

N°d'ordre :

N°de série :

République Algérienne Démocratique et Populaire



Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique



UNIVERSITÉ ECHAHID HAMMA LAKHDAR - D'EL OUED

FACULTÉ DES SCIENCES EXACTES

DÉPARTEMENT D'INFORMATIQUE

Mémoire de fin d'étude

MASTER ACADÉMIQUE

Domaine: Mathématique et Informatique

Filière: Informatique

Spécialité : **Systèmes Distribués et Intelligence Artificielle**

Thème

**Développement d'un système
de comparaison de listes des notes**

Présenté par:

**BERRIM ABDELLAH
KOULL MOHAMMED FOUAD**

Soutenu devant le jury composé de

Mr YAGOUB MOHAMMED AMINE

MAA

Université Eloued

Président

Mr GHERBI KDOUR

MAA

Université Eloued

Examinateur

Mr ZAIZ FOUZI

MCD

Université Eloued

Rapporteur

Année universitaire 2019/2020

Dédicaces

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chères sœurs, pour leurs encouragements permanents, et leur soutien moral,

A mes chers frères, pour leur appui et leur encouragement,

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible,

Merci d'être toujours là pour moi.

BERRIM Abdellah

Dédicaces

*Au nom d'Allah le plus grand merci lui revient de me avoir guider vers le droit chemin.
Je dédie ce modeste travail à:*

À mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance.

*À ma petite famille , ma chère femme qui m'a soutenu, mes petit anges MARIA ,
SERINE et MERIEM*

*À mes frères et mes sœurs, chacun à son nom ;
À toute la famille ;
À tous mes amis ;
À tous mes chers enseignants qui m'ont enseigné*

KOULL MOHAMMED FOUAD

Remerciements

*A Allah,
le tout puissant, nous rendons grâce pour nous avoir donné santé, patience, volonté et surtout raison.*

*On dit souvent que le trajet est aussi important que la destination
En premier lieu, nous aimerions remercier notre encadreur Monsieur ZAIZ FOUAZI qui nous a aidé et qui nous a donné des conseils durant ce travail.*

Nous remercions tous les enseignants qui nous ont inculqué beaucoup de connaissances.

nous remercions chaleureusement les membres du jury qui nous ont fait l'honneur de participer à notre jury et accepter de juger ce modeste travail.

En fin, nous voulons encore remercier tous ceux qui nous ont consolidé et nous ont encouragé entres autres les parents et les amis

Résumé

La reconnaissance et l'analyse de tableaux sur des images de documents imprimés est un domaine de recherche populaire de la reconnaissance de formes et du traitement d'images. Les méthodes de reconnaissance de table existantes nécessitent généralement un degré élevé de régularité, et la robustesse doit encore être améliorée de manière significative. Dans notre travail, on présente un système de comparaison des listes des notes basée sur une méthode de reconnaissance des chiffres imprimé à partir d'un texte imprimé. L'étude est reposée sur l'analyse optique du texte, un pré-traitement de l'image permet de réduire le bruit et de produire l'image binaire. La segmentation est fait par la projection horizontale isole les lignes, la projection verticale isole les mots, fragmente le mot en chiffres et détecte la position du chiffres.

Le processus de classification est effectué par des bibliothèques spéciales, dont openCV et Tesseract, et un ensemble d'algorithmes pour le filtrage et la correction d'image ainsi que la correction de son orientation.

Les résultats montrent que le système est assez performant, mais qu'il n'est pas impeccable, découvrent encore une fois la limite de la reconnaissance des chiffres et lettres imprimés qu'il faut donc compléter par une reconnaissance plus contextuelle.

Mots Clés: Reconnaissance de formes, traitement de images, réseaux de neurones convolutifs, apprentissage profonde, filtrage d'image, détection des contours, reconnaissance optique de caractères (O.C.R)

Abstract

Recognition and analysis of Tables on images of printed documents is a popular area of research in pattern recognition and image processing, Existing table recognition methods generally require a high degree of regularity, and the robustness still need to be improved significantly

Recognition of printed numbers is an open problem. This is due to difficulties such as the variability of form, style, and gradient of the writing. The writing of the printed figures is obviously cursive, difficult to process, and has great variability.

In our work, we present a system of comparison of lists of notes based on a method of recognizing figures printed from printed text. The study is based on the optical analysis of the text, a preprocessing of the image makes it possible to reduce the noise and to produce the binary image. The segmentation is done by the horizontal projection isolates the lines, the vertical projection isolates the words, fragments the word into figures and detects the position of the figures.

The classification process is carried out by special libraries, including openCV and Tesseract, and a set of algorithms for image filtering, image correction, as well as its orientation correction. The results show that the system is quite efficient, but that it is not flawless, discover once again the limit of the recognition of printed figures which must therefore be supplemented by a more contextual recognition.

Keywords: Pattern recognition, image processing, image edge detection, image filtering, convolutional neural networks, deep learning, Optical Character Recognition (OCR).

الملخص

يعد التعرف على الجداول الموجودة على صور المستندات المطبوعة وتحليلها مجالاً شائعاً للبحث في التعرف على الأنماط ومعالجة الصور. تتطلب طرق التعرف على الجدول الحالية عموماً درجة عالية من الانتظام، ولا يزال الأداء بحاجة إلى تحسين كبير. في عملنا هذا، نقدم نظاماً لمقارنة قوائم نقاط امتحانات الطلبة يعتمد على طريقة التعرف على الأشكال انطلاقاً من نص مطبوع. تعتمد الدراسة على تحليل النص البصري، والمعالجة المسبقة للصورة للتقليل من الضوضاء وذلك للحصول على صورة ثنائية. تليه عملية التقسيم عن طريق الإسقاط الأفقي لعزل الخطوط، الإسقاط العمودي لعزل الكلمات، وتجزئة الكلمة إلى أشكال والكشف عن مواضع هذه الأشكال. تتم عملية التصنيف بواسطة مكتبات خاصة و مجموعة من الخوارزميات الخاصة بفلتر الصورة و تصحيح اتجاهها و قد أظهرت النتائج أن النظام فعال للغاية، ولكنه ليس بلا عيوب، لتبين مرة أخرى محدودية التعرف على الأشكال المطبوعة التي يجب استكمالها من خلال التعرف على المعاني. الكلمات الرئيسية: التعرف على الأنماط، معالجة الصور، كشف حافة الصورة، تصفية الصور، ملامح الصورة، الشبكات العصبية التآلفية، التعلم العميق، التعرف البصري على الأحرف

Table des matières

Dédicaces	i
Dédicaces	i
Remerciements	ii
Résumé	iii
Liste des figures	xi
Liste des tableaux	xiii
Liste des Equations	xiv
Introduction	1
1 État de l'art	4
Introduction	4
1.1 Traitement des Images:	5
1.1.1 Introduction:	5
1.1.2 Traitement d'image et la vision artificielle:	5
1.1.3 Représentation d'image Numérique:	5
1.1.3.1 Définition:	5
1.1.3.2 Acquisition:	5
1.1.3.3 Les Différents transformations possibles sur l'image Numérique:	5
1.1.3.4 Composition et Caractéristiques d'une image numérique:	6
1.1.3.4.1 Le Pixel:	6
1.1.3.4.2 La Définition:	6
1.1.3.4.3 La résolution :	6
1.1.3.4.4 Codage des Couleurs (ou profondeur des couleurs):	6

1.1.3.4.5	Bruit:	7
1.1.3.4.6	Luminance:	7
1.1.3.4.7	Contraste:	7
1.1.3.4.8	La région:	7
1.1.3.4.9	Contours:	7
1.1.4	Représentation des images:	8
1.1.4.1	Numérisation des images:	8
1.1.4.2	L'échantillonnage:	8
1.1.4.3	La quantification:	8
1.1.5	Outils fondamentaux:	9
1.1.5.1	L'histogramme:	9
1.1.5.2	La convolution:	10
1.1.5.3	La transformée de Fourier:	10
1.1.6	Filtrage des Images:	11
1.1.6.1	Les types de filtres:	11
1.1.6.1.1	Les filtres passe-bas:	11
1.1.6.1.2	Les filtres passe-haut:	11
1.1.6.1.3	Filtrage dans le domaine spatial:	11
1.1.7	détection de contours:	11
1.1.7.1	Approche gradient:	12
1.1.7.2	La dérivée seconde	13
1.2	Intelligence Artificielle:	13
1.2.1	Définition de l'intelligence Artificielle:	13
1.2.2	Les Domaines d'application de l'intelligence artificielle:	14
1.2.3	L'Apprentissage Automatique:	14
1.2.3.1	Types d'apprentissage:	14
1.2.3.2	Apprentissage non supervisé:	14
1.2.3.3	Apprentissage supervisé:	15
1.2.3.3.1	Les algorithmes d'apprentissage supervisé	15
1.2.4	L'apprentissage en profondeur (deep Learning):	15
1.2.4.1	Les différentes Architectures du Deep Learning:	16
1.2.4.1.1	Les réseaux de neurones convolutifs	16
1.2.4.2	Les Éléments de base du réseau de neurones convolutifs:	16

1.2.4.2.1	Réseaux de neurones de convolution pour la détection d'objets:	16
1.2.4.2.2	Réseau de neurones récurrents:	19
1.2.4.2.3	Modèle génératif:	19
1.3	Reconnaissance de Caractères (O.C.R):	19
1.3.1	Les phases d'un système OCR:	19
1.3.2	Les différents systèmes de reconnaissance de caractères:	20
1.3.3	la reconnaissance de caractères imprimés:	20
1.3.3.1	Caractéristiques des caractères imprimés:	21
1.3.4	la reconnaissance de caractères manuscrit	21
1.3.4.1	Type de reconnaissance de l'écriture	22
1.3.4.1.1	La reconnaissance en-ligne (on-line):	22
1.3.4.1.2	Reconnaissance hors ligne (off-line):	23
1.3.4.1.3	processus de la reconnaissance OCR:	23
1.3.4.2	les Approches de reconnaissance:	24
Conclusion	25
2	Détection et extraction des tables	26
Introduction	26
2.1	Problématique d'extraction des tableaux:	26
2.1.1	Détection de table à partir de documents image:	26
2.2	Travaux connexes sur l'extraction de tables	27
2.2.1	L'apprentissage profond en action:	27
2.2.1.1	TableNet:	27
2.2.1.1.1	DataSet:	28
2.2.1.1.2	Architecture de TableNet	28
2.2.1.1.3	Les sorties	29
2.2.1.2	DeepDeSRT:	29
2.2.1.2.1	DataSet:	29
2.2.1.2.2	Architecture de DeepDeSRT:	30
2.2.1.3	Graph des réseaux de neurones:	31
2.2.1.3.1	DataSet:	31
2.2.1.3.2	Architecture:	31
2.2.1.4	CGAN et algorithmes génétiques:	32

2.2.1.4.1	DataSet:	33
2.2.1.4.2	Architecture:	33
2.2.2	Détection de table avec OpenCV:	34
2.2.3	les outils Open Source:	37
2.2.3.1	Tabula:	37
2.2.3.2	Camelot:	38
2.2.3.3	PyPDF2:	39
2.2.4	bref résumé des méthodes de reconnaissance des tableaux:	39
	Conclusion	40
3	Conception	41
	Introduction	41
3.1	Architecture globale de système:	42
3.2	Explication d'architecture:	42
3.2.1	Pré-traitement d'image:	42
3.2.1.1	Correction automatique de l'inclinaison à l'aide de détecteurs de coin et d'homographie:	44
3.2.2	Détection de table:	45
3.2.3	Allocation approfondie des cellules:	45
3.2.4	Extraction des valeurs de cellules:	45
3.2.4.1	Méthode d'extraction des valeurs proposée:	46
3.2.5	comparaison les DataFrames:	47
	Conclusion	48
4	Implémentation et résultats	49
	Introduction	49
4.1	Le choix de langage de programmation:	49
4.1.1	Python	49
4.1.2	L'environnement de développement :	50
4.1.3	L'environnement Anaconda:	50
4.2	Bibliothèques utilisées:	50
4.2.1	OpenCV (Open Source Computer Vision Library):	50
4.2.2	Pandas:	50
4.2.3	QT:	51

4.2.3.1	QT designer	51
4.2.4	Tesseract-Ocr:	51
4.2.5	Présentation de l'interface d'application:	52
	Conclusion	58
	Conclusion et perspectives	59
	Bibliographie	60

Liste des figures

1	État de l'art	4
1.1	Acquisition des Images Numériques	5
1.2	Le Bruit	7
1.3	La Région	7
1.4	Échantillonnage d'une image	8
1.5	Quantification d'une image	9
1.6	La Convolution	10
1.7	modèles de contours	12
1.8	La dérivation seconde: Laplacien	13
1.9	Apprentissage non supervisé	14
1.10	Apprentissage supervisé	15
1.11	Architecture de CNN	16
1.12	Architecture de R-CNN	17
1.13	Architecture de Fast R-CNN	17
1.14	Faster R-CNN	18
1.15	You Only Look Once(YOLO)	18
1.16	le processus général de reconnaissance de l'écriture	20
1.17	Panorama des types d'écriture	20
1.18	les différentes étapes de passage de la forme papier vers des formes élec- troniques codées	21
1.19	Systèmes de reconnaissance de l'écriture	23
2	Détection et extraction des tables	26
2.1	Architecture de TableNet	28
2.2	Exemple d'image de document et ses masques de sortie générés après le traitement à partir de TableNet	29
2.3	Détection Table par DeepDeSRT (ICDAR 2013)	30
2.4	reconnaissance Table par DeepDeSRT (ICDAR 2013)	30
2.5	Images avec différentes catégories de difficulté. Les images de catégorie 1 sont des images simples sans fusion et avec des lignes dominantes. Caté- gorie 2 ajoute différents types de frontières, y compris l'absence occasion- nelle de lignes directrices. La catégorie 3 est la plus difficile qui introduit la cellule et la colonne fusion. Modèles de catégorie 4 images capturées par caméra par transformation de perspective linéaire	31
2.6	Sorties générées par Graph Neural Networks	32
2.7	Schéma général de l'approche CGAN	33
2.8	Exemple d'estimation de la structure du table	34

2.9	charge image pour traitement OpenCv	35
2.10	Inverse Image Thresholding	35
2.11	la dilatation de l'image	35
2.12	le Méthode de FinContours	36
2.13	Extraction de lignes de tableau: (a) L'image d'origine, (b) l'image binaire, (c) le l'extraction de lignes de table verticales, (d) l'extraction de lignes de table horizontales, (e)le table combinée.	36
2.14	les Méthodes boundingRect et rectangle	36
2.15	La méthode namedWindow OpenCV	37
2.16	Contenu et Contours Image avec OpenCV	37
2.17	Interface Tabula Web Server	38
2.18	Inteface Excalibur de Camelot	38
3	Conception	41
3.1	Architecture du système pour reconnaissance et extraction de table	42
3.2	processus de pré traitement	43
3.3	Formule Homography	44
3.4	Schéma de la fonction de classification des segments des caractères	46
4	Implémentation et résultats	49
4.1	Interface Qt Designer	51
4.2	Interface principale de notre application	52
4.3	image inclinée	54
4.4	image originale vs image table après détection de contours	54
4.5	image table après la correction	55
4.6	combinaison des lignes vertical et horizontal	55
4.7	détection approfondie des cellules	56
4.8	Extraction des cellules au fichier CSV	57
4.9	Résultat de comparaison de deux liste des notes (table sous forme image)	57

Liste des tableaux

1	État de l'art	4
1.1	comparaison entre RCNN,fast RCNN,Faster RCNN	18
1.2	quelques types de standardisation des Fontes	22
2	Détection et extraction des tables	26
2.1	Résultat de destruction des tableau	29
2.2	comparaison Resultat extraction des Tables(04 Catégories)	32
2.3	résumé les méthodes de détection et d'extraction de table	39

Liste des equations

1.3	Calcule de la Convolution	10
1.4	Transformée de Fourier discrète bidimensionnelle Direct	10
1.5	Transformée de Fourier discrète bidimensionnelle Inverse	11
1.6	Calcule le Gradient (G)	12
1.7	calcule de module (m)	13
1.8	calcule de module θ	13
1.9	La dérivée seconde	13

Introduction

La reconnaissance de chiffres imprimé a pour but de transformer ce dernier en une représentation intelligible par une machine, et simplement reproductible par un traitement de texte, ceci n'est pas toujours facile parce que les chiffres peuvent avoir nombreuses représentations, différents styles et différentes polices (Gras, Italiques, . . . etc.). La reconnaissance de chiffres imprimé possède plusieurs applications, notamment en tri automatique du courrier et bureautique, pour l'enregistrement et la vérification des chèques bancaires, et le traitement automatique de dossiers administratifs. Le développement d'un système de comparaison de listes des notes utilise la reconnaissance de chiffres imprimé, l'automatisation de son traitement devient un domaine de recherche important. Une liste de note contient plusieurs types d'informations qui sont souvent tableau, caractères imprimés et chiffres imprimé. Des nombreuses solutions ont été présentées pour résoudre ce problème en créant des systèmes qui utilisent des techniques de traitement d'images et de reconnaissance des caractères OCR. Les recherches réalisées dans le domaine de la reconnaissance des caractères imprimés et la détection des tableaux à partir d'un document (pdf ou image) ont présentées des améliorations de performances en utilisant des méthodes de recherche globales. Dans ce travail, on s'intéresse à développer un système de comparaison de listes des notes en se basant sur le numéro d'inscription de l'étudiant et sa note dans de la liste. Nous avons proposé de représenter l'image scanner de la liste note venant de l'administration après un processus de détection et d'extraction des informations de l'image tableau sous un fichier Excel CSV et ensuite le comparer avec la liste (fichier Excel) qui est donnée par l'enseignant. Notre mémoire est organisé comme suit :

Dans le premier chapitre, nous présenterons quelques notions de bases sur la représentation des images numériques, l'intelligence artificielle et la reconnaissance des caractères. Dans le deuxième chapitre, nous allons expliquer et analyser quelques travaux antérieurs relative à la détection et l'extraction des tableaux à partir d'une image d'un document.

Dans le troisième chapitre, nous présenterons notre système, où nous voyons une description détaillée du système et de la méthode de comparaison proposée. Dans le quatrième chapitre, nous présenterons l'application et les résultats de de comparaison des listes basée sur la reconnaissance des chiffres et des lettres imprimés. Nous terminons le travail par une conclusion sur les résultats obtenus par la méthode utilisée, et des perspectives de ce travail.

Problématique

Les systèmes reconnaissance des formes ne sont pas assurés 100 % du fait de la très grande changement des styles d'écriture.

La complication de la reconnaissance de caractères imprimé est due principalement aux variations des styles d'écriture. Dans cette thèse, nous avons traité une problématique majeure rencontrée en reconnaissance de caractères imprimé qui est la proposition d'un système comparaison des listes des notes basée sur la reconnaissance de caractères imprimés qui s'adapter aux changements des styles d'écriture pour en augmenter les performances.

Des solutions nombreuses ont été présentées pour résoudre ce problème en créant des systèmes qui utilisent des techniques à base traitement du images et reconnaissance des caractères OCR .

Les recherches guidées dans le domaine de la reconnaissance des caractères imprimés et la detection des tableau a partir d'un document (pdf ou image)ont présenté qu'une amélioration élevé de performances peut être exécutée en utilisant des méthodes de recherche globales. Les travaux de cette thèse sont créés sur deux aspects unifiés qui sont l'intelligence en essaim et traitement de images. La démarche suivie dans cette thèse se dispose dans le cadre d'étude des systèmes émergents afin de les adapter au problème de la detection et extraction des tableaux.

Nous nous intéressons à la modélisation d'un système de détection et extraction des tableaux basée sur le traitement du images et OCR.

Contributions

Dans notre travail, on consistant à développer un système de comparaison de listes des notes appliqué sur le numéro de l'inscription de l'étudiant et sa note dans de la liste. Nous avons proposé de représenter l'image scanner de la liste note qui est à l'administration applique les processus de détection et l'extraction les informations de l'image tableau pour obtenu un fichier Excel CSV et après de comparer par la liste (fichier Excel) qui est transmettre par l'enseignant.

Organisation du travail

Notre mémoire se sectionne comme suit : Dans le premier chapitre, nous présenterons rapidement les notions de bases de la représentation des images numériques, l'intelligence artificielle et la reconnaissance des caractères

Dans le deuxième chapitre, nous intéresserons à expliquer et analyser les travaux antérieurs relative a la détection et l'extraction des image tableau,

Dans le troisième chapitre, nous présenterons la conception de notre système.

Dans le quatrième chapitre, nous présenterons l'implémentation et résultats de notre travail sur la système de comparaison de liste basé sur la reconnaissance des chiffres et des lettres imprimés.

Nous terminons le travail par une conclusion sur les résultats obtenus par la méthode utilisée, et des perspectives de ce travail.

État de l'art

Introduction

Le traitement des images concerne les méthodes et les techniques appliquées à l'image pour en extraire les différentes informations contenues dans l'image et nécessaires à une application ou à une autre. on peut classer les méthodes et algorithmes existantes en deux grandes catégories :

- traitements de bas-niveaux qui nécessitent très peu d'informations sur le contenu des images (le filtrage, l'amélioration des images, . . . etc.)
- Traitement de haut-niveaux, qui fonctionnent après ceux de bas-niveaux, et qui peuvent nécessiter des informations sur le contenu des images (la reconnaissance des formes, la reconnaissance des caractères (OCR), reconstruction 3D et autres).

Depuis des décennies, la reconnaissance optique de caractères (OCR) est largement mise en œuvre dans diverses applications en convertissant le texte imprimé en texte modifiable, comme la littérature d'archives, la bureautique et la reconnaissance des plaques d'immatriculation . Cette technologie de pointe intègre le traitement d'image numérique, la vision par ordinateur et d'autres disciplines. Le développement rapide de l'OCR a favorisé la transformation de nombreuses industries, car il peut considérablement économiser les heures de travail, ainsi que les coûts de main-d'œuvre. La reconnaissance optique de caractères désigne les procédés informatiques pour la traduction d'images de textes (imprimés, articles de journaux, livres, ...) en fichiers de textes. pour reconnaître et extraire des informations d'une image de table. Il comprend de nombreux aspects, y compris le traitement d'image, la reconnaissance des des caractères

Nous présentons dans ce chapitre, un aperçu de l'état l'art des techniques dans les principaux concepts de traitement d'images et la vision artificielle, et la Reconnaissance des caractères

1.1 Traitement des Images:

1.1.1 Introduction:

Le traitement et l'analyse d'images numérique désigne une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Dans le domaine de la vision artificielle, nous distinguons le traitement d'images juste après les étapes d'acquisition et de numérisation dans le but d'assurer les transformations d'images ainsi que la phase de calcul qui permet la réalisation d'une interprétation des images traitées.[14].

1.1.2 Traitement d'image et la vision artificielle:

le traitement d'image et la vision artificielle tirent parti des connaissances et des techniques d'intelligence artificielle pour gérer l'adaptation à un environnement changeant, l'information incertaine, les systèmes hétérogènes de connaissances et les différents niveaux de prise de décision.

1.1.3 Représentation d'image Numérique:

1.1.3.1 Définition:

Les images numériques sont composées d'un ensemble de pixels (picture éléments), juxtaposés en lignes et en colonnes. Le pixel, est le plus petit élément que l'on peut trouver dans une image. Chaque pixel possède des caractéristiques propres, couleurs, luminosité, brillance, qui permettent de les différencier et de constitué les images

1.1.3.2 Acquisition:

Le processus général pour l'acquisition d'une image numérique est illustré sur la figure suivante :

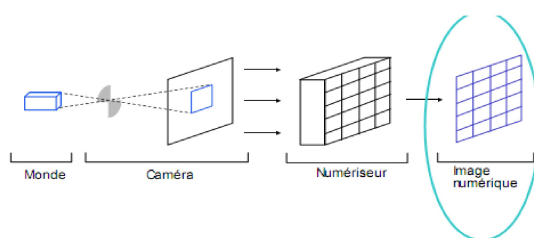


Figure 1.1: Acquisition des Images Numériques

1.1.3.3 Les Différents transformations possibles sur l'image Numérique:

On peut distinguer quatre (04) types ou modes de transformations, qu'on peut effectuer à partir ou vers une image numérique.

- **Le Traitement des Images:** Image \rightarrow Image «image processing»
- **L'analyse des Images:** Image \rightarrow mesures «image analysis» analyse sémantique.

- **Image description de haut niveau:** Images \rightarrow Classes d'objet «image understanding or recognition».
- **Reconstruction d'Images:** Opérateurs ensemble d'informations \rightarrow image.

1.1.3.4 Composition et Caractéristiques d'une image numérique:

L'image est un ensemble déstructuré d'informations caractérisé par les paramètres suivants:

1.1.3.4.1 Le Pixel: Une image numérique est composée d'un ensemble de points appelés pixels (abréviation de picture Élément) pour établir une image. Le pixel indique ainsi le plus petit élément particulier d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image.

1.1.3.4.2 La Définition: On nomme définition le nombre de points (pixels) créant une image: c'est le nombre de colonnes de l'image que multiplie son nombre de lignes.

Les Principaux formats d'affichage: Pour afficher ces images, des formats d'affichages standards ont été déterminés en fonction de l'évolution des capacités matérielles des cartes graphiques et des écrans. ci après les principaux formats d'affichage d'une image:

- **CGA** (320 x 200) 4 couleurs.
- **VGA** (640 x 480) 16 couleurs.
- **SVGA** (800x600) 256 couleurs.
- **XGA** (1024 x 768) 256 couleurs.
- **SXGA** (1280 x 1024) en 16 millions de couleurs.

Formule de Calcul: Calcul du nombre total des pixels dans une image: Nombre total des pixels = colonnes x lignes.

1.1.3.4.3 La résolution : C'est le nombre de points contenu dans une longueur donnée (en pouce). Elle est formulée en points par pouce (**PPP**, en anglais: **DPI pour (Dots Per Inch)**). Un pouce mesure 2.54 cm, c'est une unité de mesure anglaise. La résolution permet ainsi d'établir la relation entre la définition en pixels d'une image et la dimension réelle de sa représentation sur un support physique (affichage écran, impression papier...).

1.1.3.4.4 Codage des Couleurs (ou profondeur des couleurs): En plus de sa définition, une image numérique utilise plus ou moins de mémoire selon le codage des informations de couleur qu'elle possède. C'est ce que l'on appelle le codage de couleurs ou profondeur des couleurs, exprimé en bit par pixel (bpp): 1, 4, 8, 16 bits...

En connaissant le nombre de pixels d'une image et la mémoire nécessaire à l'affichage d'un pixel, il est possible de définir nettement le poids que va utiliser le fichier image sur le disque dur (ou l'espace mémoire nécessaire en RAM pour réaliser un calcul sur cette image).

Une image crée comme une matrice où chaque case contient des nombres associés à une couleur. Généralement on distingue 3 grands types de couleurs pour une image numérique : (Le noir et blanc, Les niveaux de gris et La couleur).

1.1.3.4.5 Bruit: Le bruit d'image est la existence d'informations parasites dans cette dernière, il résulte de la lumière des dispositifs optiques et électroniques du capteur.

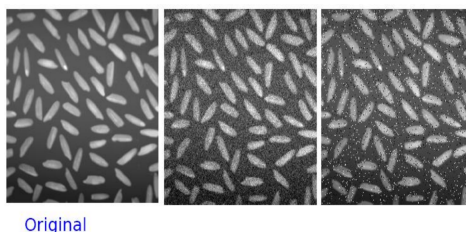


Figure 1.2: Le Bruit

1.1.3.4.6 Luminance: C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le division de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur éloigné, le mot luminance est remplacé au mot brillance, qui correspond à le part d'un objet.

1.1.3.4.7 Contraste: C'est l'opposition marquée entre deux régions d'une image, plus exactement entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si $L1$ et $L2$ sont les degrés de luminosité respectivement de deux zones voisines $A1$ et $A2$ d'une image, le contraste C est défini par le rapport.[11]

$$C = \frac{L1 - L2}{(L1 + L2)} \quad (1.1)$$

1.1.3.4.8 La région: La notion de région dans le traitement d'images, comme évoquée ci-dessus, est de regrouper des zones possédant les mêmes caractéristiques . C'est-a-dire que si plusieurs pixels adjacents s'avèrent être de couleur identique alors la zone qu'ils forment est une région. Ci-après, une illustration montrant deux régions de pixels différentes :

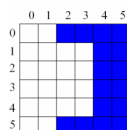


Figure 1.3: La Région

1.1.3.4.9 Contours: Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative.

En traitement d'image et en vision par ordinateur, on appelle détection de contours les procédés permettant de repérer les points d'une image matricielle qui correspondent à un changement brutal de l'intensité lumineuse.

1.1.4 Représentation des images:

1.1.4.1 Numérisation des images:

pour obtenir un image, on utilise un intermédiaire d'un capture et un numériseur, ces pour transformer un signal optique en un signale numérique.

Numériser une image c'est lui donner une représentation électronique à partir de l'objet réel qui lui sert de support (papier, film, diapo, négatif, mais aussi objet 3D). on peut obtenir le signale optique directement par l'intermédiaire d'un système d'agrandissement optique (Exemple: lentille simple, microscope) le signale électronique est repris par un convertisseur analogique ou digitale qui transforme les données continues en données numérique codés sur 1,8 16 ou 24 bits, le codage utilisé définie d'images. deux opérations permet de numériser une image:

- l'échantillonnage.
- la quantification.

1.1.4.2 L'échantillonnage:

L'échantillonnage est défini comme le passage d'une image continue à une image discrète, et la reconstruction spatiale comme celui de d'une image discrète à une image continue.[15]

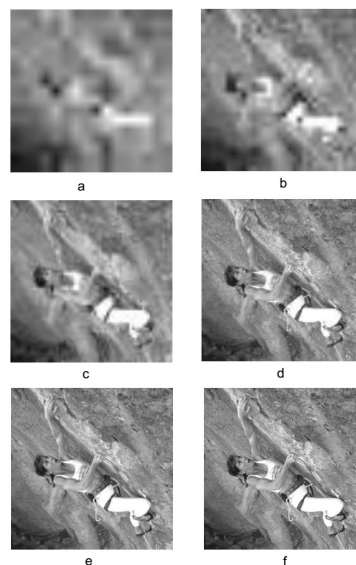


Figure 1.4: Échantillonnage d'une image

Cette figure montre une image analogique qui a été échantillonnée suivant un échantillonnage variable: 512 x 512 (f), 256 x 256 (e), 128 x 128 (d), 64 x 64 (c), 32 x 32 (b) et 16 x 16 (a).

1.1.4.3 La quantification:

Une autre façon de réduire la place mémoire nécessaire pour le stockage, elle procédé consistant à réduire l'ensemble des valeurs possibles d'une image numérique, par effectuer une valeur numérique à chaque échantillon prélevé sur le signale(après echantionallage). L'une dépend du capture, et de sa capacité effective à observer des signaux de valeurs

différentes: le rapport signal sur bruit est un indicateur de la qualité de la transmission d'une information. C'est le rapport des puissances entre :

1. l'amplitude des niveaux de gris mesurable par capteur:

$$(n_{max} - n_{min}) \quad (1.2)$$

2. le niveau de gris, en gros l'écart S_n de perturbation aléatoire qui affecte les niveaux de gris.

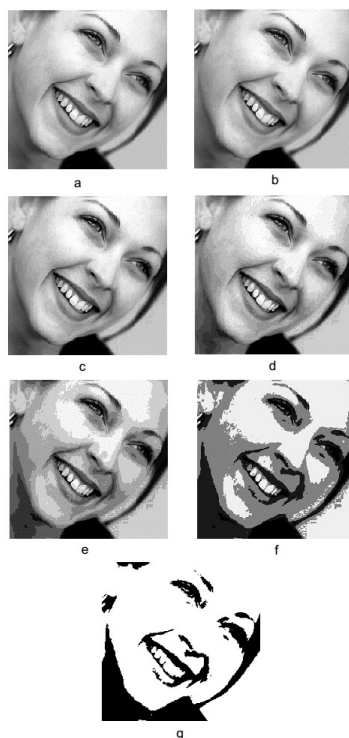


Figure 1.5: Quantification d'une image

cette figure la quantification d'une image sur 7 bits (a), sur 6 bits (b), sur 5 bits (c), sur 4 bits (d), sur 3 bits (e), sur 2 bits (f) et sur 1 bit (g)

1.1.5 Outils fondamentaux:

nous présentons une introductions aux outils d'analyse d'image numérique les plus courants, cette présentation est organisée selon les principaux modèles mathématiques qui été utilisé pour traiter les images.[15]

1.1.5.1 L'histogramme:

Outil de base pour l'étude des capteurs ou de la dynamique d'une scène, il est utilisé par certains opérateurs d'analyse. On retiendra cependant qu'il ne faut pas considérer l'histogramme comme une caractéristique fondamentale de l'image dans la mesure où on peut le transformer radicalement sans changer significativement l'image.[39]

1.1.5.2 La convolution:

C'est l'opérateur de base du traitement linéaire des images. Apparue très tôt dans les premiers systèmes d'analyse d'images sous forme empirique et justifiée par des considérations d'implantation, pour calculer une convolution, on remplace la valeur de chaque pixel par la valeur de produit scalaire entre les valeurs des noyau de la convolution et les valeurs de voisinage du pixel considéré. [26] Pour calculer la valeur du pixel (x,y)

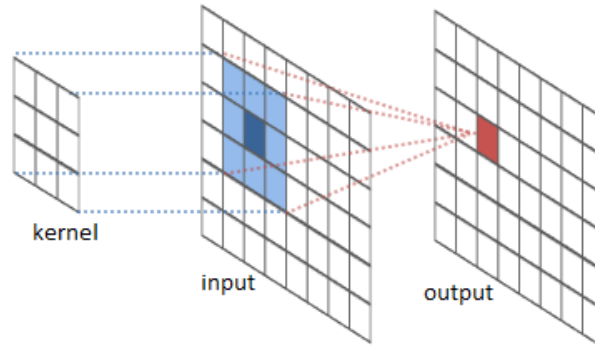


Figure 1.6: La Convolution

de l'image output (en rouge) on superpose le centre du kernel (ici une matrice 3×3) sur le pixel source en position (x,y) (en bleu foncé) et on multiplie deux à deux les valeurs du kernel et de la zone couverte en bleue entre elles puis on les additionnent pour trouver la valeur de la sortie.

La valeur du pixel sera donnée, dans le cas d'un noyau 3×3, par la formule:[26]

$$O(x, y) = \sum_{i=0}^2 \sum_{j=0}^2 k(i, j) * I(x - 1 + j, y - 1 + i) \quad (1.3)$$

1.1.5.3 La transformée de Fourier:

Outil fondamental d'analyse en traitement du signal, le pendant bidimensionnel de la **TF** et sa version discrète peut être appliqué avec profit aux images numériques. Si son utilisation en tant qu'outil analytique et algorithmique s'est estompée en traitement d'images au profit d'approches mieux adaptées à la localisation spatiale des fréquences (ondelettes). Dans certains cas, il est intéressant de faire la transformation inverse en reconstruisant une image dans le but d'obtenir celle-ci avant quantification. Cette technique permet entre autres d'effectuer un changement d'échelle (agrandissement ou réduction). soit le coefficient de fourrier $F(u,v)$, fournissent des information sur les fréquences (u,v) et permettent de manipulation dans le domaine fréquentiel. Transformée de fourrier discrète bidimensionnelle: (x,y) sont les coordonnées du domaine spatial: discrète: [26]

$$F(u, v) = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} f(x, y) e^{-2i\pi(ux+vy)/wh} \quad (1.4)$$

(u,v) sont les coordonnées du domaine fréquentiel: inverse:

$$f(x, y) = \frac{1}{wh} \sum_{u=0}^{w-1} \sum_{v=0}^{h-1} F(u, v) e^{2i\pi(ux+vy)/wh} \quad (1.5)$$

NB: h et w sont les height et weight d'image.

1.1.6 Filtrage des Images:

Un filtre est une transformation mathématique permettant, pour chaque pixel de la zone à laquelle il s'applique, de changer sa valeur en fonction des valeurs des pixels avoisinants, affectées de coefficients. Les techniques de filtrage permettent en outre de calculer ou amplifier les contrastes locaux. Le filtre est représenté par un tableau (matrice), caractérisé par ses dimensions et ses coefficients, dont le centre correspond au pixel concerné. Les coefficients du tableau déterminent les propriétés du filtre.

1.1.6.1 Les types de filtres:

Nous distinguons les types de filtres suivants :

1.1.6.1.1 Les filtres passe-bas: consistant à atténuer les composantes de l'image ayant une fréquence haute (pixels foncés). Ce type de filtrage est généralement utilisé pour atténuer le bruit de l'image, c'est la raison pour laquelle nous parlons habituellement de lissage (filtre anti-bruit). En effet, les filtres de lissage sont des opérateurs qui éliminent des éléments perturbateurs / non significatifs dans les images numériques, soit pour améliorer leur visualisation, soit pour les simplifier en vue d'un traitement postérieur.

1.1.6.1.2 Les filtres passe-haut: à l'inverse des passe-bas, atténuent les composantes de basse fréquence de l'image et permettent notamment d'accentuer les détails et le contraste, c'est la raison pour laquelle le terme de " filtre d'accentuation " est parfois utilisé. L'accentuation (ou bruitage) est l'inverse du lissage ; il s'agit d'une opération visant à accentuer les différences entre les pixels voisins. Ainsi l'accentuation peut permettre de mettre en exergue les limites entre les zones homogènes de l'image et est alors appelée extraction de contours (également contourage ou rehaussement de contours). Les filtres passe-bande permettant d'obtenir la différence entre l'image originale et celle obtenue par application d'un filtre passe-bas.

1.1.6.1.3 Filtrage dans le domaine spatial: par le filtrage spatial est un filtrage qui s'applique sur un voisinage d'un pixel dans une image. En effet, le concept de filtrage spatial consiste à déplacer un masque de convolution de pixel en pixel afin de parcourir toute l'image. Nous évoquerons un ensemble de filtres passe-haut, dont certains effectueront un filtrage linéaire et d'autres effectueront un filtrage non linéaire :

- Le filtre **moyen**, **gaussien** et **exponentiel** comme filtres linéaires passe-bas, et le filtre **médian** comme filtres non linéaires passe-bas.
- Les filtres **réhausseur**, de **Prewitt**, de **Sobel** et de **Laplace** comme filtres linéaires passe-haut.

1.1.7 détection de contours:

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative . Un contour ou bord peut être observé dans une image suite à :[39]

- Un changement brusque de luminosité peut également se produire dans l'objet.
- Changement d'orientation de surface
- Modifications de l'éclairage.
- Changements de réflectance.

le contour représente également une variation brusque de niveau de gris dans une image. Il peut être défini comme la figure ci-dessous:

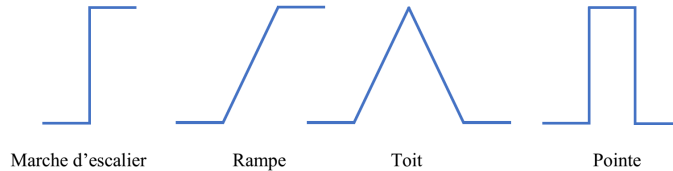


Figure 1.7: modèles de contours

- "marche d'escalier" si le contour est net.
- "rampe" si le contour est plus flou.
- "toit" s'il s'agit d'une ligne sur un fond uniforme.

Il existe plusieurs méthodes de détection des contours, Deux approches les plus utilisées :

- Approximations du gradient (dérivée première directionnelle)
- Approximations du Laplacien (dérivées secondes)

1.1.7.1 Approche gradient:

Le gradient, en un pixel d'une image numérique, est un vecteur caractérisé par son amplitude et sa direction.

La direction est orthogonale à la frontière qui passe au point considéré.

Le principe de la détection des contours par l'utilisation du gradient consiste à calculer d'abord le gradient de l'image dans deux directions orthogonales puis le module du gradient. Ensuite d'effectuer une sélection des contours les plus marqués.

Les directions des contours étant orthogonales à la direction du gradient déterminée en chaque pixel de l'image. En considérant l'image dans un repère orthogonal (Oxy) tel que (Ox) désigne l'axe horizontal et (Oy) l'axe vertical, le Gradient G de l'image (ou plutôt de la luminosité $I(x,y)$ en tout point ou pixel de coordonnées (x,y) est défini par:[18]

$$G = \nabla I(x,y) = \left(\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y} \right)^T \quad (1.6)$$

Il est donc caractérisé par un module m et une direction θ dans l'image :

$$m = \left(\frac{\partial I(x,y)}{\partial x}^2 + \frac{\partial I(x,y)}{\partial y}^2 \right)^{\frac{1}{2}} \quad (1.7)$$

$$\theta = \arctan\left(\frac{\partial I(x, y)}{\partial y} / \frac{\partial I(x, y)}{\partial x}\right) \quad (1.8)$$

1.1.7.2 La dérivée seconde

Les opérateurs de gradient vus précédemment exploitent le fait qu'un contour dans une image correspond au maximum du gradient dans la direction orthogonale au contour. Or le passage par zéro de la dérivée seconde d'une rupture d'intensité permet également de mettre en évidence le contour. La dérivée seconde est donc déterminée par le calcul du Laplacien:[18]

$$\nabla^2 I(x, y) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2} \quad (1.9)$$

Les filtres de convolution correspondant au laplacien d'une gaussienne respectivement de connexités 4 et 8 sont les suivantes:[37]

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

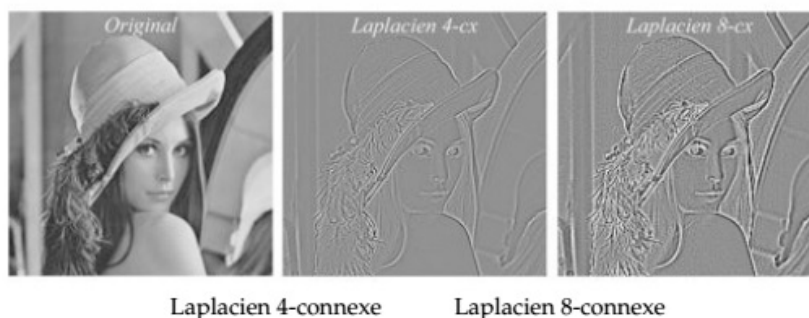


Figure 1.8: La dérivation seconde: Laplacien

1.2 Intelligence Artificielle:

1.2.1 Définition de l'intelligence Artificielle:

Le mot intelligence est du latin *intelligentia* signifiant la faculté de comprendre et plus précisément la capacité à lier les éléments entre eux. Elle est l'ensemble des capacités mentales nous permettant de comprendre ce qui nous entoure, de découvrir des relations nous permettant d'aboutir à une connaissance conceptuelle et rationnelle celle-ci étant en opposition avec les sensations et émotions. **Marvin Lee Minsky**, l'un des précurseurs

de la discipline définit l'intelligence artificielle comme la construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique.

1.2.2 Les Domaines d'application de l'intelligence artificielle:

L'intelligence artificielle se retrouve dans de nombreux domaines et sous domaines de la vie humaine:

- Apprentissage automatique.
- Reconnaissance des formes, des visages et la vision en général.
- Traitement automatique des langues (Ou NLP – “Natural Language processing” en anglais).
- Systèmes experts.

1.2.3 L'Apprentissage Automatique:

Définition:

L'apprentissage automatique (**Mitchell, 1997**) est un sous-domaine de l'intelligence Artificielle (IA) dont l'objectif est d'étudier les moyens permettant à une machine d'apprendre dans ce contexte, signifie être capable d'adapter son comportement en existence de situation inconnues (n'ayant pas été prévues par les concepteurs de la machine) et pouvoir extraire des lois à partir de bases de données d'exemples. L'apprentissage se fait donc par des outils qui permettent d'acquérir, élargir et améliorer les connaissances disponibles au système.

1.2.3.1 Types d'apprentissage:

Il existe deux types d'apprentissage : supervisé et non supervisé

1.2.3.2 Apprentissage non supervisé:

Dans le cadre de l'apprentissage non supervisé, la tâche consistera en la découverte de similarités entre les observations dans une collection d'exemples, dans le but de regrouper celles-ci en sous-ensembles, appelés clusters ou classes Parmi les algorithmes

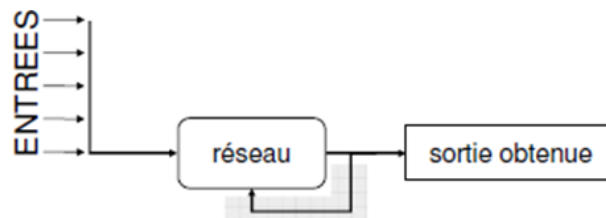


Figure 1.9: Apprentissage non supervisé

d'apprentissage non supervisé, les techniques utilisées classiquement dans le monde des statistiques :

- Classification ascendante hiérarchique (CAH).
- K-Means (Recherche des plus proches voisins).

1.2.3.3 Apprentissage supervisé:

Le principe de cette méthode c'est d'avoir une base d'apprentissage qui possède un ensemble de données étiqueté ou bien des exemples à qui en leur attribué des classes par un expert (superviseur), le but des méthodes d'apprentissage supervisé est de créer automatiquement des règles ou des fonctions de classement à partir de la base d'apprentissage, ces règles vont permettre à classifier un objet.

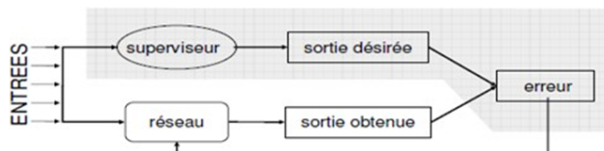


Figure 1.10: Apprentissage supervisé

1.2.3.3.1 Les algorithmes d'apprentissage supervisé

Il y a plusieurs algorithmes d'apprentissage supervisé la plus importante de ces méthodes.

Les Machines à Vecteur Support (SVMs): Les SVMs sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de la classification binaire et de la régression.

leurs Principales: la notion de la marge maximale et la notion de la fonction noyau. La marge maximale est utilisée pour les problèmes de la classification linéaire. Elle représente la distance entre la frontière de séparation et les échantillons d'apprentissage les plus proches. Ces derniers sont les vecteurs supports. Les fonctions noyau sont utilisées dans le cas des problèmes de la classification non-linéaire pour modifier l'espace de représentation des données d'entrées en un espace de plus grande dimension dans lequel il est probable qu'il existe de séparateurs linéaires.

Le classifieur du K plus proches voisins Le classifieur du **K Plus Proches Voisins (K Nearest Neighbors)** C'est un classifieur simple fondé sur le calcul de distance entre les exemples d'apprentissage et les exemples de tests, généralement la norme euclidienne est souvent utilisée comme mesure de distance, dans chaque étape de l'apprentissage, l'algorithme mémorise les k meilleurs exemples de l'ensemble d'apprentissage (kppv(x)) qui sont proches à l'exemple de test x. Cet algorithme est souvent performant s'il y a suffisamment d'exemples d'apprentissage, mais demande un temps de prédiction très long pour passer tous les exemples afin de voir les K meilleures solutions.

1.2.4 L'apprentissage en profondeur (deep Learning):

Le Deep Learning est un nouveau domaine de recherche de la machine Learning (ML), L'apprentissage en profondeur est un ensemble d'algorithmes d'apprentissage automatique qui tentent d'apprendre à plusieurs niveaux, correspondant à différents niveaux d'abstraction. Il a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations

linéaires et non linéaires et apprendre sur ces caractéristiques petites à petit à travers chaque couche avec une intervention humaine minimale.

1.2.4.1 Les différentes Architectures du Deep Learning:

1.2.4.1.1 Les réseaux de neurones convolutifs : Convolutional Neural Network (CNN) (réseaux de neurones convolutifs) sont un type de réseau de neurones spécialisés pour le traitement de données ayant une topologie semblable à une grille. Qui se sont avérés très efficaces dans des domaines tels que la reconnaissance et la classification d'images et vidéos. CNN a réussi à identifier les visages, les objets, panneaux de circulation et auto-conduite des voitures. Un réseau neuronal convolutif (CNN) est un algorithme d'apprentissage en profondeur qui peut reconnaître et classer les caractéristiques des images pour la vision par ordinateur. Récemment, les CNN ont été efficaces dans plusieurs tâches de traitement du langage naturel (telles que la classification des phrases). [12]

1.2.4.2 Les Éléments de base du réseau de neurones convolutifs:

Couche de convolution - un «filtre», parfois appelé «noyau», est passé sur l'image, affichant quelques pixels à la fois (par exemple, 3X3 ou 5X5). L'opération de convolution est un produit scalaire des valeurs de pixel d'origine avec des poids définis dans le filtre. Les résultats sont résumés en un nombre qui représente tous les pixels observés par le filtre.

Couche de regroupement(Pooling layer) réduit la quantité d'informations générées par la couche convolutive pour chaque entité et conserve les informations les plus essentielles.

Couche entièrement connectée: structure traditionnelle de perceptron multicouche. Son entrée est un vecteur unidimensionnel représentant la sortie des couches précédentes. Son résultat est une liste de probabilités pour différentes étiquettes possibles attachées à l'image (par exemple, chien, chat, oiseau). L'étiquette qui reçoit la probabilité la plus élevée est la décision de classification.

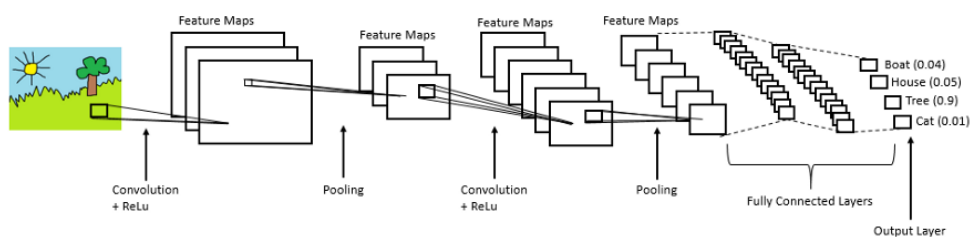


Figure 1.11: Architecture de CNN

1.2.4.2.1 Réseaux de neurones de convolution pour la détection d'objets:

-Les réseaux neuronaux convolutionnels basé sur des régions(R-CNN) : initialement proposé en 2014 par (Girshick et al), R-CNN est l'abréviation de «Region-based Convolutional Neural Networks». L'idée principale se compose de deux étapes. Premièrement, en utilisant la recherche sélective, il identifie un nombre gérable de candidats de région d'objet de boîte englobante («région d'intérêt» ou «RoI»). Et

puis il extrait les caractéristiques CNN de chaque région indépendamment pour la classification. [13]

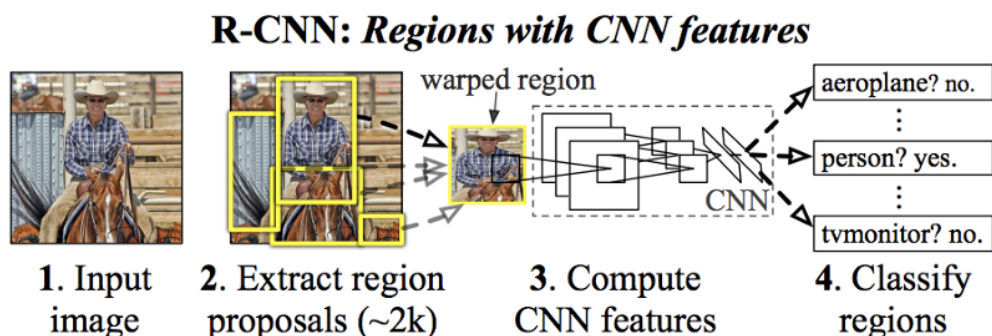


Figure 1.12: Architecture de R-CNN

Fast R-CNN Pour rendre R-CNN plus rapide, Girshick (2015) a amélioré la procédure de formation en unifiant trois modèles indépendants dans un cadre formé conjointement et en augmentant les résultats de calcul partagés, nommé Fast R-CNN. Au lieu d’extraire les vecteurs de caractéristiques CNN indépendamment pour chaque proposition de région, ce modèle les regroupe en un seul passage CNN sur toute l’image et les propositions de région partagent cette matrice de caractéristiques. Ensuite, la même matrice de caractéristiques est ramifiée pour être utilisée pour apprendre le classificateur d’objets et le régresseur de boîte englobante. En conclusion, le partage des calculs accélère R-CNN. [16]

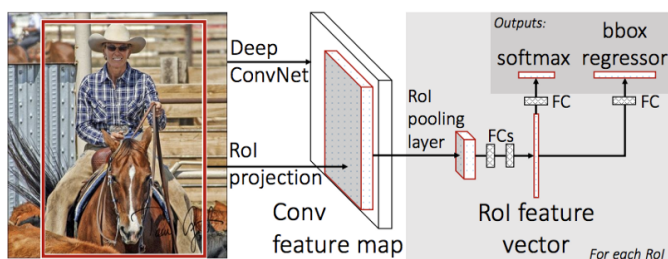


Figure 1.13: Architecture de Fast R-CNN

Faster R-CNN Les deux algorithmes ci-dessus (R-CNN et Fast R-CNN) utilisent la recherche sélective pour découvrir les propositions de région. La recherche sélective est un processus lent et chronophage qui affecte les performances du réseau. Par conséquent, Shaoqing Ren et al. a proposé un algorithme de détection d’objets qui élimine l’algorithme de recherche sélective et permet au réseau d’apprendre les propositions de région. Semblable à Fast R-CNN, l’image est fournie en tant qu’entrée à un réseau convolutif qui fournit une carte de caractéristiques convolutives. Au lieu d’utiliser un algorithme de recherche sélective sur la carte des caractéristiques pour identifier les propositions de région, un réseau séparé est utilisé pour prédire les propositions de région. Les propositions de région prédite sont ensuite remodelées en utilisant une couche de regroupement RoI qui est ensuite utilisée pour classer l’image dans la région proposée et prédire les valeurs de décalage pour les cadres de délimitation. [34]

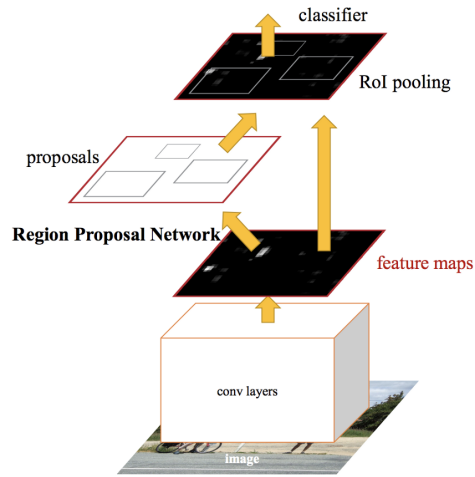


Figure 1.14: Faster R-CNN

	R-CNN	Fast R-CNN	Faster R-CNN
Test Time per Image	50 Seconds	2 Seconds	0.2 Seconds
Speed Up	1x	25x	250x

Tableau 1.1: comparaison entre RCNN,fast RCNN,Faster RCNN [21]

YOLO — You Only Look Once est un algorithme de détection d'objets très différent des algorithmes basés sur la région vus ci-dessus. Dans YOLO, un seul réseau convolutif prédit les boîtes englobantes et les probabilités de classe pour ces boîtes. Le fonctionnement de YOLO est que nous prenons une image et la divisons en une grille $S \times S$, dans chacune des grilles, nous prenons m boîtes englobantes. Pour chacune des boîtes englobantes, le réseau génère une probabilité de classe et des valeurs de décalage pour la boîte englobante. Les cadres de délimitation ayant la probabilité de classe supérieure à une valeur de seuil sont sélectionnés et utilisés pour localiser l'objet dans l'image.[21]

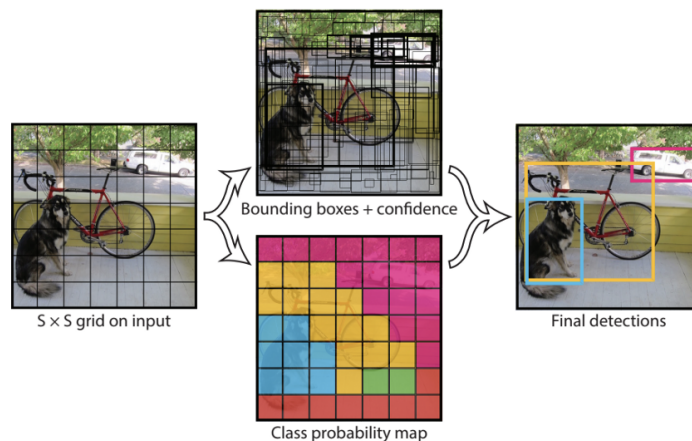


Figure 1.15: You Only Look Once(YOLO)

1.2.4.2.2 Réseau de neurones récurrents: L'idée derrière les **RNN Recurrent Neural Network** est d'utiliser des informations séquentielles. Dans un réseau neuronal traditionnel, nous supposons que toutes les entrées (et les sorties) sont indépendantes les unes des autres. Mais pour de nombreuses tâches, c'est une très mauvaise idée. Si on veut découvrir le prochain mot dans une phrase, il faut connaître les mots qui sont venus avant. Les RNN sont appelés récurrents, car ils exécutent la même tâche pour chaque élément d'une séquence, la sortie étant dépendante des calculs précédents. Une autre façon de penser les RNN est qu'ils ont une « mémoire » qui capture l'information sur ce qui a été calculé jusqu'ici. En théorie, les RNN peuvent utiliser des informations dans des séquences arbitrairement longues, mais dans la pratique, on les limite à regarder seulement quelques étapes en arrière.[12]

1.2.4.2.3 Modèle génératif: Si les modèles discriminatifs comme (CNN, RNN) sont utilisés pour prédire les données du label et de l'entrée, tant disque le modèle génératif décrit comment générer les données, il apprend et fait des prédictions en utilisant la loi de Bayes.[27] Cependant les modèles génératifs sont capables de bien plus que la simple classification comme par exemple générer de nouvelles observations.[27]

1.3 Reconnaissance de Caractères (O.C.R):

Définition:

La reconnaissance optique de caractères (OCR) est l'utilisation de la technologie pour distinguer les caractères de texte imprimés ou manuscrits à l'intérieur d'images numériques de documents physiques, tels qu'un document papier numérisé. Le processus de base de l'OCR consiste à examiner le texte d'un document et à traduire les caractères en code pouvant être utilisé pour le traitement des données. L'OCR est parfois également appelée reconnaissance de texte. [10] Les logiciels OCR peuvent également tirer parti de l'intelligence artificielle (IA) pour mettre en œuvre des méthodes plus avancées de reconnaissance intelligente de caractères (ICR), comme l'identification de langues ou de styles d'écriture manuscrite. Il n'existe pas un système universel d'O.C.R qui peut agir tous les cas de l'écriture, mais plutôt des voies d'approche dépendant du type de données traitées et bien évidemment de l'application visée . On utilise autrefois le terme O.C.R ou reconnaissance de caractère qui existe au domaine lecture automatique de document qui couvre trois axes :

- Lecture optique des caractères imprimés.
- La reconnaissance de caractères manuscrits.
- La reconnaissance de l'écriture manuscrite cursive.

1.3.1 Les phases d'un système OCR:

Un système de reconnaissance fait appel généralement aux étapes suivantes : Acquisition, pré traitements, segmentation, extraction des caractéristiques, classification, suivis éventuellement d'une phase de post-traitement. La figure 1.6 suivante représente globalement le processus général de reconnaissance de l'écriture.

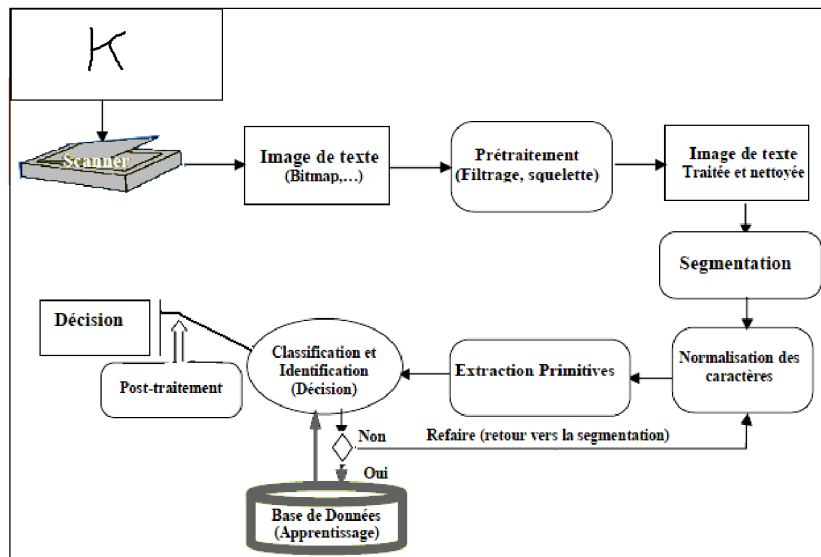


Figure 1.16: le processus général de reconnaissance de l'écriture

1.3.2 Les différents systèmes de reconnaissance de caractères:

La reconnaissance de caractères désigne le décodage de n'importe quel texte manuscrit ou imprimé ou information symbolique, par exemple les formules mathématiques etc. On présente dans la figure ci-dessous une dichotomie des types d'écriture faite selon le média, le support saisi et l'application considérée. La reconnaissance des caractères manuscrits est considérée plus difficile que la reconnaissance de l'imprimé pour lequel des systèmes opérationnels commerciaux existent.

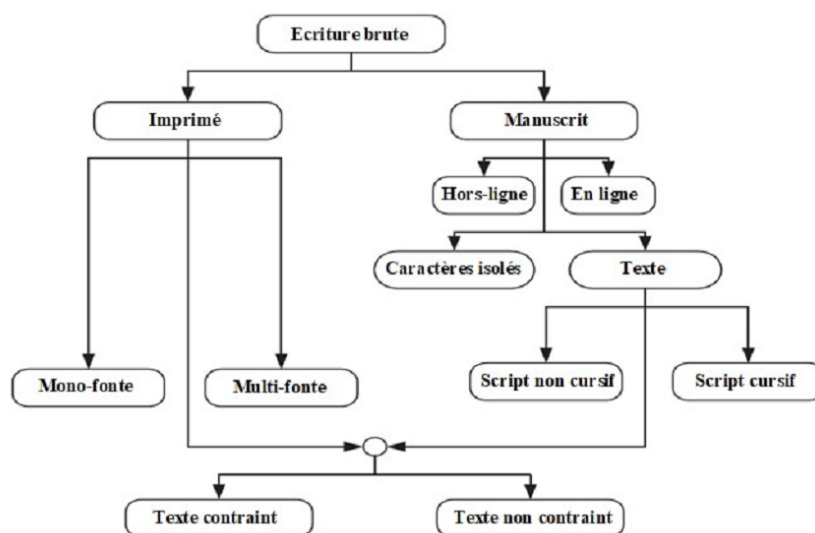


Figure 1.17: Panorama des types d'écriture

1.3.3 la reconnaissance de caractères imprimés:

L'invention de procédés d'impression par Gutenberg vers le milieu du quinzième siècle a changé notre vie par une diffusion plus large et plus rapide de connaissance, la plupart de patrimoine culturel et technique de l'humanité n'est encore disponible que sous forme de

document papier ; les entreprises et les collectivités sont ainsi confrontées à un besoin de retraitement, c'est-à-dire conversion rétrospective pour passer à un format électronique .on montre dans la figure ci-dessous les différentes étapes de passage de la forme papier vers des formes électroniques codées.

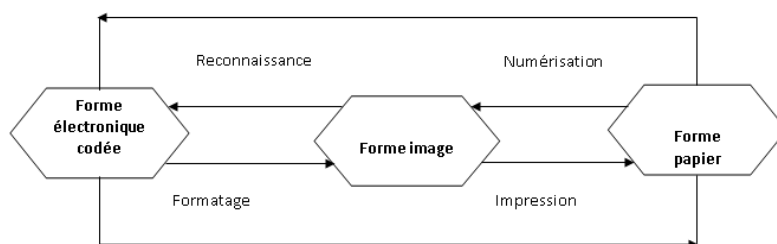


Figure 1.18: les différentes étapes de passage de la forme papier vers des formes électroniques codées

1.3.3.1 Caractéristiques des caractères imprimés:

La plupart des documents possède une partie importante où on trouve l'information textuelle nommée texte, en plus de ça il peut contenir des dessins, graphiques, schémas tableaux, formules mathématiques, photographies ...etc Dans le document présenté au système de reconnaissance de l'écriture imprimée, l'écriture est déterminée par le nombre de fontes (mono, multi ou omni-fonte). Le système de reconnaissance dans ce cas est généralement appelé OCR (pour Optical Character Recognition).

Un système OCR est dit mono-fonte s'il ne traite qu'une seule fonte à la fois. Ce cas est simple particulièrement au niveau de l'apprentissage car l'alphabet représenté est réduit.

-Un système OCR est dit multi-fonte s'il est capable de reconnaître divers types de fontes parmi un ensemble des fontes précédemment apprises.

-Un système OCR omni-fonte est capable de reconnaître toute fonte, généralement sans apprentissage, en se fondant sur les règles topologiques et morphologiques de l'écriture .

Le texte imprimé est composé de caractères minuscules, majuscules, signes de ponctuation, chiffres regroupés en ligne, paragraphes en bloc, ils ont une certaine forme et de taille changée, les unités qui sont les caractères nommés fonte ont leur attribut le nom corps qui correspond à la taille de ces fontes qui est déterminée par la hauteur totale des caractères les plus grands, plus les blancs de séparation avec les zones supérieures et inférieures, sans oublier le style de l'écriture, et comme il existe nombreux fontes pour un caractère on est obligatoire d'aller vers la standardisation des fontes. Il y a 3 types de fonte standardisés qui ont OCRA, OCR.B, OCR canadienne. **Notez que:**

- **ANSI:** American National Standard Institute
- **ECMA:** Européen Computer Manufacturers Association

1.3.4 la reconnaissance de caractères manuscrit

L'écriture manuscrite comme la parole fait partie de nos modes d'expression les plus évolués en tous les cas les deux sont très complexes, c'est un moyen particulier de codage de l'information qui concerne nos idées, pensées, nos sentiments, d'une autre façon la reconnaissance de l'écriture en général possède deux caractéristiques : Un liée au scripteur par exemple vérifier une signature d'un auteur. L'autre liée au sens

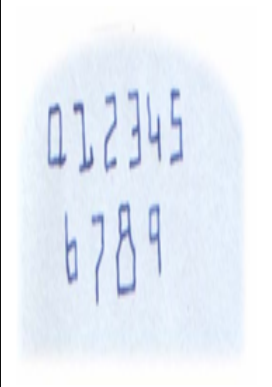
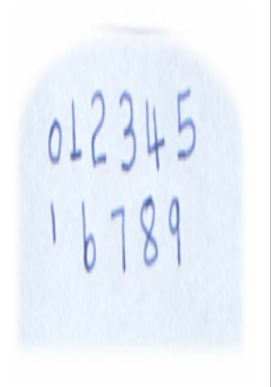
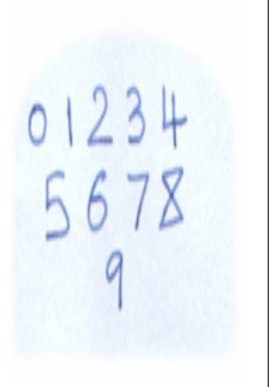
O.C.R.A 1960 ANSI	OCR.B 1965 ECMA	OCR CANADA
		

Tableau 1.2: quelques types de standardisation des Fontes

de ce qui est écrit (contextuelle). Contrairement à la reconnaissance de l'imprimé, la reconnaissance de l'écriture manuscrite soulève encore de nombreux problèmes, même si des avancées technologiques ont été enregistrées ces dernières années. Une des sources de ces problèmes est liée à l'extrême variabilité qui caractérise l'écriture. En effet, un texte manuscrit peut se présenter sous plusieurs formes: caractères isolés, texte cursif, texte non cursif, texte contraint ou non contraint. Cette diversité de l'écriture a été étudiée dans de nombreuses approches de reconnaissance proposées dans la littérature. Généralement, ces méthodes se distinguent par les données à traiter et par le style d'écriture utilisé.

1.3.4.1 Type de reconnaissance de l'écriture

La reconnaissance de l'écriture manuscrite est un traitement informatique qui a pour but de traduire un texte écrit en un texte codé numériquement. On peut distinguer deux types de reconnaissances.

1.3.4.1.1 La reconnaissance en-ligne (on-line): L'écriture en ligne est obtenue lors de sa réalisation par une saisie en continu du tracé. Elle concerne les nombreux objets électroniques de poche, tels qu'une tablette électronique munie de stylo spécial permettant de saisir du texte sans clavier. Les données se présentent alors sous la forme d'une séquence de points ordonnés dans le temps correspondant à la position du stylo. La reconnaissance en ligne est plus avantageuse que celle hors ligne à cause de la possibilité de correction et de modification de l'écriture de manière interactive, Les systèmes de reconnaissance en ligne sont principalement employés dans le domaine de sécurité, tels que la certification d'auteur ou la vérification de signature.[10]

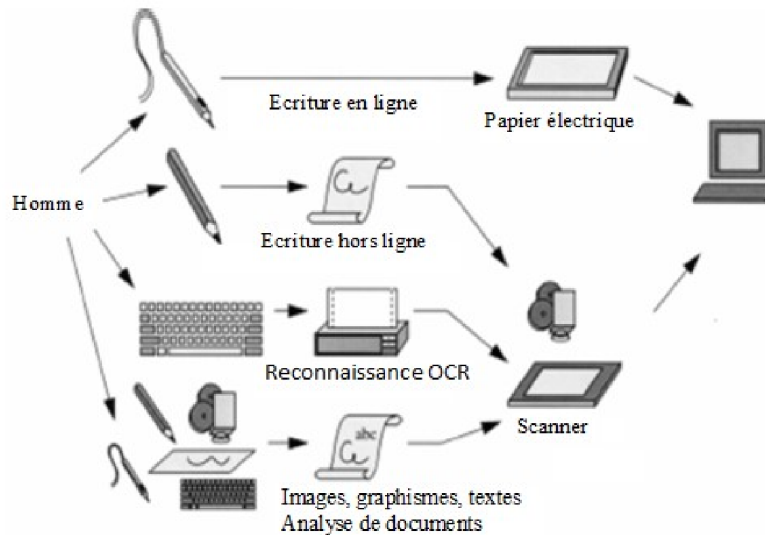


Figure 1.19: Systèmes de reconnaissance de l'écriture

1.3.4.1.2 Reconnaissance hors ligne (off-line): Dans le cas des systèmes hors ligne, la reconnaissance est effectuée une fois que l'écriture est présente sur un support en papier. L'acquisition des données se fait généralement à l'aide d'un scanner ou d'une caméra. Ainsi, l'information obtenue est bidimensionnelle. Contrairement aux systèmes en ligne, les systèmes hors ligne ne disposent pas de l'information temporelle et dynamique du tracé.

1.3.4.1.3 processus de la reconnaissance OCR: La première phase est l'acquisition a pour but la numérisation de l'image entrée par un scanner ou un capture externe.

le but de la phase des prétraitements et normalisation est la réduction de bruit et les variabilités de l'écriture pour garder seulement les informations utiles

La phase d'extraction des primitives permet de faire une description synthétique de la forme à reconnaître dans un espace à plusieurs dimensions, tous à partir de la représentation de l'image

L'étape de segmentation elle permet de décomposer l'image d'un texte en entités (mots, caractères ou graphèmes) pour réduire la complexité des modules de traitements plus tard.

La phase apprentissage consiste à trouver les modèles les plus adaptées aux entrées du problème. Le résultat est une base de données d'apprentissage qui constitue la base de référence de système

pour la phase de la classification du caractère ou du mot, la description du caractère à reconnaître a partir de la base de test est comparée aux descriptions des caractères de la base de référence.

En plus ,on pouvons trouver une phase de post-traitement qu'est une opération facultative et pour le but de consiste à vérifier et peut-être à corriger l'hypothèse de reconnaissance générée par le module de classification .

1.3.4.2 les Approches de reconnaissance:

Il y a deux approches s'opposent en reconnaissance des mots : globale et analytique:

L'approche globale: Cette approche a une vision générale du mot, elle se base sur une description unique de l'image du mot, vue comme une entité indivisible. Disposant de beaucoup d'informations, elle absorbe plus facilement les changements au niveau de l'écriture. L'approche globale est généralement appliquée pour diminuer la liste de mots candidats dans le contexte d'une reconnaissance à grands vocabulaires (mais limitée dynamiquement : pouvant être réduit de façon dynamique, comme l'ensemble des noms de rues associés à un bureau de poste distributeur).

L'approche analytique: À l'inverse à l'approche globale, l'approche analytique cherche à reconnaître les caractères ou sous-caractères (graphèmes) issus de la segmentation (séparation de mots, des caractères) pour reconstituer les mots. La difficulté d'une telle approche a été clairement évoquée par **Sayre** en 1973 et peut être résumée par le dilemme suivant : "pour reconnaître les lettres, il faut segmenter le tracé et pour segmenter le tracé, il faut reconnaître les lettres". Cette approche est la seule applicable dans le cas de grands vocabulaires. Elle peut s'adapter simplement à un changement de vocabulaire. Elle permet théoriquement une discrimination plus fine des mots puisque elle se base sur la reconnaissance des lettres qui la composent et il est possible de récupérer l'orthographe du mot reconnu.

Conclusion

Dans ce présent chapitre nous avons dressé un état de l'art sur les notions de base qui servent de fondement à la compréhension de différentes techniques de traitement d'images et la vision artificielle. Comme en traitement du signal, le filtrage, la détection des contours est un outil de base en traitement d'images .

Nous avons également examiné brièvement l'intelligence artificielle et sa relation et la reconnaissance des caractères.

Dans le second chapitre, nous parlons plus en détail sur les divers méthodes de détection et extraction des tableaux dans un image

Détection et extraction des tables

Introduction

2.1 Problématique d'extraction des tableaux:

Le problème de la détection de table est extrêmement difficile en raison du degré élevé de variabilité intra-classe, Ça signifie il est difficile de donner une définition formelle de ce à quoi ressemble un tableau en raison des différentes dispositions, de l'utilisation erratique de la règle lignes pour la délimitation de table ou de structure, ou simplement en raison de contenus de table très divers, De plus, il y a souvent un degré significatif de similitude entre les classes avec d'autres objets potentiellement présents dans les documents (par exemple : graphiques, listes de codes ou organigrammes) . Cela rend particulièrement difficile de créer manuellement une Dataset pour décrire les structures tabulaires. En raison de l'utilisation continue de documents papier, en particulier Les universités et les environnements commerciaux et d'entreprise, ainsi que l'abondance de données tabulaires à l'intérieur, les pipelines de traitement de documents dépendent de mécanismes de compréhension de table très précis. Il existe déjà des approches disponibles pour détecter et décomposer les tableaux mais ces systèmes reposent généralement sur des heuristiques ad hoc et des métadonnées supplémentaires extraites par exemple de fichiers PDF. L'extraction de tableaux à partir de fichiers PDF atténue certaines des complexités du travail avec des images brutes en raison des métadonnées disponibles pendant le traitement. Le problème est beaucoup plus difficile lorsque la détection et la reconnaissance de structure ont besoin à effectuer sur des images brutes.

2.1.1 Détection de table à partir de documents image:

l'extraction des tableaux ont fait l'objet d'un grand volume de recherche. De nombreuses approches tentent d'extraire des tableaux à partir de documents image. Ils se concentrent principalement sur les composants de mise en page (limites de tableau, limites de cellule, etc.) pour détecter et décomposer les tableaux. Habituellement, ils détectent les zones de table potentielles de l'ensemble documents et isolent les cellules du tableau en analysant les informations d'espace. La plupart des recherches dans ce domaine se concentre sur l'analyse de l'image raster. Certaines œuvres adoptent l'optique technique de reconnaissance de caractères (OCR) pour extraire le texte des parties segmentées.

Et il existe d'autres travaux récents qui utilisent des techniques d'intelligence artificielle pour détecter et extraire des tables, notamment le Deep learning. Du point de vue de la méthodologie, toutes les méthodes peuvent être divisées en trois catégories: basées sur une mise en page prédéfinie, basées sur des heuristiques et basées sur des statistiques.[7] Les algorithmes prédéfinis basés sur la mise en page fonctionnent généralement bien pour un domaine, mais sont difficiles à étendre en raison de leur dépendance à des règles artisanales. Heuristique les méthodes basées sur les méthodes nécessitent un post-traitement complexe et les performances reposent largement sur le choix des fonctionnalités et la qualité des données d'entraînement (Dataset).[7]

2.2 Travaux connexes sur l'extraction de tables

au cours des 15 dernières années, plus de deux cents articles ont été publiés sur table champ de reconnaissance. Zanibbi [33] fournit une enquête avec une description détaillée des approches et systèmes. Une bibliographie détaillée sur la segmentation et l'extraction de tableaux peut être trouvée dans ce lien <http://www.visionbib.com/bibliography/char969.html>[8]. Pour extraire les données des tables, nous devons accomplir deux tâches distinctes: détecter la limite du tableau à partir du document et extraire également les cellules du tableau.

2.2.1 L'apprentissage profond en action:

Le Deep Learning, ou apprentissage profond, est un sous-ensemble du Machine Learning, ou apprentissage automatique, basé sur des réseaux neuronaux artificiels. Le processus d'apprentissage est qualifié de profond parce que la structure des réseaux neuronaux artificiels se compose de plusieurs couches d'entrée, de sortie et masquées. Chaque couche contient des unités qui transforment les données d'entrée en informations que la couche suivante peut utiliser une tâche prédictive spécifique. Grâce à cette structure, une machine est capable d'apprendre au travers de son propre traitement de données.[24]

Les réseaux neuronaux artificiels sont formés par couches de nœuds connectés. Les modèles de Deep Learning utilisent des réseaux neuronaux comportant grand nombre de couches.

Il existe différentes architectures telles que les NN à convolution, les NN récurrents, les auto-encodeurs, les NN Adversaires génératifs pour traiter différents types de données. Celles-ci sont complexes mais présentent des performances élevées pour résoudre les problèmes en temps réel. Examinons maintenant les recherches qui ont été menées dans le domaine de l'extraction de tables à l'aide des réseaux de neurones et examinons-les brièvement.

2.2.1.1 TableNet:

Introduction: TableNet est une architecture moderne d'apprentissage en profondeur qui a été proposée par une équipe de **TCS Research** en 2019. La motivation principale était d'extraire des informations de tables numérisées via des téléphones portables ou des caméras. Ils ont proposé une solution qui comprend une détection précise de la région tabulaire dans une image, puis la détection et l'extraction d'informations à partir des lignes et des colonnes du tableau détecté.[29]

2.2.1.1.1 DataSet: le DataSet utilisé était **Marmot**. Il contient 2000 pages au format PDF qui ont été rassemblées avec les vérités de terrain correspondantes. Cela inclut également les pages chinoises.[29]

2.2.1.1.2 Architecture de TableNet L'architecture est basée sur **Long et al.**, Un modèle codeur-décodeur pour la segmentation sémantique. Le même réseau encodeur / décodeur est utilisé comme architecture **FCN (Fully Convolutional Network)** pour l'extraction de table.[5] Les images sont pré-traitées et modifiées à l'aide de l'**OCR Tesseract** . Le modèle est dérivé en deux phases en soumettant l'entrée à des techniques d'apprentissage en profondeur. Dans la première phase, ils ont utilisé les pondérations d'un réseau **VGG-19(est un réseau neuronal convolutif de 19 couches de profondeur)** pré-formé.[6] Ils ont remplacé les couches entièrement connectées du réseau **VGG** utilisé par des couches convolutionnelles **1x1** . Toutes les couches convolutives sont suivies de l'activation **ReLU** et d'une couche d'abandon de probabilité 0,8. Ils appellent la deuxième phase le réseau décodé qui se compose de deux branches. C'est selon l'intuition que la région de colonne est un sous-ensemble de la région de table. Ainsi, le réseau de codage unique peut filtrer les régions actives avec une meilleure précision en utilisant les caractéristiques des régions de table et de colonne. La sortie du premier réseau est distribuée aux deux branches. Dans la première branche, deux opérations de convolution sont appliquées et la carte des caractéristiques finale est mise à l'échelle pour respecter les dimensions de l'image d'origine. Dans l'autre branche de détection des colonnes, il y a une couche de convolution supplémentaire avec une fonction d'activation **ReLU (Unité Linéaire Rectifiée)** et une couche d'abandon avec la même probabilité d'abandon que celle mentionnée précédemment.[5] Les cartes d'entités sont suréchantillonnées en utilisant des convolutions fractionnées après une couche de convolution (1x1). Ci-dessous, une image de l'architecture:[29]

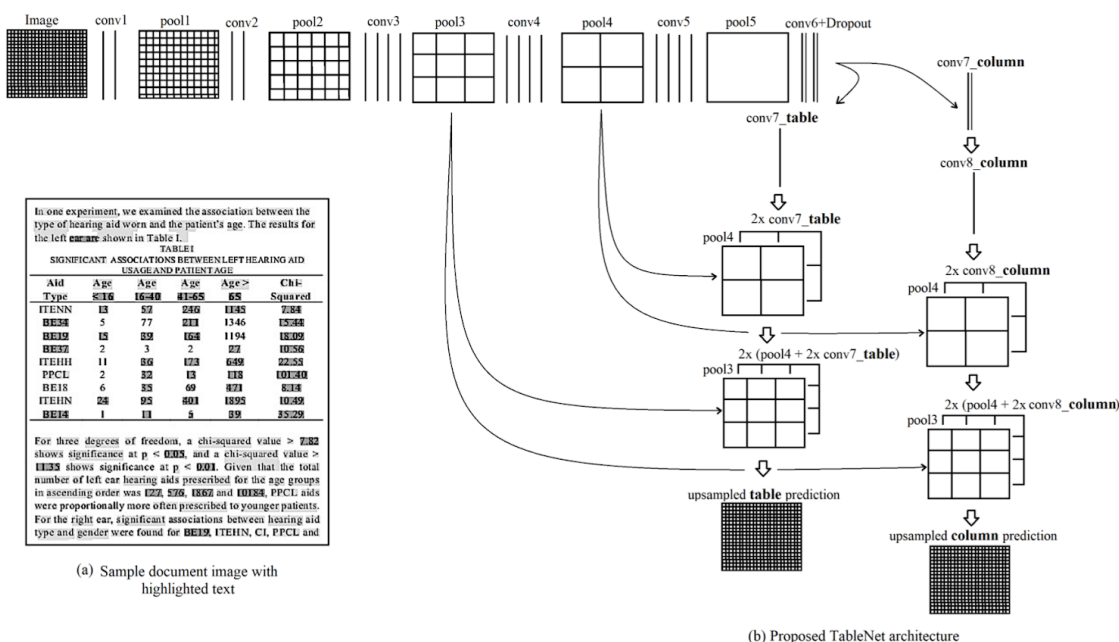
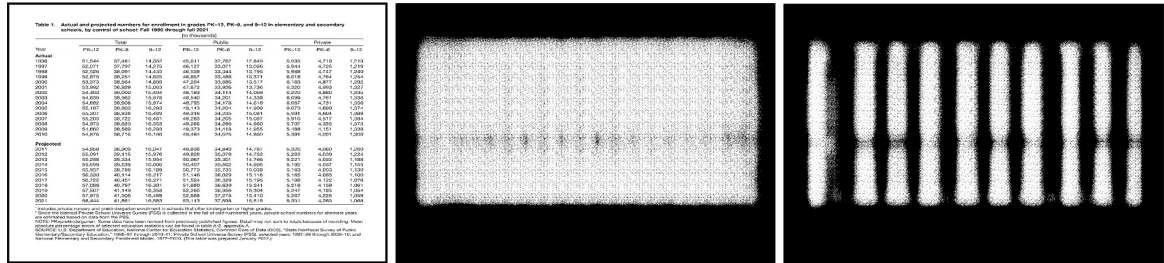


Figure 2.1: Architecture de TableNet

2.2.1.1.3 Les sorties Une fois les documents traités à l'aide du modèle, les masques des tables et des colonnes sont générés. Ces masques sont utilisés pour filtrer le tableau et ses régions de colonne de l'image. En utilisant désormais l'OCR Tesseract, les informations sont extraites des régions segmentées. Ci-dessous, une image montrant les masques générés puis extraits des tables.[5]



(a) Figure showing the raw document image. (b) Generated table mask after processing. (c) Generated column mask after processing.

Figure 2.2: Exemple d'image de document et ses masques de sortie générés après le traitement à partir de TableNet

Ils ont également proposé le même modèle qui est affiné avec **ICDAR(International Conference on Document Analysis and Recognition)** qui a mieux performé que le modèle original. Le rappel, la précision et le score F1 du modèle affiné sont:[5]

Model	Recall	Precision	F1-Score
TableNet + Semantic Features (fine-tuned on ICDAR)	0.9628	0.9697	0.9662
TableNet + Semantic Features	0.9621	0.9547	0.9583
TableNet	0.9501	0.9547	0.9547

Tableau 2.1: Résultat de destruction des tableau

2.2.1.2 DeepDeSRT:

Introduction: DeepDeSRT (**Deep Learning for Detection and Structure Recognition of Tables in Document Images**) est un framework de réseau neuronal utilisé pour détecter et comprendre les tables dans les documents ou images, Il a deux solutions comme mentionné dans le nom de cette méthode:[36]

- Il présente une solution basée sur l'apprentissage en profondeur pour la détection de table dans les images de document.
- Il propose une nouvelle approche basée sur l'apprentissage en profondeur pour la reconnaissance de la structure des tables, c'est-à-dire l'identification des lignes, des colonnes et des positions de cellule dans les tables détectées.

Le modèle proposé est entièrement basé sur des données, il ne nécessite pas d'heuristique ou de métadonnées des documents ou des images. L'un des principaux avantages de la technique est qu'ils n'ont pas utilisé de grands DataSets. au lieu de cela, ils ont utilisé le concept d'apprentissage par transfert et d'adaptation de domaine pour la détection de table et la reconnaissance de la structure de table.

2.2.1.2.1 DataSet: L'ensemble de données utilisé est un compétition de DataSet de table **ICDAR 2013** contenant 67 documents avec 238 pages au total.[36]

2.2.1.2.2 Architecture de DeepDeSRT:

Détection de table: Le modèle proposé utilisait **Fast RCNN** (**F**ast **R**egion-based **C**onvolutional **N**etwork) comme méthode de base pour la détection des tables. L'architecture se décompose en deux parties différentes[17].

Dans la première partie, ils ont généré des propositions de région basées sur l'image d'entrée par un réseau de proposition de région (RPN).[17]

Dans la deuxième partie, ils ont classé les régions en utilisant Fast-RCNN. Pour soutenir cette architecture, ils ont utilisé **ZFNet** [42] et les **VGG-16**.

Reconnaissance de la structure: Une fois qu'une table a été détectée avec succès et que son emplacement est connu par du système, le défi suivant pour comprendre son contenu est de reconnaître et de localiser les lignes et les colonnes qui constituent la structure physique de la table. Par conséquent, ils ont utilisé un réseau entièrement connecté avec les poids de **Network VGG-16** qui extrait les informations des lignes et des colonnes. Voici les résultats de DeepDeSRT:[36]

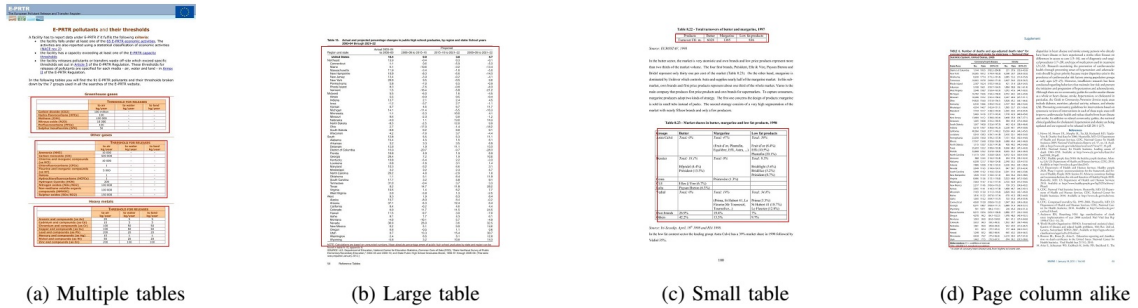
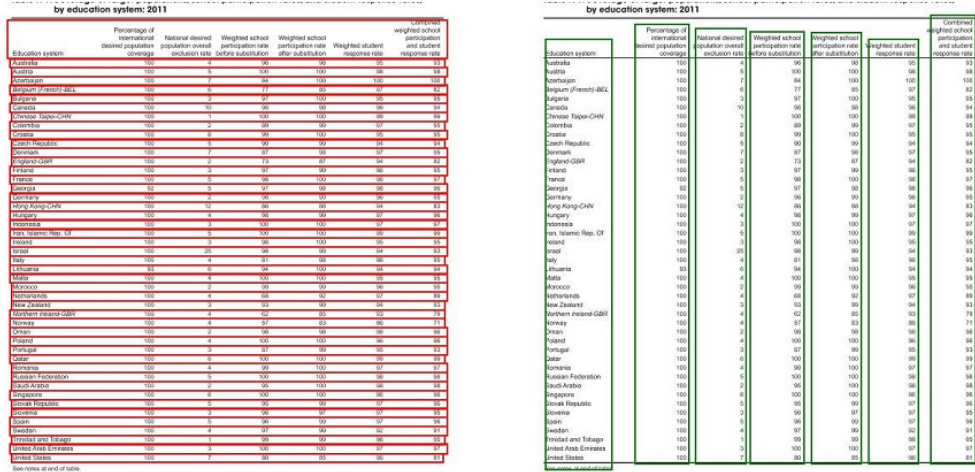


Figure 2.3: Détection Table par DeepDeSRT (ICDAR 2013)



(a) Row detection, no ruling lines present (b) Column detection, no ruling lines present

Figure 2.4: reconnaissance Table par DeepDeSRT (ICDAR 2013)

Les résultats de l'évaluation révèlent que DeepDeSRT pour la détection de table et la

reconnaissance de structure et et atteint des F1-mesures de 96,77% et 91,44% pour la détection de table et la reconnaissance de structure, respectivement jusqu'en 2015.[36]

2.2.1.3 Graph des réseaux de neurones:

Introduction: Dans cette recherche, les auteurs du Deep Learning Laboratory du National Center of Artificial Intelligence (NCAI) ont proposé des réseaux de neurones graphiques pour extraire des informations de tableaux. Ils ont fait valoir que les réseaux de graphes sont un choix plus naturel pour ces problèmes et ont exploré plus en détail deux réseaux de neurones graphiques basés sur le gradient.

Ce modèle proposé combine les avantages des réseaux de neurones convolutifs pour l'extraction de caractéristiques visuelles et des réseaux de graphes pour traiter le problème de structure.[32]

2.2.1.3.1 DataSet: Les auteurs ont proposé un nouveau grand DataSet généré synthétiquement de 0,5 million de tableaux répartis en quatre catégories. qui sont visualisées dans la figure ci après.[32]

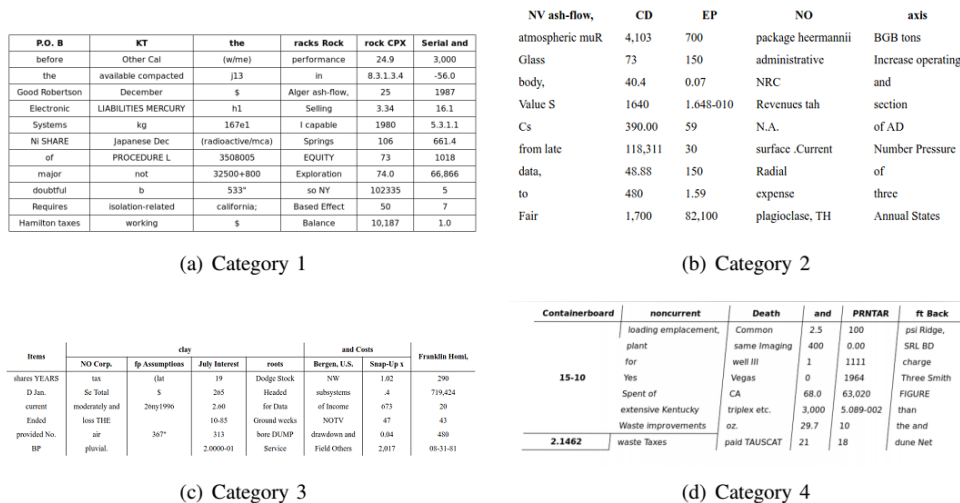


Figure 2.5: Images avec différentes catégories de difficulté. Les images de catégorie 1 sont des images simples sans fusion et avec des lignes dominantes. Catégorie 2 ajoute différents types de frontières, y compris l'absence occasionnelle de lignes directrices. La catégorie 3 est la plus difficile qui introduit la cellule et la colonne fusion. Modèles de catégorie 4 images capturées par caméra par transformation de perspective linéaire

- Les images sont des images simples sans fusion et avec des lignes dominantes.
- Introduit la fusion de cellules et des colonnes.
- Les images ont différents types de bordure, y compris l'absence occasionnelle de lignes dominantes.
- La caméra a capturé des images avec la transformation de perspective linéaire.

2.2.1.3.2 Architecture: ils ont utilisé un réseau convolutif peu profond qui génère les caractéristiques convolutives respectives. Si les dimensions spatiales des entités en

sortie ne sont pas les mêmes que l'image d'entrée, elles collectent des positions qui sont redimensionnées linéairement en fonction du rapport entre les dimensions d'entrée et de sortie et les envoient à un réseau d'interaction qui a deux réseaux de graphes appelés **DGCNN** (Deep Graph Convolutional Neural Network)[43] et **GravNet**[31]. Les paramètres du réseau de graphes sont les mêmes que le CNN d'origine. En fin de compte, ils ont utilisé un échantillonnage de paires d'exécution pour classer le contenu extrait qui utilisait en interne l'algorithme basé sur **Monte Carlo**: [35]

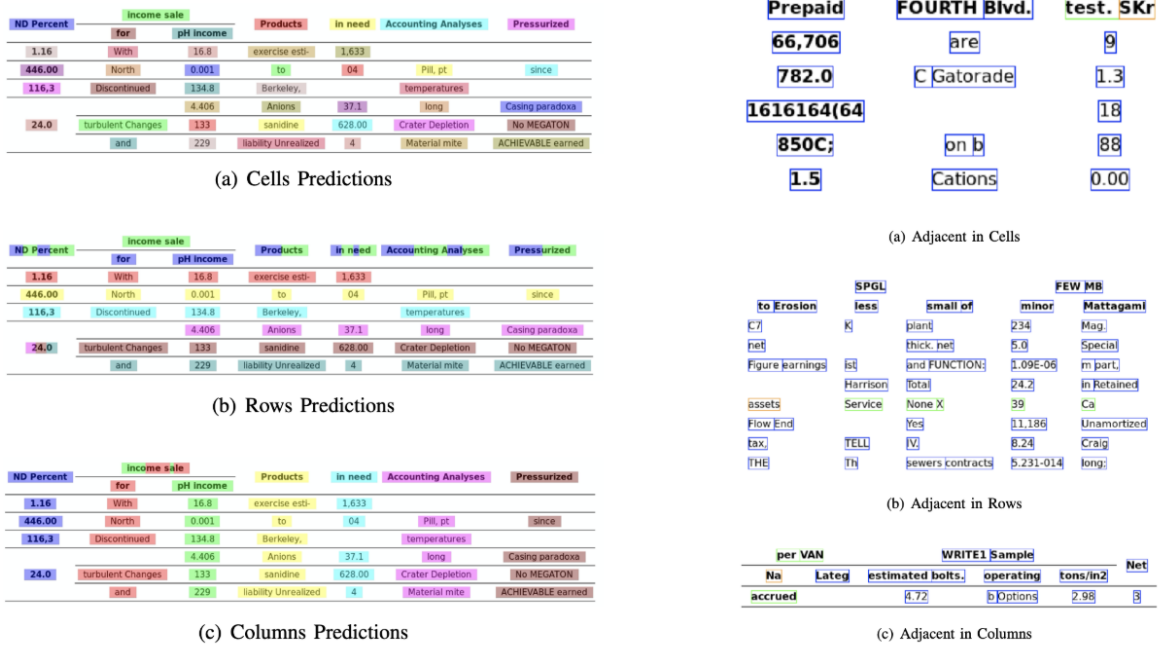


Figure 2.6: Sorties générées par Graph Neural Networks

ci-dessous les résultats d'extraction des tables générés par les réseaux pour quatre catégories du réseau:

Méthode	Catégorie 1	Catégorie 2	Catégorie 3	Catégorie 4
FCNN	42.4	54.6	10.9	31.9
GravNet	65.6	58.6	13.1	31.5
DGCNN	96.9	94.7	52.9	68.5

Tableau 2.2: comparaison Resultat extraction des Tables(04 Catégories)

2.2.1.4 CGAN et algorithmes génétiques:

Introduction: CGAN (Conditional Generative Adversarial Networks) Dans cette recherche, les auteurs ont utilisé une approche descendante au lieu d'utiliser une approche ascendante(intégrer des traits dans des cellules, des lignes ou des colonnes). Dans cette méthode, en utilisant un réseau antagoniste génératif, ils ont mappé l'image de la table dans une forme de table standardisée «squelette». Ce tableau squelette indique les bordures approximatives des lignes et des colonnes sans le contenu du tableau, Ensuite, ils adaptent les rendus des structures de table latentes candidates à la structure du squelette en utilisant une mesure de distance optimisée par un algorithme génétique. [23]

2.2.1.4.1 DataSet: Les auteurs ont utilisé leur propre ensemble de données contenant 4 000 tables.

2.2.1.4.2 Architecture: Le modèle proposé se compose de deux parties. Dans la première partie, les images d'entrée sont résumées dans des tableaux squelettes à l'aide d'un réseau neuronal antagoniste génératif conditionnel.

Un **GAN** a encore deux réseaux, le générateur qui génère des échantillons aléatoires et un discriminateur qui dit si les images générées sont fausses ou originales. Le générateur **G** est un réseau codeur-décodeur où une image d'entrée est passée à travers une série de couches de sous-échantillonnage progressif jusqu'à une couche de goulot d'étranglement où le processus est inversé. Pour transmettre suffisamment d'informations aux couches de décodage, une architecture **U-Net** avec des connexions de saut est utilisée et une connexion de saut est ajoutée entre les couches **i** et **n-i** via la concaténation, où **n** est le nombre total de couches, et **i** est le numéro de couche dans l'encodeur. Une architecture **PatchGAN** est utilisée pour le discriminateur **D**. Cela pénalise la structure de l'image de sortie à l'échelle des patches. Ceux-ci produisent la sortie sous forme de tableau squelette.[23]

Dans la seconde partie, ils optimisent l'ajustement des structures de données latentes candidates à l'image squelette générée en utilisant une mesure de la distance entre chaque candidat et le squelette. C'est ainsi que le texte à l'intérieur des images est extrait. Ci-dessous, une image illustrant l'architecture.

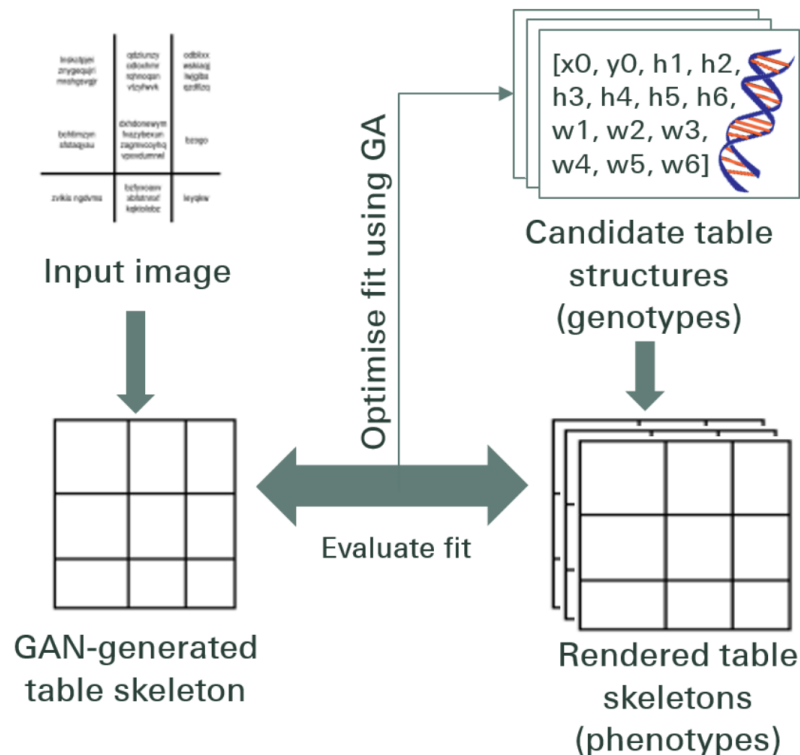


Figure 2.7: Schéma général de l'approche CGAN

Sortie: Les structures de table estimées sont évaluées en comparant - Numéro de ligne et de colonne, position du coin supérieur gauche, hauteurs de ligne et largeurs de colonne.

L'algorithme génétique a donné une précision de 95,5% par ligne et 96,7% de précision par colonne lors de l'extraction des informations des tableaux.[23]

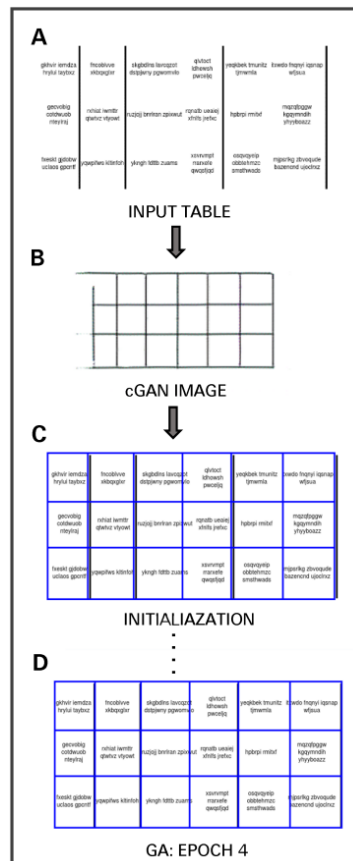


Figure 2.8: Exemple d'estimation de la structure du table

2.2.2 Détection de table avec OpenCV:

Introduction:

OpenCV (Open Source Computer Vision Library):est une bibliothèque de logiciels open source de vision par ordinateur et d'apprentissage automatique.

ils utilisèrent les techniques de vision par ordinateur pour extraire les informations des tables numérisées, ils besoin d'un image de table d'abord (les tableaux d'où nous devons extraire les informations), puis l'utilisation la vision par ordinateur, puis ils essaieront de trouver les bordures, les bords et les cellules. il utiliseront différents filtres et contours, et ils mettrons en évidence les fonctionnalités principales des tableaux.[25] Cette méthode est généralement utilisée lorsque les informations de table sont prédéfinies.

Nous avons d'abord besoin d'une photo, qui peut être prise ou utiliser une photo existante.[22]

ci-dessous l'image qui a été chargée.

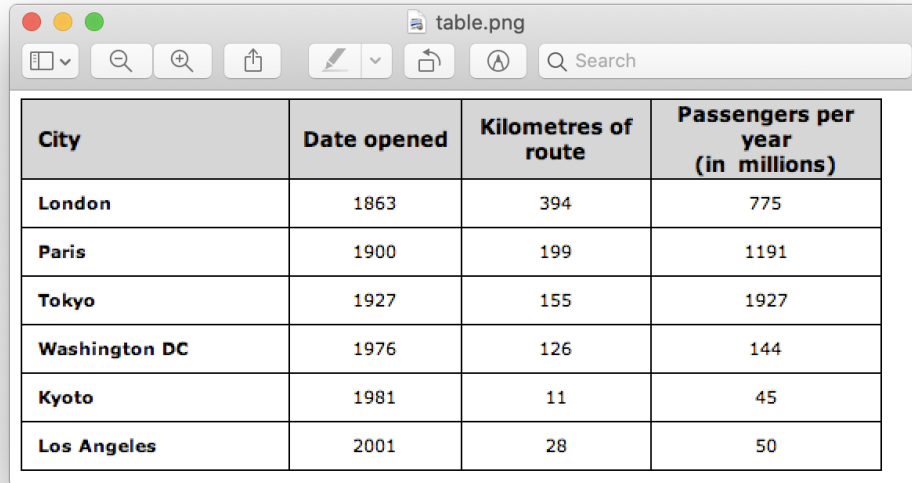


Figure 2.9: charge image pour traitement OpenCv

puis l'utilisation d'une technique appelée **Inverse Image Thresholding** qui améliore les données présentes dans l'image chargée.

```
ret, thresh_value = cv2.threshold(
    table_image_contour, 180, 255, cv2.THRESH_BINARY_INV)
```

Figure 2.10: Inverse Image Thresholding

Une autre étape importante du pré-traitement est la dilatation de l'image. La dilatation est une opération mathématique simple appliquée aux images binaires (noir et blanc) qui élargit progressivement les limites des régions de pixels.

```
kernel = np.ones((5,5),np.uint8)
dilated_value = cv2.dilate(thresh_value,kernel,iterations = 1)
```

Figure 2.11: la dilatation de l'image

Dans OpenCV, on utilise la méthode, **findContours** pour obtenir les contours dans l'image. Cette méthode prend trois arguments, le premier est l'image dilatée (la méthode **findContours** ne prend en charge que les images binaires), le second est le **cv2.RETR_TREE** qui est utilisé pour la récupération de contour, le troisième est le **cv2.CHAIN_APPROX_SIMPLE** qui est le mode d'approximation de contour. Le **findContours** décompresse deux valeurs, il faut donc ajouter une autre variable nommée **hierarchy**. Lorsque les images sont imbriquées, les contours dégagent une interdépendance. Pour représenter de telles relations, le variable hiérarchie est utilisée.[38]

```

contours, hierarchy = cv2.findContours(
    dilated_value, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

```

Figure 2.12: le Méthode de FinContours

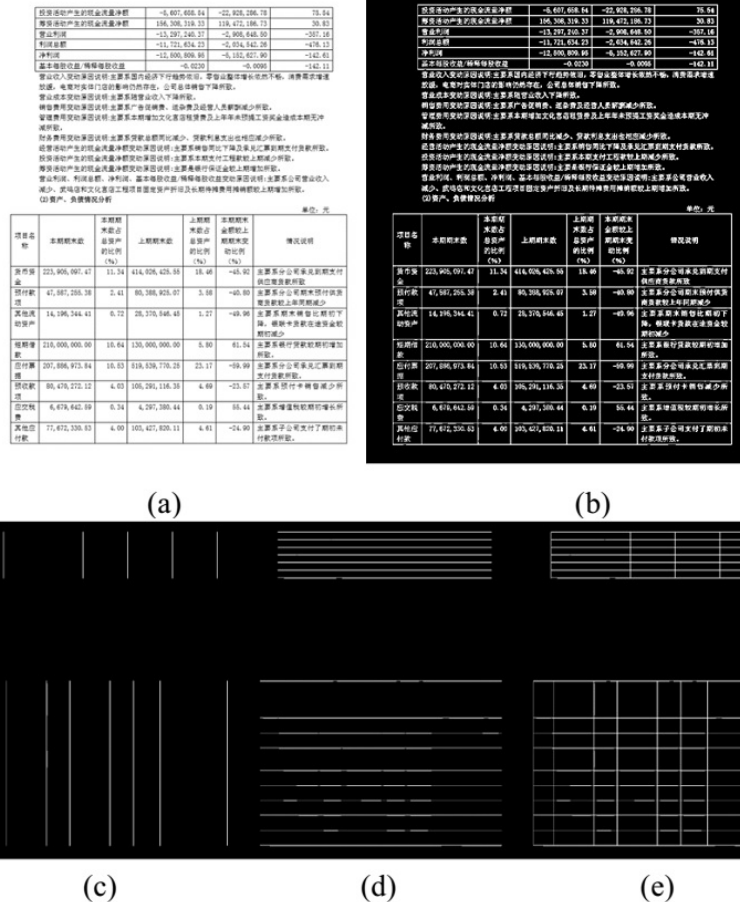


Figure 2.13: Extraction de lignes de tableau: (a) L'image d'origine, (b) l'image binaire, (c) le l'extraction de lignes de table verticales, (d) l'extraction de lignes de table horizontales, (e) le table combinée.

Les contours indiquent exactement où les données sont présentes dans l'image. Maintenant, ils parcourent la liste des contours que nous avons calculée à l'étape précédente et calculent les coordonnées des boîtes rectangulaires telles qu'observées dans l'image d'origine à l'aide de la méthode, **cv2.boundingRect**. Dans la dernière itération, ils ont placé ces boîtes sur l'image originale **table_image** en utilisant la méthode, **cv2.rectangle()**.^[30]

```

for cnt in contours:
    x, y, w, h = cv2.boundingRect(cnt)
    # bounding the images
    if y < 50:
        table_image = cv2.rectangle(table_image, (x, y), (x +
w, y + h), (0, 0, 255), 1)

```

Figure 2.14: les Méthodes boundingRect et rectangle

C'est la dernière étape. Ici, ils utilisent la méthode **namedWindow** pour rendre le table avec le contenu extrait et les contours incorporés dessus. ci-dessous le code:[30]

```
plt.imshow(table_image)
plt.show()
cv2.namedWindow('detecttable', cv2.WINDOW_NORMAL)
```

Figure 2.15: La méthode namedWindow OpenCV

实 例	种 属	重量 (平均)	颜色 (主)	生命周期/保质期
非洲象	动物	5t	土灰色	70年
大白鲨	动物	3.2t	灰白色	70年
苹果	植物	250g	红色	5-15天
梨	植物	300g	黄色	5-10天

(a)

实 例	种 属	重量 (平均)	颜色 (主)	生命周期/保质期
非洲象	动物	5t	土灰色	70年
大白鲨	动物	3.2t	灰白色	70年
苹果	植物	250g	红色	5-15天
梨	植物	300g	黄色	5-10天

(b)

Figure 2.16: Contenu et Contours Image avec OpenCV

et quand les tables extraites, nous pouvons exécuter chaque recadrage de contour (image de chèque cellule) via le moteur Tesseract OCR. Une fois que nous avons des boîtes de chaque texte, nous pouvons les regrouper en fonction de leurs coordonnées x et y pour dériver à quelle ligne et colonne elles appartiennent..

2.2.3 les outils Open Source:

Il existe une gamme d'outils disponibles pour détecter et extraction de tableaux à partir de documents. Certains d'entre eux fournissent API en Python pour utiliser facilement les outils d'extraction. Certains des outils que nous ont étudié sont Tabula, Camelot et PyPDF2.

2.2.3.1 Tabula:

Tabula est un outil open-source développé à l'origine pour Java. maintenant, Tabula fournit également une API en Python qui fonctionne sans problème. L'API Python prend le PDF comme d'entrée ainsi qu'un certain nombre d'arguments qui aident à détecter différentes dispositions de tableaux. L'un des arguments est l'argument «pages» qui prend le numéro de page input et permet d'extraire des tables spécifiques. Il a également un Argument «stream» qui prend une valeur booléenne. Il y a deux formats pour extraire les PDF: le format «stream» est pour l'extraction de tableaux sans bordure alors que le format «(lattice)treillis» est utilisé pour l'extraction de tables clairement délimitées. Tabula fournit également une application qui a une interface utilisateur pour aider l'extraction de tableaux. L'application offre la possibilité de sélectionner la zone de la page qui est ensuite analysée par Tabula pour détecter une structure tabulaire.[4]

nous pouvons installer tabula-py via PIP, on utilise la commande: `pip install tabula-py` pour installer Tabula sous plateforme Windows ou Mac: <https://tabula.technology/>

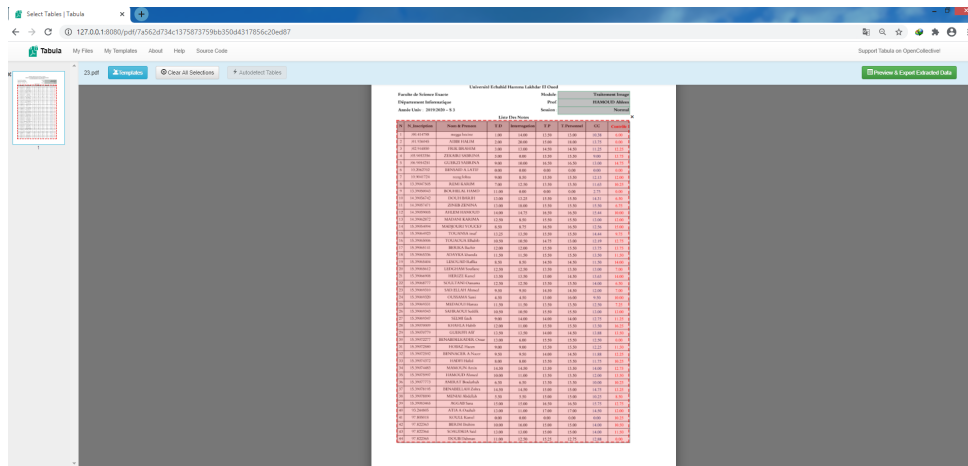


Figure 2.17: Interface Tabula Web Server

2.2.3.2 Camelot:

Camelot est un autre package disponible en Python pour extraction de tables. Il présente de nombreuses similitudes avec Tabula. Il a également des arguments pour le numéro de page et le (stream/lattice) format. Camelot fournit également une interface Web appelée **Excalibur** qui est similaire à Tabula UI en termes de fonctionnalité. La différence entre les deux est que Camelot fonctionne mieux sur les images.

Camelot et Tabula fonctionnent bien pour détecter des tables bien définies, bien que leurs performances sur les mises en page non structurées peuvent être erronées. Les deux packages donner un objet dataframe comme sortie qui peut être converti au format csv ou au format Excel facilement. Celles-ci les bibliothèques sont pratiques lors de l'extraction de quelques tables connues des documents, mais travaillant sur de grandes DataSet avec des tables non structurées peuvent se révéler difficile sans une logique robuste pour détecter un type similaire de les tables.[1]

nous pouvons installer au python : `pip install "camelot-py[cv]"`

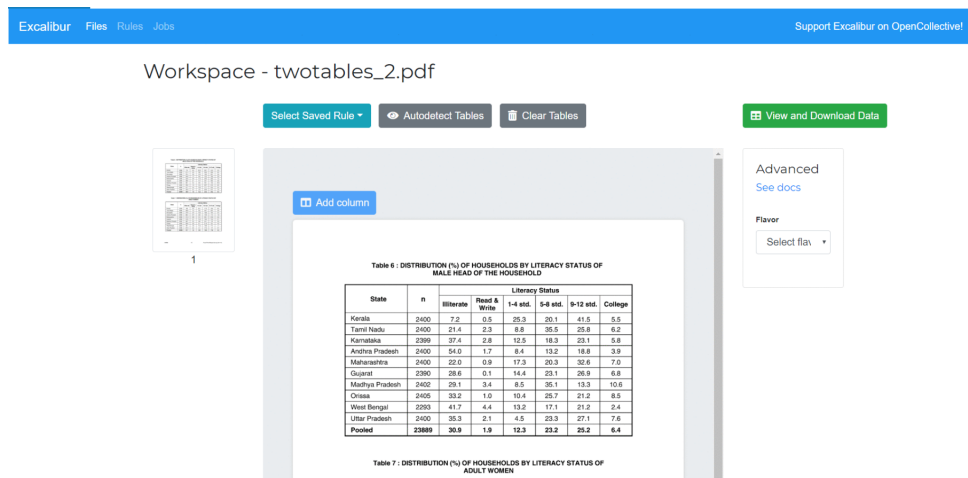


Figure 2.18: Interface Excalibur de Camelot

2.2.3.3 PyPDF2:

PyPDF2 est un package open-source en Python qui peut extraire des données de PDF au format texte, il ne fait pas reconnaître la disposition des tableaux, il extrait juste les données au format texte. Ce package est utile pour extraire du texte à partir de nombreux fichiers PDF peut être utilisé plus tard dans le traitement d'Application de la langue Naturel (LNP).[9]

commande installation : pip install PyPDF2 [2]

2.2.4 bref résumé des méthodes de reconnaissance des tableaux:

Dans le tableau ci après, nous avons brièvement résumé les méthodes de détection et d'extraction de table, sur la base de son méthodologie

Méthode	Auteur	méthodologie
DeepDeSRT	Sebastian et al	Deux modèles d'apprentissage en profondeur distincts pour la détection de table et la reconnaissance de structure de table
TabelNet	Shubham et al	Un modèle unique d'apprentissage en profondeur de end to end pour la détection et la reconnaissance de structure de table
TableBnak	Mingho Li et al	Basé sur l'image modèle d'apprentissage en profondeur (deeplearning) de détection et de reconnaissance de table
OpenSource outils		Input: PDF Output: Dataframe objects A simple API is provided in Python Example: Camelot, Tabula
OpenCv et pytesseract		Input: Image Output: Dataframe objects programmation script pour détecte le table par OpenCv et extraire le texte par Pytesseract
utilisation Graph Neural Networks	Shah Rukh Qasim, Hassan Mahmood, Faisal Shafait	il est divisé en trois parties: la convolution réseau de neurones pour l'extraction des caractéristiques de l'image, l'interaction réseau de communication entre les sommets et le partie de classification pour étiqueter tous les sommets appariés comme étant adjacent ou non adjacent
CGANs et Genetic Algorithms	Nataliya Le Vine Matthew Zeigenfuse Mark Rowan	les images d'entrée sont résumées dans des tableaux squelettes pour ce la on utilisant un réseau neuronal antagoniste génératif conditionnella , la deuxième partie, ils optimisent l'ajustement des structures de données latentes candidates à l'image squelette générée

Tableau 2.3: résumé les méthodes de détection et d'extraction de table [9]

Conclusion

Dans ce chapitre, nous avons examiné en détail l'extraction d'informations à partir de tables. Nous avons vu comment des technologies modernes telles que le Deep Learning et la vision par ordinateur peuvent automatiser les tâches banales en créant des algorithmes robustes pour produire des résultats précis. Dans les sections initiales, nous avons découvert le rôle de l'extraction de tables dans la facilitation des tâches des individus, des industries et des secteurs d'activité, codé un algorithme utilisant la vision par ordinateur pour trouver la position des informations dans les tableaux en utilisant des techniques de seuillage, de dilatation et de détection des contours, et l'extraction des informations de tableau.

Chapitre 3

Conception

Introduction

Le but de notre projet est la réalisation d'une application utilisant pour la comparaison deux liste des notes. Dans ce chapitre, nous allons présenter le systèmes de détection et extraction de tableau , puis la comparaison avec un fichier, Dans cet chapitre, nous allons présenter une conception du système en donnant une vue globale du système, ensuite nous allons détailler chaque module composant le système séparément.

3.1 Architecture globale de système:

On générale notre système se compose de Quatre parties :le premier est la pré traitement de l'image de table, le second et détection des horizontale de table puis détecte le table lui-même, le troisième est l'allocation approfondie des cellules à la ligne et à la colonne appropriées , la quatrième partie est l'extraction de chaque cellule allouée et la reconnaissance optique de caractères (OCR) avec Pytesseract, le cinquième est de comparaison entre les Résultat d'extraction et le fichier original. Tout d'abord, nous avons besoin des données d'entrée, qui sont dans notre cas une image au format png ou Jpeg. Le but est d'avoir une image et un fichier Excel avec la même structure tabulaire, où chaque cellule peut être modifiée et utilisée pour une analyse plus approfondie. Ci-dessous l'architecture de notre système :

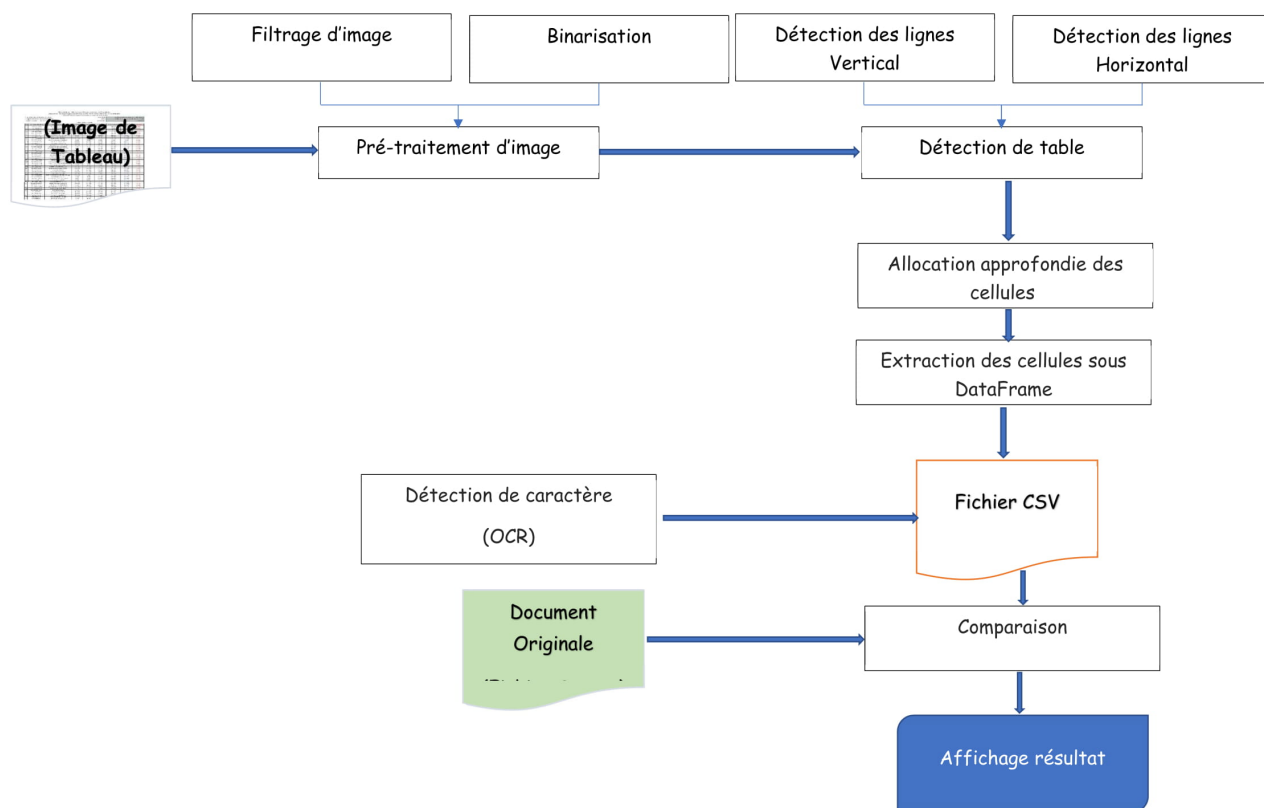


Figure 3.1: Architecture du système pour reconnaissance et extraction de table

3.2 Explication d'architecture:

3.2.1 Pré-traitement d'image:

Pour améliorer les fonctionnalités des images de table pour un traitement ultérieur, un algorithme de prétraitement a été proposé pour la reconnaissance de table avec ses procédures globales illustrées. Comme le montre la figure ci dessous, l'image de la table est

d'abord débruitée [41], puis l'image en niveaux de gris est obtenue pour rendre le contour et le contenu du tableau plus clairs. La méthode de binarisation d'image est optimisée et sa robustesse est améliorée, conduisant à une diminution significative de la quantité d'informations soumises à un traitement ultérieur. Ensuite, une détection de bord sur l'image binaire est effectuée pour obtenir des informations de contour périphérique. Ensuite, la correction d'inclinaison et la correction de perspective sont appliquées pour les rectangles réguliers et irréguliers, respectivement. Il existe de nombreux algorithmes de réduction du bruit développés pour le pré traitement des images de table, tels que le filtrage moyen, le filtrage lisse de masque et le filtrage médian. Parmi ces algorithmes de réduction de bruit existants, le filtrage moyen peut être défini comme l'algorithme de filtrage médian, en tant que méthode efficace largement utilisée dans le prétraitement d'image, traite l'image avec une fenêtre glissante. Son processus général consiste en deux étapes : définir la fenêtre glissante avec le pixel cible comme centre et remplacer la valeur du pixel cible par la médiane de tous les pixels de la fenêtre. Les algorithmes de la binarisation, qui peuvent améliorer le composant de premier plan et affaiblir le composant d'arrière-plan, jouent également un rôle essentiel dans le pré traitement des images de table. La précision de détection des informations de bord et des caractères dans la reconnaissance de table dépend de l'effet de la binarisation. Après la binarisation, les images du tableau ne conservent que les informations principales telles que les lignes du tableau et les caractères de texte.

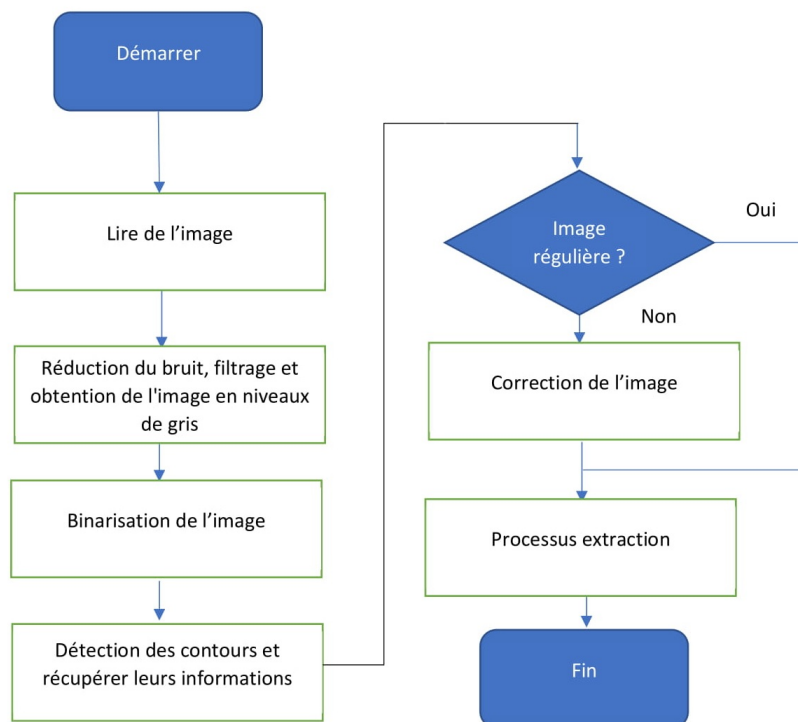


Figure 3.2: processus de pré traitement

3.2.1.1 Correction automatique de l'inclinaison à l'aide de détecteurs de coin et d'homographie:

Dans cet sous étape, nous parlerons de l'une des principales applications de l'homographie : la correction du biais et comment nous pouvons y parvenir. Nous utiliserons, `cv2.findHomographie()` pour calculer la matrice d'homographie et `cv2.warpPerspective()` pour transformer les images.

Formellement, l'homographie (ou parfois la colinéation) est une cartographie d'un plan à lui-même de sorte que la colinéarité de tout ensemble de points soit préservée. Il transforme les points colinéaires en points colinéaires dans un plan.

Le mappage d'homographie H entre deux points x et x' représenté par : $x' = H*x$ [28]

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Figure 3.3: Formule Homography

par exemple, nous allons essayer de corriger le biais dans une image plus réaliste. soit une image de table (liste des notes). L'image est inclinée, notre objectif ici est de transformer cette image en vue aérienne. Pour atteindre notre objectif, nous devons trouver des coins au bord du plan extérieur image et non à l'intérieur. Pour cela, nous masquons la zone extérieure et utilisons la détection de contour, La première étape est de lire l'image la convertissons en RVB format.

Masquage, seuillage et détection des contours: les étapes de masquage, seuillage et détection des contours, pour l'étap principale de correction de l'image de table sont qui suites:

- Pour masquer l'image de table, nous créons un noyau 5x5 et l'appliquons sur l'image en niveaux de gris.
- on appliquons le seuil d'Otsu[20] pour la classification binaire de l'image; c'est-à-dire classer chaque pixel en noir ou en blanc. Cela facilite les choses car maintenant, nous pouvons détecter le contour de la table d'image sans aucun problème.
- Pour la détection des contours, on utilise `cv2.findContours ()`. Nous ne nous intéressons qu'au plus grand contour. Alors, c'est ce que nous prenons.
- `cv2.drawContours ()` est utilisé pour dessiner les points de contour

Détection des coins (en utilisant l'algorithme de Douglas-Peucker) Maintenant que nous avons obtenu le contour d'image de tableau, nous devons trouver les points d'angle. Nous utiliserons `cv2.approxPolyDP ()`. Cette fonction utilise l'algorithme de **Douglas-Peucker**[40] pour approcher une courbe ou un polygone avec moins de sommets de sorte que la distance entre eux soit inférieure à epsilon. Par conséquent, en utilisant `cv2.approxPolyDP ()` avec la valeur epsilon appropriée, nous pouvons obtenir une assez bonne approximation de notre contour . Nous pouvons exploiter cette fonctionnalité pour obtenir nos points d'angle.

Calcul d'homographie et transformation de perspective : Maintenant nous utiliserons `get_destination_points()` pour obtenir les sommets de coin correspondants de notre image de table non déformée et `unwarp()` pour effectuer une transformation de perspective en utilisant l'homographie. Enfin, nous allons recadrer la zone de l'image de table pour supprimer l'arrière-plan indésirable.

3.2.2 Détection de table:

L'étape suivante consiste à définir un noyau pour détecter les boîtes rectangulaires, puis la structure tabulaire. Tout d'abord, nous définissons la longueur du noyau et suivons les noyaux verticaux et horizontaux pour détecter toutes les lignes verticales et toutes les lignes horizontales. Nous combinons les lignes horizontales et verticales pour détecter chaque cellule. Après avoir eu la structure tabulaire, nous utilisons la fonction `findContours` pour détecter les contours. Cela nous aide à récupérer les coordonnées exactes de chaque boîte. (In OpenCV, finding contours is like finding white object from black background. So remember, object to be found should be white and background should be black.)[19]

Nous combinons les lignes horizontales et verticales en une troisième image, en pondérant les deux avec 0,05. Le but est d'obtenir une structure tabulaire claire pour détecter chaque cellule.

3.2.3 Allocation approfondie des cellules:

Les étapes supplémentaires sont nécessaires pour définir le bon emplacement, c'est-à-dire la colonne et la ligne appropriées, de chaque cellule. Tout d'abord, la récupération de la hauteur de chaque cellule et la stocker dans les hauteurs de la liste. puis, nous prenons la moyenne des hauteurs. Ensuite, la récupération de la position, la largeur et la hauteur de chaque contour et le stockons dans la liste de boîtes. Ensuite, nous dessinons des rectangles autour de toutes nos boîtes et traçons l'image. Maintenant que nous avons chaque cellule, son emplacement, sa hauteur et sa largeur, nous devons obtenir le bon emplacement dans le tableau. Par conséquent, nous devons savoir dans quelle ligne et dans quelle colonne elle se trouve. Nous calculons le nombre maximum de colonnes (c'est-à-dire les cellules) pour comprendre le nombre de colonnes de notre trame de données / table finale. Après avoir eu le nombre maximum de cellules, nous sauvegardons le point médian de chaque colonne dans une liste, créons un tableau et trions les valeurs.

À ce stade, nous avons toutes les rectangles (les boîtes) et leurs valeurs. enfin on stockons la séquence appropriée dans les boîtes finales (`CelluleFin`) de la liste.

3.2.4 Extraction des valeurs de cellules:

Dans cette étape, on utilise nos (`CelluleFin`) de liste. Chaque image de cellule est dilatée et érodée avant de réaliser la reconnaissance optique de caractères. Ensuite, nous appliquons deux méthodes d'extraction des informations ou des valeurs des cellules:

- Nous donnons les images des cellules au `pytesseract` afin de reconnaître les chaînes qui contiennent de l'information. Les mêmes étapes sont faites pour chaque cellule et la valeur est sauvegardée dans une autre liste. on utilise la fonction de `openCV` (`pytesseract.image_to_string`). Finalement, on convertit la liste en un `DataFrame` pour la sauvegarder dans un fichier sous Format CSV.

- Nous appliquons la méthode d'extraction d'informations ou valeurs proposée.

3.2.4.1 Méthode d'extraction des valeurs proposée:

La méthode proposée commence par la segmentation de l'image de cellule en cours en un ensemble de segments d'images de caractères (chiffres) en se basant sur l'espace blanc entre ces caractères ("0..9", "." et "/"). Par la suite, ces différents petits segments sont donnés à une fonction de miniaturisation afin de minimiser la taille des images. Finalement, les segments d'images miniaturisés sont passés à une fonction de classification à base de ratio et de distance euclidienne (Voir la figure ci-dessous). Cette distance est calculé entre les deux matrices des images l'une est celle du caractère à reconnaître et l'autre est celle des chiffres de bases.

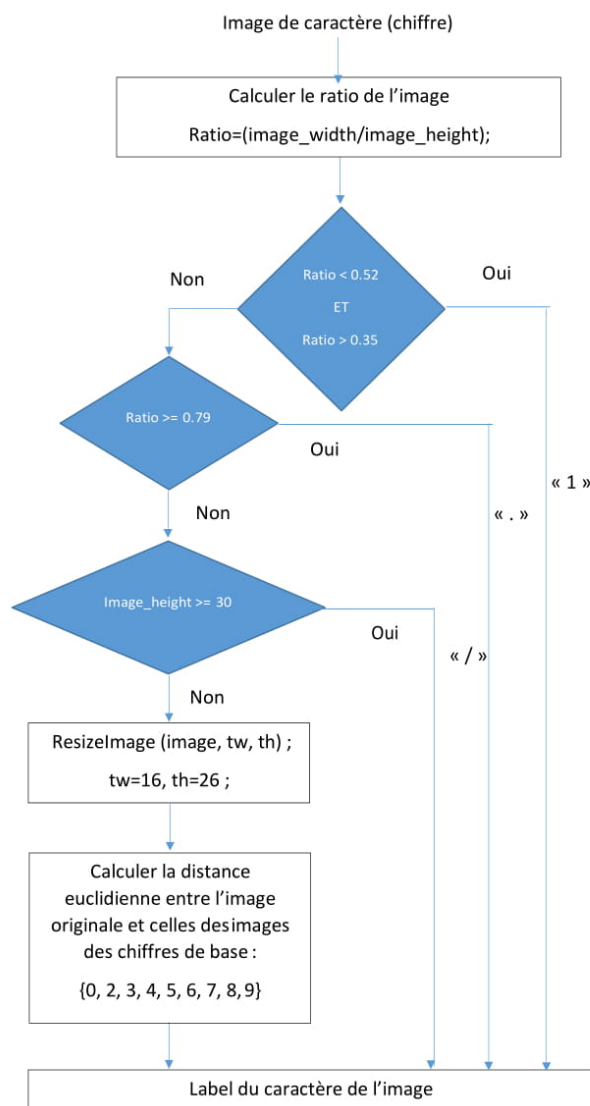


Figure 3.4: Schéma de la fonction de classification des segments des caractères

3.2.5 comparaison les DataFrames:

Dans cette étape, Nous convertissons le fichier originale (Fichier Excel) on DataFrame et , nous faisons une comparaison avec notre résultat, Les résultats de comparaison sont sous forme logique, cela montre nous les cellules correspondantes et les cellules non correspondantes

Conclusion

Dans ce chapitre, nous avons donné un aperçu sur notre travail, nous avons parlé en détail du programme et de toutes les étapes qu'il prend, de la lecture de l'image jusqu'à la comparaison des données extraites avec les données du fichier d'origine.

Dans ce travail, nous avons utilisé des algorithmes de traitement d'image et de détection de leurs contours, ainsi que des outils de détection de caractères pour extraire les caractères de chaque cellule du tableau. dans le chapitre suivantes nous allons parler de l'implémentation de notre systèmes, aussi nous allons présenter les différentes interfaces

Implémentation et résultats

Introduction

L'implémentation d'un logiciel devient après une série de plusieurs étapes dans le processus de développement, le but principal est de réaliser un système efficace de résoudre les problèmes posés en utilisant des outils et des algorithmes. Nous avons donné dans le précédent chapitre la conception dans façon globale et détaillée, le présent chapitre donne une vue sur le système et les outils utilisées afin de créer un système fiable. Dans ce chapitre nous allons présenter l'environnement du développement avec les différentes bibliothèques importantes utilisées, ainsi que les structures de données choisies pour implémenter ce type de système. Nous terminons ce chapitre par une conclusion.

4.1 Le choix de langage de programmation:

Pour choisir un langage de programmation qui se spécialise dans l'apprentissage profond, et le traitement des images, il doit considérer les compétences répertoriées dans les offres d'emploi actuelles ainsi que les bibliothèques disponibles dans différentes langues qui peuvent faire les processus d'apprentissage profond. Python est le langage de programmation le plus affecté dans le domaine de l'apprentissage automatique et l'apprentissage profond. Python est suivi par Java, puis par le R, puis C ++.

4.1.1 Python

Python est une interprétation , de haut niveau , à usage général langage de programmation . Créée par **Guido van Rossum** et publiée pour la première fois en 1991, la philosophie de conception de Python met l'accent sur la lisibilité du code avec son utilisation notable d'espaces blancs importants. Ses constructions de langage et son approche orientée objet visent à aider les programmeurs à écrire du code clair et logique pour des projets à petite et à grande échelle ,il fournit des constructions qui permettent une programmation claire à petite et à grande échelle.

Python dispose d'un système de type dynamique et d'une gestion automatique de la mémoire. Il prend en charge de multiples paradigmes de programmation, y compris



orientés objet, impératifs, fonctionnels et procéduraux, et dispose d'une bibliothèque standard vaste et complète. Les interpréteurs de Python sont disponibles pour de nombreux systèmes d'exploitation. le lien de site officiel de python <https://www.python.org/>

4.1.2 L'environnement de développement :

Pour l'environnement de développement, nous avons utilisé l'environnement du PyCharm(version Community 2020.1.3). utilisé pour la programmation en Python. Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégré, une intégration avec des systèmes de contrôle de version (VCS) et prend en charge le développement Web avec Django. Il permet l'analyse de code et contient un débogueur graphique. Développé par l'entreprise tchèque JetBrains, c'est un logiciel multi-plateforme qui fonctionne sous Windows, Mac OS X et Linux (<https://www.jetbrains.com/fr-fr/pycharm/>).



Il est décliné en édition professionnelle, diffusé sous licence propriétaire, et en édition communautaire diffusé sous licence Apache.

4.1.3 L'environnement Anaconda:

Anaconda regroupe un ensemble d'outils gravitant autour des langages de programmation Python et R : il fournit notamment les deux environnements d'exécution. Cette distribution de Python est orientée « Data Science » (science des données) et « Machine Learning » (l'apprentissage automatique) : dans ces domaines, elle est certainement la plus populaire.



Anaconda s'installe aussi bien sur Windows, MacOS ou Linux. Vous pouvez trouver le logiciel Anaconda à partir de son site officiel <https://www.anaconda.com/>. Anaconda propose son propre gestionnaire de paquets appelé « conda ».

4.2 Bibliothèques utilisées:

4.2.1 OpenCV (Open Source Computer Vision Library):

Est une bibliothèque offrant un ensemble de plus de 2500 algorithmes de vision par ordinateur spécialisé dans le traitement d'images, accessible au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (libre) pour les plateformes Windows, GNU/Linux, Android et MacOS, nous avons utilisé cette bibliothèque pour la détection des images introduites.



4.2.2 Pandas:

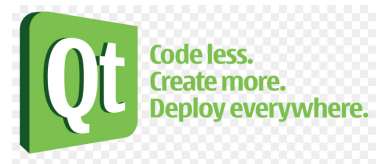
Pandas est une bibliothèque open source sous licence BSD fournissant des structures de données hautes performances et faciles à utiliser, ainsi que des outils d'analyse des données pour le langage de programmation Python . Pandas est un projet sponsorisé par NumFOCUS . Cela participera au succès du développement de pandas en tant que projet open source de classe mondiale et permettra de faire un don au projet. Pandas est une librairie python qui permet de manipuler facilement des données à analyser :



- Manipuler des tableaux de données avec des étiquettes de variables (colonnes) et d'individus (lignes).
- Ces tableaux sont appelés DataFrames, similaires aux DataFrames sous R.
- On peut facilement lire et écrire ces DataFrames à partir ou vers un fichier tabulé.
- On peut tracer des graphes à partir de ces DataFrames grâce à Matplotlib

4.2.3 QT:

Qt est une bibliothèque logicielle offrant essentiellement des composants d'interface graphique (communément appelés widgets), mais également d'autres composants non-graphiques permettant entre autres l'accès aux données, les connexions réseaux, la gestion des files d'exécution, etc. Elle a été développée en C++ par la société Trolltech et est disponible pour de multiples environnements Unix utilisant X11 (dont Linux), Windows et Mac OS. Qt est un toolkit qui présente de nombreux avantages. Il est intéressant de les souligner puisque ces avantages se retrouvent dans PyQt.



4.2.3.1 QT designer

Est un module qui permet de lier le langage Python avec la bibliothèque Qt. Il permet ainsi de créer des interfaces graphiques en python. Une extension de QtDesigner (utilitaire graphique de création d'interfaces Qt) permet de gérer le code python d'interfaces graphiques. PyQt dispose de tous les avantages liés à Qt.

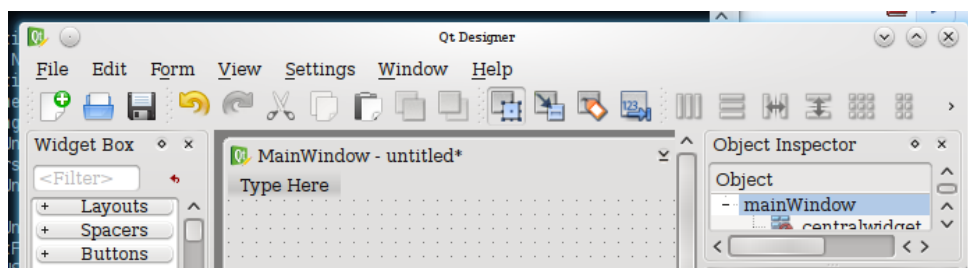


Figure 4.1: Interface Qt Designer

4.2.4 Tesseract-Ocr:

Le projet traitant de la reconnaissance de documents et l'extraction de données d'intérêt, il est important d'avoir une idée précise des solutions d'OCR existantes sur le marché. Il existe plusieurs logiciels OCR comme (FineReader (Abbyy), Omnipage (Nuance Communications), Tesseract, OCRopus) Tesseract est un logiciel de reconnaissance optique de caractères sous licence Apache. Conçu par les ingénieurs de Hewlett Packard de 1985 à 1995, son développement est abandonné pendant les dix années suivantes, en 2005, les sources du logiciel sont libérées sous licence Apache et le logiciel est actuellement développé par Google. Initialement limité aux caractères ASCII, Ce moteur supporte plus de 100 langages (de base) et propose un système d'apprentissage permettant de supporter plus.[3]



4.2.5 Présentation de l'interface d'application:

ci dessous l'interface principale de notre système de comparaison liste des notes: comme nous déjà dites dans le chapitre de conception, nous avons besoin tout d'abord de lire une image qui contient table de liste des notes.

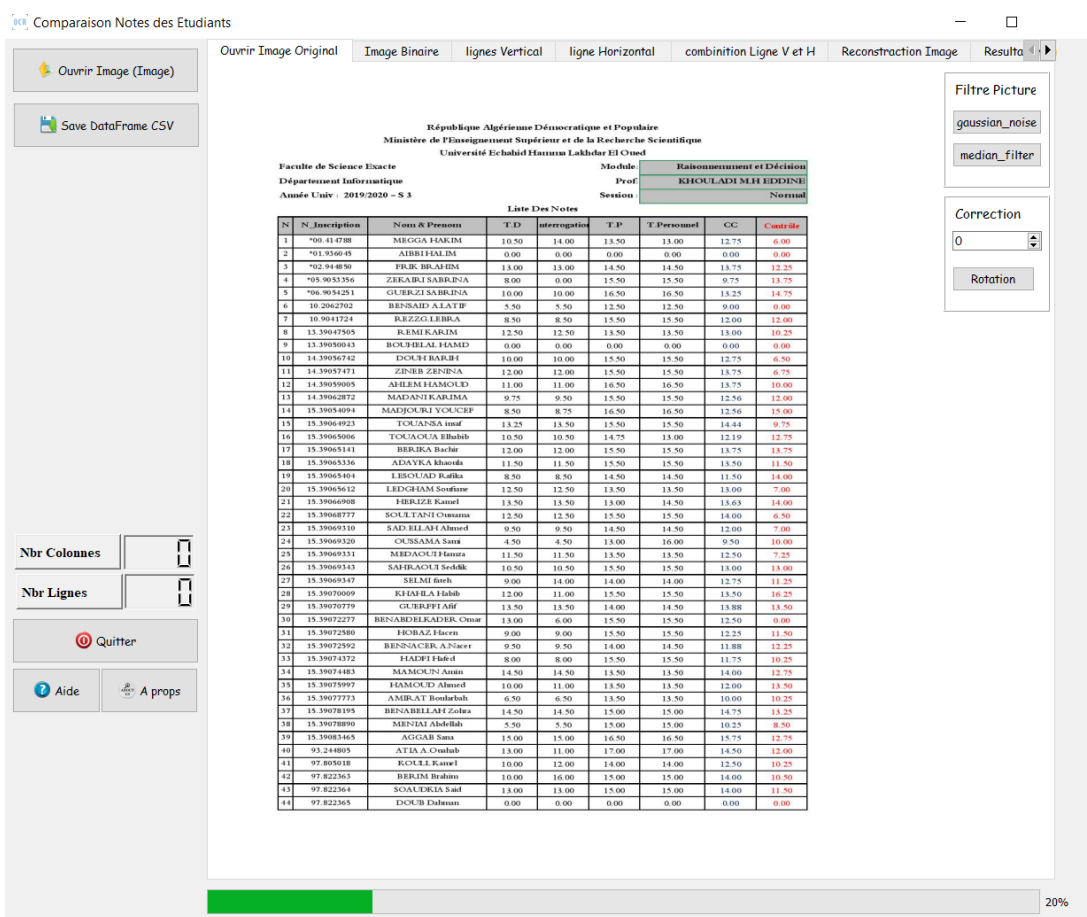


Figure 4.2: Interface principale de notre application

Vous pouvez modifier et améliorer l'image en appliquant des filtres (Médian ou Laplacien), ainsi de corriger l'orientation de l'image (rotation des rectangle) la figure suivante représente l'étape de combinaison les ligne vertical et horizontal détectes après la mise

seuil de l'image sur une image binaire.

correction d'image de table comme nous avons parlé dans la troisième chapitre, les étapes d'acquisition et la pré-traitement sont les plus essentielles étapes pour obtenir une belle résultats.

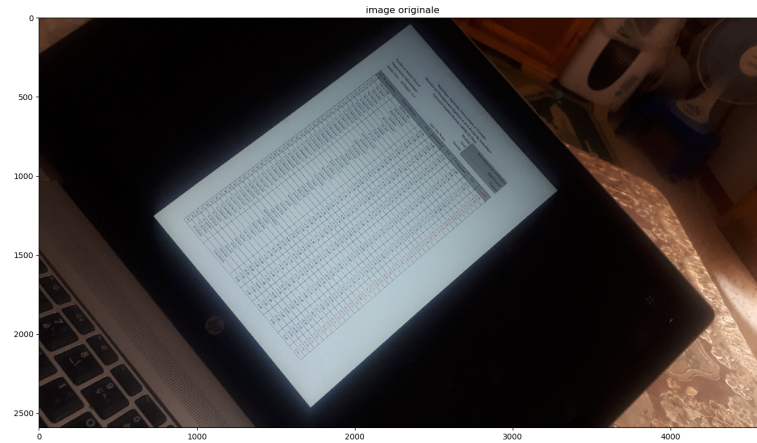


Figure 4.3: image inclinée

Afin de corriger l'image et d'isoler l'espace extérieur de l'espace réel de l'image de la table, nous prenons les mesures correctives et les résultats sont présentés dans les images suivantes.

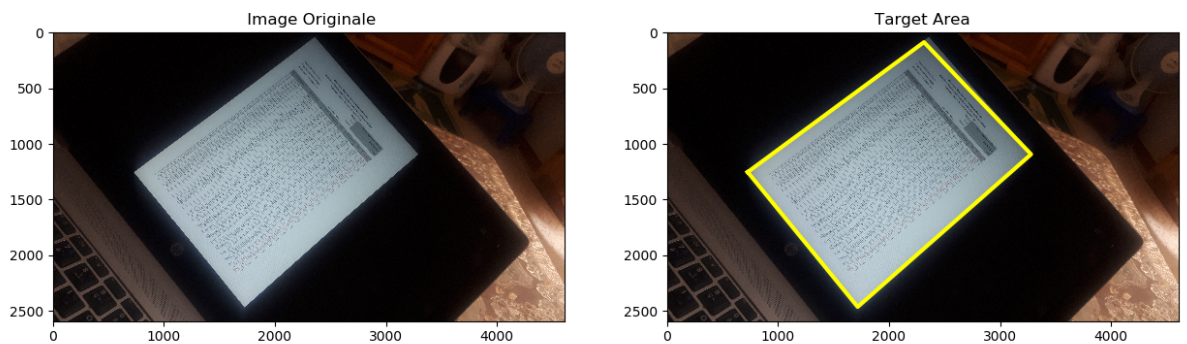


Figure 4.4: image originale vs image table après détection de contours

l'image ci après représente l'allocation approfondie de cellules, dernière étape avant l'extraction de table.

Comparison Notes des Etudiants

Ouvrir Image (Image) Save DataFrame CSV

Original Image Binaire lignes Vertical ligne Horizontale combinaison Ligne V et H Reconstruction Image Resultat Informations

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Echahid Hamou Lakhdar El Oued

Faculté de Sciences Exactes
Département Informatique
Année Univ : 2019-2020 - S 3

Module : **Reconnaissance et Décision**
Prat : **RECONNAISSANCE**
Séances : **Sciences**

Liste Des Notes

N°	N° Inscription	Nom & Prénom	T.P	Interrogat	T.P	V.Présentiel	CC	Classe
1	15 39501706	ABDELHAZI	10.00	14.00	13.50	13.00	12.75	13.00
2	15 39501703	ABDELHAZIM	0.00	0.00	0.00	0.00	0.00	0.00
3	15 39501682	ABDELKACIM	13.00	13.00	14.50	14.50	13.75	12.75
4	15 39501695	ABDEL	0.00	0.00	13.50	15.50	9.75	13.75
5	15 39501681	ABDELKADIR	10.00	10.00	10.50	10.50	13.25	14.75
6	15 39501692	ABDEL	12.50	8.50	13.50	12.50	11.00	13.25
7	15 39511124	ZINEB	12.50	8.50	13.50	15.50	13.00	13.25
8	15 39501695	ABDEL	12.50	12.50	13.50	13.50	13.00	10.25
9	15 39501694	ABDEL	0.00	0.00	0.00	0.00	0.00	0.00
10	15 39501692	ABDELKADIR	10.00	10.00	13.50	15.50	12.75	8.50
11	15 39501691	ABDEL	12.00	12.00	13.50	13.50	13.75	8.75
12	15 39501695	ABDELKADIR	11.00	11.00	10.50	10.50	12.75	10.00
13	15 39501692	ABDEL	9.75	9.50	13.50	15.50	12.50	17.00
14	15 39501695	ABDEL	8.50	8.50	13.50	15.50	14.00	8.50
15	15 39501692	ABDEL	13.25	13.50	13.50	15.50	14.50	9.75
16	15 39501695	ABDEL	10.50	10.50	14.75	13.00	12.75	12.75
17	15 39501691	ABDEL	13.00	12.00	13.50	15.50	13.75	13.75
18	15 39501695	ABDEL	11.50	11.50	13.50	15.50	13.00	11.50
19	15 39501695	ABDEL	8.50	8.50	14.50	14.50	11.50	14.00
20	15 39501692	ABDEL	12.50	12.50	13.50	15.50	13.00	9.00
21	15 39501695	ABDEL	13.50	13.50	13.00	14.50	13.00	13.00
22	15 39501695	ABDEL	13.50	12.50	13.50	14.50	14.00	8.50
23	15 39501691	ABDEL	6.50	9.50	14.50	14.50	12.00	9.00
24	15 39501692	ABDEL	4.50	4.50	13.00	16.00	9.50	10.00
25	15 39501691	ABDEL	11.50	11.50	13.50	15.50	12.00	12.50
26	15 39501691	ABDEL	10.50	10.50	13.50	15.50	13.00	13.00
27	15 39501691	ABDEL	9.50	14.00	14.00	14.00	12.50	13.25
28	15 39501691	ABDEL	12.50	11.50	13.50	15.50	13.25	9.25
29	15 39501691	ABDEL	13.50	13.50	14.00	14.50	13.50	13.50
30	15 39501691	ABDEL	13.00	6.00	9.50	15.50	12.00	0.00
31	15 39501691	ABDEL	9.00	9.00	13.50	15.50	12.25	11.50
32	15 39501692	ABDEL	9.50	9.50	14.00	14.50	11.00	12.25
33	15 39501692	ABDEL	9.00	9.00	10.50	15.50	11.75	10.25
34	15 39501691	ABDEL	14.50	14.50	13.50	15.50	14.00	12.75
35	15 39501691	ABDEL	10.00	11.00	13.50	15.50	12.00	13.50
36	15 39501691	ABDEL	6.25	6.25	13.50	15.50	10.00	10.25
37	15 39501691	ABDEL	14.50	14.50	13.00	15.00	14.75	13.25
38	15 39501691	ABDEL	8.50	8.50	13.00	15.00	10.25	8.50
39	15 39501691	ABDEL	15.00	15.00	16.50	16.50	15.75	14.75
40	15 39501691	ABDEL	13.00	11.00	17.00	17.00	14.50	13.00
41	15 39501691	ABDEL	10.00	12.00	14.00	14.00	12.50	10.25
42	15 39501691	ABDEL	10.00	16.00	13.00	15.00	14.00	10.75
43	15 39501691	ABDEL	13.00	13.00	13.00	14.00	14.00	13.00
44	15 39501691	ABDEL	0.00	0.00	0.00	0.00	0.00	0.00

Nbr Colonnes Nbr Lignes

Quitter Aide A props

70%

Figure 4.7: détection approfondie des cellules

nous enregistrons DataFrame obtenu par l'extraction des cellule a un fichier sous forme CSV,Ceci est pour faciliter le processus de comparaison.

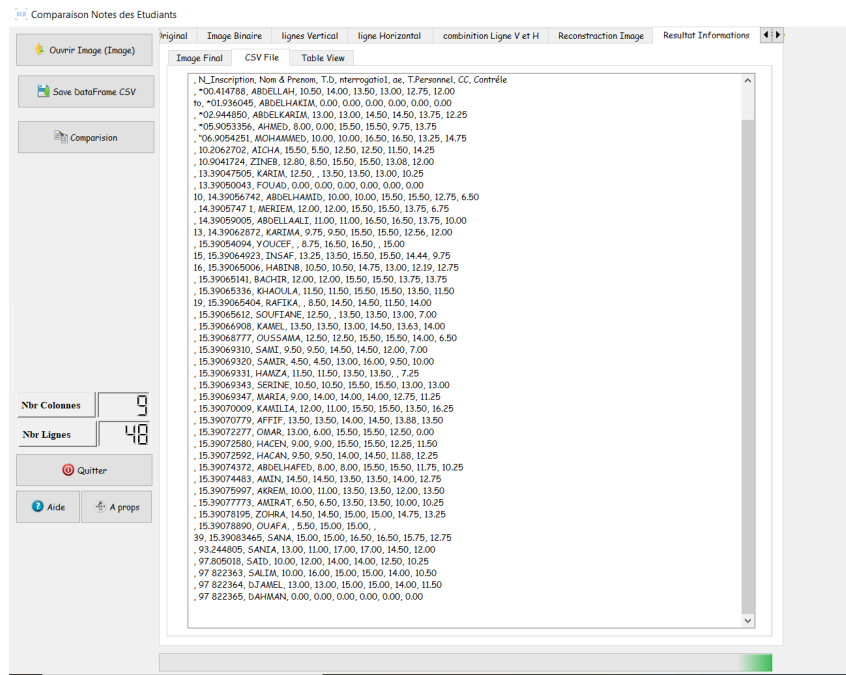


Figure 4.8: Extraction des cellules au fichier CSV

la dernière étape de notre application est la Comparaison du résultat obtenu avec le fichier ou la photo fourni par le service de l'éducation.

True : 97.73 % False : 2.27 % Resultat de comparaison de deux liste des notes (table)

Ninscription	NomPrenom	TD	INT	TP	TRVPER	CC	Contrôle
'00.414788	ABDELLAH	10.50	14.00	13.50	13.00	12.75	12.00
'01.936045	ABDELHAKIM	0.00	0.00	0.00	0.00	0.00	0.00
'02.944850	ABDELKARIM	13.00	13.00	14.50	14.50	13.75	12.25
'05.9053356	AHMED	8.00	0.00	15.50	15.50	9.75	13.75
'06.9054251	MOHAMMED	10.00	10.00	16.50	16.50	13.25	14.75
10.2062702	AICHA	15.50	5.50	12.50	12.50	11.50	14.25
10.9041724	ZINEB	12.80	8.50	15.50	15.50	13.75	12.00
13.39047505	KARIM	12.50	12.50	13.50	13.50	13.00	10.25
13.39050043	FOUAD	0.00	0.00	0.00	0.00	0.00	0.00
14.39056742	ABDELHAMID	10.00	10.00	15.50	15.50	12.75	6.50
14.39057471	MERIEEM	12.00	12.00	15.50	15.50	13.75	6.75
14.39059005	ABDELLAALI	11.00	11.00	16.50	16.50	13.75	10.00
14.39062872	KARIMA	9.75	9.50	15.50	15.50	12.56	12.00
15.3904094	YOUCEF	8.50	8.75	16.50	16.50	12.56	15.00
15.39064923	INSAAF	13.25	13.50	15.50	15.50	14.44	9.75
15.39065006	HABINB	10.50	10.50	14.75	13.00	12.19	12.75
15.39065141	BACHIR	12.00	12.00	15.50	15.50	13.75	13.75

Figure 4.9: Résultat de comparaison de deux liste des notes (table sous forme image)

Conclusion

Les modèles actuels OCR reposent sur un bon traitement des données et sur le choix de l'algorithme approprié, mais les spécialistes ne sont pas encore parvenus à définir des paramètres statiques ni un modèle général permettant de résoudre tous les problèmes similaires,. Dans ce chapitre , nous avons expliqué le critère de choix de langage de programmation et l'environnement du développement avec les différentes bibliothèques importantes utilisées, ainsi que les structures de données choisies pour implémenter ce type de système. enfin la présentation de l'interface d'application.

Conclusion et perspectives

Jusqu'aujourd'hui, La détection et l'extraction des tableaux présente un défi très grand, malgré tous les efforts et les travaux intensifs réalisés dans ce domaine, aucun de ces systèmes n'est jugé fiable à 100%, Mais au fur et à mesure d'autres travaux essayent d'améliorer les résultats. Dans ce travail, Nous avons proposé un système de comparaison de liste des notes des étudiants. Il s'articule essentiellement autour de deux parties. Une partie détection du tableau qui contient les notes à partir d'une image du document, et une partie d'extraction de contenu le contenu du tableau. Le taux de reconnaissance obtenu par l'algorithme traditionnel Open CV que nous avons utilisé dans ce travail et de 97.70% qui est très satisfaisant, mais comme dans tout logiciel, les améliorations restent toujours à apporter. L'optimisation des résultats peut être proposée par le pré traitement de l'image et l'utilisation d'une bonne qualité de papiers et pour les filtres utiliser pour minimiser les bruits. Nous concluons que cet algorithme est très fiable lorsqu'on travaille sur des tableaux avec une structure fixe

Nous pouvons envisager plusieurs perspectives possibles pour étendre ce travail :

- L'évaluation de ce algorithme sur d'autres ainsi que sur d'autres types des problèmes
- Les méthodes d'extraction des textes a partir les tableaux peuvent être envisagées (par exemple en utilisant les algorithmes qui ne contiennent pas la bibliothèque Tesseract-Ocr).
- Ajout des autres filtres sur l' image pour minimiser les bruits
- L'utilisation les bonnes qualités de papiers et une bonne acquisition d'image pour donne les bonnes résultats



Bibliographie

- [1] “Camelot: Pdf table extraction for humans” – <https://camelot-py.readthedocs.io/en/master/>, Date visite site: 03/09/2020.
- [2] “Pypdf2 1.26.0” – <https://pypi.org/project/PyPDF2/>, Date Visite site: 03/09/2020.
- [3] “Reconnaissance d’éléments textuels ou graphiques dans les documents juridiques numérisés”.
- [4] “Tabula” – <https://tabula.technology/>, Date Visite site: 03/09/2020.
- [5] “Unité linéaire rectifiée” – https://datafranca.org/wiki/Unit%C3%A9_lin%C3%A9aire_rectif%C3%A9e, Accessed: 03/09/2020.
- [6] “Vgg-19 convolutional neural network” – <https://www.mathworks.com/help/deeplearning/ref/vgg19.html>, Accessed: 03/09/2020.
- [7] “Dictionnaire la rousse” – <https://www.larousse.fr>, sep 2020, date accès :02/09/2020.
- [8] “Table segmentation, extract tables or forms” – <http://www.visionbib.com/bibliography/char969.html>, sep 2020, date accès :02/09/2020.
- [9] A. G. P. P. S. H. ADITYA KEKARE, ABHISHEK JACHAK – “Techniques for detecting and extracting tabular data from pdfs and scanned documents: A surve”.
- [10] M. U. AMROUCH – “Reconnaissance de caractères imprimés et manuscrits, textes et documents basée sur les modèles de markov cachés”, Thèse, Université Ibn Zohr, AGADIR.
- [11] D. BOUKHLOUF – “Résolution de problèmes par écosystèmes: Application au traitement d’images”, Thèse, Université Mohamed Khider-Biskra, 2005.
- [12] P. BOURRET, J. REGGIA et M. SAMUELIDES – “Réseaux neuronaux (une approche connexionniste de l’intelligence artificielle)”, (1991).
- [13] J. O. DU TERRAIL – “Réseaux de neurones convolutionnels profonds pour la détection de petits véhicules en imagerie aérienne”, Thèse, Normandie Université, 2018.
- [14] J. GASTELLU-ETCHEGORRY – *Acquisition et traitement d’image numérique*, Université Paul Sabatier. IUT - Département de Mesures Physiques, Avril 2008.
- [15] — , *Acquisition et traitement d’image numérique*, Université Paul Sabatier. IUT - Département de Mesures Physiques, Avril 2008.
- [16] R. GIRSHICK – “Fast r-cnn”, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [17] — , “Fast r-cnn”, *Proceedings of the IEEE international conference on computer vision*, 2015, p. 1440–1448.
- [18] B. M. HABIB – “Intitulé développement de méthodes d’extraction des contours sur des images à niveaux de gris”, Thèse, Université des Sciences et de la Technologie d’Oran Mohamed-Boudiaf, 2016/2017.
- [19] A. M. . A. K – “Opencv-python tutorials documentation”.
- [20] A. M. . A. K – “Image thresholding”, https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html, 2013.
- [21] S. S. S. G. R. R. N. K. M. R. K G SHREYAS DIXIT, MAHIMA GIRISH CHADAGA – “Evaluation and evolution of object detection techniques yolo and r-cnn”.

- [22] V. KURAMA – “Table detection, information extraction and structuring using deep learning”, <https://nanonets.com/blog/tag/table-detection/>, Date Visite site: 04/09/2020.
- [23] N. LE VINE, M. ZEIGENFUSE et M. ROWAN – “Extracting tables from documents using conditional generative adversarial networks and genetic algorithms”, *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, p. 1–8.
- [24] Y. LECUN – “L’apprentissage profond, une révolution en intelligence artificielle”, *La lettre du Collège de France* (2016), no. 41, p. 13.
- [25] Q. LIANG, J. PENG, Z. LI, D. XIE, W. SUN, Y. WANG et D. ZHANG – “Robust table recognition for printed document images”, *Mathematical Biosciences and Engineering* **17** (2020), no. 4, p. 3203.
- [26] A. MANZANERA – “Traitement d’images et vision artificielle”, *Unité d’Électronique et d’Informatique* (2005).
- [27] A. Y. NG et M. JORDAN – “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes”.
- [28] A. PACHAURI, V. MORE, P. GAIDHANI et N. GUPTA – “Autonomous ingress of a uav through a window using monocular vision”, *arXiv preprint arXiv:1607.07006* (2016).
- [29] S. S. PALIWAL, D. VISHWANATH, R. RAHUL, M. SHARMA et L. VIG – “Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images”, *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019, p. 128–133.
- [30] M. PRADHAN – “How to detect tables in images using opencv and python”, <https://medium.com/analytics-vidhya/how-to-detect-tables-in-images-using-opencv-and-python-6a0f15e560c3#:~:text=First%20we%20need%20to%20import,like%20OpenCV%2C%20numpy%20and%20matplotlib.&text=Then%20we%20will%20read%20the,on%20the%20untouched%20image%20im.>, Accessed: 03/09/2020.
- [31] S. R. QASIM, J. KIESELER, Y. IYAMA et M. PIERINI – “Learning representations of irregular particle-detector geometry with distance-weighted graph networks”, *The European Physical Journal C* **79** (2019), no. 7, p. 1–11.
- [32] S. R. QASIM, H. MAHMOOD et F. SHAFAIT – “Rethinking table recognition using graph neural networks”, *2019 International Conference on Document Analysis and Recognition (ICDAR)*, IEEE, 2019, p. 142–147.
- [33] D. B. R. ZANIBBI et J. CORDY – “A survey of table recognition: Models, observations, transformations, and inferences”.
- [34] S. REN, K. HE, R. GIRSHICK et J. SUN – “Faster r-cnn: Towards real-time object detection with region proposal networks”, *Advances in neural information processing systems*, 2015, p. 91–99.
- [35] M. ROGER – “Méthodes de monte-carlo”.
- [36] S. SCHREIBER, S. AGNE, I. WOLF, A. DENGEL et S. AHMED – “Deepdesrt: Deep learning for detection and structure recognition of tables in document images”, *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, vol. 1, IEEE, 2017, p. 1162–1167.
- [37] Y. SCULO – *Introduction au traitement d’images détection de contours et segmentation*, Université de Technologie de Troyes, 2009.
- [38] A. SINHA, J. BAYER et S. S. BUKHARI – “Table localization and field value extraction in piping and instrumentation diagram images”, *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 1, IEEE, 2019, p. 26–31.
- [39] S. S. TAUUVY ALEXANDRE, CARAYON NICOLAS – “Egalisation d’histogramme”, <http://tsi.telecom-paristech.fr>, august 2020.
- [40] OPENCV DEV TEAM. – “Structural analysis and shape descriptors”, https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html#approxpolydp, 2014.
- [41] C. TIAN, Y. XU et W. ZUO – “Image denoising using deep cnn with batch renormalization”, *Neural Networks* **121** (2020), p. 461–473.

- [42] M. D. ZEILER et R. FERGUS – “Visualizing and understanding convolutional networks”, *European conference on computer vision*, Springer, 2014, p. 818–833.
- [43] M. ZHANG, Z. CUI, M. NEUMANN et Y. CHEN – “An end-to-end deep learning architecture for graph classification”, *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.