permis d'obtenir un rappel de $48,5\%$ pour une précision de $15,5\%$.

Contrairement à cette méthode où la localisation et la reconnaissance des champs numériques sont effectuées conjointement, la deuxième méthode effectue de manière indépendante la localisation puis la reconnaissance de champs numériques. La particularité de la localisation est qu'elle n'est pas basée sur des techniques classiques de reconnaissance d'entités manuscrites via un processus de classification des composantes connexes de manière individuelle. Elle consiste plutôt à interpréter globalement la séquence de composantes qui constituent une ligne de texte à l'aide d'un modèle neuro-markovien. Il s'agit d'un modèle de Markov permettant de modéliser la séquence de caractères alphanumériques constituant une ligne où les observations sont estimées à l'aide d'un classifieur de type réseau de neuronnes. Ce classifieur fournit, pour chaque composante, une liste ordonnée d'hypothèses de classification (chiffre, chiffre-double, séparateur ou rejet), avec les mesures de confiance correspondantes. le treillis de ces hypothèses est soumis au modèle de Markov afin d'obtenir le meilleur alignement sur une syntaxe donnée (voir Figure 3.13).

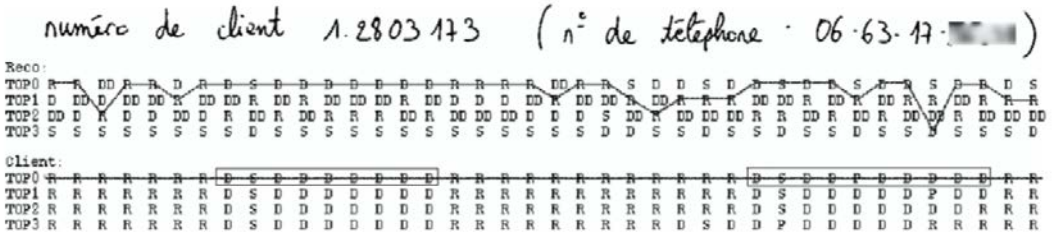


FIGURE 3.13 – Alignement du treillis d'hypothèses de classification d'une ligne de texte manuscrit sur le modèle syntaxique d'une ligne contenant un code client [Chatelain2006c].

Le résultat de l'étape de localisation (classification + alignement) consiste en un ensemble de champs numériques constitués des composantes qui ont été toutes classées en tant que numérique (chiffre, chiffre-double, séparateur). L'étape de reconnaissance consiste à déterminer les valeurs de ces composantes. Celles qui sont labellisées comme étant des chiffres isolés, sont soumises à un classifieur pour être reconnues. Les composantes chiffres-doubles sont soumises à un système de segmentation-reconnaissance. Ce module repose sur la génération de plusieurs hypothèses de segmentation qui seront fournies au classifieur chiffre pour se prononcer sur les valeurs des deux chiffres. La meilleure hypothèse est ensuite choisie en fonction des confiances fournies par le classifieur. Les séparateurs sont ignorés. Une étape de vérification a été introduite en post-traitement du système. Elle consiste à analyser les hypothèses de champs afin d'accepter uniquement ceux qui étaient effectivement à détecter. Cette étape utilise un vecteur de caractéristiques provenant des deux étapes précédentes et des informations sur les boîtes englobantes et le fournit à un classifieur PMC qui décide l'acceptation ou le rejet d'un champ. Ce système

a été évalué sur une base de 300 courriers manuscrits et a permis de reconnaître plus de 60% des champs. On note une amélioration d'environ 6 points de pourcentage par rapport à [Koch2005]. Ceci est dû principalement à l'apport des étapes de reconnaissance et de vérification ajoutées au système.

Une autre méthode traitant de l'extraction de champs numériques dans des courriers manuscrits, imprimés et mixtes est proposée dans [Haji2011]. La chaîne de traitement permettant la mise en place de cette méthode est composée des étapes suivantes : extraction de lignes, segmentation en caractères, élagage, et vérification.

La première étape consiste à segmenter les documents en lignes en utilisant la carte adaptative de connexités locales (ALCM) générée à partir de l'image à l'aide d'un filtre à orientation sélective [Shi2009]. La deuxième étape consiste à segmenter en caractères chaque ligne de texte. Un algorithme basé sur un graphe modélisant le squelette de l'image (les nœuds sont les points de jonction et les points terminaux du squelette et les arcs représentent les liens entre ces points) est utilisé pour cet objectif. Pour résoudre le problème des caractères sur-segmentés (à cause d'un défaut de numérisation, de binarisation ou même de l'algorithme de segmentation), les auteurs utilisent un algorithme de fusion basé sur la technique de partitionnement de graphe avec recherche heuristique. La troisième étape effectue une sélection préliminaire des caractères numériques en se basant sur la régularité de largeur, de hauteur et d'espace inter-caractères. La dernière étape de la chaîne de traitement consiste à vérifier si une séquence de caractères (pré-sélectionnée dans l'étape précédente) est numérique ou non. Pour ce faire, une classification alphabétique/numérique de chaque caractère est effectué à l'aide d'un classifieur neuronal en utilisant des descripteurs à base de gradient. Afin d'éviter d'éventuelles confusions entre chiffres et lettres présentant une grande similarité (comme par exemple 'S' et '5', '2' et 'Z', etc.), des classifieurs binaires spécifiques ont été utilisés.

Cette méthode est conçue pour extraire tout type de champs numériques : elle n'utilise aucune information a priori sur les champs recherchés. Toutefois, elle a été testée sur l'extraction des codes clients dans des courriers. Une base d'environ 600 courriers de différents scripts et structures a été utilisée pour l'évaluation de la méthode, dont 100 ont été utilisés pour l'apprentissage des classifieurs. Les résultats ont permis d'atteindre une F-mesure de 94%. Les performances de cette méthode dépendent fortement du résultat de la segmentation en caractères, ce qui est a priori une opération très délicate. Ceci constitue une limite principale de cette méthode car nous pensons que, malgré les efforts fournis, les résultats de la segmentation en caractères dans les documents manuscrits sont encore loin d'être satisfaisants.

Nous constatons que la plupart des méthodes de localisation des champs numériques font appel à une opération de reconnaissance. Si la reconnaissance des chiffres isolés peut être considérée comme un problème résolu en grande partie [Diem2013], la reconnaissance de chiffres connectés reste encore un challenge en vertu du paradoxe de Sayre [Sayre1973]. Celui-ci stipule que pour bien reconnaître les caractères composant une chaîne, il faut d'abord les avoir segmentés mais pour les segmenter, il faut les avoir reconnus. De nombreuses approches ont été proposées dans la littérature afin de résoudre ce dilemme de segmentation-reconnaissance pour les chiffres connectés. C'est ce que nous explorons dans la section suivante.

3.5.2 Reconnaissance de chiffres connectés

Il existe peu d'approches holistiques qui sont proposées pour la reconnaissance de chiffres connectés [Wang2000, Zhou2005]. Ceci est dû au grand nombre de classes qu'on doit prendre en compte dans une telle approche (10^n classes pour des chaînes de n chiffres), ce qui pose un problème pour collecter suffisamment d'exemples de chaque classe. Motivé par le fait que naturellement la majorité (plus de 80%) des chiffres connectés est composée uniquement de deux chiffres, Wang et al. [Wang2000] ont

adopté une approche holistique pour reconnaître les chiffres doubles. Ils ont étendu un classifieur utilisé pour la reconnaissance de chiffres isolés [Favata1996] à une classification à 100 classes correspondant à toutes les combinaisons possibles de deux chiffres $\{00..99\}$. Des descripteurs structurels, de gradients et de concavité ont été utilisés pour caractériser les images des chiffres-doubles. Les expérimentations menées dans ce travail ont montré que les résultats de l'approche holistique surpassent ceux de deux approches analytiques [Fenrich1991, Shi1997].

Dans [Zhou2005], une nouvelle implémentation d'un classifieur multi-cleasses à base de SVM est proposée. Le classifieur est conçu de manière qu'il puisse effectuer des classifications en n classes avec n très grand, en offrant un bon compromis entre la complexité/temps et la précision. Il est adapté à la classification de toutes les paires de chiffres. Appris sur environ 2400 échantillons de chiffres doubles et testé sur environ 500 échantillons, ce classifieur a permis d'obtenir un taux de 94% de bonne reconnaissance.

Bien que les méthodes holistiques aient montré une efficacité quant à la reconnaissance de chiffres doubles, elles restent difficiles à être mises en œuvre pour les chiffres connectés dont on ne connaît pas le nombre. Pour cette raison, les méthodes les plus répandues pour la reconnaissance de chiffres connectés sont des approches analytiques. Celles-ci peuvent être subdivisées en deux grandes classes :

— stratégie de segmentation puis reconnaissance [Pal2002, Lu1999] ;
— stratégie de segmentation-reconnaissance [Wu2014, Gattal2015, Kim2002].

La première stratégie consiste à segmenter d'abord la chaîne numérique et d'envoyer les fragments résultant à un moteur de reconnaissance. Elle présente un point délicat quant au choix du meilleur chemin de segmentation. Elle nécessite alors des métriques d'évaluation très sélectives pour pouvoir choisir la bonne segmentation car ce choix est définitif et ne peut pas être mis en cause ultérieurement. En revanche, la deuxième stratégie consiste à générer un ensemble de chemins de segmentation possibles et de choisir le meilleur en se basant sur la qualité de la segmentation et la reconnaissance combinées.

3.5.2.1 Stratégie de segmentation puis reconnaissance

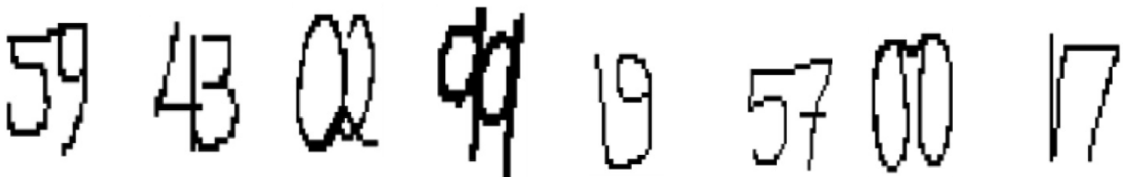
La stratégie de segmentation puis reconnaissance repose sur les deux étapes, effectuées de manière séquentielle et indépendante. L'étape de la reconnaissance est relativement immédiate puisque, une fois segmentés, les chiffres isolés peuvent être transmis à un classifieur pour être reconnus. Une grande variété de classifieurs ont été utilisés pour la reconnaissance de chiffres isolés, dont nous pouvons citer à titre d'exemple, les SVMs [Gorgevik2002, Razafindramanana2013], les réseaux de neurones [Dan2012, Kaensar2013], les arbres de décision [Jiang2006], le KPPV [Lee1991], la combinaison de plusieurs classifieurs [Gorgevik2004, Lee1991], etc. En outre, les résultats obtenus à l'issue d'une reconnaissance, sont généralement bons et peuvent atteindre plus que 99% [Ranzato2007, Kegl2009, Dan2012]. Ainsi, dans les méthodes basées sur la stratégie de segmentation puis reconnaissance, l'étape clé est la segmentation. Pour cela, dans cette section, nous présentons en plus des systèmes complets de segmentation puis reconnaissance, les méthodes de segmentation seule (non suivie par une reconnaissance.)

Le travail présenté dans [Lu2011] effectue d'abord une segmentation en utilisant une méthode basée sur la binarisation adaptative et l'analyse des composantes connexes. Ensuite, la reconnaissance des segments obtenus est confiée à un moteur de reconnaissance constitué par la combinaison d'un SVM et de deux classifieurs neuronaux : un SOM (Self-Oganizing Map) et un BPNN (back-Propogation Neural Network). Cette méthode a été testée, dans le cadre de la compétition HSDRC [Diem2014], sur 3 bases publiques de chaînes numériques et a donné un taux moyen de bonne segmentation/reconnaissance d'environ 50%. Ce modeste résultat peut être expliqué par l'insuffisance de la méthode de segmentation utilisée. En effet, aucune analyse morphologique ou structurelle pouvant informer sur les positions de liaisons des chiffres dans l'image d'une chaîne numérique n'est effectuée.

Dans la mesure où la connexion entre deux chiffres engendrent généralement des caractéristiques morphologiques spéciales (des vallées, des jonctions, etc.) dans la composante connexe, nous pensons que les méthodes de segmentation les plus robustes sont celles qui cherchent directement ces caractéristiques. La méthode présentée dans [Pal2002], vise à détecter les réservoirs d'eau (une métaphore pour décrire des vallées supérieures ou inférieures dans une image) pour s'en servir dans la détermination du chemin de coupure. Les positions et les tailles des réservoirs sont analysées afin de déterminer un réservoir principal qui est situé dans la zone de connexion. Selon le type de ce réservoir (haut ou bas) et sa base, la position de connexion (haut, milieu ou bas) est décidée. Un ensemble de règles basées sur les caractéristiques structurelles ainsi que les positions et les tailles des autres réservoirs et des boucles est utilisé afin de décider définitivement du chemin de segmentation. Dans cette méthode, les auteurs se contentent uniquement de la segmentation et non de la reconnaissance des chiffres qui peut être obtenue par n'importe quel classifieur. L'évaluation de cette méthode est effectuée sur environ 2200 chiffres-doubles extraits des chèques bancaires français et a permis de segmenter correctement $94,8\%$ des chiffres connectés.

Une autre méthode de segmentation qui tend à exploiter les caractéristiques spéciales engendrées par une connexion entre deux chiffres, est proposée dans [Elnagar2003]. Afin d'analyser au mieux ces caractéristiques, différents cas de figures de connexion sont énumérés :

- connexion par un seul point de contact (Figure 3.14(a).) ;
- connexion par un segment de contact (Figure 3.14(b).) ;
- connexion lisse où les deux chiffres sont liés souplement (Figure 3.14(c)) ;
- connexion avec ligature où les deux chiffres sont liés avec un segment supplémentaire (Figure 3.14(d))



(a) (b) (c) (d)

FIGURE 3.14 – (a) Différents cas de connexion entre deux chiffres [Elnagar2003].

Les auteurs cherchent à extraire des points qui caractérisent ces cas de connexion : les points de branchement, de croisement et les points terminaux. Parmi ces points, ceux qui sont proches de la plus grande vallée supérieure ou inférieure (colline) sont considérés comme des points de segmentation. Un ensemble d'heuristiques sur la configuration de ces points (distances, positions relatives, nature de la partie de la composante qui les sépare) est utilisé pour décider d'un chemin de segmentation. La méthode est testée sur un ensemble de chiffres-doubles extrait des bases CEDAR et NIST19. Les résultats de segmentation sont vérifiés par un système de reconnaissance puis manuellement pour les chiffres rejetés ou mal reconnus. Une performance globale de 96% a été obtenue. Cette performance élevée peut être expliquée par le fait que le système est évalué uniquement sur des chiffres-doubles à connexion simple puisque les cas énumérés précédemment ne couvrent pas toutes les connexions possibles.

En conclusion, nous constatons que la majorité des méthodes basées sur la stratégie de segmentation puis reconnaissance ont recours à des règles ou à des heuristiques pour construire un chemin de segmentation à partir d'un ensemble de caractéristiques de la composante connexe. De plus, à notre connaissance, il n'existe pas de méthodes basées sur cette stratégie qui traitent des chaînes numériques

contenant plus de 2 chiffres. Ces dernières sont plutôt segmentées en se basant sur une stratégie de segmentation-reconnaissance.

3.5.2.2 Stratégie de segmentation-reconnaissance

Les travaux plus récents traitant de la reconnaissance des chiffres connectés en adoptant une stratégie de segmentation-reconnaissance sont décrits dans la compétition HDSRC 2014. Le système de Wu et al. [Wu2014] génère un ensemble de chemins de coupures verticales sur les composantes connexes représentant des chaînes numériques, en analysant leurs profils de projection supérieur et inférieur [Liu2004]. A l'issue de cette opération, une composante connexe est représentée par une séquence de segments primitifs qui sont par la suite fusionnés pour former des images candidates de chiffres. Un treillis d'hypothèses de segmentation est alors obtenu à partir des différentes possibilités de fusion des segments consécutifs. Les fragments candidats sont reconnus en utilisant un classifieur polynomial prenant en entrée le descripteur HOG (Histogramme de Gradient Orienté). Un algorithme de recherche en faisceau (beam search) est utilisé pour déterminer le meilleur chemin de segmentation parmi le treillis d'hypothèses. Cette méthode a montré sa supériorité sur toutes les autres méthodes qui ont participé à la compétition HSDRC 2014 avec 85% comme taux moyen de bonne reconnaissance sur les trois bases utilisées pour l'évaluation. Elle tire sa force principalement de la robustesse de son algorithme de segmentation qui extrait toutes les positions possibles de connexion de chiffres.

Une autre approche qui a montré son efficacité pour la segmentation des chiffres connectés est celle présentée dans [Oliveira2000] et réutilisée/améliorée après dans [Oliveira2002, Sadri2007, Gattal2015]. Elle est basée sur l'extraction de trois types de points caractéristiques informant sur les zones susceptibles de contenir des liaisons entre les chiffres : les points terminaux et les points de jonction extraits à partir du squelette de la composante connexe et les points de base extraits à partir du contour extérieur ou des profils inférieur et supérieur (voir Figure 3.15). Les points terminaux et les points de jonction sont extraits en utilisant les chaînes de Freeman en 8-connexité : les premiers sont des points ayant un seul voisin et les deuxièmes sont des points ayant plus de 2 voisins. Quant aux points de base, elles correspondent aux extrema dans les contours ou dans les profils de projection supérieur et inférieur. Une fois extraits, ces points sont utilisés pour générer des chemins possibles de segmentation en utilisant un ensemble de règles définies sur le nombre, la disposition et les distances entre eux. Ces chemins peuvent correspondre à des coupures intra-caractères ou inter-caractères. La distinction entre ces deux types de segmentation est effectuée au niveau reconnaissance. Différents classifieurs ont été employés à cette fin comme par exemple un PMC dans [Oliveira2000], un SVM dans [Gattal2015] ou une combinaison de plusieurs PMCs dans [Oliveira2002]. La méthode présentée dans [Oliveira2000] a été testée sur les chiffres connectés (principalement des chiffres-doubles et triples) extraits des chèques brésiliens et a permis d'obtenir une performance avoisinant $98,5\%$ de bonne segmentation. Les résultats obtenus dans [Sadri2007] sont relativement moins bons avec un taux allant de 90% à 97% selon le nombre des chiffres connectés puisque la méthode a été testée sur des chaînes numériques de différentes longueurs allant de 2 à 10 extraits de la base NIST SD19 [Grother1995]. A peu près le même résultat sur cette même base a été obtenu par la méthode présentée dans [Gattal2015].

Une autre méthode proposée pour la première fois dans [Congedo1995] et réutilisée ensuite dans [Dey1999, Chatelain2006b], consiste à générer des chemins de segmentation en simulant les trajectoires de la chute d'une goutte d'eau (drop-fall) verticalement le long de la composante connexe. Lorsque la goutte rencontre un obstacle, elle coupe la composante et continue son chemin. Selon la direction de la chute de la goutte (descente ou ascension), et selon la direction privilégiée quand elle rencontre un obstacle (gauche ou droite), quatre chemins de coupure sont générés. Pour chaque chemin, les fragments obtenus sont reconnus en utilisant plusieurs classifieurs de chiffres isolés dont les sorties sont combinés à

l'aide de la règle de vote majoritaire. Si les fragments sont bien reconnus, la segmentation est considérée bonne. Cette méthode a été testée sur des chiffres connectés extraits de la base CEDAR. Le taux de segmentation correcte obtenu est de 91%.

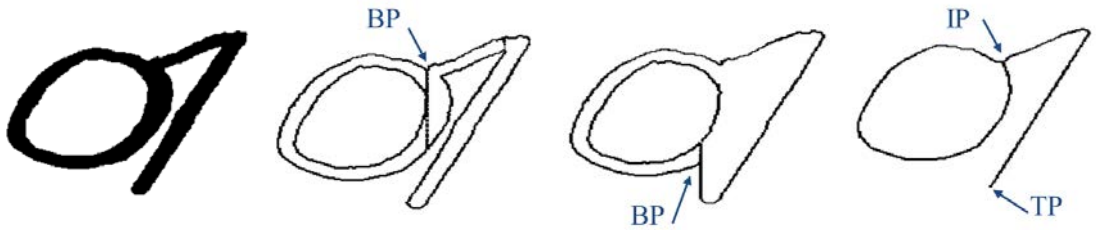


FIGURE 3.15 – (a) Image binarisée de deux chiffres connectés 0 et 7 , (b) et (c) respectivement le contour (intérieur et extérieur) et le profil (inférieur et supérieur) de l'image avec les points de base (BP) et (c) squelette de l'image avec les points d'intersection (IP) et les points terminaux (TP).

Une méthode en quatre étapes est proposée dans [Kim2002] pour la segmentation et la reconnaissance des chiffres doubles. La première étape consiste à détecter les chiffres cassés en se basant sur les positions relatives des composantes connexes et leurs dimensions afin de fusionner les fragments d'un même caractère. Dans la deuxième étape, un ensemble de points de coupures potentiels est déterminé à l'aide d'une analyse structurelle du contour et des profils de projection supérieur et inférieur. Ces points sont examinés, dans la troisième étape, afin de déterminer le type de connexion entre les deux chiffres. Selon la connexion, les points "non sérieux" sont éliminés et seuls les principaux points sont gardés. Dans la dernière étape, différentes segmentations basées sur ces points sont générées. La probabilité de reconnaissance de chacune des sous-composantes engendrées est estimée par un classifieur PMC de chiffre isolé. Un vecteur de 6 descripteurs (maille, chaincode, nombre de trous, ratio hateur/largeur, transition et distance) est fourni à l'entrée du classifieur. Si la probabilité combinée des deux fragments dépasse un certain seuil, la segmentation est considérée bonne. Les tests ont été effectués sur 3500 paires de chiffres connectés extrait de la base NIST SD19. Ils ont montré un taux de 92.5% de segmentation correcte. Cette méthode présente la particularité d'utiliser le type de connexion pour la détermination des positions de coupures, ce qui n'est pas évident dans le cas de chaînes numériques de tailles supérieur à 2.

3.6 Conclusion

Les méthodes présentées dans ce chapitre portent sur différents niveaux de traitements dans la chaîne d'analyse d'images de documents. Au niveau de la segmentation, nous avons exploré les méthodes qui visent à extraire les principales régions d'un document, telles que la séparation texte/graphique, l'extraction de tableaux et la segmentation de texte en lignes. Au niveau de la reconnaissance, notre étude a été focalisée sur l'extraction de champs numériques dans les documents.

Chapitre 4

Extraction de formules chimiques

Sommaire

4.1 Introduction 53
4.2 Difficultés 53
4.3 Prétraitement 54
4.4 Extraction de la formule chimique 56
4.4.1 Segmentation 56
4.4.2 Caractérisation des structures linéaires 60
4.4.3 Classification des structures linéaires 62
4.5 Expérimentation et résultats 66
4.5.1 Évaluation de la segmentation 66
4.5.2 Apprentissage du classifieur 67
4.5.3 Évaluation de l'extraction de la formule chimique 67
4.6 Conclusion 69

4.1 Introduction

Dans ce chapitre, nous nous intéressons à l'extraction du bloc graphique contenant une formule chimique dans un document de chimie. La méthodologie retenue est de passer par la classification de structures linéaires extraites du document, pour séparer le graphique (formule chimique) d'une part, et le texte (reste du document) d'autre part. En effet, la formule chimique aussi bien que les lignes de texte constituent des structures linéaires rassemblant des composantes connexes horizontalement alignées. La discrimination entre ces deux types de structures est effectuée en se basant sur un ensemble de descripteurs structurels.

Le reste de ce chapitre est organisé comme suit. Nous commençons par présenter les difficultés de la segmentation Texte/Graphique dans les documents de chimie. Ensuite, nous décrirons la chaîne de traitement mise en œuvre pour l'extraction de la formule chimique. Enfin, nous présenterons les expérimentations réalisées sur une base conséquente de documents et les performances obtenues et nous finirons par une conclusion.

4.2 Difficultés

Étant manuscrites, les formules chimiques héritent des mêmes difficultés liées à la segmentation des documents manuscrits composites non contraints, dues principalement à la variabilité de l'écriture au

niveau du style, de la taille et de la qualité. De plus, certaines difficultés leurs sont spécifiques, notées dans les points suivants :

— Une formule chimique est composée en majorité d'éléments graphiques (lignes, polygones, cercles, etc.) mais comprend aussi des chaînes de caractères représentant les noms de molécules utilisées.
— Les éléments graphiques peuvent être de tailles quelconques et non parfaitement tracés, ce qui les rapproche des éléments textuels (caractères, pseudo-mots, mots, etc.).
— Les composants textuels de la formule peuvent soit coller aux éléments graphiques et rendre difficile leur détection, ou au contraire, se trouver éloignés d'eux et risquer d'être considérés comme externes à la formule.

Pour faire face à ces difficultés, nous proposons un système en plusieurs étapes (voir Figure 4.1).

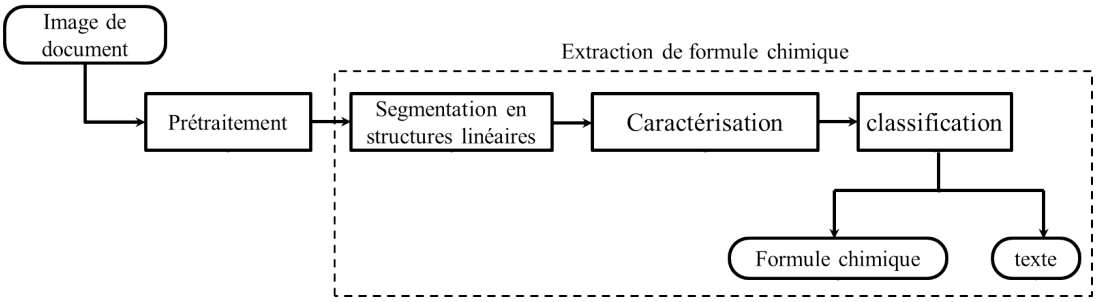


FIGURE 4.1 – Chaîne de traitements pour l'extraction de formules chimiques.

4.3 Prétraitement

La première opération effectuée est la binarisation des images. Plusieurs algorithmes de binarisation sont proposés dans la littérature [Otsu1979, Niblack1985, Sauvola2000, Wolf2002, Gupta2007]. Le premier, basé sur un seuillage global, est adapté aux images bimodales, où un mode représente le texte et l'autre représente le fond. Les autres algorithmes, basés sur un seuillage local, sont souvent utilisés pour binariser les images dont le contraste est faible.

Nous nous sommes basés sur l'histogramme de niveaux de gris pour sélectionner les images bimodales et les binariser avec l'algorithme d'Otsu (voir Figure 4.2(a)). Pour les autres images dont l'histogramme est multimodal ou lisse, nous avons utilisé la méthode de Wolf (voir Figure 4.2(e)). Elle utilise une fenêtre glissante pour déterminer un seuil de binarisation pour chaque pixel, en fonction de la moyenne et de l'écart type des niveaux de gris des pixels situés dans la fenêtre.

En deuxième étape, nous avons effectué des opérations de nettoyage sur les images binaires en vue d'améliorer leur qualité en éliminant les défauts de mauvaise capture de documents. Ces défauts sont principalement :

— Les bordures noires.
— Le bruit impulsionnel, appelé également poivre et sel.

Le nettoyage des images de ces documents est constitué des opérations de filtrage suivantes :

— Suppression des composantes connexes formant les bordures en se basant sur leurs formes et leurs positions (voir figure 4.3).
— Suppression du bruit impulsionnel (voir figure 4.3) en utilisant l'algorithme KFill. Comme décrit dans [Khaffaf2008], cet algorithme utilise une fenêtre de taille $k \times k$ pixels qui doit être déplacée sur l'image entière. La suppression du bruit de type sel est effectuée en mettant en noir le pixel

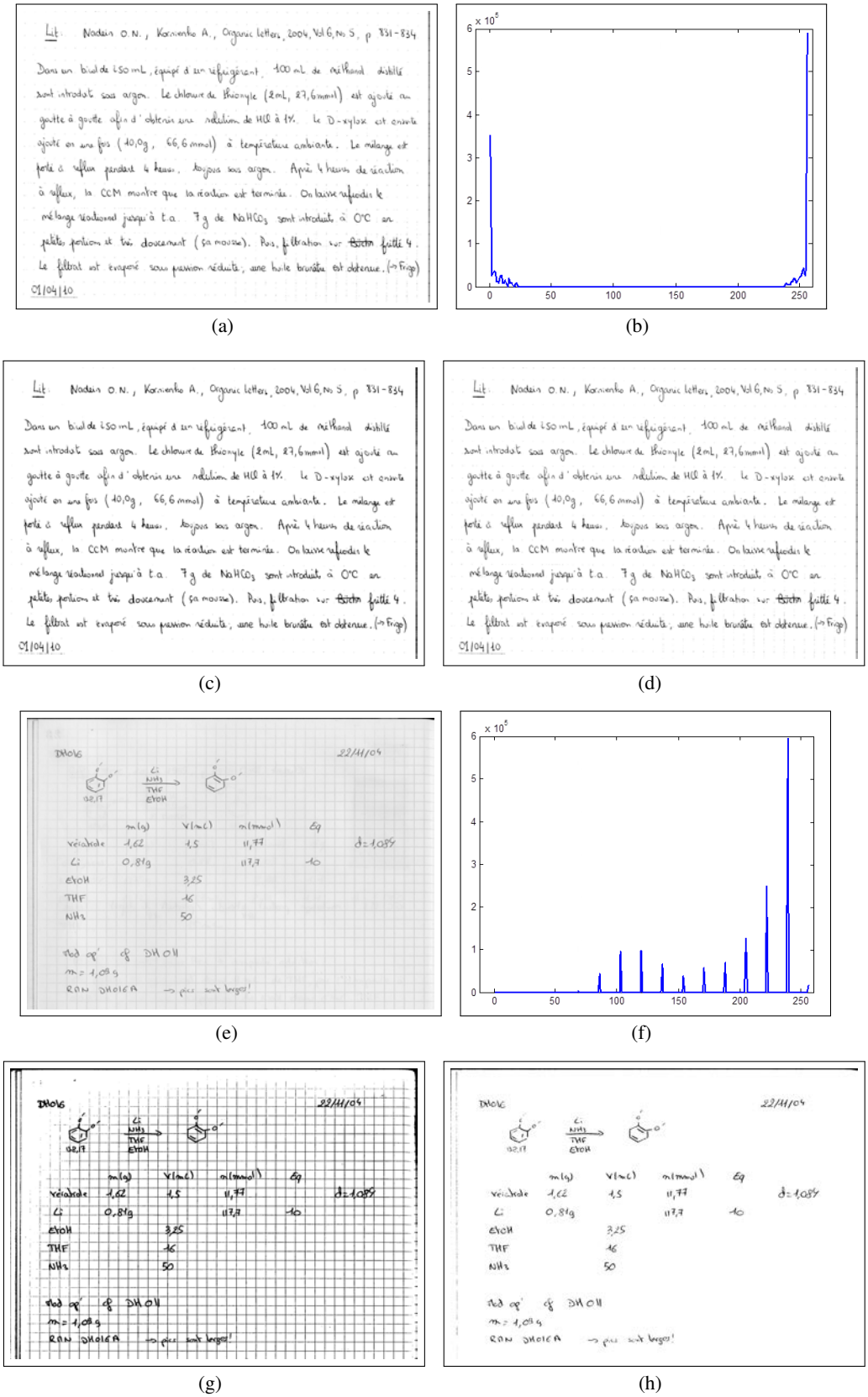
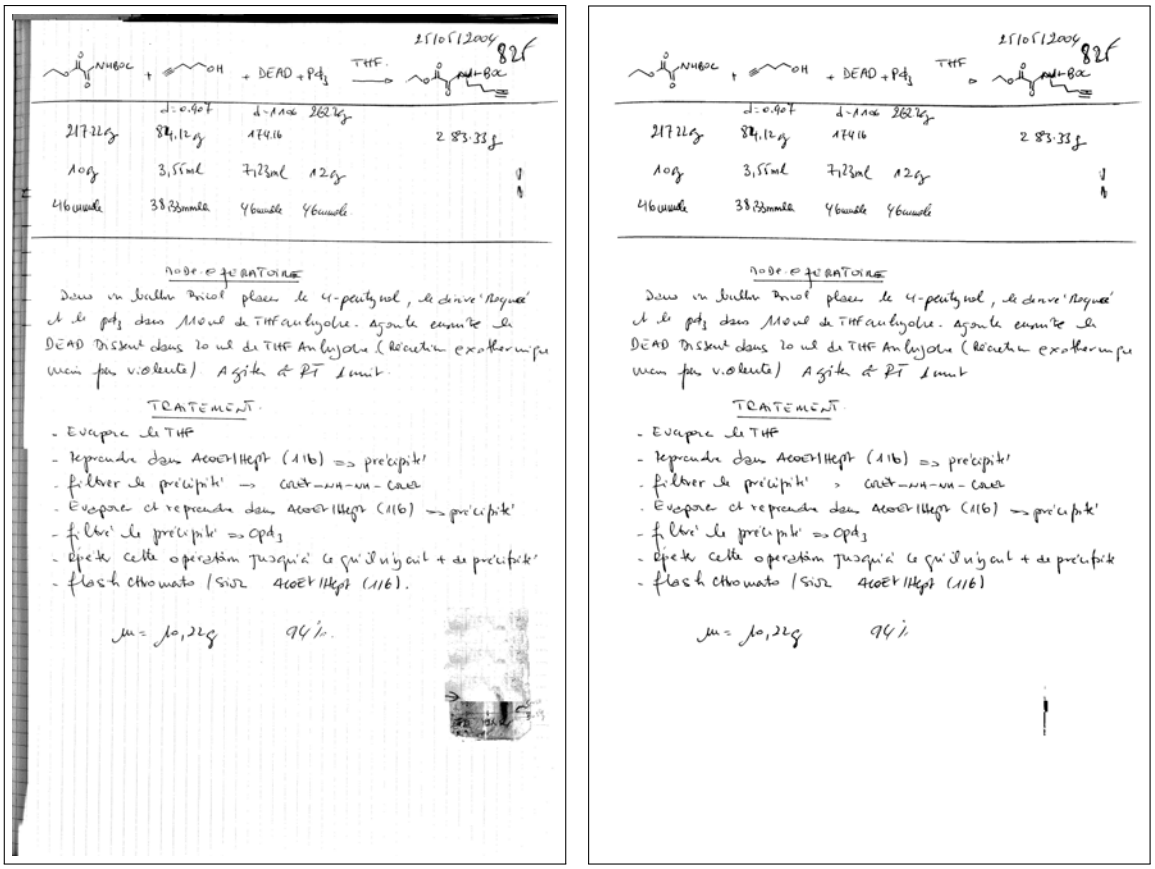


FIGURE 4.2 – (a) et (e) images en niveaux de gris, (b) et (f) les histogrammes de niveaux de gris correspondants, (c) et (g) images binarisées avec l'algorithme d'Otsu, et (d) et (h) images binarisées avec l'algorithme de Wolf.



(a) (b)

FIGURE 4.3 – Nettoyage des images de documents : (a) image d'origine, (b) image nettoyée.

situé au centre de la fenêtre si la condition suivante est satisfaite :

$$(c = 1)\ et\ ((n > 3k - 4)\ ou\ ((n = 3k - 4)\ et\ (r = 2))) \tag{4.1}$$

où c est le nombre de composantes connexes situées dans la fenêtre, n est le nombre de pixels noirs dans la fenêtre, r est le nombre de pixels noirs situés aux coins de la fenêtre. La suppression du bruit de type poivre est effectuée de la même manière mais sur l'image inversée.

Après avoir nettoyer l'image, les composantes très petites et extrêmement grandes (notamment les lignes graphiques très longues) sont exclues.

4.4 Extraction de la formule chimique

Comme illustré dans la Figure 4.1, l'extraction de la formule chimique suit les étapes suivantes : segmentation, caractérisation et classification. L'étape de segmentation vise à extraire des structures linéaires servant comme unités de base pour la classification. Cette dernière, quant à elle, permet d'assigner à chaque bloc, la classe Texte ou Graphique en se basant sur des caractéristiques discriminantes.

4.4.1 Segmentation

4.4.1.1 Granularité de la segmentation : les structures linéaires

Quand l'objectif de la segmentation est de produire des structures qui seront utilisées comme unité de base pour une opération de classification, la question principale à laquelle nous devons répondre est : quel est le degré de granularité le plus approprié pour faire cette classification ? En effet, un bon choix de la granularité est déterminant pour espérer atteindre un bon niveau de séparation.

Dans la littérature, plusieurs niveaux de granularité ont été adoptés pour des objectifs de classification, tels que : les pixels, les composantes connexes ou les caractères, des blocs de tailles fixes appelés aussi sites, les mots ou les pseudo-mots et les lignes. Les pixels sont utilisés comme unité de base dans [Kumar2007, Chaudhury2009] pour une classification visant la segmentation de documents manuscrits en fond, texte et graphique (ou ton continu). L'étiquetage des composantes connexes est le procédé le plus utilisé en segmentation d'images de documents. Elles ont été utilisées pour la classification texte/graphique dans [Fletcher1988, Barlas2014, Mollah2009] ou la séparation manuscrit/imprimé dans [Kandan2007]. Le caractère a été utilisé dans [Fan1998] pour la segmentation manuscrit/imprimé dans des documents chinois où les caractères sont faciles à extraire. Des sites de taille 9×9 ont été utilisés dans [Nicolas2006] pour l'étiquetage des principales zones d'intérêts, notamment les zones de texte et de rature dans les manuscrits de l'auteur Gustave Flaubert. D'autres de taille 16×16 ont été utilisés dans [Tony2005] pour la classification du contenu des images générées par ordinateur en deux classes : texte ou illustration. Les mots ou les pseudo-mots sont eux aussi utilisés pour la discrimination manuscrit/imprimé dans [Zheng2004]. Les lignes ont été utilisées pour la séparation texte/graphique dans des pages de magazines [Pavlidis1992] et pour l'extraction des formules mathématiques dans des pages de journaux scientifiques [Jin2003].

En conclusion, le choix du niveau de la segmentation dépend de la structure des documents traités, c'est-à-dire de la disposition et de la composition des régions à extraire. Dans les documents de chimie, les structures linéaires représentent le niveau de segmentation le plus stable et le plus approprié en vue de l'extraction de la formule chimique.

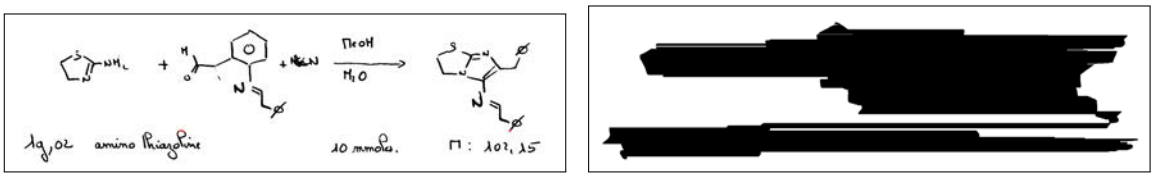
4.4.1.2 Segmentation de la page en structures linéaires

L'algorithme adopté pour cette segmentation doit satisfaire les deux contraintes suivantes :

1. toutes les composantes connexes horizontalement alignées doivent être regroupées dans une même structure linéaire puisqu'elles sont probablement de même type ;
2. des composantes connexes susceptibles d'être de types différents doivent appartenir à des structures linéaires différentes. Ceci consiste à empêcher le regroupement de composantes connexes verticalement éloignées.

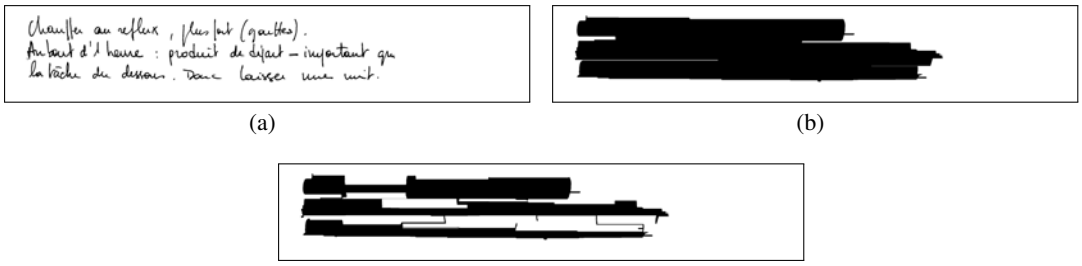
Pour cela, nous avons proposé un algorithme hybride en deux passages.

1. un premier passage ascendant qui part des composantes connexes et essaie de construire des structures linéaires en se basant sur le critère d'alignement horizontal ;
2. un deuxième passage descendant qui consiste à diviser les structures linéaires obtenues lorsqu'elles se trouvent fusionnées. La fusion (sous-segmentation) constitue un problème qui affecte le résultat de la classification du fait qu'elle peut engendrer une structure linéaire mixte. En revanche, la sur-segmentation ne pose pas de problème puisqu'au moment de la classification, les deux parties de la même structure linéaire (sur-segmentée) vont avoir la même classe et seront regroupées après la classification.



(a) (b)

FIGURE 4.4 – Fusion des deux structures linéaires en utilisant un seuil égal à la largeur de la page. (a) image binaire avec en rouge des pixels alignés mais appartenant à deux structures linéaires différentes, (b) résultat du lissage horizontal de l'image.



(a) (b) (c)

FIGURE 4.5 – (a) Image binaire, (b) résultat d'un lissage horizontal en utilisant un seuil égal à la largeur de la page, (c) résultat d'un lissage horizontal en utilisant un seuil plus fin (calculé à partir des distances entre les composantes connexes alignées).

Premier passage : lissage horizontal

Le lissage horizontal permet de créer des bandes noires horizontales en comblant les espaces entre les paires de pixels séparés par une distance inférieure à un seuil donné. L'extraction des composantes connexes de l'image résultante permet d'obtenir une première détection des structures linéaires.

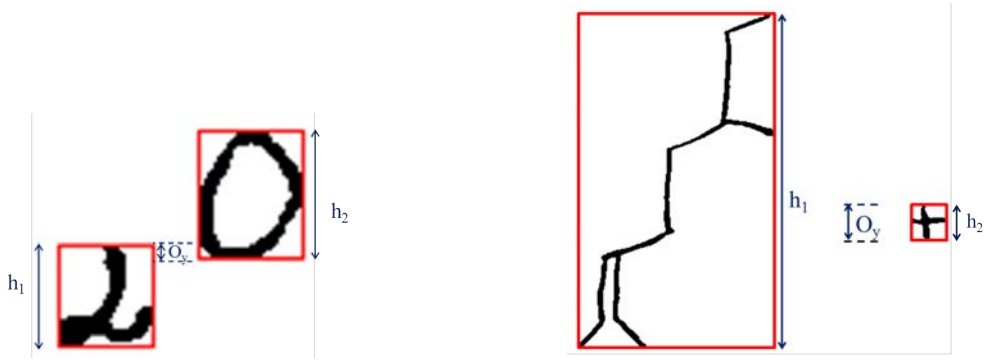
La principale difficulté dans l'utilisation de cet algorithme est le réglage du seuil de lissage. Même si dans notre cas, les structures linéaires s'étalent horizontalement sur toute la page et le choix d'un seuil égal à la largeur de la page semble être approprié, ce choix conduira souvent à des erreurs de fusion (qu'on aurait pu éviter en utilisant un seuil plus fin) à cause de la présence de pixels alignés mais qui appartiennent à des composantes connexes situées dans deux structures linéaires adjacentes (voir Figure 4.4). De plus, les fusions engendrées en utilisant une telle valeur de seuil sont plus difficiles à détecter et à corriger que celles engendrées en utilisant un seuil plus fin (voir Figure 4.5).

Pour cela, nous avons choisi un seuil S_l égal à la valeur maximale des distances entre toutes les paires de composantes adjacentes qui sont horizontalement alignées.

Soient deux composantes connexes $C_1(x_1, y_1, w_1, h_1)$ et $C_2(x_2, y_2, w_2, h_2)$, où x_i, y_i sont respectivement l'abscisse et l'ordonnée du point supérieur gauche de la boîte englobante de la composante C_i, et w_i et h_i représentent respectivement la largeur et la hauteur de la composante C_i. L'alignement horizontal de C_1 et C_2 est déterminé en fonction du taux de chevauchement vertical défini par l'équation 4.2.

$$ovlp_y(c_1, c_2) = \frac{O_y}{\min(h_1, h_2)} \tag{4.2}$$

où O_y est la hauteur du chevauchement vertical des boîtes englobantes de C_1 et C_2 (voir Figure 4.6). Le taux de chevauchement est normalisé en le divisant par le minimum des hauteurs des composantes connexes pour pouvoir regrouper des composantes dont la configuration spatiale est similaire à la fi-



(a) (b)

FIGURE 4.6 – Illustration du chevauchement vertical entre paires de composantes connexes.

(a) (b) (c)

FIGURE 4.7 – (a) Image nettoyée, (b) application d'un lissage horizontal et extraction des composantes connexes dans l'image lissée, (c) localisation (en rouge) des structures linéaires dans l'image initiale.

gure 4.6(b). Une telle configuration est très fréquente notamment dans le bloc contenant la formule chimique. Après avoir calculé les taux de chevauchement, nous considérons toutes les paires de composantes connexes adjacentes qui présentent un chevauchement vertical significatif (supérieur à un seuil fixé expérimentalement à 0.3) pour déterminer la valeur maximale des distances horizontales qui les séparent. Cette valeur est considérée comme un seuil de lissage horizontal de l'image.

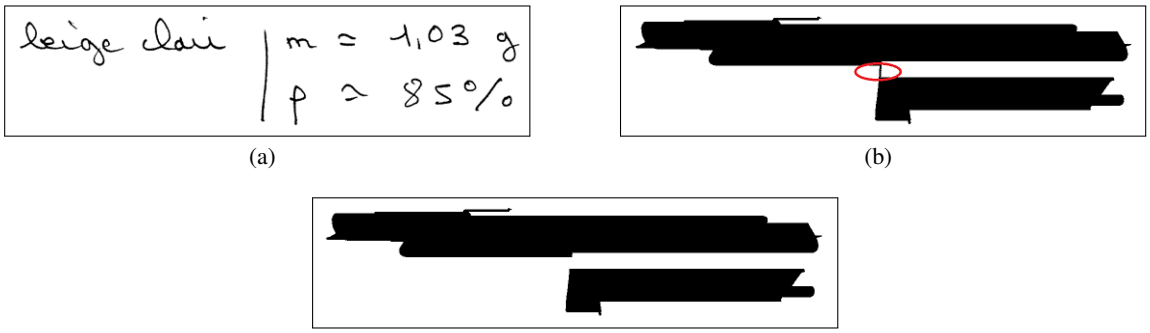
Un exemple d'extraction de structures linéaires par un lissage horizontal est illustré dans la Figure 4.7).

Deuxième passage : correction des sous-segmentations

Les structures linéaires du document peuvent se toucher ou se chevaucher verticalement. Ceci peut être à l'origine d'erreurs de fusion : les structures en question sont combinées en un seul bloc (voir figure 4.9). Nous nous sommes inspirés des travaux de Huaigu et al. [Huaigu2007] pour développer un algorithme de correction de ce type d'erreur. L'algorithme proposé repose sur les deux points suivants :

— une connexion ou un chevauchement entre deux structures linéaires est présente sous forme d'une courte séquence horizontale de pixels noirs dans l'image lissée I_l. On note par $I_l(i, j_1 : j_2)$ cette séquence de pixels noirs, située dans la ligne i entre les colonnes j_1 et j_2 de l'image I_l ;

— dans l'image initiale I, la zone de chevauchement entre deux structures linéaires est une zone à



(a) (b) (c)

FIGURE 4.8 – Correction de la fusion de structures linéaires due à une connexion (encerclée en rouge) par les ascendants et les descendants de deux structures adjacentes, (a) image contenant deux structures linéaires, (b) résultat du lissage horizontal, (c) correction de l'erreur de fusion.

faible densité de pixels. De plus, la distance horizontale entre deux pixels adjacents dans cette zone est souvent plus grande que celle qui sépare deux pixels adjacents dans une même structure linéaire. Ceci engendre, dans l'image lissée, plus qu'une connexion verticale entre les deux structures linéaires en question. Soit $nbpixel_I(i)$ le nombre de pixels dans la ligne i de l'image I et soit $cross\_count_{I_l}(i)$ le nombre de transitions blanc-noir (qui permet de compter le nombre de connexions) dans l'image lissée I_l au niveau de la ligne i.

La correction des erreurs de fusion consiste à parcourir l'image lissée ligne par ligne et à transformer en blanc toute séquence $I_l(i, j_1 : j_2)$ si l'une des conditions suivantes est satisfaite :

$$j_2 - j_1 < S_l \text{ et } I(i, j_1 : j_2) < S_1 \tag{4.3a}$$

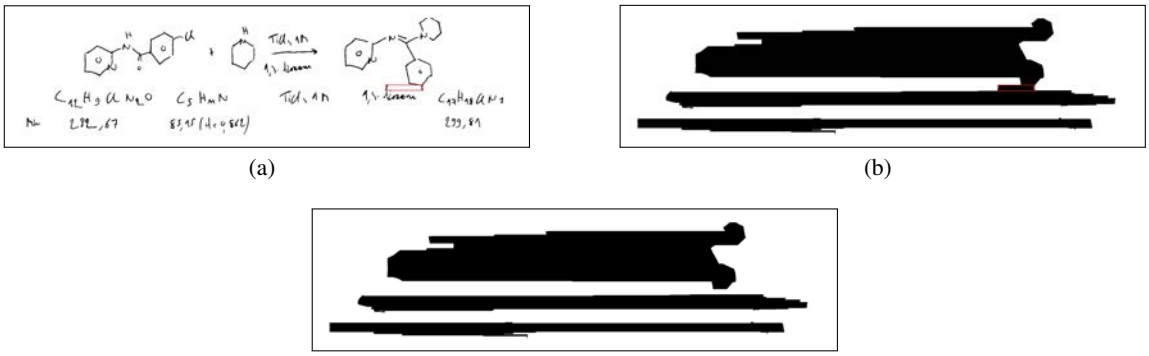
$$(cross\_count_{I_l}(i) \geq 2) \text{ et } (nbpixel_I(i) < S_2) \tag{4.3b}$$

— la condition exprimée par l'équation 4.3a permet de séparer deux structures linéaires qui se touchent ou se chevauchent par deux traits, l'un se prolongeant au-dessus de la structure inférieure et l'autre, en-dessous de la structure supérieure (voir Figure 4.10(b)). Ainsi, S_1 est fixé à une valeur égale à 2 fois l'épaisseur moyen du trait d'écriture estimé sur toute la page ;
— la condition 4.3b permet de séparer deux structures linéaires qui se chevauchent même sans se toucher (voir Figure 4.10(c)) par plusieurs descendants et ascendants . S_2 est fixé à une valeur égale au nombre moyen de pixels par ligne (de pixels), contenus dans la plus courte structure linéaire (obtenue par l'algorithme de lissage).

4.4.2 Caractérisation des structures linéaires

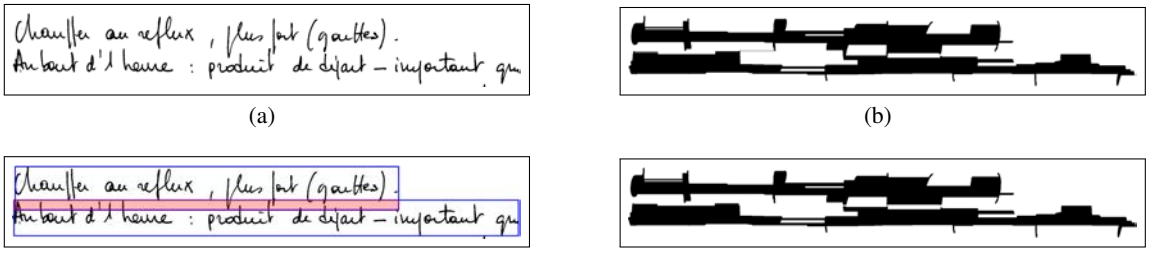
Par rapport à une structure textuelle, une structure graphique est caractérisée par :
— la présence de longues séquences de pixels noirs alignés horizontalement et/ou verticalement ;
— un nombre réduit de transitions blanc-noir horizontales ;
— la présence de plusieurs segments de droites.

En se basant sur ce constat, nous avons extrait de chaque structure linéaire les descripteurs suivants :
— mean_{h_rl} resp. mean_{v_rl} : la moyenne run-length horizontal resp. vertical. Ces descripteurs sont calculés à partir des deux matrices de run-length horizontale et verticale de la structure linéaire ;



(a) (b) (c)

FIGURE 4.9 – Correction de la fusion de structures linéaires due à un chevauchement (encadré en rouge) par un ascendant et un descendant de deux structures adjacentes, (a) image contenant trois structures linéaires, (b) résultat du lissage horizontal donnant uniquement deux structures : la première et la deuxième sont fusionnées, (c) correction de l'erreur de fusion.



(a) (b) (c) (d)

FIGURE 4.10 – Correction de la fusion de structures linéaires due à un chevauchement par les ascendants et les descendants de deux structures adjacentes. (a) Image contenant deux structures linéaires, (b) résultat du lissage horizontal avec une erreur de sous-segmentation, (c) Illustration de la zone de chevauchement des deux lignes (en rouge), (d) correction de l'erreur de fusion.

TABLE 4.1 – Performances des différents classifieurs pour la discrimination Texte/Graphique

	Texte		Graphique		Taux global de bonne classification
Classifieur	Rappel	Précision	Rappel	Précision	
SVM à noyau polynomial	97.2	96.4	96.4	97.2	96.8
SVM à noyau RBF	**97.6**	**96.8**	**96.8**	**97.6**	**97.2**
PMC	95.2	96.4	96.4	95.3	95.8
Arbre de décision (J48)	95.2	94.4	94.4	94.8	94.8
LogitBoost	95.6	96.4	96.4	95.6	96.0
Forêt aléatoire	96.8	95.3	95.2	96.7	96.0

— nb_{tr} : le nombre de transitions blancs-noirs horizontales ;
— nb_{seg} : le nombre de segments de droites. Ce descripteur est calculé en utilisant une version probabiliste de la transformée de Hough.

4.4.3 Classification des structures linéaires

Cette opération est classiquement réalisée à l'aide d'un classifieur qui permet d'affecter la classe Texte ou Graphique à chacune des structures linéaires en se basant sur les valeurs de ses caractéristiques. Le défi rencontré lors de cette classification est lié à la répartition déséquilibrée des structures linéaires entre les deux classes. En effet, dans chaque document, la classe Graphique est souvent représentée par une voire deux structures linéaires tandis que la classe Texte est représentée par autant de structures linéaires que de lignes. Il en résulte que, dans toute la base, seulement une structure sur 15 est de type graphique. Ce problème peut influencer les performances du classifieur.

Dans ce qui suit, nous allons tout d'abord, choisir expérimentalement le classifieur le plus performant parmi un ensemble de classifieurs usuels. Ensuite, nous allons étudier le problème de données déséquilibrées et présenter la solution que nous avons adoptée.

4.4.3.1 Choix du classifieur

Les classifieurs les plus courants ont été testés en utilisant un même jeu de données constitué de 500 structures linéaires homogènes, réparties équitablement entre les deux classes : Texte et Graphique, en utilisant l'outil WEKA [Hall2009]. Nous avons employé la technique de validation croisée à 10 sous-ensembles pour l'évaluation de ces classifieurs. Cette technique consiste à répartir aléatoirement les structures linéaires en 10 sous-ensembles et de faire la classification 10 fois. A chaque fois, 9 sous-ensembles sont utilisés pour l'apprentissage et un seul pour les tests. La moyenne des performances obtenues est considérée pour l'évaluation du classifieur. Les résultats obtenus sont reportés dans le tableau 4.1.

En se basant sur ces résultats, nous avons retenu le classifieur SVM à noyau RBF (Radial Basis Function kernel) pour la classification des structures linéaires. Il convient de noter que ces résultats sont obtenus avec des données équilibrées manuellement et sont utilisés juste pour sélectionner le classifieur à utiliser.

4.4.3.2 Problème de données déséquilibrées

Comme la plupart des algorithmes de classification visent à minimiser le taux de mauvaise classification global, les résultats obtenus par des classifieurs appris sur des données déséquilibrées sont généralement insatisfaisants. Ceci peut être illustré par le scénario suivant : disposant d'un corpus de structures linéaires où le ratio du nombre d'exemples dans la classe Graphique (classe minoritaire) par

rapport à celui de la classe Texte (classe majoritaire) est égal à $1/15$, si on décide de considérer toutes les structures linéaires comme Texte, on aura un taux global de bonne classification de $93,33\%$. Ce taux est trompeur car il montre une performance considérable du classifieur alors qu'en réalité, on n'a réussi à identifier aucune structure graphique.

Les méthodes traitant du problème de déséquilibre des données en apprentissage supervisé peuvent être regroupées en deux catégories :

— **Au niveau des données**, les approches consistent à rééquilibrer les classes par les techniques de sur-échantillonnage et/ou sous-échantillonnage. La première consiste à créer de nouvelles instances de la classe minoritaire tandis que la deuxième vise à éliminer quelques instances de la classe majoritaire. Ces deux techniques peuvent être effectuées de façon aléatoire ou dirigée.

— **Au niveau des algorithmes**, les solutions proposées consiste à modifier les algorithmes d'apprentissage afin de prendre en compte le déséquilibre. L'idée principale consiste à introduire des coûts spécifiques à chaque erreur de classification et modifier les algorithmes d'apprentissage de façon à ce qu'ils tiennent compte de ces coûts. Ainsi, au lieu de prédire la classe ayant la probabilité la plus élevée, la classe ayant le coût minimal est prédite. Il existe 3 façons d'introduire cette notion de coût, d'après [He2008a]. La première consiste à définir une matrice de coûts qui sera utilisée comme une forme de pondération de la probabilité de décision [Zadrozny2003]. La seconde se compose de divers méta-techniques où les algorithmes d'apprentissage standards sont combinés avec la théorie des ensembles pour développer des classifieurs sensibles aux coûts [Domingos1999]. La dernière intègre des fonctions sensibles aux coûts directement dans les algorithmes de classification [He2008a]. Quelques adaptations des algorithmes de classification telles que les arbres de décision [Ling2004, Maloof2003], les SVMs [Xiao-yan2007] et les réseaux de neurones [Kukar1998], ont été proposées pour traiter avec une bonne performance le cas des données déséquilibrées.

Les performances des solutions au niveau des algorithmes sont fortement influencées par les valeurs exactes de coûts de mauvaise classification. Ainsi, la détermination de ces valeurs est une étape délicate qui nécessite une expertise dans le domaine étudié. Dans le cas d'absence de connaissances suffisantes pour un choix judicieux des valeurs des différents coûts, les solutions au niveau des données s'avèrent les plus appropriées. Pour cela, nous nous sommes orientés vers les approches au niveau de données dont nous détaillons davantage les principales techniques dans la suite de cette section.

Sur-échantillonnage et sous-échantillonnage aléatoires. Comme son nom l'indique, le mécanisme de sur-échantillonnage aléatoire consiste à ajouter un ensemble d'exemples aléatoirement sélectionnés à la classe minoritaire. Cet ajout se fait par duplication des exemples sélectionnés. Ceci permet d'équilibrer la distribution des données au niveau souhaité. Le sous-échantillonnage quant à lui, est un mécanisme simple qui consiste à supprimer quelques exemples aléatoirement choisis de la classe majoritaire. Les deux mécanismes altèrent la taille de la base originale. Par conséquent, ils génèrent des problématiques qui peuvent potentiellement nuire à l'apprentissage. Le sur-échantillonnage est généralement à l'origine d'un problème de sur-apprentissage (en anglais, overfitting) [Mease2007]. En particulier, ce problème est plus sérieux quand le classifieur utilisé produit plusieurs règles associées aux différentes copies du même exemple. Le problème du sous-échantillonnage est relativement plus évident. En effet, enlever des exemples de la classe majoritaire, risque de causer une perte d'informations importantes qui devraient contribuer positivement à l'élaboration de la bonne règle de classification.

Sous-échantillonnage dirigé. L'objectif des méthodes dirigées de sous-échantillonnage est de remédier au problème de perte d'information causé par la méthode aléatoire. Deux algorithmes de sous-échantillonnage dirigé qui ont montré de bonnes performances, ont été proposés dans [Liu2006] : EasyEnsemble et BalanceCascade. Le premier peut être considéré comme un algorithme d'apprentissage non supervisé qui effectue plusieurs sous-échantillonnages aléatoires indépendants pour entraîner plusieurs classifieurs dont il combine les sorties. Le deuxième est basé sur une approche supervisée qui utilise un ensemble de classifieurs pour sélectionner automatiquement les exemples à considérer de la classe majoritaire. Il s'agit d'une procédure itérative où à chaque étape, les exemples de la classe majoritaire qui sont correctement classés par un classifieur déjà appris, sont ignorés dans l'étape suivante. Une autre technique de sous-échantillonnage dirigé basée sur l'algorithme KPPV, a été aussi proposée dans [Zhang2003]. L'idée principale est de choisir des exemples de la classe majoritaire qui sont proches des exemples de la classe minoritaire. Selon les caractéristiques des données étudiées, plusieurs variantes de KPPV ont été proposées.

Sur-échantillonnage par génération de données. Cette technique repose sur l'échantillonnage synthétique qui consiste à produire "artificiellement" des données pour augmenter le nombre d'exemples dans la classe minoritaire. L'algorithme SMOTE (Synthetic Minority Oversampling Technique) est un algorithme très utilisé dans l'échantillonnage synthétique. Cet algorithme consiste à créer de nouvelles instances de la classe minoritaire en se basant sur les similarités entre les caractéristiques des exemples déjà existants dans cette classe. Une nouvelle instance créée par cet algorithme est une combinaison linéaire d'une instance existante (germe) et de la différence entre celle-ci et une autre prise aléatoirement parmi ses K plus proches voisins. La qualité des instances créées par cet algorithme est influencée par les exemples utilisés comme germe. Afin d'améliorer le choix des germes, des algorithmes adaptatifs ont été proposés tels que Borderline-SMOTE [Han2005] et ADASYN [He2008b]. Un autre problème de l'échantillonnage synthétique est dû au fait que les nouvelles instances peuvent augmenter le chevauchement inter-classes.

4.4.3.3 Règle de sélection

Pour résoudre le problème de déséquilibre des données, nous avons opté pour un sous-échantillonnage de la classe majoritaire (Texte). Nous nous sommes basés sur le travail de Zhang [Zhang2003] où une étude comparative de différentes méthodes de sous-échantillonnage a été menée. Cette étude a montré que les deux méthodes de sélection suivantes ont donné les meilleurs résultats :

— sélection aléatoire des exemples ;
— sélection des exemples (de la classe majoritaire) qui sont proches de l'ensemble des exemples de la classe minoritaire.

Nous avons testé ces deux méthodes afin de choisir la plus appropriée aux données que nous traitons.

Pour la deuxième méthode, une règle de sélection doit être définie pour guider le sous-échantillonnage de la classe Texte. Pour ce faire, nous nous sommes basés sur le constat fait dans la section 4.4.2, pour définir un score en fonction des descripteurs des structures linéaires, qui permet de discerner le plus possible les deux classes. L'idée est inspirée de la technique d'analyse discriminante descriptive, pour proposer un nouveau système de représentation des structures linéaires à partir d'une combinaison linéaire des descripteurs, et permettre une représentation graphique dans un espace réduit.

D'abord, les valeurs des descripteurs sont normalisées entre 0 et 1 en utilisant la formule 4.4

$$d_{i\_n} = \frac{d_i - d_{min}}{d_{max} - d_{min}} \tag{4.4}$$

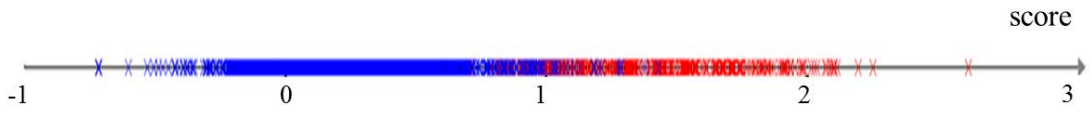


FIGURE 4.11 – Distribution des structures linéaires en fonction du score. En rouge les instances de la classe minoritaire (Graphique) et en bleu, les instances de la classe majoritaire (Texte).

TABLE 4.2 – Description de la base utilisée pour la comparaison des méthodes de sous-échantillonnage

	Classe	nombre d'échantillons
Base d'apprentissage	Texte	4190
	Graphique	280
Base de test	Texte	3783
	Graphique	220

où d_i désigne la valeur originale d'un descripteur d, et d_{i_n} est la valeur normalisée, et d_{max} et d_{min} dénotent respectivement la valeur maximale et minimale de d.

Ensuite, le score est calculé sur chaque structure linéaire sl, selon l'équation 4.5.

$$score(sl) = nb_{seg} + mean_{rl} - nb_{tr} \tag{4.5}$$

où $mean_{rl}$ est la moyenne des deux descripteurs $mean_{h\_rl}$ et $mean_{v\_rl}$.

La représentation graphique des structures linéaires en fonction du score est illustrée dans la Figure 4.11. Les exemples de chaque classe sont regroupés ensemble (avec une zone de chevauchement au milieu), ce qui montre le pouvoir discriminatif de ce score.

En se basant sur cette observation, nous proposons une règle de sélection qui doit permettre de sélectionner les structures linéaires textuelles qui sont proches de celles graphiques. Il s'agit alors d'écarter les structures dont le score est inférieur à un seuil S_s. Ce seuil doit être fixé de manière à ce que :

1. aucune structure graphique ne soit écartée. Ceci peut être exprimé par l'équation : $S_s \leq s_{min}$ où $s_{min} = \min(\{score(sl\}_{sl \in Graphique})$ est la valeur minimale des scores de toutes les structures linéaires graphiques ;
2. le nombre de structures textuelles sélectionnées doit être plus ou moins égal au nombre total des structures graphiques. Ceci revient à minimiser $|nbT_{selected} - nbG|$ où $nbT_{selected} =$ card$(\{sl / score(sl) \geq S_s$ et $sl \in Texte\}_{sl})$ est le nombre de structures linéaires textuelles ayant un score supérieur au seuil S_s, et nbG est le nombre total des structures graphiques.

Après avoir défini la règle de sélection permettant de sous-échantillonner la classe majoritaire, nous avons effectué trois scénarios d'apprentissage : 1) le classifieur (un SVM) est entraîné sur toute la base d'apprentissage, 2) le classifieur est entraîné sur la base d'apprentissage où la classe majoritaire est sous-échantillonnée aléatoirement, puis 3) en utilisant la règle de sélection. La base utilisée pour l'apprentissage des classifieurs est décrite dans le tableau 4.2. Pour pouvoir comparer les résultats obtenus, les classifieurs ainsi appris ont été testés sur la même base de test décrite dans le tableau 4.2. Les résultats obtenus sont présentés dans le tableau 4.3.

Interprétation

En utilisant la base complète (sans résoudre le problème de déséquilibre), les résultats obtenus confirment les résultats théoriques attendus. En effet, le classifieur obtenu est biaisé vers la classe majoritaire avec

TABLE 4.3 – Résultats obtenus pour les différents scénarios d'apprentissage

	Classe	Précision	Rappel	F-mesure
Base complète	Texte	99.1	99.6	99.4
	Graphique	93.0	84.9	88.7
Sous-échantillonnage aléatoire	Texte	99.9	98.7	99.3%
	Graphique	80.8	98.2	88.6
Sous-échantillonnage dirigé	Texte	99.2	99.6	**99.4**
	Graphique	93.0	95.8	**94.3**

une F-mesure de 99.4% contre une F-mesure de 88.7% pour la classe minoritaire. Concernant les résultats obtenus dans le cas du sous-échantillonnage aléatoire, il convient de noter que les résultats reportés dans ce tableau représentent la moyenne de 10 sous-échantillonnages aléatoires (un seul sous-échantillonnage aléatoire ne permet pas de juger de l'apport de cette méthode pour rééquilibrer la base). Les résultats obtenus dans ces 10 expérimentations varient d'un sous-échantillonnage à l'autre. Parmi les 10, un seul a donné des résultats meilleurs que le sous-échantillonnage dirigé, mais en moyenne, il donne des résultats plus mauvais que celui du sous-échantillonnage dirigé. Ce dernier donne la meilleure F-mesure de la classe minoritaire (Graphique) qui constitue la classe d'intérêt de notre étude. Pour cela, nous avons utilisé cette technique pour rééquilibrer le nombre des structures linéaires dans les deux classes.

4.5 Expérimentation et résultats

La base ChemicalPage composée de 500 documents a été utilisée pour l'expérimentation de notre système.

4.5.1 Évaluation de la segmentation

L'évaluation de cette étape est effectuée de manière manuelle et subjective. Une segmentation est qualifiée de "bonne" si elle fournit un nombre de structures linéaires hétérogènes (contenant à la fois des parties de la formule chimique et de la région de texte) minime. Elle est qualifiée de "mauvaise" dans le cas contraire, c'est-à-dire, plusieurs structures linéaires mixtes sont obtenues.

Cette évaluation est effectuée en considérant toutes les images des structures linéaires issues de la segmentation de l'ensemble des 500 documents. Nous calculons le taux de bonne segmentation défini par l'équation 4.6.

$$\tau_{seg} = \frac{|sl_C|}{|sl|} \tag{4.6}$$

où $|sl_C|$ est le nombre de structures linéaires correctes, c'est-à-dire contenant uniquement du graphique ou du texte, et $|sl|$ est le nombre total des structures linéaires extraites dans tous les documents. Le taux de segmentation obtenu est de $97,15\%$. Le taux d'erreur de $2,85\%$ qui représente le pourcentage des structures linéaires mixtes, est principalement dû à un grand chevauchement vertical du texte situé au-dessous ou au-dessus de la formule chimique avec cette dernière. La Figure 4.12 illustre un exemple d'erreur de segmentation.

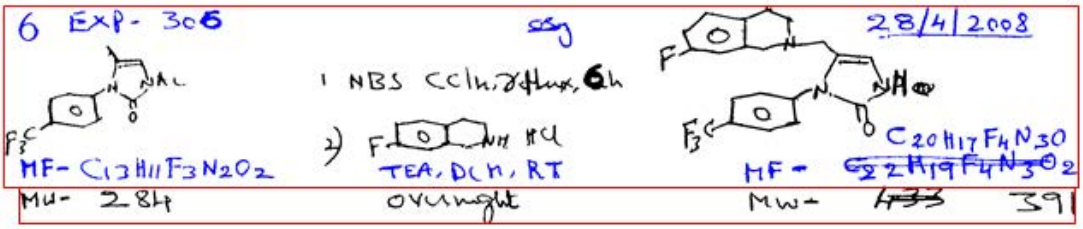


FIGURE 4.12 – Exemple d'erreur de segmentation, avec en rouge les boîtes englobantes des structures linéaires extraites. Celle en haut est incorrecte (hétérogène) : elle contient toute la formule chimique et d'autres composantes textuelles (en bleu).

4.5.2 Apprentissage du classifieur

A partir de la base ChemicalPage, 280 documents ont été utilisés pour l'apprentissage du SVM [8] qui sera utilisé pour la classification. Les structures linéaires extraites sur ces documents sont d'abord corrigées manuellement (de façon à ce que chacune soit homogène). Ensuite, elles sont annotées à partir des documents de vérité au niveau régions selon qu'elles appartiennent à la classe Graphique ou à la classe Texte. Les structures linéaires textuelles sont sous-échantillonnées en utilisant la règle de sélection décrite dans la section 4.4.3.3. Enfin, le SVM est entraîné sur l'ensemble composé par les structures textuelles sous-échantillonnées et toutes les structures graphiques. Ses hyper paramètres C (paramètre de pénalité) et γ (paramètre de noyau) sont déterminés empiriquement en cherchant à minimiser le taux d'erreur. Les valeurs retenues sont $C = 10$ et $\gamma = 1.05$.

4.5.3 Évaluation de l'extraction de la formule chimique

Cette évaluation est faite sur les 220 documents restants suivant une métrique inspirée de celle employée dans [Shafait2008, Chen2011]. La métrique est basée sur le chevauchement entre les boîtes englobantes des formules chimiques de la vérité terrain et celles détectées par notre système.

Soient $F\_G = \{F\_G_i\}_i$ l'ensemble des formules chimiques définies dans les documents de vérité et $F\_S = \{F\_S_j\}_j$ l'ensemble des formules chimiques extraites par notre système. Soient G_i et S_j respectivement les boîtes englobantes de F_G_i et F_S_j. Le taux de chevauchement de F_G_i et F_S_j est donné par l'équation 4.7 :

$$O(F\_G_i, F\_S_j) = \frac{2 \times \mathcal{A}(G_i \cap S_j)}{\mathcal{A}(G_i) + \mathcal{A}(S_j)} \tag{4.7}$$

où $\mathcal{A}()$ désigne l'aire de la zone entre parenthèses. Le taux de chevauchement O varie entre 0 et 1 et mesure à quel degré la zone détectée est superposable à celle de la vérité terrain. En particulier, pour deux zones strictement disjointes, $O = 0$ et pour deux zones parfaitement superposables, $O = 1$.

En se basant sur le taux de chevauchement, nous définissons les différentes configurations dans lesquelles peuvent se retrouver les éléments de F_S par rapport aux éléments de F_G :

— Correct (C) : F_G_i est correctement détectée s'il existe une formule chimique F_S_j identifiée par le système telle que : $O(F\_G_i, F\_S_j) \geq S_c$; où S_c est un seuil de chevauchement à partir duquel la détection est considérée correcte.

— Partiel (P) : F_S_j correspond à une détection partielle s'il existe une formule chimique F_G_i de la vérité terrain telle que : $F\_S_j \subset F\_G_i$ et $S_p \leq O(F\_G_i, F\_S_j) < S_c$; où S_p est un seuil de chevauchement à partir duquel la détection est considérée partielle.

8. Utilisation de la librairie C++ LibSVM [Chang2001]

TABLE 4.4 – résultats de l'extraction des formules chimiques

$Rappel(\%)$	$Précision(\%)$	$err_P(\%)$	$err_S(\%)$	$err_F(\%))$	$err_M(\%)$
$88,63$	$89,44$	$5,45$	$3,18$	$1,83$	2.72

— Sur-détection (S) : F_S_j correspond à une sur-détection s'il existe une formule chimique F_G_i de la vérité terrain telle que : $F\_G_i \subset F\_S_j$ et $S_p \leq O(F\_G_i, F\_S_j) < S_c$.
— Faux (F) : F_S_j correspond à une fausse détection si elle ne se chevauche pas suffisamment avec aucune formule chimique de la vérité, i.e. $O(F\_G_i, F\_S_j) < S_p, \forall i$.
— Manqué (M) : F_G_i est dite manquée si elle ne se chevauche avec aucune zone détectée par le système en tant que formule chimique, i.e. $O(F\_G_i, F\_S_j) < S_p, \forall j$.

Les valeurs de S_c et S_p sont fixées selon la précision souhaitée dans la détection de la zone. Par exemple, fixer S_c à 1 signifie qu'on désire être précis à 100% dans la délimitation de la région et plus cette valeur s'éloigne de 1, plus on est tolérant dans la détection. S_p correspond à un taux de chevauchement au-dessous duquel on considère que la zone détectée est fausse. Les valeurs de S_c et S_p sont fixées respectivement à 0.9. et 0.1.

Une fois les formules chimiques de la vérité et celles détectées par le système sont regroupées selon les cinq configurations, nous calculons le rappel, la précision et les taux d'erreurs err_P, err_A, err_F et err_M relatifs respectivement aux configuration P, A, F et M. Ces métriques sont définies selon les équations suivantes :

$$Précison = \frac{|C|}{|F\_S|} \tag{4.8a}$$

$$Rappel = \frac{|C|}{|F\_G|} \tag{4.8b}$$

$$err_P = \frac{|P|}{|F\_G|} \tag{4.8c}$$

$$err_S = \frac{|S|}{|F\_G|} \tag{4.8d}$$

$$err_F = \frac{|F|}{|F\_S|} \tag{4.8e}$$

$$err_M = \frac{|M|}{|F\_G|} \tag{4.8f}$$

où $|\ |$ désigne la cardinalité de l'ensemble donné en argument. Les tests effectués ont donné les résultats illustrés dans le tableau 4.4

Les erreurs obtenues sont dues à des erreurs de segmentation ou de classification ou de la combinaison de ces deux étapes. Dans ce qui suit, nous allons analyser les différents types d'erreurs.

Les sur-détections sont engendrées suite à des erreurs de segmentation similaires à celle présentée dans la figure 4.12. Dans un tel cas, nous obtenons une structure linéaire hétérogène (formule chimique + texte), mais classée comme étant une formule chimique car elle est composée majoritairement de graphique.

Les détections partielles sont produites suite à des cas de segmentation où la formule chimique est décomposée en deux (ou plus) structures linéaires à cause de la présence d'espace vertical important entre

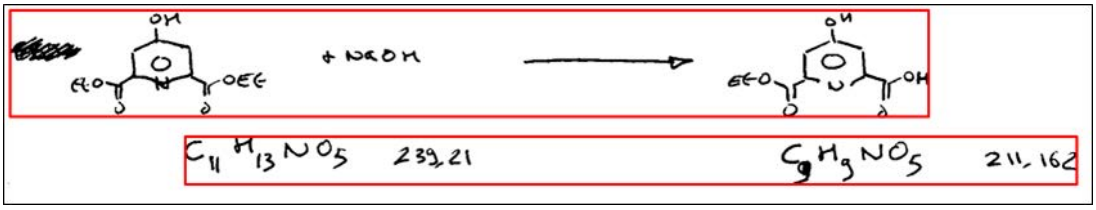


FIGURE 4.13 – Exemple de détection partielle d'une formule chimique. Cette dernière est décomposée en deux structures linéaires (les boîtes englobantes correspondantes sont en rouge), mais seulement la première est classée comme graphique. Cette erreur pourrait être corrigée par la suite, après la détection du tableau et la non attribution de cette ligne au tableau.

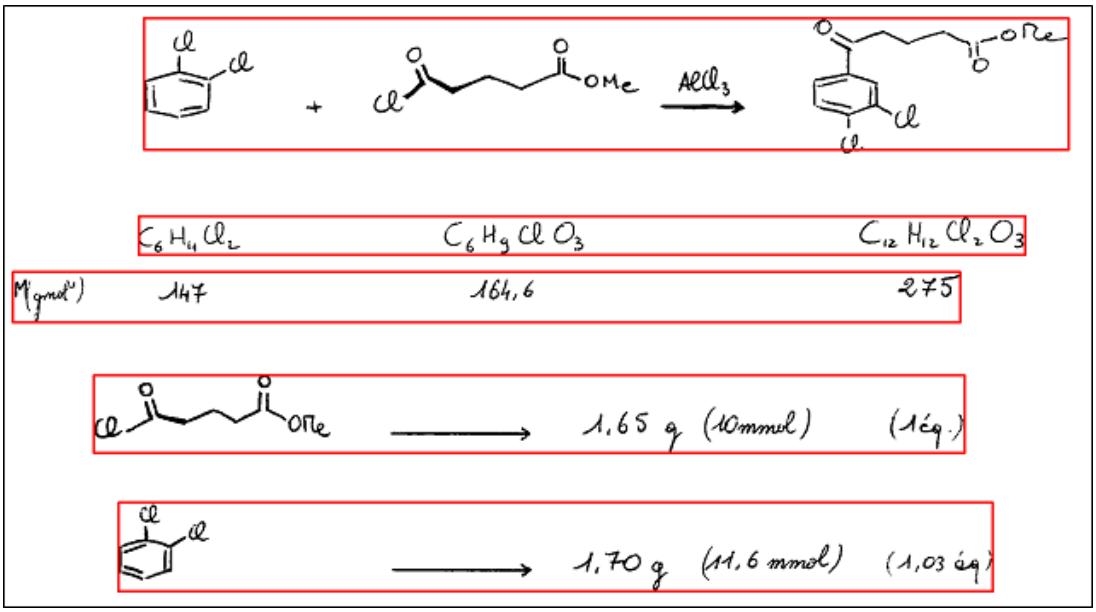


FIGURE 4.14 – Exemple de fausse détection d'une formule chimique. Les structures linéaires issues de la segmentation sont encadrées en rouge. L'avant dernière structure linéaire est classée en tant que formule chimique alors qu'en réalité, elle représente une ligne de tableau.

ses composantes. Parmi ces structures, il peut y avoir quelques-unes qui sont composées majoritairement de texte, elles seront classées comme du texte. Ainsi, la formule chimique est partiellement détectée puisque seulement une partie (celle qui contient suffisamment de graphique) est correctement détectée (voir Figure 4.13).

Les fausses détections se produisent dans le cas où des schémas de molécules ou de petits dessins qui sont présents dans le texte sont regroupés avec peu de composantes textuelles adjacentes en une seule structure linéaire. Cette dernière sera classée en tant que formule chimique. La présence de ratures sous formes de segments de droites peut être aussi à l'origine de ce type d'erreur. La Figure 4.14 illustre une erreur de fausse détection.

Les formules chimiques manquées sont généralement des formules où les composantes graphiques sont de petites tailles et peuvent ressembler à des composantes textuelles. Par conséquent, le classifieur se trompe dans la classification des structures linéaires correspondantes. Ce type d'erreur peut aussi avoir lieu sur les formules chimiques composées majoritairement de texte. La Figure 4.15 illustre un exemple de formule chimique non détectée.

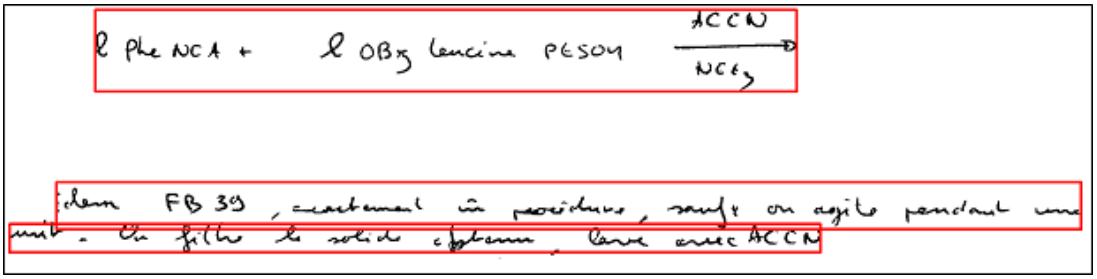


FIGURE 4.15 – la première structure linéaire est une formule chimique qui n'a pas été détectée.

4.6 Conclusion

Nous avons présenté dans ce chapitre les différentes étapes d'un système d'extraction de formules chimiques dans des documents manuscrits composites. Le problème a été considéré comme une tâche de classification texte/graphique. A la différence des méthodes existantes qui sont basées sur la classification des composantes connexes ou de zones de taille fixe ou variable, notre méthode est basée sur la classification de structures linéaires. Tenant compte de la structure des documents traités, ce niveau de segmentation s'avère le plus approprié pour cette classification. En effet, cette granularité n'est pas très grossière pour contenir des données mixtes (texte et graphique). Elle n'est pas non plus très fine pour être ambigüe au moment de la classification. Un algorithme hybride en deux passages a été mis en œuvre pour effectuer cette segmentation. Le premier passage basé sur la technique de lissage permet de regrouper des composantes connexes horizontalement alignées qui sont de même type pour former une même structure linéaire. Le deuxième passage permet de corriger d'éventuelles erreurs de sous-segmentation dues à des connexions ou des chevauchements entre des composantes verticalement adjacentes. L'avantage de notre méthode réside dans la détermination automatique des seuils utilisés dans les deux phases. Les structures linéaires ainsi obtenues sont classées en texte ou graphique. Au niveau de la classification, nous avons proposé une solution basée sur la technique de sous-échantillonnage dirigé pour résoudre le problème de classes déséquilibrées. Notre méthode de sélection des exemples de la classe majoritaire diffère de celles utilisée dans la littérature. Elle est basée sur un score calculé défini en fonction de quatre descripteurs de façon à pouvoir sélectionner des exemples informatifs de la classe majoritaire.

Chapitre 5

Détection de tableaux

Sommaire

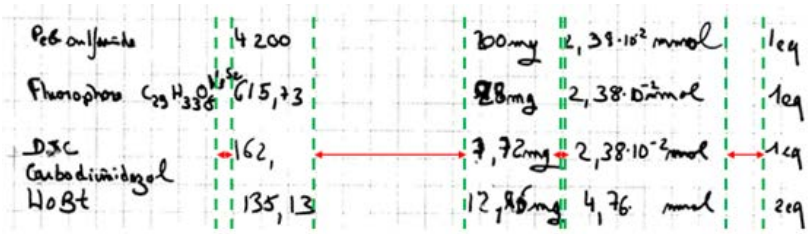
5.1 Introduction 71
5.2 Étiquetage de séquences de données 73
5.2.1 Les modèles génératifs 73
5.2.2 Les modèles discriminants 74
5.3 Les champs aléatoires conditionnels 74
5.3.1 Formalisme des CACs 74
5.3.2 Quelques applications de CACs dans l'ananlyse des images de documents . . . 76
5.4 Proposition d'un modèle CAC pour la détection de tableaux 79
5.4.1 Modélisation du problème 79
5.4.2 Extraction des descripteurs 80
5.4.3 Apprentissage du modèle 85
5.4.4 Décodage 87
5.5 Expérimentation et résultats 88
5.5.1 Évaluation de la segmentation 88
5.5.2 Évaluation de la détection de tableau 89
5.6 Conclusion 91

5.1 Introduction

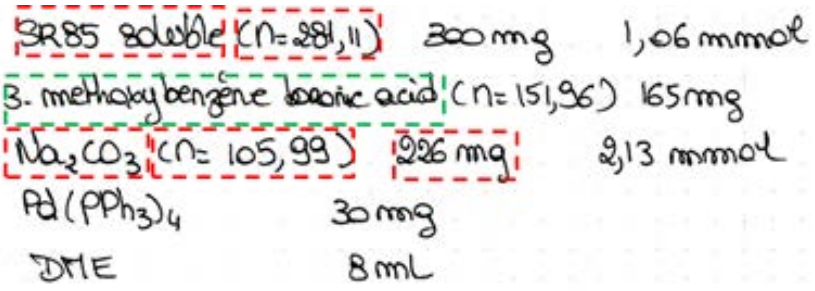
A l'issue de la séparation texte/graphique présentée dans le chapitre précédent, nous obtenons pour chaque document deux images séparées, l'une contenant un bloc graphique et l'autre contenant du texte. Nous nous focalisons dans ce chapitre sur la détection de tableaux dans les images contenant du texte. Nous abordons ce problème de la même manière que dans le chapitre précédent, comme un problème de classification qui consiste à séparer les lignes de texte de celles d'un tableau.

Dans les documents de chimie, les tableaux présentent plusieurs imperfections qui altèrent leurs structures physiques et rendent difficile leur localisation en utilisant les techniques classiques basées sur la détection de lignes graphiques (si elles existent), d'espaces et d'alignement de texte. Ces imperfections sont résumées dans les points suivants :

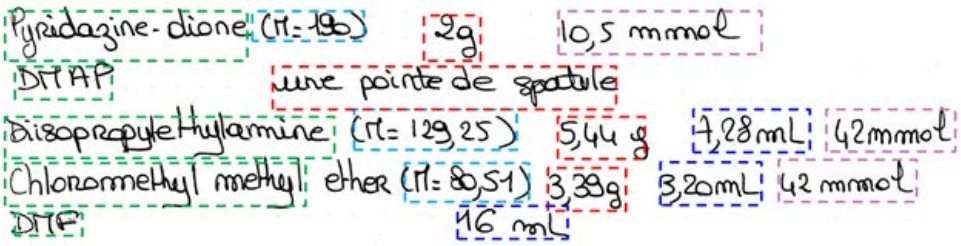
— non-régularité des espaces inter-colonnes (voir Figure 5.1(a)). Ceci rend difficile la distinction entre les espaces inter-mots et les espaces inter-cellules, une distinction souvent utilisée sur les lignes comme révélateur de la présence d'une ligne de tableau ;



(a)



(b)



(c)

FIGURE 5.1 – Exemples de tableaux présentant les imperfections de type : (a) non-régularité d'espace inter-colonnes, (b) cellules étendues sur plusieurs colonnes et (c) mauvais alignement vertical des cellules d'une même colonne.

— présence de cellules qui sont étendues sur deux ou (plusieurs) colonnes (voir Figure 5.1(b)). Il en résulte la présence de texte dans les espaces inter-colonnes, ce qui perturbe le raisonnement basé sur l'alignement des espaces ;
— mauvais alignement vertical des cellules de la même colonne (voir Figure 5.1(c)). Ceci rend inefficace les méthodes qui utilisent l'alignement vertical de texte.

En plus des imperfections listées ci-dessus, certains tableaux particuliers sont difficiles à détecter comme les tableaux contenant une seule ligne ou une seule colonne.

Pour faire face à ces difficultés, nous proposons une approche de classification de lignes basée sur les Champs Aléatoires Conditionnels (CAC). La modélisation à l'aide de CAC présente deux avantages principaux permettant de surpasser les difficultés mentionnées ci-dessus. En effet, il s'agit d'un modèle graphique probabiliste qui permet à la fois :

— une modélisation probabiliste capable d'absorber la variabilité des données à étiqueter ;
— une modélisation structurelle permettant d'exploiter les informations contextuelles.

Ce chapitre est organisé comme suit. Nous allons commencer par présenter le problème d'étiquetage de séquences de données ainsi que les principaux modèles utilisés. Notre étude sera ensuite focalisée sur

l'utilisation des modèles CACs. Puis, nous décrirons le modèle de détection de tableau dans les documents de chimie. Enfin, nous présenterons quelques expérimentations et nous finirons par une conclusion.

5.2 Étiquetage de séquences de données

L'étiquetage d'une séquence de données consiste à affecter une étiquette à chaque élément de la séquence en utilisant des valeurs observées qui décrivent ces données. Il peut être vu comme un ensemble de tâches de classification indépendantes, une par élément de la séquence. Cependant, un tel étiquetage est généralement plus fiable s'il est effectué de manière à optimiser globalement l'ensemble des étiquettes de la séquence entière au lieu d'optimiser séparément l'étiquette de chaque élément.

Formellement, l'étiquetage de séquence peut être présenté comme suit : étant donné une séquence d'observations $x = \{x_1, x_2, ..., x_n\}$ où $\forall i, x_i \in O$ l'espace d'observation, et une séquence d'étiquettes $y = \{y_1, y_2, ..., y_n\}$ où $\forall j, y_j \in E$ l'espace des étiquettes, trouver la séquence d'étiquettes $\hat{y}$, telle que :

$$\hat{y} = \arg\max_{y \in E}(p(y \mid x)) \tag{5.1}$$

$\hat{y}$ est appelée la séquence d'étiquettes optimale sachant les observations x. Le problème revient à déterminer la probabilité conditionnelle $p(y \mid x)$ appelée aussi probabilité a posteriori.

De part leur capacité à gérer les incertitudes causées par la variabilité des observations et à intégrer les dépendances entre les éléments d'une séquence, les modèles graphiques probabilistes sont les plus utilisés pour résoudre ce problème. Nous distinguons deux types de modélisations permettant de déterminer $p(y \mid x)$: les modèles génératifs et les modèles discriminants.

5.2.1 Les modèles génératifs

Pour déterminer la probabilité conditionnelle $p(y \mid x)$, les modèles génératifs utilisent la règle de Bayes exprimée par la formule 5.2.

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)} \tag{5.2}$$

En utilisant cette formule, l'équation 5.1 devient :

$$\hat{y} = \arg\max_{y \in E}(\frac{p(x \mid y)p(y)}{p(x)}) \tag{5.3}$$

Dans l'équation 5.3, il s'agit de maximiser $\frac{p(x|y)p(y)}{p(x)}$ pour une séquence d'observations x donnée, c'est-à-dire, pour une probabilité $p(x)$ constante. Cette équation peut être alors simplifiée comme suit :

$$\hat{y} = \arg\max_{y \in E^n}(p(x \mid y)p(y)) \tag{5.4}$$

Ainsi, dans ce type de modèle, toutes les données (observées et cachées) sont modélisées, en estimant les distributions de probabilité $p(y)$ et $p(x \mid y)$. La première désigne la probabilité a priori des étiquettes et la deuxième modélise la façon dont les observations sont générées sachant les valeurs des étiquettes, d'où l'appellation "génératif".

Il existe plusieurs exemples de modèles génératifs dont les plus connus sont les Modèles de Markov Cachés (MMC) [Takasu2003, Seymore1999], les Champs Aléatoires de Markov (CAM) [Lemaitre2007, Nicolas2006] et les classifieurs Bayésiens Naïfs (BN).

Bien qu'ils aient été utilisés avec succès dans plusieurs applications, les modèles génératifs présentent les inconvénients suivants :

— ils transforment le problème de départ qui consiste à modéliser la distribution d'un seul processus statistique $p(y \mid x)$ en un problème plus compliqué qui consiste à modéliser les distributions de deux processus $p(y)$ et $p(x \mid y)$;
— ils nécessitent un grand nombre de données d'apprentissage afin de couvrir toutes les combinaisons possibles des valeurs des étiquettes avec les valeurs des observations. Pour remédier à ce problème, ces modèles adoptent l'hypothèse d'indépendance des observations conditionnellement aux étiquettes permettant de réduire le nombre de données nécessaires pour l'apprentissage. Cette solution est elle-même problématique puisqu'en réalité l'hypothèse d'indépendance n'est pas toujours vérifiée.

Les modèles discriminants se présentent comme une alternative aux modèles génératifs. Ils permettent de surpasser les inconvénients mentionnés ci-dessus.

5.2.2 Les modèles discriminants

A la différence des modèles génératifs, les modèles discriminants n'estiment pas la probabilité de génération des observations $p(x \mid y)$ mais ils modélisent directement la distribution de la probabilité a posteriori $p(y \mid x)$ en se basant sur l'équation 5.5

$$p(y \mid x) = p(y_1, ..., y_n \mid x) = p(y_1 \mid x)p(y_2 \mid x, y_1)....p(y_n \mid x, y_1, ...y_{n-1}) = \prod_{i=1}^{n} p(y_i \mid x, Y_1^{n-1}) \tag{5.5}$$

où Y_1^{n-1} représente tous les états de la séquence autres que y_i. En pratique, chaque probabilité locale est supposée être dépendante uniquement d'un voisinage restreint noté $y_{\mathcal{N}i}$. Ainsi, $p(y \mid x)$ est calculée à partir des probabilités conditionnelles locales $p(y_i \mid x, y_{\mathcal{N}i})$ appelées probabilités conditionnelles de transitions.

$$p(y \mid x) = \prod_{i=1}^{n} p(y_i \mid x, y_{\mathcal{N}i}) \tag{5.6}$$

Parmi les modèles discriminants, nous pouvons citer les Modèles de Markov à Entropie Maximale (MMEM) [McCallum2000] et les Champs Aléatoires Conditionnels (CAC) [Lafferty2001].

Par rapport aux CACs, les MMEMs présentent un comportement indésirable qui est connu dans la littérature sous le nom du *biais de label*. Ce comportement est dû au fait que les probabilités de transitions à partir d'un état donné sont normalisées localement, c'est-à-dire, sans tenir compte des transitions dans tout le modèle. Cette normalisation locale implique que toute la probabilité qui arrive à un état doit être répartie entre tous les états successeurs possibles. Dans le cas extrême où un état a un seul successeur, toute la masse de probabilité est transmise à ce successeur sans tenir compte de l'observation. Dans les modèles CACs, ce problème est résolu en effectuant une normalisation globale des probabilités de transitions.

Dans la section suivante, nous allons présenter le formalisme des CACs et les différentes applications de ces modèles dans l'analyse des images de documents.

5.3 Les champs aléatoires conditionnels

5.3.1 Formalisme des CACs

Définition. Soient X et Y deux variables aléatoires représentant respectivement la séquence d'observation et les étiquettes associées. Soit un graphe non orienté $G(V, E)$ tel que, à chaque variable aléatoire

Y_i de Y est associé un nœud $i \in V$. (X, Y) est un champ aléatoire conditionnel si chaque variable aléatoire Y_i vérifie la propriété de Markov dans le graphe G, c'est-à-dire :

$$p(Y_i \mid X, Y_j, i \neq j) = p(X \mid Y_i, Y_j, i \sim j) \tag{5.7}$$

avec $i \sim j$ indique que les nœuds i et j sont voisins dans le graphe G.
Autrement dit, (X, Y) est un champ aléatoire conditionnel si chaque variable aléatoire Y_i dépend uniquement de la séquence d'observation X et de ses voisins dans le graphe G. En théorie, La structure du graphe G peut être arbitraire, à condition qu'elle représente les dépendances conditionnelles dans la séquence d'étiquettes modélisées.

Théorème de Hammersley-Clifford. La distribution de probabilité p d'un champ de Markov représenté par un graphe G peut être définie comme un produit normalisé de fonctions de potentiel définies sur des sous-graphes complètement connectés appelés cliques de G.

En utilisant la structure de graphe et le théorème de Hammersley-Clifford, la probabilité d'un étiquetage y sachant la réalisation d'observations x peut être écrite sous la forme :

$$p(y \mid x) = \frac{1}{Z(x)} \prod_{c \in cl(G)} \phi_c(y_c, x) \tag{5.8}$$

avec :

$Z(x)$ un coefficient de normalisation de la distribution de probabilité conditionnelle. Ce facteur est donné par la formule suivante :

$$Z(x) = \sum_{y} \prod_{c \in cl(G)} \phi_c(y_c, x) \tag{5.9}$$

$cl(G)$ désigne l'ensemble des cliques de G.
y_c est la réalisation des variables aléatoires associées aux noeuds de la clique c.
ϕ_c désigne une fonction de potentiel sur la clique c. Cette fonction est définie dans [Lafferty2001] sous la forme :

$$\phi_c(y_c, x; w) = \exp(\sum_{k=1}^{K} w_k f_k(y_c, x)) \tag{5.10}$$

Comme on peut le voir dans l'équation 5.10, la fonction de potentiel est définie comme l'exponentielle d'une somme de K fonctions f_k appelées fonctions de caractéristiques, pondérées respectivement par des poids w_k.

— Les fonctions de caractéristiques (en anglais *feature functions*) permettent d'intégrer les connaissances du domaine de l'application, dans le modèle. Elles décrivent les occurrences des différentes combinaisons d'observation(s) et d'étiquette(s). La forme de ces fonctions dépend de l'application.
— Les poids w_k représentent les paramètres du modèle. Ils expriment l'importance des fonctions de caractéristiques correspondantes.

En utilisant l'équation 5.10 dans l'équation 5.9, la probabilité d'un étiquetage y sachant la réalisation d'observations x devient :

$$p(y \mid x) = \frac{1}{Z(x)} \prod_{c \in cl(G)} \exp(\sum_{k=1}^{K} w_k f_k(y_c, x)) = \frac{1}{Z(x)} \exp(\sum_{c \in cl(G)} \sum_{k=1}^{K} w_k f_k(y_c, x)) \tag{5.11}$$

5.3.2 Quelques applications de CACs dans l'ananlyse des images de documents

Dans [Shetty 07], un modèle CAC utilisant des informations contextuelles est utilisé pour étiqueter des segments d'écriture extraits des images de documents, en 3 types : imprimé, manuscrit ou bruit. D'abord, l'image est segmentée en un nombre de patchs disjoints en utilisant un algorithme de croissance de régions qui consiste à regrouper des composantes connexes proches (selon un seuil de distance) en un seul patch. Ensuite, un graphe d'adjacence est établi en considérant pour chaque patch un voisinage composé de ses six voisins les plus proches : un en-dessus, un en-dessous, deux à gauche et deux à droite.

Pour déterminer la probabilité d'un étiquetage sachant une séquence d'observations, deux fonctions de potentiels ont été utilisées. La première, appelée potentiel d'association, permet d'affecter un label à un patch en utilisant des descripteurs qui sont extraits sur ce dernier seul. La deuxième, appelée potentiel d'interaction, utilise des descripteurs contextuels extraits sur le patch en considérant son voisinage.

Pour définir le potentiel d'association, un vecteur de 23 caractéristiques telles que la hauteur, la largeur, la densité, le chevauchement, le nombre de composantes connexes, etc. est d'abord extrait sur le patch. Ensuite, ce vecteur est transformé en appliquant sur chaque descripteur f_k^s une fonction tangente hyperbolique $tanh$ multipliée par un facteur θ_k^s. Cette transformation est équivalente à l'application d'une couche cachée d'un réseau de neurones. Elle permet d'introduire une non-linéarité dans la frontière de décision. Enfin, le potentiel d'association est défini comme la somme pondérée des caractéristiques transformées.

Le potentiel d'interaction est définie de manière similaire au potentiel d'association mais en considérant des descripteurs contextuels (appelés aussi descripteurs de transitions). Pour chaque voisin, 4 descripteurs contextuels, qui sont la position relative, la distance, le ratio des hauteurs, le ratio des nombres de composantes connexes, sont extraits. Pour introduire la non-linéarité dans la frontière de décision, le noyau quadratique est appliqué sur ces descripteurs. La fonction de potentiel d'interaction est définie comme la somme pondérée des descripteurs contextuels transformés.

Les paramètres du modèle (les poids de pondération du potentiel d'association et ceux du potentiel d'interaction) sont estimés par la méthode de maximisation de la pseudo-vraisemblance. C'est une version approchée de la méthode de maximum de vraisemblance qui est utilisée afin de réduire la complexité du calcul des probabilités conditionnelles. Le maximum de la pseudo-vraisemblance est estimé en utilisant la méthode de gradient conjugué avec une recherche linéaire [Hestenes1952].

L'étape d'inférence consiste à utiliser le modèle estimé pour assigner à chaque patch le label correspondant. Les auteurs utilisent un algorithme d'inférence basée sur l'échantillonnage de Gibbs [Casella1992]. La méthode a été testée sur des documents extraits de l'archive de l'industrie de tabac aux états unis. Les résultats obtenus sont 95% de données correctement étiquetés. La comparaison du modèle proposé avec des classifieurs appliqués séparément sur chaque patch, tels que les réseaux de neurones et les réseaux bayésiens, a montré la supériorité du modèle CAC du fait de sa capacité à contextualiser l'étiquetage.

Nicolas et al. [Nicolas2008] ont proposé un modèle basé sur les CACs pour segmenter et étiqueter les différentes zones dans des documents manuscrits. L'image étant décomposée en des zones de taille 50×50 appelés sites, il s'agit d'affecter à chaque site l'une des six étiquettes suivantes : corps de texte, bloc de texte, numéro de page, marge, entête et pied de page. Pour ce faire, un modèle CAC est utilisé pour calculer les probabilités de tous les étiquetages possibles et de retenir le plus probable. Cette probabilité est déterminée selon la formule 5.11 en ne considérant que des cliques d'ordre 1. Les fonctions de caractéristiques sont définies sur chaque site en considérant 3 niveaux d'analyse : local, contextuel et global. Ainsi, 3 fonctions de caractéristiques f_l, f_c et f_g définies comme étant les sorties de 3 classifieurs neuronaux (PMCs) appliqués respectivement sur des descripteurs locaux, contextuels et globaux sont définies. Les descripteurs locaux, extraits sur l'image du site, sont des densités de niveaux de gris multi-résolution et la position relative du site. La sortie du classifieur appliqué sur ces descripteurs

exprime la relation entre le label associé à un site et l'observation sur ce site. Les descripteurs contextuels sont définis comme étant les probabilités d'étiquetage des 4 sites voisins du site courant. La sortie du classifieur appliqué sur ces descripteurs permet de déterminer la probabilité d'étiquetage du site courant sachant les étiquettes de ses voisins. Les descripteurs globaux sont des descripteurs statistiques extraits en fonction de la configuration globale d'étiquettes dans un voisinage plus large que celui considéré dans le niveau contextuel.

Globalement, le modèle peut être vu comme une combinaison de classifieurs qui permettent d'effectuer un étiquetage en tenant compte des caractéristiques de l'image et des configurations des étiquettes.

L'apprentissage du modèle consiste à déterminer les poids des fonctions de caractéristiques et à entraîner les 3 classifieurs. D'abord, l'apprentissage du classifieur local est effectué sur une base d'apprentissage en considérant uniquement les descripteurs locaux. Sa sortie est ensuite utilisée pour entraîner le classifieur contextuel, puis le classifieur global.

Étant donné la structure compliquée du graphe du modèle, une inférence exacte n'était pas possible. Par conséquent, les auteurs ont utilisé l'algorithme itératif ICM (Iterated Conditional Modes) permettant de déterminer une solution approximative pour l'étiquetage optimal.

Le modèle a été testé sur un ensemble de 69 images des manuscrits de Flaubert, divisé équitablement en 3 bases : d'apprentissage, de validation et de test. Un taux d'environ 94% de pixels correctement étiquetés est obtenu. Ce taux représente le meilleur résultat en comparaison avec ceux obtenus en utilisant un PMC local ou un modèle aléatoire de Markov (MRF).

Montreuil et al. utilisent un modèle CAC hiérarchique pour l'extraction de structures dans des courriers manuscrits non-contraints extraits de la base RIMES [Montreuil2010]. Le modèle global est une combinaison séquentielle de 4 modèles CACs permettant de combiner différents niveaux d'analyse de l'image. Les 3 premiers permettent de segmenter l'image respectivement en composantes connexes, mots et lignes. Des informations extraites sur les segments obtenus dans chaque niveau de segmentation sont combinées et exploitées par un quatrième CAC pour extraire la structure logique des blocs de texte. Pour la segmentation en composantes connexes, un graphe d'adjacence où les nœuds représentent les pixels de l'image est établi. Un CAC basé sur ce graphe, permet d'attribuer une même étiquette pour les pixels d'une même composante connexe. Pour ce faire, deux fonctions de potentiel définies de façon similaire à celle dans [Shetty2007], sont utilisées. Le potentiel d'association est défini en fonction de la moyenne des niveaux de gris extraits à 3 résolutions différentes. Le potentiel d'interaction est défini en fonction du ratio de la moyenne du niveau de gris du pixel courant et de celle d'un pixel proche.

La segmentation en mots est réalisée en utilisant un CAC appliqué sur un graphe d'adjacence dont les nœuds sont les composantes connexes obtenues par la segmentation précédente. Les arcs relient une composante connexe à ses 4 plus proches voisins : un au-dessus, un en-dessous, un à gauche et un à droite. Le regroupement des composantes connexes en mots est effectué par partitionnement du graphe d'adjacence, en un ensemble de sous-graphes représentant chacun un mot. De manière identique au premier niveau de segmentation, deux fonctions de potentiels ont été utilisées. La première utilise des descripteurs locaux extraits sur un mot et la deuxième utilise des descripteurs extraits sur un mot en tenant compte de ses voisins. La segmentation en lignes est effectuée de la même manière que la segmentation précédente mais en considérant des mots au lieu des composantes connexes.

Le dernier CAC permet de regrouper des lignes proches afin de créer des blocs et de leur attribuer une étiquette logique choisie parmi un ensemble de 9 étiquettes telles que : expéditeur, date/lieu, adresse, objet, etc. Le CAC est défini de manière similaire à ceux utilisés dans les étapes précédentes. Il permet d'assigner une étiquette à chaque ligne en utilisant des descripteurs locaux et contextuels. Pour créer des blocs, les lignes voisines ayant la même étiquette sont regroupées.

Testée sur 100 images de courriers, la méthode s'est montrée plus efficace que trois autres (2 MRFs

utilisant des descripteurs différents et une méthode basée sur la grammaire).

Dans [Hebert2011], un modèle CAC a été proposé pour la segmentation des documents images issus de la numérisation des pages de journaux anciens. La segmentation est réalisée par un modèle CAC horizontal dédié à l'étiquetage des pixels. Les auteurs ont introduit un processus de quantification des fonctions de caractéristiques afin d'adapter le formalisme de CAC à des observations continues. Ils utilisent des fonctions de caractéristiques binaires dont chacune rend compte de l'occurrence d'une combinaison observation(s)/étiquette(s) donnée. Deux types de fonctions ont été utilisées : locales et contextuelles. Les premières utilisent des descripteurs extraits en chaque pixel et les deuxièmes considèrent les 4 pixels horizontalement adjacents au pixel courant (2 à gauche et 2 à droite). Toutes les fonctions sont de premier ordre, c'est-à-dire une seule caractéristique est associée à une seule étiquette. Les caractéristiques utilisées sont le "run-length" horizontal et le "run-length" vertical. L'étendue des valeurs de ces deux descripteurs est très grand (le nombre de valeurs possibles du premier est égal à la largeur de l'image et celui du deuxième, est égal à la hauteur de l'image), ce qui induit un nombre très élevé des fonctions de caractéristiques. Pour pallier ce problème, les auteurs utilisent des fonctions de caractéristiques quantifiées. Tenant compte de la distribution des valeurs des descripteurs, 9 pas de quantification ont été utilisés. Ceci a permis une réduction théorique de $56,4\%$ dans le nombre des fonctions de caractéristiques. L'apprentissage des paramètres du CAC est effectué en utilisant l'algorithme L-BFGS. Les fonctions de caractéristiques dont les poids sont très faibles sont ignorées puisqu'elles ne contribuent pas significativement dans la détermination de l'étiquette du pixel. L'inférence est réalisée à l'aide d'un algorithme similaire à l'algorithme de Viterbi. L'étiquetage de la page entière est obtenu en concaténant les résultats de chaque ligne de pixels.

L'expérimentation du système est réalisée sur 23 documents extraits du quotidien de la ville de Rouen. 10 étiquettes, telles que titre, ligne de texte, séparateur horizontal, séparateur vertical, bruit, etc. sont considérées. Les résultats sont évalués au niveau pixel en se basant sur l'indice de Jaccard calculé séparément pour chaque étiquette. Les indices obtenus varient de $0,41$ pour le bruit à $0,98$ pour les lignes de texte. En plus du gain dans le temps d'exécution, la quantification des fonctions de caractéristiques a permis aussi une amélioration des performances d'étiquetage.

Dans [Awal2014], les auteurs ont proposé un modèle CAC pour la séparation manuscrit/imprimé dans des documents administratifs. Le document étant segmenté en pseudo-lignes qui elles-mêmes sont segmentées en pseudo-mots, la séparation manuscrit/imprimé est considérée comme une tâche d'étiquetage des pseudo-mots. S'appuyant sur le fait que le type d'un pseudo-mot dépend du type de ses voisins horizontaux, un modèle CAC horizontal a été défini pour modéliser une séquence horizontale de pseudo-mots (qui correspond à une pseudo-ligne). Ce modèle permet l'étiquetage des pseudo-mots en utilisant deux fonctions de caractéristiques. La première est modélisée par un SVM qui prend en entrée un vecteur de 137 caractéristiques locales extraites sur un pseudo-mot. La deuxième est modélisée par un classifieur PMC qui prend en entrée des caractéristiques contextuelles extraites en considérant pour chaque pseudo-mot ses deux voisins, celui de gauche et celui de droite. Les caractéristiques contextuelles sont des informations sur les étiquettes des voisins et 3 autres descripteurs géométriques, extraits de l'image du pseudo-mot et de chacun de ses deux voisins. Les auteurs ont utilisé un deuxième modèle CAC similaire au précédent, avec une seule différence qui consiste à considérer un voisinage plus grand. Les descripteurs contextuels sont extraits en tenant compte de toute la pseudo-ligne. Les résultats obtenus sur un ensemble de 202 documents ont montré que l'utilisation d'un voisinage global (la pseudo-ligne entière) donne des résultats meilleurs que ceux obtenus par l'utilisation d'un voisinage local (seulement deux voisins horizontaux).

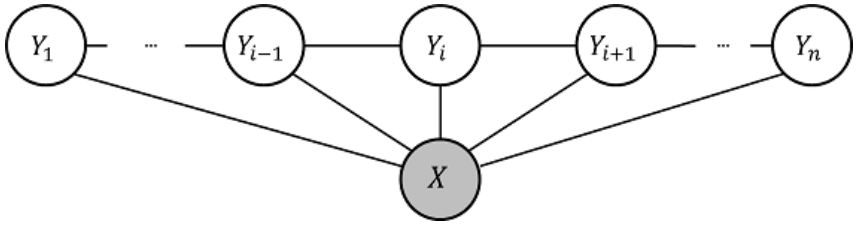


FIGURE 5.2 – Structure graphique d'un L-CAC pour l'étiquetage de la séquence de lignes dans un document.

5.4 Proposition d'un modèle CAC pour la détection de tableaux

5.4.1 Modélisation du problème

Nous considérons le problème de détection de tableaux comme une tâche d'étiquetage des lignes d'un document. Afin de tirer parti des informations contextuelles, il est intéressant d'effectuer un étiquetage global de la séquence entière au lieu d'étiqueter séparément chaque ligne. Ainsi, le document est vu comme une séquence de n lignes $L = \{l_i\}$, $i = 1..n$, modélisée par un graphe d'adjacence vertical où les nœuds correspondent aux lignes de texte et les arcs relient une ligne à ses deux voisines : une au-dessus et une en-dessous. Les lignes sont décrites par une séquence d'observations $X = \{x_1, x_2, ..., x_n\}$, où $\forall i$, $x_i \in O$. O est l'ensemble des valeurs possibles des observations, c'est-à-dire les valeurs des descripteurs extraits sur les lignes (ils seront décrits dans la section 5.4.2.3). L'étiquetage des lignes consiste à déterminer parmi tous les étiquetages possibles $Y = \{y_1, y_2, ..., y_n\}$, la meilleure séquence d'étiquettes $\hat{Y}$, telle que

$$\hat{Y} = \arg\max_{Y \in E}(p(X \mid Y)) \tag{5.12}$$

où $\forall j$, $y_j \in E$, l'ensemble des valeurs possibles que peut avoir une étiquette. Ici, nous traitons un cas binaire où l'étiquette y_j prend ses valeurs dans un ensemble d'états finis $E = \{$TextLine,TableRow$\}$. Le champ (X, Y) est supposé être un champ aléatoire conditionnel, c'est-à-dire que chaque variable aléatoire y_i dépend des observations X et de ses voisins dans le graphe d'adjacence. Nous nous proposons d'utiliser un modèle CAC linéaire (L-CAC) pour déterminer le meilleur étiquetage de la séquence de lignes. La structure graphique du modèle est représentée dans la figure 5.2

Nous nous sommes inspirés des modèles proposés dans [Montreuil2010] et [Nicolas2008] pour modéliser les fonctions de caractéristiques par des classifieurs. L'étiquetage est effectué en utilisant des cliques unaires que nous étiquetons en considérant deux niveaux d'information local (l) et contextuel (c). Ainsi, l'équation 5.11 utilisée pour la détermination de la probabilité conditionnelle globale devient :

$$p(Y/X) = \frac{1}{Z(X)} \exp(\sum_{i=1}^{n} w_l f_l(y_i, X, i) + \sum_{i=1}^{n} w_c f_c(y_i, y_{\mathcal{N}i}, X, i)) \tag{5.13}$$

• La fonction de caractéristiques locales f_l est modélisée par un classifieur neuronal (PMC) qui prend en entrée des descripteurs locaux de la ligne.

• La fonction de caractéristiques contextuelles f_c est également modélisée par un classifieur PMC qui prend en entrée des observations contextuelles sur la ligne. Ces observations sont des descripteurs extraits sur la ligne i dans son voisinage et les étiquettes $y_{\mathcal{N}i}$ des lignes voisines, où $\mathcal{N}i$ représente le voisinage de la ligne i, composé de la ligne au-dessus et la ligne en-dessous. Pour la première ligne qui n'a pas un

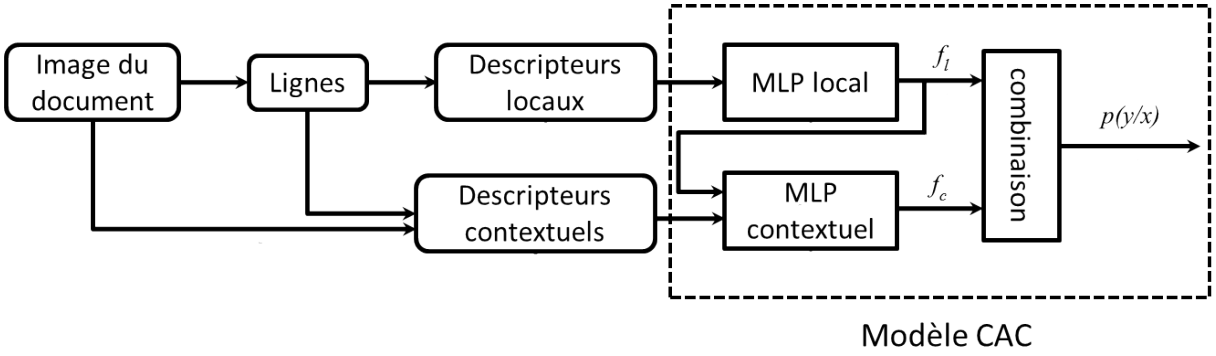


FIGURE 5.3 – Modèle CAC proposé pour l'étiquetage de lignes.

voisin au-dessus, nous l'utilisons elle-même en tant que voisin au-dessus, et de même pour la dernière ligne qui n'a pas un voisin en-dessous.

Il convient de signaler que nous utilisons la fonction sigmoïde comme une fonction de transfert dans le PMC, ce qui permet d'interpréter la sortie du perceptron comme une probabilité. En utilisant les deux fonctions f_l et f_c, la probabilité conditionnelle individuelle $p(y_i/X, y_{\mathcal{N}i})$ en une ligne i s'exprime comme suit :

$$p(y_i \mid x_i, y_{\mathcal{N}i}) = w_l f_l(y_i, x_i) + w_c f_c(y_i, y_{\mathcal{N}i}, x_i) \tag{5.14}$$

Au final, le modèle proposé est vu comme une combinaison de classifieurs (voir Figure 5.3). Nous allons maintenant décrire les descripteurs utilisés, la méthode adoptée pour l'apprentissage du modèle ainsi que la méthode de décodage.

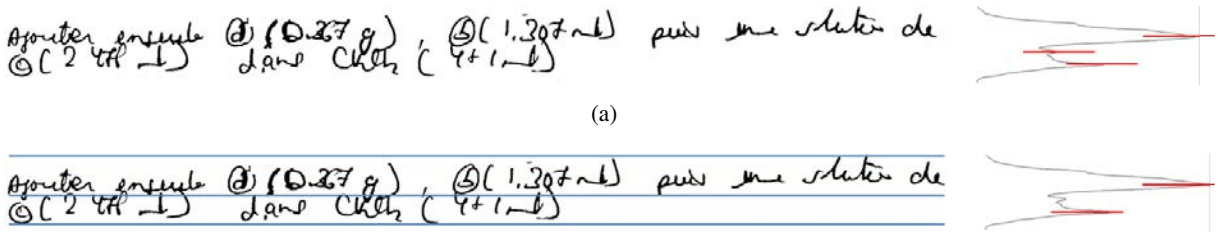
5.4.2 Extraction des descripteurs

Nous nous proposons d'extraire des descripteurs de lignes qui doivent être discriminants au sens de la détection de tableaux. Ils doivent décrire les dispositions et les dimensions des espaces, des composantes connexes et des mots dans les lignes de texte pour détecter les alignements verticaux des espaces et de texte classiquement utilisés pour la détection de tableaux. Pour cela, le document doit être tout d'abord segmenté en lignes et en mots.

5.4.2.1 Segmentation en lignes

Cette segmentation consiste à examiner les structures linéaires textuelles (extraites dans le chapitre précédent) et à affiner leur segmentation en exploitant l'information sur la hauteur des composantes connexes textuelles (une telle information n'était pas exploitable avant la séparation texte/graphique, à cause de la présence des composantes graphiques). L'idée consiste à utiliser la technique de projection pour segmenter toute structure linéaire susceptible de contenir plus qu'une ligne, c'est-à-dire, ayant une hauteur supérieure à $2 * H_m$, avec H_m la hauteur moyenne des composantes connexes, estimée dans tout le texte. L'algorithme proposé est composée des étapes suivantes :

1. Extraction des composantes connexes dans toute la zone textuelle et estimation de la hauteur moyenne H_m.
2. Construction de l'histogramme de projection horizontale.



(a)

(b)

FIGURE 5.4 – Exemple de segmentation en lignes d'une structure textuelle : (a) le texte et le profil lissé de la projection horizontale avec en rouge la position de tous les maxima, (b) exclusion des fausses détections et détermination des bords de lignes.

3. Lissage de l'histogramme en utilisant un filtre moyenneur $1D$ de longueur 5 afin d'éliminer les pics locaux.
4. Détection de pics dans le profil de projection.
5. Exclusion des fausses détections : pour deux pics consécutifs p_i et p_j tels que $|pos(p_i) - pos(p_j)| < 0.5 * H_m$, seulement le plus grand pic $p = max(p_i, p_j)$ est retenu. La condition $|pos(p_i) - pos(p_j)| < 0.5 * H_m$ signifie que la distance entre deux lignes doit être supérieure à la moitié de la hauteur moyenne des composantes connexes.
6. Détermination des bords des lignes détectées : identifier la plus grande vallée entre toute paire de pics consécutifs. Les lignes horizontales qui passent par ses vallées constituent les bords des lignes de texte. Il convient de noter ici que nous n'effectuons pas de traitements spécifiques (comme par exemple la technique de minimum cost path [F-Mota2014], le suivi de pixels [Ouwayed2010], l'utilisation de règles [Marti2001]), pour couper les composantes connexes situées dans les espaces inter-lignes car nous ne cherchons pas à extraire les lignes de manière très précise. Une délimitation grossière de la structure centrale de la ligne nous est suffisante pour extraire les descripteurs nécessaires pour la classification de lignes.

La Figure 5.4 illustre les principales étapes de segmentation en ligne d'un bloc de texte.

Un exemple de segmentation en lignes de texte d'un document, est illustré dans le Figure 5.5.

5.4.2.2 Segmentation en mots

Nous avons utilisé une méthode de regroupement qui se base sur l'hypothèse que les distances intra-mots sont inférieures à celles inter-mots. Dans chaque ligne, les composantes connexes sont extraites et celles qui se chevauchent horizontalement sont regroupées. Ensuite, les distances horizontales entre les bords des boîtes englobantes des composantes connexes adjacentes sont déterminées. Cela permet de construire un histogramme qui présente généralement deux pics caractéristiques correspondant aux deux distances les plus fréquentes d_1 et d_2. Parmi ces deux distances, la plus petite représente une approximation de l'écart entre les fragments d'un même mot. L'autre distance est une approximation de l'écart entre les mots de la ligne. La segmentation en mots consiste à regrouper les composantes connexes qui sont séparées par une distance inférieure à un seuil que nous définissons comme suit : $ds = \frac{d_1+d_2}{2}$. A la différence de la méthode présentée dans [Louloudis2009] où le seuil de décision est calculé de manière globale dans tout le document, le seuil que nous utilisons est calculé localement sur chaque ligne. Ceci permet de prendre en compte la variabilité des espaces d'une ligne à une autre, surtout que le document peut contenir des lignes de texte et des lignes de tableaux.

La Figure 5.6 illustre le résultat de la segmentation en mots d'un bloc de texte.

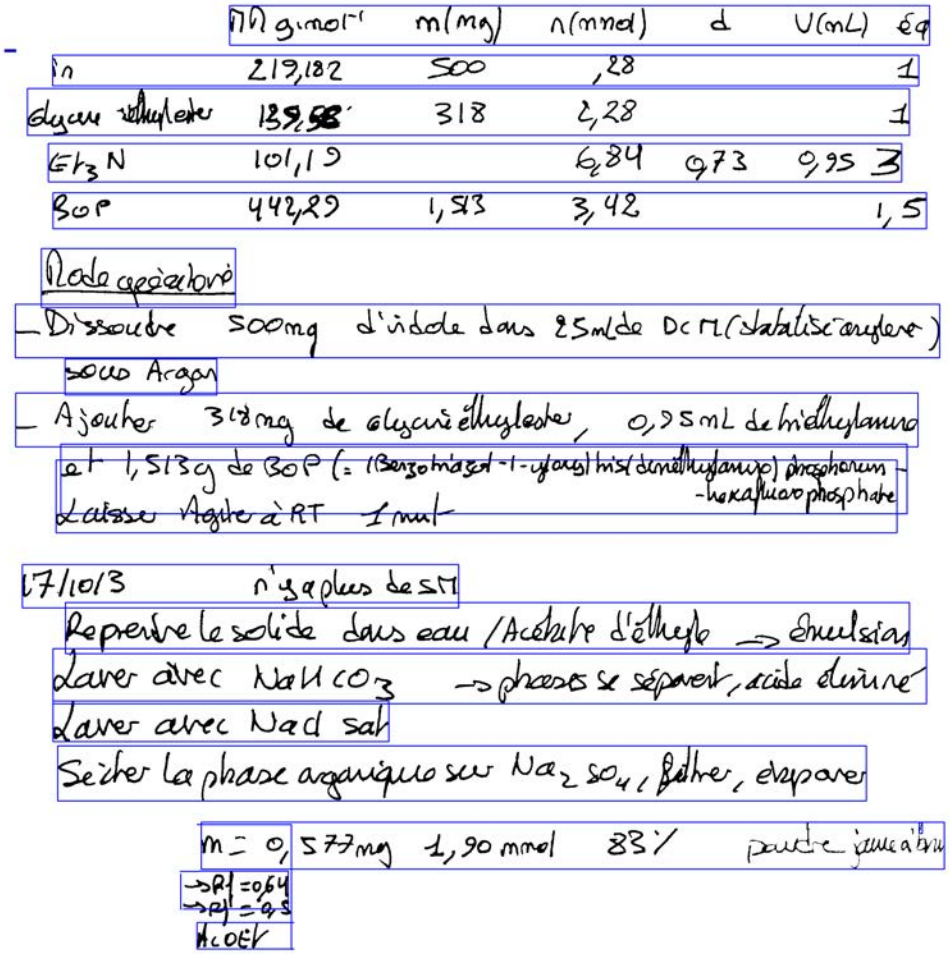


FIGURE 5.5 – Exemple de segmentation en lignes d'un document. Les rectangles bleus délimitent les lignes extraites.

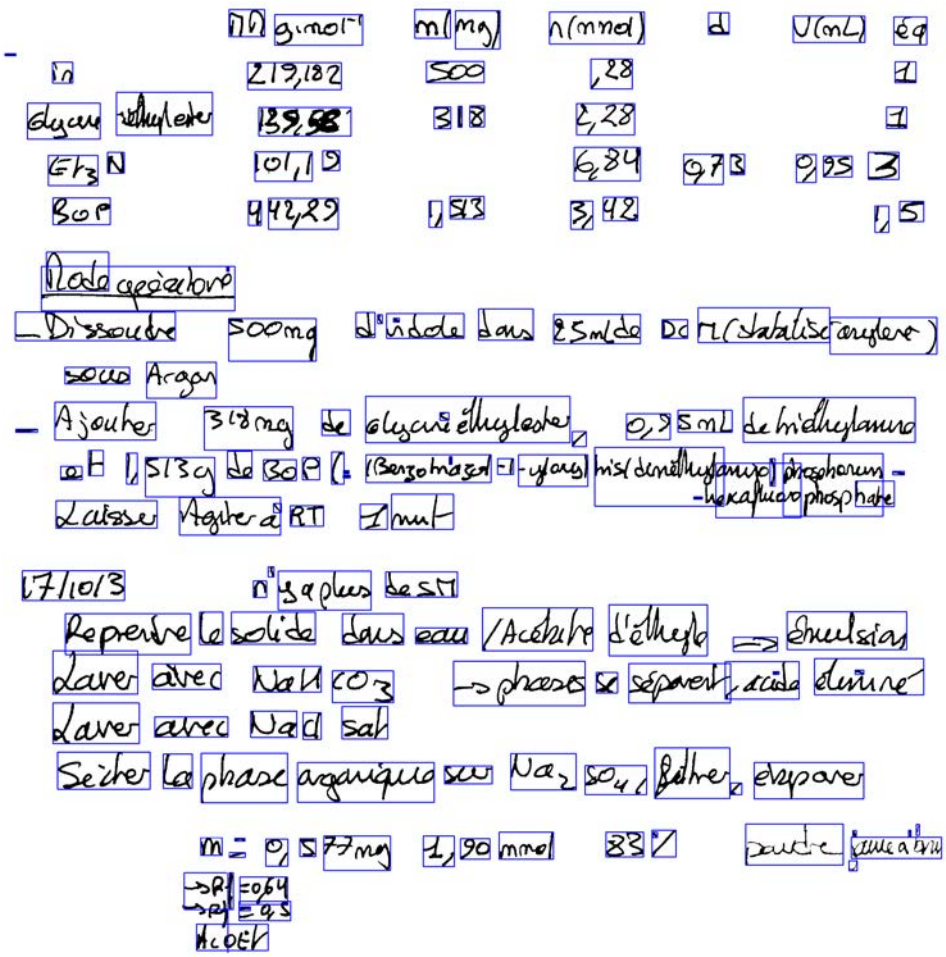


FIGURE 5.6 – Exemple de segmentation en mots d'un document. Les rectangles bleus délimitent les mots extraits.

TABLE 5.1 – liste des descripteurs locaux

Descripteurs	Commentaires
Pourcentage d'espace (PE)	La somme des longueurs des espaces divisée par la longueur de la ligne
Moyenne des longueurs des espaces (ME)	La somme des longueurs des espaces divisées par leur nombre dans une ligne
Variance des longueurs des espaces (VE)	Dans une ligne d'un tableau, il existe deux types d'espaces de longueurs différents : les espaces inter-mots et les espaces inter-cellules, ce qui induit une variance élevée. Par contre, dans une ligne de texte, il existe un seul type d'espace, les espaces inter-mots, qui sont plus ou moins réguliers, d'où une variance faible.
Nombre de mots (NM)	Le nombre de mots dans une ligne
Longueur moyenne des mots (MM)	La somme des longueurs de mots dans une ligne divisée par leur nombre
Moyenne des longueurs de composantes connexes (MC)	La somme des longueurs des composantes connexes divisée par leur nombre

5.4.2.3 Extraction de descripteurs de lignes

Les descripteurs locaux. Ils sont utilisés par le classifieur afin d'associer une étiquette à chaque ligne en utilisant les caractéristiques de cette ligne seule. Six descripteurs sont extraits sur chaque ligne comme décrit dans le tableau 5.1. Les 3 premiers caractérisent les espaces contenus dans une ligne et les 3 autres caractérisent le texte. Il convient de signaler que seuls les espaces inter-mots sont considérés car les espaces intra-mots n'apportent pas d'information vu qu'ils sont invariants entre une ligne de texte et une ligne de tableau.

Les descripteurs contextuels. Ils prennent en compte des informations dans le voisinage de la ligne. Ils sont de deux types :

1. des observations sur l'image d'une ligne l_i et chacune des deux lignes adjacentes l_{i-1} et l_{i+1} ;
2. des observations sur les étiquettes des lignes adjacentes l_{i-1} et l_{i+1}.

Pour les observations sur l'image, nous avons opté pour des descripteurs qui mesurent la similarité de la ligne courante avec chacune des deux lignes voisines. L'extraction de tels descripteurs est motivée par le fait suivant : par rapport aux lignes de texte, les lignes d'un tableau présentent une grande similarité dans les positions horizontales du texte et des espaces. Les descripteurs que nous avons extraits sont les suivants :

— **Similarité basée sur les espaces (SE)** : ce descripteur mesure le chevauchement horizontal des espaces à l'intérieur de deux lignes consécutives. Les espaces entre les boîtes englobantes de mots sont considérés. Le taux de chevauchement horizontal est défini de la manière suivante (voir Figure 5.7) : soient (x_{11}, x_{12}) et (x_{21}, x_{22}) les étendues horizontales respectives des espaces S_1 et S_2 situés respectivement dans deux lignes adjacentes l_1 and l_2. Sans perte de généralité, nous supposons que $x_{11} < x_{21}$ et $x_{12} < x_{22}$. Le taux de chevauchement horizontal τ_o entre S_1 et S_2 est défini par :

$$\tau_o(S_1, S_2) = \frac{x_{12} - x_{21}}{\min(x_{12} - x_{11}, x_{22} - x_{21})} \tag{5.15}$$

La similarité des espaces entre l_1 et l_2 est calculée comme suit. Soient $\{S_{1i}, i \leq N_1\}$ et $\{S_{2j}, j \leq$

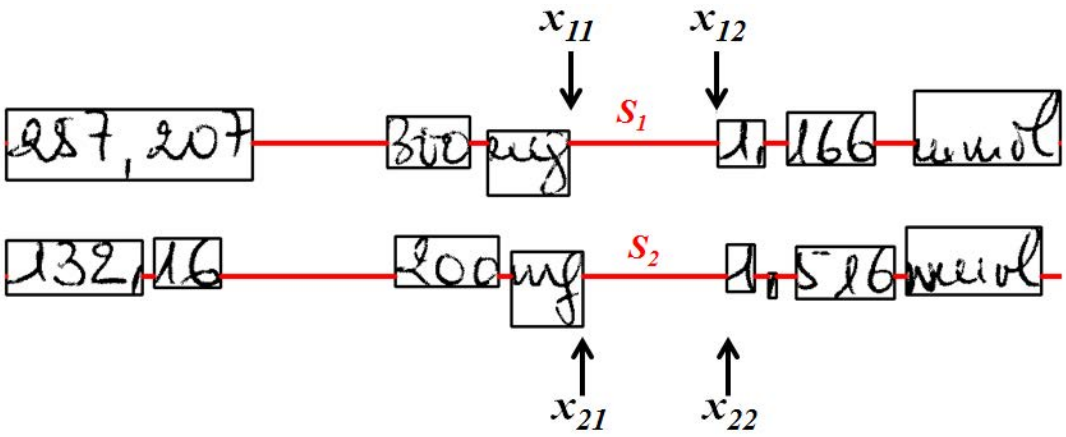


FIGURE 5.7 – Similarité d'espaces entre deux lignes consécutives.

$N_2\}$ l'ensemble des espaces contenus respectivement dans l_1 et l_2.

$$SE(l_1, l_2) = \frac{\sum_{i \leq N_1, j \leq N_2} \tau_o(S_{1i}, S_{2j})}{\min(N_1, N_2)} \tag{5.16}$$

où N_1 et N_2 représentent respectivement le nombre des espaces dans l_1 et l_2.

— **Similarité basée sur les boîtes englobantes des mots (SM)** : ce descripteur décrit l'alignement vertical des boîtes englobantes des mots dans deux lignes adjacentes. Deux mots sont considérées verticalement alignés si leurs boîtes englobantes se chevauchent horizontalement de façon significative. Le taux de chevauchement horizontal entre les boîtes englobantes de deux mots est défini de la même manière que dans l'équation 5.15 mais en considérant les positions et les étendues horizontales des boîtes englobantes au lieu des espaces. La similarité de deux lignes adjacentes est ensuite calculée en considérant tous les chevauchements des boîtes englobantes de leurs mots de manière similaire à celle définie dans l'équation 5.16.

— **Similarité basée sur le nombre de mots (SNM)** : c'est un descripteur binaire qui prend la valeur 1 si deux lignes adjacentes ont le même nombre de mots et 0 sinon.

Pour les observations sur les étiquettes, nous considérons pour une ligne l_i, les probabilités conditionnelles locales $p(y_{l_{i-1}} = y_k)$ et $p(y_{l_{i+1}} = y_k)$ avec $y_k \in \{\text{TextLine,TableRow}\}$. Ainsi, 4 probabilités conditionnelles sont utilisées comme observations contextuelles sur les étiquettes. Elles sont déterminées par le classificateur local.

La figure 5.8 illustre l'extraction des descripteurs locaux et contextuels sur une image de document.

5.4.3 Apprentissage du modèle

L'apprentissage du modèle CAC consiste à entraîner les deux PMC et à estimer les valeurs optimales des paramètres w_l et w_c. Pour ce faire, nous utilisons un ensemble de documents dont les lignes ont été extraites et étiquetées en TextLine ou TableRow.

Apprentissage des PMC[9]**.** D'abord, le classifieur local est entraîné avec les descripteurs locaux en utilisant la technique de rétropropagation du gradient (backpropagation en anglais). Les deux principaux

9. Utilisation de la librairie C FANN [Nissen2003]

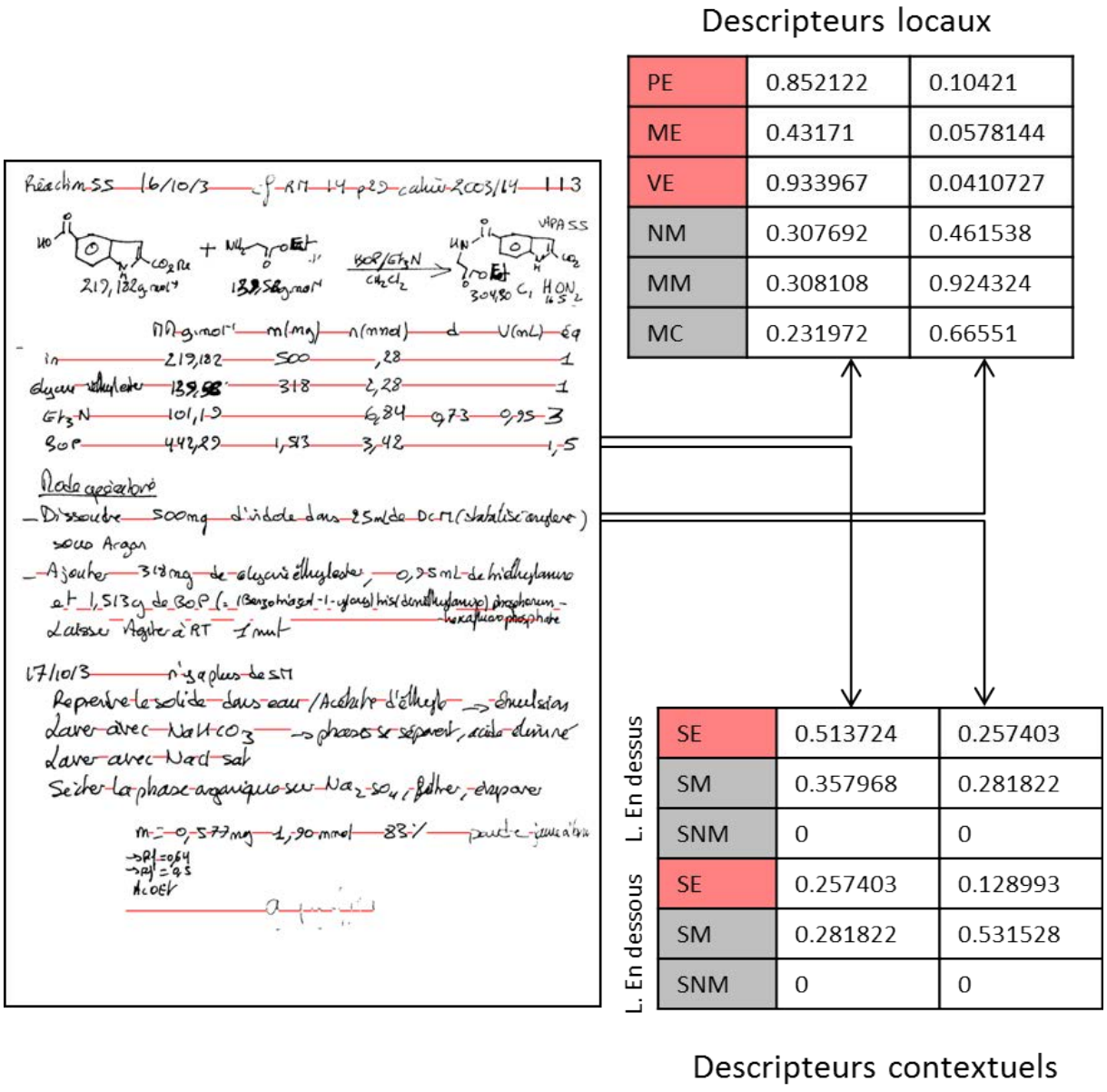


FIGURE 5.8 – Exemples de valeurs de descripteurs locaux et contextuels extraits sur une image de document.

paramètres du PMC, à savoir le nombre de couches cachées et le nombre de neurones par couche sont optimisés expérimentalement. Différentes combinaisons de valeurs associées à ces deux paramètres ont été utilisées et à chaque fois, le classifieur est évalué par validation croisée. Ensuite, le PMC local (déjà appris) est utilisé pour estimer les probabilités $p(y_{l_i} = y_k)$ pour toutes les lignes l_i de la base d'apprentissage et pour tout $y_k \in \{\text{TextLine,TableRow}\}$. Ces probabilités sont utilisées avec les descripteurs contextuels extraits sur l'image des lignes correspondantes pour entraîner le classifieur contextuel. Les paramètres de ce classifieur sont optimisés de la même manière que ceux du PMC local.

Détermination des paramètres w_l et w_c. Il s'agit de déterminer les valeurs optimales des paramètres $w = \{w_l, w_c\}$ qui minimisent globalement l'erreur d'étiquetage sur un ensemble $\{(x^{(i)}, y^{(i)})\}_{i=1..nbL}$ de nbL lignes étiquetées. L'erreur d'étiquetage globale $R(w)$ est égale à l'opposé de la log-vraisemblance conditionnelle (cette erreur est appelée aussi *perte-log*) :

$$R(w) = \sum_{i=1}^{nbL} -\log p(y^{(i)} \mid x^{(i)}; w) \tag{5.17}$$

L'estimation des paramètres optimaux revient à déterminer $\hat{w}$ tel que :

$$\hat{w} = \arg\min_w(R(w)) = \arg\min_w(\sum_{i=1}^{nbL} -\log p(y^{(i)} \mid x^{(i)}; w)) \tag{5.18}$$

Il s'agit d'un problème de minimisation d'une fonction convexe (la fonction $-log$) qui n'a qu'un seul minimum global. Dans la littérature, il existe plusieurs algorithmes permettant de résoudre ce problème, dont les plus utilisés sont l'algorithme L-BFGS (Limited-memory Broyden Fletcher Goldfarb Shanno) [Liu1989] et la méthode de descente du gradient [Nocedal2006].

Nous avons utilisé l'algorithme de descente de gradient avec recherche linéaire. C'est un algorithme itératif qui consiste à évaluer la fonction R avec des paramètres w différents à chaque étape. La variation des paramètres est effectuée à l'aide d'une descente qui produit une séquence $w^{(k)}$ telle que $w^{(k+1)} = w^{(k)} + \epsilon^{(k)} d^{(k)}$ où $d^{(k)}$ est une direction de descente, $\epsilon^{(k)} > 0$ est le pas, de sorte à avoir $R(w^{(k+1)}) < R(w^{(k)})$ et dès que cette inégalité n'est pas vérifiée, $w^{(k)}$ est optimal. Le pas $\epsilon^{(k)}$ est déterminé en employant la technique de recherche linéaire.

La partie coûteuse dans ce calcul est la détermination du coefficient de normalisation $Z(x)$ qui nécessite le calcul des probabilités non normalisées $(p(y \mid x)Z(x))$ pour tous les étiquetages possibles de x. Dans notre cas où le modèle est un CAC 1D, ces calculs sont effectués en employant une version adaptée aux modèles CACs de l'algorithme forward-backword, classiquement utilisés dans les MMCs [Jousse2006].

5.4.4 Décodage

Le décodage consiste à déterminer, en utilisant le modèle CAC appris, la configuration optimale $\hat{y}$ du champ d'étiquettes Y sachant une réalisation x du champ d'observation X. Ceci peut être considéré comme un problème d'optimisation au sens d'un critère donné. Le critère le plus utilisé est celui du Maximum A posteriori (MAP). En utilisant ce critère, il s'agit de déterminer la configuration $\hat{y}$ qui maximise la probabilité a posteriori $p(y \mid x)$:

$$\hat{y} = \arg\max_y p(y \mid x) = \arg\max_y(\sum_{i=1}^{n} w_l f_l(y_i, X, i) + \sum_{i=1}^{n} w_c f_c(y_i, y_{\mathcal{N}i}, X, i)) \tag{5.19}$$

Le calcul des probabilités de tous les étiquetages possibles est une opération impossible en pratique. Le graphe de notre modèle étant 1D, nous avons utilisé l'algorithme de Viterbi de façon similaire à son utilisation dans les MMCs.

La procédure de décodage est effectuée comme suit :

1. La séquence de lignes est étiquetée en utilisant seulement les descripteurs locaux. Des probabilités locales d'étiquetage sont obtenues.
2. Les probabilités locales auxquelles sont ajoutés les descripteurs contextuels extraits à partir de l'image, sont utilisées par le classifieur contextuel pour effectuer un étiquetage contextuel de la séquence de lignes. Des probabilités contextuelles d'étiquetage sont obtenues.
3. Les probabilités locales et celles contextuelles sont combinées (en tenant compte de leurs pondération w_l et w_c) et la séquence optimale des étiquettes $\hat{y}_i$ est déterminée par l'algorithme de Viterbi. Les probabilités globales d'étiquetage obtenues sont mémorisées.
4. Les étapes 2 et 3 sont répétées en utilisant les probabilités globales (de l'itération précédente $i-1$) comme des probabilités locales, jusqu'à ce que la séquence optimale d'étiquettes soit stable, c'est-à-dire $\hat{y}_i = \hat{y}_{i-1}$ ou un nombre d'itérations maximal soit atteint.

5.5 Expérimentation et résultats

La base ChemicalText composée des images contenant les régions textuelles (texte régulier et tableau) est utilisée pour l'évaluation de notre système. Nous évaluons les tâches suivantes :

— les segmentations en lignes et en mots puisqu'elles représentent des tâches qui influencent les résultats de la détection de tableaux ;
— la détection de tableau.

5.5.1 Évaluation de la segmentation

Ne nécessitant aucun apprentissage, cette étape est évaluée sur la totalité de la base ChemicalText. Nous avons utilisé les métriques d'évaluation qui ont été adoptées dans les dernières compétitions portant sur la segmentation en mots et en lignes, qui ont été organisées dans les conférences ICDAR2009 [Gatos2009], ICDAR2013 [Stamatopoulos2013] et ICFHR2010 [Gatos2010].

Ces mesures sont basées sur le nombre de correspondances entre les éléments (mots ou lignes) détectés par le système et ceux de la vérité terrain. Pour ce faire, une matrice *MatchScore* dont les valeurs sont calculées en se basant sur le taux d'intersection entre un élément de la vérité terrain d'une part, et un élément détecté par le système d'autre part.

$$MatchScore(i,j) = \frac{T(G_j \cap R_i)}{T(G_j \cup R_i)} \tag{5.20}$$

où G_j et R_i désignent respectivement un élément de la vérité terrain et un élément détectée par le système. T est une fonction qui compte le nombre de pixels noirs de la zone donnée en paramètre. Nous avons modifié cette fonction de façon qu'elle mesure l'aire de la zone donnée en paramètre pour l'adapter à notre vérité terrain et notre système où les éléments sont définis par leurs boîtes englobantes.

Une paire d'éléments est considérée comme une correspondance un-à-un, si la valeur correspondante dans la matrice $matchScore$ est supérieure ou égale à un seuil d'acceptation S_a défini par l'évaluateur.

Soient N le nombre d'éléments dans la vérité terrain et M le nombre d'éléments obtenus par le système et soit $o2o$ (comme *one-to-one*) le nombre de correspondances entre des paires d'éléments. Le

TABLE 5.2 – Evaluation des résultats de la segmentation.

	N	M	$o2o$	$Rappel(\%)$	$Précision(\%)$	$FM(\%)$	$SM(\%)$
Lignes	8691	8769	8190	94.23	93.40	93.81	83.84
Mots	59970	56966	43197	72.03	75.83	73.88	

rappel et la précision de la segmentation sont définis comme suit :

$$Rappel = \frac{o2o}{N} \tag{5.21a}$$

$$Précision = \frac{o2o}{M} \tag{5.21b}$$

La F-mesure FM est ensuite définie par la combinaison des deux métriques DR et RA comme suit

$$FM = \frac{2 * Rappel * Précision}{Rappel + Précision} \tag{5.22}$$

Une mesure de performance globale, notée SM, est définie comme étant la moyenne des F-mesures obtenues pour la segmentation en lignes et la segmentation en mots.

Dans toutes les compétitions, les évaluateurs ont utilisé les valeurs 0.95 et 0.9 comme seuils d'acceptation respectivement pour les lignes et les mots. Nous utilisons ces mêmes valeurs dans notre évaluation.

Les résultats obtenus sont reportés dans le tableau 5.2.

Il convient de noter que les résultats de la segmentation varient d'un document à un autre selon la qualité de l'écriture (voir Figure 5.9(a) et 5.9(b)). Pour la segmentation en lignes, les principales erreurs sont des erreurs de fusion dues au grand chevauchement entre les lignes adjacentes. Quant à la segmentation en mots, les erreurs sont dues à des espacements non uniformes entre les mots adjacents. Il en résulte des cas où des fragments de deux mots consécutifs sont fusionnés et des cas où des fragments d'un même mot sont répartis en deux ou plus (voir Figure 5.9(f)).

5.5.2 Évaluation de la détection de tableau

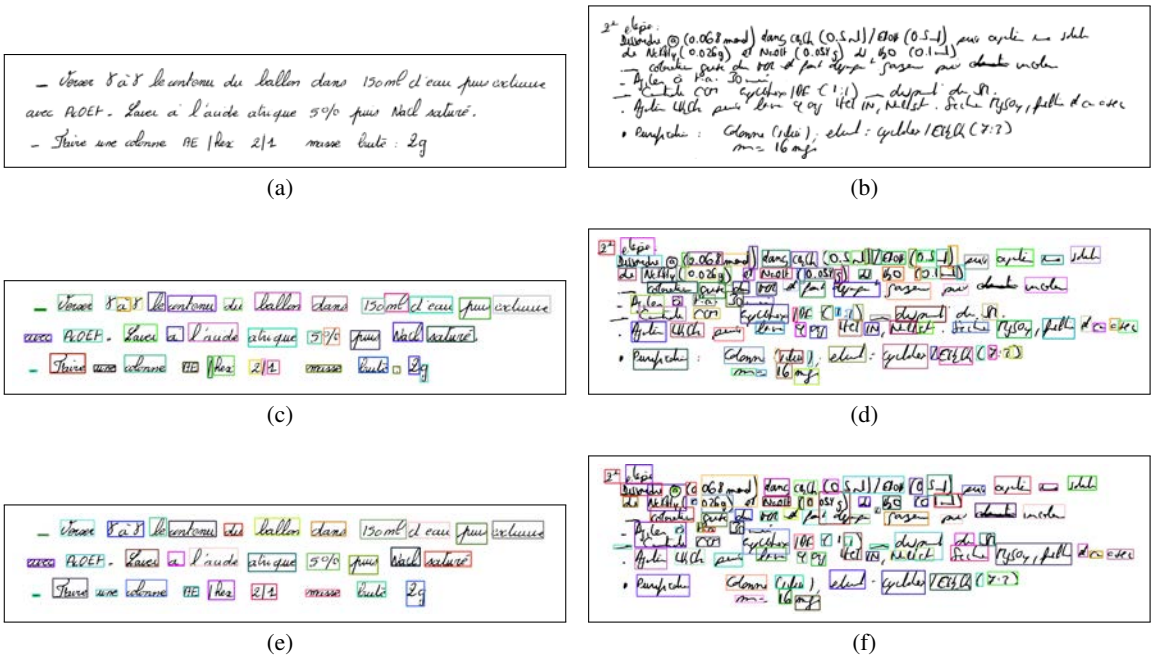
A partir de la base ChemicalText, nous avons utilisé 280 documents pour l'apprentissage et 220 documents pour les tests. L'évaluation de la détection de tableaux est effectuée :

— au niveau ligne ;
— au niveau tableau.

Évaluation au niveau ligne. Nous avons adopté les mêmes métriques de performance utilisées pour l'évaluation de l'étape de segmentation, mais en considérant séparément l'ensemble des lignes de texte et ceux de tableaux avec un seuil d'acceptation égal à 0.9. Nous avons évalué les résultats obtenus en utilisant :

— le classifieur local seul ($w_c = 0$) ;
— le classifieur contextuel seul ($w_l = 0$) ;
— le modèle CAC combinant les deux classifieurs.

Les performances obtenues sont présentées dans le tableau 5.3. Elles montrent la supériorité du modèle CAC par rapport à chacun des deux classifieurs quand ils sont appliqués individuellement sur chaque ligne. Ceci s'explique par le grand pouvoir du CAC à prendre en compte les relations entre les lignes et à intégrer des informations contextuelles. Notamment, les informations sur la similarité entre les lignes



(a) (b) (c) (d) (e) (f)

FIGURE 5.9 – Extraits des documents sur lesquels nous avons obtenu la meilleure performance (à gauche) et la plus faible performance (à droite). (a) et (b) représentent les extraits d'images, (c) et (d) la vérité terrain, (e) et (f) le résultat obtenu.

TABLE 5.3 – Performance de l'étiquetage de lignes.

	Classifieur local		Classifieur contextuel		CAC combinant les 2 classifieurs	
	Rappel(%)	Précision(%)	Rappel(%)	Précision(%)	Rappel(%)	Précision(%)
TextLine	90, 27	87, 81	84, 57	89, 62	90, 57	91, 71
TableRow	76, 50	84, 22	83, 50	71, 15	86, 9	85, 79
Moy. pondérée	86, 51	86, 83	84, 28	84, 58	89, 57	90, 09

adjacentes et leurs étiquettes qui sont introduites par les potentiels contextuels, permettent une bonne discrimination.

Il convient de signaler que les résultats d'étiquetage des lignes rendent compte du cumul des erreurs qui se produisent au niveau de l'étape de segmentation et de l'étape de classification. En observant ces résultats, nous constatons que les principales erreurs de classification se produisent sur :

— des lignes de texte composées d'un seul mot, qui sont situées au-dessus ou en-dessous d'un tableau. De telles lignes sont souvent classées comme TableRow parce que leurs caractéristiques locales n'apportent pas suffisamment d'informations pour être bien classées, d'une part ; d'autre part, lorsque le mot est aligné avec la première colonne du tableau, ce qui est souvent le cas, les caractéristiques contextuelles de ces lignes sont similaires à celles d'un tableau ;

— la première ligne d'un document. Elle représente souvent un entête sous forme de champs dispersés horizontalement (numéro de manipulation, date, etc.), d'où, elle a des caractéristiques locales similaires à celles d'une ligne de tableau. De plus, ses caractéristiques contextuelles ne sont pas très pertinentes du fait qu'elle n'a pas un voisin au-dessus ;

— les lignes de tableaux dont la disposition des cellules est très altérée. Elles constituent des cas difficiles même pour un opérateur humain s'il ne se base pas sur une interprétation du contenu.

Évaluation au niveau tableau. Chaque ensemble de lignes adjacentes ayant été étiquetées comme TableRow sont regroupées pour former un tableau. L'ensemble des tableaux $TS = \{TS_i\}_{i\in[1,M]}$ ainsi obtenus est évalué par rapport à l'ensemble des tableaux de la vérité terrain $TV = \{TV_j\}_{j\in[1,N]}$ en calculant la matrice *MatchScore* telle que définie dans l'équation 5.20. A partir de cette matrice, nous déterminons :

— le nombre de tableaux correctement détectés (nb_c). C'est le nombre de tableaux $TS_i \in TS$, tel qu'il existe un seul $TV_j \in TV$, avec $MatchScore(i,j) \geq 0.9$;

— le nombre de tableaux partiellement détectées (nb_p). C'est le nombre de tableaux $TS_i \in TS$, tel qu'il existe un seul $TV_j \in TV$, avec $0.2 \leq MatchScore(i,j) \leq 0.9$;

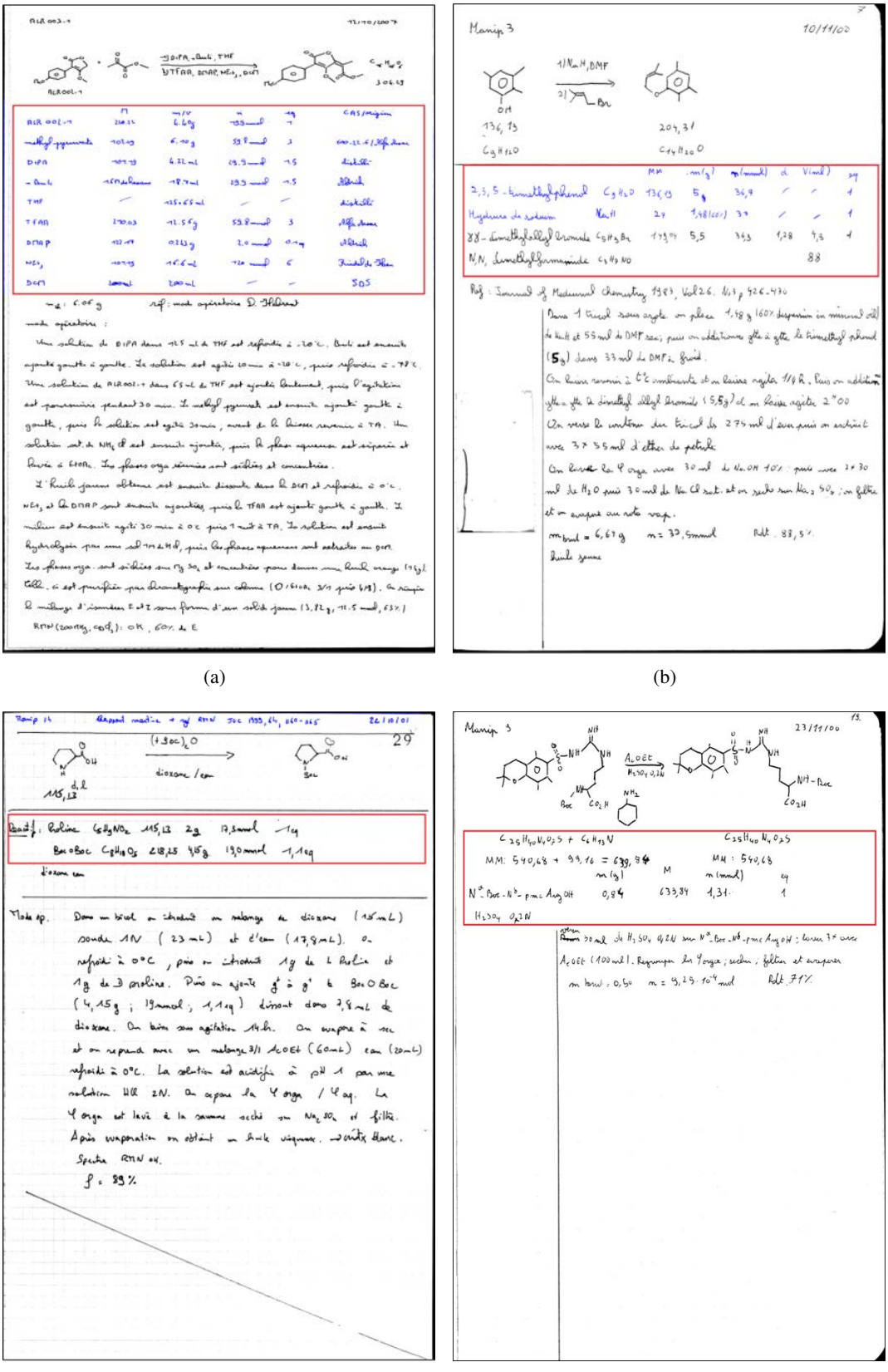
— le nombre de fausses détections (nb_f). C'est le nombre de tableaux $TS_i \in TS$ n'ayant pas de chevauchement significatif avec aucun tableau $TV_j \in TV$, c'est-à-dire $\forall j, MatchScore(i,j) < 0.2$;

— le nombre de tableaux manqués (nb_m). C'est le nombre de tableaux $TV_j \in TV$ n'ayant pas de chevauchement significatif avec aucun tableau $TS_i \in TS$, c'est-à-dire $\forall i, MatchScore(i,j) < 0.2$;

Un exemple de chacun de ces cas de figure est illustré dans la figure 5.10. Sur l'ensemble des 220 documents contenant 220 tableaux, le système permet de détecter 251 tableaux répartis suivant les cas décrits ci-dessus, de la manière suivante : 136 tableaux correctement détectés, 73 partiellement détectés et 42 fausses détections, tout en manquant 7 tableaux.

5.6 Conclusion

Nous avons présenté dans ce chapitre une approche basée sur un modèle CAC pour détecter les tableaux dans les documents manuscrits de chimie. L'idée consiste à considérer le problème de détection de tableaux comme un problème d'étiquetage de lignes selon qu'elles appartiennent à un tableau ou à un bloc de texte. L'utilisation du modèle CAC est motivée par sa capacité à modéliser la variabilité des lignes et à intégrer des caractéristiques locales et contextuelles pour déterminer la séquence d'étiquettes optimale correspondante à la séquence de lignes d'un document. Le modèle proposé est réalisé en combinant deux classifieurs discriminants de type PMC. Il est entraîné en utilisant la méthode de descente du gradient avec recherche linéaire. L'étape de décodage est réalisée en utilisant l'algorithme de Vi-



(a) (b) (c) (d)

FIGURE 5.10 – Différents cas de détection de tableaux : le résultat du système est en bleu et la vérité terrain est encadrée en rouge. (a) Détection correcte, (b) détection partielle, (c) fausse détection + tableau manqué et (d) tableau manqué.

terbi. L'expérimentation du modèle sur une base de 220 documents a montré que les résultats obtenus dépassent ceux des classifieurs classiques appliqués individuellement sur chaque ligne.

Chapitre 6

Extraction de la structure de tableaux

Sommaire

6.1 Introduction 95
6.2 Système proposé 95
6.3 Niveau structurel 98
6.3.1 Détection des lignes graphiques 98
6.3.2 Analyse de la grille graphique 99
6.3.3 Projection des boîtes englobantes 100
6.4 Niveau contenu 100
6.4.1 Extraction et filtrage des CCs 101
6.4.2 Discrimination chiffre/non chiffre 102
6.4.3 Correction de la segmentation par l'analyse du contenu 110
6.5 Expérimentation et résultats 110
6.6 Conclusion 113

6.1 Introduction

Dans ce chapitre, nous présentons une méthode automatique de segmentation de tableaux en cellules. Elle prend en entrée des tableaux qui sont extraits par le module précédent et cherche à déterminer les limites entre les cellules. Ces limites peuvent être explicites sous forme de lignes graphiques ou de larges espaces verticalement alignés, ou implicites pouvant être identifiées à partir des informations syntaxiques relatives au contenu.

6.2 Système proposé

Pour une meilleure compréhension du système, nous présentons quelques caractéristiques des tableaux traités.

— Du point de vue de la **structure**, certains tableaux sont structurés à l'aide de lignes graphiques qui séparent la totalité ou un sous-ensemble des cellules. D'autres ne contiennent aucune ligne graphique mais ils présentent une disposition particulière de leurs cellules (alignement, espacement, etc.).

— Du point de vue du **contenu**, nous distinguons trois types de régions dans un tableau : la ligne entête, la colonne entête et le corps du tableau. L'une des entêtes contient des libellés de produits

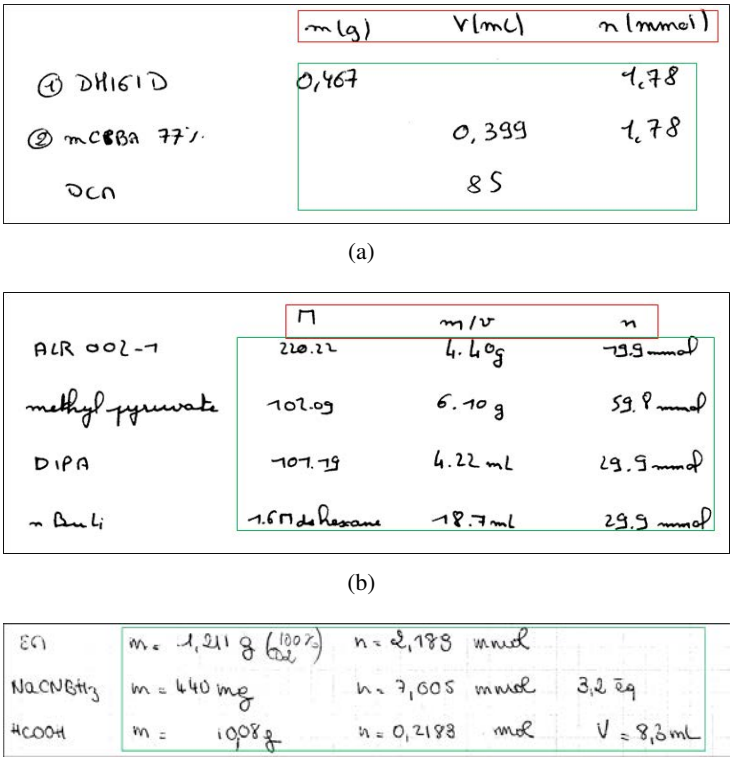


FIGURE 6.1 – Syntaxe des cellules de données en fonction de l'entête contenant les grandeurs, (a) l'entête est sous la forme $G(U)$, les cellules de données contiennent uniquement des valeurs numériques, (b) l'entête est sous la forme G, l'unité est reportée dans les cellules de données, d'où la syntaxe NU, (c) le tableau est sans entête, les cellules de données ont une syntaxe $G = NU$.

chimiques et l'autre contient des grandeurs qui peuvent être accompagnées d'unités. Selon la présence ou non, et la forme de l'entête contenant les grandeurs, le contenu d'une cellule de donnée (située dans le corps du tableau) est une chaîne numérique ou alphanumérique (voir les tableaux dans la figure 6.1). La syntaxe générale de ces cellules est de la forme :

$$[G =]N[U] \tag{6.1}$$

où les termes entre crochets sont optionnels, G désigne une grandeur telle que masse (m), volume (v), nombre de môles (n), densité (d), etc. N est une chaîne numérique et U est l'unité correspondante à la grandeur G, telle que gramme (g), millilitre (ml), môle (mol), etc.

Pour s'affranchir de la variabilité et des imperfections des tableaux dues à leur nature manuscrite, nous proposons une méthode de segmentation en cellules qui fait coopérer deux niveaux d'analyse.

— **Structurel**, les lignes graphiques et l'alignement vertical des espaces dans les lignes (de texte) sont analysées afin de déterminer les limites entre les cellules d'un tableau.
— **Syntaxique**, la syntaxe des cellules est étudiée en utilisant les connaissances a priori sur le contenu. Les limites entre les cellules sont déterminées en localisant des modèles (patterns) syntaxiques récurrents le long d'une ligne.

Le schéma du système proposé est illustré dans la figure 6.2.

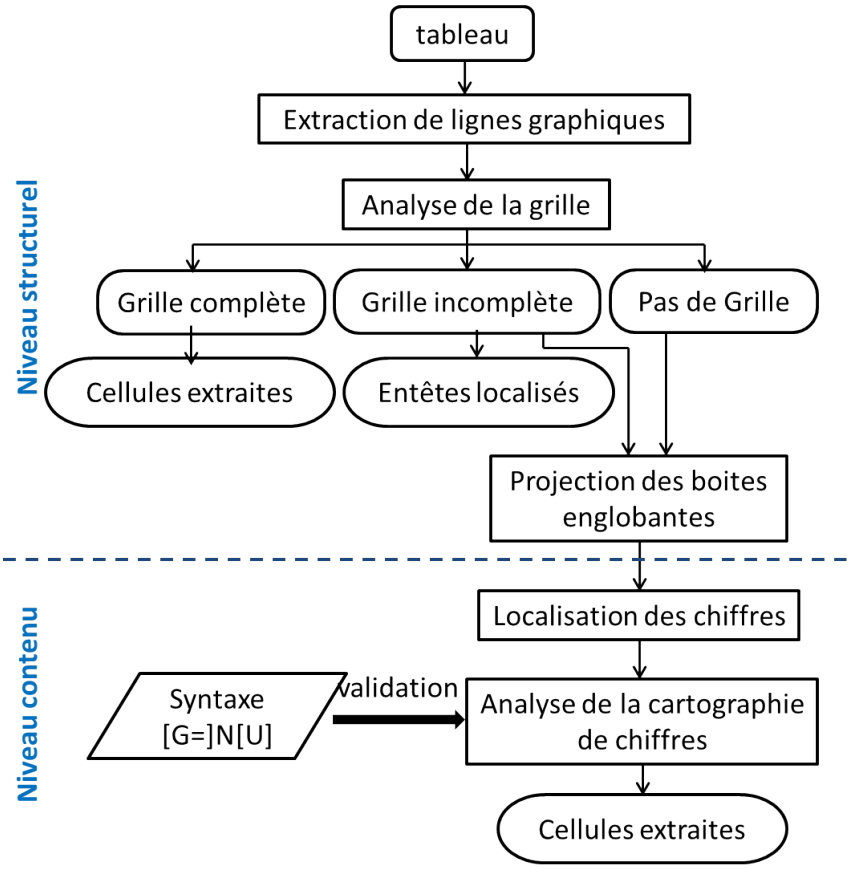


FIGURE 6.2 – Schéma global du système proposé pour la segmentation de tableaux.

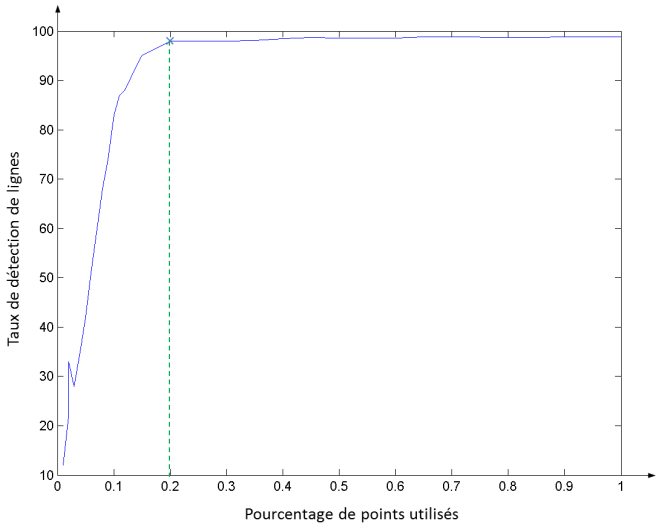


FIGURE 6.3 – Le taux moyen de détection de lignes en fonction de la fraction de points utilisée pour la transformée de Hough.

6.3 Niveau structurel

6.3.1 Détection des lignes graphiques

Nous avons utilisé une variante probabiliste de la transformée de Hough [Kiryati1991] pour extraire les lignes graphiques dans l'image d'un tableau. Tout comme la transformée de Hough standard, une correspondance des pixels de l'image avec l'espace de paramètre (ρ, θ), est effectuée. La différence est qu'au lieu d'utiliser le total des N pixels de l'image comme points votants, nous utilisons seulement un sous-ensemble de n pixels, avec $n \leq N$. Ceci permet une réduction considérable du temps d'exécution.

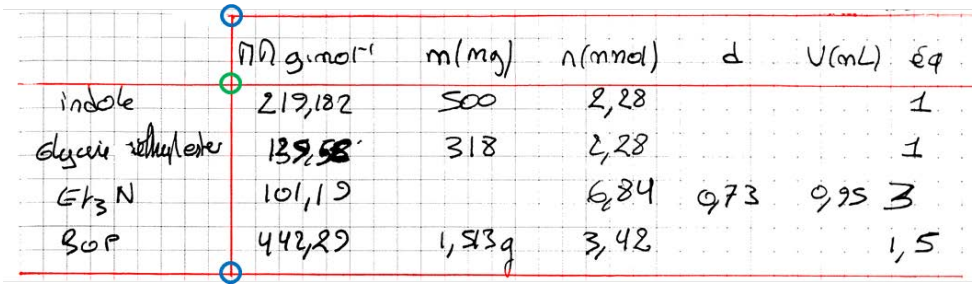
La valeur de n dépend de l'application et est souvent représentée comme une fraction du nombre total de pixels dans une image. Nous avons essayé différentes valeurs pour choisir celle qui maximise le taux de détection de lignes sur un ensemble d'images. La courbe illustrant ce taux en fonction du nombre de points utilisés dans la transformée de Hough est donnée par la figure 6.3.

Nous remarquons qu'en utilisant uniquement 20% du nombre total de pixels noirs de l'image, nous pouvons extraire les lignes graphiques avec une bonne précision et que l'utilisation de points supplémentaires n'améliore que très légèrement le taux de détection de lignes. Nous avons utilisé cette valeur comme le nombre de points votants dans la transformée de Hough probabiliste.

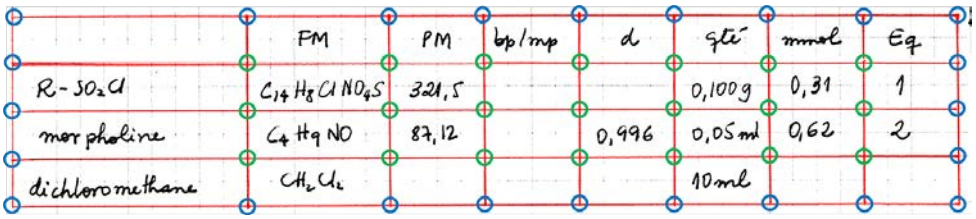
Afin de pouvoir détecter les lignes dégradées ou présentant de petites coupures, deux points alignés sont considérés dans une même ligne s'ils sont séparés par un espace de longueur inférieure ou égale à un seuil que nous avons fixé expérimentalement à 15 pixels. Ce seuil permet aussi de compenser l'effet de l'utilisation de seulement une fraction du nombre total de pixels.

Une fois les lignes graphiques extraites, nous effectuons un filtrage afin de sélectionner uniquement celles qui sont susceptibles d'appartenir à la grille du tableau. Pour cela, les points d'extrémité de chaque ligne sont considérés pour sélectionner celles qui sont horizontales ou verticales et qui ont une longueur supérieure à un seuil donné. Soient une ligne l et ses deux points d'extrémité (x_1, y_1) et (x_2, y_2), l est susceptible d'appartenir à un tableau, si :

$$\arctan(\frac{|y_2 - y_1|}{|x_2 - x_1|}) < \theta_{th} \text{ et } |x_2 - x_1| > Lx_{th} \tag{6.2}$$



(a)



(b)

FIGURE 6.4 – Les deux formes de grilles possibles : (a) incomplète, (b) complète.

ou

$$(x_2 = x_1 \text{ ou } \arctan(\frac{|y_2 - y_1|}{|x_2 - x_1|}) > \frac{\pi}{2} - \theta_{th}) \text{ et } |y_2 - y_1| > Ly_{th} \tag{6.3}$$

où θ_{th} est un seuil, fixé à $\frac{\pi}{18}$, qui est utilisé pour tolérer une petite inclinaison dans les directions horizontale ou verticale. Lx_{th} et Ly_{th} désignent respectivement la longueur minimale d'une ligne horizontale et d'une ligne verticale. Ces deux seuils sont déterminés en fonction de la largeur W et de la hauteur H de l'image du tableau ($Lx_{th} = \frac{3}{4}W$ et $Ly_{th} = \frac{3}{4}H$).

Pour analyser au mieux la grille formée par l'ensemble des lignes, nous avons effectué une correction de l'inclinaison de ces lignes. Soit $l_h = \{(x_i, y_i)_i\}$ une ligne horizontale composée de N points $(x_i, y_i)_i$ avec $1 \leq i \leq N$. L'inclinaison de l_h est corrigée en alignant l'ensemble de ses points dans la position verticale d'ordonnée y_h égale à la moyenne des ordonnées de tous les points, c'est-à-dire $y_h = \frac{\sum_{i=1}^{N} y_i}{N}$. De la même manière, l'inclinaison d'une ligne verticale composée de M points $\{(x_j, y_j)_j\}$ avec $1 \leq j \leq M$ est corrigée en alignant l'ensemble de ses points dans la position horizontale d'abscisse x_v égal à la moyenne des ordonnées de tous les points, c'est-à-dire $x_v = \frac{\sum_{i=1}^{M} x_i}{M}$

6.3.2 Analyse de la grille graphique

Les lignes graphiques d'un tableau se croisent pour constituer une grille. Classiquement, deux formes de grilles structurant un tableau peuvent être rencontrées. La première est une grille incomplète où seulement des lignes séparant les entêtes du corps du tableau sont présentes (voir Figure 6.4(a)). La deuxième est un grille complète où toutes les cellules sont séparées par des lignes graphiques (voir Figure 6.4(b)).

La différenciation entre ces deux grilles est effectuée en analysant les points d'intersection des lignes. Pour ce faire, ces points sont déterminés tout en distinguant ceux qui sont limitrophes (entourés en bleu dans la figure 6.4) et ceux qui sont internes au tableau (entourés en vert dans la figure 6.4). Une grille est incomplète si elle contient un seul point d'intersection interne, et elle est complète dans le cas contraire.

— Dans le cas où la grille est complète, le tableau est entièrement segmenté en cellules en considérant chaque quadruplet de points d'intersection adjacents comme boîte englobante d'une cellule.

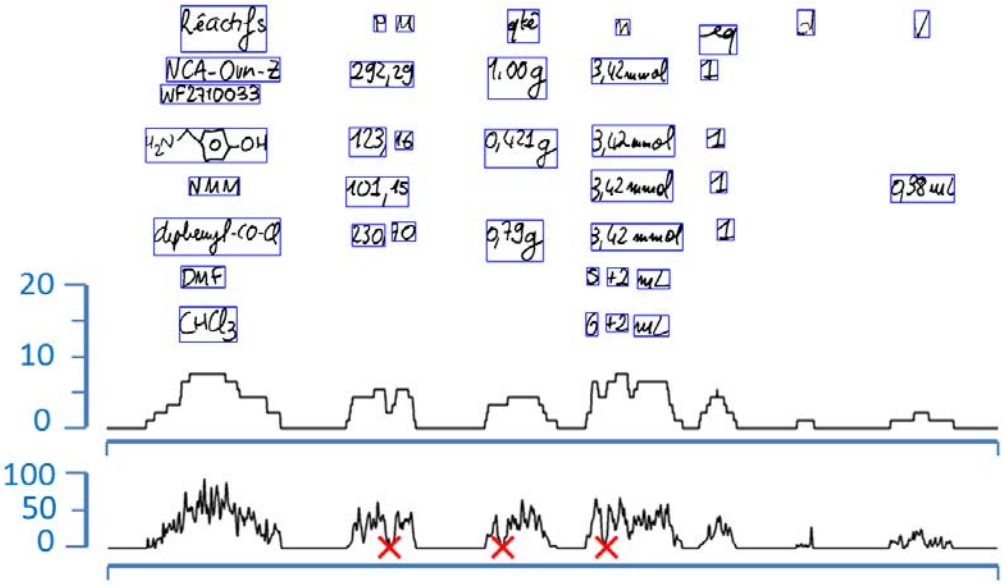


FIGURE 6.5 – Image d'un tableau segmenté en mots et les profils de projection des boîtes englobantes et des pixels.

— Dans le cas où la grille est absente ou incomplète, nous utilisons la technique de projection pour effectuer une première localisation des cellules, qui sera raffinée par l'analyse syntaxique du contenu.

6.3.3 Projection des boîtes englobantes

Nous effectuons une projection verticale des boîtes englobantes des mots en vue de détecter des flux d'espaces verticaux traversant le tableau en hauteur. Projeter les boîtes englobantes des mots plutôt que les pixels, permet d'ignorer les petits espaces (intra-mots) et de détecter seulement les espaces qui sont susceptibles d'être inter-colonnes (voir Figure 6.5).

Le profil de projection verticale obtenu est utilisé pour séparer des blocs de texte verticaux que nous appelons pseudo-colonnes. Ils correspondent aux colonnes d'un tableau si ce dernier a une structure parfaite (pas de chevauchement horizontal entre les cellules). Le tableau est ensuite segmenté en pseudo-cellules (par analogie à l'appellation pseudo-colonnes) qui sont définies comme les intersections des lignes et des pseudo-colonnes.

Les pseudo-cellules ainsi obtenues seront raffinées par une analyse du contenu que nous décrivons dans la section suivante.

6.4 Niveau contenu

L'analyse du contenu d'un tableau repose sur l'extraction de champs numériques qui constituent des éléments structurants de la syntaxe des cellules. L'observation du texte dans un tableau montre qu'il est majoritairement composé de caractères isolés (principalement des chiffres) qui ont à peu près la même taille (les statistiques sur une base de 220 documents montrent que les tableaux contiennent 7527 chiffres isolés et 5627 lettres isolés sur un total de 20348 composantes connexes). Aussi, une première opération consiste à estimer les dimensions de caractères isolés et regrouper les composantes connexes en trois groupes en fonction de leurs dimensions. Les petites composantes connexes représentent des fragments

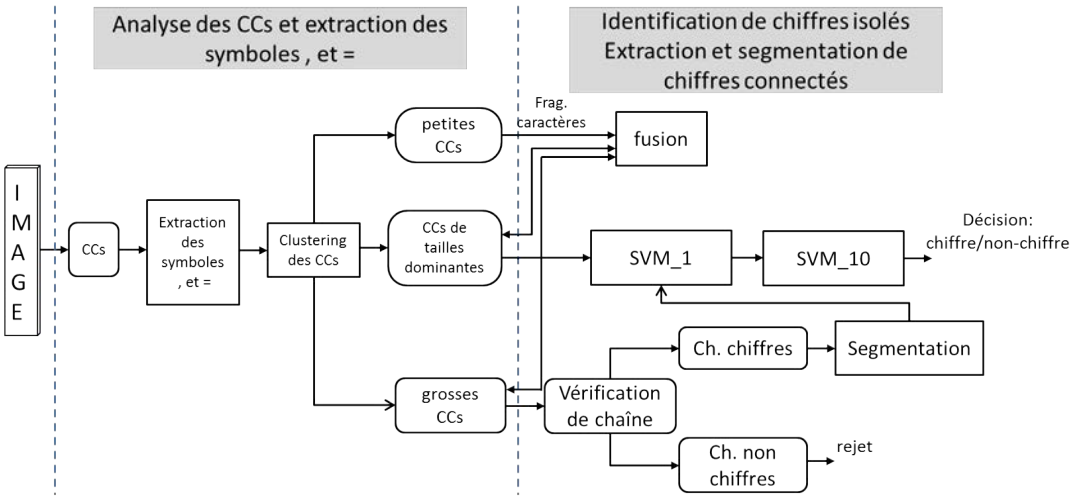


FIGURE 6.6 – Étapes de l'extraction de chiffres.

de caractères et doivent donc être fusionnées avec d'autres composantes en vue de constituer des caractères complets. Les composantes de taille dominante sont des caractères isolés. Elles sont envoyées à un système de reconnaissance afin de décider si elles sont des chiffres ou pas. Les grandes composantes représentent des chaînes de lettres ou de chiffres connectés. En se basant sur leurs structures, celles qui sont très probablement des chaînes non-numériques sont exclues. Les autres seront segmentées afin de reconnaître séparément leurs fragments et de décider de leur nature (chaînes numériques ou non).

Ces différentes étapes sont illustrées dans la figure 6.6.

6.4.1 Extraction et filtrage des CCs

L'objectif de ce module est d'extraire les composantes connexes dans un tableau, identifier les deux symboles spéciaux "," (virgule décimale) et "=" (opérateur égal) et regrouper les autres composantes connexes en 3 groupes en fonction de leurs dimensions.

Les symboles spéciaux sont identifiés dans chaque ligne en utilisant des conditions géométriques définies sur les formes et les positions des composantes connexes. Soient $\{C_i\}$ l'ensemble de ces composantes et $\{bb_i\} = \{(x_i, y_i, w_i, h_i)\}$ leurs boîtes englobantes respectives. La ligne de base est déterminée en appliquant une régression linéaire sur les centres des côtés bas des boîtes englobantes. Soit y_{bl} l'ordonnée de cette ligne.

La virgule décimale se présente comme une composante connexe $CC(x_{cc}, y_{cc}, w_{cc}, h_{cc})$ satisfaisant les conditions suivantes :

1. Sa plus grande partie est située en dessous de la ligne de base.
2. Elle a une élongation verticale importante.
3. Elle ne présente pas de fluctuation.

Ces critères sont exprimés par les équations suivantes :

$$\begin{cases} y_{cc} + \alpha.w_{cc} \leq y_{bl} & \text{(6.4a)} \\ h_{cc}/w_{cc} > r_{min} & \text{(6.4b)} \\ \max\limits_{x}(cross\_count(CC, x)) < 2,\ x \in [\![x_{cc}, x_{cc} + w_{cc}]\!] & \text{(6.4c)} \end{cases}$$

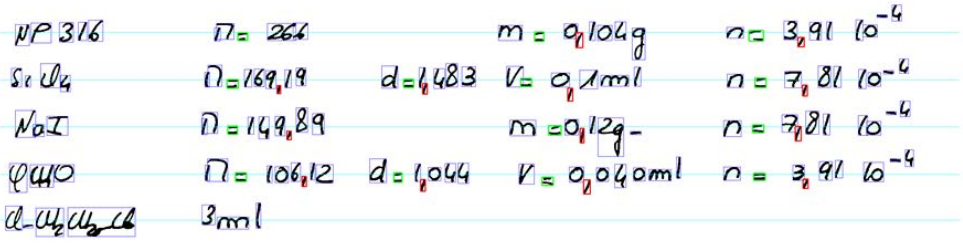


FIGURE 6.7 – Extraction des symboles "," et "=" dans un tableau. Les composantes connexes sont encadrées en bleu et les symbole "=" et "," sont respectivement encadrés en vert et en rouge.

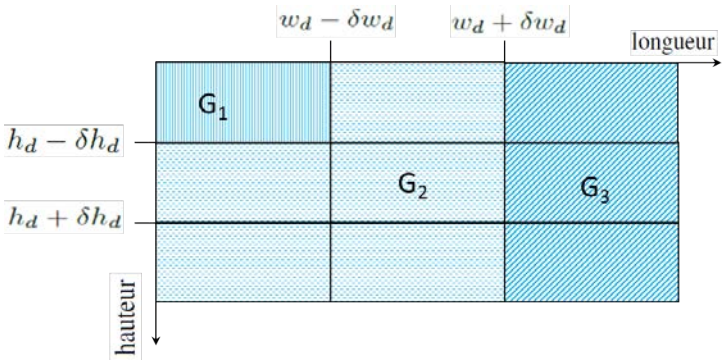


FIGURE 6.8 – Regroupent des composantes en 3 groupes selon leurs dimensions.

où $cross\_count(CC, x))$ est le nombre de points d'intersection entre une ligne verticale d'abscisse x et la composante CC ; α et r_{min} sont deux paramètres fixés respectivement à 0.2 et 1.5.

L'opérateur égal est identifié comme étant une paire de composantes (CC_1, CC_2) telle que :

— chacune des composantes CC_1 et CC_2 est allongée horizontalement ;

— CC_1 et CC_2 se chevauchent horizontalement.

Ces critères sont exprimés par les équations suivantes :

$$\begin{cases} O_h(CC_1, CC_2) > 0 & \text{(6.5a)} \\ h_{CC_i}/w_{CC_i} > r_{max},\ i = 1, 2 & \text{(6.5b)} \end{cases}$$

La figure 6.7, illustre la localisation des symboles "," et "=" dans un exemple de tableau.

6.4.2 Discrimination chiffre/non chiffre

Une fois les symboles spéciaux identifiés dans tout le tableau, toutes les autres composantes connexes sont regroupées selon leurs dimensions. Pour ce faire, les histogrammes des longueurs et des hauteurs de ces composantes sont établis. Chaque histogramme présente un pic qui correspond à la dimension la plus fréquente : celle des chiffres isolés. Soient h_d et w_d respectivement la longueur et la hauteur la plus fréquente. Les composantes connexes sont réparties en 3 groupes, comme illustré dans la figure 6.8, où δ_d est un facteur (fixé expérimentalement à 0.2), introduit pour tolérer une petite variation des dimensions estimées.

Une petite composante connexe (du groupe G_1) correspond à un fragment de caractère. Elle doit être fusionnée avec une autre composante afin de former des caractères entiers. Nous avons utilisé la même stratégie présentée dans [Ha1998] et qui est basée sur le chevauchement horizontal et la distance verticale entre une petite composante connexe et ses voisines, pour effectuer cette fusion.

Les composantes connexes du groupe G_2 correspondent à des caractères isolés (chiffres ou lettres) et celles du groupe G_3 correspondent à des fragments de mots, des mots entiers ou des chaînes de chiffres connectés. Parmi toutes les composantes connexes, celles qui ont des descendants sont des composantes non numériques, elles sont rejetées.

Les traitements des composantes connexes du groupe $G2$ et $G3$ sont expliqués respectivement dans les sections 6.4.2.1 et 6.4.2.2.

6.4.2.1 Détection de chiffres isolés

Cette tâche vise à identifier et reconnaître les chiffres isolés parmi toutes les composantes de la classe G_2. Pour filtrer au mieux les composantes non chiffres (rejet), nous avons adopté une stratégie en deux étapes organisées de manière séquentielle.

• La première étape consiste à distinguer les chiffres des non-chiffres sans aucune reconnaissance de leurs valeurs. Ceci peut se faire de deux façons différentes, en utilisant une classification en 2 classes : la classe des chiffres et la classe de rejet ou en utilisant une classification à une classe. Les éléments de la classe rejet présentent une grande variabilité et par conséquent, il est très difficile de collecter des exemples représentatifs de cette classe afin de les utiliser dans l'apprentissage du classifieur, si l'on veut utiliser une classification à deux classes. Nous avons donc opté pour une classification à une classe où nous avons besoin seulement des exemples de la classe des chiffres dans la phase d'apprentissage.

En se basant sur une étude complète et détaillée sur les techniques de classification à une classe présentée dans [Khan2014], nous avons choisi d'utiliser un SVM-1classe. Cette technique se révèle être la plus performante et la plus utilisée notamment dans la reconnaissance de chiffres [Scholkopf1999, Tax2001, Hao2008].

La particularité de cette étape est de pouvoir filtrer le maximum de rejet tout en acceptant tous les chiffres. En particulier, les composantes connexes situées à gauche et à droite d'une virgule décimale "," ainsi que celles situées à droite d'un opérateur "=" ne sont pas traitées dans cette étape et elles sont automatiquement acceptées.

• La seconde étape, consiste à filtrer davantage les rejets parmi les composantes qui ont été acceptées dans la première étape. Nous avons utilisé un classifieur à 10 classes correspondant aux 10 chiffres. Le filtrage des rejets est effectué en se basant sur les scores de confiance de chaque composante [Pitrelli2003]. Si, pour une composante donnée, le meilleur score obtenu est inférieur à un certain seuil (seuil de rejet noté T_r), la composante est considérée comme non-chiffre et elle sera rejetée. Ce seuil est optimisé sur la base d'apprentissage de manière à ce qu'aucun chiffre ne soit rejeté (un taux de faux négatif nul).

Même si nous effectuons une reconnaissance des chiffres lors de cette étape, le critère le plus important, dans le choix du meilleur classifieur, est toujours de maximiser le taux de vrai rejet (VR) tout en minimisant le taux de faux rejet (FR) et ceci sans prendre en compte si les chiffres sont bien reconnus ou pas. Le taux de VR est égal au nombre de non-chiffres qui sont rejetés par le classifieur, divisé par le nombre total des non-chiffres. Le taux de faux rejet est le nombre de chiffres qui sont rejetés, divisé par le nombre total de chiffres. Nous avons comparé deux classifieurs, un PMC et un SVM, en utilisant la courbe ROC qui permet la représentation graphique du couple (taux de VR ; taux de FR). Afin d'obtenir les points nécessaires pour établir la courbe ROC, nous avons calculé ces deux taux pour différentes valeurs du seuil de rejet T_r. La courbe obtenue est illustrée dans la Figure 6.9. A partir de cette courbe, nous constatons que pour un taux de faux rejet nul, le SVM permet d'effectuer plus de vrais rejets (41.2%) que le PMC (33.7%). Nous avons donc choisi d'utiliser un SVM.

Il convient de noter que pour utiliser le SVM, qui est un classifieur initialement conçu pour des classifications binaires, nous avons adopté la stratégie "un contre un" qui consiste à utiliser un classifieur

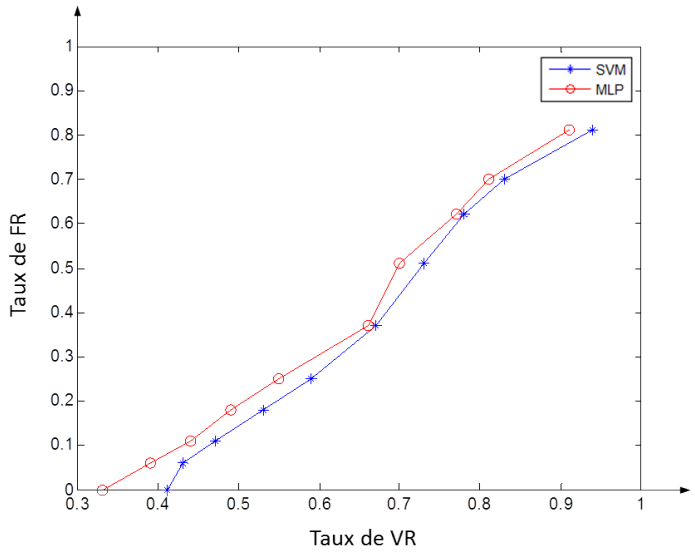


FIGURE 6.9 – Courbe ROC pour les deux classifieurs PMC et SVM.

par paire de classes [Guermeur2007]. S'agissant d'une classification en 10 classes (correspondant aux 10 chiffres), nous avons utilisé $C_{10}^2 = 45$ SVMs binaires, où C_{10}^2 dénote la combinaison de 2 parmi 10. Chaque classifieur est appris sur un ensemble d'exemples (d'apprentissage) représentant les deux classes correspondantes. Dans la phase de prédiction, un exemple est présenté aux 45 classifieurs et la décision de sa classe est prise en effectuant un vote majoritaire. La voix de chaque classifieur est pondérée par la valeur de la sortie calculée.

Pour la classification d'une composante connexe, l'image binaire elle-même est fournie en entrée du SVM. D'abord, l'image est redimensionnée en 20×20 pixels tout en conservant son ratio hauteur/largeur. Ensuite, elle est translatée pour positionner le centre de masse de pixels au centre d'une boîte 28×28.

Le processus d'identification des chiffres isolés est lancé sur chaque document. Les composantes connexes classées comme chiffres sont utilisées pour estimer la largeur moyenne w_m d'un chiffre dans le document. Cette largeur est utilisée pour guider la localisation de zones de connexion dans les chaînes de chiffres connectés, dont le traitement est décrit dans la section suivante.

6.4.2.2 Détection et segmentation de chiffres connectés

Dans cette étape, nous traitons les composantes connexes du groupe G_3 afin de détecter celles qui sont numériques. Nous supposons qu'une composante de ce groupe est, soit une chaîne alphabétique pure, soit une chaîne numérique pure. Cette hypothèse est basée sur l'observation des composantes connexes contenues dans un tableau, où les connexions entre des chiffres et des lettres sont très rarement rencontrées.

L'analyse des chaînes alphabétiques, nous a permis de les classer en deux types :

— type 1 : rassemble des chaînes possédant une forme très différente des chaînes numériques. Elles contiennent des lettres ayant des hauteurs différentes, ce qui les rend facilement distinguables des chaînes numériques qui sont composées de chiffres ayant à peu-près la même hauteur.

— type 2 : rassemble des chaînes dont la forme est proche des chaînes numériques. Elles contiennent des lettres de même hauteur, ce qui rend leur différenciation des chaînes numériques plus difficile que celles de type 1.

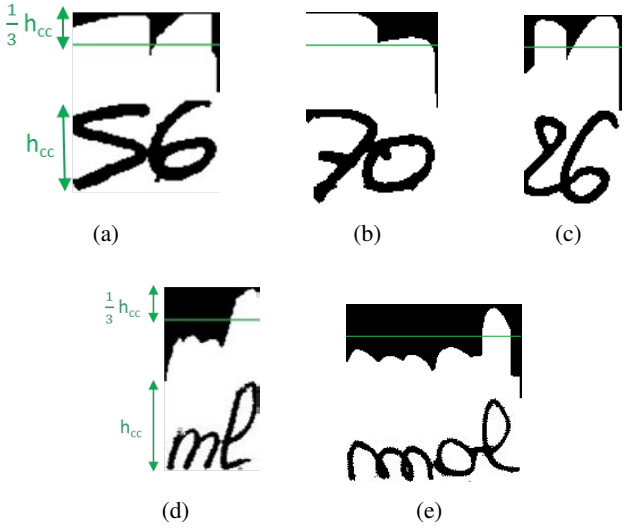


FIGURE 6.10 – Profils de projection supérieur de chaînes numériques (a, b and c) et de chaînes alphabétiques (d and e).

Cette observation nous a conduit à adopter une stratégie en deux étapes pour rejeter les chaînes alphabétiques. La première est basée sur la régularité de la hauteur pour rejeter les chaînes de type 1, et la deuxième est basée sur la segmentation-reconnaissance pour rejeter les chaînes de type 2.

Rejet des chaînes alphabétiques de type 1. Cette étape est effectuée sans aucune segmentation ni reconnaissance et ceci en se basant uniquement sur la régularité des hauteurs des composantes connexes. Pour étudier cette caractéristique, nous analysons le profil supérieur de chacune de ces composante (voir Figure 6.10). Dans cette analyse, nous supposons que la hauteur des lettres basses telles que 'a', 'c', 'e', 'm', 'n', 'o', etc. est en général inférieure ou égale à $2/3$ de la hauteur des lettres avec des ascendants tels que 'b', 'd', 'f', 'h', 'k', 'l', 't'. Ainsi, la présence de caractères de hauteurs différentes dans une composante connexe CC_i de hauteur h_{cc_i} se manifeste dans son profil supérieur par des vallées de profondeur supérieure à $h_{cc_i}/3$ qui s'étendent sur une partie importante de la largeur de CC_i.

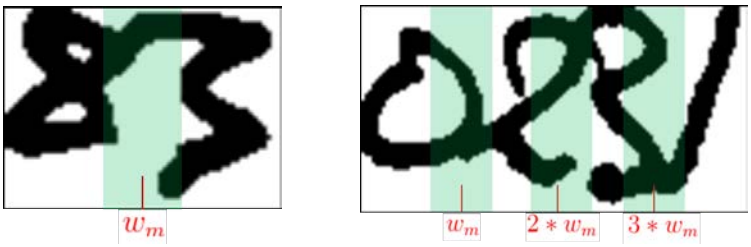
Soit lv_{max} la largeur de la plus grande vallée dont la profondeur est supérieure à $h_{cc_i}/3$. Si lv_{max} est supérieure ou égale à la largeur moyenne d'un caractère w_m (estimée dans la section 6.4.2.1), ce qui implique la présence d'au moins une lettre basse, alors CC_i est considérée comme une chaîne alphabétique et elle est rejetée. Dans l'autre cas ($lv_{max} \leq w_d$), la vallée peut correspondre à la région où deux caractères successifs se touchent (un réservoir d'eau comme appelé dans [Pal2002]). La composante CC_i est alors retenue pour décider de sa nature (numérique ou alphabétique) en utilisant une méthode de segmentation-reconnaissance qui sera expliquée dans la section suivante.

Rejet des chaînes de type 2. L'idée est de segmenter chacune des chaînes qui ont été retenues dans l'étape précédente, de reconnaître les fragments résultants et de décider de la nature de la chaîne en se basant sur les scores de reconnaissance.

Nous utilisons un algorithme de segmentation similaire à celui proposé dans [Sadri2004] qui consiste en les étapes suivantes :

— l'extraction de points caractéristiques issus d'une analyse du contours de la forme et du fond (arrière plan) de la composante connexe à segmenter ;

— la construction de chemins de segmentation (appelés aussi chemins de coupure) à partir des points



(a) (b)

FIGURE 6.11 – Zone de connexion potentielle (en vert) estimée en se basant sur la longueur moyenne d'un chiffre isolé.

caractéristiques en utilisant des règles sur leurs positions et les distances qui les séparent ainsi que des contraintes sur le chevauchement et les hauteurs des fragments obtenus.

Cet algorithme présente l'inconvénient de générer un grand nombre de chemins de segmentation surtout que les composantes connexes que nous traitons peuvent être non-numériques et de longueurs quelconques. Pour pallier ce problème, nous utilisons la largeur moyenne w_m de chiffres (déterminée dans la section 6.4.2.1) pour estimer les zones susceptibles de contenir des connexions entre deux chiffres dans la composante connexe : ce sont les zones centrées horizontalement sur les points d'abscisses $x_{cc} + i * w_m$, où $i = 1, ..., n$ et $i * w_m < w_{cc}$ (voir Figure 6.11). Ainsi, au lieu de considérer tous les points caractéristiques de la composante connexe, nous examinons seulement les points qui sont situés dans les zones de connexion. Ceci permet de réduire considérablement le nombre de chemins de coupure résultants.

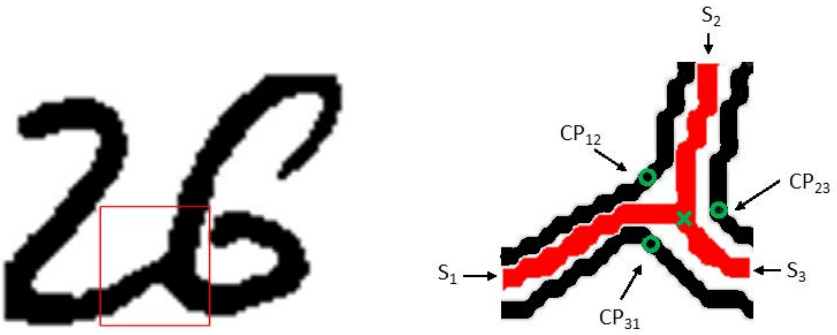
Dans ce qui suit, nous décrivons l'extraction des points caractéristiques à partir d'une composante connexe et la construction de chemins de coupure à partir de ces points.

Les points caractéristiques du contour. Ils sont extraits à partir des points de jonctions de la composante connexe. Un point de jonction est un pixel du squelette de la composante, ayant au moins trois pixels voisins noirs. Pour le détecter, l'image est d'abord squelettisée en utilisant l'algorithme de Zhang-Suen [Zhang1984]. Ensuite, chaque point du squelette est visité et les valeurs des pixels situés dans son 8-voisinage sont examinées. Tous les points ayant au moins trois pixels voisins noirs sont des points de jonction.

A partir de chaque point de jonction, nous déterminons un ensemble de points caractéristiques situés sur le contour extérieur de la composante connexe. Ces points sont obtenus par la projection d'un point de jonction sur chaque fragment du contour délimité par une paire de segments qui passent par ce point. A la différence de [Sadri2007] qui utilise la bissectrice de l'angle formée par chaque paire de segments, nous cherchons le point le plus proche dans la partie du contour située dans cet angle.

La Figure 6.13(b) illustre le squelette (en rouge) et le contour extérieur (en noir) de la forme encadrée en rouge dans la Figure 6.12(a). 3 segments S_1, S_2 et S_3 passent par le point de jonction (représenté par une croix verte) extrait à partir du squelette. Entre chaque paire de segments S_i et S_j, est extrait un point de contour CP_{ij}.

Les points caractéristiques du fond. Comme dans [Sadri2007], nous n'utilisons pas tous les pixels du fond y compris ceux qui sont internes à la composante connexe, mais seulement, les parties essentielles sont considérées. Ce sont les pixels de fond vus du dessus et de dessous de l'image de la composante connexe. D'abord, les profils de projection supérieur et inférieur sont construits et squelettisés. Ensuite, les points terminaux du squelette sont déterminés. Un point terminal est un pixel du squelette qui a un

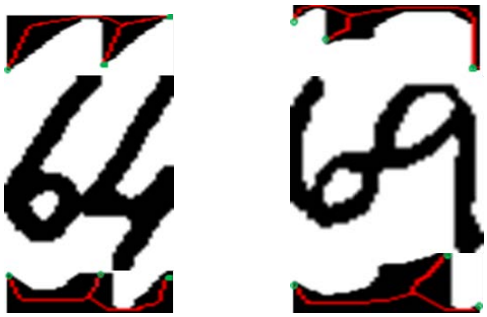


(a) (b)

FIGURE 6.12 – Extraction des points caractéristiques de contours.

(a) (b)

FIGURE 6.13 – Exemples de points caractéristiques de contour extraits à partir des images de chiffres connectés. Le squelette de l'image est en rouge. Les points de jonctions sont représentés par des croix vertes et les points caractéristiques du contour sont représentés par des petits cercles verts.



(a) (b)

FIGURE 6.14 – Exemples de points caractéristiques de fond extraits à partir des profils de projection supérieur et inférieur des images de chiffres connectés. Les profils de projection sont illustrés en haut et en bas de chaque composante connexe. Les squelettes correspondants sont en rouge. Les points terminaux sont présentés sous formes de petits cercles verts.

seul pixel noir dans son 8-voisinage.

La Figure 6.14 illustre l'extraction des points caractéristiques de fond, à partir de deux images de chiffres connectés.

Construction du chemin de segmentation. La composante connexe est balayée de gauche à droite et l'ensemble des points caractéristiques situés dans chacune des zones de connexion sont considérés pour construire le chemin de segmentation. La construction du chemin de segmentation est décrite dans l'algorithme 1. D'abord, l'ensemble des points caractéristiques du contour est parcouru verticalement (de haut en bas ou inversement) pour sélectionner et relier ceux qui sont séparés par une distance horizontale inférieure à un certain seuil T_x. Ensuite, le chemin CS obtenu est complété, en haut et en bas, jusqu'à atteindre les limites inférieure et supérieure de la composante connexe. Cette opération est effectuée en reliant le point le plus bas CS_b (respectivement le plus haut CS_u) du chemin CS au point caractéristique BP_s du profil inférieur (respectivement le point BU_s du profil supérieur) le plus proche horizontalement, si la distance horizontale entre CS_b et BP_s (respectivement entre CS_u et BU_s) est inférieure au seuil T_x. Sinon, le chemin est complété par un segment vertical jusqu'aux limites inférieure et supérieure de la composante connexe. Dans le cas, où aucun point caractéristique de contour n'est trouvé, le chemin de coupure est construit selon le nombre de points de profils supérieur et inférieur et la distance horizontale qui les sépare. Les distances verticales de ces points vers les limites supérieure et inférieure de la composante connexe sont aussi prises en compte.

Les figures 6.15 et 6.16 illustrent la construction de chemins de segmentation de deux composantes connexes composées respectivement de deux et de trois chiffres.

Il convient de noter que la méthode de segmentation proposée génère un seul chemin de coupure. Ceci provient du fait de considérer uniquement les points caractéristiques situés dans la zone de connexion estimée et d'utiliser un ensemble de contraintes sur les positions et les distances entre ces points.

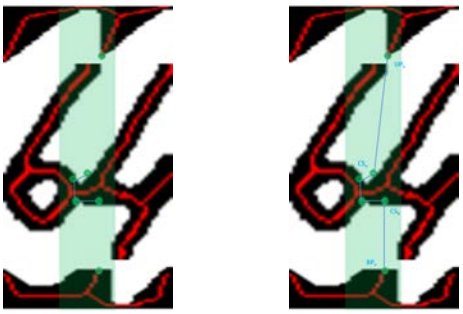
Identification des chaînes numériques. Une fois la segmentation d'une composante connexe est effectuée, les fragments obtenus sont envoyés au système de détection de chiffres isolés (décrit dans la section 6.4.2.1). Ceci permet de déterminer le type (chiffre ou rejet) de chaque fragment ainsi que sa valeur s'il s'agit d'un chiffre. La composante connexe est considérée comme une chaîne numérique si tous ses fragments sont reconnus comme étant des chiffres.

Algorithm 1 Construction de chemin de segmentation

Entrées: points de contour $CP = \{CP_1, CP_2, ...CP_k\}$, points de profil inférieur $BP = \{BP_1, BP_2, ...BP_m\}$, points de profil supérieur $UP = \{UP_1, UP_2, ..., UP_n\}$

Sortie: chemin de segmentation $CS = \{p_1, p_2, ..., p_s\}$

si $|CP| > 1$ **alors**
//sélectionner les points de CP horizontalement proches
    $CS = selectCP(CP)$
//compléter le chemin jusqu'à la limite inférieure de la CC
    $CS_b = plusBasPoint(CS)$
    $BP_s = plusProcheH(CS_b, BP)$
    si $distH(BP_s, CS_b) \leq T_x$ **alors**
        $CS.ajouter(BP_s)$
    sinon
        $CS.ajouter(Point(CS_b.x, CC.buttom))$
    fin si
//compléter le chemin jusqu'à la limite supérieure de la CC
    $CS_u = plusHautPoint(CS)$
    $UP_s = plusProcheH(CS_u, UP)$
    si $distH(UP_s, CS_u) \leq T_x$ **alors**
        $CS.ajouter(UP_s)$
    sinon
        $CS.ajouter(Point(CS_u.x, CC.top))$
    fin si
sinon //aucun point caractéristique de contour n'est détecté.
    si $|BP| \geq 1$ **et** $|UP| \geq 1$ **alors**
        $(BP_s, BU_s) = \underset{i \in [1,n], j \in [1,m]}{\arg\min} (|distH(BP_i, UP_j)|)$
        si $|distH(BP_s, UP_s)| \leq T_x$ **alors**
            $CS = \{BP_s, UP_s\}$
        sinon
            $UP_b = plusBasPoint(UP)$
            $UP_v = Point(UP_b.x, CC.buttom)$ //projection verticale de UP_b sur le bord inférieur de la CC
            $BP_u = plusHautPoint(BP)$
            $BP_v = Point(BP_u.x, CC.top)$
            si $|distV(UP_b, UP_v| \leq |distV(BP_u, BP_v|$ **alors**
                $CS = \{UP_b, UP_v\}$
            sinon
                $CS = \{BP_u, BP_v\}$
            fin si
        fin si
    sinon
        si $|BP| \geq 1$ **alors**
            $BP_u = plusBasPoint(BP)$
            $BP_v = Point(BP_b.x, CC.top)$
            $CS = \{BP_u, BP_v\}$
        fin si
        si $|UP| \geq 1$ **alors**
            $UP_b = plusBasPoint(UP)$
            $UP_v = Point(UP_b.x, CC.buttom)$
            $CS = \{UP_b, UP_v\}$
        fin si
    fin si
fin si

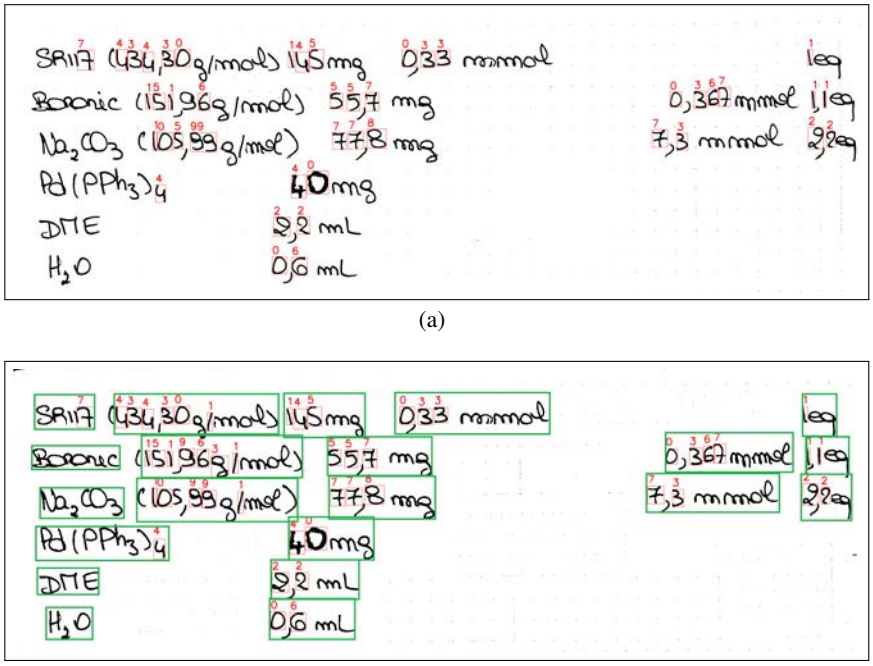


(a) (b)

FIGURE 6.15 – Construction du chemin de coupure d'une chaîne composée de deux chiffres, (a) les points caractéristiques de contours sont tous reliés entre eux car les distances verticales qui les séparent sont faibles, (b) le chemin est complété jusqu'aux limites inférieure et supérieure de la composante connexe en connectant respectivement le point de contour le plus bas et le plus haut au point caractéristiques de profil inférieur et supérieur.

(a) (b) (c) (d)

FIGURE 6.16 – Construction de chemins de coupure d'une chaîne composée de trois chiffres, (a) les points caractéristiques de contours et de profils situés dans les zones de connexions sont présentés sous formes de petits cercles verts, (b) le premier chemin de coupure est déterminé en utilisant le point de profil supérieur et sa projection verticale sur le bord inférieur de la composante connexe, (c) les points de contours situés dans la deuxième zone de connexion sont reliés pour initialiser le deuxième chemin de coupure, (d) ce chemin est complété jusqu'aux limites inférieure et supérieure de la composante connexe.



(a)

(b)

FIGURE 6.17 – (a) Exemple de tableau où les composantes numériques sont localisées. (b) Délimitation des cellules en vert.

6.4.3 Correction de la segmentation par l'analyse du contenu

Cette tâche consiste à exploiter la cartographie des composantes numériques dans chaque ligne pour corriger les pseudo-cellules obtenues par la projection orthogonale des boîtes englobantes de façon à obtenir des cellules ayant une syntaxe uniforme. La première ligne pouvant être un entête contenant des libellés qui ne vérifient aucune syntaxe est examiné séparément. Si la densité en composantes numériques dans cette ligne est inférieure à un certain seuil, elle est considérée comme entête et ses cellules seront corrigées en fonction du résultat de la segmentation du corps du tableau.

Pour la segmentation d'une ligne du corps du tableau, nous distinguons les deux cas suivants :

— Toutes les composantes connexes de la ligne sont numériques et des virgules décimales. Dans ce cas, les cellules sont supposées être suffisamment espacées et le résultat de la segmentation obtenu par la projection verticale des boîtes englobantes est considérée correcte.
— En plus des composantes numériques, la ligne contient des composantes non numériques (correspondant aux grandeurs chimiques ou aux unités). Dans ce cas, la syntaxe exacte des cellules est déterminée en tenant compte du symbole spécial "=" et/ou des positions relatives des composantes numériques par rapport aux autres composantes. Ensuite, la ligne est balayée de droite à gauche et les cellules sont corrigées selon la syntaxe déterminée.

Considérons par exemple le tableau illustré dans la figure 6.17(a). En balayant une ligne de droite à gauche, la première composante connexe est non-numérique et comme il n'y a pas de symbole "=", nous déduisons que la syntaxe des cellules est NU. Pour avoir des cellules vérifiant cette syntaxe, chaque groupe de composantes non-numériques adjacentes doit être regroupé avec le groupe de composantes numériques situé à sa gauche. Le résultat obtenu est illustré dans la figure 6.17(b).

TABLE 6.1 – Résultats de l'extraction des cellules

Tableau			Rappel (%)	Précision (%)
Type	nombre	nombre de cellules		
avec grille complète	64	2460	95.12	97,78
avec grille incomplète	8	120	93,33	91,80
sans grille	148	2534	85.51	87,80

6.5 Expérimentation et résultats

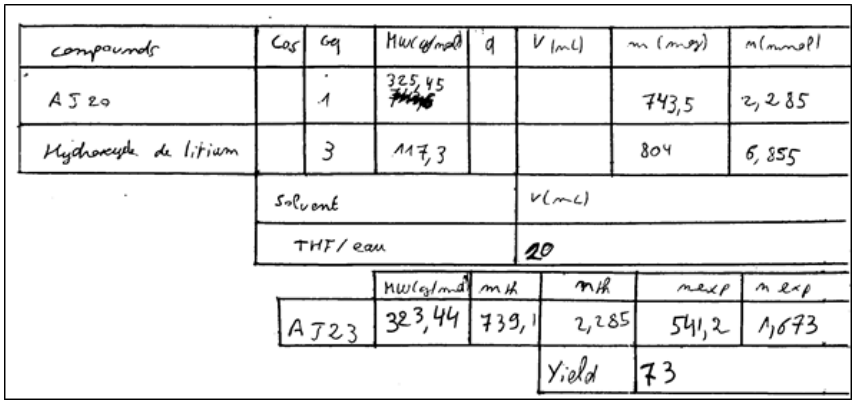
Nous présentons dans cette section, les résultats de la méthode d'extraction de cellules sur 220 tableaux de la base ChemicalTable (les 280 documents restants sont utilisés pour l'apprentissage des deux classifieurs de chiffres ainsi que pour le réglage des différents seuils utilisés). Comme il s'agit d'une chaîne de traitement, nous présentons d'abord les performances du système vis-à-vis de l'objectif final : l'extraction des cellules. Ensuite, dans le but d'une analyse plus fine des résultats obtenus, nous présentons quelques résultats intermédiaires, notamment l'extraction des chiffres dans le tableau.

Évaluation de l'extraction des cellules. Les métriques de performance utilisées dans ces expérimentations sont là encore le rappel et la précision déterminés à partir de la matrice MatchScore (définie dans le chapitre précédent) que nous calculons entre les cellules extraites par le système et celles définies dans la vérité terrain. Le seuil d'acceptation est fixé à 0.9. Les résultats obtenus sont présentés dans le tableau 6.1. Nous constatons que les meilleures performances sont obtenues sur les tableaux à grilles complètes. Ceci s'explique par la fiabilité de la méthode d'extraction des lignes graphiques. Les quelques erreurs produites sur les tableaux de ce type sont dues à la dégradation de certaines lignes graphiques au cours des étapes de numérisation et/ou de pré-traitements des images. D'autres erreurs sont dues à l'inadéquation des paramètres utilisés dans la sélection des lignes qui appartiennent au tableau, notamment les deux paramètres Lx_{th} et Ly_{th}. Par exemple, dans la figure 6.18, les lignes pointées par les flèches bleues ont été détectées par la transformée de Hough mais elles ont été rejetées (lors du filtrage des lignes susceptibles d'appartenir au tableau) car elles ont une hauteur inférieure au seuil Ly_{th}.

Les résultats obtenus sur les tableaux à grilles incomplètes viennent en deuxième place. Même si ces tableaux, ne sont pas complètement structurés avec des lignes graphiques, leurs structures présentent très peu d'imperfections car les cellules de données tendent à être alignées avec celles des entêtes, ce qui garantit une certaine régularité d'espacement et d'alignement. Par conséquent, la projection orthogonale des boîtes englobantes peut être efficace pour extraire les cellules avec des performances acceptables. Nous rappelons que les cellules obtenues sont raffinées à l'aide d'une analyse au niveau contenu. Cependant, l'amélioration apportée par cette analyse reste modérée (environ 1.5% en rappel et 0.8% en précision).

Les plus faibles performances sont obtenues sur les tableaux sans lignes graphiques car elles présentent une disposition physique plus altérée que celle des autres types de tableau. L'amélioration apportée par l'analyse au niveau contenu est considérable (5.21% en rappel et 3.58% en précision) mais elle reste atténuée par :

— l'incohérence de la syntaxe au sein d'une ligne du tableau qui est due à un oubli de l'un des éléments syntaxiques, notamment les unités, ce qui perturbe le regroupement des composantes connexes en cellules ;
— les erreurs d'extraction des composantes numériques que nous allons analyser de manière détaillée dans le paragraphe suivant.

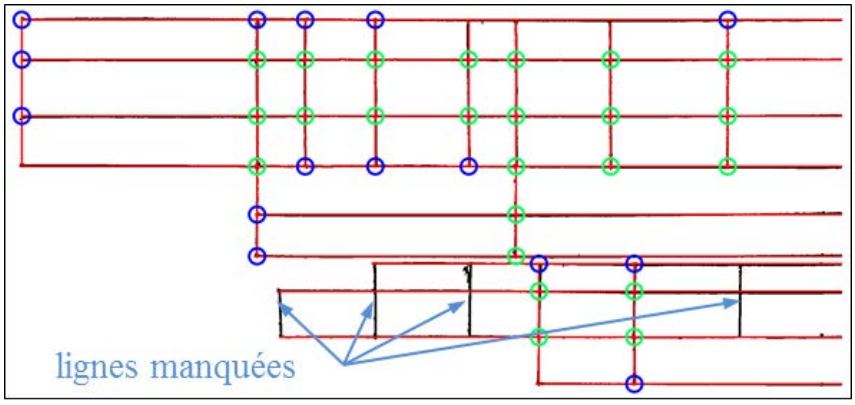


compounds	Cas	Eq	MW (g/mol)	q	V (mL)	m (mg)	n (mmol)
AJ20		1	325,45			743,5	2,285
Hydroxyde de lithium		3	117,3			804	6,855

Solvent	V (mL)
THF/eau	20

	MW (g/mol)	m th	n th	m exp	n exp
AJ23	323,44	739,1	2,285	541,2	1,673
			Yield	73	

(a)



(b)

FIGURE 6.18 – (a) Exemple d'extraction de cellules dans un tableau à grille complète. Les lignes graphiques détectées comme étant des lignes du tableau sont en rouges.

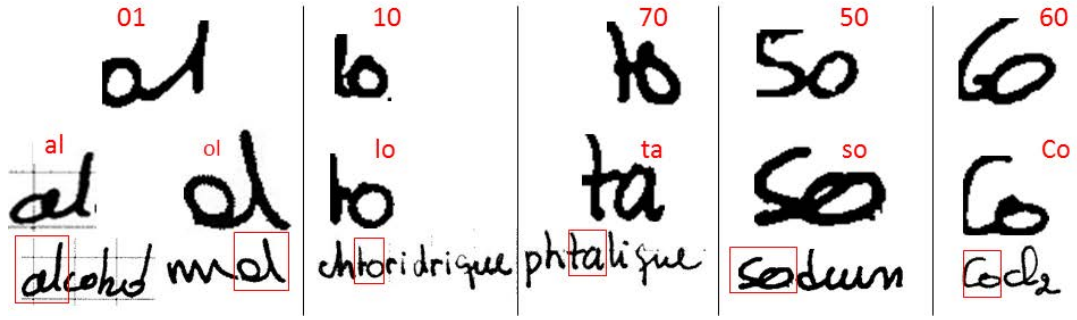


FIGURE 6.19 – Exemples de faux rejets de chiffres connectés (en haut) présentant une grande ressemblance avec des chaînes alphabétiques (en bas) avec les mots auxquels ils appartiennent.

Évaluation de l'extraction des chiffres. Nous calculons le rappel et la précision de la détection des composantes numériques (chiffres isolés et connectés) sans tenir compte de la reconnaissance de leurs valeurs exactes (le fait d'être reconnu comme chiffre suffit). Les résultats sont reportés dans le tableau 6.2.

TABLE 6.2 – performance de l'extraction des composantes numériques.

composantes	nombre total	détectés	corrects	Rappel (%)	Précision (%)
chiffre isolé	7527	8151	6902	91, 69%	84, 67%
chaîne double	462	435	379	82, 03%	87, 12%
chaîne triple	106	94	85	80, 19	90, 42%

Les résultats montrent que même pour les chiffres isolés, les taux obtenus ne sont pas très satisfaisants. Ceci est dû à la difficulté du problème de reconnaissance des chiffres manuscrits en présence de données aberrantes. Pour les chiffres connectés, on note que la précision est meilleure que le rappel. Ceci s'explique par l'opération de filtrage basée sur la hauteur, qui n'accepte que les chaînes susceptibles d'être numériques. Mais au moment de la segmentation-reconnaissance, ces chaînes peuvent être mal reconnues, ce qui explique le faible rappel.

Les principales erreurs observées proviennent de :

— la confusion de chiffres et de lettres isolés tels que "S" et "5", "u" et "4", "o" et "0", etc.

— le rejet de quelques chaînes numériques qui ne vérifient pas l'hypothèse de régularité de la hauteur, ce qui les fait plus similaires à des chaînes alphabétiques (voir Figure 6.19) ;

— l'acceptation de quelques chaînes alphabétiques similaires à des chaînes numériques. En plus de la régularité de la hauteur, la segmentation de ces chaînes permet de fournir des formes similaires à des chiffres isolés (voir Figure 6.20) ;

— des erreurs de segmentation qui sont dues à des cas de connexions très compliquées telles que illustrées dans la figure 6.20.

6.6 Conclusion

Nous avons présenté dans ce chapitre une méthode basée sur l'analyse de la structure et du contenu pour l'extraction des cellules d'un tableau. L'analyse de la structure consiste à détecter les lignes graphiques et à analyser la grille résultante. Pour un tableau qui ne dispose pas d'une grille graphique, nous avons utilisé la projection des boîtes englobantes des mots pour une première détection de cellules, qui est ensuite corrigée par une analyse de la syntaxe du contenu du tableau. Cette opération est effectuée

(a) (b)

FIGURE 6.20 – (a) Exemples de chaînes alphabétiques qui ont été acceptées comme des chaînes numériques. A gauche, les chaînes avec leurs vraies valeurs en vert et à droite, le chemin de segmentation(en vert) avec les valeurs obtenues en rouge.

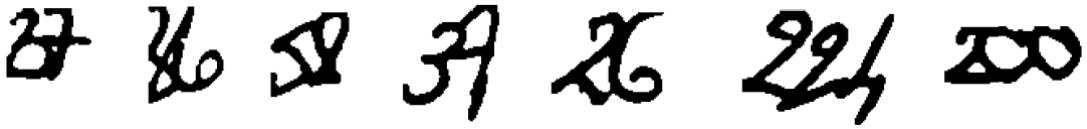


FIGURE 6.21 – Exemples de connexions difficiles à segmenter.

en analysant la cartographie des chiffres présents dans le tableau. Pour cela, nous avons proposé une méthode de segmentation-reconnaissance adaptée à l'extraction des chiffres isolés et connectés.

Chapitre 7

Conclusion et perspectives

Nous avons abordé dans cette thèse le problème de la segmentation des documents de chimie comme un préalable à la reconnaissance et à l'indexation. Les travaux réalisés concernent l'extraction de formules chimiques, la localisation de tableaux et leur segmentation en cellules. Les difficultés de ces tâches viennent de la grande variabilité des documents tant au niveau de style et de la taille d'écriture que de la qualité des images. Cette variabilité provient du fait que les documents sont manuscrits, hétérogènes, non-contraints et multi-scripteurs.

Après avoir exploré l'état de l'art relatif à ces travaux et noté l'inadéquation et la difficulté d'adaptation des méthodes existantes pour traiter ce type de documents, nous avons proposé des méthodes qui tiennent compte des différentes difficultés.

Nous avons abordé les deux premiers problèmes, à savoir l'extraction de formules chimiques et la localisation de tableaux, comme un problème de classification. Pour le premier, nous avons extrait des structures linéaires que nous avons classées en Texte ou Graphique en se basant sur un ensemble de descripteurs structurels. Le choix des structures linéaires comme unité de base pour cette classification, a été motivé par la structure physique des documents, qui peut être vue comme une succession de blocs horizontaux et verticalement séparables. Pour le deuxième, les structures linéaires textuelles ont été, d'abord, affinées pour extraire des lignes de texte. Ces dernières sont ensuite classées selon qu'elles appartiennent à un tableau ou à une région de texte brute. Nous avons proposé pour cette classification, un modèle CAC qui permet de déterminer la séquence optimale d'étiquettes associées à la séquence des lignes d'un document. Le choix de ce type de modèle a été motivé par sa capacité à absorber la variabilité des lignes et à exploiter les informations contextuelles.

Pour le problème de la segmentation de tableaux en cellules, nous avons proposé une méthode hybride qui fait coopérer deux niveaux d'analyse : structurel et syntaxique. Le premier tire avantage de la présence des lignes graphiques et de l'alignement de texte et d'espaces ; et le deuxième tend à exploiter la cohérence de la syntaxe des cellules.

Les différentes phases de notre système ont toutes été implémentées et testées sur une base de 500 documents de chimie, réparties en 280 documents pour l'apprentissage et 220 pour les tests. Nous n'avons pas pu faire de comparaison avec d'autres méthodes à cause de l'absence dans la littérature, des travaux qui traitent ce type de documents. Les résultats obtenus sont :

- $88,63\%$ de formules chimiques parfaitement extraites et de $5,45\%$ partiellement extraites.
- $61,81\%$ de tableaux parfaitement localisés et de $33,18\%$ partiellement localisés.
- les taux de segmentation des tableaux en cellules varient entre $85,53\%$ dans les tableaux sans lignes graphiques et $95,12\%$ dans les tableaux plus structurés, avec des lignes graphiques.

Pour atteindre l'objectif ultime du projet dans lequel est située cette thèse, et qui vise une lecture complète des documents de chimie, plusieurs points pourront être envisagés dans le futur. Certains portent sur l'amélioration des tâches réalisées, et d'autres, sur les phases suivantes du processus de reconnaissance.

— La première perspective consiste à intégrer, de manière intelligente, les trois contributions que nous avons réalisées, dans un système global. Au lieu de les assembler de manière séquentielle, nous pourrons envisager une stratégie d'intégration avec retour en arrière, c'est-à-dire, qui permet d'exploiter les résultats d'un module pour corriger et raffiner les traitements qui le précèdent.

— La deuxième perspective consiste à élargir la base de documents utilisée dans les expérimentations. Pour ce point, nous soulignons que la difficulté réside dans la création de la vérité terrain, et non dans la collecte des documents. D'ailleurs, nous avons pu recevoir dernièrement environ 30000 documents, de notre partenaire industriel eNovalys.

Les autres travaux à planifier pour la suite de ce travail, concernent :

— la reconnaissance et l'interprétation des formules chimiques, comme dans [Ouyang2007, Fujiyoshi2011, Sadawi2013] ;

— le repérage de mots clés dans la région de texte qui décrit l'expérience présentée dans le document. Ceci constituera une alternative à la lecture intégrale du texte qui demeure encore une opération difficile à réaliser avec des résultats satisfaisants. Les informations extraites seront utilisées pour l'indexation des documents pour être exploités par les chimistes en utilisant le moteur de recherche eNovalys ;

— l'interprétation complète du tableau par extraction de sa structure logique et reconnaissance de son contenu qui est constitué d'un vocabulaire relativement réduit : des montants, des grandeurs et des unités.

Annexe A

Liste de publications

— N. Ghanmi and A. Belaïd. *Extraction de formules chimiques dans des documents manuscrits composites*. Colloque International Francophone sur l"Ecrit et le Document, pp. 185-197, 2014.

— N. Ghanmi and A. Belaïd. *Table Detection in Handwritten Chemistry Documents Using Conditional Random Fields*. International Conference on Frontiers in Handwriting Recognition, pp. 146-151, 2014.

— N. Ghanmi and A. Belaid. *Table Structure Extraction in Handwritten Chemistry Documents*. International Conference on Document Analysis and Recognition, pp. 296-300, 2015,

— N. Ghanmi and A. Belaid, *Recognition-based Approach of Numeral Extraction in Handwritten Chemistry Documents using Contextual Knowledge*. Document Analysis System, pp. 251-256, 2016.

Bibliographie

[Aho1983] J. E. Hopcroft, J. D. Ullman, and A. V. Aho. *Data structures and algorithms*, volume 175. Addison-Wesley Boston, MA, USA :, 1983.

[Anisimov1995] V. Anisimov, N. Gorski, D. Price, O. Baret, and S. Knerr. Bank check reading : Recognizing the courtesy amount. In Lecture Notes in Computer Science, editor, *Proceedings of International Computer Science Conference on Image Analysis Applications and Computer Graphics*, pages 161–172, 1995.

[Arivazhagan2007] M. Arivazhagan and S. N. Srihari H. Srinivasan. A statistical approach to handwritten line segmentation. In *Document Recognition and Retrieval, Proceedings of SPIE*, 2007.

[Awal2014] A. M. Awal and A. Belaïd. Handwritten/printed text separation using pseudolines for contextual re-labeling. In *Proceedings of International Conference on Frontiers Handwriting Recognition*, pages 29–34, 2014.

[Barlas2014] P. Barlas, C. Chatelain, and S. Adam et T. Paquet. Détection et segmentation des blocs de texte manuscrits et imprimés dans des documents complexes. In *Colloque International Francophone sur l'écrit et les documents*, pages 125–137, 2014.

[Casella1992] G. Casella and E. I. George. Explaining the gibbs sampler. *The American Statistician*, 46 :167–174, 1992.

[Chang2001] C. Chang and C. Lin. Libsvm : A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3) :27 :1–27 :27, 2011.

[Chatelain2006a] C. Chatelain. *Extraction de séquences numériques dans des documents manuscrits quelconques*. PhD thesis, LITIS - Université de Rouen, 2006.

[Chatelain2006b] C. Chatelain, L. Heutte, and T. Paquet. Segmentation-driven recognition applied to numerical field extraction from handwritten incoming mail documents. In *Proceedings of International Workshop on Document Analysis System*, pages 564–575, 2006.

[Chatelain2006c] C. Chatelain, L. Heutte, and T. Paquet. Extraction automatique de champs numériques dans des documents manuscrits. *Revue des Nouvelles Technologies de l'Information*, 6 :23–34, 2006.

[Chaudhury2009] S. Chaudhury, M. Jindal, and D. R. Sumantra. Model-guided segmentation and layout labelling of document images using a hierarchical conditional random field. In *Proceedings of International Conference on Pattern Recognition and Machine Intelligence*, pages 375–380, 2009.

[Chen2011] J. Chen and D. Lopresti. Table detection in noisy offline handwritten documents. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 399–403, 2011.

[Chen2012] J. Chen and D. Lopresti. Model-based tabular structure detection and recognition in noisy handwritten documents. In *Proceedings of International Conference on Frontiers in Handwriting Recognition*, pages 75–80, 2012.

[Chen2013] J. Chen and D. Lopresti. Ruling-based table analysis for noisy handwritten documents. In *International Workshop on Multilingual OCR*, pages 1–5, 2013.

[Congedo1995] G. Congedo, G. Dimauro, S. Impedovo, and G. Pirlo. Segmentation of numeric strings. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1038–1041, 1995.

[Couasnon2014] B. Coüasnon and A. Lemaitre. Recognition of tables and forms. In *Handbook of Document Image Processing and Recognition*, pages 647–677. 2014.

[Dan2012] C. Dan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012.

[Dey1999] S. Dey. Adding feedback to improve segmentation and recognition of handwritten numerals. Master's thesis, Massachusetts Institute of Technology, 1999.

[Diem2013] M. Diem, S. Fiel, A. Garz, M. Keglevic, F. Kleber, and R. Sablatnig. Icdar 2013 competition on handwritten digit recognition (hdrc 2013). In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1454–1459, 2013.

[Diem2014] M. Diem, S. Fiel, F. Kleber, R. Sablatnig, J. M. Saavedra, D. Contreras, J. Barrios, and L. S. Oliveira. Icfhr 2014 competition on handwritten digit string recognition in challenging datasets (hdsrc 2014). In *Proceedings of International Conference on Frontiers in Handwriting Recognition*, pages 779–784, 2014.

[Domingos1999] P. Domingos. Metacost : A general method for making classifiers cost-sensitive. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[Elnagar2003] A. Elnagar and R. Alhajj. Segmentation of connected handwritten numeral strings. *Pattern Recognition*, 36(3) :625–634, 2003.

[F-Mota2014] D. Fernández-Mota, J. Lladós, and A. Fornés. A graph-based approach for segmenting touching lines in historical handwritten documents. *International Journal on Document Analysis and Recognition*, 17(3) :293–312, 2014.

[Fan1998] K. C. Fan, L. S. Wang, and Y. t. Tu. Classification of machine-printed and handwritten texts using character block layout variance. *Pattern Recognition*, 31(9) :1275–1284, 1998.

[Favata1996] J. T. Favata and G. Srikantan. A multiple feature/resolution approach to handprinted digit and character recognition. *Interantional Journal of Imaging Systems Technology*, 7(4) :304–311, 1996.

[Fenrich1991] R. Fenrich. Segmentation of automatically located handwritten words. In *International Workshop on Frontiers in Handwriting Recognition, pp.*, pages 33–34, 1991.

[Fletcher1988] L. A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6) :910–91, 1988.

[Fujiyoshi2011] A. Fujiyoshi, K. Nakagawa, and M. Suzuki. Robust method of segmentation and recognition of chemical structure images in cheminfty. In *Proceedings of the Workshop on Graphics Recognition*, 2011.

[Gatos2005] B. Gatos, D. Danatsas, I. Pratikakis, and S. Perantonis. Automatic table detection in document images. In *Proceedings of International Conference on Advances in Pattern Recognition*, 609-618, 2005.

[Gatos2009] B. Gatos, N. Stamatopoulos, and G. Louloudis. Icdar 2009 handwriting segmentation contest. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1393–1397, 2009.

[Gatos2010] B. Gatos, N. Stamatopoulos, and G. Louloudis. Icfhr 2010 handwriting segmentation contest. In *Proceedings of International Conference on Frontiers Handwriting Recognition*, pages 737–742, 2010.

[Gattal2015] A. Gattal and Y. Chibani. Svm-based segmentation-verification of handwritten connected digits using the oriented sliding window. *International Journal of Computational Intelligence and Applications*, 14(1) :1–17, 2015.

[Gorgevik2002] D. Gorgevik and D. Cakmakov. Combining svm classifiers for handwritten digit recognition. In *Proceedings of International Conference on Pattern Recognition*, pages 529–551, 2002.

[Gorgevik2004] D. Gorgevik and D. Cakmakov. An efficient three-stage classifier for handwritten digit recognition. In *510*, pages 507–510. Proceedings of International Conference on Pattern Recognition, 2004.

[Grother1995] P. J. Grother. *NIST Special Database 19-Handprinted Forms and Characters Database*. National Institute of Standards and Technology (NIST, 1995.

[Guermeur2007] Y. Guermeur. Svm multiclasses, théorie et applications. *HDR thesis - Université de Nancy 1*, 2007.

[Gupta2007] M. R. Gupta, N. P. Jacobson, and E. K. Garcia. Ocr binarization and image preprocessing for searching historical documents. *Pattern Recognition*, 40 :389–397, 2007.

[Ha1998] T.M. Ha, M. Zimmermann, and H. Bunke. Off-line handwritten numeral string recognition by combining segmentation-based and segmentation-free methods. *Pattern Recognition*, 31(3) :257–272, 1998.

[Haji2011] M. M. Haji, T. D. Bui, and C. Y. Suen. Automatic extraction of numeric strings in unconstrained handwritten document images. In *Proccedings of Document Recognition and Retrieval, pp.*, pages 2–10, 2011.

[Hall2009] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software : An update. *SIGKDD Explor. Newsl.*, 11(1) :10–18, 2009.

[Han2005] H. Han, W. Y. Wang, and B. H. Mao. Borderline-smote : A new over-sampling method in imbalanced data sets learning. In *Proceedings of International Conference on Intelligent Computing*, pages 878–887, 2005.

[Hao2008] P. Y. Hao. Fuzzy one-class support vector machines. *Fuzzy Sets and Systems*, pages 2317–2336, 2008.

[He2008a] H. He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE transactions on knowledge and data engineering*, 21(9) :1263–1284, 2009.

[He2008b] H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn : Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of International Conference on Neural Networks*, pages 1322–1328, 2008.

[Hebert2011] D. Hebert, T. Paquet, and S. Nicolas. Continuous crf with multi-scale quantization feature functions application to structure extraction in old newspaper. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 493–497, 2011.

[Hestenes1952] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6) :409–436, 1952.

[Hirayama1995] Y. Hirayama. A method for table structure analysis using dp matching. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 583–586, 1995.

[Hu2000] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Medium-independent table detection. In *Proceedings of the Document Recognition and Retrieval (IST/SPIE Electronic Imaging)*, pages 44–55, 2000.

[Hu2001] J. Hu, R. S. Kashi, D. P. Lopresti, and G. Wilfong. Table structure recognition and its evaluation. In *Document Recognition and Retrieval*, pages 44–55, 2001.

[Huaigu2007] C. Huaigu, P. Rohit, N. Premkumar, and M. Ehry. Robust page segmentation based on smearing and error correction unifying top-down and bottom-up approaches. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 392–396, 2007.

[Jang2005] I. H. Jang, C. H. Kim, and N. C. Kim. Region analysis of business card images in pda using dct and information pixel density. In *Advanced Concepts for Intelligent Vision Systems, 7th International Conference, ACIVS 2005*, pages 243–251, 2005.

[Jiang2006] W. Jiang, Z. Sun, W. Zheng, and W. Xu. User-independent online handwritten digit recognition. In *Proceedings of International Conference on Machine Learning and Cybernetics*, pages 3359–3364, 2006.

[Jin2003] J. Jin, X. Han, and Q. Wang. Mathematical formulas extraction. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1138–1141, 2003.

[Journet2005] N. Journet, V. Eglin, J. Y. Ramel, and R. Mullot. Text/graphic labelling of ancient printed documents. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1010–1014, 2005.

[Jousse2006] F. Jousse, R. Gilleron, I. Tellier, and M. Tommasi. Champs conditionnels aléatoires pour l'annotation d'arbres. In *Proceedings of Conférence francophone sur l'apprentissage automatique*, pages 171–186, 2006.

[Kaensar2013] C. Kaensar. A comparative study on handwriting digit recognition classifier using neural network, support vector machine and k-nearest neighbor. In *Proceedings of International Conference on Computing and Information*, pages 155–163, 2013.

[Kamdi2011] S. Kamdi and R. K. Krishna. Image segmentation and region growing algorithm. *International Journal of Computer Technology an d Electronics Engineering*, 2(1) :103–107, 2011.

[Kandan2007] R. Kandan, N. K. Reddy, K. R. Arvind, and A. G. Ramakrishnan. A robust two level classification algorithm for text localization in documents. In *Proceedings of International conference on Advances in visual computing*, pages 96–105, 2007.

[Kasar2013] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet. Learning to detect tables in scanned document images using line information. In *Proceedings of International Conference on Document Ananlysis and Recognition*, pages 1185–1189, 2013.

[Kegl2009] B. Kégl and R. Busa-Fekete. Boosting products of base classifiers. In *Proceedings of International Conference on Machine Learning*, pages 497–504, 2009.

[Khaffaf2008] H. Al-Khaffaf, A. Z. Talib, and R. A. Salam. Removing salt-and-pepper noise from binary images of engineering drawings. In *Proceedings of the International Conference on Pattern Recognition*, pages 1–4, 2008.

[Khan2014] S. S. Khan and M. G. Madden. One-class classification : taxonomy of study and review of techniques. *Knowledge Engineering Review*, 29(3) :345–374, 2014.

[Kieninger1998] T. G. Kieninger. Table structure recognition based on robust block segmentation. In *Proceedings of SPIE Conference on Document Recognition*, pages 22–32, 1998.

[Kim2002] K. K. Kim, J. H. Kim, and C. Y. Suen. Segmentation-based recognition of handwritten touching pairs of digits using structural features. *Pattern Recognition Letters*, 23(1) :13–24, 2002.

[Kiryati1991] N. Kiryati, Y. Eldar, and A. M. Bruckstein. A probabilistic hough transform. *Pattern Recognition*, 24(4) :303–316, 1991.

[Koch2005] G. Koch, L. Heutte, and T. Paquet. Automatic extraction of numerical sequences in handwritten incoming mail documents. *Pattern Recognition Letters*, 26(8) :1118–1127, 2005.

[Kukar1998] M. Z. Kukar and I. Kononenko. Cost-sensitive learning with neural networks. In *Proceedings of European Conference on Artificial Intelligence*, pages 445–449, 1998.

[Kumar2007] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, and S. D. Joshi. Text extraction and document image segmentation. *IEEE Transactions on Image Processing*, 16(8) :2117–2128, 2007.

[Lafferty2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, pages 282–289, 2001.

[Lam2002] L. Lam, Q. Xu, and C. Y. Suen. Differentiation between alphabetic and numeric data using nn ensembles. In *Proceedings of International Conference on Pattern Recognition*, pages 40–43, 2002.

[Laurentini1992] A. Laurentini and P. Viada. Identifying and understanding tabular material in compound documents. In *Proceedings of International Conference on Pattern Recognition*, pages 405–409, 1992.

[Lee1991] Y. Lee. Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks. *Neural Computation Journal*, 3(3) :440–449, 1991.

[Lemaitre2007] M. Lemaitre. Approche markovienne bidimensionnelle d'analyse et de reconnaissance de documents manuscrits. *Université René Descartes, Paris 5*, 2007.

[Li2006] Yi Li, Yefeng Zheng, David Doermann, and Stefan Jaeger. A New Algorithm for Detecting Text Line in Handwritten Documents. In *Proceedings of 10th International Workshop on Frontiers in Handwriting Recognition*, pages 35–40, 2006.

[Likforman-S1995] L. Likforman-Sulem, A. Hanimyan, and C. Faure. A hough based algorithm for extracting text lines in handwritten document. In *International Conference on Document Analysis and Recognition*, pages 774–777, 1995.

[Likforman-Sulem1994] L. Likforman-Sulem and C. Faure. Extracting text lines in handwritten documents by perceptual grouping. *Advances in handwriting and drawing :a multidisciplinary approach*, pages 117–135, 1994.

[Ling2004] C. X. Ling, Qiang Yang, Jianning Wang, and Shichao Zhang. Decision trees with minimal costs. In *Proceedings of International Conference on Machine learning*, pages 69–76, 2004.

[Liu1989] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(3) :503–528, 1989.

[Liu2004] C. Liu, H. Sako, and H. Fujisawa. Effects of classifier structures and training regimes on integrated segmentation and recognition of handwritten numeral strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11) :1395–1407, 2004.

[Liu2006] X. Y. Liu, J. Wu, and Z. H. Zhou. Exploratory under sampling for class imbalance learning. In *Proceedings of International Conference on Data Mining*, pages 965–969, 2006.

[Louloudis2009] G. Louloudis, B. Gatos, and C. Halatsis I. Pratikakis. Text line and word segmentation of handwritten documents. *Pattern recognition*, 42(12) :3169–3183, 2009.

[Lu1999] Z. Lu, Z. Chi, W. C. Siu, and P. Shi. A background-thinningbased approach for separating and recognizing connected handwritten digit strings. *Pattern Recognition*, 32 :921–933, 1999.

[Lu2011] S. Lu X. Tu and Y. Lu. A handwritten bangla numeral recognition scheme based on expanded two-layer som. *International Journal of Intelligent Systems Technologies and Applications*, 10(2) :203–213, 2011.

[Malleron2009] V. Malleron, V. Eglin, s. Dord-Crouslé H. Emptoz, and P. Régnier. Text lines and snippets extraction for 19th century handwriting documents layout analysis. In *International Conference on Document Analysis and Recognition*, pages 1001–1005, 2009.

[Maloof2003] M. A. Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *Proceedings of International Conference on Machine Learning, Workshop Learning from Imbalanced Data Sets II*, 2003.

[Mandal2006] S. Mandal, S. Chowdhury, A. Das, and B. Chanda. A simple and effective table detection system from document images. *International Journal on Document Analysis and Recognition*, 8(2-3) :172–182, 2006.

[Mandal2012] R. Mandal, P. P. Roy, and U. Pal. Date field extraction in handwritten documents. In *Proceedings of International Conference on Pattern Recognition*, pages 533–536, 2012.

[Manmatha2005] R. Manmatha and J. Rothfeder. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8) :1212–1225, 2005.

[Marti2001] U. Marti and H. Bunke. On the influence of vocabulary size and language models in unconstrained handwritten text recognition. In *Proceedings ofInternational Conference on Document Analysis and Recognition*, pages 260–265, 2001.

[McCallum2000] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of International Conference on Machine Learning*, pages 591–598, 2000.

[Mease2007] D. Mease, A. J. Wyner, and A. Buja. Boosted classification trees and class probability/quantile estimation. *Journal of Machine Learning Research*, 8 :409–439, 2007.

[Mollah2009] A. F. Mollah, S. Basu, M. Nasipuri, and D. K. Basu. Text/graphics separation for business card images for mobile devices. In *International Workshop on Graphics Recognition*, pages 263–270, 2009.

[Montreuil2010] F. Montreuil, S. Nicolas, E. Grosicki, and L. Heutte. A new hierarchical handwritten document layout extraction based on conditional random field modeling. In *Proceedings of International Conference on Frontiers Handwriting Recognition*, pages 31–36, 2010.

[Neves2006] L. Neves, J. M. De Carvalho, J. Facon, and F. Bortolozzi. A new table extraction and recovery methodology with little use of previous knowledge. In *Proceedings of International Workshop on Frontiers in Handwriting Recognition*, 2006.

[Niblack1985] W. Niblack. *An Introduction to Digital Image Processing*. Strandberg Publishing Company, Birkeroed, Denmark, Denmark, 1985.

[Nicolas2004] S. Nicolas, T. Paquet, and L.Heutte. Text line segmentation in handwritten document using a production system. In *Proceedings of the International Workshop on Frontiers Handwriting Recognition*, pages 245–250, 2004.

[Nicolas2006] S. Nicolas, T. Paquet, and L. Heutte. Extraction de la structure de documents manuscrits complexes à l'aide de champs markoviens. In *Proceedings of Colloque International Francophone sur l'Ecrit et le Documen*, pages 124–129, 2006.

[Nicolas2008] S. Nicolas, T. Paquet, and L. Heutte. 2d markovian models for document structure analysis. In *Proceedings of Intenational Conference on Frontiers in Handwriting Recognition*, pages 658–663, 2008.

[Nikolaou2010] N. Nikolaou, M. Makridis, B. Gatos, Nikolaos Stamatopoulos, and Nikos Papamarkos. Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths. *Image and Vision Computing*, 28(12) :590–604, 2010.

[Nissen2003] S. Nissen. Implementation of a fast artificial neural network library (fann). Technical report, Department of Computer Science University of Copenhagen (DIKU), 2003.

[Nocedal2006] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[Oliveira2000] L. S. Oliveira, E. Lethelier, F. Bortolozzi, and R. Sabourin. A new segmentation approach for handwritten digits. In *Proceedings of International Conference on Pattern Recognition*, pages 323–326, 2000.

[Oliveira2002] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Automatic recognition of handwritten numerical strings : A recognition and verification strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11) :1438–1454, 2002.

[Osher2003] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*. Applied mathematical science. Springer, 2003.

[Otsu1979] N. Otsu. A threshold selection method from grey scale histogram vol. 1. *IEEE Transactions on Systems, Man, and Cybernetics*, 1 :62–66, 1979.

[Ouwayed2010] N. Ouwayed, A. Belaïd, and F. Auger. General text line extraction approach based on locally orientation estimation. In *Proceedings of Document Recognition and Retrievel*, pages 1–10, 2010.

[Ouyang2007] T. Y. Ouyang and R. Davis. Recognition of hand drawn chemical diagrams. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, pages 846–851, 2007.

[Pal2002] U. Pal, A. Belaïd, and Ch. Choisy. Touching numeral segmentation using water reservoir concept. *Pattern Recognition Letters*, 24 :261–272, 2002.

[Papavassiliou2010] V. Papavassiliou, T. Stafylakis, V. Katsouros, and G. Carayannis. Handwritten document image segmentation into text lines and words. *Pattern Recognition*, 43(1) :369–377, 2010.

[Pavlidis1992] T. Pavlidis and J. Zhou. Page segmentation and classification. *Graphical Model and Image Processing*, 54(6) :484–496, 1992.

[Peterman1997] C. Peterman, Ch. Chang, and H. Alam. A system for table understanding. In *Proccedings of Conference on Document Image Understanding Technology*, pages 55–62, 1997.

[Pinto2003] D. Pinto, A. McCallum, X. Wei, and W. B. Croft. Table extraction using conditional random fields. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 235–242, 2003.

[Pitrelli2003] J. Pitrelli and M. Perrone. Confidence-scoring post-processing for off-line handwritten-character recognition verification. In *Proceedings of International Conference on Document Analysis and recognition*, pages 278–282, 2003.

[Ranzato2007] M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Proceedings of Computer Vision and Pattern Recognition Conference*, pages 1–8, 2007.

[Razafindramanana2013] O. Razafindramanana, F. Rayar, and G. Venturini. Alpha*-approximated delaunay triangulation based descriptors for handwritten character recognition. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 440–444, 2013.

[Roy2010] P. Roy, U. Pal, and J. Llados. Touching text character localization in graphical documents using sift. In *Graphics Recognition. Achievements, Challenges, and Evolution, 8th International Workshop, GREC*, pages 199–211, 2010.

[Roy2012] P. P. Roy, U. Pal, J. Lladós, and M. Delalandre. Touching text character segmentation in graphical documents using dynamic programming. *Pattern Recognition*, 45(5) :1972–1983, 2012.

[Saabni2011] R. Saabni and J. El-Sana. Language-independent text lines extraction using seam carving. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 563–568, 2011.

[Sadawi2013] N. Sadawi. *A rule-based approach for recognition of chemical structure diagrams*. PhD thesis, University of Birmingham, 2013.

[Sadri2004] J. Sadri, C. Y. Suen, and T. D. Bui. Automatic segmentation of unconstrained handwritten numeral strings. In *International Workshop on Frontiers in Handwriting Recognition, pp.*, pages 317–322, 2004.

[Sadri2007] J. Sadri, C. Y. Suen, and T. d. Bui. A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings. *Pattern Recognition*, 40(3) :898–919, 2007.

[Saha2010] S. Saha, S. Basu, M. Nasipuri, and D k. Basu. A hough transform based technique for text segmentation. *Journal of computing*, 2(2) :134–141, 2010.

[Sarkar2011] R. Sarkar, S. Moulik, N. Das, S. Basu, M. Nasipuri, and M. Kundu. Suppression of non-text components in handwritten document images. In *Proceedings of International Conference on Image Information Processing*, pages 1–7, 2011.

[Sauvola2000] J. Sauvola and M. Pietikainen. Adaptive document image binarization. *Pattern Recognition*, 33(2) :225–236, 2000.

[Sayre1973] K. Sayre. Machine recognition of handwritten words : A project report. *Pattern Recognition*, 5(3) :213–228, 1973.

[Scholkopf1999] B. Scholkopf, R. C. Williamson, A. Smola, and J. S. Taylor. *SV estimation of a distribution's support*. Advances in Neural Information Processing Systems, 1999.

[Seymore1999] K. Seymore, A. McCallum, and R. Rosenfeld. Learning hidden markov model structure for information extraction. In *AAAI 99 Workshop on Machine Learning for Information Extraction*, 1999.

[Shafait2008] F. Shafait, D. Keysers, and T. Breuel. Performance evaluation and benchmarking of six-page segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6) :941–954, 2008.

[Shafait2010] F. Shafait and R. Smith. Table detection in heterogeneous documents. In *Proceedings of the International Workshop on Document Analysis Systems*, pages 65–72, 2010.

[Shetty2007] S. Shetty, H. Srinivasan, M. Beal, and S. N. Srihari. Segmentation and labeling of documents using conditional random fields. In *Proceedings of Document Recognition and Retrieval*, pages 1–11, 2007.

[Shi1997] Z. Shi and V. Govindaraju. Segmentation and recognition of connected handwritten numeral strings. *Pattern Recognition*, 30(9) :1501–1504, 1997.

[Shi2004] Z. Shi and V. Govindaraju. Line separation for complex document images using fuzzy runlength. In *International Workshop on Document Image Analysis for Libraries*, pages 23–24, 2004.

[Shi2009] Z. Shi, S. Setlur, and V. Govindaraju. A steerable directional local profile technique for extraction of handwritten arabic text lines. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 176–180, 2009.

[Simon1997] A. Simon, J. c. Pret, and A. P. Johnson. A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis anad Machine Intelligence*, 19(3) :273–277, 1997.

[Stamatopoulos2013] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei. Icdar 2013 handwriting segmentation contest. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 1402–1406, 2013.

[Takasu2003] A. Takasu. Bibliographic attribute extraction from erroneous references based on a statistical model. In *Proceedings of ACM/IEEE-CS joint conference on Digital libraries*, pages 49–60, 2003.

[Tax2001] D. M. J. Tax. *One-class Classification*. PhD thesis, Delft University of Technology, 2001.

[Tombre2002] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch. Text/graphics separation revisited. In *Proceedings of International Workshop on Document Analysis Systems, Lecture Notes in Computer Science*, volume 2423, pages 200–211, 2002.

[Tony2005] L. Tony, H. Pengwei, and U. L. Sang. Efficient coding of computer generated compound images. In *Proceedings of International Conference on Image Processing*, pages 561–564, 2005.

[Tripathy2004] N. Tripathy and U. Pal. Handwriting segmentation of unconstrained oriya text. In *International Workshop on Frontiers in Handwriting Recognition*, pages 306–311, 2004.

[Tsuruoka2001] S. Tsuruoka, T. Tanaka, T. Yoshikawa, T. Shinogi, and K. Takao. Region segmentation for table image with unknown complex structure. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 709–713, 2001.

[Wang2000] X. Wang, V. Govindaraju, and S. Srihari. Holistic recognition of handwritten character pairs. *Pattern Recognition*, 33(12) :1967–1973, 2000.

[Wolf2002] C. Wolf, J m Jolion, and F. Chassaing. Text localization, enhancement and binarization. In *Proceedings of the International Conference on Pattern Recognition*, pages 1037–1040, 2002.

[Wu2014] Y. Wu, F. Yin, and C. Liu. Evaluation of geometric context models for handwritten numeral string recognition. In *Proceedings of International Conference on Frontiers in Handwriting Recognition*, pages 193–198, 2014.

[Xiao-yan2007] T. Xiao-yan and J. I. Hong-bing. A modified psvm and its application to unbalanced data classification. In *Proceedings of International Conference on Natural Computation*, 2007.

[Zadrozny2003] B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of International Conference on Data Mining*, pages 435–442, 2003.

[Zhang1984] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the Association for Computing Machinery*, 27(3) :236–239, 1984.

[Zhang2003] J. Zhang and I. Mani. Knn approach to unbalanced data distributions : A case study involving information extraction. In *Proceedings of International Conference on Machine Learning, Workshop Learning from Imbalanced Data Sets II*, 2003.

[Zhao2010] M. Zhao, S. Li, and J. Kwok. Text detection in images using sparse representation with discriminative dictionaries. *Image and Vision Computing*, 28(12) :1590–1599, 2010.

[Zheng2004] Y. Zheng, H. Li, and D. Doermann. Machine printed text and handwriting identification in noisy document images. *IEEE Transactions on on Pattern Analysis and Machine Intelligence*, 26 :337–353, 2004.

[Zhou2005] J. Zhou and C. Y. Suen. Unconstrained numeral pair recognition using enhanced error correcting output coding : A holistic approach. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 484–488, 2005.

[Zuyev1997] K. Zuyev. Table image segmentation. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 705–708, 1997.