

533

17004, 533

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 - Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique



Mémoire de fin d'études Master

16/921

Filière : Informatique

Option : Ingénierie de médias

Thème :

---

---

## Reconnaissance de la structure logique des pages du journal

---

---

Encadré Par :

Dr. Abderrahmane Kefali

Présenté par :

Hassina Bouressace

Amina Zebiri

Juin 2016

# Abstract

The domain of documents analysis and recognition relates to automatic document identification methods. Its goal is to pass from a simple image to a structured set of information exploitable by machine. After the revolution of the handwriting recognition over the past two decades, document analysis and recognition are moving towards of the logical structure recognition of documents. This latter is a high-level representation in the form of a structured document of the components contained in the document image. The goal of extracting the logical structure of a document is to understand the hierarchical organization of its components and the relationships between them.

This work proposes a system to recognize the logical structure (hierarchical organization) of Arabic newspapers pages. Those latter are characterized by a rich and variable structure. They may contain several articles composed each one of titles, figures, author's names, figure captions, etc. However, logical structure recognition of a newspaper page is preceded by the extraction of its physical structure. This extraction is done in our system using a mixed method which is essentially based on the algorithm RLSA (Run Length Smearing / Smoothing Algorithm), projections profile analysis, and connected components labeling. Logical structure extraction is then performed based on certain rules of sizes and positions of the physical elements extracted earlier, and also on an a priori knowledge of certain properties of logical entities (titles, figures, authors, captions, etc.). Finally, the hierarchical organization of the document is represented as an XML file generated automatically. To evaluate the performances of our system, we tested it on a set of images and the results are encouraging.

**Keywords:** document analysis, document recognition, physical structure, logical structure, document image.

# Résumé

Le domaine de l'analyse et la reconnaissance de documents se concerne aux méthodes d'identification automatique des documents par ordinateur. Son but est de faire passer d'une image brute à un ensemble d'informations structurées exploitables par la machine. Après la révolution de la reconnaissance de l'écriture au cours des deux dernières décennies, l'analyse et la reconnaissance de documents s'orientent vers la reconnaissance de la structure logique de documents. Cette dernière est une représentation de haut niveau sous la forme d'un document structuré des composants contenus dans l'image de document. Le but de l'extraction de la structure logique d'un document est de comprendre l'organisation hiérarchique de ses éléments et les relations entre eux.

Ce mémoire propose un système pour reconnaître la structure logique (l'organisation hiérarchique) des pages de journaux arabes. Ces dernières sont caractérisées par une structuration riche et variable. Elles peuvent contenir plusieurs articles composé chacun par des titres, figures, nom de l'auteur, légendes des figures...etc. Cependant, la reconnaissance de la structure logique d'une page de journal est précédée par l'extraction de sa structure physique. Cette extraction est effectuée dans notre système en utilisant une méthode mixte qui repose essentiellement sur l'algorithme de lissage RLSA (Run Length Smearing/Smoothing Algorithm), l'analyse des profils de projections, et l'étiquetage des composantes connexes. L'extraction de la structure logique est ensuite réalisée en se basant sur certaines règles de la taille et la position des éléments physiques extraits précédemment, et aussi sur la connaissance à priori de certaines propriétés des entités logiques (titres, figures, auteurs, légendes ...etc.). Finalement, l'organisation hiérarchique du document est représentée sous forme d'un fichier XML généré automatiquement. Afin d'évaluer les performances de notre système, nous l'avons testé sur un ensemble d'images et les résultats obtenus sont encourageants.

**Mots clés :** analyse de document, reconnaissance de document, structure physique, structure logique, image de document.

## Remerciement

*Nous rendons grâce au bon Dieu de nous avoir donné le courage et la force pour mener à bout ce modeste travail. Nous remercions toute personne ayant participé à l'élaboration de ce présent mémoire. Mention spéciale à notre encadreur Dr. Abderrahmane Kefali qui suit fidèlement notre travail. Nous le remercions pour son encadrement, son soutien, et pour la confiance qu'il nous témoignée en notre confiant ce travail.*

*Nos remerciements les vifs s'adressent à monsieur le président ainsi qu'au membre de jury d'avoir accepté d'examiner et d'évaluer ce modeste travail. Nous adressons aussi nos sincères remerciements à tous les enseignants et les travailleurs du département d'Informatique de l'université 08 Mai 1945 - Guelma.*

*ET nous ne saurons terminer sans laisser un grand merci à tous nos camarades étudiants de la promotion du département informatique. En bref, nous remercions tous ceux qui nous ont aidé de près ou de loin que ce soit par leur soutien moral ou physique pour la réalisation de ce projet.*



## *Dédicace*

*Nous dédions ce travail à :*

*Nos très chers parents,*

*A toutes nos familles,*

*A toutes nos amies,*

*A tous ceux qui nous ont soutenus et encouragés*

*Pendant nos études de près ou de loin,*

*A tous nos chers.*

# Table de matière

Abstract .....	i
Résumé .....	ii
Remerciement.....	iii
Dédicace .....	iv
Table de figures.....	4
Liste de tableaux.....	5
Liste des Abréviations et Acronymes.....	6
<b>Introduction générale.....</b>	<b>7</b>
<b>Chapitre 1. Documents et structures .....</b>	<b>10</b>
1. Introduction .....	11
2. Les documents structurés .....	11
2.1. Qu'est-ce qu'un document ? .....	11
2.2. Notion de structure.....	11
2.3. Différents types de documents.....	13
3. Reconnaissance de document.....	14
3.1. Présentation.....	14
3.2. Etapes de reconnaissance de document .....	15
4. Analyse de structures de documents .....	15
4.1. Extraction de la structure physique.....	15
4.1.1. Approche ascendante (Bottom-up) .....	16
4.1.2. Approche descendante (Top-down) .....	17
4.1.3. Approche mixte (Hybrid).....	19
4.2. Extraction de la structure logique .....	19
4.2.1. Approche structurelle .....	20
4.2.2. Approche de type « Intelligence Artificielle ».....	20
4.2.3. Approche probabiliste .....	20
5. Représentation des structures .....	20
5.1. Représentation des structures physiques.....	20
5.2. Représentation des structures logiques .....	21
5.3. Structures génériques.....	22
6. Conclusion.....	22
<b>Chapitre 2. Etat de l'art sur l'analyse et la reconnaissance de documents structurés ....</b>	<b>23</b>
1. Introduction .....	24
2. Documents à structure simple et documents à structure complexe.....	24
2.1. Les documents à structure simple .....	24
2.2. Les documents à structure complexe .....	25
3. Méthodes existantes de reconnaissance de documents .....	26
3.1. Reconnaissance des documents à structure simple.....	26
3.1.1. Reconnaissance de la structure physique .....	27

3.1.1.1. Méthodes ascendantes .....	27
3.1.1.2. Méthodes descendantes .....	28
3.1.2. Reconnaissance de la structure logique.....	29
3.1.2.1. Méthodes structurelles .....	29
3.1.2.2. Méthodes de type Intelligence Artificielle (à base de règles) .....	29
3.1.2.3. Méthodes probabilistes .....	30
3.2. Reconnaissance des documents à structure complexe et stable (les formulaires) ...	30
3.2.1. Reconnaissance de la structure physique .....	30
3.2.1.1. Méthodes utilisant les filets .....	31
3.2.1.2. Méthodes utilisant les intersections .....	31
3.2.2. Reconnaissance de la structure logique.....	31
3.2.2.1. Méthodes basées sur la représentation en arbres hiérarchiques.....	32
3.2.2.2. Méthodes basées sur une grammaire de graphe .....	32
3.3. Reconnaissance des documents à structure complexe et variables.....	32
3.3.1. Reconnaissance de la structure physique .....	33
3.3.1.1. Méthodes ascendantes .....	33
3.3.1.2. Méthodes descendantes .....	34
3.3.1.3. Méthodes mixtes.....	34
3.3.2. Reconnaissance de la structure logique.....	36
3.3.2.1. Méthodes structurelles .....	37
3.3.2.2. Méthodes de type Intelligence Artificielle (à base de règles) .....	37
4. Conclusion.....	38
<b>Chapitre 3. Conception</b> .....	39
1. Introduction .....	40
2. L'entrée / Sortie.....	40
3. Caractéristiques des pages de journaux utilisées.....	40
4. Exposé de la démarche .....	42
4.1. Extraction de la structure physique.....	43
4.1.1. Prétraitement .....	43
4.1.1.1. Transformation en niveau de gris.....	44
4.1.1.2. Seuillage .....	44
4.1.2. Segmentation.....	46
4.1.2.1. Etiquetage des composantes connexes.....	46
4.1.2.2. Détection et élimination des graphiques .....	47
4.1.2.3. Segmentation du texte en articles.....	49
4.1.2.4. Segmentation des articles en blocs.....	51
4.1.2.5. Segmentation des blocs en lignes.....	52
4.1.2.6. Segmentation des lignes en mots .....	53
4.2. Extraction de la structure logique .....	54



4.2.1. Etiquetage.....	54
4.2.1.1. Etiquetage de l'entête et le pied de page.....	54
4.2.1.2. Etiquetage des articles.....	54
4.2.1.3. Etiquetage des blocs.....	54
4.2.1.4. Etiquetage des autres entités logiques.....	55
4.2.2. Génération d'un fichier XML .....	55
4.3. Génération d'un arbre de composants de la page .....	56
5. Conclusion.....	56
<b>Chapitre 4. Implémentation et résultats .....</b>	<b>57</b>
1. Introduction .....	58
2. Environnement de développement .....	58
2.1. Environnement matériel.....	58
2.2. Environnement logiciel .....	58
3. Corpus de documents utilisé .....	59
4. Architecture et fonctionnalités du système .....	60
4.1. Description de l'application .....	60
4.2. Scénario d'utilisation complet .....	62
4.2.1. Chargement d'une image .....	62
4.2.2. Prétraitement de l'image chargée.....	62
4.2.2.1. Transformation en niveaux de gris .....	62
4.2.2.2. Seuillage adaptatif .....	63
4.2.3. Extraction de la structure physique .....	63
4.2.3.1. Etiquetage des composants connexes.....	64
4.2.3.2. Détection des éléments graphiques .....	64
4.2.3.3. Élimination des éléments graphiques .....	65
4.2.3.4. Détection et élimination des titres .....	65
4.2.3.5. Lissage par RLSA et étiquetage des composantes connexes .....	65
4.2.3.6. Segmentation en articles.....	66
4.2.3.7. Extraction des blocs, des lignes et des mots.....	66
4.2.4. Extraction de la structure logique .....	67
4.2.4.1. Étiquetage des entités physiques .....	67
4.2.4.2. Génération d'un fichier XML.....	68
4.2.5. Génération d'un arbre de composants de la page.....	68
5. Expérimentations et résultats .....	69
6. Conclusion.....	70
<b>Conclusion générale et perspectives .....</b>	<b>72</b>
<b>Bibliographie.....</b>	<b>75</b>



## Table de figures

<b>Figure 1.1</b> : Structure physique d'une page de journal.....	12
<b>Figure 1.2</b> : Structure logique d'une page de journal. ....	13
<b>Figure 1.3</b> : Schéma de reconnaissance d'un document .....	15
<b>Figure 1.4</b> : Segmentation RLSA .....	17
<b>Figure 1.5</b> : Découpage XY d'un document.....	18
<b>Figure 1.6</b> : Représentation d'une structure physique .....	21
<b>Figure 1.7</b> : Représentation d'une structure logique .....	21
<b>Figure 1.8</b> : Structures génériques sous forme de DTD. ....	22
<b>Figure 2.1</b> : Exemples de documents imprimés ayant une structure simple .....	25
<b>Figure 2.2</b> : Exemples de documents à structure complexe (composite) .....	26
<b>Figure 2.3</b> : Diversité des mises en pages d'un journal à l'autre.....	33
<b>Figure 3.1</b> : L'entrée / Sortie de notre application .....	40
<b>Figure 3.2</b> : Composants d'une page du journal Echorouk. ....	42
<b>Figure 3.3</b> : Schéma illustre notre processus d'analyse et de reconnaissance de documents..	43
<b>Figure 3.4</b> : Binarisation d'une page de journal.....	46
<b>Figure 3.5</b> : Etiquetage des composantes connexes. ....	47
<b>Figure 3.6</b> : Détection et élimination des graphiques.....	48
<b>Figure 3.7</b> : (a) Elimination des titres, (b) étiquetage des composantes connexes après un lissage RLSA vertical. ....	50
<b>Figure 3.8</b> : Segmentation de la page. ....	51
<b>Figure 3.9</b> : Segmentation des blocs en lignes .....	53
<b>Figure 4.1</b> : Exemples des pages du journal Echorouk utilisées dans notre corpus de test....	59
<b>Figure 4.2</b> : Interface d'accueil de notre application.....	60
<b>Figure 4.3</b> : Interface principale de notre application. ....	60
<b>Figure 4.4</b> : Interface d'affichage du fichier XML.....	61
<b>Figure 4.5</b> : Les modules principaux de l'application. ....	61
<b>Figure 4.6</b> : Chargement et transformation de l'image en niveaux de gris. ....	63
<b>Figure 4.7</b> : Binarisation par seuillage adaptatif.....	63
<b>Figure 4.8</b> : Etiquetage des composantes connexes. ....	64
<b>Figure 4.9</b> : Détection des éléments graphiques.....	64
<b>Figure 4.10</b> : Elimination des éléments graphiques détectés auparavant. ....	65
<b>Figure 4.11</b> : Elimination des titres .....	65
<b>Figure 4.12</b> : Lissage par RLSA et étiquetage des composantes connexes.....	66
<b>Figure 4.13</b> : Segmentation du texte en articles .....	66
<b>Figure 4.14</b> : Extraction des blocs, des lignes et des mots de chaque article .....	67
<b>Figure 4.15</b> : Etiquetage logique des entités physiques extraites précédemment. ....	67
<b>Figure 4.16</b> : Génération du Fichier XML .....	68
<b>Figure 4.17</b> : Génération d'un outil de navigation à l'intérieur de la page de journal. ....	68

## Liste des tableaux

<b>Tableau 4.1</b> : Caractéristiques du matériel utilisé.....	58
<b>Tableau 4.2</b> : Etiquettes détectées manuellement et automatiquement de l'image de la figure 4.1.b.....	69
<b>Tableau 4.3</b> : Résultats du test. ....	70

## Liste des Abréviations et Acronymes

<b>OCR</b>	Optical Character Recognition
<b>DAL</b>	Document Architecture Language
<b>DAN</b>	Document Analysis on Network
<b>DTD</b>	Document Type Definition
<b>EDI</b>	Environnement de Développement Intégré
<b>EPL</b>	Eclipse Public License
<b>GNU</b>	General Public License
<b>ICDAR</b>	International Conference on Document Analysis and Recognition
<b>PC</b>	Personal Computer
<b>RLSA</b>	Run Length Smearing/Smoothing Algorithm
<b>XML</b>	eXtensible Markup Language

---

---

# Introduction générale

---

---



De nos jours, la plupart des informations sont encore enregistrées, stockées et distribuées en format papier. L'utilisation généralisée des ordinateurs pour l'édition de documents, avec l'introduction des PC (Personal Computer) et des logiciels de traitement de textes, et face à l'évolution de l'informatique et la quantité importante d'informations, le document papier ne reste pas un support primordial car il devient difficile pour pouvoir les transmettre.

Cependant, le document électronique est devenu un vecteur inévitable pour l'échange d'idées et d'informations lors d'un processus de communication entre ou hors organisations. Le grand nombre de documents existants et la production de nouveaux documents chaque année soulèvent des questions importantes dans la recherche d'un traitement efficace et un stockage de ces derniers et les informations qu'ils contiennent. Cela a conduit à l'apparence de nouveaux domaines de recherche tel que l'analyse et la compréhension des documents par ordinateur et la reconnaissance des éléments qu'ils contiennent : les images, les mots, les caractères, les blocs manuscrits,...etc. Ces éléments sont organisés en structure qui porte des informations sur le contenu de document afin simplifier l'étape de lecture et d'interprétation.

Pour concevoir un système permettant la reconnaissance, l'indexation, la recherche et la classification automatique ou tout autre système visant à comprendre les documents de nature imprimée et manuscrite, il faut d'abord reconnaître la structure des documents. Après la révolution de la reconnaissance optique de caractères (OCR) et la reconnaissance de l'écriture au cours des deux dernières décennies, l'analyse et la reconnaissance de documents s'orientent vers une autre application très intéressante aussi qui est la reconnaissance des structures de documents. Cette dernière vise à la représentation du document sous une forme structurée en suivant un ensemble de techniques informatiques pour faciliter la réutilisation et la récupération.

Nous nous intéressons dans ce mémoire à reconnaître la structure logique ou l'organisation hiérarchique d'une catégorie des documents à structure complexe à savoir les pages de journal. Les pages de journal sont caractérisées par une structuration riche et variable. Elles peuvent contenir plusieurs articles composé chacun par des titres, figures, nom de l'auteur, légendes des figures...etc. La reconnaissance de la structure logique d'une page de journal est précédée par le découpage de la page en articles, les articles en blocs, les blocs en paragraphes, les paragraphes en lignes, et les lignes en mots, ou en d'autres termes par une analyse de la structure physique de la page.

Ce mémoire est structuré en 4 chapitres comme suit.

Le chapitre 1 présente une vue d'ensemble sur les documents et les leurs structure définissons d'abord le document et ses différentes structures, puis nous expliquons les bases de classification des différents types de documents. Ensuite nous donnons une présentation générale du domaine de reconnaissance de documents tout en exposons ses différentes étapes. Après nous adressons les différentes approches d'analyse des structures physiques et logiques proposées dans la littérature, et nous décrivons les déférentes formes de représentation des structures.

Le chapitre 2 présente un état de l'art sur le domaine de l'analyse et la reconnaissance des structures de document. D'abord nous distinguons les documents structurés à base de leur complexité : simple ou complexe. Puis, nous exposons les principales méthodes existantes dans la littérature pour la reconnaissance de la structure physique et logique des documents à structure simple, les documents à structure complexe et stable, et les documents à structure complexe et variable, tout en présentant les caractéristiques de chaque type de documents structurés.

Le chapitre 3 présente une conception générale de notre proposition. Nous décrivons d'abord les caractéristiques des pages de journal Arabe qui sont l'objet de notre étude. Après, nous donnons une suite détaillée des différentes étapes suivies pour arriver à un système permettant l'analyse et la reconnaissance des pages de journal.

Le chapitre 4 est consacré à l'implémentation et l'expérimentation de notre système. Nous décrivons les tests effectués et les résultats obtenus.

---

---

# Chapitre 1.

## Documents et structures

---

---



## 1. Introduction

La typologie des documents s'enrichit chaque jour et introduit de nouvelles façons de les exploiter. L'analyse et la reconnaissance de document c'est la première étape qui conduit à l'extraction de la structure physique et logique du document et elle devient une étape importante dans toute application de reconnaissance de l'écriture, indexation, recherche, et classification automatique de documents.

L'objectif de l'analyse et la reconnaissance de document est d'extraire des parties spécifiques d'après un support électronique d'un document numérisé qui est à l'origine une image, afin de faciliter la recherche et l'indexation, et la classification automatique de documents. Pour atteindre ce but nous devons connaître l'ensemble des documents étudiés pour analyser leurs structures, c'est-à-dire les différents types de documents et leurs familles de classification. L'analyse et la reconnaissance de document regroupe l'ensemble des techniques de segmentation guidées par des règles structurelles, afin de trouver deux types de structures, la structure physique, et la structure logique.

Le présent chapitre présente le domaine de l'analyse des structures de documents. Son but est de décrire les étapes nécessaires à l'analyse et la reconnaissance des structures de documents pour les préparer à l'étape suivante de traitement.

## 2. Les documents structurés

### 2.1. Qu'est-ce qu'un document ?

Un document peut être décrit comme une collection d'objets comportant des objets de plus haut niveau composés d'objets plus primitifs. Les relations entre ces objets représentent les relations logiques entre les composants du document. Par exemple un livre est divisé en chapitres, chaque chapitre en sections, sous-sections, paragraphes, etc. Une telle organisation documentaire est appelée représentation de **document structuré**. (Traduit depuis la préface de Structured document [AF89])

### 2.2. Notion de structure

D'après LE PETIT ROBERT : « une structure décrit la manière dont un édifice est construit ; en architecture par exemple, elle désigne l'agencement des parties d'un bâtiment ». Pour un document, le terme « structure » désigne l'organisation du document en blocs, niveaux, etc. et leurs relations.

On peut cependant distinguer deux types de structures dans un document : la structure *physique* et la structure *logique*.



### 2.2.1. La structure physique

Une image de document est composée de régions telles que des blocs de texte, des lignes, des mots, des tableaux, des figures et du fond. Ces objets présentent l'apparence du document. La structure physique d'un document décrit la mise en page du document, les différentes zones de texte, leur agencement les unes par rapport aux autres, ainsi que l'ensemble de leurs caractéristiques typographiques (police, couleur, gras, italique, ...etc.)

La figure suivante présente un exemple d'une structure physique d'une image de document (une page de journal).



(a) Page de journal

(b) structure physique

Figure 1.1 : Structure physique d'une page de journal. [Jou07]

### 2.2.2. La structure logique

La structure logique d'un document est une désignation au contenu sémantique du document et ainsi la correspondance entre les régions physiques et leur fonction c'est-à-dire les attribuer des descriptions logiques. C'est l'étiquetage qui consiste à attribuer aux différentes régions physiques une étiquette peut être un titre, un résumé, un sous-titre, un paragraphe, un en-tête, un pied de page, un numéro de page,... etc. Cet étiquetage est le but des systèmes d'analyse et de reconnaissance de documents.

La figure suivante présente un exemple d'une structure logique d'une image de document (une page de journal).



(a) Page de journal

(b) Structure logique

Figure 1.2 : Structure logique d'une page de journal. [Jou07]

### 2.3. Différents types de documents

Il n'existe pas une méthode générique permettant le traitement et l'analyse de tous les types de documents existants. Les traitements sont plutôt adaptés à une famille de documents. A cet effet les documents peuvent être regroupés en classes selon divers critères :

#### 2.3.1. Classification basée sur le contenu

Dans [Nag00], Nagy a proposé une classification des documents en documents structurés et en documents graphiques suivant la prédominance des zones textuelles ou graphiques. Les documents structurés sont composés de colonnes, de paragraphes, de lignes de texte, de mots, de caractères, de figures et de schémas. Les dessins techniques, les cartes géographiques, les plans, les partitions musicales, les diagrammes, les schémas électroniques, les organigrammes sont des exemples de documents graphiques.

#### 2.3.2. Classification basée sur les usages

Une classification des documents selon leurs usages, leurs natures et leurs contenus a été proposée par Dupoirier [Dup94]. Dans cette classification on distingue : les documents de bureaux, la documentation technique, la presse et les ouvrages de labeur, les ouvrages de savoir, les systèmes documentaires, les documents de gestion de configuration et les plans.



### 2.3.3. Classification selon la structure

Selon Karim Hadjar [Had06], les documents peuvent être répartis selon leur structuration en trois groupes :

- **Documents à Structures linéaires** : ce sont des documents à structures simples telles que les Romans par exemple.
- **Documents à Structures hiérarchiques simples** : dites aussi des documents à structures complexes et stables. Ils ont des structures arborescentes et des organisations en chapitres, sections, titres... comme les livres, les articles scientifiques, les formulaires, ...
- **Documents à Structures complexes** : ces documents possèdent une typographie riche et variable (journaux, magazines,...)

## 3. Reconnaissance de document

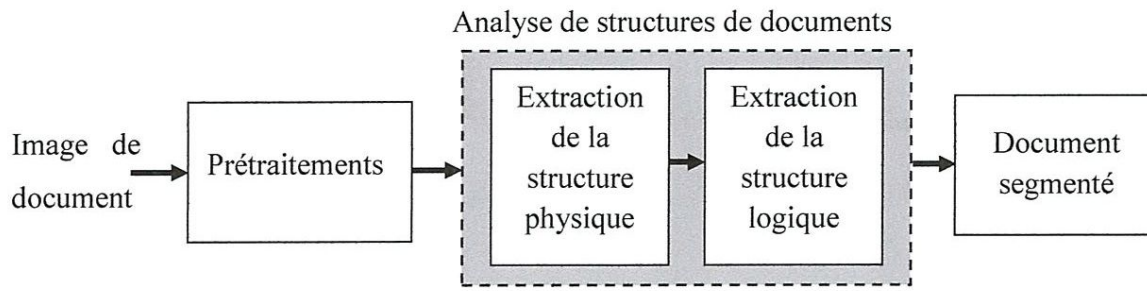
### 3.1. Présentation

La reconnaissance de document consiste à convertir un document papier en document électronique en se basant sur l'analyse et l'interprétation du document [KK10]. C'est alors le processus inverse de la production de document. Cette discipline regroupe un ensemble de techniques informatiques dont le but est de reconstituer le contenu d'un document à partir de son image.

Pendant longtemps, les efforts dans le domaine de la reconnaissance de document se sont concentrés sur la problématique de la reconnaissance optique de caractères (OCR), ainsi que sur quelques domaines très spécifiques exploitant l'OCR, comme la lecture de chèques, le tri de courriers postaux, et l'acquisition de formulaires. Cependant, le principal intérêt d'un document ne se trouve pas uniquement au niveau caractères, mais également dans d'autres informations de haut niveau sur le document comme la taille et le type de police utilisée, la couleur, et surtout l'organisation hiérarchique de ses entités (sa structure). Cela fait sortir la reconnaissance de document de la simple OCR à des objectifs beaucoup plus larges pour répondre aux besoins actuels. La reconnaissance de document s'oriente aujourd'hui vers l'interprétation complète des documents pour l'indexation ou la réédition du fait de l'accroissement constant de la puissance de calcul des machines et de l'amélioration des performances des modules de reconnaissance de caractères.

### 3.2. Etapes de reconnaissance de document

Comme nous avons déjà dit, la reconnaissance sert à construire une version électronique exploitable à partir d'un document papier. Ainsi, cette tâche est effectuée en plusieurs étapes : le prétraitement, la segmentation de la page (extraction de la structure physique), l'étiquetage logique des segments (extraction de la structure logique). Ces étapes sont illustrées par la figure suivante :



*Figure 1.3: Schéma de reconnaissance d'un document. [Mon11]*

Le prétraitement c'est un ensemble de techniques et opérations qu'on applique sur l'image numérisée avant toute autre traitement, afin de réduire le bruit superposé aux données, d'éliminer les défauts, diminuer les dégradations, et d'améliorer ainsi la qualité de l'image bruitée. Les prétraitements peuvent être effectués à des fins de visualisation, et/ou en vue de préparer le terrain aux traitements ultérieurs. Le résultat des prétraitements est une image nettoyée de bruit et qui serait la meilleure possible pour les traitements suivants dans la chaîne.

## 4. Analyse de structures de documents

L'analyse de structures de documents est une étape essentielle dans processus de reconnaissance de documents. Elle a comme but d'analyser le document afin d'en extraire ses différentes structures : physique et logique. Dans cet objectif, l'analyse de structures de documents est une étape indispensable permettant de générer une représentation structurée du document.

Le processus d'extraction de la structure physique et de la structure logique de documents consiste à décomposer une image de document en régions et à comprendre leur fonction et leurs relations dans le document. Notons qu'il est parfois nécessaire de reconsidérer la structure physique à partir des résultats obtenus pour la structure logique.

### 4.1. Extraction de la structure physique

Une image de document est composée de différentes entités physiques ou régions telles que des blocs de texte, des lignes, des mots, des chiffres, des tableaux ainsi qu'un fond.



L'extraction de la structure physique ou l'analyse de document consiste à segmenter l'image de document en composantes homogènes et de classifier chaque zone en texte, image, graphique, etc. Les méthodes classiques d'extraction de la structure physique sont généralement applicables sur des documents imprimés à prédominance textuelle et présentant une structure simple. Pour des documents plus complexes, un modèle de document est utilisé. Ce modèle peut-être introduit manuellement ou construit par apprentissage incrémentale à l'aide d'une intervention d'un expert.

Les méthodes classiques d'extraction des structures physiques de documents sont souvent réparties en trois grandes classes ou approches : *approche descendante*, *approche ascendante* et *approche mixte*.

#### 4.1.1. Approche ascendante (Bottom-up)

Le principe des méthodes ascendantes est d'extraire des primitives locales à partir de l'image analysée, par exemple les composantes connexes, indépendantes des objets à reconnaître, puis de les assembler en introduisant de la connaissance. Ceci aura pour but d'une part de les valider et d'autre part de tenter de reconstruire petit à petit les objets et donc de les reconnaître. Dans le cas des composantes de connexes, on fusionne les morceaux jusqu'à l'assemblage complet de la page du document.

Les problèmes de cette approche sont les suivants:

- Avec ce type de méthodes, il n'est pas toujours possible d'extraire des primitives indépendantes, et le système fait alors des choix (notamment de segmentation) sans aucune information contextuelle sur les entités supérieures [Mon11].
- Elle nécessite de connaissance à priori sur les typographies utilisées et encore une très grande précision dans la résolution des images est requise pour la manipulation de gros volumes de données.

Un exemple d'algorithme utilisant la stratégie ascendante est le fameux algorithme de lissage RLSA.

##### 4.1.1.1. L'algorithme RLSA

L'algorithme RLSA (Run Length Smearing/ Smoothing Algorithm) a été développé par Srihari et Wang dans [SW89], C'est un algorithme de segmentation appliqué sur des images binaires. Il permet le regroupement des pixels noirs voisins en régions (composantes connexes lignes, etc.), en procédant à un lissage horizontal et vertical de l'image. Le lissage est effectué en remplaçant les pixels blancs par de pixels noirs dans les séquences de pixels blancs de longueur inférieure ou égale à un certain seuil. L'image finale est obtenue par la réunion des deux images obtenues après le lissage horizontal et vertical.

Cet algorithme a certaines limites comme le choix des seuils qui doivent être fixés à priori et dans ce cas ils peuvent causer une sur-segmentation (quand les seuils sont trop faibles) ou une sous-segmentation (avec des seuils trop élevés). La figure suivante présente un exemple de la segmentation RLSA.

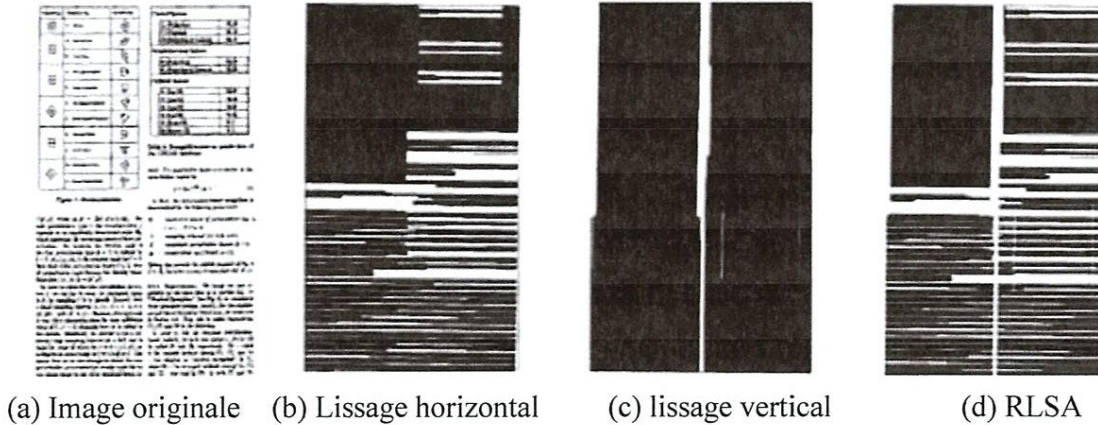


Figure 1.4: Segmentation RLSA. [BM07]

#### 4.1.2. Approche descendante (Top-down)

Les méthodes de cette classes sont basées sur le découpage de l'image en zones de grande taille découpées ensuite en petites zones par analyse de propriétés spécifiques en relation avec la nature du document traité à partir d'une connaissance a priori, puis à vérifier leur présence. Cette approche cherche donc à diviser les entités du document itérativement pour vérifier les hypothèses. Le problème de ces méthodes est qu'elles ne fonctionnent que si la connaissance et le contexte sont très bien définis et elles nécessitent donc de connaître la structure à priori du document (documents très hiérarchisés).

De cette approche il peut découler plusieurs algorithmes. Nous citons dans cette section deux algorithmes parmi les plus célèbres.

##### 4.1.2.1. L'algorithme de découpage XY

Introduit par Nagy et Seth [NS84], il peut-être bien adapté à des images de documents imprimés (formulaires, journaux, ouvrages, ...) qui sont composées en majorité de lignes de texte horizontales organisées en paragraphes, et de graphiques aux formes bien séparées du texte. L'algorithme décompose récursivement l'image du document en sous-rectangle. Pour faire cela on trace les profils de projection horizontaux et verticaux utilisant un seuil (correspondant respectivement à la somme des pixels le long de l'axe X et le long de l'axe Y). Ainsi le découpage se fait récursivement sur les zones d'espace les plus denses [NS84].

Le résultat d'une telle segmentation peut être représenté dans un arbre X-Y appelée aussi découpe récursive en utilisant le profil de projection, dans lequel la racine correspond à la



page toute entière et les feuilles représentent les blocs de la page et chaque niveau de l'arbre représente alternativement les résultats de la segmentation horizontale ou verticale [Had06].

La figure suivante présente un exemple de l'algorithme de découpage XY d'une image de document (une page de journal).



Figure 1.5: Découpage XY d'un document. [BM07]

#### 4.1.2.2. L'analyse des profils de projection

Cet algorithme permet la séparation des blocs de texte et la détection des lignes. Cet il consiste à projeter les valeurs des pixels noirs ou l'épaisseur du rectangle circonscrit des caractères, dans des directions horizontales et verticales de façon à obtenir deux histogrammes. L'histogramme des projections horizontales possède des maxima qui représentent les centres des lignes et des minima qui délimitent les bords inférieurs et supérieurs des lignes. L'histogramme des projections verticales donne les bords extérieurs gauches et droits des colonnes.

Cette méthode ne fonctionne correctement qu'avec des documents de structure simple, et elle suppose que les lignes soient correctement alignées horizontalement. Elle nécessite donc une correction préalable de la courbure et de l'inclinaison. De plus il faut binariser correctement l'image de façon à séparer correctement les lignes. Cette méthode donc n'est pas utilisable sur toutes les images de documents. Cependant les méthodes de projection peuvent être appliquées sur des morceaux des lignes de façon à réduire la sensibilité à l'inclinaison et éviter l'imbrication multiple avec les zones graphiques [LET03].

### 4.1.3. Approche mixte (Hybrid)

Elle combine l'analyse ascendante pour extraire les primitives locales et l'analyse descendante pour rechercher des primitives globales. Les méthodes exploitant à la fois les primitives locales et globales constituent aujourd'hui de nouvelles pistes de recherche.

L'objectif de l'approche mixte est d'augmenter la robustesse des résultats. Elle est plus efficace pour l'analyse du fond (par les espaces blancs).

Les méthodes dites mixtes peuvent être ascendantes au maximum et descendantes au minimum ou inversement.

## 4.2. Extraction de la structure logique

L'extraction de la structure logique d'un document consiste à attribuer des étiquettes logiques (titre, sous-titre, paragraphe, légende,...etc.) aux régions physiques identifiées lors de l'extraction de sa structure physique. L'étiquetage logique n'est possible qu'à partir de connaissances a priori sur la classe des documents à traiter, par exemple un titre, un résumé, un paragraphe sont les étiquettes logiques possibles pour un journal scientifique tandis que la date, l'objet, les coordonnées expéditrices sont des étiquettes logiques correspondant à un courrier [Mon11].

La reconnaissance de la structure logique comprend deux étapes : *l'étiquetage des blocs* et *la transformation de la structure physique en structure logique*. La première étape peut avoir lieu avant, après ou en même temps que la deuxième étape [Rob01].

- **L'étiquetage des blocs** : consiste à assigner des étiquettes logiques aux blocs physiques extraites précédemment. Les étiquettes des blocs permettent de fournir une indication sur le rôle de ces blocs. L'étiquetage des blocs lui-même se fait en deux étapes successives : l'extraction des caractéristiques et la classification
- **La transformation de la structure physique en structure logique** : consiste à fusionner des blocs physiques appartenant à la même entité logique et déterminer un ordre de lecture entre les entités logiques.

D'après [Had06], les travaux de recherche sur l'extraction de la structure logique sont moins génériques par rapport au celles de la structure physique. Cela est dû principalement au fait que la structure logique n'est pas standard mais elle est fortement dépendante de l'application à traiter. Cependant, plusieurs méthodes ont été proposées pour l'extraction des structures logiques, et elles sont regroupées en quatre grandes approches [Duo05]:



#### 4.2.1. Approche structurelle

Cette approche regroupe des méthodes qui s'appliquent directement sur les structures de représentation de données. Il s'agit d'algorithmes de transformation de graphes, d'arbres, etc. ou d'inférence de grammaires [Duo05].

La transformation permet de passer d'une structure physique représentée sous forme d'arbre ou de graphe à une structure logique en s'appuyant sur diverses techniques.

#### 4.2.2. Approche de type « Intelligence Artificielle »

Cette approche regroupe des méthodes issues de l'intelligence artificielle. Elles reposent sur la construction de règles à partir de différentes informations extraites au niveau physique pour trouver la structure logique. Dans un tel schéma, les relations éventuelles entre les composantes logiques ne peuvent pas être directement représentées. Ces méthodes utilisent des heuristiques et un langage de description DAL (Document Architecture Language) pour la construction de règles.

Les règles extraites sont ensuite exploitées par un système expert, un algorithme d'apprentissage automatique, un classifera, etc. pour effectuer l'étiquetage logique des entités physiques extraites précédemment.

#### 4.2.3. Approche probabiliste

Cette approche est choisie pour les documents à typographie riche. Les méthodes probabilistes consistent à considérer que les éléments ont été générés par un ensemble de distributions de probabilité. Le but est de s'adapter, au moyen des probabilités, au manque de régularité qui est dû à la structure du document même ou engendré par des erreurs de segmentation au niveau physique du traitement du document.

Ainsi, plusieurs techniques probabilistes ont été employées : les réseaux bayésiens, les n-grammaires généralisés, l'analyse grammaticale probabiliste, etc.

### 5. Représentation des structures

Un document peut être représenté au cours de sa reconnaissance par plusieurs formes peut-être images, arbre ou format structurée comme XML (eXtensible Markup Language) par exemple). Nous nous intéressons plus particulièrement aux formes physiques et logiques car ce sont les structures visées par la reconnaissance.

#### 5.1. Représentation des structures physiques

Les structures physiques sont parfois décrites par un arbre pour transcrire les liens hiérarchiques visibles qui existent entre les objets (*exemple* : un mot fait partie d'une ligne dans le cas de la structure physique)

Un autre moyen plus ouvert de décrire la structure physique avec la norme XML qui permet de spécifier n'importe quel format désiré. La figure 1.6 met en parallèle la représentation de la structure physique d'un document sous forme d'arbre avec la proposition de représentation sous forme XML

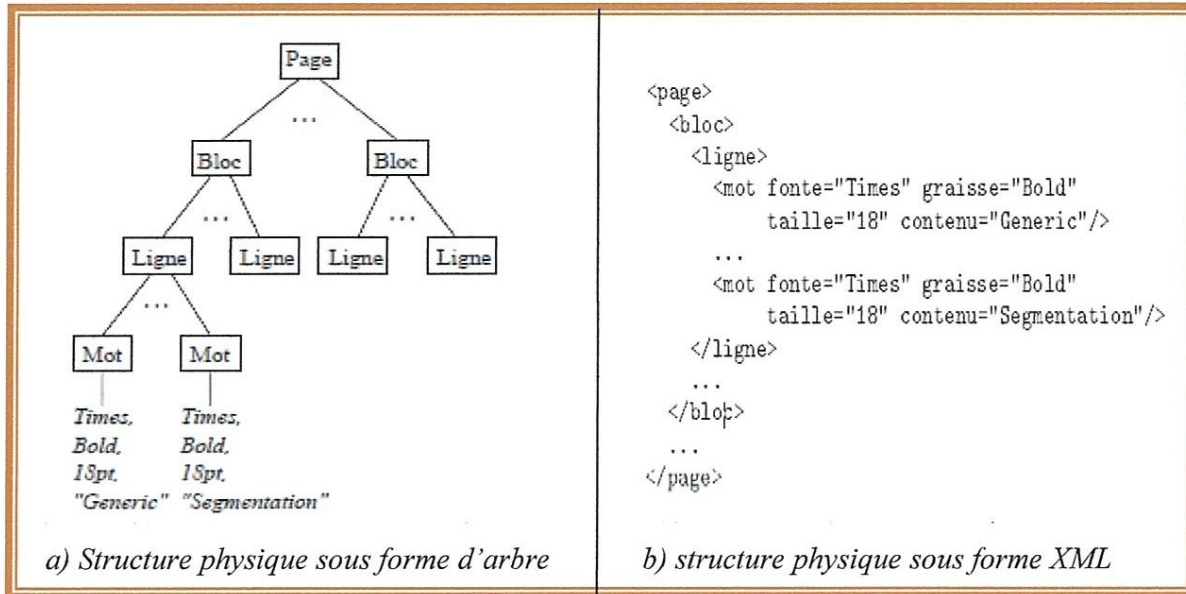


Figure 1.6 : Représentation d'une structure physique. [Rob01]

## 5.2. Représentation des structures logiques

La structure logique d'un document décrit son contenu sémantique, elle peut être représentée par un arbre comme la structure physique et encodée en XML. La figure 1.7 met en parallèle la représentation de la structure logique d'un document sous forme d'arbre avec la proposition de représentation sous forme XML.

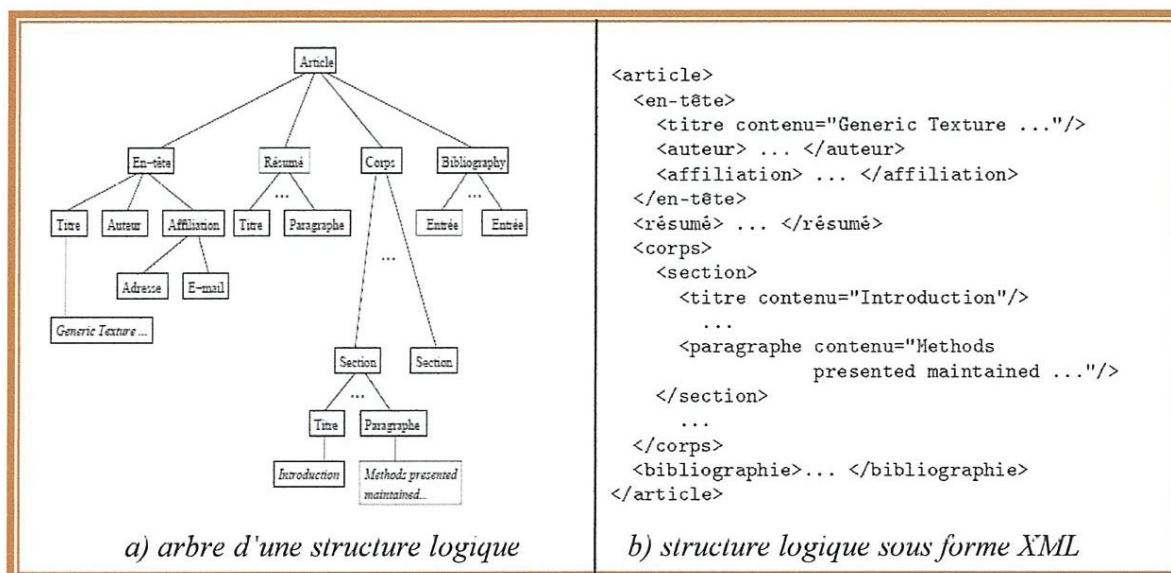


Figure 1.7 : Représentation d'une structure logique. [Rob01]



### 5.3. Structures génériques

En effet, on peut avoir deux structures plus ou moins similaires de deux documents différents. Si leurs structures comportent les mêmes étiquettes organisées hiérarchiquement de manière similaire, on dira que les deux documents appartiennent la même classe. Une classe de documents physiques est décrite par une structure physique appelée générique, de même une classe de documents logiques est décrite par une structure logique générique. Les structures qui décrivent une instance de ces classes (un document particulier) sont appelées spécifiques. Les structures des figures 1.6 et 1.7 sont des structures spécifiques. La structure générique physique ou logique d'une classe de documents est aussi appelée modèle physique ou logique de cette classe.

En XML les structures génériques sont décrites par des DTD (Document Type Definition). La figure 1.8 contient des extraits des DTDs des structures physique et logique du document. La description XML de la figure 1.6 est conforme au DTD de la figure 1.8.a) et la description XML de la figure 1.7 est conforme au DTD de la figure 1.8.b).

<pre> &lt;!ELEMENT page (bloc)*&gt; &lt;!ELEMENT bloc (ligne)*&gt; &lt;!ELEMENT ligne (mot)*&gt; &lt;!ELEMENT mot EMPTY&gt; &lt;!ATTLIST mot fonte NMTOKEN #REQUIRED&gt; &lt;!ATTLIST mot taille NMTOKEN #REQUIRED&gt; &lt;!ATTLIST mot contenu NMTOKEN #REQUIRED&gt; </pre> <p>a) structure physique générique</p>	<pre> &lt;!ELEMENT article (en-tête, résumé, corps,                     bibliographie)&gt; &lt;!ELEMENT en-tête (titre, auteur,                   affiliation)&gt; &lt;!ELEMENT titre EMPTY&gt; &lt;!ATTLIST titre contenu NMTOKEN #REQUIRED&gt; ... &lt;!ELEMENT résumé EMPTY&gt; &lt;!ATTLIST résumé contenu NMTOKEN #REQUIRED&gt; &lt;!ELEMENT corps (section)*&gt; &lt;!ELEMENT section (titre, paragraphe)*&gt; ... </pre> <p>b) structure logique générique</p>
---	---

Figure 1.8 : Structures génériques sous forme de DTD. [Rob01]

## 6. Conclusion

L'analyse de structures de documents est souvent une étape initiale sur laquelle de nombreux systèmes de reconnaissance de documents sont construits. Elle consiste à réaliser l'extraction séquentielle ou combinée des structures physique et logique des documents. Les documents papiers manipulés sont tellement divers qu'il est difficile d'avoir un système qui permet de reconnaître n'importe quel document.

Dans ce chapitre on a présenté une vue d'ensemble sur l'état actuel du domaine de l'analyse des structures de documents basé sur les méthodes classiques pour passer à l'étape de reconnaissance et cette dernière se fait en fonction des catégories de documents.

---

---

# Chapitre 2.

Etat de l'art sur L'analyse et la  
reconnaissance de documents structurés

---

---



## **1. Introduction**

L'analyse et la reconnaissance de document est une étape importante dans tout processus de traitement et d'analyse de document comme l'indexation, la recherche, la catégorisation et la classification automatique de documents. Elles désignent une discipline scientifique qui regroupe un ensemble de techniques informatiques dont le but est de reconstituer le contenu d'un document à partir de son image. L'analyse et la reconnaissance de document constituent donc un processus inverse de la production de document.

Nous avons vu dans le chapitre précédant que l'objectif de l'analyse et la reconnaissance d'un document est l'extraction de ses structures, et que l'organisation structurelle de document est soit physique ou logique. Nous avons vu également les différentes approches d'analyse et de reconnaissance de ses structures.

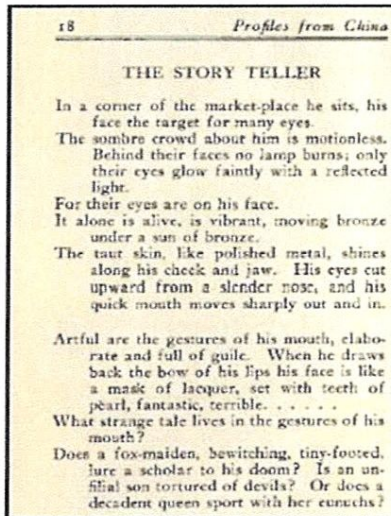
Le présent chapitre est une suite du chapitre précédent, dans lequel nous allons présenter un état de l'art sur les principaux travaux effectués dans la littérature sur le domaine de reconnaissance de document. Ce chapitre est organisé comme suit. Tout d'abord, nous commençons par la présentation des différents types de documents structurés. Ensuite, nous détaillons dans la section 3 les différentes méthodes de reconnaissance des structures physiques et logiques proposées dans la littérature en les regroupant selon le type de document traité, avant de conclure.

## **2. Documents à structure simple et documents à structure complexe**

Les documents se caractérisent suivant l'organisation spatiale de ses différentes zones, quel que soit dans la disposition de ses éléments et leur structuration ou dans son contenu. Ainsi, les documents peuvent être structurés lorsqu'ils sont créés autour d'une structure ou non structurés. D'après [Had06], les documents structurés peuvent être imprimés ou manuscrits. Parmi les documents imprimés nous distinguons les documents à structures simples et les documents à structures complexes.

### **2.1. Les documents à structure simple**

Ce sont les documents qui obéissent à des règles de mise en page connues et standardisées. Elles peuvent être linéaires et régulières comme les œuvres littéraires tels que les romans, ou à structures hiérarchiques simples comme les articles scientifiques ou les livres. Les deuxièmes possèdent une organisation en chapitres, sections, articles et paragraphes que l'on peut les représentées sous forme d'arbre. La figure suivante présente deux exemples de documents à structure simple.



(a) Exemple d'une page de livre



(b) Exemple d'une page de journal scientifique

Figure 2.1 : Exemples de documents imprimés ayant une structure simple. [Mon11]

## 2.2. Les documents à structure complexe

Un document à structure complexe est un document imprimé composé essentiellement de blocs hétérogènes textuels et pouvant contenir des expressions mathématiques, des tableaux, des graphiques et des photographies [Azo95]. Les documents à structure complexe sont caractérisés par la variabilité de positionnement, de forme et d'apparition des zones, dans lesquelles les différents blocs de texte ne sont pas parfaitement alignés, avec une mise en page complexe qui peut présenter plusieurs colonnes avec des tailles de corps et d'interligne irrégulières.

Les deux familles des documents composites sont les documents complexes et à structure stable (formulaire, lettre commerciale, etc.), et les documents complexes et à structure variable où du texte intervient l'inclusion de blocs dans d'autres (les journaux, les magazines, les dépliants publicitaires, etc.). Les deux familles des documents composites possèdent une typographie riche et ne sont pas composées uniquement de texte mais d'une combinaison, selon une disposition variable, de textes, de graphiques et d'images.

La figure suivante présente deux exemples d'images de documents à structure complexe.





### 3.1.1. Reconnaissance de la structure physique

Dans le cas des documents à structure simple et linéaire (les pages d'une revue scientifique par exemple), la segmentation physique est une séparation des zones textuelles des zones non-textuelles. Le texte peut être découpé en blocs représentant les paragraphes et les zones non-textuelles regroupent des figures, des traits, des symboles, des logos, des schémas, des délimiteurs entre zones de texte ...etc.

Comme nous avons dit dans le premier chapitre, les méthodes classiques de reconnaissance des structures physiques de documents peuvent être *ascendantes*, *descendantes*, ou *mixtes*.

#### 3.1.1.1. Méthodes ascendantes

Ces méthodes consistent à assembler suivant des normes typographiques, les éléments du plus bas niveau (simples connexités) en éléments successifs définissant les caractères, les mots, les lignes de texte, les paragraphes, les colonnes.

Parmi les premières méthodes sur la reconnaissance de la structure physique de document proposées dans la littérature nous citons celle de Wong et al. [WCW82]. C'est une méthode de regroupement de pixels basée sur l'algorithme RLSA (*Run Length Smearing/ Smoothing Algorithm*) décrit dans le premier chapitre. L'algorithme applique le regroupement des pixels en séquence binaire avec 0 pour le pixel blanc et 1 pour le pixel noir. L'algorithme permet donc de transformer une séquence  $x$  en une séquence  $y$  suivant certaines règles: toutes suite consécutifs dans  $x$  dont le nombre est inférieure ou égale à un seuil  $C$  sont changés en 1 dans  $y$ , les 1 dans  $x$  restent changés. Dans cette méthode, l'algorithme RLSA est appliqué horizontalement et verticalement avec deux valeurs de seuils différentes ce qui produit deux images lissées différentes. Ces dernières sont finalement combinées par un ET logique pour fournir un résultat final. Le résultat de cette méthode sont des blocs étiquetés suivant certaines caractéristiques en texte, non-texte ou filets.

Une autre célèbre méthode de reconnaissance de la structure physique de certains types de documents techniques, notamment les pages de revues scientifiques et les cartes de visite, est celle d'O'Gorman [Ogo93]. C'est une méthode basée sur le *clustering*. Dans le cas des pages de journal, la méthode d'O'Gorman décompose la page en paragraphes et en colonnes. Le principe de cette méthode est de segmenter le document en composantes connexes (caractères) puis chercher les *k-plus proches voisins* de chaque composante. Les lignes de texte sont ensuite déterminées par association des *k-plus proches voisins* possédant un angle et une distance similaires. Les blocs sont finalement obtenus en examinant les lignes de texte deux à deux ; si deux lignes respectent certaines conditions, approximativement parallèles,



proches en distance perpendiculaire et se chevauchent ou séparées par une faible distance parallèle alors elles appartiennent au même bloc. Le problème de cette méthode est qu'elle n'est pas conçue pour des documents composites puisqu'il ne gère pas les régions non textuelles. En plus, dans le cas des espacements très irréguliers, l'algorithme peut causer un sur-échantillonnage des segments recherchés.

L'algorithme de lissage RLSA a été utilisé également dans [Sun06]. Dans ce travail, la méthode proposée utilise la segmentation par lissage divisé en deux exécutions de RLSA. La première est effectuée pour extraire les composantes textuelles de petites tailles. Ainsi, un contexte local a été utilisé dans la première exécution pour déterminer la taille des seuils. La seconde exécution a pour but d'extraire les composantes textuelles ayant des polices plus grandes, en utilisant des informations globales pour les réaliser sur le résultat de lissage de la première exécution.

### ***3.1.1.2. Méthodes descendantes***

Ces méthodes consistent à la division d'une image de document en plus petites régions, et nécessitent de définir des connaissances a priori sur la structure du document. Elles procèdent à un découpage itératif en régions à partir de ces connaissances, en se basant soit sur segmentation par découpage X-Y, soit sur une segmentation par analyse du fond de l'image.

Une des premières méthodes descendantes est celle de Nagy et Seth [NS84]. La méthode proposée repose sur le découpage récursif du document en un arbre X-Y, tel que chaque nœud de l'arbre X-Y correspond à un bloc rectangulaire dans le document. Les successeurs de chaque nœud correspondent à des blocs qui sont obtenus par les découpages horizontales ou verticales du rectangle parent. A chaque étape du découpage récursif, une projection verticale et horizontale des profils de niveaux de gris de chaque bloc sont calculées. Après, les espaces blancs (les vallées) des profils de projections sont comparés à un certain seuil pour déterminer les limites des blocs. Du point de vue représentation, quand l'espace blanc est plus grand que le seuil prédéterminé, le nœud correspondant au bloc père est alors divisé en deux nœuds fils. Le processus opère jusqu'à ce que les nœuds ne puissent plus être divisés.

Ha et al. ont ajouté dans [HHP95] une étape de division du document à partir des boîtes englobantes des composantes connexes au lieu des pixels dans la méthode de découpage X-Y. Le principe de cette méthode est de compter le nombre de boîtes qui se chevauchent pour chaque coordonnée. Comme résultat, un document est découpé au niveau des espaces qui doivent être supérieurs à un certain seuil, dans les profils de projections.

Pour la segmentation par analyse du fond de l'image, le principe général de ce type de méthodes est de rechercher un ensemble de rectangles maximaux qui ne contiennent pas de

pixels noirs (du premier plan). Ces rectangles une fois fusionnés permettent de construire les régions qui délimitent les blocs.

L'analyse de fond a été utilisée dans [BFJ90] pour trouver des rectangles blancs maximaux. Cette méthode fusionne des rectangles blancs non maximaux fournis en entrée tant que le critère d'arrêt basé sur des heuristiques n'est pas atteint.

Dans [PZ91], les auteurs utilisent aussi l'algorithme d'analyse du fond de l'image pour trouver les colonnes blanches les plus larges possible localement, pour les fusionner selon deux critères : la taille et le rapprochement. Donc, si ces colonnes ont des tailles similaires et si elles sont suffisamment proches, alors elles sont fusionnées.

### **3.1.2. Reconnaissance de la structure logique**

Les entités logiques de la première page d'un article scientifique par exemple sont le titre, les noms d'auteurs, leurs emails, affiliations et adresses, le résumé, les mots clés, le ou les colonnes de texte, etc.

#### **3.1.2.1. Méthodes structurelles**

Ces méthodes reposent sur l'utilisation de grammaires et des représentations par des arbres ou par des graphes.

Krishnamoorthy et al. [KNS93] ont utilisé des grammaires déterministes pour segmenter et étiqueter les blocs à partir des profils de projection obtenus après l'utilisation de la technique de découpage X-Y. Pour faire cette segmentation elles interviennent quatre étapes : la première est de repérer les suites consécutives de 0 et de 1 dans ces profils, appelées atomes. La seconde c'est de grouper les atomes contigus en molécules. Ensuite, la troisième étape est d'affecter des étiquettes à des entités de document. La quatrième étape c'est la fusion des entités de documents à l'aide de grammaires de bloc (peuvent être écrites pour un résumé, une colonne de texte régulière, etc).

La méthode proposée par Lee et al. [LCC03] consiste à regrouper les lignes de texte adjacentes proches entre elles et possédant les mêmes caractéristiques géométriques pour différencier les blocs en classes d'entête ou de corps de texte. Après la production, la séquence des types d'entête et le corps de texte sont réunie à partir de la structure fonctionnelle pour obtenir l'arbre de structure logique.

#### **3.1.2.2. Méthodes de type Intelligence Artificielle (à base de règles)**

Ces méthodes issues de l'intelligence artificielle visent à construire des règles en se basant sur différents types d'informations.



Une méthode ascendante a été proposée par Ingold [Ing91] pour la reconnaissance de structures logique d'un document. Dans cette méthode, une description formelle, comprenant des règles de composition et des règles de présentation, est effectuée pour chaque classe de documents. Cette description sert à la construction d'une série d'automates qui permet d'effectuer l'analyse. L'expérimentation a été effectuée sur les textes juridiques.

### **3.1.2.3. Méthodes probabilistes**

Une méthode statistique a été proposée par Brugger et al. Dans [BIZ97]. La méthode proposée utilise la structure physique pour trouver la structure logique à l'aide de n-grams. Le modèle de n-grams a été utilisé pour les structures linéaires, mais il est généralisé pour être appliqué sur des structures d'arbre qui sont des informations en deux dimensions. Le processus de reconnaissance consiste à construire l'arbre optimal logique à partir des entités physiques, le modèle d'un document est géré par apprentissage.

Un autre modèle probabiliste a été proposé par Souafi dans [SP01]. Le modèle proposé est représenté par les réseaux Bayésiens pour l'étiquetage logique de documents. Ainsi, le classifieur (les réseaux Bayésiens) est utilisé pour représenter les relations entre l'ensemble des attributs et la classe d'étiquetage correspondante. Le modèle proposé utilise un apprentissage supervisé. L'expérimentation a été effectuée sur les tables de matières de plusieurs magazines commerciaux et scientifiques.

## **3.2. Reconnaissance des documents à structure complexe et stable (les formulaires)**

Un formulaire est une structure complexe avec une organisation irrégulière et est composée de cases dont la disposition est spécifique à chaque formulaire [Mar09]. Les objets de formulaire sont des items de texte dont la position est assez précisément connue et qui représentent des montants, une adresse ou d'autres types d'informations dont on a presque toujours une étiquette sémantique précise. La figure 2.2.b présente un exemple de formulaire.

D'après [Mar09], l'extraction de la structure de formulaire est effectuée à l'aide des filets. Si les filets sont cassés, il sera difficile de reconnaître le formulaire.

### **3.2.1. Reconnaissance de la structure physique**

La structure physique des formulaires est très variable, car on peut représenter la structure logique de différentes manières : on peut changer les dimensions des items et faire varier la nature des séparateurs (lignes continues, plages blanches, etc.).



Selon [Mar09], la reconnaissance de la structure physique d'un formulaire permet d'extraire ses différentes cellules séparées par des filets. Les différentes méthodes de reconnaissance des formulaires proposées dans la littérature reposent soit sur les filets soit sur les intersections.

### **3.2.1.1. Méthodes utilisant les filets**

Parmi les méthodes utilisant les filets pour reconnaître la structure physique des formulaires nous citons celle de Xingyuan et al. [XGDO99]. Cette méthode commence par l'extraction des segments en appliquant une érosion et une dilatation sur l'image de formulaire. Ensuite, elle regroupe les segments extraits précédemment en filets suivant leur position. Les filets dont la longueur des segments est inférieure à un seuil donné sont rejetés. Les rectangles des cellules sont finalement extraits suivant 3 contraintes.

Le nombre de filets et leur position ont été utilisés par Chen et Tseng dans [CT97]. La méthode proposée consiste à apprendre, à partir d'un formulaire non rempli, le nombre de filets horizontaux, verticaux et diagonaux, les listes des filets, les coordonnées des coins haut-gauche et bas-droite du formulaire, et les coordonnées de chaque cellule du formulaire.

### **3.2.1.2. Méthodes utilisant les intersections**

Dans [FN00], les auteurs ont utilisé une méthode de reconnaissance des cellules de formulaires en basant sur les intersections. La méthode proposée permet de détecter les intersections manquantes et les fausses intersections. Pour chaque type d'intersection et pour une direction donnée celle-ci donne une liste d'intersections voisines acceptées et une liste d'intersection voisines rejetées. Les erreurs sont récursivement détectées puis corrigées et les corrections se font dans un certain ordre.

La même idée a été utilisée par Hadano et al. Dans [HMSS01]. La méthode proposée commence par la détection des cellules à partir des intersections. Les intersections manquantes sont ensuite détectées et ajoutées.

## **3.2.2. Reconnaissance de la structure logique**

Pour un formulaire, la structure logique est intimement liée à la structure physique. Les objets sont des items de texte dont la position est assez précisément connue et qui représentent des montants, une adresse ou d'autres types d'informations dont on a presque toujours une étiquette sémantique précise. Les méthodes mises en œuvre sont guidées par la structure physique et se fondent sur la position absolue des items, leur alignement et l'emplacement des traits séparant les différentes cases. Cette structuration, qui semble triviale, cache de sérieux problèmes. D'abord, la structure physique est très variable, car on peut représenter la structure logique de différentes manières : on peut changer les dimensions des items et faire varier la

nature des séparateurs (lignes continues, plages blanches, etc.). Ensuite, la structure logique est complexe, car elle met en relation des items qui ne sont pas toujours physiquement voisins.

### ***3.2.2.1. Méthodes basées sur la représentation en arbres hiérarchiques***

Une méthode permettant d'extraire la structure logique d'un formulaire en reconnaissant la hiérarchie entre les cases pré-remplies et les cases vides a été proposée dans [ZLJO00]. Cette extraction est effectuée en trois phases. La première phase effectue une division globale du formulaire qui détecte tous les rectangles du formulaire englobant le maximum de champs de données, puis à partir de ces rectangles en les étendant on détecte les sous-formulaires (rectangle de formulaire basique) contenant les champs de données et les champs de titres. La deuxième phase analyse localement la structure logique à l'intérieur de chaque sous-formulaire et associe les cellules de données aux cellules de titres. La troisième phase est une re-division du formulaire pour obtenir les sous-formulaires composés [Mar09].

### ***3.2.2.2. Méthodes basées sur une grammaire de graphe***

Les grammaires ont été également utilisées pour reconnaître la structure logique des formulaires. L'auteur dans [AA03] a défini une grammaire de graphe en ajoutant certaines relations entre les cellules comme « même hauteur », « inclus », « non connectés ». Cette méthode suppose que les cellules ont été correctement détectées.

## **3.3. Reconnaissance des documents à structure complexe et variables (les pages de journal)**

Les pages de journaux ont des structures complexes et variables. Généralement, tous les journaux sont bâtis autour des mêmes entités physiques comme : les filets, les cadres, les images, les textes et les blocs. Néanmoins, l'utilisation de ces entités pour bâtir la structure physique du journal diffère d'un éditeur à l'autre. Par exemple, les filets utilisés pour séparer les articles du journal, peuvent être représentés par un segment continu chez l'un et par des segments discontinus chez un autre éditeur. La grande majorité des articles sont séparés des espaces blancs, les autres sont séparés par des filets ou par des cadres. Certains journaux possèdent des régions vides, à cause de la présence de publicités, et d'autres possèdent comme fond une image. C'est pourquoi le journal a été choisi comme sujet d'études tout au long de ce travail. La figure 2.3 présente trois exemples de pages de journaux ayant trois structures différentes.





Figure 2.3 : Diversité des mises en pages d'un journal à l'autre. [Dum05]

### 3.3.1. Reconnaissance de la structure physique

Les journaux font partie d'une catégorie de documents à structure complexe. La structure physique de ce type de documents se caractérise par : la variabilité intra-classe, la découpe en articles, l'utilisation d'objets structurants, les entrefilets, le mode d'intégration des illustrations au texte et l'organisation des blocs dans la page.

Contrairement aux autres types de documents (à structure simple), la reconnaissance des pages de journal n'a pas suscité suffisamment de recherches. Les premiers travaux ont essayé d'appliquer les méthodes de reconnaissance des documents à structures simples sur les pages de journal qui sont à structures complexes. D'autres ont proposé des nouvelles méthodes mieux adaptées à ce genre de documents.

Cependant la plupart des méthodes de reconnaissance des documents à structures complexes proposées dans la littérature sont soit ascendantes ou mixtes. Cela est justifié par le fait que les méthodes descendantes nécessitent (comme nous avons mentionné dans le chapitre 1) de connaître la structure à priori du document ce qui n'est pas possible pour le cas des documents à structures complexes et variables comme les pages de journal.

#### 3.3.1.1. Méthodes ascendantes

Pour la reconnaissance des différentes entités d'une page de journal: les filets, les images, les graphiques et les textes, Liu et al. [LLYH01] ont utilisé une méthode ascendante basée sur le regroupement des composantes connexes. Pour la fusion des lignes de texte en blocs, les composantes connexes voisines sont prises en considération et seulement la paire de composantes connexes la plus valable est choisie pour la fusion. Un filtrage est ensuite appliqué pour supprimer les composantes connexes de petites tailles. L'étape dernière est



l'étiquetage du texte en titre et la séparation graphique image à partir d'un graphe. Notons que cette méthode a participé à la première compétition de segmentation des documents à structures complexes organisée instaurée au sein de la conférence international ICDAR 2011 (*International Conference on Document Analysis and Recognition*).

Une autre méthode ascendante a été proposée par Mitchell et Yan [MY01] dans le cadre du concours de segmentation des documents à structures complexes en 2001. Après la segmentation de l'image, la méthode procède au regroupement des régions rectangulaires qui contiennent le plus de pixels du premier plan, afin de construire des patterns. Ces derniers sont plus grands et moins nombreux que les composantes connexes ; cependant ils garantissent la segmentation de composants séparés par plus que trois pixels. La taille, la forme et la plage de valeurs de pixels sont les caractéristiques utilisées lors de la classification de l'entité. Enfin, les patterns sont regroupés pour former les lignes et les blocs.

Hadjar et Ingold [HI03] proposaient un algorithme utilisant une approche ascendante basée sur les composantes connexes. L'algorithme consiste en un ensemble d'étapes : extraction de filets, extraction de cadres, séparation texte/ images, extraction des lignes de texte et regroupement des lignes en blocs. Cependant, les composantes connexes sont utilisées pour l'extraction des images, filets et cadres. Elles sont également utilisées pour l'extraction des lignes de textes après l'application de l'algorithme de lissage RLSA. Le regroupement des lignes en blocs se fait suivant certaines règles tout en prenant en considération les caractéristiques de la langue arabe.

### ***3.3.1.2. Méthodes descendantes***

Cinque et al. [CLM02] ont proposé une méthode baptisée DAN (Document Analysis on Network) composée de trois étapes. Premièrement, un prétraitement est appliqué sur l'image de document (la page de journal) afin d'améliorer sa qualité et d'éliminer le bruit. Sur l'image résultante, elle applique une technique de quad-arbre dans le but de découper le document en des petits blocs. Le résultat de découpage constitue l'entrée de la troisième étape : la fusion. Cette étape applique des critères de pré-classification pour fusionner les blocs similaires en des régions plus larges. Des opérateurs locaux ont été utilisés conjointement avec des seuils variables afin de calculer la phase de pré-classification. Cette méthode a participé à la compétition de segmentation des documents complexes organisée par l'ICDAR en 2003 et elle a présenté des résultats encourageants.

### ***3.3.1.3. Méthodes mixtes***

La méthode de Srihari et Wang [SW89] est considérée comme la première tentative d'appliquer des techniques de reconnaissance des documents à structures simples sur des

documents à structures complexes. Cette méthode combine l'algorithme de lissage RLSA et l'algorithme de découpage récursif X-Y pour extraire les blocs rectangulaires homogènes d'une page de journal. Les blocs sont ensuite classés en fonction de caractéristiques textuelles statistiques et des techniques de décision de l'espace.

Une méthode similaire utilisant le même principe est celle de Govindaraju et al. [GLNS90]. La seule différence est au niveau de l'obtention des blocs qui se fait en fusionnant les composantes connexes en de grandes zones.

Dans [CGM99], les auteurs utilisent une méthode mixte pour la segmentation de pages de journaux numérisés ainsi que l'identification des articles, basée sur un ensemble d'algorithmes intégrés. Elle combine une technique ascendante (l'étiquetage des composantes connexes pour l'extraction de blocs), avec une technique descendante (analyse du fond de l'image).

Hadjar et al. ont proposé dans [HHI01] une technique de segmentation de pages de journaux basée sur le découpage et la fusion de zones (split and merge). C'est une méthode mixte utilisant le principe de découpage et de fusion. Les étapes de cette technique sont: l'extraction de l'image, extraction des filets horizontaux et verticaux, découpage de l'image du journal en de petites zones à partir des filets verticaux et horizontaux extraits et fusion de ces petites zones pour former des régions plus grandes, extraction des lignes de texte, étiquetage des blocs en zones de texte et en zones de titre. Cette technique est la troisième méthode participante au concours de segmentation des documents à structures complexes organisé au sein de l'ICDAR 2001.

La deuxième méthode participante à la compétition organisée par l'ICDAR 2003 est celle de Chowdhuri et al. [CCD02]. Cette méthode s'applique sur des images en niveaux de gris et elle commence par le lissage de l'image d'entrée (en utilisant un filtre moyen  $3 \times 3$ ). Après, elle extrait des régions en niveaux gris en appliquant les opérations d'ouverture et de fermeture morphologiques. Une binarisation est ensuite effectuée suivie par une analyse des composantes connexes permettant d'éliminer les grandes zones de bruits. Si l'image est inclinée, une étape de détection et de correction de l'inclinaison doit ensuite intervenir. Le processus de détection commence par les lignes de séparation si elles existent. Puis, les régions de texte sont détectées comme des régions individuelles. Les zones restantes constituent soit du bruit, soit des figures, soit des lignes. Finalement, le bruit est séparé des lignes et des figures par analyse des composantes connexes et opérateurs morphologiques.

Dans le cadre de la compétition de l'ICDAR 2003 toujours, Goey a proposé une méthode mixte qui s'exécute comme suit. D'abord, les composantes connexes sont identifiées et réparties en les classes suivantes: petits caractères, grands caractères, graphiques, caractères



normaux, lignes horizontales, lignes verticales, et bruit. Cette classification est effectuée en utilisant un arbre de décision construit manuellement et basé sur des caractéristiques comme la hauteur, la largeur, le nombre de pixels, etc. Le résultat de classification aide au découpage de l'image en 3 sous-images: (a) image contenant des figures, photos, et bruit, (b) image contenant des lignes, et (c) image contenant du texte. Dans le dernier cas, les blocs de texte, dans lesquelles la majorité des composantes connexes sont classées comme des grands caractères sont extraits dans une image séparée. Ainsi, la zone contenant du texte est divisée en 2 images, (c1) une image contenant du texte normal et petit, et (c2) une autre contenant des titres. Après, les composants dans l'image (c1) et ceux dans l'image (a) sont fusionnés en blocs en employant un lissage RLSA. Les blocs résultants sont ensuite classés par un arbre de décision entraîné qui prend des statistiques sur la classe des composantes connexes comme entrée. Dans l'image (b), chaque ligne est considérée comme un bloc séparé et étiquetée « Séparateur ». Les blocs dans l'image (c2) sont trouvés en appliquant un algorithme de regroupement des composantes connexes suivi par une post-classification afin de s'assurer que les blocs contiennent effectivement du texte.

Dans le concours de segmentation des documents complexes de l'ICDAR 2009, la méthode gagnante est la méthode *Fraunhofer Newspaper Segmenter*. Cette méthode commence par la binarisation de l'image d'entrée. Ensuite, elle procède à la détection des séparateurs noirs, suivie par la détection des séparateurs blancs. Ces derniers sont des rectangles vides satisfaisant certains critères : de hauteur, largeur, etc. L'étape suivante est la segmentation de la page en utilisant une technique hybride. La technique utilisée comprend une étape ascendante guidée par des informations descendantes dans la forme de la disposition des colonnes logiques de la page. Les régions de texte sont séparées du non-texte en employant des propriétés statistiques du texte (les caractères alignés sur les lignes de base, etc.). Finalement, on extrait les lignes de texte et les régions. Les lignes de texte sont détectées à partir des régions de texte extraites précédemment. Des caractéristiques de police (largeur de trait, italique, gras, ...) sont calculées pour chaque ligne de texte et utilisées pour dériver les régions de texte avec des propriétés similaire.

### 3.3.2. Reconnaissance de la structure logique

Les pages de journal sont des documents à structure complexe et riche. Les entités logiques souvent trouvées sont : article, titres, colonne, paragraphe, figure, entête, numéro de page, séparateur, lettrine...etc.



### **3.3.2.1. Méthodes structurelles**

Belaid [Bel97], propose une méthode basée sur l'inférence de grammaires d'arbres et l'apprentissage pour la création d'un modèle générique adaptable à des documents variés et fortement structurés. Le modèle est présenté sous forme d'arbre où les nœuds sont des boîtes rectangulaires disposées de manière hiérarchique. Il est exprimé à l'aide de constructeurs et de qualificatifs. La construction du modèle générique est précédée par la construction de modèle spécifique pour chaque document en entrée.

### **3.3.2.2. Méthodes de type Intelligence Artificielle (à base de règles)**

Ces méthodes reposent sur les modèles à base de règles implicites comme les modèles statistiques à base de classifieurs ou sur l'apprentissage automatique.

Toyoda et al. [TNN82] ont été intéressés à la reconnaissance de la structure logique des pages d'un journal japonais. L'étiquetage des différentes régions extraites lors de la phase de reconnaissance de la structure physique est effectué en utilisant des heuristiques assez précises sur la mise en forme. Ces heuristiques sont codées sous forme d'un ensemble de règles portant sur les positions de différentes zones. Les articles sont finalement extraits par le regroupement des zones qui les constituent.

Pour reconnaître la structure logique des documents, Asada et Tsujimoto [AT90] ont proposé des modèles à base de transformation de l'arbre physique en arbre logique où les opérations de transformation de l'arbre et d'étiquetage se font successivement à partir de règles. Ces modèles donnent de bons résultats lorsque les structures sont relativement stables. Le problème de ces méthodes est qu'elles sont trop rigides dans le cas de structures de documents variables. L'utilisation de ces modèles exigerait donc de définir un nombre trop important de règles.

Un système nommé DeLoS limité aux journaux a été présenté dans [NS95]. Ce système repose sur l'utilisation des règles pour l'extraction des structures logiques. Ainsi, le nombre de règles employés s'élève à 160 règles. Pour inférer les classes et les étiquettes des blocs dans une image de document, et afin de créer les unités logiques, le système conçu repose sur l'utilisation des heuristiques appliquées sur la structure physique et sur la combinaison des blocs étiquetés. Le DeLoS permet aussi de restituer l'ordre de lecture.

L'auteur dans [Rob01] a proposé un modèle appelé 2(CREM) pour la reconnaissance des pages de journal basé sur des patterns bidimensionnels. Des méthodes à base de retours de pertinence peuvent être appliquées pour affiner les modèles d'apprentissage. Pour construire

des modèles plus généralistes certains auteurs utilisent des méthodes structurelles utilisant des grammaires.

Dans [HHI01], conjointement avec l'extraction de la structure logique des pages de journal, les auteurs proposent d'étiqueter les blocs extraits en *figures*, *titres* et *textes*. Les figures sont séparées du texte lors de la première étape de l'extraction de la structure physique. Des règles relatives à la hauteur dominant de caractères et la distance moyenne entre les lignes de texte pour accomplir l'étiquetage logique des blocs textuelles en *titres* et *textes*.

Dans [PHT12], les auteurs ont proposé une méthode destinée à la segmentation logique d'articles dans des journaux anciens. La segmentation a pour but d'extraire des métadonnées à partir des images numérisées grâce à l'utilisation conjointe d'une méthode de classification de séquence de pixels basée sur les champs aléatoires conditionnels, associé à un ensemble de règles définissant la notion même d'article au sein d'un numéro de journal. La méthode suit les étapes suivantes : segmentation de l'image par champs aléatoires conditionnels, lissage de la segmentation par vote majoritaire, extraction des lignes de texte, génération d'une grille de séparateurs, et analyse récursive pour l'extraction des articles et du sens de lecture.

#### **4. Conclusion**

Nous avons présenté dans ce chapitre un état de l'art sur quelques méthodes existantes dans le domaine de la reconnaissance de documents. Ce dernier constitue un processus important dans toute application de traitement et d'analyse de documents. La reconnaissance de documents sert à construire une version électronique exploitable à partir d'un document papier. La diversité des documents papiers rendre difficile de concevoir une méthode générique capable de reconnaître n'importe quel type de documents.

---

---

# Chapitre 3.

## Conception

---

---



## 1. Introduction

Après avoir cerné le problème et les différentes techniques de reconnaissance de la structure logique des documents, on va présenter dans ce chapitre l'approche que nous avons adoptée pour le résoudre.

Ce chapitre est destiné à la description de l'approche proposée pour la reconnaissance de la structure logique de documents, et plus précisément des pages de journaux arabes, et les différentes étapes intervenues. Nous essayons à travers ce chapitre d'expliquer et de démontrer comment nous puissions faire passer d'une image brute de page de journal à un ensemble d'informations structurées exploitables représentant l'organisation logique du document.

On commence par présente l'entrée et la sortie de notre application. Puis, on énumère les caractéristiques des pages de journaux arabes utilisées tout au long de ce travail. Le reste du chapitre est consacré à l'exposition de la démarche suivie tout en détaillant les différentes étapes incluses.

## 2. L'entrée / Sortie

L'entrée de notre application est une image correspondante à une page d'un journal numérisée. Notons qu'il y a trois paramètres qui doivent être respectés :

- ♦ La structure des pages du journal doit être unie.
- ♦ Les conditions de numérisation et d'éclairage sont parfaites.
- ♦ Le journal du sujet doit être en langue arabe.

La sortie de notre application est un fichier XML. (Figure 3.1).

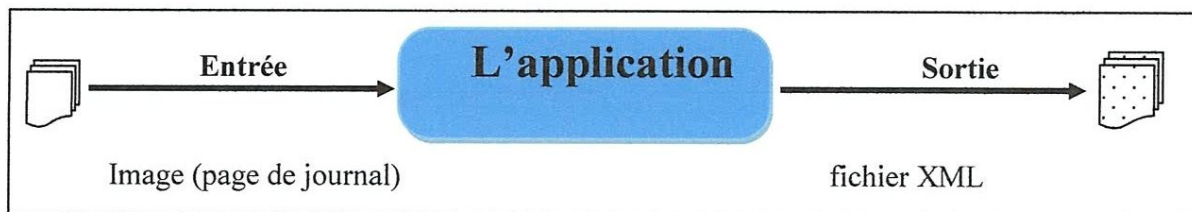


Figure 3.1 : L'entrée / Sortie de notre application.

## 3. Caractéristiques des pages de journaux utilisées

Notre application a été conçue pour traiter des pages de journal en langue arabe, et nous avons choisi le journal quotidien Echorouk pour notre corpus de test. Les pages de ce journal possèdent une grande variabilité dans leur structure rendant leur traitement et analyse très difficiles.

Les caractéristiques des pages de journal quotidien Echorouk utilisées dans le test sont les suivants. Une page du journal Echorouk contient (Figure 3.2):

- Un entête englobant le nom du journal (en arabe), le numéro de la page, la date, et la catégorie des articles de la page (sportifs, politiques, etc.). L'entête est délimité au-dessous par une ligne droite horizontale.
- Certaines pages contiennent également un pied de page exposant les informations de contact du journal.
- Zones publicitaires. La présence de ces zones n'est pas obligatoire dans la page.
- Plusieurs articles avec différents niveaux (articles longs, courts, principaux,...etc.).  
Chaque article contient :
  - Un ou plusieurs titres de différents niveaux.
  - Un résumé de l'article au-dessous des titres (peut être absent).
  - Une ou plusieurs figures accompagnées chacune ou pas par une légende. L'article peut ne contenir aucune figure.
  - Un nom d'auteur. Ce nom peut être trouvé soit après le résumé ou bien à la fin de l'article.
  - Une ou plusieurs colonnes de texte.
  - Une ou plusieurs bandes (rectangles pleines) englobant du texte (blanc), et elles correspondent soit à des titres ou à des articles secondaires. Ces bandes peuvent être absentes.
  - Certains articles sont entourés par un rectangle noir. Certains autres incluent des lignes droites horizontales servant de séparateurs entre des parties de l'article.
- On peut trouver dans la page des lignes droites horizontales ou verticales, utilisées comme séparateurs entre les différents articles.

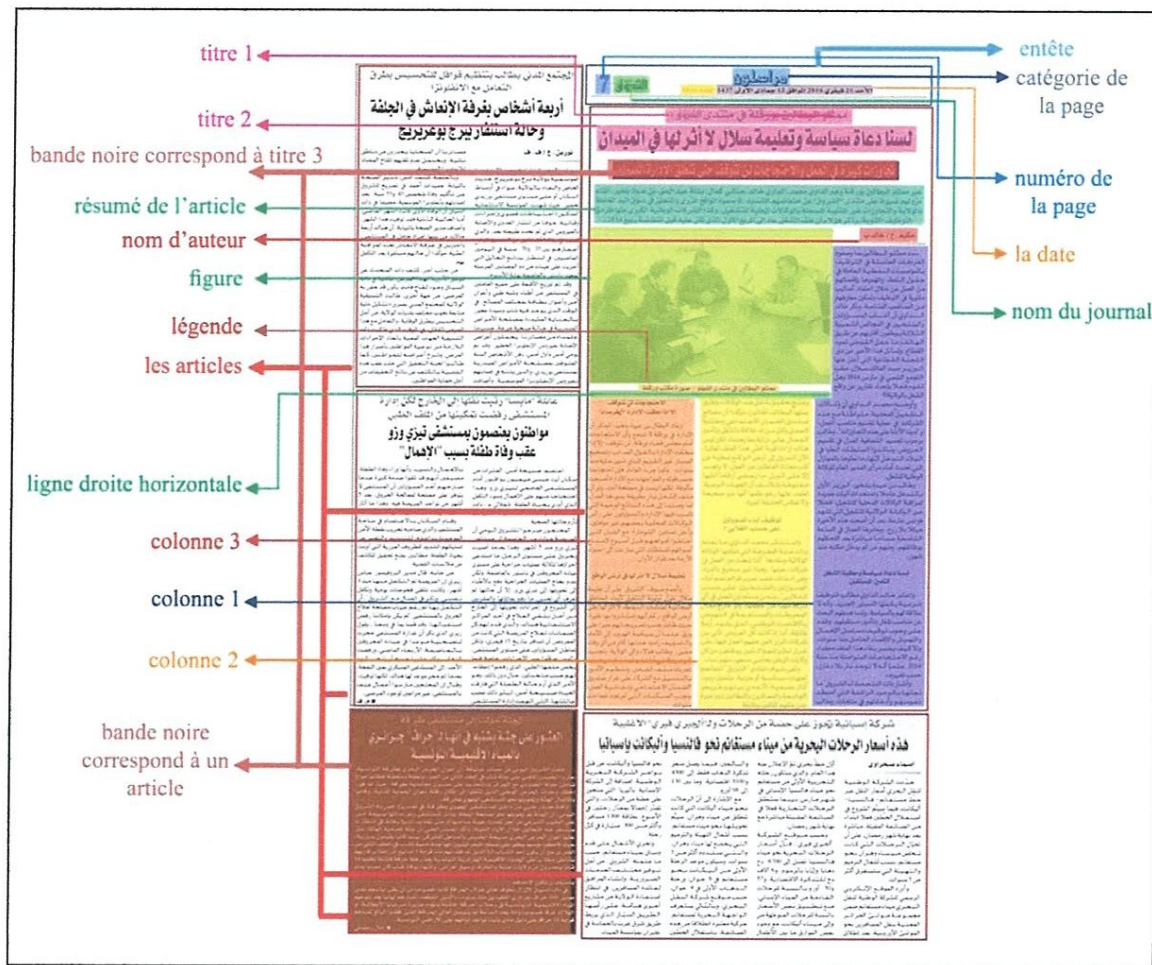


Figure 3.2 : Composants d'une page du journal Echrouk.

#### 4. Exposé de la démarche

La figure 3.3 illustre notre démarche pour la réalisation de notre système de reconnaissance de la structure logique des pages du journal. La démarche suivie inclue les parties suivantes :

- Extraction de la structure physique**, cette partie vise à analyser l'image de document traitée en vue de reconnaître sa structure physique. Elle regroupe ainsi deux phases : prétraitement visant à améliorer la qualité de l'image d'entrée, et segmentation permettant de séparer les entités physiques composant le document.
- Reconnaissance de la structure logique**, et elle regroupe deux phases aussi : l'étiquetage par des étiquettes logiques, les entités physique extraites précédemment, et la génération d'un fichier XML structuré représentant l'organisation logique du document.
- Génération d'un arbre dynamique**, représentant l'organisation hiérarchique du document.



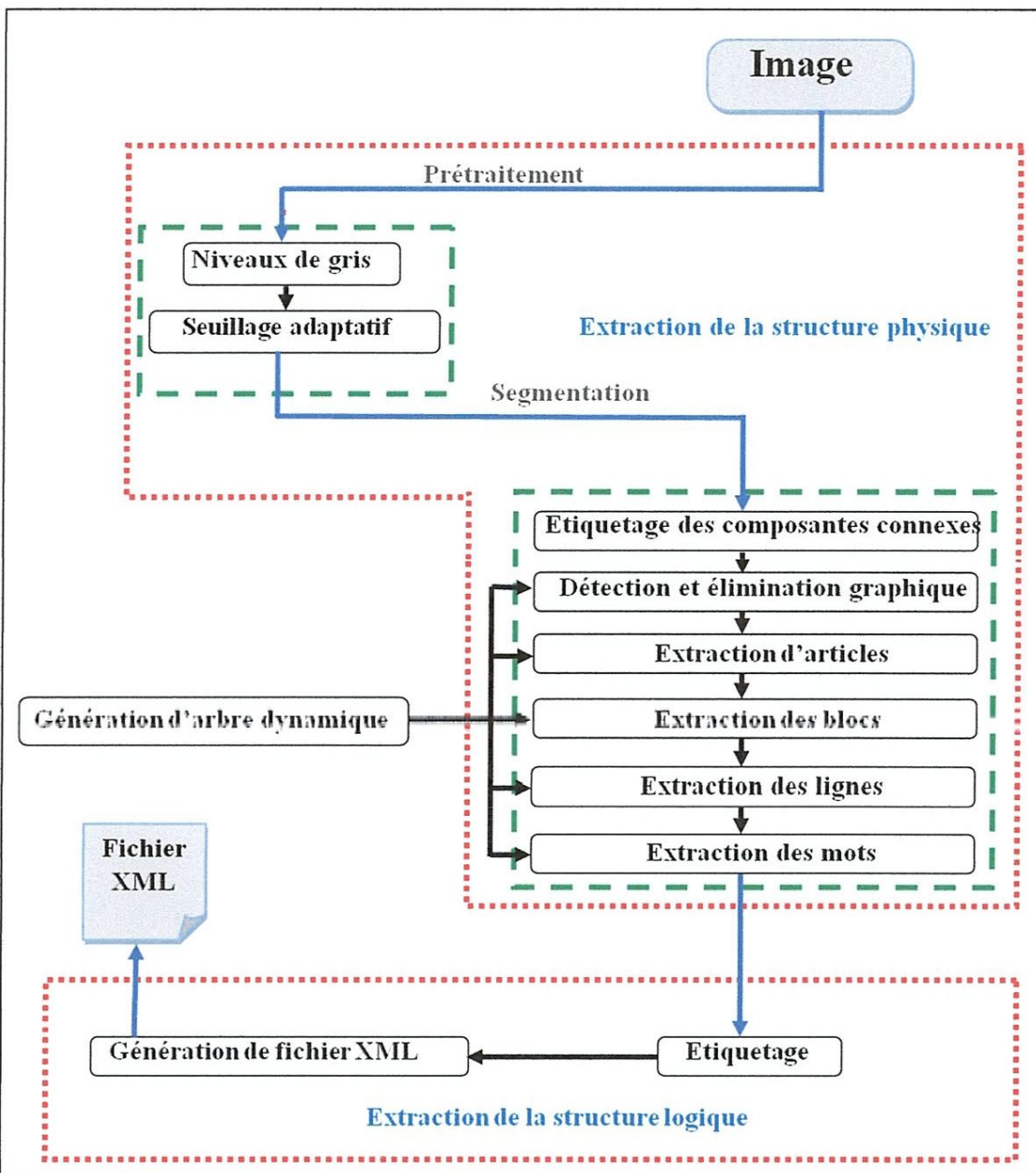


Figure 3.3 : Schéma illustre notre processus d'analyse et de reconnaissance de documents.

## 4.1. Extraction de la structure physique

Nous détaillons dans cette section les des deux phases incluses dans cette partie.

### 4.1.1. Prétraitement

Le prétraitement est une étape essentielle et nécessaire dans tous les systèmes de traitement, d'analyse, et de reconnaissance de documents. Elle regroupe un ensemble d'opérations visant à éliminer le bruit superposé aux données, de réduire les dégradations (s'il y a une variation d'illumination par exemple) et de ne préserver que les informations utiles de l'image. Cela permettra de préparer le terrain aux étapes suivantes dans le processus de reconnaissance dans le but d'avoir des résultats meilleurs et plus précises. Dans notre approche le prétraitement regroupe la transformation en niveaux de gris, et le seuillage.

#### 4.1.1.1. Transformation en niveau de gris

Le but de cette transformation est de construire une image pour le seuillage adaptatif que nous utiliserons dans l'étape suivante car cette méthode de binarisation n'est applicable que sur des images en niveaux de gris. Nous pouvons utiliser le pseudo-code suivant pour faire cette transformation :

##### Algorithme de transformation en niveaux de gris

```
Entrée : image en couleurs ( $I$ )
Sortie : image en niveaux de gris ( $I_g$ )
Début
Pour chaque pixel  $(x,y)$  de l'image  $I$  faire :
     $R \leftarrow$  la quantité de la couleur rouge du pixel  $(x,y)$  ;
     $V \leftarrow$  la quantité de la couleur verte du pixel  $(x,y)$  ;
     $B \leftarrow$  la quantité de la couleur bleue du pixel  $(x,y)$  ;
     $I_g(x,y) = (R+V+B)/3$  ;
Fin Pour
Fin.
```

#### 4.1.1.2. Seuillage

La binarisation (le seuillage) est la technique de classification la plus simple où les pixels de l'image sont séparés par un ou plusieurs seuils en deux classes : pixels de fond et pixels de l'avant-plan [Thi07]. Ainsi, le seuillage peut être global ou local. Les méthodes de seuillage globales sont généralement rapides et donnent des bons résultats pour des images de documents de bonne qualité. Lorsque l'image de document soit de mauvaise qualité, les méthodes globales deviennent inefficaces, et nous devons avoir recours aux techniques locales. Ces dernières sont généralement plus précises mais elles sont très lentes.

Nous avons ainsi proposé d'intégrer dans notre système une méthode de seuillage adaptatif. Cette dernière combine le seuillage dynamique ce qui tend à concentrer les informations de l'image dans des zones bien spécifiques qui seront donc très détaillées. Ces zones apparaîtront dans l'image binarisée comme des zones très texturées.

Le principe de base de cette méthode est de découper l'image en sous-images de manière judicieuse afin d'adapter le seuil à chacune d'entre elles. Le découpage se fait à partir de la variance des tons de la sous-image par rapport à l'histogramme (multimodal ou bimodal). Ensuite, elle cherche un seuil optimal qui maximise la variance interclasses ou qui minimise la variance intra-classe (le seuil d'Otsu [Ots79]) pour chaque sous-image.

### Algorithme de seuillage adaptatif

Entrée : image en niveaux de gris ( $I$ )

Sortie : image binaire ( $I_b$ )

Début

Décomposer l'image  $I$  en 4 ;

Pour chaque sous image  $I_i$  de  $I$  faire

$H \leftarrow$  l'histogramme de  $I_i$  ;

Si  $H$  est bimodal alors binariser  $I_i$  par la méthode d'Otsu ;

Sinon réitère l'algorithme récursivement pour  $I_i$  ;

Fin Pour

Fin.

### Algorithme de la méthode d'Otsu

Entrée : image en niveaux de gris ( $I$ )

Sortie : image binaire ( $I_b$ )

Début

Calculer  $h_2$  l'histogramme normalisé de  $I$  ;

Pour chaque niveau de gris  $S$  faire

$$q_1(S) = \sum_{i=0}^{S-1} h_2(i) ; \quad q_2(S) = \sum_{i=S}^{255} h_2(i) ;$$

$$\mu_1(S) = \frac{1}{q_1(S)} \sum_{i=0}^{S-1} h_2(i) \times i ; \quad \mu_2(S) = \frac{1}{q_2(S)} \sum_{i=S}^{255} h_2(i) \times i ;$$

$$\sigma_{inter}^2 = q_1(S) \times q_2(S) \times [\mu_1(S) - \mu_2(S)]^2 ;$$

Fin pour

$T \leftarrow$  le niveau de gris dont la variance est minimale ;

Pour chaque pixel  $(x, y)$  de l'image  $I$  faire :

Si  $I(x, y) < T$  alors  $I_b(x, y) \leftarrow$  noir ;

Sinon  $I_b(x, y) \leftarrow$  blanc ;

Fin Pour

Fin.

La figure 3.4 illustre le résultat de binarisation d'une page de journal en niveaux de gris en utilisant la technique de seuillage adaptatif.





(a) image en niveaux de gris

(b) résultat du seuillage adaptatif

Figure 3.4 : Binarisation d'une page de journal.

#### 4.1.2. Segmentation

La segmentation d'image consiste à partitionner l'image en plusieurs régions connexes. Comme nous avons vu dans les chapitres précédents, les trois approches de segmentation de document sont l'approche ascendante, l'approche descendantes, et l'approche mixte.

Dans notre travail, nous procédons à une segmentation mixte. Nous commençons par une segmentation ascendante qui part des pixels de l'image et les fusionne en composantes connexes. Ensuite les informations des composantes connexes sont utilisées pour séparer les composantes graphiques de la page (les figures, les bandes, les rectangles, et les lignes droites). Après, et pour diviser le texte de la page du journal en articles, nous utilisons une segmentation mixte basée sur l'analyse des profils de projections, l'algorithme de lissage RLSA, et l'étiquetage des composantes connexes. Finalement, nous appliquons une segmentation descendante pour diviser les articles de la page en blocs, les blocs en lignes et lignes en mots.

##### 4.1.2.1. Etiquetage des composantes connexes

L'étiquetage des composantes connexes consiste à fusionner les pixels noirs voisins dans une unité distincte, et nous utilisons pour cela la méthode d'agrégation des pixels.

Le résultat de l'étiquetage des composantes connexes est une image en couleurs dont chaque composante connexe est attachée par une couleur différente. La figure suivante illustre le résultat de l'étiquetage des composantes connexes de l'image présentée dans la figure 3.5.b.



Figure 3.5 : Etiquetage des composantes connexes.

#### 4.1.2.2. Détection et élimination des graphiques

La séparation entre les composantes graphiques et le texte est une étape importante avant la décomposition du texte de la page, et elle regroupe plusieurs étapes. Elle commence par la détection de l'entête/ pied de page, ensuite la détection des figures, puis la détection des bandes /rectangles/ lignes, et enfin l'élimination de tous les composantes détectées.

##### a) Détection de l'entête / pied de page

En tenant compte que l'entête et le pied de page sont toujours délimités en haut ou en bas par une ligne droite horizontale, la détection de l'entête et du pied de page repose sur la détection de ces lignes séparatrices. Pour détecter la ligne séparatrice de l'entête (respectivement du pied), on extrait la composante connexe la plus large qui se trouve à la partie haute (respectivement basse) de la page ( $1/6$  de la hauteur de la page). Si la largeur de cette composante est supérieur à (la largeur de l'image  $/2$ ) alors, cette composante est considérée comme la ligne séparatrice de l'entête (respectivement du pied de la page).

Finalement, toutes les composantes connexes qui se trouvent au-dessus de la ligne séparatrice de l'entête sont considérées comme des composantes de l'entête, et toutes les composantes se trouvant au-dessous de la ligne séparatrice du pied sont considérées comme des composantes du pied de page. On procède de la même manière pour l'extraction le pied de page. L'entête est coloré en jaune dans la figure 3.6.a.





(a) détection des graphiques

(b) élimination des graphiques

Figure 3.6 : Détection et élimination des graphiques

b) Détection des figures/ bandes / rectangles/ lignes

Pour la détection des figures, bandes, rectangles, et lignes droites, nous avons utilisé des formules et des conditions sur la taille des composantes connexes, le rapport entre la hauteur et la largeur, et la densité des pixels noirs dans la composante connexe. Chaque condition affecte l'autre.

On calcule d'abord pour chaque composante connexe  $CCi$ , le nombre  $F$  de pixels noirs dans  $CCi$ , la densité  $D$  des pixels noirs, et le rapport  $R$  entre la largeur  $L$  et la hauteur  $H$ . On calcule après la taille moyenne  $T$  des composantes connexes. Ces valeurs sont données par :

$$T = \text{taille moyennes des composantes connexes} \quad , \quad F = \text{nb de pixels noirs dans } CCi$$

$$R(CC_i) = \begin{cases} L/H & \text{si } L > H \\ H/L & \text{sinon} \end{cases} \quad , \quad D(CC_i) = \frac{F}{L \times H}$$

Ensuite, les éléments graphiques sont détectés en comparant ces statistiques avec des valeurs prédéterminées de seuils. La détection se fait dans l'ordre comme suit :

- **Les bandes:** une composante  $CCi$  est considérée comme bande si et seulement si :
  - $30 > R(CC_i) > 2$                       -  $D(CC_i) > 0.7$                       -  $F > 10 \times T$
- **Les lignes :** une composante  $CCi$  est considérée comme une ligne droite si et seulement si :
  - $R(CC_i) > 30$                               -  $D(CC_i) > 0.9$                       -  $F > 3 \times T$
- **Les rectangles :** une composante  $CCi$  est considérée comme rectangle si et seulement si :
  - $D(CC_i) < 0.05$                               -  $F > 5 \times T$
- **Les figures :** une composante est considérée comme figure si et seulement si :
  - $2 > R(CC_i) > 0$                               -  $D(CC_i) > 20$



Notons que les valeurs des seuils précédentes ont été déterminées par expérimentations. Chacun de ces éléments est représenté par une couleur distincte dans la figure 3.6.a. Les figures en bleu, les lignes droites en vert, les rectangles en rouge, et les bandes en violet.

**c) Élimination des graphiques**

Après la détection, on supprime les figures, les lignes, les rectangles, et les bandes tout en gardant le texte inclus dans les bandes. Ce texte sera donc coloré en noir (voir la figure 3.6.b).

**4.1.2.3. Segmentation du texte en articles**

Après la séparation entre le texte et les graphiques, l'étape suivante est la décomposition du texte en articles. La décomposition du texte en article est effectuée dans notre système en se basant l'algorithme RLSA. Pour segmenter le texte en articles on procède comme suit :

**a) Détection et élimination les titres**

Les titres sont les éléments clé des articles car il n'y a pas d'articles sans titres. La détection des titres se fait en tenant compte que les hauteurs des titres sont plus grandes que les hauteurs des lignes de texte simple. On peut extraire ces titres à l'aide d'une technique ascendante. Partant de l'étude bibliographique que nous avons effectuée, l'algorithme RLSA se révèle un bon choix. On applique ainsi un lissage RLSA horizontal sur l'image résultante des étapes précédentes pour éliminer les espaces entre les mots d'une même ligne de texte, et un lissage RLSA vertical afin de relier les signes diacritiques aux mots correspondants. Les seuils de lissage horizontal et vertical sont fixés par expérimentations égale à 20 et 25 respectivement. En effet, les caractères des grands titres sont toujours plus grands que ceux des lignes de texte simple (et des petits titres) et dans ce cas-là le seuil du lissage RLSA horizontal précédent n'est pas suffisant pour relier les mots d'un grand titre. Pour remédier à ce problème, nous appliquons un deuxième lissage RLSA horizontal avec un seuil plus grand (seuil = 30) uniquement sur les parties de l'image contenant des grands titres probables. Ces derniers sont composés des composantes connexes dont la hauteur est plus grande que  $(1.5 \times$  la hauteur de texte la plus présente dans le document). Nous appliquons ensuite un autre étiquetage des composantes connexes sur l'image lissée par RLSA. Comme les mots d'une même ligne de texte (simple ou titre) deviennent connectés, chaque ligne de texte est considérée comme une composante connexe à part. Une composante connexe est considérée comme un titre si sa hauteur est plus grande que  $(1.2 \times$  la hauteur de texte la plus présente dans le document), sinon c'est une ligne de texte simple.

Cependant, cette condition nous a fourni uniquement les titres principaux, pas tous les titres. Il faut donc recourir à d'autres critères afin d'ajouter les autres titres. Pour cela nous

avons pensé d'intégrer deux critères : la taille de la composante connexe et sa position par rapport aux titres principaux, ici les autres titres sont extraits à partir des profils de projection.

Notons  $L1$  et  $L2$  les lignes de texte qui se trouvent respectivement au-dessus et de dessous d'un titre principal  $T$ . Mettons :

- $(x1, y1)$  ← les coordonnées du coin haut gauche de  $L1$
- $(x2, y2)$  ← les coordonnées du coin bas droit de  $L1$
- $(xx1, yy1)$  ← les coordonnées du coin haut gauche de  $T$
- $(xx2, yy2)$  ← les coordonnées du coin bas droit de  $T$

La ligne de texte  $L1$  est considérée comme titre si elle vérifie les conditions suivantes :

- $34 < \text{hauteur de } L1$  ,  $xx1 - x2 < 25$
- $(y1 - yy1 < 100)$  et  $(y2 - yy2 < 100)$
- $(yy1 - y1 > 2)$  et  $(yy2 - y2 > 2)$  ou  $(y1 - yy1 > 2)$  et  $(yy2 - y2 > 2)$

On procède de la même manière pour  $L2$  en mettant  $(x1, x2)$  les coordonnées du coin haut gauche de  $T$ ,  $(x2, y2)$  les coordonnées du coin bas droit de  $T$ ,  $(xx1, yy1)$  les coordonnées du coin haut gauche de  $L2$ ,  $(xx2, yy2)$  les coordonnées du coin bas droit de  $L2$ .

Le résultat de suppression des titres de l'image de la figure 3.6.b est illustré par la figure 3.7.a.

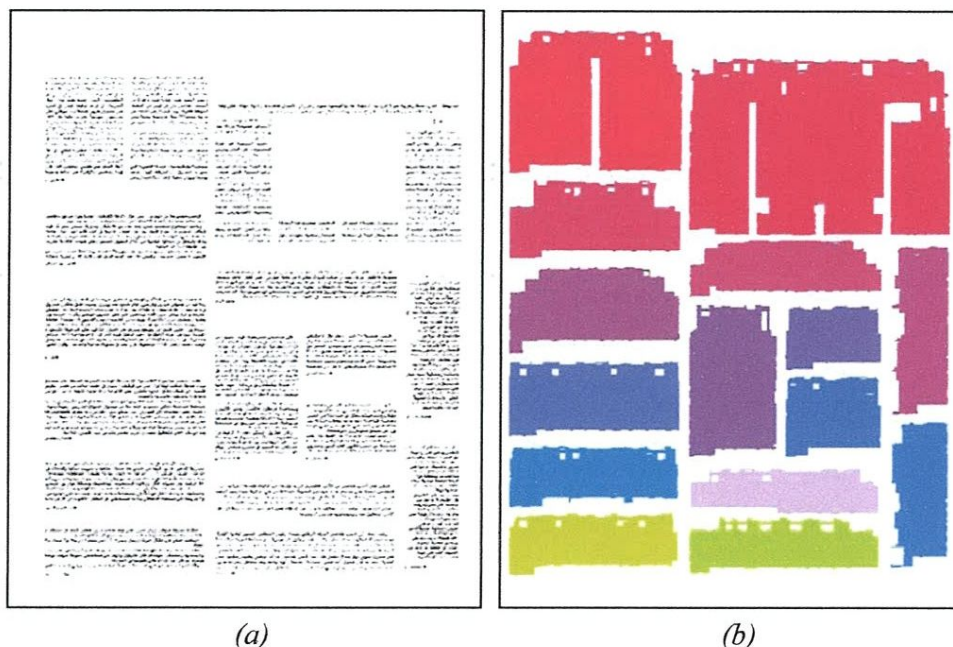


Figure 3.7 : (a) Elimination des titres, (b) étiquetage des composantes connexes après un lissage RLSA vertical

#### b) Lissage par RLSA vertical et étiquetage des composantes connexes

Après la restauration des titres sur l'image lissée précédente, on applique un deuxième lissage RLSA vertical avec deux valeurs de seuils différentes : 150 pour les pixels des titres,



et 30 pour les pixels des lignes de texte simple. Cela a pour effet de connecter les titres et lignes de texte d'un même article. L'étiquetage des composantes connexes permet ensuite de localiser les différents articles de la page (Figure 3.7.b).

**c) Extraction d'articles à partir de l'image binarisée**

Finalement, à partir des coordonnées des composantes connexes dans l'image binarisée (Figure 3.4.b), on extrait les différents articles. Notons que chaque composante est considérée comme un article à part (voir la Figure 3.8.a).



(a) segmentation en articles

(b) Segmentation en blocs

**Figure 3.8 : Segmentation de la page.**

**4.1.2.4. Segmentation des articles en blocs**

Après la séparation entre les articles, l'étape suivante est la décomposition de chaque article en blocs. Ainsi, on distingue deux types de blocs de texte, le bloc d'entête représentant les titres, et les colonnes de texte. La décomposition d'article en blocs est effectuée dans notre système en se basant sur les profils de projections horizontales et verticales. Cette méthode consiste à calculer le nombre de pixels noirs accumulés dans les directions horizontale ou verticale, pour identifier les emplacements de séparation.

Les étapes suivies pour la division du texte d'un article en blocs sont les suivantes :

**a) Extraction du bloc d'entête**

Les projections horizontales et verticales sont appliquées sur chaque article résultant de l'étape précédente dans le but d'identifier les titres. Ainsi le bloc d'entête est caractérisé par sa position en haut de chaque article, et de sa par disposition sur la largeur de l'article entier.



L'histogramme des projections horizontales est d'abord obtenu en calculant le nombre de pixels noirs dans chaque ligne de l'image. L'histogramme des projections horizontales correspondant sera constitué de pics et de vallées. Les vallées représentent des espaces de séparation entre des lignes du texte.

Ensuite, on calcule l'histogramme des projections verticales sur la partie de l'article délimitée par une vallée de l'histogramme des projections horizontales et la fin de l'article.

Les vallées dans l'histogramme des projections verticales correspondent aux espaces de séparation entre les colonnes de texte. Le calcul de l'histogramme des projections verticales est répété pour chaque vallée de l'histogramme des projections horizontales, se trouvant au premier tiers de l'article (car comme a dit l'entête se trouve toujours en haut de l'article). La vallée considérée comme séparateur entre le bloc d'entête et le reste de l'article est celle avec laquelle le nombre de colonnes trouvé à partir des projections verticales soit maximal. Le bloc d'entête sera donc la partie de l'article comprise entre le début de l'article et ce séparateur.

#### **b) Extraction des colonnes**

Les colonnes, sont obtenues à partir de l'histogramme des projections verticales de la partie de l'article au-dessous du bloc d'entête. Les vallées de cet histogramme constituent les espaces de séparation entre les articles.

Le résultat de segmentation des articles en blocs est illustré dans la figure 3.8.b.

#### **4.1.2.5. Segmentation des blocs en lignes**

L'étape suivante dans l'extraction de la structure physique est la décomposition de chaque bloc en lignes. Pour ce faire, nous avons réutilisée la technique de segmentation en lignes implémentée dans [BF15]. Cette technique repose sur l'application d'une projection horizontale sur chaque bloc séparément afin d'extraire les lignes qui le composent. Elle consiste à:

- a) **Calcul de l'histogramme des projections horizontales du bloc** : comme précédemment.
- b) **Extraction des minima locaux** : si on considère l'histogramme des projections comme une fonction discrète  $f(x)$ , pour  $k$  allant de 1 jusqu'à la taille de l'histogramme-1,  $k$  est considéré comme un minimum local si  $f(k-1) > f(k)$  et  $f(k+1) > f(k)$ .
- c) **Filtrage des minima locaux** : en deux passes :  
Dans la première passe, on élimine les minima locaux ayant une largeur plus grande qu'un seuil donné. Le seuil est choisi comme la largeur du plus long *minimum local* /2. L'espace entre deux minima successifs correspond à la hauteur d'une ligne du texte. Dans la deuxième passe, on enlève l'un des deux minima très proches, car la hauteur

de lignes du texte est presque la même dans tout le bloc. Pour ce faire, on calcule d'abord la distance moyenne *distanceMoyen* entre deux minima successifs. Si la distance entre deux minima successifs est  $< 2 * (distanceMoyen) / 3$ , le plus long d'entre eux sera enlevé. Les minima restants correspondent aux zones de séparation entre les lignes du texte.

- d) **Résolution de conflits** : en attribuant les pixels noirs existant dans les zones séparatrices à la ligne du texte la plus proche par analyse de proximité.

La figure 3.9 illustre le résultat de segmentation en lignes :

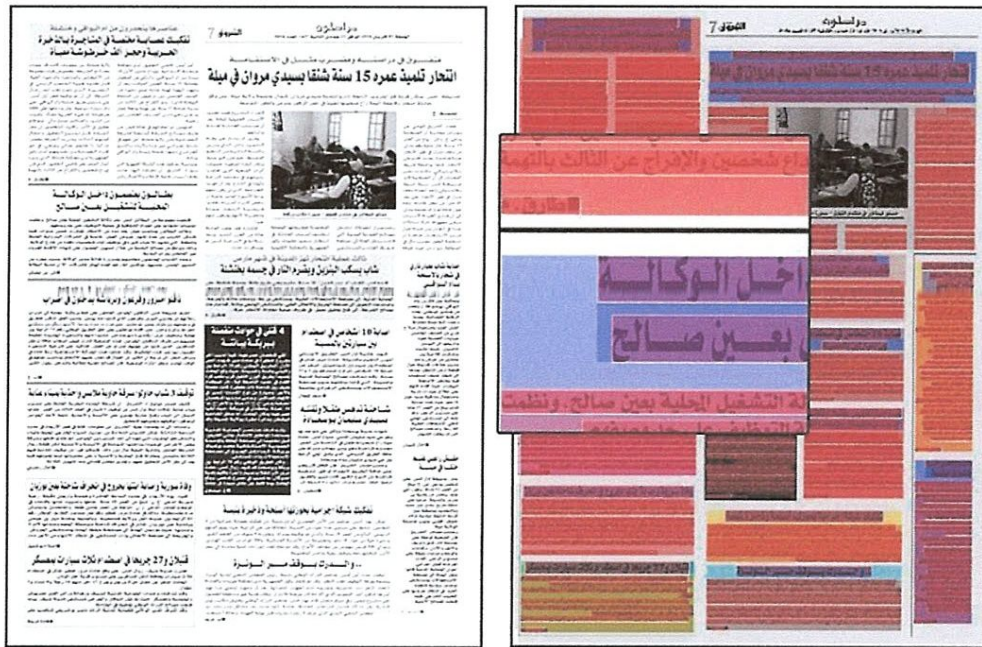


Figure 3.9 : Segmentation des blocs en lignes.

#### 4.1.2.6. Segmentation des lignes en mots

L'étape finale de segmentation est la décomposition de chaque ligne en mots. Pour ce faire, l'étiquetage des composantes connexes et le lissage RLSA sont appliqués sur chaque ligne séparément afin d'en extraire les mots qui le composent. Ainsi la segmentation d'une ligne en mots s'effectue comme suit:

- Lissage RLSA vertical pour interconnecter les points diacritiques aux mots. Le seuil de lissage est fixé égale à 30
- Lissage RLSA horizontal avec un seuil égal à 2 pour connecter les sous-mots (PAW en anglais pour Part of Arabic Word) d'un même mot.
- Etiquetage des composantes connexes pour chaque ligne.
- Filtrage pour améliorer la séparation entre les mots. On élimine les espaces ayant une largeur plus petite qu'un seuil donné. Le seuil est choisi par rapport à largeur et la hauteur de chaque ligne. Il est donné par les conditions suivantes :



- Si  $0 > largeur > 250$  et  $50 > hauteur > 0$  alors seuil  $\leftarrow 4$ ;
- Si  $250 > largeur > 500$  et  $50 > hauteur > 70$  alors seuil  $\leftarrow 5$  ;
- Si  $500 > largeur$  et  $70 > hauteur$  alors seuil  $\leftarrow 8$  ;

## 4.2. Extraction de la structure logique

Cette partie a pour objectif l'étiquetage de toutes les composantes extraites dans la première partie du système, la génération d'un arbre dynamique représentant l'organisation hiérarchique des éléments logique et physique du page de journal, et la génération d'un fichier XML représentant la structure logique de la page.

### 4.2.1. Etiquetage

L'étiquetage consiste à reconnaître tous les composants de la page de journal. Les composants logiques considérées dans notre système sont : les articles, l'entête de la page, le pied de la page, les colonnes (avec leur numéro), les titres (avec leur niveau), le nom d'auteur de chaque article, les figures, les lignes de textes, et les légendes des figures.

#### 4.2.1.1. Etiquetage de l'entête et le pied de page

L'entête et le pied de la page sont les premiers éléments identifiés lors de l'extraction de la structure physique. L'étiquetage de ces éléments c'est l'attribution des étiquettes « Entête de page » et « Pied de page » aux zones d'entête et de pied localisées dans l'image.

#### 4.2.1.2. Etiquetage des articles

Les articles sont reconnus lors de la première phase d'extraction de la structure physique. Dans cette étape il ne nous reste que la numérotation de ces articles par rapport à leur position dans la page en tenant compte que les pages de journaux arabes se lisent de droite à gauche. L'article le plus haut à droite est étiqueté « *Article 1* », le suivant « *Article 2* », etc.

#### 4.2.1.3. Etiquetage des blocs

Il y a le bloc d'entête représentant la partie où il y a les titres et les autres blocs représentant les colonnes du texte, et aussi l'étiquetage ici est basé sur la position des blocs.

- **Bloc de titres** : le bloc le plus haut de l'article dont la disposition est sur la largeur de l'article est étiqueté « *bloc de titres* ».
- **Colonnes** : les autres blocs sont des colonnes. Ainsi l'étiquetage des colonnes consiste à les assigner l'étiquette « *Colonne* » en plus d'un numéro décrivant l'ordre de disposition des colonnes. Comme nous traitons des pages de journal en langue arabe, les colonnes sont numérotées de droite à gauche. Ainsi la colonne la plus à droite de chaque article est étiquetée « *Colonne 1* », et la colonne qui suit « *Colonne 2* » et ainsi de suite.



#### 4.2.1.4. Etiquetage des autres entités logiques

- **Titres** : les titres sont étiquetés par leur niveau dans le bloc de titres. En classant les lignes de ce bloc par ordre décroissant selon leur position. Le « *Titre 1* » correspond à la ligne la plus haute du bloc de titres, et ainsi de suite.
- **Lignes du texte** : les lignes du texte de chaque colonne ont été extraites précédemment dans la première partie. Il ne reste que dès les étiqueter en les attribuant un numéro correspondant à leur ordre dans le bloc (du haut en bas). Ainsi la première ligne dans chaque colonne est étiquetée « *Ligne 1* », et ainsi de suite.
- **Nom d'auteur** : le nom de l'auteur d'un article se trouve souvent après les titres, et avant le corps (le texte principal) de l'article, dans la première ligne de la première colonne. Il peut être également trouvé après le texte de l'article, dans la dernière ligne de la dernière colonne. La position du nom d'auteur est trouvée en examinant la condition suivante :  
Si la première ligne de la première colonne a une largeur inférieure ou égale à un certain seuil, cela indique la présence du nom d'auteur. Le seuil est choisi égale à la  $(2 \times \text{largeur de la ligne} / 3)$ . Si ce n'est pas le cas on vérifie si le nom d'auteur est à la fin de l'article. Si la largeur de la dernière ligne de la dernière colonne est inférieure ou égale au seuil précédent, alors cette ligne est le nom d'auteur. Il ne reste qu'étiqueter la ligne correspondante au nom d'auteur par l'étiquette « *Auteur* ».
- **Légendes** : les légendes ou les titres des figures peuvent être trouvées dans les pages de journaux souvent au-dessous des figures. Dans notre système, pour la détection de la légende d'une figure, nous examinons la ligne du texte qui se trouve juste au-dessous de la figure. Si la largeur de cette ligne est inférieure à la largeur de la figure, et l'espace entre cette ligne et la ligne qui la suit est supérieure ou égale à un certain seuil, cela indique la présence d'une légende. Le seuil est choisi égal à  $(\text{l'espace entre la ligne } 3 \text{ et la ligne suivante}) \times 2$ . Les légendes sont étiquetées par l'étiquette « *Légende* ».

#### 4.2.2. Génération d'un fichier XML

Cette étape est très importante dans notre système car elle résume toute la structure logique extraite dans une forme bien organisée et structurée. Nous avons choisi le format XML (eXtensible Markup Language) parce qu'il est très utilisé dans le milieu de la Gestion Electronique Documentaire et aussi permet facilement l'échange des résultats. Cependant, nous faisons correspondre à chaque page de journal, un fichier d'annotation XML correspondant. Chaque fichier d'annotation contient les informations suivantes sur l'image:

- Nom, format, hauteur, et largeur de l'image, en plus du nombre d'articles dans la page.
- La position de l'entête et la position du pied de la page. Pour chaque article :
  - La position de l'article dans la page, son auteur, le nombre de colonnes, le nombre de figures.

- La position du bloc d'entête, et les niveaux de titres existants.
- La position de chaque titre, son niveau, et le nombre de mots dans le titre.
- La position de chaque figure, et la légende de celle-ci si elle existe.
- La position de chaque colonne, son numéro, et le nombre de lignes dans la colonne.
- La position de chaque ligne, son numéro, et le nombre de mots dans la ligne.
- La position de chaque mot, et son numéro dans la ligne ou le titre.

Un exemple de fichier XML et sa forme est le suivant:

```
<? xml version="1.0" encoding="UTF-8"?>
<PAGE nom="k" type="jpg" hauteur="2000" largeur="3080" nb_Articles="6">
  <En_tete box="797;0;1887;239">
    <ARTICLE num="1" BOX="811;1197;1591;1477" nb_Colonne="3" nb_Figures="1">
      <Bloc_Titres BOX="871;1197;1528;1293" nb_Titres="1">
        <Titre num="1" box ="926;1197;1477;1231" nb_mots="10">
          <Mot num="1" box ="1006;1197;1064;1231"/>
          <Mot num="2" box ="1106;1197;1189;1231"/>
          ..... </Titre>
        </Bloc_Titres>
      <FIGURE num="1" box="....." legende="....."/>
      <Colonne num="1" box="....." nb_Lignes="27">
        <LIGNE num="1" box="....." nb_Mots="12" />
        <Mot num="1" box ="1006;1197;1064;1231">
          .....
```

### 4.3. Génération d'un arbre de composants de la page

Cette partie de l'application permet de construire un arbre dynamique qui s'enrichit et s'organise à chaque étape de traitement dans notre système. Cet arbre donne toutes les informations de la page sous une forme dynamique et bien organisée, structurée et hiérarchique, et il peut être considéré comme un outil de navigation à l'intérieur de la page. Cependant, l'arbre de composants de la page permet de localiser facilement en un seul clique n'importe quel élément physique ou logique de la page (les titres, les articles, les lignes, les auteurs, les légendes, les figures,...etc.).

## 5. Conclusion

Dans ce chapitre nous avons exposé les modèles que nous avons choisis pour résoudre les problèmes de la reconnaissance de la structure logique des pages du journal arabe. Nous avons essayé à travers ce chapitre de présenter notre approche en détails afin de montrer l'effort que nous avons réalisé pour résoudre les problèmes rencontrés dans notre travail dans le but d'atteindre les objectifs visés.

Dans le prochain chapitre, nous présenterons l'implémentation de notre conception, avec un scénario complet d'utilisation du système.



---

---

# **Chapitre 4.**

## Implémentation et résultats

---

---

## 1. Introduction

Dans le but de résoudre le problème de la reconnaissance de la structure logique des pages du journal, et pour atteindre les objectifs de notre projet, nous aborderons dans ce chapitre la réalisation pratique de notre système, en appliquant la conception proposée dans le chapitre précédent.

Nous y décrivons l'environnement de travail et les outils nécessaires, le corpus d'images utilisé dans les tests, l'architecture générale de notre application ainsi que ses interfaces principales. Nous exposons également les différentes étapes de traitement incluses dans notre application avec les diverses fenêtres. Nous présentons finalement quelques résultats des tests et nous discutons ces résultats tout en essayant d'expliquer les raisons des bonnes et mauvaises performances.

## 2. Environnement de développement

On désigne par l'environnement de développement tous les moyens matériels et logiciels utilisés pour l'implémentation de notre application.

### 2.1. Environnement matériel

Le matériel utilisé constitué d'un PC dont les caractéristiques sont présentées dans le tableau suivant:

<b>Modèle</b>	<b>PC portable</b>
<b>Processeur</b>	<b>Intel core i3 duo (2 .10 GHZ)</b>
<b>RAM</b>	<b>4 Go</b>
<b>Disque dur</b>	<b>80 Go</b>

*Tableau 4.1 : Caractéristiques du matériel utilisé.*

### 2.2. Environnement logiciel

Notre application a été développée en langage de programmation Java sous Eclipse version 3.5 Galileo.

#### 2.2.1. Eclipse

*Eclipse Public License* (ou EPL) est une licence libre à copyright faible utilisée à l'origine par l'environnement de développement intégrée (EDI) Eclipse. Elle succède à la *Common Public License* et supprime certains termes relatifs aux litiges de brevet. L'*Eclipse Public License* est conçue pour être plus favorable aux entreprises souhaitant faire du logiciel propriétaire que ne l'est la GNU (*General Public License*) car elle n'impose pas de devoir contributif. Comme tout logiciel libre, un programme sous licence EPL peut être utilisé, modifié, copié et redistribué librement. Le code source des versions dérivées (modifiées) doit être divulgué sous la même licence. En revanche, un logiciel propriétaire peut inclure un programme sous licence EPL, à partir du moment où il n'est pas un dérivé de ce programme.



Par exemple, il peut l'utiliser en tant que bibliothèque. L'EPL est approuvée par l'*Open Source Initiative* et reconnue comme libre par la *Free Software Foundation* [Jea09].

### 2.2.2. Java

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses Caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprise et mobiles [Dou10].

## 3. Corpus de documents utilisé

Comme nous avons dit précédemment, nous nous intéressons dans le présent travail aux pages de journaux numérisées. Le journal considéré dans notre étude est le quotidien algérien Echorouk. Ce journal existe sous forme électronique en format PDF et peut être téléchargé depuis Internet à partir du lien : [www.echoroukonline.com](http://www.echoroukonline.com). Cependant, notre corpus utilisé tout au long de notre travail est composé de trente images résultantes de la conversion des pages d'un journal téléchargé au format PDF en format image. Ces images sont des images couleurs en format *jpg* et elles sont dites idéales c'est-à-dire ne contiennent pas de bruit.

La conversion du PDF en images a été effectuée à l'aide du convertisseur *Pdf2Jpg*. Ce dernier est un convertisseur gratuit accessible en ligne à partir du lien : <http://pdf2jpg.net/fr/>. On lui fournit un seul fichier en PDF correspondant à un journal, et il nous renvoie plusieurs images en *jpg* dont chacune correspond à une page du journal.



(a)

(b)

Figure 4.1 : Exemples des pages du journal Echorouk utilisées dans notre corpus de test.

## 4. Architecture et fonctionnalités du système

L'application que nous avons développée contient plusieurs fonctionnalités qui nous permettent de reconnaître la structure logique des pages du journal, en visualisant les résultats intermédiaires à chaque étape de traitement. Dans cette section, nous expliquons l'architecture et les fonctionnalités de notre système en détails.

### 4.1. Description de l'application

A ce moment, on va présenter quelques interfaces, les modules principaux de l'interface.

La figure ci-dessous illustre l'interface d'accueil de notre application.

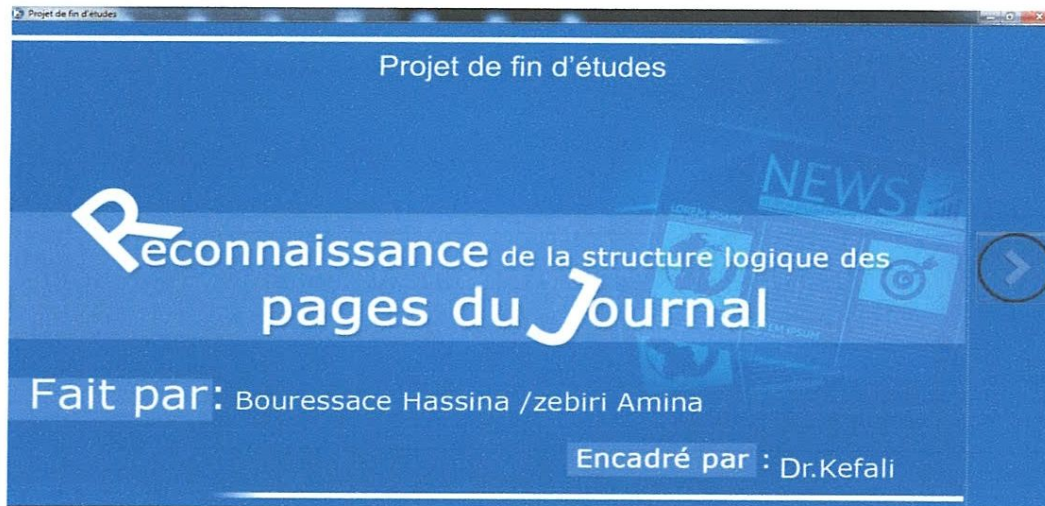



Figure 4.2 : Interface d'accueil de notre application.

En cliquant sur le bouton , l'interface principale de l'application s'affiche. Elle contient : une barre de menus, une barre d'outils, une zone d'affichage de l'image d'origine et de l'image traitée, un arbre dynamique, en plus de tous les boutons de l'application.

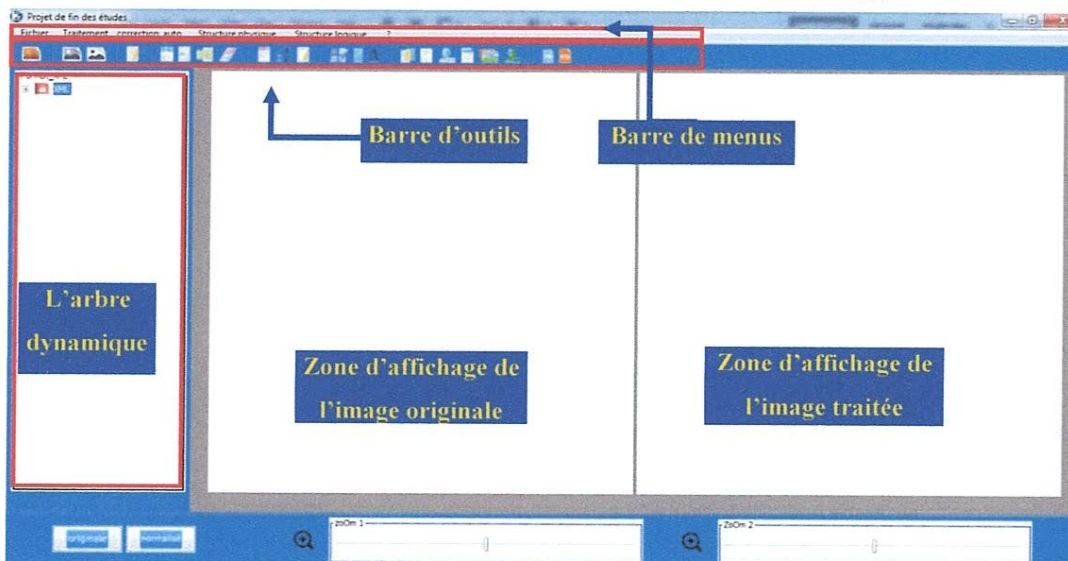


Figure 4.3 : Interface principale de notre application.



La troisième interface de notre application est la fenêtre chargée d'afficher la structure logique finale d'une page de journal, sous forme d'un fichier XML. La figure 4.4 montre un exemple.

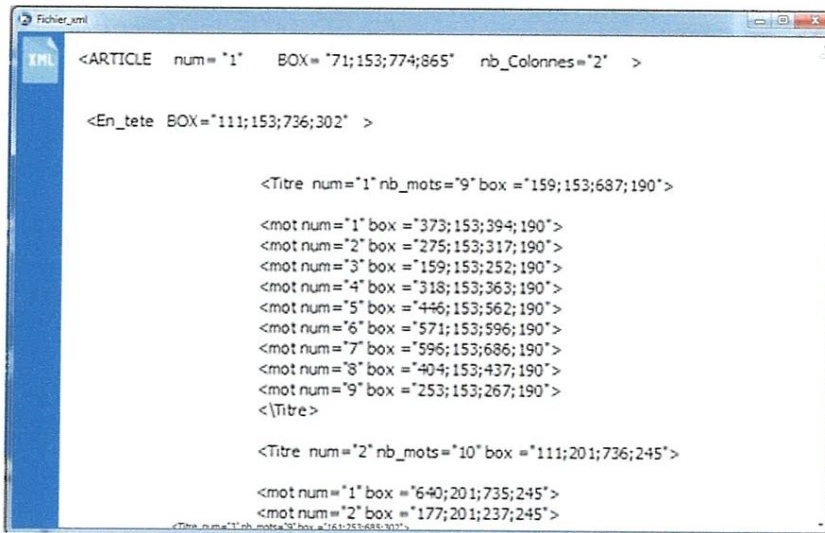


Figure 4.4 : Interface d'affichage du fichier XML.

Les modules principaux de notre application sont récapitulés par la figure suivante :

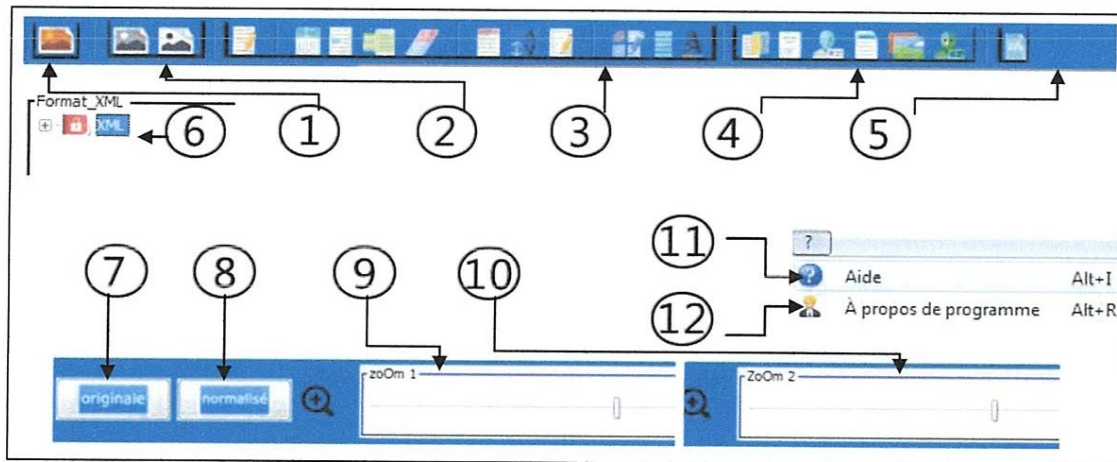


Figure 4.5: Les modules principaux de l'application.

Ces modules sont les suivants :

- 1:** le bouton qui permet le chargement de l'image à traiter
- 2:** regroupe les boutons permettant le prétraitement : la transformation en niveaux de gris, et le seuillage adaptatif.
- 3:** regroupe les boutons permettant la segmentation de l'image prétraitée : l'étiquetage des composants des connexes, la détection de l'entête / pied de page, la détection des figures, la détection des bandes /rectangles/ lignes, l'élimination des bandes /rectangles/ lignes, la détection et l'élimination des titres, l'étiquetage des composants des connexes de l'image lissée par RLSA, la segmentation en articles, blocs, lignes et en mots.

**4 :** contient des boutons pour l'étiquetage logique des entités physiques extraites précédemment (articles, colonnes, lignes, mots, entête, titres, figures, auteurs, légendes).

**5 :** bouton chargé de la génération d'un fichier XML.

**6 :** l'arbre dynamique des éléments extraits.

**7 et 8 :** boutons permettant de redimensionner les images (originale et traitée) à la taille originale et normalisée respectivement.

**9 et 10 :** ascenseurs permettant d'agrandir et amoindrir les images affichées.

**11 et 12 :** l'aide et l'à propos de l'application.

## **4.2. Scénario d'utilisation complet**

Comme tout système informatique, notre système fonctionne selon un scénario spécifique. Nous déroulons dans cette section une utilisation complète, qui explique le fonctionnement de notre système.

### **4.2.1. Chargement d'une image**

Nous commençons par le chargement d'une image correspondante à une page numérisée du journal Echorouk à partir du sous menu « Ouvrir » du menu « Fichier » ou en cliquant directement sur le bouton « Ouvrir » de la barre d'outils. Une boîte de dialogue s'affiche et nous propose de sélectionner une image. (Figure 4.6.a).

### **4.2.2. Prétraitement de l'image chargée**

Après le chargement de l'image, vient l'étape de prétraitement. On commence par la transformation en niveaux de gris puis le seuillage adaptatif.

#### **4.2.2.1. Transformation en niveaux de gris**

L'image de la page est transformée en niveaux de gris et affichée dans la zone d'affichage de l'image traité (Figure 4.6.b).





(a) image chargée

(b) image en niveaux de gris

Figure 4.6 : Chargement et transformation de l'image en niveaux de gris.

#### 4.2.2.2. Seuillage adaptatif

L'étape suivante est le seuillage adaptatif appliqué sur l'image en niveaux de gris (Figure 4.7).



Figure 4.7 : Binarisation par seuillage adaptatif.

#### 4.2.3. Extraction de la structure physique

L'extraction de la structure physique regroupe plusieurs traitements. Ces traitements sont accessibles depuis le menu « Structure physique » ou à partir de la barre d'outils. On débute par l'étiquetage des composantes connexes jusqu'à la segmentation des lignes en mots.



#### 4.2.3.1. Etiquetage des composantes connexes

Comme nous avons expliqué dans le chapitre précédent, nous avons utilisé une méthode itérative d'étiquetage des composantes. Dans la figure suivante chaque composante connexe est étiquetée par une couleur distincte.

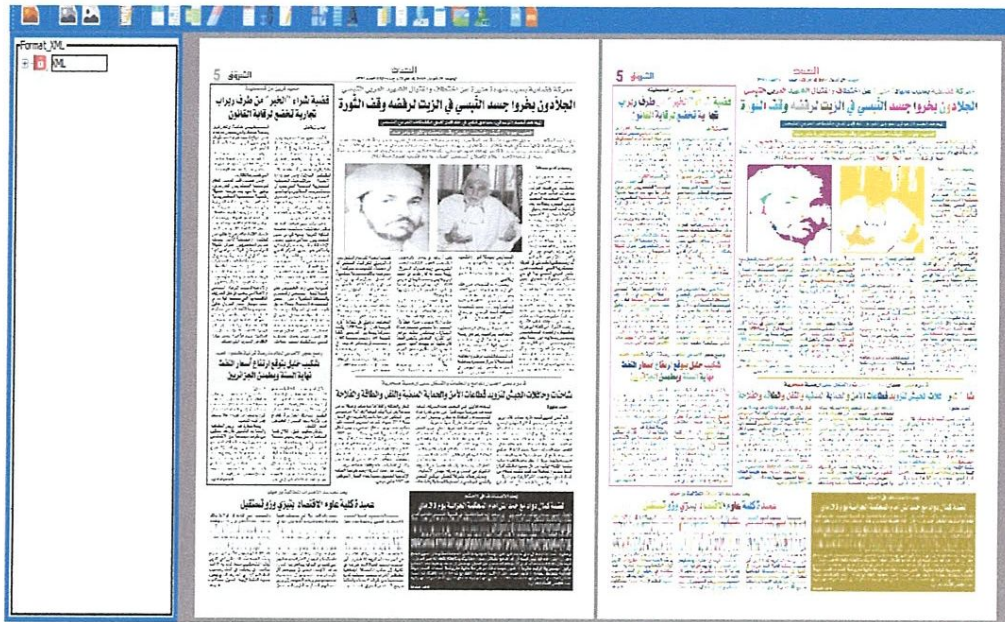


Figure 4.8 : Etiquetage des composantes connexes.

#### 4.2.3.2. Détection des éléments graphiques

L'étape suivante pour l'extraction de la structure physique est la détection des éléments non-textuels. Cette étape regroupe la détection de l'entête / pied de page, la détection des figures, et la détection des bandes /rectangles/ lignes s'ils existent. Chacun de ces éléments est étiqueté par une couleur distincte comme dans la Figure 4.9.



Figure 4.9 : Détection des éléments graphiques.



#### 4.2.3.3. Élimination des éléments graphiques

Après la détection, on élimine tous les éléments détectés dans l'étape précédente tout en gardant le texte inclus dans les bandes noires. Ce texte sera donc coloré en noir. La figure suivante illustre le résultat de cette étape :



Figure 4.10 : Elimination des éléments graphiques détectés auparavant.

#### 4.2.3.4. Détection et élimination des titres

Cette étape est très importante dans l'extraction de la structure physique. Elle consiste à détecter les titres et puis de les éliminer (Figure 4.11).



Figure 4.11 : Elimination des titres.

#### 4.2.3.5. Lissage par RLSA et étiquetage des composantes connexes

En appliquant l'algorithme RLSA sur l'image précédente, cette dernière est lissée en regroupant les composantes connexes proches (horizontalement et verticalement) ensemble en



une seule unité. Le résultat d'étiquetage des composantes connexes de l'image lissée par RLSA est illustré par la figure suivante. Chaque composante connexe ainsi obtenue représente un article (avec ses titres et son corps) et est étiquetée par une couleur différente.



Figure 4.12 : Lissage par RLSA et étiquetage des composantes connexes.

#### 4.2.3.6. Segmentation en articles

La segmentation en articles permet d'extraire les différents articles de la page. Le résultat est illustré par la figure suivante :



Figure 4.13 : Segmentation du texte en articles.

#### 4.2.3.7. Extraction des blocs, des lignes et des mots

Cette étape permet d'extraire les blocs de chaque article (bloc de titres et colonnes de texte), et pour chaque bloc on extrait les lignes et les mots (Figure 4.14).





Figure 4.14 : Extraction des blocs, des lignes et des mots de chaque article.

#### 4.2.4. Extraction de la structure logique

C'est la deuxième phase dans notre application, et elle inclue plusieurs traitements accessibles depuis le menu « Structure logique » ou depuis la barre d'outils. On commence par l'étiquetage des entités physiques extraites, jusqu'à la génération d'un fichier XML.

##### 4.2.4.1. Étiquetage des entités physiques

Cette étape permet d'assigner une étiquette logique à toute entité physique extraite dans la phase précédente. Dans la figure 4.15, nous pouvons remarquer sur chaque article les étiquettes suivantes : *Entête*, *Colonne* (1..3), *Titre* (1..6), *Auteur*, *Figure*, *Légende*, *Ligne*.



Figure 4.15 : Étiquetage logique des entités physiques extraites précédemment.



#### 4.2.4.2. Génération d'un fichier XML

L'étape finale c'est la génération automatique d'un fichier XML décrivant la structure logique du document. Cela est possible à partir la barre de menus ou de la barre d'outils. Le code XML généré est affiché dans nouvelle fenêtre (voir la figure suivante).

```

<ARTICLE num="2" BOX="820;276;1905;2202" nb_Colonne="3" >
<FIGURE num="1" BOX="821;705;1599;1159" >
<LENGENDE num="1" BOX="994;1165;1423;1188" >
<FIGURE>

<En_tete BOX="833;276;1904;650" >

<Titre num="1"nb_mots="10" box="1080;276;1647;319">

<mot num="1" box="1368;276;1372;319">
<mot num="2" box="1330;276;1378;319">
<mot num="3" box="1564;276;1646;319">
<mot num="4" box="1093;276;1176;319">
<mot num="5" box="1215;276;1281;319">
<mot num="6" box="1292;276;1321;319">
<mot num="7" box="1439;276;1556;319">
<mot num="8" box="1186;276;1210;319">
<mot num="9" box="1379;276;1427;319">
<mot num="10" box="1080;276;1085;319">
</Titre>

<Titre num="2" nb_mots="11" box="845;332;1861;411">

<mot num="1" box="895;332;979;411">
<mot num="2" box="1060;332;1111;411">
<mot num="3" box="1235;332;1339;411">
<mot num="4" box="1353;332;1503;411">
<mot num="5" box="1775;332;1860;411">
    
```

Figure 4.16 : Génération du Fichier XML.

#### 4.2.5. Génération d'un arbre de composants de la page

Notre application génère automatiquement un arbre des éléments détectés à chaque étape de traitement d'une manière dynamique. Cet arbre sert d'outil de navigation à l'intérieur de la page du journal traitée. A partir de cet arbre, on peut accéder et sélectionner n'importe quel composant physique ou logique de la page (un entête, une colonne, un titre, un nom d'auteur, une figure, une légende, une ligne) en cliquant dessus sur l'arbre (Figure 4.17).



Figure 4.17 : Génération d'un outil de navigation à l'intérieur de la page de journal.



## 5. Expérimentations et résultats

Dans la discipline de l'informatique, tout système (logiciel) doit être passé par la phase d'expérimentations, pour définir les performances et les limites du système.

Pour l'expérimentation de notre système, nous avons utilisé toutes les images du corpus de test. Tout d'abord, les étiquettes logiques des différents éléments des images de test ont été établies manuellement. Ensuite, nous avons appliqué notre système sur toutes les images de test afin de les étiqueter automatiquement. Les résultats d'étiquetage automatique de chaque image sont comparés aux étiquettes réelles (établies manuellement) dans le but de déterminer le taux de reconnaissance. Afin de vérifier la généralité de notre système, nous avons essayé de varier les images de test, pour qu'elles contiennent un nombre varié d'articles, avec des dispositions différentes, et aussi contiennent ou non des lignes droites, des bandes, des figures, etc. Le tableau 4.2 présente un exemple de comparaison entre les étiquettes réelles et les étiquettes extraites par notre système de l'image de la figure 4.1.b.

Etiquette \ Images	Image détectés automatiquement	Image détectés manuellement
Entête	1	1
Pied de page	0	0
Figures	2	2
Bandes noires	3	3
Bordures	1	1
Lignes droites horizontales	1	1
Articles	6	6
Blocs	6	6
Lignes	318	318
mots	1572	1306
Titres	14	14
Colonnes	15	15
Légendes	0	0
Auteurs	6	6

*Tableau 4.2 : Etiquettes détectées manuellement et automatiquement de l'image de la figure 4.1.b.*

Le tableau 4.3 résume le taux de reconnaissance moyen de chaque entité logique :

Etiquette \ taux	Taux de bonne reconnaissance	Taux de mauvaise reconnaissance
Entête	95%	5%
Pied de page	80%	20%
Figures	90.20%	09.80%
Bandes noires	85%	15%
Bordures	70.55%	29.45%
Lignes droites horizontales	90.87%	09.13%
Articles	88%	12%
Blocs	80%	20%
Lignes	78.85%	21.15%
Mots	55%	45%
Titres	76.20%	23.80%
Colonnes	90%	10%
Légendes	88.10%	11.90%
Auteurs	78%	22%

*Tableau 4.3 : Résultats du test.*

A partir du tableau précédent, nous pouvons remarquer que le système a réussi à reconnaître la plupart des entités logiques existantes, ce qui égale à un taux de reconnaissance de **87.57 %**.

Cependant, les problèmes ont été rencontrés avec des images contenant:

- Des titres très petites, dans ce cas le système considère ces titres comme des lignes de texte simple;
- Des espaces irréguliers entre les mots, compliquant la segmentation entre les mots d'une même ligne ;
- Lorsque l'espace entre le titre et le paragraphe qui suit soit petit et égale à l'espace entre les lignes ;
- Lorsque le d'auteur s'agit d'une abréviation (les deux premiers lettres) ;
- Une figure très lumineuse, ou qui contient d'écriture ;
- La forme des bandes noires est non-rectangulaire ;
- Le pied de page n'est pas délimité par une ligne de séparation.

En général l'application a donné de bons résultats. Ces résultats sont à notre avis très encourageants considérant la structuration riche et complexe des pages de journaux.

## 6. Conclusion

Dans ce chapitre, nous pouvons dire que nous avons présenté d'une manière détaillée et globale, la manière d'utilisation du système que nous avons proposé.



Dans l'état actuel de notre projet, nous sommes arrivés à réaliser une application qui répond brillamment aux objectifs fixés aux débuts, mais notre application à sa première version comme toute autre application a besoin des améliorations. Ces améliorations restent comme perspectives de notre travail.

---

---

# **Conclusion générale et perspectives**

---

---



Les innovations technologiques y compris l'ordinateur et l'informatique engendrent une quantité de documents et d'images de plus en plus complexes. Cette grande masse de documents a obligé l'être humain à chercher des moyens pour les exploiter facilement et plus efficacement.

Le travail adressé dans ce mémoire a pour objectif de faciliter l'accès et la compréhension des documents, et plus précisément les pages de journal arabe, en contribuant à l'automatisation de la reconnaissance de la structure logique des documents. L'extraction de cette dernière a pour but de reconnaître la nature des composants du document qui sont organisés hiérarchiquement, ainsi que les relations entre ces composants. Cependant, l'analyse et la reconnaissance des structures des documents est une étape très importante dans toute application de reconnaissance et de compréhension de documents comme par exemple la reconnaissance de formulaires et de courriers, l'indexation, la recherche et la classification automatique de documents. Ces applications ont pour but de réaliser le rêve « un monde sans papier » ; malgré les efforts et les progrès réalisés dans le domaine, on est encore loin de l'atteindre.

Ce mémoire présente un système qui sert à faire passer d'une image brute de page de journal à un ensemble d'informations structurées exploitables représentant l'organisation logique du document. Pour réaliser ce système, nous avons procédé à l'extraction des structures physique et logique des documents. L'extraction de la structure physique permet d'accomplir la séparation entre les articles, les blocs, les colonnes de texte, les lignes, etc. Ainsi, pour ce but nous avons utilisé une méthode mixte ; nous avons effectué une segmentation ascendante pour étiqueter les différentes composantes connexes de la page. Ces dernières sont ensuite regroupées et utilisées pour séparer les éléments textuels et les éléments graphiques. Après, nous avons suivi une segmentation descendante (par projections horizontales et verticales) afin d'extraire les articles, les blocs et les lignes de texte. D'autre part, l'extraction de la structure logique d'une page de journal a pour but de comprendre l'organisation hiérarchique de ses éléments et de créer une interface de navigation à l'intérieur de la page de journal en énumérant tous les éléments (titres, colonnes, figure, auteur, légende, pied de page, etc.). L'extraction de la structure logique est faite en étiquetant les différents éléments physiques extraits. Cet étiquetage repose principalement sur certaines règles de la taille et la position des éléments de la page et aussi sur des connaissances à priori de certaines propriétés des entités logiques (titres, figure, auteur, légende ...etc.).

Plusieurs tests ont été menés pour évaluer les performances du système développé et les résultats obtenus sont encourageants.

### **Perspectives**

Plusieurs perspectives peuvent être envisagées :

- Développer le système proposé pour qu'il soit générique et applicable sur d'autres journaux pas uniquement Echorouk ;
- Etendre le système développé pour qu'il permette l'indexation et la recherche des pages de journal. La recherche peut être par image exemple ou par requête textuelle correspondante à un mot ou à une phrase.
- Réviser l'étape de segmentation notamment la segmentation en mots ;
- Inclure d'autres modules de prétraitement notamment pour permettre l'analyse et la reconnaissance des images issues de la numérisation des pages de journal par un scanner ou un appareil photo;
- Introduire d'autres traitements visant à extraire le texte à partir des figures et des tableaux ;
- Penser à augmenter la précision du système en intégrant un module d'apprentissage automatique ;
- Améliorer l'étape de reconnaissance des titres;
- Améliorer l'étape d'extraction des figures pour qu'elle soit applicable sur toutes les formes des figures dans le journal, pas sur la forme rectangulaire uniquement;
- L'adaptation des étapes de traitement selon les caractéristiques et l'organisation du document.



---

---

# Bibliographie

---

---

## Bibliographie

- [AA03] A. Amano, N. Asada, « Graph grammar based analysis system of complex table form document ». International Conference on Document Analysis and Recognition (ICDAR), pp. 916 – 920, 2003.
- [AF89] J. André, R. Furuta, « Structured documents ». Cambridge University Press, 1989.
- [AT90] H. Asada, S. Tsujimoto, « Understanding Multi-articled Documents ». International Conference on Pattern Recognition (ICPR), pp. 551 - 556, 1990.
- [Azo95] A. S. Azokly, « Une approche uniforme pour la reconnaissance de la structure physique de documents composites fondée sur l'analyse des espaces ». Thèse de doctorat, Institut d'Informatique - Université de Fribourg (Suisse) ,1995.
- [Bel97] A. Belaid, « Conception assistée de modèles de page en vue de leur utilisation en reconnaissance de documents ». Lausanne – Atelier sur les modèles de pages électroniques (Lausanne), 17p, 1997.
- [BF15] H. Boufersaoui, I. Frihi, « Extraction de la structure logique des documents ». Mémoire de Master, Option : Ingénierie de médias, Université 08 Mai 1945 – Guelma, Juin 2015.
- [BFJ90] H.S. Baird, S. Fortune, S.E. Jones, «Image Segmentation by Shape-Directed Covers ». International Conference on Pattern Recognition, pp. 820 - 825, 1990.
- [BIZ97] R. Brugger, R. Ingold, A. Zramdini, « Modeling documents for structure recognition using generalized n-grams ». International Conference on Document Analysis and Recognition, pp. 56-60, 1997.
- [BM07] F. Bouchara, E. Murisasco, «Segmentation d'image : Application aux documents anciens ». Mémoire de master de recherche, Laboratoire de sciences de l'information et des systèmes, 2007.
- [CCD02] S.P. Chowdhuri, B. Chanda, A.K. Das, « A Complete System for Document Image Segmentation ». National Workshop on Computer Vision, Graphics and Image Processing (WVGIP 2002), pp. 9–16, Madurai - India, February 2002.
- [CGM99] K.V. Chandrinos, B. Gatos, S.L. Mantzaris, « Integrated Algorithms for Newspaper Page Segmentation and Article Tracking ». Proceedings of the 5th International Conference on Document Analysis and Recognition (ICDAR), pp. 559-562, Bangalore - India, 1999.
- [CLM02] L. Cinque, S. Levialdi, A. Malizia, « DAN: an automatic segmentation and classification engine for paper documents ». Fifth IAPR International Workshop



- on Document Analysis Systems (DAS 2002), LNCS 2423, pp. 491-502, Princeton, New Jersey – USA, August 2002.
- [CT97] R.C. Chen, L.Y. Tseng, « The recognition of form documents based on three types of line segments ». International Conference on Document Analysis and Recognition (ICDAR), pp. 71 – 75, 1997.
- [Dou10] J.M. Doudoux, « Développons en Java ». 2010.
- [Dum05] B. Dumas, « Structexed : un outil pour la reconstruction des structures logiques ». Travail de Master en Informatique, Université de Fribourg, 2005.
- [Duo05] J. Duong, « Étude des documents imprimés: Approche statistique et contribution méthodologique ». Thèse de doctorat, Université Claude Bernard, 2005.
- [Dup94] G. Dupoirier, « Technologie de la GED: l'édition électronique ». Hermès, 1994.
- [FN00] J. Facon, L. Neves, «Methodology of automatic extraction of table-form cells». Symposium on Computer Graphics and Image Processing, pp. 15 – 21, 2000.
- [GLNS90] V.Govindaraju, S.W.Lam, D.Niyogi, D.B.Sher, « Newspaper Image Understanding ». In S. Ramani, R. Chandrasekar editors, Knowledge Based Computer Systems, Narosa Publishing House, New Delhi India, pp. 375-384, 1990.
- [Had06] K. Hadjar, «Une étude de l'évolutivité des modèles pour la reconnaissance de documents arabes dans un contexte interactif ». Thèse de doctorat, Université Fribourg (Suisse), 2006.
- [HHI01] K. Hadjar, O. Hitz , R. Ingold, « Newspaper Page Decomposition using a Split and Merge Approach ». 6<sup>th</sup> International Conference on Document Analysis and Recognition (ICDAR), pp. 1186-1189, Seattle - USA, 2001.
- [HHP95] J. Ha, R. Haralick, I. Phillips, «Recursive x-y cut using bounding boxes of connected components ». International Conference on Document Analysis and Recognition (ICDAR), pp. 952 - 955, 1995.
- [HI03] K. Hadjar, R. Ingold, « Arabic Newspaper Page Segmentation ». 7<sup>th</sup> International Conference on Document Analysis and Recognition, pp. 895-899, Edinburgh - Scotland, August 2003.
- [HMSS01] E. Hadano, K. Marukawa, H. Shinjo, Y. Shima, «A recursive analysis for form cell recognition». International Conference on Document Analysis and Recognition (ICDAR), pp. 694 – 698, 2001.
- [Ing91] R. Ingold, « A Top-Down Document Analysis Method for Logical Structure Recognition ». 1<sup>st</sup> International Conference on Document Analysis and Recognition, pp. 41 - 49, Saint-Malo, France, 1991.

Recognition, pp. 472-475, Montreal, Canada, 1995.

- [Ogo93] L.O’Gorman, « The document spectrum for page layout analysis ». IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pp. 1162 – 1173, 1993.
- [Ots79] N. Otsu, « A Threshold Selection Method from Gray-Level Histograms ». IEEE transactions on Systems, Man, and Cybernetics, vol. 9, No.1, pp. 62-66, 1979.
- [PHT12] T. Palfray, D. Hébert, P. Tranouez, « Segmentation logique d’images de journaux anciens ». Laboratoire LITIS, UFR de sciences Avenue de l’université 76800 Saint Etienne du Rouvray, 2012.
- [PZ91] T. Pavlidis, J. Zhou, «Page Segmentation by White Streams ». International Conference on Document Analysis and Recognition (ICDAR), pp. 945 - 953, 1991.
- [Rob01] L. Robadey, «2 (CREM): Une méthode de reconnaissance structurelle de documents complexes basée sur des patterns bidimensionnels ». Thèse de doctorat, Université de Fribourg – Suisse, 2001.
- [SP01] S. Souafi-Bensafi, M. Parizeau, « Logical Labeling using Bayesian Networks ». 6<sup>th</sup> International Conference on Document Analysis and Recognition, pp. 832-836, Seattle - USA, September 2000.
- [Sun06] H.M. Sun, «Enhanced Constrained Run-Length Algorithm for Complex Layout Document Processing». International Journal of Applied Science and Engineering, vol. 4, No. 3, pp. 297 - 309, 2006.
- [SW89] S. N. Srihari, D. Wang, « Classification of newspaper image blocks using texture analysis». Computer Vision Graphics and Image Processing, vol. 47, No. 3, pp. 327–352, 1989.
- [Thi07] Nguyen Thioanh, « Binarisation d’images de documents graphiques». Maître de conférences à Université de Nancy 2 Chercheur à l’équipe QGAR, INRIA Lorraine Nancy, France, 2007.
- [TNN82] J. Toyoda, Y. Noguchi, Y. Nishimura, « Study of extracting Japanese newspaper article ». 6<sup>th</sup> International Conference on Pattern Recognition, pp. 113 – 115, Munich Germany, 1982.
- [WCW82] K.Y. Wong, R.G. Casey, F.M. Wahl, « Document analysis system ». IBM Journal of Research and Development, vol. 26, No. 6, pp. 647 – 656, 1982.
- [XGDO99] L. Xingyuan, W. Gao, D. Doermann, W.G. Oh, « A robust method for unknown forms analysis ». 5<sup>th</sup> International Conference on Document Analysis and Recognition ICDAR, pp. 531 – 534, Bangalore – India, 1999.



- [Jea09] Benjamin Jean, «Option libre du bon usage des licences libres». CC By-SA 3.0, 2009.
- [Jou07] N. Journet, « Analyse d'images de documents anciens: une approche texture ». Thèse de doctorat, Université de La Rochelle – France, 2007.
- [KK10] D. Ketata, M. Khemakhem, « Un survol sur l'analyse et la reconnaissance de documents: imprimé, ancien et manuscrit ». Colloque International Francophone sur l'Ecrit et le Document (CIFED), p. 12 pages, 2010.
- [KNS93] M. Krishnamoorthy, G. Nagy, S. Seth, « Syntactic segmentation and labeling of digitized pages from technical journals ». IEEE transactions on Pattern Analysis and machine Intelligence archive, vol. 15, No. 7, pp. 737-747, 1993.
- [LCC03] K.H. Lee, Y.C. Choy, S.B. Cho, « Logical structure analysis and generation for structured documents: A syntactic approach ». IEEE transaction on Knowledge and a Data Engineering, vol. 15, No. 5, pp. 1277-1294, 2003.
- [LET03] F. Lebourgeois, H. Emptoz, E. Trinh, « Wp4.3-4 Numérisation, Traitement et Interprétation des Image du Documents Anciens ». Projet Debora Telematics Application Programme n 5608-2003.
- [LLYH01] F. Liu, Y. Luo, M. Yoshikawa, D. Hu, « A New Component based Algorithm for Newspaper Layout Analysis». Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR), pp. 1176-1179, Seattle, USA, 2001.
- [Mar09] I. Martinat, « Conception d'un langage de structures tabulaires et du système de reconnaissance associé. Application aux tableaux dans les documents d'archive ». Thèse de doctorat, INSA de Rennes – France, 2009.
- [Mon11] F. Montreuil, « Extraction de structures de documents par champs aléatoires conditionnels : application aux traitements des courriers manuscrits ». Thèse de doctorat, Université de Rouen, 2011.
- [MY01] P. E. Mitchell, H. Yan, « Newspaper Document Analysis featuring Connected Line Segmentation». 6<sup>th</sup> International Conference on Document Analysis and Recognition, pp. 1181-1185, Seattle, USA, September 2001.
- [Nag00] G. Nagy, « Twenty years of document image analysis in PAMI ». IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, No.1, pp. 38-62, 2000.
- [NS84] G. Nagy, S. Seth, « Hierarchical representation of optically scanned documents ». 17<sup>th</sup> Conference on Pattern Recognition, pp. 347–349, 1984.
- [NS95] D. Niyogi, S. Srihari, « Knowledge-Based Derivation of Document Logical Structure ». 3<sup>rd</sup> International Conference on Document Analysis and

[ZLJO00] H. Zhao, B. Liu, Z. Jian, T. Ostgathe, « Global-local-global method for logical structure extraction of form document image ». Journal of Electronic Imaging, pp. 296 – 304, 2000.