

UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ
INSTITUT FRANCOPHONE INTERNATIONAL



Option : Systèmes Intelligents et Multimédia (SIM)

Promotion : XXII

RECONNAISSANCE DES FORMES
Groupe 5

**DETECTION ET EXTRACTION DE TEXTE DANS UN
DOCUMENT NUMÉRIQUE**

Hugues MADIMBA KANDA,
Jean Claude SERUTI ZAGABE
MASTER II

Encadrant :

Dr Ho Tuong Vinh
ho.tuong.vinh@ifi.edu.vn

Année académique 2018-2019

Table des matières

1 Définition et Analyse du Sujet	2
1.1 Contexte et problématique	2
1.2 Objectifs	3
1.3 Domaine de recherche	3
1.4 Outils et Techniques	3
1.5 Les Problèmes à résoudre	3
1.6 Difficultés rencontrées	3
2 Etat de l'art	4
2.1 Historique et Concepts	4
2.2 Travaux de recherches relatifs au sujet	4
2.3 Développement du contexte	6
2.3.1 L'analyse de documents	6
2.3.2 Détection des lignes de texte	7
2.3.3 Détection d'objets	8
2.4 Proposition d'un méta-modèle pour la détection d'objets textuels	10
2.4.1 Prise en compte du contexte	10
2.4.2 Méta-modèle	11
3 Solution Proposée	12
3.1 Reconnaissance optique de caractères	12
3.2 Architecture de l'étape de traitement	12
3.3 Implémentation	14
4 Expérimentation	15
4.1 Lancement du programme	15
4.2 Interprétation de résultat	15
5 Conclusion et Perspective	18

1 Définition et Analyse du Sujet

1.1 Contexte et problématique

Ce travail se place dans un contexte industriel, les entreprises de dématérialisation souhaitent déterminer si des documents semi-structurés tels qu'une carte d'identité, un ticket de train, une facture de téléphone, etc. sont présents sur une page numérisée. Ce que nous appelons "image" qui est une page numérisée.

Si un document est présent dans une image à analyser, il doit être localisé précisément afin de pouvoir exploiter les informations qu'il contient, telles que le nom, le prénom, etc. Notre problématique se rapproche donc de la recherche de sous-image.

Les contraintes industrielles imposent de privilégier la précision au rappel : en effet il est préférable que les décisions prises par l'algorithme de localisation soient correctes le plus souvent possible pour ne pas avoir à contrôler la sortie du système. Ce travail présente une méthode adaptée à la résolution de documents semi-structurés. Un document est semi-structuré si l'ensemble des documents de la même "famille" ont une partie de leur structure en commun. Par exemple les cartes d'identité vietnamiennes sont des documents "semi-structurés" dans la mesure où une partie de l'information ne change pas d'une carte à l'autre. Seules les informations relatives à la personne (nom, prénom, photo...) changent d'un exemplaire à un autre. Le processus de comparaison proposé dans ce travail repose sur la détection puis extraction de textes. La nature même des documents à comparer rend complexe cet objectif d'extraction de texte. Par exemple, les documents tels que les pièces d'identité contiennent une photo, des logos, des textures et du texte protégés des contrefaçons. Certaines informations (comme des hologrammes) disparaissent ou sont déformées lors de la numérisation. Des documents tels que les tickets de caisse ou récépissés sont des documents de mauvaise qualité (encre effacée, papier déformé...) et leur numérisation génère une image bruitée. L'étape de numérisation du document physique introduit également un degré de complexité supplémentaire à la recherche et la comparaison de documents. Les documents présents dans l'image peuvent être placés d'une manière quelconque (différentes orientations et positions).

Tout comme l'apprentissage en profondeur a touché presque toutes les facettes de la vision par ordinateur, il en va de même pour la reconnaissance des caractères et la reconnaissance de l'écriture.

Les modèles fondés sur l'apprentissage en profondeur ont réussi à obtenir une précision de reconnaissance de texte sans précédent, bien au-delà des méthodes traditionnelles d'extraction de caractéristiques et d'apprentissage automatique.

Ce n'était qu'une question de temps avant que Tesseract incorpore un modèle d'apprentissage approfondi pour renforcer encore la précision de la reconnaissance optique des caractères et en fait, le moment est venu. La dernière version de Tesseract (v4) prend en charge la reconnaissance optique de caractères basée sur l'apprentissage en profondeur, qui est nettement plus précise. Le moteur OCR sous-jacent utilise lui-même un réseau de mémoire à court terme (LSTM), une sorte de réseau de neurones récurrents (RNN). Nous en parlerons dans les lignes qui suivent.

1.2 Objectifs

Il peut également arriver que l'image du document ne soit pas à l'échelle standard et que certaines parties soient coupées ou recouvertes par d'autres informations. De plus, il peut y avoir des documents de différents types sur une même image. L'objectif poursuivi est de donner à l'ordinateur la capacité de reconnaître les caractères sur un document numérique à l'aide d'un programme en vue d'extraire le texte pour le sauvegarder en format txt.

1.3 Domaine de recherche

Ce travail se situe dans un domaine d'apprentissage automatique(ou profond) et de Traitement automatique de langue.

1.4 Outils et Techniques

Pour mettre en oeuvre notre cas d'étude, il faudra bien utiliser des algorithmes avec l'outil approprié pour former notre modèle pour ce cas d'étude.

Hormis Python, il en existe d'autre outil permettant de résoudre ce problème de manière automatique. Dans ce travail nous nous sommes basés en python, avec l'aide de OpenCV OCR et Tesseract afin d'implémenter les algorithmes. Ce projet implique la mise en œuvre de techniques de traitement d'images et d'algorithmes d'apprentissage automatique, d'où l'utilisation d'algorithme de classification supervisé et de réseau de mémoire à court terme (LSTM) , une sorte de réseau de neurones récurrents (RNN).

1.5 Les Problèmes à résoudre

De nos jours, il existe une énorme demande pour stocker les informations disponibles dans les documents papier dans un ordinateur, un disque de stockage, puis pour les réutiliser ultérieurement en procédant à une recherche. Un moyen simple de stocker les informations de ces documents papier dans le système informatique consiste tout d'abord à numériser les documents, puis à les stocker sous forme d'images. Mais pour réutiliser ces informations, il est très difficile de lire le contenu individuel et de rechercher le contenu de ces documents ligne par ligne et mot par mot. Cela pose un inconvénient car l'image n'est ni interrogeable ni modifiable. Même lorsque nous souhaitons convertir des images numérisées directement en PDF, celles-ci ne sont ni au format éditable ni au format interrogeable.

Dans ce travail nous allons tenter de résoudre le problème lié à l'identification et la reconnaissance de texte à partir d'une image et de le convertir en un format modifiable (.txt, etc.) sans avoir à saisir à nouveau manuellement le document texte. Le problème principal peut être expliqué par la différence entre l'information présente dans un document et celle donnée par une séquence textuelle ainsi que les méthodes de stockage de chaque type.

1.6 Difficultés rencontrer

- La première difficulté est l'implémentation des algorithmes ;

- La deuxième difficulté est la méthodologie à appliquer pour la résolution des images.
- La troisième difficulté c'est la prise en main du nouveau domaine d'apprentissage approfondi.

2 Etat de l'art

2.1 Historique et Concepts

La méthode d'extraction de texte à partir d'images s'appelle également Reconnaissance Optique de Caractères (OCR) ou parfois simplement reconnaissance de texte.

Tesseract a été développé en tant que logiciel propriétaire par Hewlett Packard Labs. En 2005, HP a ouvert le code source libre en collaboration avec l'Université du Nevada à Las Vegas. Depuis 2006, il a été activement développé par Google et de nombreux contributeurs open source.

Tesseract a acquis de la maturité avec la version 3.x en prenant en charge de nombreux formats d'image et en ajoutant progressivement un grand nombre de scripts (langages). Tesseract 3.x est basé sur des algorithmes de vision par ordinateur traditionnels. Au cours des dernières années, les méthodes basées sur l'apprentissage en profondeur ont dépassé de loin les techniques traditionnelles d'apprentissage automatique en termes de précision dans de nombreux domaines de la vision par ordinateur. La reconnaissance de l'écriture manuscrite est l'un des exemples les plus marquants. Donc, ce n'était qu'une question de temps avant que Tesseract ait également un moteur de reconnaissance basé sur Deep Learning. Dans la version 4, Tesseract a mis en place un moteur de reconnaissance basé sur la mémoire à court terme (LSTM).

Remarque : pour reconnaître une image contenant un seul caractère, nous utilisons généralement un réseau de neurones convolutionnels (CNN). Le texte de longueur arbitraire est une séquence de caractères. De tels problèmes sont résolus à l'aide de RNN et le LSTM est une forme populaire de RNN. La version 4 de Tesseract contient également l'ancien moteur OCR de Tesseract 3, mais le moteur LSTM est la valeur par défaut et nous l'utilisons exclusivement dans ce travail.

La bibliothèque Tesseract est livrée avec un outil de ligne de commande très pratique appelé tesseract . Nous pouvons utiliser cet outil pour effectuer une OCR sur des images et la sortie est stockée dans un fichier texte.

2.2 Travaux de recherches relatifs au sujet

De nombreux chercheurs ont effectué leurs travaux sur l'extraction du texte de l'image et récupération du information bien qu'il y ait beaucoup de défis. Ces recherches sont basées sur différentes images de texte techniques de détection et d'extraction qui ont leur propres avantages ainsi que des limitations. Examen de ces littératures sont données d'une manière séquentielle selon les lignes qui suivent.

Chowdhury Md. Mizan, Tridib Chakraborty et Suparna Karmakar[3], algorithme proposé pour reconnaître la copie imprimée et convertir en requis mettre en forme le texte en

utilisant l'OCR (caractère optique Reconnaissance) et techniques de traitement d'image. L'algorithme reconnaît le personnage hors ligne, est efficace pour extraire des images bimodales et est applicable lors de la récupération d'images, de vidéos, de textes de pages Web, etc. suggéré que les futurs chercheurs doivent faire sur Zone OCR.

Akhilesh A. Panchal, Shrugal Varde, M.S.Panse[4], a utilisé une combinaison de deux approches, Composant connecté et basé sur la région à fournir accès de la technologie de vision par ordinateur pour visuellement de personnes handicapées en extrayant et en convertissant l'image texte dans la parole avec une précision et une rapidité approuvées. La combinaison des résultats des techniques d'approche est plus rapide et meilleur système. Le système n'est pas vérifié avec image complexe et texte de petite taille et varié alignement. Les auteurs suggèrent une combinaison de techniques d'amélioration de la précision et de la vitesse.

Najwa-Maria Chidiac, Pascal Damien, Charles Yaacoub[5], utilisait MSER (au maximum Régions extrêmes stables) et largeur de trait DéTECTeurs pour détecter et extraire le texte de naturel scène indépendamment de l'orientation avec l'amélioration Précision sur image floue et bruyante. Mais le système proposé n'a pas pu détecter une image comportant du texte avec petite taille ou mince largeur et effet d'ombre.

Jack Greenhalgh et Majid Mirmehdi[6] créé un nouveau système de détection et de reconnaissance de texte dans les panneaux de signalisation automatiquement à l'aide de MSER (Régions extrêmes extrêmement stables) et HSV (Teinte-Saturation-Valeur). Le résultat s'est amélioré exactitude de reconnaissance F-mesure de 87%. L'image sur le panneau doit être capturée lorsqu'elle est plus grande. **Rashedul Islam, Md. Rafiqul Islam, Kamrul Hasan Talukder**[7], a proposé des techniques hybrides (Basé sur Edge et sur composant connecté) qui permet d'augmenter la précision de la détection de zone de texte et techniques d'extraction par combinaison. Dans ce la précision de l'algorithme du système d'extraction est amélioré (87,25%). Ils testent en utilisant seulement 08 images pour évaluer, mais pas considéré images dégradées et texte de petite taille, non vérifié par OCR à reconnaître personnages. Ces travaux futurs consistent à créer une base de données pour intérêt de la formation.

Arvind, Mohamed Rafi[8] a utilisé de connection de méthode des composants pour maximiser la détection et extraction du texte de l'image et catégorisation. Ils ont essayé d'améliorer les performances en précision (65,06%) et le taux de rappel (89,25%), et a présenté les résultats dans le graphique.

Vaishnav Ganesh, Dr L. G. Malik[9], a analysé Big Data par Google Apis et proposé la cadre pour Big Data Image utilisée Couleur basée méthode de partition et méthode de regroupement des lignes de texte utilisant le détecteur de bord de Canny et la transformation de Hough méthodes respectivement. En appliquant des classificateurs formés le temps ou l'efficacité seront améliorés.

Harpreet Singh, Deepinder Singh[10], a utilisé morphologie mathématique pour extraction d'image texte utilisant les performances améliorées et faibles bruit. Mais il ne pouvait pas détecter les petits textes de complexes Contexte. Le document indique que les travaux futurs seront d'extraire un petit texte et convertir en texte éditable forme.

Partha Sarathi Giri[11] a comparé deux bases Approches pour extraire une région de texte

dans des images : à base de composants connectés et à base périphérique utilisant un ensemble d'images qui varient selon les dimensions de éclairage, échelle et orientation. Ce travaux futurs proposés pour concevoir la région de texte d'extraction en vérifiant SVM et HMM, puis pour concevoir le logiciel de reconnaissance système pour les régions de texte d'extraction.

Niti Syal, Naresh Kumar Garg[12], le journal est basé sur l'intégration de Daubechies DWT, Différence de gradient et SVM, extrait du texte résultant région efficacement. Les travaux futurs proposés sont la mise en œuvre du système OCR pour reconnaître le texte, utilisez la meilleure méthode pour la suppression de texte.

2.3 Développement du contexte

Les approches classiques de reconnaissance de documents (Chen, Blostein, 2007) reposent davantage sur des techniques de classification supervisées utilisant des caractéristiques basées a priori sur le texte, la mise en page ou encore la présence d'illustrations ou de photos.

L'approche en trois étapes « prétraitement + segmentation + classification » est classique mais n'a jamais été testée sur des bases complexes telles que la notre. La littérature regorge d'articles sur chacune des trois étapes (séparation), pour les rendre génériques et applicable à des bases plus ou moins complexes. Cependant, la combinaison des trois étapes a été proposé uniquement pour des cas particuliers (facture, chèque, etc.). Dans (Augereau et al., 2011), ont testé une méthode de classification de ce type. Les limites sont liées à l'apprentissage supervisé volumineux, à la sélection de caractéristiques en fonction des classes de documents à reconnaître et à la qualité de la segmentation. Dans un contexte industriel, il faut être en mesure de classer plusieurs dizaines de milliers de documents numérisés chaque jour. De ce fait, il paraît impensable de mettre en place une chaîne d'analyse ou de traitement nécessitant en permanence de devoir labelliser des données pour une phase d'apprentissage, de paramétrier des prétraitements ou des algorithmes de segmentation. La possibilité de combiner un système sans prétraitement (segmentation des documents composites, redressement, autres) avec une légère phase d'apprentissage (mode requête par l'exemple) nous a poussés à rechercher la faisabilité d'une approche par recherche de sous-images. Cette approche a été largement abordée dans le domaine des images naturelles grâce à l'utilisation de détecteurs (Tuytelaars, Mikolajczyk, 2008) et de descripteurs (Mikolajczyk, Schmid, 2005) de points d'intérêt.

2.3.1 L'analyse de documents

D'importants volumes d'informations sont stockés au format papier. Numérisation nécessaire pour pouvoir les traiter efficacement. L'analyse de documents vise à automatiser ce processus de digitalisation.

Une illustration ci-dessous présente la Chaîne de reconnaissance pleine page



FIGURE 1 – Extraction des zones de textes



FIGURE 2 – Détection des lignes de textes



FIGURE 3 – Reconnaissance du texte contenu dans les lignes.

2.3.2 Détection des lignes de texte

- Spécificités de la tâche : Les objets se touchent et se superposent ; il y a de nombreux petits objets par page.
- Contrainte liée à la taille réduite des jeux de données annotés.
- Défis pour traiter des flux de données fortement hétérogènes.

Cette phase de détection n'est rien d'autre équivalent à la partie d'analyse de document, selon deux auteurs (Shi et al, 2009) et (Nicolaou et al, 2009) ont présenté respectivement les approches de la phase de détection respectivement dont :

- **Approche ascendante** : grouper les composants
- **Approche descendante** : séparer les éléments

Il y a des bons résultats sur les tâches spécifiques et les difficultés pour généraliser sur des bases hétérogènes. la figure 4 illustre un exemple de base hétérogènes :

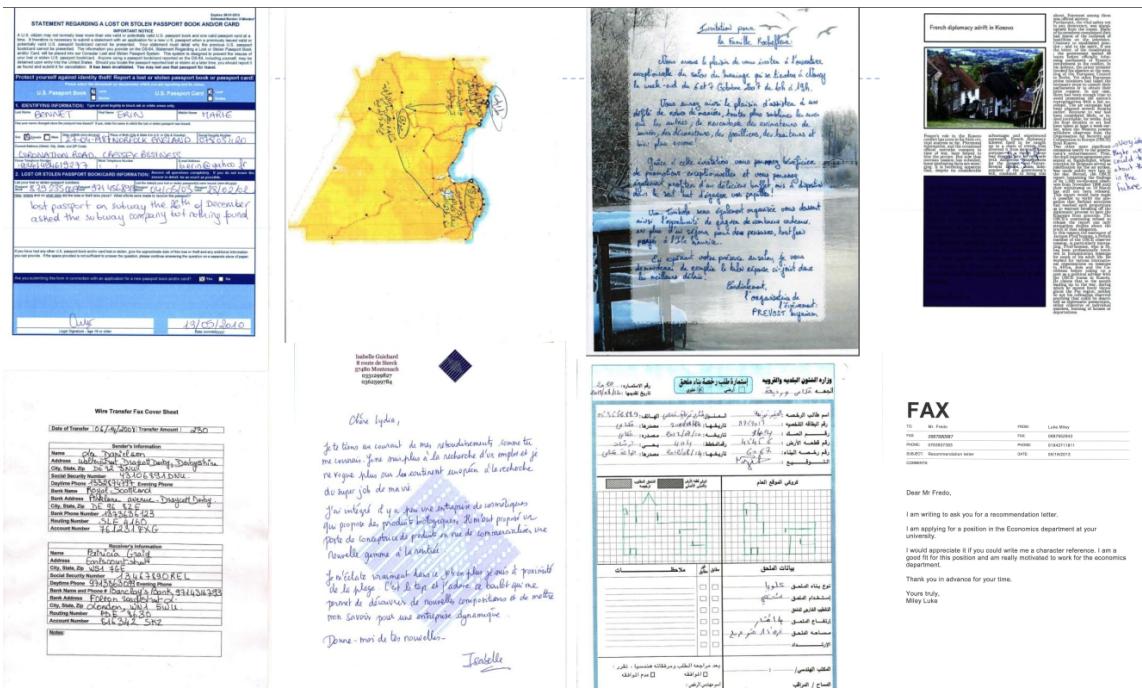


FIGURE 4 – Base Hétérogènes

2.3.3 Détection d'objets

Détection d'objets peut être compris selon ce schéma présenté sur la figure 5

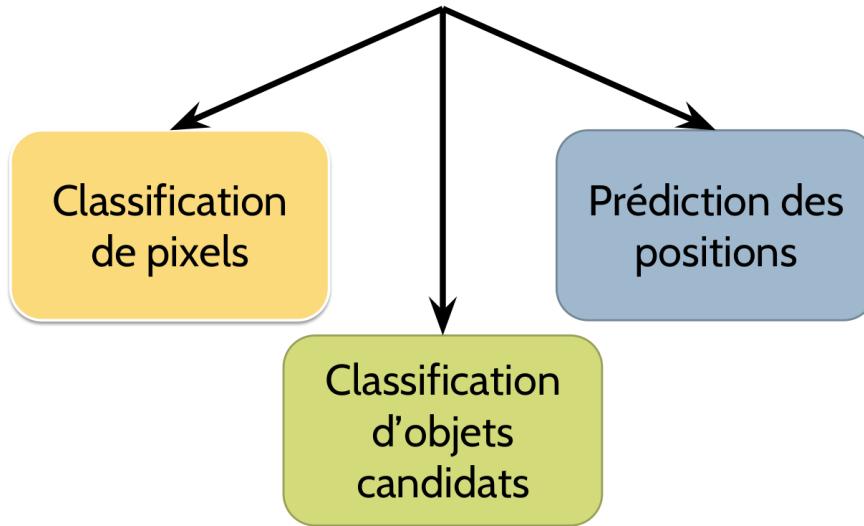


FIGURE 5 – Schéma de Détection d'objets

—Classification des pixels :

La figure 6 illustre une idée décrite dans (Fully Convolutional Networks for Semantic Segmentation, Long et al, 2016)

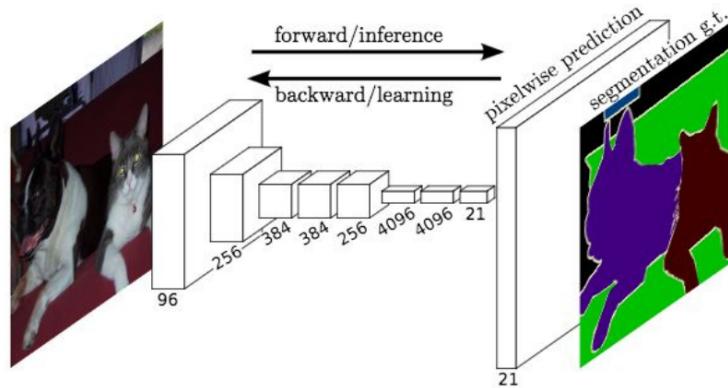
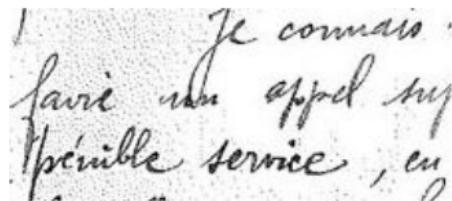


FIGURE 6 – Réseaux convolutionnels pour la segmentation sémantique

Problème : les caractères des différents objets se touchent et les boîtes se superposent. Avec comme exemple,



—Classification d'objets candidats :

Selon l'article de Rich feature hierarchies for accurate object detection and semantic segmentation, Girshick et al, 2014, les auteurs nous présentent d'une manière compréhensive que je résume comme présenté sur la figure 8

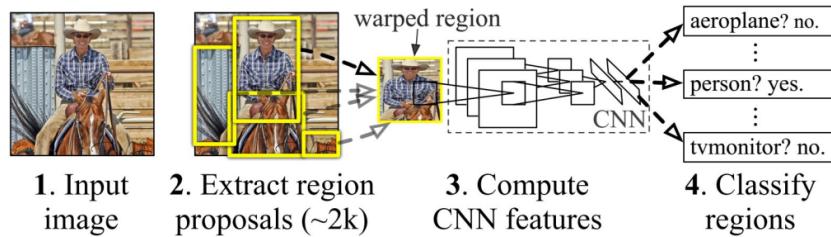
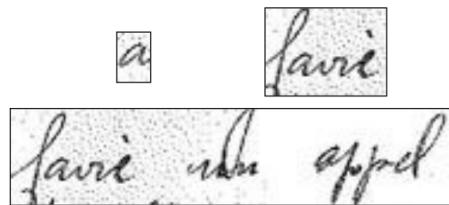


FIGURE 7 – R-CNN :régions dotées de fonctionnalités CNN

Avec comme exemple au texte :



Problème : une partie de ligne ressemble fortement à une ligne.

—Prédiction des positions des objets :

Sur cette phase on peut retenir ce qui suit :

- On prédit les coordonnées des objets et la confiance dans le fait que ces objets existent.
- On prédit un nombre variable d'objets.
- Pas de problème si les objets se touchent.

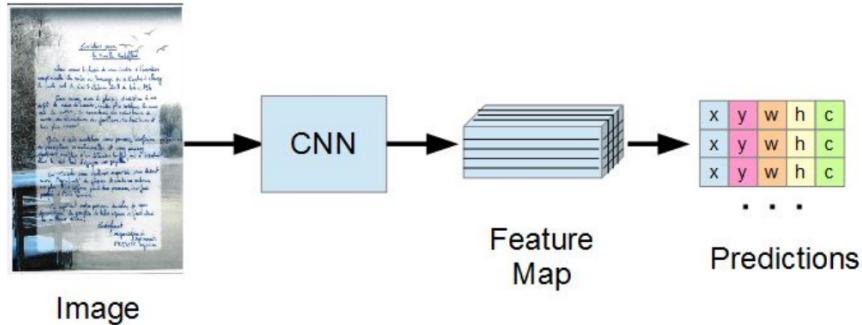


FIGURE 8 – Prédiction des positions

2.4 Proposition d'un méta-modèle pour la détection d'objets textuels

2.4.1 Prise en compte du contexte

Certains objets sortent des champs réceptifs. Impossible de prédire précisément les positions des objets. Perte des informations du contexte de la page. Notre contribution serait d'insérer de couches de récurrences 2D-LSTM entre les couches de convolution pour transmettre le contexte entre les positions. Une illustration de RNN 1D bidirectionnel selon Schuster and Paliwal (Bidirectional recurrent neural networks, 1997).

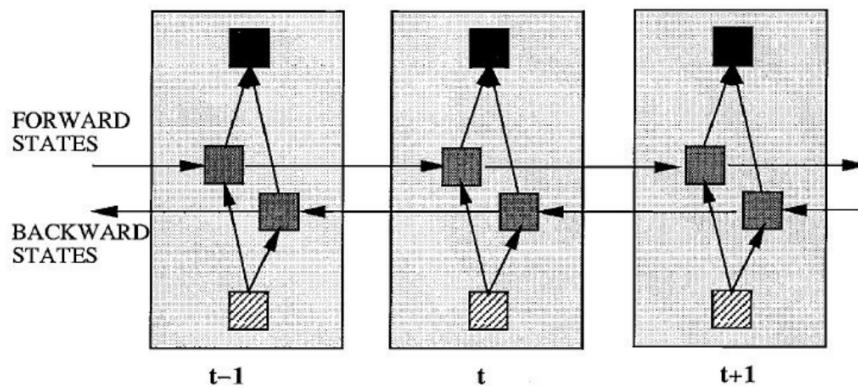


FIGURE 9 – RNN 1D

Cette figure 9 a une direction temporelle. On parcourt le signal dans les deux sens. En parallèle et on en combine les sorties. Par contre le RNN 2D à deux dimensions. On ajoute aux entrées les sorties précédentes horizontalement et verticalement. Avec le RNN 2D multi-directionnel, on parcourt l'image dans les quatre directions comme le montre la figure 10.

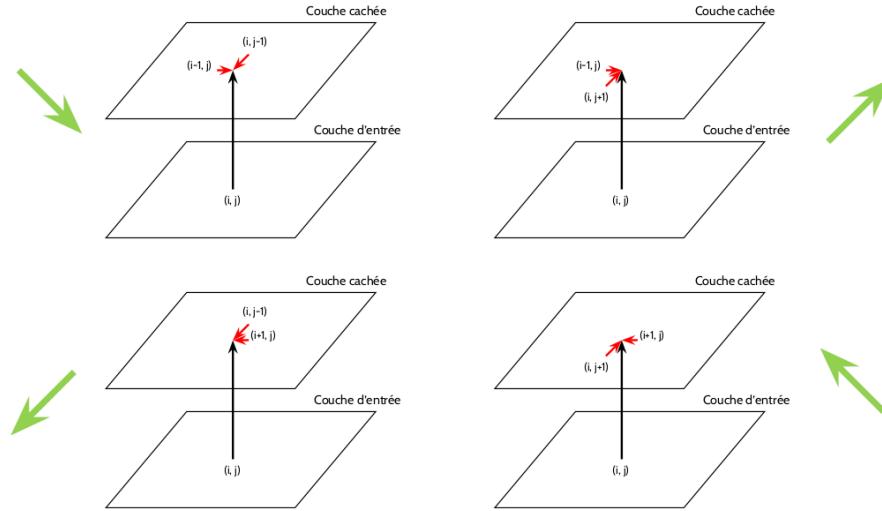


FIGURE 10 – RNN 2D multi-directionnel

Les réseaux de mémoire à long terme à long terme - généralement appelés simplement "LSTM" sont un type particulier de RNN, capable d'apprendre des dépendances à long terme. Ils ont été introduits par Hochreiter et Schmidhuber (1997) et ont été affinés et popularisés par de nombreuses personnes dans des travaux ultérieurs . Ils fonctionnent extrêmement bien sur une grande variété de problèmes et sont maintenant largement utilisés.

Les LSTM sont explicitement conçus pour éviter le problème de dépendance à long terme. Se souvenir des informations pendant de longues périodes est pratiquement leur comportement par défaut, ce n'est pas quelque chose qu'ils ont du mal à apprendre !

Tous les réseaux de neurones récurrents se présentent sous la forme d'une chaîne de modules répétitifs de réseau de neurones. Dans les RNN standard, ce module répétitif aura une structure très simple, telle qu'une seule couche de bronzage.

Les LSTM ont également cette chaîne comme structure, mais le module répétitif a une structure différente. Au lieu d'avoir une seule couche de réseau neuronal, il y en a quatre, qui interagissent de manière très particulière.

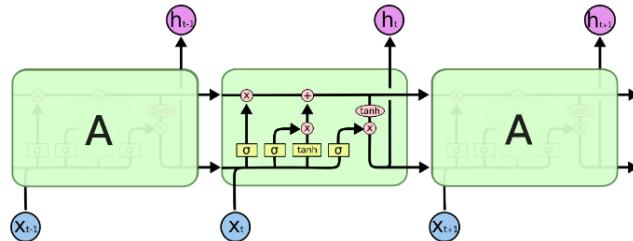


FIGURE 11 – Le module de répétition dans un LSTM contient quatre couches en interaction

2.4.2 Méta-modèle

- Prédiction des coordonnées des positions : pas de problème avec les recouvrements.
- Modèle local : nombreux objets détectés avec moins de paramètres. Modèle récurrent : pas

de perte des informations contextuelles.

- Terme de confiance et appariement global pour apprendre efficacement le nombre d'objets dans l'image.

3 Solution Proposée

3.1 Reconnaissance optique de caractères

Partant de la recherche bibliographique, nous nous proposons une solution à mettre en oeuvre pour répondre aux besoins de la société. La reconnaissance optique de caractères implique la détection du contenu textuel sur les images et la traduction des images en texte codé que l'ordinateur peut facilement comprendre. Une image contenant du texte est numérisée et analysée afin d'identifier les caractères qu'elle contient. Lors de l'identification, le caractère est converti en texte codé par machine.

Comment est-ce vraiment réalisé ? Pour nous, le texte sur une image est facilement discernable et nous pouvons détecter des caractères et lire le texte, mais pour un ordinateur, il s'agit d'une série de points.

L'image est d'abord numérisée et les éléments de texte et graphiques sont convertis en un bitmap, qui est essentiellement une matrice de points noirs et blancs. L'image est ensuite pré-traitée, où la luminosité et le contraste sont ajustés pour améliorer la précision du processus.

L'image est maintenant divisée en zones identifiant les zones d'intérêt telles que l'emplacement des images ou du texte, ce qui permet de lancer le processus d'extraction. Les zones contenant du texte peuvent désormais être subdivisées en lignes, mots et caractères et le logiciel est désormais en mesure de faire correspondre les caractères par comparaison et divers algorithmes de détection. Le résultat final est le texte de l'image que nous avons reçu.

Le processus peut ne pas être précis à 100% et peut nécessiter une intervention humaine pour corriger certains éléments qui n'ont pas été numérisés correctement. La correction d'erreur peut également être obtenue à l'aide d'un dictionnaire ou même du traitement de langage naturel (NLP). La sortie peut maintenant être convertie vers le support txt

3.2 Architecture de l'étape de traitement

Dans cette partie du traitement on a différentes étapes à effectuer dans l'image pour vérifier si elle contient du texte ou non, identifiez pour localiser la zone de texte sur l'image, et différencier le premier plan et l'arrière-plan de cette image et le texte de l'image.

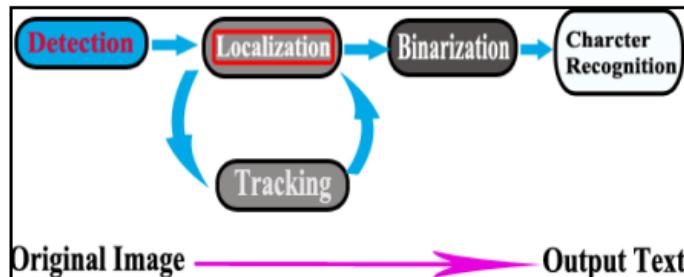


FIGURE 12 – Architecture de l'étape du traitement de l'extraction de texte

La figure 12 présente les étapes suivantes :

- **Détection de texte** : prend une image améliorée en entrée et décide qu'il contient ou non du texte et identifie le régions de texte dans une image.
- **Localisation de texte** : fusionne les régions de texte avec formuler les objets de texte et définir le serré limites autour des objets texte. Détection de texte, les modules de localisation et de suivi sont étroitement liés liés à chacun.
- **Suivi du texte** : est également utilisé pour accélérer le texte. processus d'extraction en n'appliquant pas le Binarisation et reconnaissance à chaque détection objet.
- **Binarisation du texte** : utilisé pour segmenter l'objet texte de l'arrière-plan dans les objets texte délimités. Il convertit l'image en niveaux de gris en image binaire, où les pixels de texte et les pixels d'arrière-plan apparaissent dans deux niveaux binaires différents, comme du texte blanc sur noir fond ou vice versa. La binarisation peut aussi être fait avant les autres étapes.
- **Reconnaissance des caractères** : la dernière étape est la reconnaissance des caractères. Ce module convertit l'objet texte binaire en texte ASCII à l'aide de l'outil OCR. .

La figure 13 ci-dessous montre la zone de texte localisée de l'image d'origine, en cours de traitement. La zone reconnue en tant que texte est délimitée par des rectangles. Cela se fait par étape de localisation, afin que sa sortie puisse être utilisée comme entrée pour la prochaine étape de segmentation d'objets texte et non-texte, ce qui simplifie le processus de reconnaissance des caractères.



FIGURE 13 – Localisation de zone de texte

3.3 Implémentation

Pour ce projet, nous utiliserons la bibliothèque Python-Tesseract , ou simplement PyTesseract , qui est un wrapper pour le moteur Google Tesseract-OCR. Nous choisissons cela parce qu'il est complètement open-source et en cours de développement et de maintenance. Au cours de notre développement il faut savoir que nous avons créer un modèle standard capable d'apprendre n'importe quelle base d'image en entré pour de fournir un résultat selon la qualité d'image comme expliquer précédemment.

—**Base Image** : cette fig 14 présente la base de nos images à exploiter en entrée.

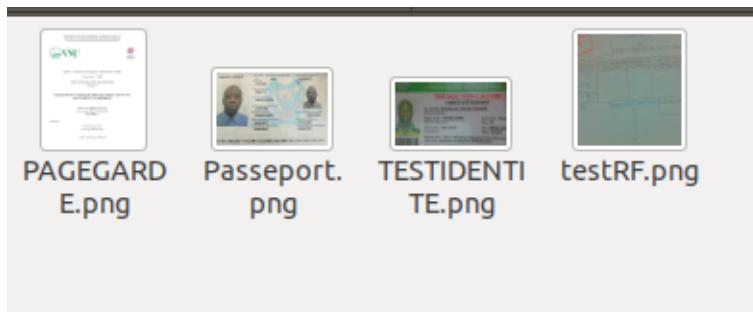


FIGURE 14 – Base d'image

—**Résultat** : ci dessous le résultat obtenu après processus de la détection et reconnaissance de la fig 14

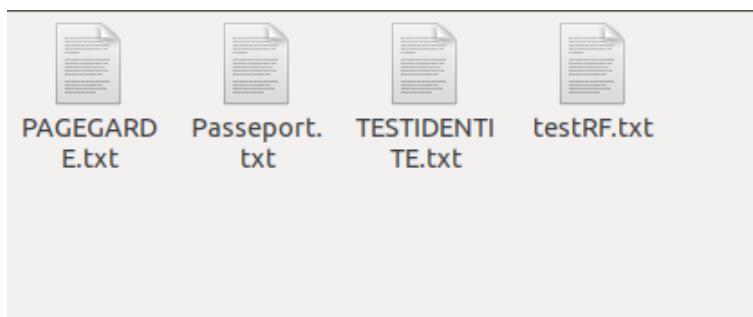
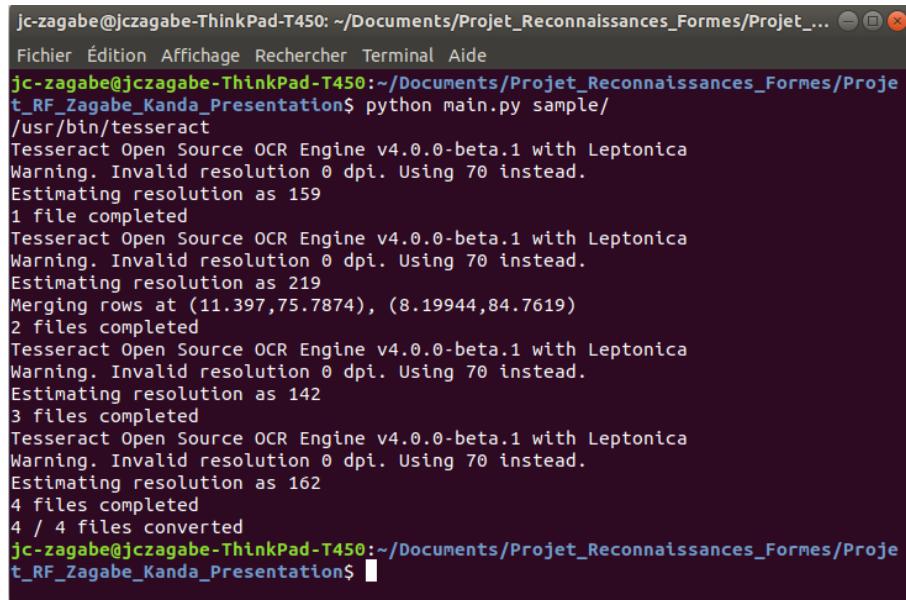


FIGURE 15 – Résultat après traitement

4 Expérimentation

4.1 Lancement du programme

—**Résultat :** Nous avions utilisé 4 images dans notre base pour tester notre modèle.



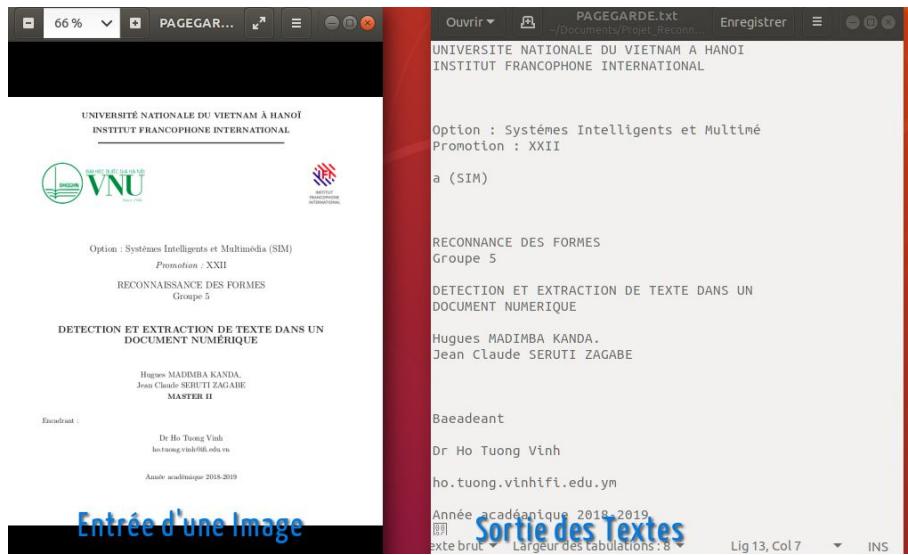
```
jc-zagabe@jczagabe-ThinkPad-T450: ~/Documents/Projet_Reconnaissances_Formes/Projet_... Fichier Édition Affichage Rechercher Terminal Aide jc-zagabe@jczagabe-ThinkPad-T450:~/Documents/Projet_Reconnaissances_Formes/Proje t_RF_Zagabe_Kanda_Presentation$ python main.py sample /usr/bin/tesseract Tesseract Open Source OCR Engine v4.0.0-beta.1 with Leptonica Warning. Invalid resolution 0 dpi. Using 70 instead. Estimating resolution as 159 1 file completed Tesseract Open Source OCR Engine v4.0.0-beta.1 with Leptonica Warning. Invalid resolution 0 dpi. Using 70 instead. Estimating resolution as 219 Merging rows at (11.397,75.7874), (8.19944,84.7619) 2 files completed Tesseract Open Source OCR Engine v4.0.0-beta.1 with Leptonica Warning. Invalid resolution 0 dpi. Using 70 instead. Estimating resolution as 142 3 files completed Tesseract Open Source OCR Engine v4.0.0-beta.1 with Leptonica Warning. Invalid resolution 0 dpi. Using 70 instead. Estimating resolution as 162 4 files completed 4 / 4 files converted jc-zagabe@jczagabe-ThinkPad-T450:~/Documents/Projet_Reconnaissances_Formes/Proje t_RF_Zagabe_Kanda_Presentation$
```

FIGURE 16 – Résultat après traitement

La figure 16 nous présente à la sortie le résultat obtenu et sauvegardé dans un dossier sous format .txt, donc sur les 4 images entrées dans le système, ce dernier les a tous traités avec une précision de résultat différent selon l'image d'entrée.

4.2 Interprétation de résultat

—**Première image :** Notre modèle récupère en entrée l'image dénommée PAGEDE-GARDE, voir la figure 14 et procède à toutes les procédures comme expliquées à la figure 12 de notre architecture d'étape du traitement de l'extraction de texte.



A la sortie de ce résultat, on constate que le programme à donner un résultat avec une forte précision, qui est significatif, il n'extrait que les caractères pertinents.

—Deuxième image : Notre modèle récupère en entrée l'image dénommée Passeport, voir la figure 14 et procède à toute les procédures comme expliqué à la figure 12 de notre architecture d'étape du traitement de l'extraction de texte.



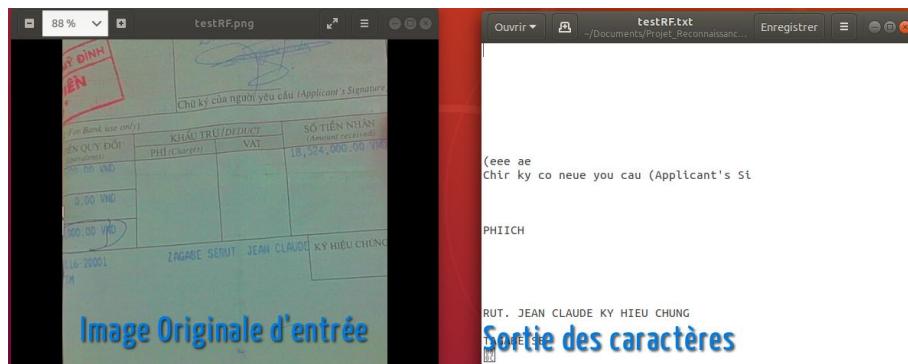
Le programme à pu détecter les textes afin de l'extraire tout en isolant les images de fond, en quelque sorte , on comprend clairement comment les techniques de traitement d'image joue derrière ce programme et l'importance de l'utilisation de Tesseract, de Même s'il y a une légère inclinaison dans le texte, Tesseract fait un travail raisonnable avec très peu d'erreurs. On arrive qu'à même à résoudre ces genres de problèmes de texte de longueur arbitraire à l'aide de RNN et LSTM.

—Troisième image : Cette image contient les informations d'un étudiant, avec notre programme nous allons l'exploiter et il se nomme TESTIDENTITE dans notre base.

Le résultat obtenu est presque parfait avec une forte précision car on remarque une similarité des caractères à la sortie.



—**Quatrième image :** Elle contient les informations de payement, c'est un reçu de preuve de perception d'argent, nommée dans notre base testRF.



Un exemple un peu difficile est un reçu qui présente une disposition de texte non uniforme et plusieurs polices. Vu la mauvaise numérisation, le résultat obtenu n'est pas bon car sa précision est faible. Cependant, si nous aidons un peu notre programme en scaannant avec un bon matériel, rognant la zone de texte, le résultat sera très bon.

Les autres résultat ci-dessus illustre la nécessité de la détection de texte avant la reconnaissance de texte. Un algorithme de détection de texte a généré un cadre de sélection autour des zones de texte, qui sont introduites dans un moteur de reconnaissance de texte tel que Tesseract pour une sortie de haute qualité.

5 Conclusion et Perspective

Partant de différente étude durant le développement de ce travail, nous avions pu mettre en place un système de reconnaissance des textes dans une image soit un document scanné. Nous avons présenté un réseau de neurones récurrents et le système OCR qui utilise avec la haute technologie de pointe tout en décrivant ses modules comportant les étapes de l'acquisition, le prétraitement, la segmentation, l'extraction des caractéristiques (texte), la classification et le post-traitement ainsi que les approches développées pour chaque module.

Donc toutes ces étapes ont été résumé par une architecture que nous avions proposé pour bien mener cette étude qui constitue la première phase de notre projet en reconnaissance de formes.

Les différents éléments de l'état actuel de notre système doivent être optimisés mais la structure générale est maintenant stable. La continuité de ce travail se situe dans la phase de reconnaissance. Nous avons déjà procédé à des essais sur les principaux outils. De cette étude, nous pouvons déduire les caractéristiques minimales que doivent valider les textes inclus dans l'image pour être reconnus. Si les caractéristiques sont trop contraignantes, il sera alors nécessaire de développer un outil spécifique de reconnaissance.

Le travail réalisé nous ouvre plusieurs perspectives. Nous allons implémenter le système conçu et munir des tests permettant de comparer différentes méthodes afin d'améliorer le taux de performance du système. Nous allons aussi créer le corpus permettant d'effectuer les tests en vue de savoir réellement si les informations extraites correspondent réellement à la personne.

Lien github : https://github.com/ZAGABE7S/Reconnaissance_de_Forme/

Références

- [1] Learning to detect, localize and recognize many text objects in document images from few examples, IJDAR 2018.
- [2] Full-page text recognition : Learning where to start and when to stop, B. Moysset, C.Kermorvant and C. Wolf, ICDAR 2017.
- [3]. Ch. Md Mizan, T. Chakraborty* and S. Karmakar, “Text Recognition using Image Processing”, International Journal of Advanced Research in Computer Science (IJARCS), 2017
- [4]. A. A. Panchal, Sh. Varde, M.S. Panse, “Character Detection and Recognition System for Visually Impaired People”, IEEE, International Conference on Recent Trends in Electronics Information Communication Technology, 2016, pp.1492-1496.
- [5]. Najwa-Maria Chidiac, P. Damien, Ch.Yaacoub, “A Robust Algorithm for Text Extraction from Images”, IEEE, 2016, pp.493-497.
- [6]. J. Greenhalgh and M. Mirmehdi, “Recognizing Text-Based Traffic Signs,” IEEE Transactions on Intelligent Transportation Systems, 2015, pp.1360- 1369.
- [8] Arvind, M. Rafi, «Extraction de texte à partir d’images à l’aide de la méthode du composant connecté», JoAIRA, Journal de la STM, 2014, 13-18.
- [9]. V. Ganesh, Dr. L. G. Malik, “Extraction of Text from Images of Big Data” International Journal of Advance Research in Computer Science and Management Studies, IJARCSMS, 2014, pp.40-46.
- [10]. H. Singh, D. Singh, “Text Confining and Extraction in Image Using Mathematical Morphology,” International Journal of Science and Research (IJSR), 2012, pp.288-290.
- [11]Learning text-line localization with shared and local regression neural networks, B.Moysset, J. Louradour, C. Kermorvant and C. Wolf, ICFHR 2016
- [12]. N. Syal, N. K. Garg, “Text Extraction in Images Using DWT, Gradient Method And SVM Classifier”, 2014, pp.477-481
- [13]. P. S. Giri, “Text Information Extraction and Analysis from Images Using Digital Image Processing Techniques,” International Journal on Advanced Computer Theory and Engineering (IJACTE), 2013, pp.66-71.
- [14] Chen N., Blostein D. (2007). A survey of document image classification : problem statement, classifier architecture and performance evaluation. International Journal on Document Analysis and Recognition, vol. 10, n o 1, p. 1–16.
- [15]Augereau O., Journet N., Domenger J.-P. (2011). Document images indexing with relevance feedback : an application to industrial context. In Document analysis and recognition (ic-dar), 2011 international conference on, p. 1190–1194.
- [16] Mikolajczyk K., Schmid C. (2005). A performance evaluation of local descriptors. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 27, n o 10, p. 1615–1630