Projet

Implémentation d'un système de Word Spotting en langage Python

Réaliser Par : Nasreddine MENACER

Résumé

Ce projet consiste à implémenter un système de word spotting en langage python, ce système contient une interface web pour interagir avec l'utilisateur, et un script de traitement en python qui est exécuté en arrière-plan, l'utilisateur est mené en premier temps à ouvrir dans cette interface l'image du document qui veut traiter, puis choisir (encadrer) à l'aide de l'outil de capture le mot recherché, un résultat est retourné après exécution sous forme d'image ou le mot recherché est encadré en rouge, et un texte donnant le nombre d'occurrence du mot dans le document.

Mots Clés: Word spotting, analyse de documents, approches word spotting

Abstract

This project consists in Implementing a word spotting system in python language, this system contains a web interface to interact with the user. , and a python processing script that runs in the background, the user is first led to open in this interface the image of the document that wants to process, then choose (frame) using the capture tool to search the word, a result is returned after execution as an image where the searched word is framed in red, and a text giving the number of occurrence of the word in the document

Key Words: Word spotting, document analysis, word spotting approaches

Table des matières

Tableau des Figures

INTRODUCTION

Actuellement les bibliothèques mondiales détiennent un grand nombre de documents historiques sous forme manuscrite. En numérisant ces documents, leur contenu peut être conservé et mis à la disposition d'une grande communauté via Internet.
Ces données peuvent aujourd'hui être partagées, mais elles sont souvent grands, non structurés, et seulement disponibles dans des formats d'image, ce qui rends leurs exploitation difficile.

La solution pour ce problème peut être d'annoter manuellement ces documents, ce qui est très coûteux en termes de temps et d'argent.
C'est pour cela qu'il existe des méthodes automatiques d'analyse de texte pour extraire de l'information à partir d'un document, tel que : les systèmes optiques de reconnaissance des caractères, qui ne sont pas assez robustes pour les écritures manuscrites.

On va s'intéressés dans ce projet à une autre technique qui est le Word Spotting qui est considéré comme une alternative à l'OCR traditionnel, le Word Spotting est une approche qui consiste à trouver toutes les images de mots similaires à un mot donné en requête sous forme d'image.

OBJECTIF DU PROJET ET L'ETAT DE L'ART

1. Objectif et Cahier de charge du projet

Le but de notre TER est de programmer en langage python un algorithme de word spotting capable de découper une image d'un document texte, en plusieurs images qui constituent les mots ou les chaine de caractères présents l'image et faire la correspondance entre une requête fournie en entrée sous forme d'image et les autres images de mots, l'algorithme doit être capable à la fin de nous fournir un résultat qui nous informe sur le nombre d'occurrence du mot recherché, à travers une interface graphique.

De ce fait notre projet peut se diviser en deux parties :

- Séparation des mots, extraction des caractéristiques sur les imagettes et comparaison.
- Affichage des résultats sur l'interface web.

Le diagramme en bêtes à cornes ci-dessous nous permet d'identifier les besoins de notre système.

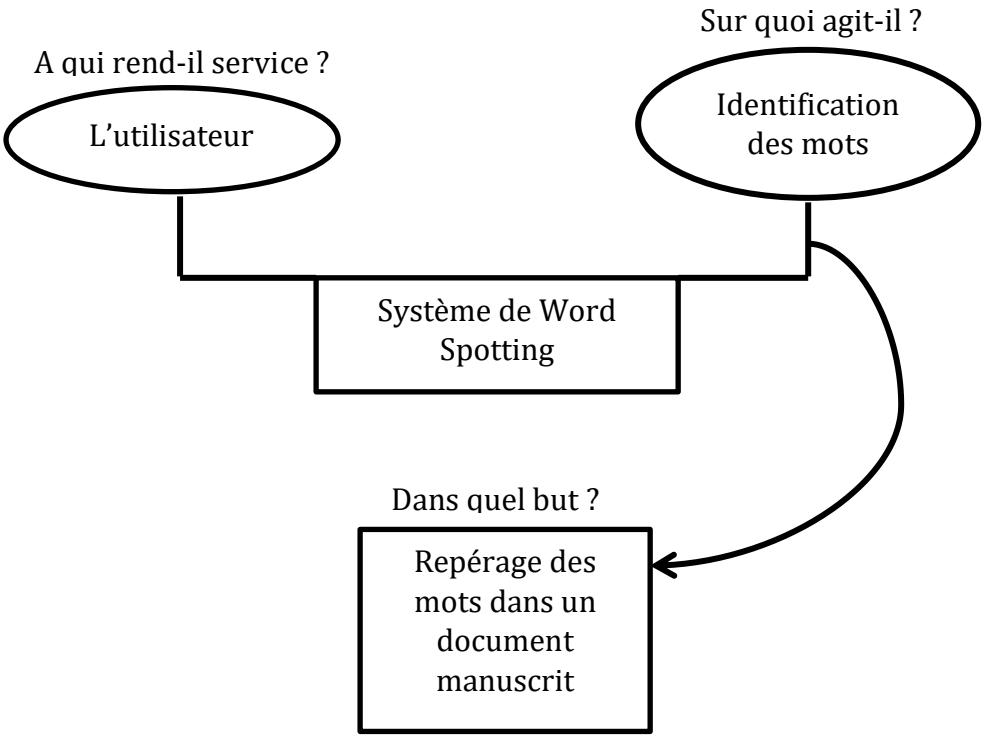


Figure 1: diagramme des besoins

2. Analyse de l'existant

Le word spotting est une technique qui a pour objectif de trouver dans une image de documents les occurrences d'une requête, qui est désignée par l'utilisateur et qui représente une image d'un mot se trouvant dans le document.

Dans le word spotting, on ne s'intéresse pas à trouver les lettres des mots, mais plutôt à faire des mesures de similarité entre images de mots. Cela permet donc de créer des indexes partiels pour le document manipulé. Et ainsi récupérer l'information suggérée comme une requête dans des documents historiques ou modernes quand ils sont relativement complexes et dégradés.

Le word spotting est très utile quand il s'agit de traité des documents en latins, arabes ou grecs, par exemple. Où le nombre de caractères, la direction de l'écriture sont différente.

- **Les différentes catégories du Word Spotting**

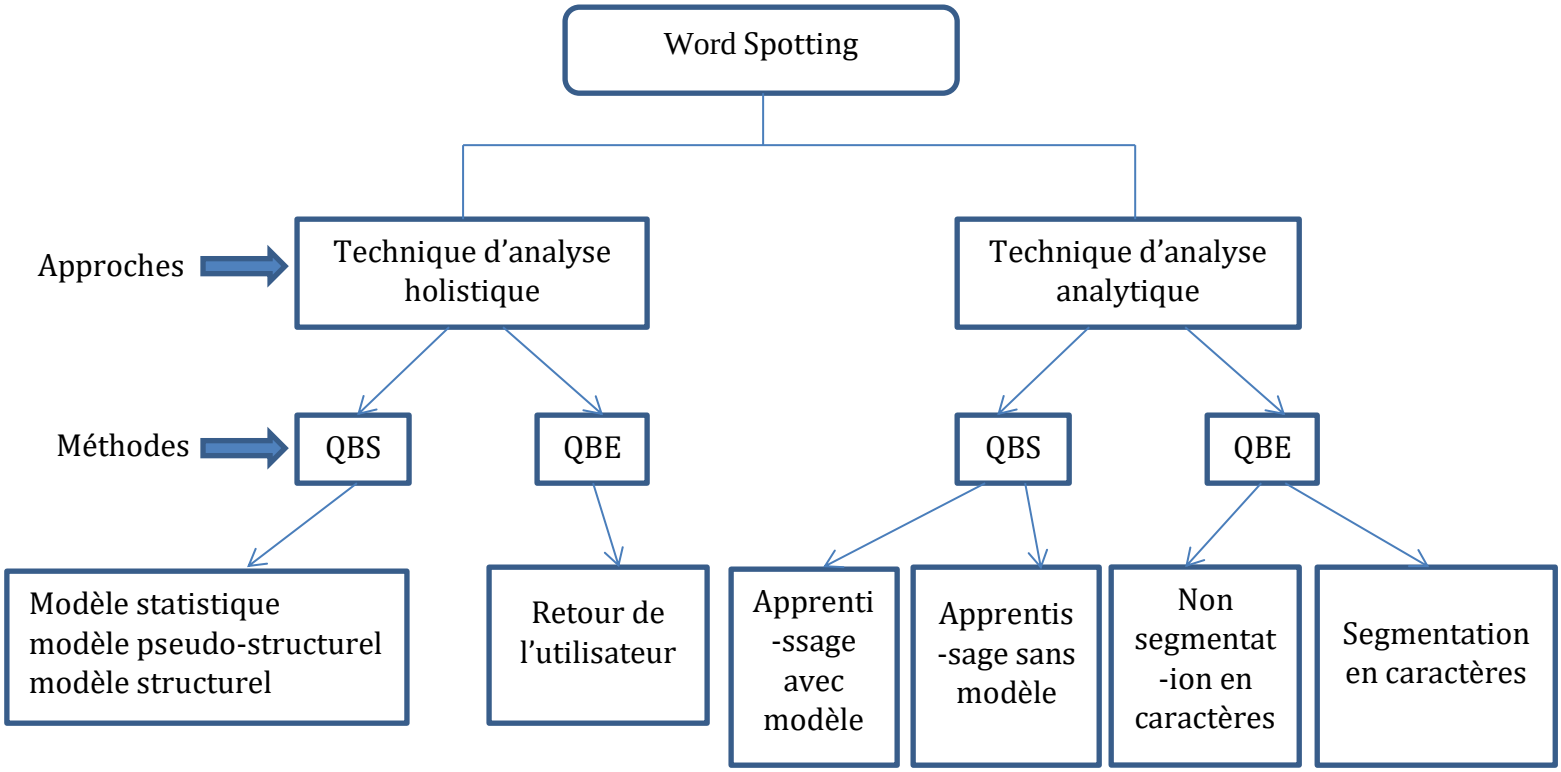


Figure 2 : Les catégories du word spotting

QBS (Query-by-String) : la requête est sous forme d'une chaine de caractères.

QBE (Query-by-Example) : la requête est sous forme d'une image.

- **Les techniques d'analyse holistique :**

C'est des techniques qui considèrent chaque image de mot comme une seule unité. Ces techniques s'appuient sur un processus de segmentation réalisé sur le document à traiter, la qualité de la segmentation influence le résultat obtenu.

Comme décrit dans la figure 2 les techniques d'analyse holistique sont divisées selon la méthode et la nature de la requête en : QBE / QBS.

Dans la sous-classe technique holistique utilisant des requêtes images (QBE) : chaque mot est représenté par un modèle statistique, modèle pseudo-structurel, un modèle structurel.
Les modèles statistiques représentent l'image comme un vecteur de caractéristiques à (n) dimensions. Ce vecteur est constitué soit de caractéristiques globales (calculées à partir de la totalité de l'image telles que la hauteur, la largeur) ou locales (calculés à partir de régions locales de l'image : des points d'intérêt, des croisements, nombre de trous, position de trous).
Les modèles pseudo-structurels quant à lui accumulent des informations (séquence de primitives géométriques et topologiques) pour représenter les images.

Dans la sous-classe technique holistique utilisant des requêtes images (QBS) : plusieurs propositions d'occurrence du mot recherché sont présentées à l'utilisateur après une phase automatique et c'est à l'utilisateur d'indique si la proposition lui convient, en fonction de ce retour, le système peut améliorer les résultats de spotting.

- **Les techniques d'analyse analytique :**

Dans les techniques d'analyse analytiques on trouve deux catégories : la première consiste à réalise une segmentation de l'image de mot pour donner un meilleur résultat de reconnaissance, et le deuxième qui consiste à segmenter l'image de document en unités plus petites.

Comme décrit dans la figure 2 les techniques d'analyse analytique sont divisées selon la méthode et la nature de la requête en : QBE / QBS.

Dans la sous-classe technique analytique utilisant des requêtes images (QBE) : on trouve deux autres sous-classes basées soit sur la segmentation en caractères, soit sur la non segmentation en caractères
Dans la sous-classe technique analytique utilisant des requêtes images (QBS) : on trouve deux autres sous-classes basées soit sur un apprentissage avec un modèle, soit basées sur un apprentissage sans modèle, tous les deux englobe les techniques de (matching DTW, , les modèles de HMM et les kNNs).

A partir de cette étude, nous avons appris que les approches analytiques sont plus robustes et donnent de meilleurs résultats que les approches holistiques.

Donc le choix que nous avons fait pour un premier travail est de suivre une approche Analytique donc de segmenté l'image du document, pour séparer les mots, et non les caractères, avec une méthode de recherche basé QBE (requete sous forme d'image capturé a partir du document à traiter).

PARTIE EXPERIMENTALE

Ce chapitre présente les différentes parties réalisées durant notre projet et qui vont servir d'outils pour notre système de word spotting. Nous présentons aussi l'expérimentation menée et les résultats obtenus.

1. Interface graphique

- L'interface est constituée d'un petit texte au début qui d'écrit brièvement le word spotting, une partie dédiée à l'ouverture du document à traiter, et l'autre pour visualiser le mot capturé pour le recherché.

Figure 3 : L'interface graphique

- En premier lieu l'utilisateur est mené à ouvrir l'image du document en cliquant sur le bouton choisir une image, et à l'afficher en cliquant sur le bouton afficher, en cliquant sur le bouton capturer , l'utilisateur capture le mot qui veut chercher dans le document, ce dernier est afficher à droite sur l'outil de capture, en cliquant sur le bouton trouver, en ayons en possession les deux images (requête et document) le scripte en python réalise le traitement (segmentation, extraction des caractéristiques, comparaison).

Figure 4 : Outil de Capture

- Après le traitement le scripte renvoie le résultat final, qui indique le nombre de fois ou le mot donné en requête apparait dans le document, tout en les encadrant en rouge.

Figure 5 : Résultat du traitement

2. Script Python

- **Découpage des mots**

Dans cette partie nous segmentons notre image originale en mots, pour cela notre script vérifie si l'image donnée est en niveau de gris, si elle est en RVB nous extrayions le premier plan de l'image couleur.

La segmentation de l'image est une étape importante pour la suite du programme, nous avons implémenté dans cette partie la méthode des projections verticales et horizontales.

Segmentation en lignes :

Pour segmenter l'image en lignes, nous utilisons le vecteur de la projection verticale de l'image.

La projection verticale d'une image est la somme des niveaux de gris de chaque colonne de notre image. Pour une image de taille [n, m] nous obtenons un vecteur de taille ligne de taille m.

Nous analysons ce vecteur afin d'extraire le début et la fin de chaque ligne.

Segmentation en mots :

Une fois qu'on a séparé les lignes, nous avons découper les mots, pour cela nous utilisons la projection horizontale, qui est calculée sur chaque ligne découpée précédemment.

La projection horizontale d'une image est la somme des niveaux de gris de chaque ligne d'une image. Pour une image de taille [n, m] nous obtenons un vecteur de taille colonne de taille n.

Ce vecteur est ensuite analysé pour déterminer le début et la fin de chaque mot.

Tous les mots segmentés sont sauvegardés dans une liste qui contient également les indices du mot dans l'image original.

Exemple :

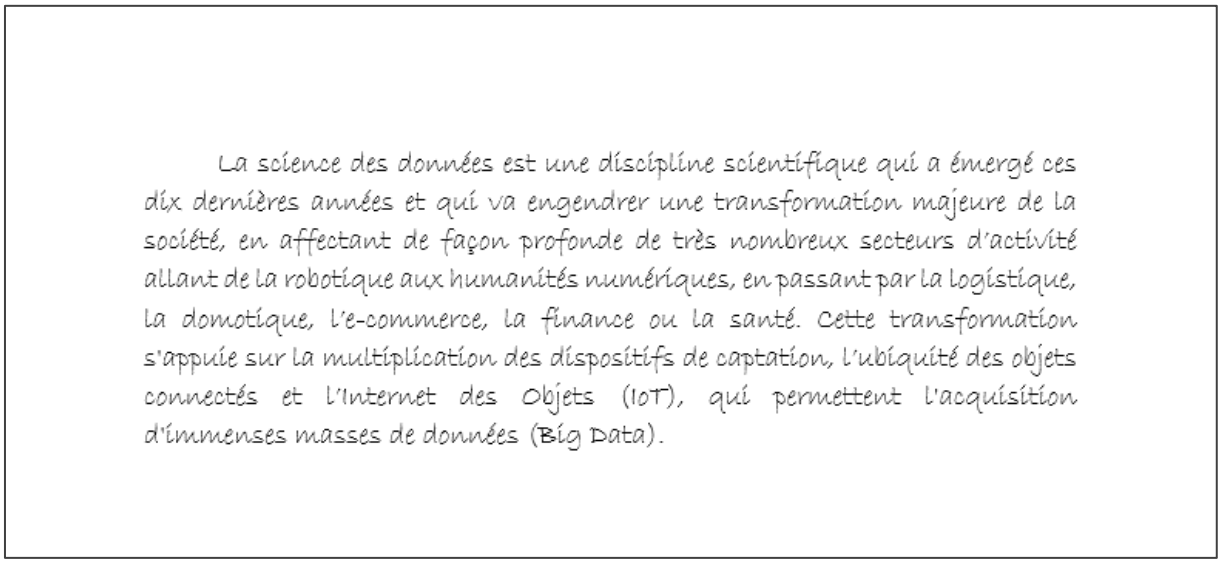


Figure 6 : Image de texte à découper

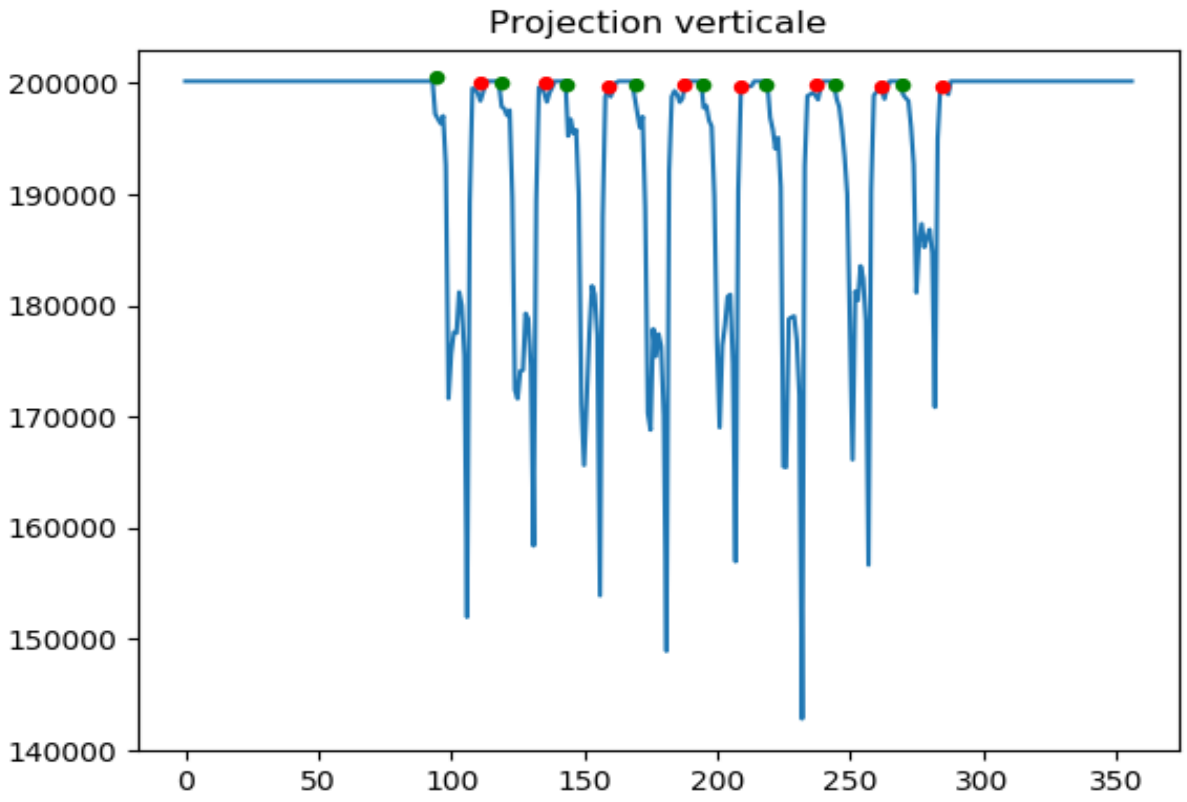


Figure 7: Projection vertical de l'image fournie

- Les points verts correspondent aux débuts des lignes.
- Les points rouges correspondent aux débuts des lignes.

On a 8 couples de points (verts, rouges) ce qui correspond aux 8 lignes de texte de l'image.

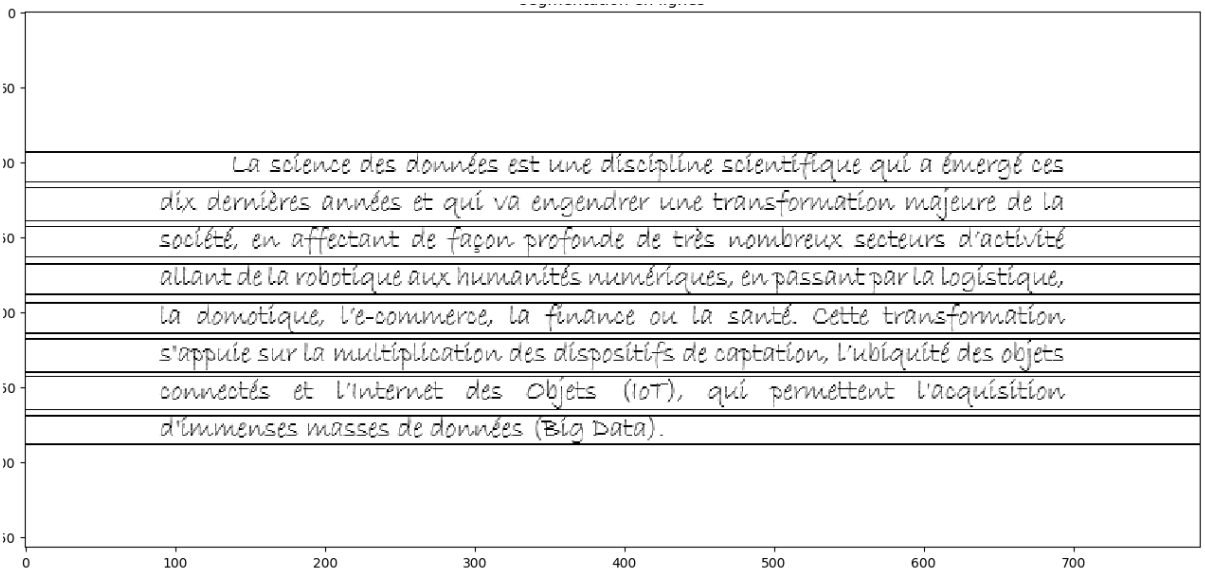


Figure 8: Résultat de la segmentation en lignes

Pour découper les mots de la première ligne par exemple on a besoin de sa projection horizontale.

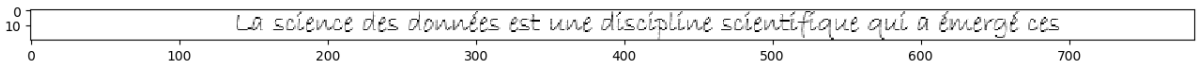


Figure 9: Image de la 1ère ligne découpée

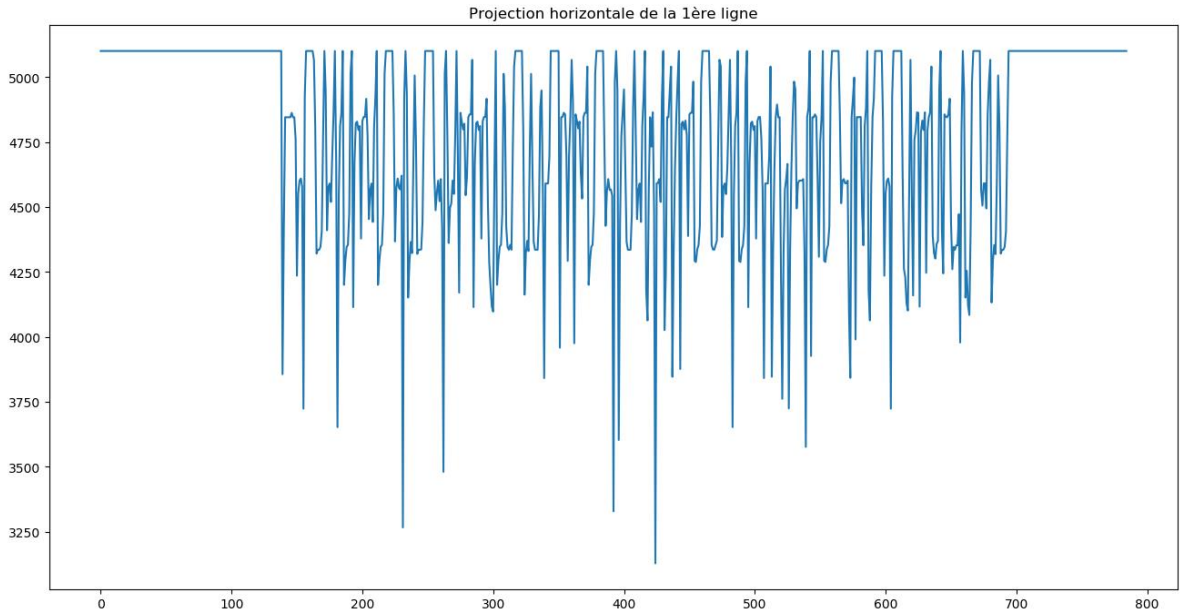


Figure 10: Projection horizontale sur la 1ère ligne

En manipulant ce vecteur, on obtient les indices de début et de fin de chaque de mot.

Tous les mots sont sauvegardés dans une liste nommés mots

Nom	Type	Taille
mots	list	83

Figure 11: Le programme a découpé 83 mots

Cette méthode de segmentation présente des défauts, la ponctuation par exemple est confondue avec le texte, comme dans les deux derniers mots de notre exemple :

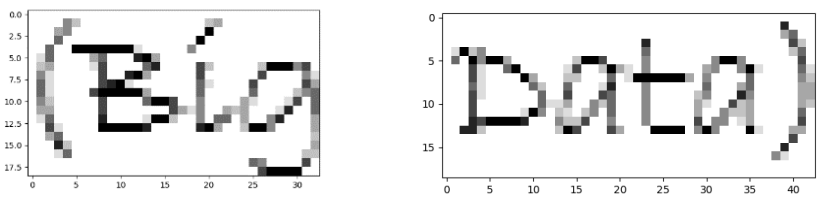


Figure 12:Exemple de mots découpés avec une ponctuation

Le point présent à la fin du texte, il n'est pas confondu avec le texte puisqu'il a été découpé comme un seul mot

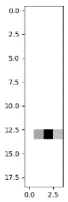


Figure 13:Exemple de ponctuation reconnue comme mot

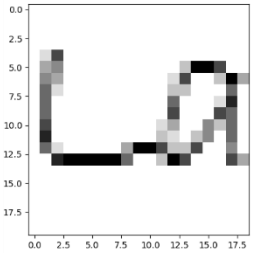
Extraction des caractéristiques :

Pour pouvoir comparer le mot recherché à l'ensemble des mots découpés, on a besoin de faire des comparaisons afin de détecter les similitudes.

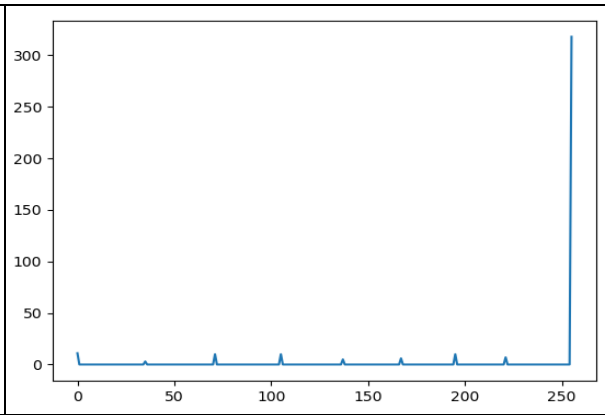
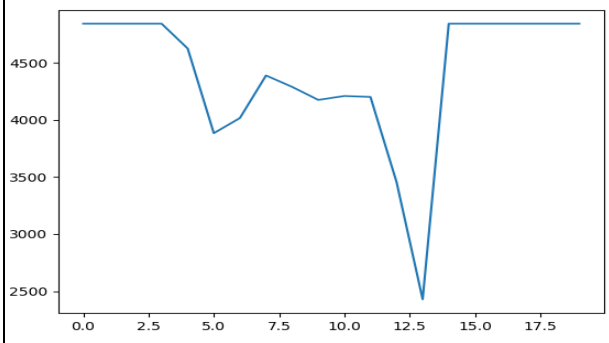
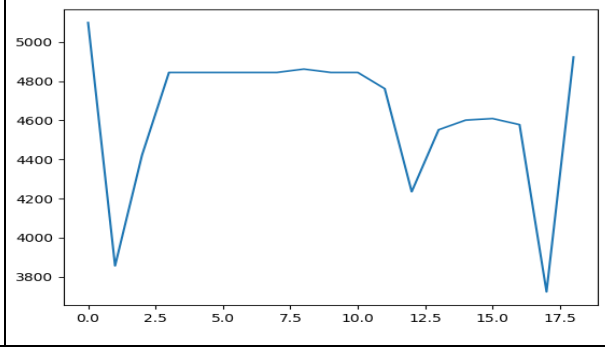
C'est pour cela qu'on extrait des différentes caractéristiques, chaque image de mot alors est définie par son vecteur de caractéristiques qui sera comparé au vecteur de caractéristique de l'image du mot recherché.

Les caractéristiques extraites sont : la hauteur, la largeur, la surface, le niveau de gris moyen, l'histogramme des niveaux de gris, projection vertical, projection horizontal.

Le tableau suivant lustre les résultats obtenus sur l'image suivante prise comme exemple :



Les caractéristiques	
la hauteur	20 pixels
la largeur	56 pixels
la surface	1220 pixels²
le niveau de gris moyen	232.05

l'histogramme des niveaux de gris	
projection vertical	
projection horizontal.	

TRAVAIL REALISE ET DIFFICULTE RENCONTRE

- **Travail réalisé :**

Durant la période de la réalisation de notre projet nous avons réussi à réaliser :

- Une interface graphique pour le système où l'utilisateur peut charger une image, capturer le mot qui veut chercher, et avoir un résultat en retour.
- Script en python capable de segmenter l'image du document, donc séparer en ligne , et séparation en mots.
- Extraction des caractéristiques de chaque image de mot (la hauteur, la largeur, la surface, le niveau de gris moyen, l'histogramme des niveaux de gris, projection vertical, projection horizontal).

- **Contrainte technique :**

- Liaison entre l'interface graphique et le scripte en python (le lancement du programme ne se fais pas automatiquement, on doit chercher une solution pour que le traitement se fais quand l'utilisateur manipule l'interface)
- L'extraction des caractéristiques (quelles sont les caractéristiques optimales à choisir pour que le taux de reconnaissance soit bon, et le taux d'erreur soit minimal).
- Quel traitement réaliser sur l'image avant de procéder à l'extraction des caractéristiques.
- Normalisation des tailles des images avant extraction et comparaison des caractéristiques.

CONCLUSION ET PERSPECTIVES

Dans ce projet, nous avons étudié les différentes approches du word spotting , et nous avons essayé de réaliser un système complet capable de traiter des documents texte, où l'utilisateur est mené à ouvrir une image de document à travers une interface graphique, et capturer le mot qui veut chercher, à travers les caractéristiques que nous avons choisis, et sur lesquelles le système se base pour comparer les mots nous avons obtenus des mauvais résultats, on déduit donc que la performance du système dépend fortement des caractéristiques extraites, le projet de word spotting est un travail compliqué où on doit optimaliser les résultats, donc au futur on pourra continuer sur ce travail en essayant de trouver la meilleure approche et les meilleures méthodes pour un système plus performant.

WEBOGRAPHIE

- Fischer. A,Keller. A, Frinken. V, Bunke. H. 2012. «Lexicon-Free Handwritten Word Spotting Using Character HMMs. »

- Ghorbel, A, Ogier. M. Vincent. 2015. «A Segmentation Free Word Spotting for Handwritten Documents »

- Adam Ghorbel, Jean-Marc Ogier, Nicole Vincent du Laboratoire SIP-LIPADE, Université Sorbonne Paris Descartes « Adaptation des caractéristiques pseudo-Haar pour le word spotting dans les documents manuscrits »

- Angelos P. Giotis , Giorgos Sfikas , Basilis Gatos , Christophoros Nikou « A survey of document image word spotting techniques»

- Marçal Rusiñol, DavidAldavert, RicardoToledo, JosepLladós « Efficient segmentation-freekeywordspottinginhistorical document collections»