

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

polytech.univ-tours.fr

Projet de programmation et génie logiciel 2017-2018

Iwordspotting Outil

Tuteur académique

Nicolas RAGOT

Étudiants

Alafate ABULIMITI (DI4)

Yuanyuan LI (DI4)

Liste des intervenants

Nom	Email	Qualité
Alafate ABULIMITI	alafate.abulimiti@etu.univ-tours.fr	Étudiant DI4
Yuanyuan LI	yuanyuan.li@etu.univ-tours.fr	Étudiant DI4
Nicolas RAGOT	nicolas.ragot@univ-tours.fr	Tuteur académique, Département Informatique



Avertissement

Ce document a été rédigé par Alafate ABULIMITI et Yuanyuan LI susnommés les auteurs.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Nicolas RAGOT susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

Les auteurs reconnaissent assumer l'entièr responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

Les auteurs attestent que les propos du document sont sincères et assument l'entièr responsabilité de la véracité des propos.

Les auteurs attestent ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

Les auteurs attestent que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

Les auteurs reconnaissent qu'ils ne peuvent diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

Les auteurs autorisent l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

Alafate ABULIMITI et Yuanyuan LI, *Iwordspotting Outil*, Projet de programmation et génie logiciel, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2017-2018.

```
@mastersthesis{  
    author={ABULIMITI, Alafate and LI, Yuanyuan},  
    title={Iwordspotting Outil},  
    type={Projet de programmation et génie logiciel},  
    school={Ecole Polytechnique de l'Université François Rabelais de Tours},  
    address={Tours, France},  
    year={2017-2018}  
}
```



Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	iii
1 Objectifs du projet	1
1 Objectif	1
2 Environnement et outils	1
2 Introduction des outils	2
1 Description générale.....	2
2 Binarization Tool	3
2.1 Introduction	3
2.2 Utilisation	3
2.3 Exemple	3
3 SegmentationTool	4
3.1 Introduction	4
3.2 Utilisation	4
3.3 Exemple	4
4 ExtracCaracsTool	6
4.1 Introduction	6
4.2 Utilisation	6

TABLE DES MATIÈRES

4.3	Exemple	6
5	MatchingToolBox.....	8
5.1	Introduction	8
5.2	Utilisation	8
5.3	Exemple	8
3	Spécification du logiciel	10
1	Description de la compétition.....	10
2	Description générale du logiciel.....	10
3	Environnement du projet	10
4	Caractéristiques des utilisateurs	11
5	Description interface homme/machine	11
6	Description du fichier du résultat.....	12
4	Modélisation du logiciel	13
1	Diagramme de séquence	13
2	Diagramme de classes	13
5	Implémentation	15
1	Exécution des outils.....	15
2	Amélioration d'outil ExtracCaracsTool.....	15
3	Amélioration d'outil MatchingToolBox	15
4	Utilisation d'outil.....	16
5	Implémentation de classes	16
6	Exécution d'outil en série	16
7	Répertoire de travail.....	16
6	Test	18
1	Exécution du document jar	18
2	Processus et résultat	19
3	Vérification du document XML.....	20
4	Vérification d'image	21
4.1	Segmentation	21
4.2	MatchingToolBox	22
7	Amélioration d'outil	23
1	SegmentationTool	23
2	MatchingToolBox.....	24



Table des figures

2 Introduction des outils

1	Les étapes d'utilisations d'outils	2
2	Binarisation Tool : Exécution	3
3	Image originale	3
4	Image binaire	4
5	Segementation Tool : Exécution	5
6	Résultat	5
7	Document XML	5
8	ExtracCaracs : Exécution	6
9	CSV de caractéristiques.....	7
10	Contenu du document CSV.....	7
11	MatchingToolBox : Exécution.....	8
12	Résultat de SegmentationTool.....	9

3 Spécification du logiciel

1	La commande sur Window cmd	11
2	Interface graphique	11
3	Résultat document XML.....	11

4 Modélisation du logiciel

1	Diagramme de séquence	13
2	Diagramme de classes	14

5 Implémentation

1	Répertoire de travail	16
2	Dossiers de travail	17

6 Test

1	Exécution	18
2	Mot 'ce'	18
3	Mot 'au'	18
4	Document 1	19
5	Document 2	19
6	Processus de la segmentation	19
7	Status final	20
8	Document du résultat.....	21
9	Résultat de segmentation	21
10	Document marqué.....	21
11	Documents du résultat	22
12	Bon résultat.....	22

7 Amélioration d'outil

1	Résultat de SegetationTool	23
2	SegmentationTool : Document marqué non-correspondant	23
3	MatchingToolBox : Document marqué non-correspondant	24

1

Objectifs du projet

1 Objectif

L'objectif du projet consiste à répondre aux spécifications permettant de concourir à des compétitions scientifiques de recherche de mots dans des images de documents par utiliser les quatre outils (Binarisation, SegmentationTool, ExtractTool, MatchingToolBox) dans la plate-forme "iwordspotting" en C # / C++.

2 Environnement et outils

En réalisant ce projet, nous programmons en Eclipse pour obtenir les fonctionnalités requises dans l'environnement Window.

2

Introduction des outils

1 Description générale

Le projet de <iwordspotting> qui peut être utilisé pour recherche mot-image dans des collections d'images étant développé dans le cadre de la formation d'ingénieur à l'École polytechnique de l'université Tours. Il est encadré par M. Nicolas RAGOT et réalisé et amélioré par Zheng ZHANG, Dawei SHEN, Loreen LAMBIN, M. Frédéric RAYAR, Bastien MEUNIER et Yamin ZAIDOU.

L'ordre d'utilisation de ces outils comme Figure 1 ci-dessous. Nous pouvons diviser ces étapes en 3 phases : (1) Traitement de mot-image : d'abord, il faut convertir l'image originale en noir et blanc. Après, il faut utiliser l'outil ExtracCaracTool avec le résultat d'outil Binarization pour obtenir les caractéristiques de mot-image. (2) Traitement d'image de document : d'abord, il faut segmenter l'image originale en ligne ut en mot. Après, il faut convertir le résultat d'outil SegmentationTool en noir et blanc. Finalement, comme le traitement de mot-image, il faut utiliser l'outil ExtracCaracTool avec le résultat d'outil Binarization pour obtenir les caractéristiques d'image de document. (3) Comparaison : finalement, il faut comparer les caractéristiques de mot-image et d'image de document avec utiliser l'outil MatchingToolBox.

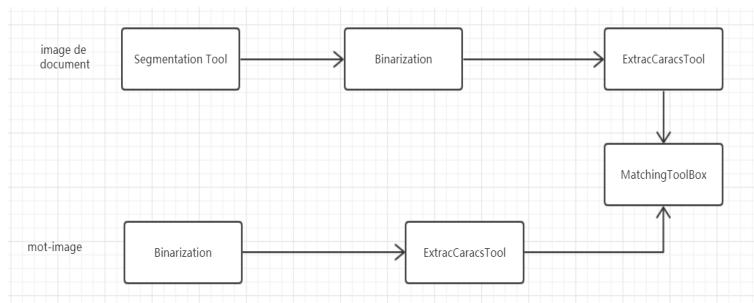


Figure 1 – Les étapes d'utilisations d'outils

2 Binarization Tool

2.1 Introduction

Le premier outil – Binarization, il permet de convertir des images en noir et blanc. Il est développé en C++, programmé et compilé en C++ Visual Studio. Il utilise la bibliothèque de OpenCV pour traiter.

2.2 Utilisation

Nous utilisons le document exécutable de cet outil. Cet outil peut traiter tous les images dans un dossier.

Entrée : le chemin absolu de dossier.

Sortie : Il va créer un dossier qui s'appelle 'Binary' dans le dossier entré, tous les résultats (image de format JPG en noir et blanc) sont enregistrés dans ce dossier.

2.3 Exemple

Exécuter : convertir l'image en noir et blanc.

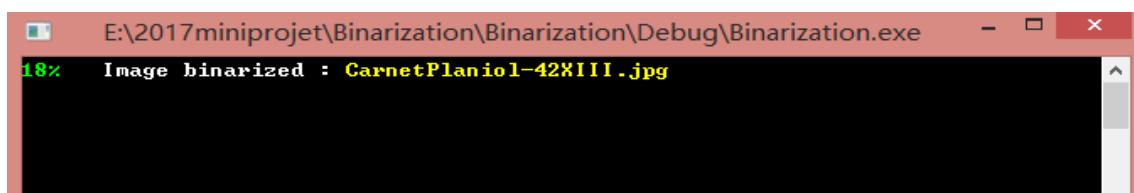


Figure 2 – Binarisation Tool : Exécution

L'image originale : CarnetPlaniol-42XIII.jpg.

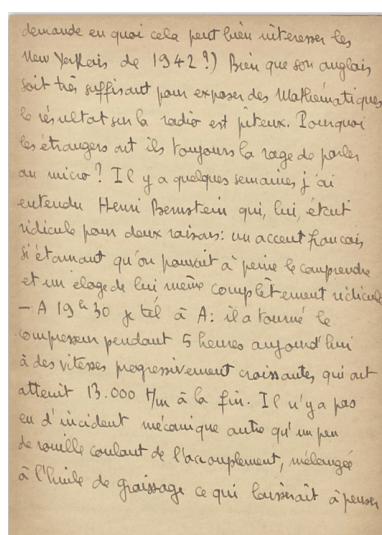


Figure 3 – Image originale

Résultat : l'image en noir et blanc.

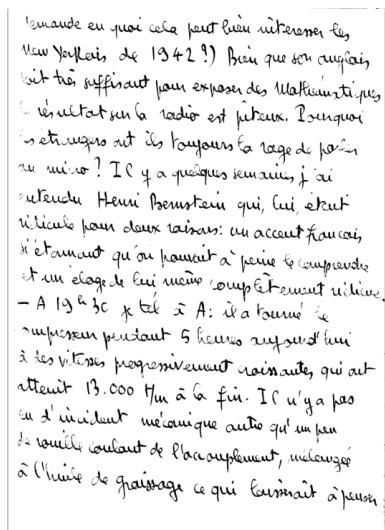


Figure 4 – Image binaire

3 SegmentationTool

3.1 Introduction

Le deuxième outil : SegmentationTool, il permet de séparer de l'image originale selon ligne ou mot pour bien comparer avec mot-image. Il est développé en C#, programmé et compilé en Visual Studio.

3.2 Utilisation

Nous utilisons le document exécutable de cet outil. Cet outil peut traiter toutes les images dans un dossier.

Entrée : Le chemin absolu de dossier

Sortie : Il va créer un dossier ‘Training’ dans quel Il y a trois dossiers : ‘componentImg’, ‘Coordonnees’ et ‘features’. Tous les fragments d’image originale sont enregistrés dans le dossier ‘componentImg’. Le document XML dans le dossier ‘Coordonnees’ indique les caractéristiques de fragments d’image originale.

3.3 Exemple

Exécuter :

entrée : l'image originale : CarnetPlaniol-42XIII.jpg

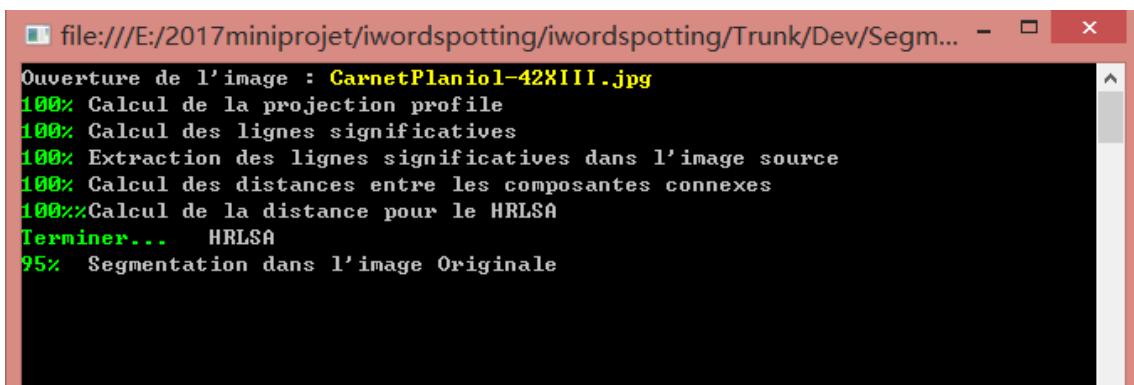


Figure 5 – Segmentation Tool : Exécution

Résultat : Un des images composantes :

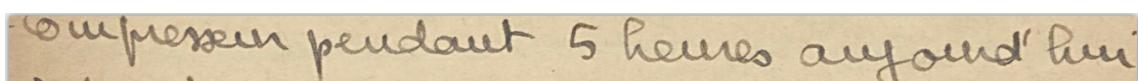


Figure 6 – Résultat

Le document XML des fragments : Chaque fragment représente une ligne ou un mot.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <root>
3      <Element index="CarnetPlaniol-42XIII_0">
4          <Coordinate coord="147,48,203,2658" />
5      </Element>
6      <Element index="CarnetPlaniol-42XIII_1">
7          <Coordinate coord="45,2421,208,2774" />
8      </Element>
9      <Element index="CarnetPlaniol-42XIII_2">
10         <Coordinate coord="71,1375,217,2735" />
11     </Element>
12     <Element index="CarnetPlaniol-42XIII_3">
13         <Coordinate coord="84,1570,222,2821" />
14     </Element>
15     <Element index="CarnetPlaniol-42XIII_4">
16         <Coordinate coord="92,2598,235,2929" />
17     </Element>
18     <Element index="CarnetPlaniol-42XIII_5">
19         <Coordinate coord="108,272,237,2861" />
20     </Element>
21     <Element index="CarnetPlaniol-42XIII_6">
22         <Coordinate coord="97,692,242,2820" />
23     </Element>
24     <Element index="CarnetPlaniol-42XIII_7">
25         <Coordinate coord="81,1131,244,2583" />
26     </Element>

```

Figure 7 – Document XML

Explication : Pour chaque élément : il s'agit que la position du fragment 'CarnetPlaniol – 42 XIII_1.jpg' dans l'image 'CarnetPlaniol – 42 XIII.jpg'. 'coord' indique coordonnée-X, coordonnée-Y, la hauteur de ce fragment, le longueur de ce fragment.

4 ExtracCaracsTool

4.1 Introduction

Le troisième outil - ExtracCaracsTool, il permet de traiter des fragments d'image originale pour obtenir des caractéristiques de ces fragments. Ces caractéristiques permettent de bien décrire l'image pour faire la correspondance entre deux images. Il est développé en C#, programmé et compilé en Visual Studio.

4.2 Utilisation

Nous utilisons le document exécutable de cet outil. Cet outil peut traiter tous les fragments dans le dossier 'componentImg'.

Entrée : Le chemin absolu de dossier 'Training'

Sortie : Toutes les caractéristiques de fragments sont enregistrées dans le document du format CSV.

4.3 Exemple

Exécuter :

```

file:///E:/2017miniprojet/iwordspotting/iwordspotting/Trunk/Dev/Extra...
Line extracted and saved !
E:/2017miniprojet/testImage/imageOriginale/Training/features/CarnetPlaniol-42XII
I_0.csv : SavedLine extracted and saved !
E:/2017miniprojet/testImage/imageOriginale/Training/features/CarnetPlaniol-42XII
I_1.csv : SavedLine extracted and saved !
E:/2017miniprojet/testImage/imageOriginale/Training/features/CarnetPlaniol-42XII
I_2.csv : SavedLine extracted and saved !
E:/2017miniprojet/testImage/imageOriginale/Training/features/CarnetPlaniol-42XII
I_3.csv : SavedLine extracted and saved !
E:/2017miniprojet/testImage/imageOriginale/Training/features/CarnetPlaniol-42XII
I_4.csv : SavedLine extracted and saved !
E:/2017miniprojet/testImage/imageOriginale/Training/features/CarnetPlaniol-42XII
I_5.csv : Saved

```

Figure 8 – ExtracCaracs : Exécution

Résultat :

Le document CSV de caractéristiques :

CarnetPlaniol-42XIII_0.csv	2018/1/6 18:17	Microsoft Excel ...	43 KB
CarnetPlaniol-42XIII_1.csv	2018/1/6 18:17	Microsoft Excel ...	42 KB
CarnetPlaniol-42XIII_2.csv	2018/1/6 18:17	Microsoft Excel ...	40 KB
CarnetPlaniol-42XIII_3.csv	2018/1/6 18:17	Microsoft Excel ...	41 KB
CarnetPlaniol-42XIII_4.csv	2018/1/6 18:17	Microsoft Excel ...	39 KB
CarnetPlaniol-42XIII_5.csv	2018/1/6 18:17	Microsoft Excel ...	36 KB
CarnetPlaniol-42XIII_6.csv	2018/1/6 18:17	Microsoft Excel ...	35 KB
CarnetPlaniol-42XIII_7.csv	2018/1/6 18:17	Microsoft Excel ...	31 KB
CarnetPlaniol-42XIII_8.csv	2018/1/6 18:17	Microsoft Excel ...	33 KB
CarnetPlaniol-42XIII_9.csv	2018/1/6 18:17	Microsoft Excel ...	35 KB
CarnetPlaniol-42XIII_10.csv	2018/1/6 18:17	Microsoft Excel ...	30 KB
CarnetPlaniol-42XIII_11.csv	2018/1/6 18:17	Microsoft Excel ...	31 KB
CarnetPlaniol-42XIII_12.csv	2018/1/6 18:17	Microsoft Excel ...	26 KB
CarnetPlaniol-42XIII_13.csv	2018/1/6 18:17	Microsoft Excel ...	29 KB
CarnetPlaniol-42XIII_14.csv	2018/1/6 18:17	Microsoft Excel ...	32 KB
CarnetPlaniol-42XIII_15.csv	2018/1/6 18:17	Microsoft Excel ...	44 KB
CarnetPlaniol-42XIII_16.csv	2018/1/6 18:17	Microsoft Excel ...	47 KB

Figure 9 – CSV de caractéristiques

Le contenu de document CSV :

	A	B	C	D	E	F	G
1	0.667787	0	0.085714	0.185714	0.1	0.1	0.140816
2	0.642801	0	0.128571	0.185714	0.057143	0.071429	0.157143
3	0.636022	0	0.1	0.314286	0.214286	0.171429	0.185714
4	0.645098	0	0.1	0.314286	0.214286	0.128571	0.204762
5	0.661345	0	0.2	0.271429	0.071429	0.028571	0.235714
6	0.691709	0	0.371429	0.371429	0	0.014286	0.371429
7	0.703922	0	0.442857	0.442857	0	0.014286	0.442857
8	0.711204	0	0	0	0	0	0
9	0.707339	0	0	0	0	0	0

Figure 10 – Contenu du document CSV

Il y a sept caractéristiques pour chaque enregistrement :

La première caractéristique : il décrit le niveau gris de la colonne. Il d'abord calculer la somme du niveau de gris des pixels de la colonne. Après, il est normalisé entre 0 et 1.

La deuxième caractéristique : il décrit le nombre de Transitions blanc en noir. Il doit être nombre entier.

La troisième caractéristique : il décrit le premier pixel noir au bord haut de l'image. Il est normalisé entre 0 et 1 en divisant par la hauteur de l'image.

La quatrième caractéristique : il décrit le dernier pixel noir au bord haut de l'image. Il est normalisé entre 0 et 1 en divisant par la hauteur de l'image.

La cinquième caractéristique : il décrit la distance entre le pixel noir le plus haut et celui du plus bas d'image. Il est normalisé entre 0 et 1 en divisant par la hauteur de l'image.

La sixième caractéristique : il décrit le pourcentage de pixel noir dans une colonne. Il calcule le nombre de pixel noir, après cette valeur est normalisé en divisant par la hauteur de la colonne.

La septième caractéristique : il décrit le centre de gravité de la colonne. Il d'abord calcule la somme de la position de tous les pixels noirs, après, cette valeur est divisée par le nombre de pixel noirs. Finalement, la valeur obtenue est normalisée en divisant par la hauteur de l'image.

5 MatchingToolBox

5.1 Introduction

Le quatrième outil - MatchingToolBox, il permet de faire la correspondance entre deux images par comparer les caractéristiques de ces images. Il y a plusieurs algorithmes pour comparer ces caractéristiques. Il est développé en C++, programmé et compilé en Visual Studio.

5.2 Utilisation

Nous utilisons le document exécutable de cet outil.

Entrée : Le chemin absolu d'image originale.

Sortie : Toutes les correspondances sont enregistrées dans le document du format XML.

5.3 Exemple

Exécuter :

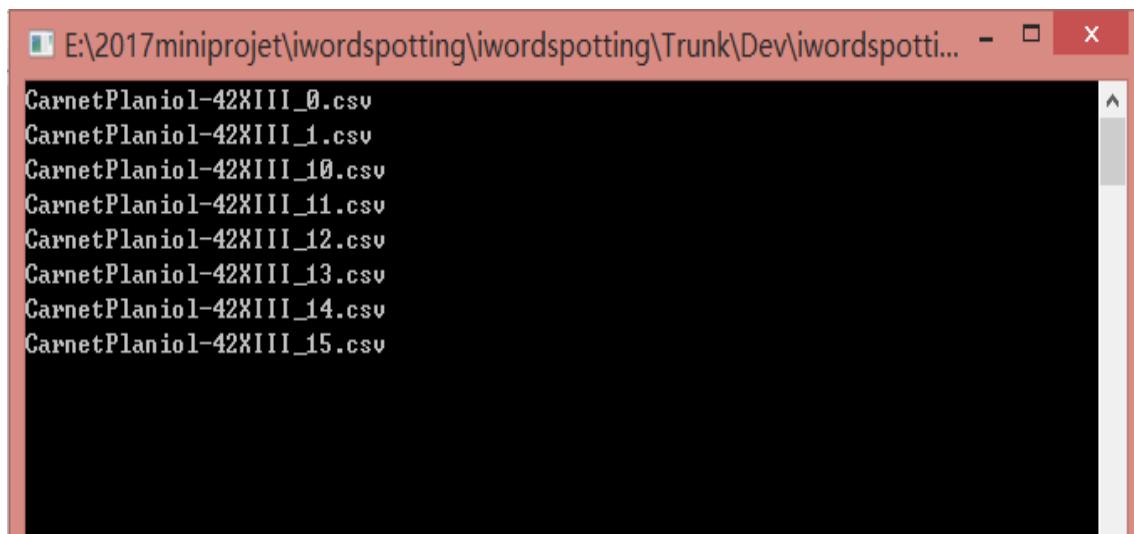


Figure 11 – MatchingToolBox : Exécution

Résultat : le document XML de correspondances(figure 12).

Explication : chaque résultat décrit une correspondance. ‘coord’ décrit la position de la correspondance dans l’image originale. Ces quatre valeurs représentent : coordonnée-X, coordonnée-Y, la hauteur et la longueur.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <root>
3   <Result id="1">
4     <Image path="C:\Users\Administrateur\Desktop\SVN\Ressources\Archives\Images\CarnetPlaniol-42XIII-003.jpg"/>
5     <Coordinate coor="770,349,121,249"/>
6   </Result>
7   <Result id="2">
8     <Image path="C:\Users\Administrateur\Desktop\SVN\Ressources\Archives\Images\CarnetPlaniol-42XIII-003.jpg"/>
9     <Coordinate coor="560,585,136,278"/>
10  </Result>
11  <Result id="3">
12    <Image path="C:\Users\Administrateur\Desktop\SVN\Ressources\Archives\Images\CarnetPlaniol-42XIII-005.jpg"/>
13    <Coordinate coor="991,1931,185,383"/>
14  </Result>
15  <Result id="4">
16    <Image path="C:\Users\Administrateur\Desktop\SVN\Ressources\Archives\Images\CarnetPlaniol-42XIII-011.jpg"/>
17    <Coordinate coor="1156,2262,167,346"/>
18  </Result>
19  <Result id="5">
20    <Image path="C:\Users\Administrateur\Desktop\SVN\Ressources\Archives\Images\CarnetPlaniol-42XIII-015.jpg"/>
21    <Coordinate coor="690,793,130,268"/>
22  </Result>
23  <Result id="6">
24    <Image path="C:\Users\Administrateur\Desktop\SVN\Ressources\Archives\Images\CarnetPlaniol-42XIII-015.jpg"/>
25    <Coordinate coor="1322,933,135,278"/>
26  </Result>
27 </root>
..n

```

Figure 12 – Résultat de SegmentationTool

3

Spécification du logiciel

1 Description de la compétition

Notre tâche principale est de répondre aux exigences la compétition ICDAR2015

La compétition ICDAR2015 sur “Keyword Spotting” pour les documents manuscrits est organisé dans le cadre de la compétition ICDAR 2015. L’objectif de cette compétition est de comparer et de promouvoir différentes approches utilisées dans le domaine du “Keyword Spotting” (KWS).

Les exigences de cette compétition

Entrée : Le chemin des requêtes et le chemin des documents.

Sortie : Un fichier de XML qui contient les IDs des résultats , les chemins des documents correspondants , les coordonnées correspondantes (X, Y, Width, Height) et les distances pour chaque résultat.

2 Description générale du logiciel

Ce projet est créé en JAVA. Il utilise l’outil de ‘Binarisation’, ‘SegmentationTool’, ‘ExtraCaracTool’, ‘MatchingToolBox’ pour rechercher l’image de mot clé dans l’image de document. Il permet de chercher des mot clé dans plusieurs images de document. L’utilisateur transfère des mot-images et images de document, après, il permet de retourner un document XML de coordonnées où le mot clé apparaît dans l’image de texte. Il utilise l’algorithme de ‘Continuous Dynamic Programming’ pour comparer caractéristiques d’image de document et celles de mot-image. Tous les résultats de coordonnées est mise en ordre par distance croissant.

3 Environnement du projet

L’outil de ‘Binarisation’, de ‘SegmentationTool’, de ‘ExtraCaracTool’ et de ‘MatchingToolBox’ sont créés et développés sous Windows. Donc, notre projet est créé et développé sous système Windows. notre projet est programmé en Eclipse.

4 Caractéristiques des utilisateurs

Ce projet est développé pour tous les utilisateurs qui utilisent Windows, et veulent rechercher l'image de mot clé dans l'image de documents.

5 Description interface homme/machine

l'utilisateur peut appeler le document JAR de notre projet dans Windows cmd. D'abord, comme figure 12, l'utilisateur entre le chemin de 'requête' et le chemin de "Document".

```
G:\workspace_eclipse\Competition>java -jar competition.jar "E:\2017miniprojet\test\request" "E:\2017miniprojet\test\image5"
```

Figure 1 – La commande sur Window cmd

Il peut aussi utiliser interface graphique comme figure 2.

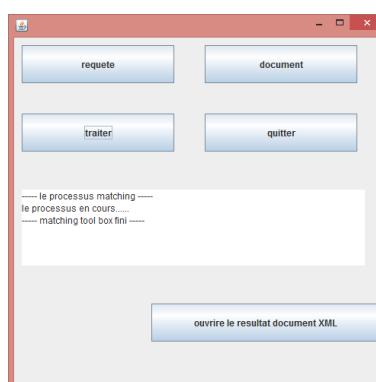


Figure 2 – Interface graphique

Quand le processus finit, nous pouvons ouvrir le fichier de résultat.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root>
    <au_0 id="1">
        <Image path="G:/workspace_eclipse/Competition/base/2018-01-04-21-20-45/M0275_02_0209.jpg"/>
        <Coordinate coor="164, 106, 64, 83"/>
        <Distance dis="0.19097"/>
    </au_0>
    <au_0 id="2">
        <Image path="G:/workspace_eclipse/Competition/base/2018-01-04-21-20-45/M0275_02_0209.jpg"/>
        <Coordinate coor="101, 291, 62, 81"/>
        <Distance dis="0.19671"/>
    </au_0>
    <au_0 id="3">
        <Image path="G:/workspace_eclipse/Competition/base/2018-01-04-21-20-45/M0275_02_0209.jpg"/>
        <Coordinate coor="95, 1382, 55, 71"/>
        <Distance dis="0.207346"/>
    </au_0>
    ...

```

Figure 3 – Résultat document XML

Le résultat est enregistré dans le fichier ResultatParOrdre.xml.

6 Description du fichier du résultat

<au_0 id= ‘1’> : c'est-à-dire que l'image de mot clé est au_0.jpg.

<Image path=’XXXXXX’> : il enregistre l'image de document.

<Coordinate coor=’164,106,64,83’> : c'est la coordonnée de résultat fragment, coordonnée-X=164, coordonnée-Y=106, largeur de fragment = 64, hauteur de fragment = 83.

<Distance dis= “xxxx”> : c'est la similarité de ce résultat avec le mot-image (utilisant l'algorithme CDP).

4

Modélisation du logiciel

1 Diagramme de séquence

Pour expliquer les fonctions d'outil, nous avons créé un diagramme de séquence.

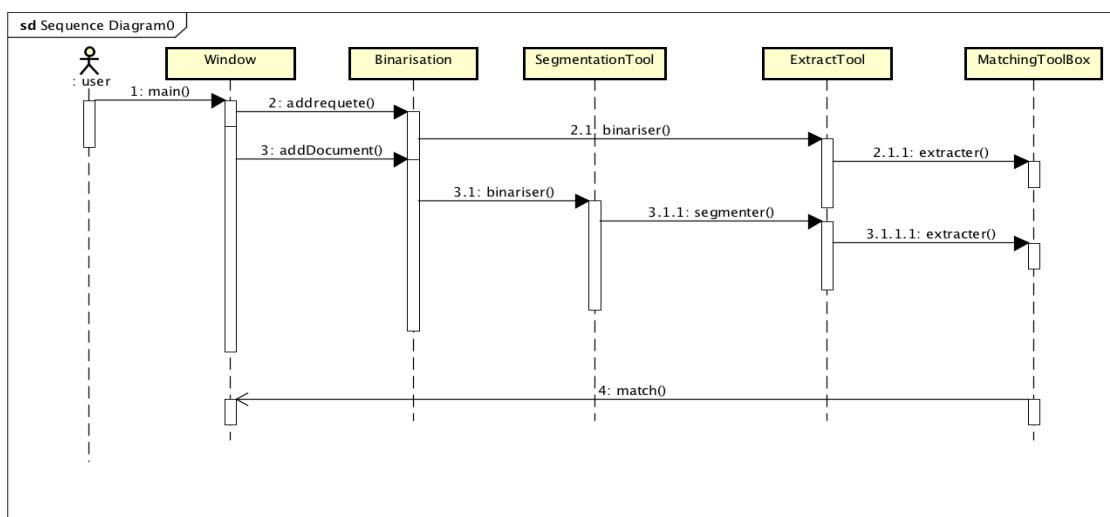


Figure 1 – Diagramme de séquence

Nous utilisons le design pattern de MVC pour construire notre projet. Le package modele est pour enregistrer des résultats en utilisant des outils. Le package contrôleur est pour gérer ces résultats et les fichiers de ces résultats. Nous pouvons utiliser la classe «VerifierResultat» dans le package verifier pour dessiner les résultats de SegmentationTool et de MatchingToolBox sur l'image.

2 Diagramme de classes

Nous avons utilisé le modèle de MVC pour réaliser cet objectif, et créé un diagramme de classes pour comprendre mieux de cette structure de logiciel.

Nous ajoutons notamment un nouveau package appelé "vérifier", il y a une classe : VerifierResultat. Nous pouvons marquer les documents en utilisant les coordonnées dans le fichier XML pour vérifier les résultats par les fonctions dans cette classe.

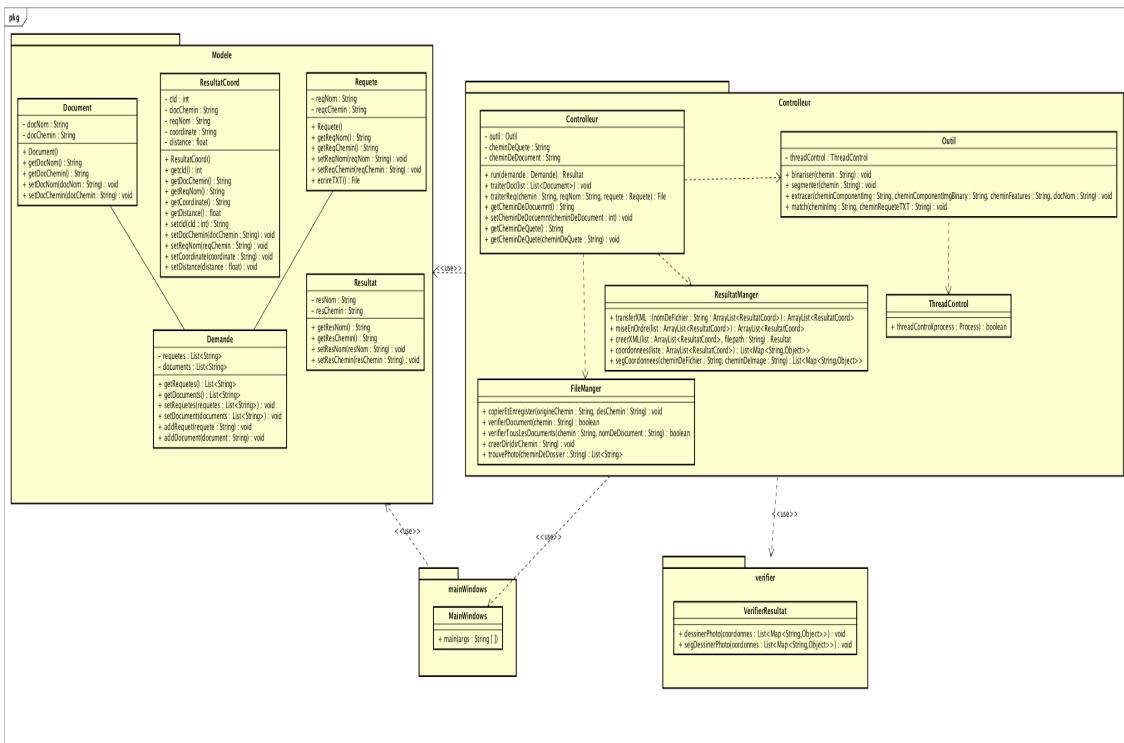


Figure 2 – Diagramme de classes

5

Implémentation

1 Exécution des outils

(1) Langage C# et C++ :

Ces outils sont créés et développés en C++ et C#, en Visual Studio, en Windows.

(2) Visual Studio 2012 :

Ces outils ne peuvent qu'exécuter sur la version de Visual Studio 2012.

(3) Implémenter OpenCV :

Ces outils utilisent la bibliothèque de OpenCV. Nous utilisons Visual studio 2012 pour exécuter ces outils, donc il faut ajouter la bibliothèque de OpenCV 2.4.9-VC11.

(4) Faire les liaisons parmi les différents outils :

D'abord nous travaillons pour comprendre l'entrée et la sortie de chaque outil. Après, nous changeons le paramètre de l'entrée et quelque paramètre du document configuration, par exemple, le chemin du document exécutable d'outil Binarization.

2 Amélioration d'outil ExtracCaracsTool

Outil ExtracCaracsTool permet de traiter des fragments d'une image de texte à la fois. Mais il y a des fragments images de plusieurs images de texte dans notre dossier, donc, quand cet outil lit des images, il peut-être rencontrer l'erreur de dépasser des limites d'une liste. En conséquence, nous ajoutons une étape avant lire le document. Avant lequel, il faut déterminer si le nom de fragment est même que celle d'image originale.

3 Amélioration d'outil MatchingToolBox

Avant, cet outil ne retourne pas la distance de correspondance, mais nous voulons obtenir le résultat qui mise en ordre par distance. Par conséquent, nous ajoutons un attribut dans la classe «ResultCoord» qui s'appelle distance pour enregistrer la correspondance. Donc, quand cet outil enregistre les résultats, il enregistre la distance dans le document en même temps.

4 Utilisation d'outil

Outil Binarization permet de traiter plusieurs images à la fois.

Outil Segmentation aussi permet de traiter plusieurs images à la fois. Outil ExtracCaracsTool permet de traiter juste une image à la fois. Outil MatchingToolBox permet de comparer une image de mot clé avec plusieurs images de texte..

5 Implémentation de classes

Nous utilisons le design pattern de MVC pour construire notre projet. le package modele est pour enregistrer des résultats en utilisant des outils. le package contrôleur est pour gérer ces résultats et les fichiers de ces résultats. Nous pouvons utiliser la classe «VerifierResultat» dans le package verifier pour dessiner les résultats de segmentation Tool et de MatchingToolBox Tool sur l'image.

6 Exécution d'outil en série

Nous pouvons connaître que le processus de traiter d'image de mot de clé est sériel, c'est-à-dire que le processus de ExtracCaracsTool ne peut pas commencer avant le processus de Binarisation déjà fini. Donc, pour l'image de mot clé, le processus de ExtracCaracsTool faudrait attendre que le processus de Binarization finit. Pour l'image de document, le processus de ExtracCaracsTool faudrait attendre le processus de Binarization, le processus de Binarization faudrait attendre le processus de Segmentation Tool. Pour les deux (image mot clé & image document), le processus de matchingToolBox faudrait attendre le processus de ExtracCaracsTool d'eux.

Afin que nous pouvons réaliser exécuter les outil en série. nous utiliser la classe <ThreadControl> pour nettoyer 'InputStream' et 'ErrorStream' qui évite la suspension du processus.

7 Répertoire de travail

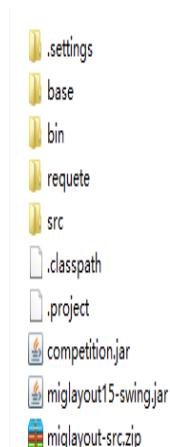


Figure 1 – Répertoire de travail

Pour bien enregistrer les images, quand le programme exécute, il va créer deux dossier, le dossier “base” est le répertoire de traiter des mot-images., le dossier “requete” est le répertoire de traiter des images de document. comme figure 1.

Chaque fois on exécute le document JAR, il va créer le dossier pour enregistrer et traiter les images que l’utilisateur demande. le nom de ce dossier est le temps. comme figure 2.

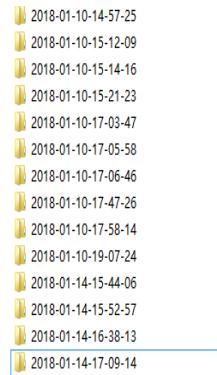


Figure 2 – Dossiers de travail

6 Test

1 Exécution du document jar

Exécuter le document jar sur Windows cmd comme figure 1.

```
G:\workspace_eclipse\Competition>java -jar competition.jar "E:\2017miniprojet\test\request" "E:\2017miniprojet\test\image5"
```

Figure 1 – Exécution

Rechercher le mot ‘ce’ (figure 2) et le mot ‘au’ (figure 3) dans les images (figure 4, figure 5).



Figure 2 – Mot ‘ce’



Figure 3 – Mot ‘au’

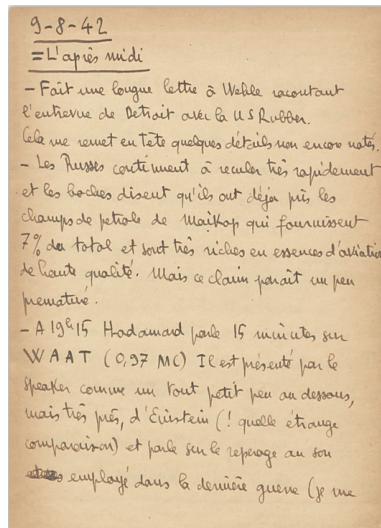


Figure 4 – Document 1

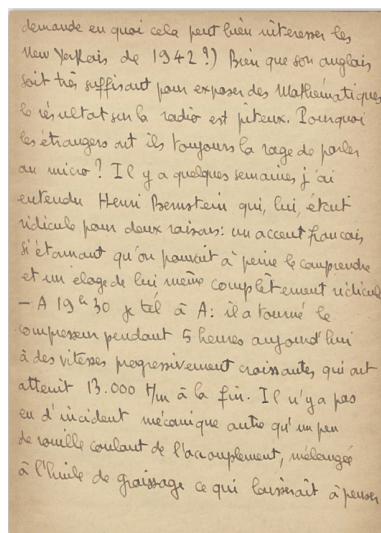


Figure 5 – Document 2

2 Processus et résultat

Quand il marche, nous pouvons savoir le programme exécute quel outil.

```
G:\workspace_eclipse\Competition>java -jar competition.jar "E:\2017miniprojet\test\request" "E:\2017miniprojet\test\image5"
copier le dossierG:/workspace_eclipse/Competition/base le dossier objectif déjà existe
copier le dossierG:/workspace_eclipse/Competition/base\non pas succès
copier le dossierG:/workspace_eclipse/Competition\requete le dossier objectif déjà existe
copier le dossierG:/workspace_eclipse\Competition\requete\non pas succès
copier le dossierG:/workspace_eclipse\Competition\base/2018-01-14-17-09-14\succes!
copier le dossierG:/workspace_eclipse\Competition\requete/2018-01-14-17-09-14\succes!
copier le dossierG:/workspace_eclipse\Competition\requete/2018-01-14-17-09-14\au\succes!
copier le dossierG:/workspace_eclipse\Competition\requete/2018-01-14-17-09-14\element\succes!
copier le dossierG:/workspace_eclipse\Competition\requete/2018-01-14-17-09-14\segmenter\succes!
copier le dossierG:/workspace_eclipse\Competition\requete/2018-01-14-17-09-14\segmenter\commence
E:\2017miniprojet\Binarization\Binarization\Debug\Binarization.exe
Image binarized : CarnetPlanilo-42X111-003.jpg
```

Figure 6 – Processus de la segmentation

Comme figure 6, nous savons que l'outil SegmentationTool commence d'exécuter.

```

creer le dossierG:/workspace_eclipse/Competition/base/2018-01-14-17-09-14\succes!
creer le dossierG:/workspace_eclipse/Competition/requete/2018-01-14-17-09-14\succes!
creer le dossierG:/workspace_eclipse/Competition/requete/2018-01-14-17-09-14/au\succes!
----copier succes ----
creer le dossierG:/workspace_eclipse/Competition/requete/2018-01-14-17-09-14/element\succes!
----copier succes ----
----copier succes ----
----copier succes ----
segmenter commence
segmenter finit
Binarisation commence
Binarisation finit
ExtracCaracsTool commence
ExtracCaracsTool finit
ExtracCaracsTool commence
ExtracCaracsTool finit
creer le dossierG:/workspace_eclipse/Competition/requete/2018-01-14-17-09-14/au/txtfile\succes!
Binarisation commence
Binarisation finit
creer le dossierG:/workspace_eclipse/Competition/requete/2018-01-14-17-09-14/au/features\succes!
ExtracCaracsTool commence
ExtracCaracsTool finit
----document txt enregistrer----
matching commence
matching finit
creer le dossierG:/workspace_eclipse/Competition/requete/2018-01-14-17-09-14/element/txtfile\succes!
Binarisation commence
Binarisation finit
creer le dossierG:/workspace_eclipse/Competition/requete/2018-01-14-17-09-14/element/features\succes!
ExtracCaracsTool commence
ExtracCaracsTool finit
----document txt enregistrer----
matching commence
matching finit
fini
---- matching tool box fini ----
le resultat de document XML est: G:/workspace_eclipse/Competition/base/2018-01-14-17-09-14/ResultatsParOrdre.xml

```

Figure 7 – Status final

Comme le figure 7, quand il affiche le chemin de résultat de document XML, il indique que tous les processus finissent. et nous pouvons savoir le chemin de résultat de document XML.

3 Vérification du document XML

Le document XML bien enregistre le nom de mot-image, le chemin d'image de document, le cordonnée et la distance (la correspondance). tous les correspondance sont mise en ordre croissant. Comme figure 8.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root>
    <au_0 id="1">
        <Image path="G:/workspace_eclipse/Competition/base/2018-01-14-17-09-14/CarnetPlaniol-42XIII-003.jpg"/>
        <Coordinate coor="245,45,90,118"/>
        <Distance dis="0.094952"/>
    </au_0>
    <au_0 id="2">
        <Image path="G:/workspace_eclipse/Competition/base/2018-01-14-17-09-14/CarnetPlaniol-42XIII.jpg"/>
        <Coordinate coor="556,48,203,267"/>
        <Distance dis="0.095383"/>
    </au_0>
    <au_0 id="3">
        <Image path="G:/workspace_eclipse/Competition/base/2018-01-14-17-09-14/CarnetPlaniol-42XIII.jpg"/>
        <Coordinate coor="2016,272,237,314"/>
        <Distance dis="0.106144"/>
    </au_0>
    <au_0 id="4">
        <Image path="G:/workspace_eclipse/Competition/base/2018-01-14-17-09-14/CarnetPlaniol-42XIII-003.jpg"/>
        <Coordinate coor="148,1110,127,165"/>
        <Distance dis="0.122146"/>
    </au_0>
    <au_0 id="5">
        <Image path="G:/workspace_eclipse/Competition/base/2018-01-14-17-09-14/CarnetPlaniol-42XIII-003.jpg"/>
        <Coordinate coor="1380,2106,161,211"/>
        <Distance dis="0.135014"/>
    </au_0>
</root> ...

```

Figure 8 – Document du résultat

4 Vérification d'image

4.1 Segmentation

Pour l'outil SegmentationTool, un des résultat comme ci-dessous (figure 9) :

Figure 9 – Résultat de segmentation

Nous vérifions sur l'image comme figure 10 :

Figure 10 – Document marqué

4.2 MatchingToolBox

Il y a plan de résultat comme figure 11 :



Figure 11 – Documents du résultat

Un des résultats (recherche le mot ‘au’) comme figure 12 :

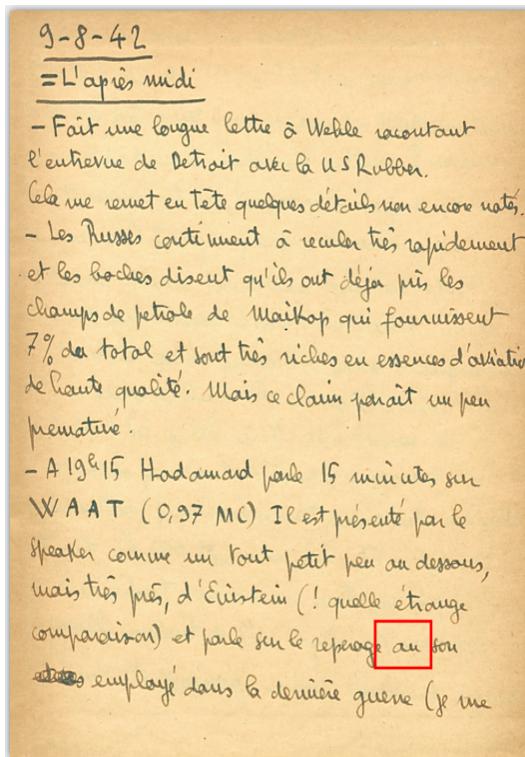


Figure 12 – Bon résultat

7

Amélioration d'outil

1 SegmentationTool

Quand nous utilisons l'outil SegmentationTool, nous trouvons qu'il y a quelques fragments d'image ne correspondent pas de coordonnées enregistrant dans le document XML, par exemple : le fragment comme figure 1 :

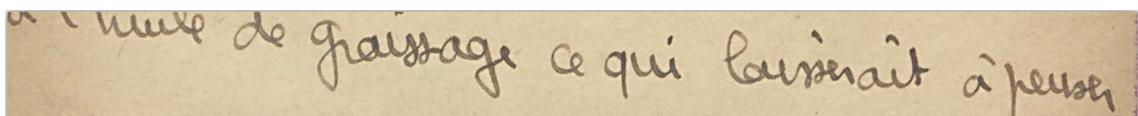


Figure 1 – Résultat de SegmentationTool

Mais quand nous dessinons sur image selon la coordonnée qui est enregistrée dans le document XML, nous obtenons l'image comme figure 2 :

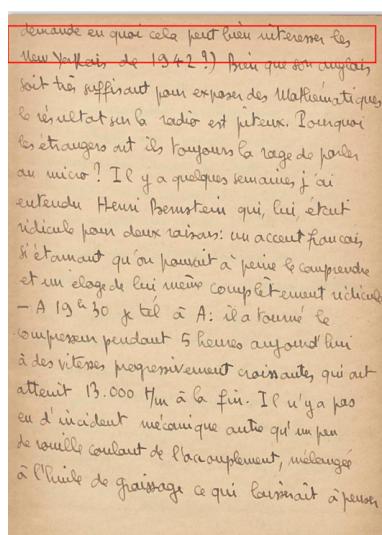


Figure 2 – SegmentationTool : Document marqué non-correspondant

Cela ne correspond pas de figure 1.

donc, dans la future de développement, c'est mieux d'améliorer l'outil segmentation Tool.

2 MatchingToolBox

Quand nous utilisons l'outil MatchingToolBox, nous trouvons qu'il y a quelques résultats ne correspondent pas de mot-image, par exemple comme figure 3 (recherche du mot 'au') :

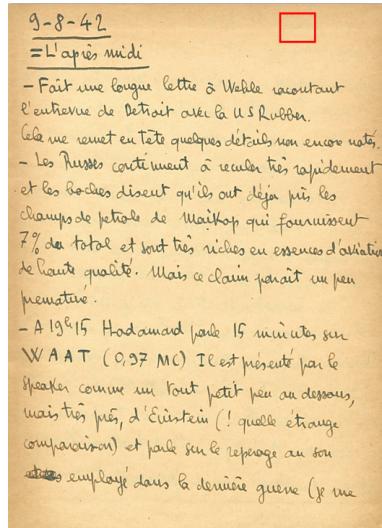


Figure 3 – MatchingToolBox : Document marqué non-correspondant

Donc dans la future de développement, c'est mieux d'améliorer l'outil MatchingToolBox.

Iwordspotting Outil

Résumé

L'objectif de notre projet est de réaliser un outil sur la plateforme JAVA en utilisant quatre outils (Binarisation, SegmentationTool, ExtracTool, et MatchingToolBox) dans la plateforme "iwordspotting" utilisant la bibliothèque de OpenCV, Afin de générer un fichier de XML pour répondre aux exigences de la compétition ICDAR2015 : Utiliser le mot-clé image pour trouver l'emplacement de cette image dans le document, et trier par la similitude. En réalisant cet outil , nous avons utilisé le MVC pour créer la structure du projet, Nous utilisons la façon de transmettre des paramètres sur la ligne de commande pour faciliter les travaux des testeurs.

Mots-clés

iwordspotting, ICDAR, C#, OpenCV, MVC

Abstract

The objective of our project is to build a tool on the JAVA platform by using four tools (Binarization, SegmentationTool, ExtracTool, and MatchingToolBox) in the platform "iwordspotting" which uses the OpenCV library, in order to generate an XML file to answer the requirements of the ICDAR2015 competition : Use the image keyword to find the location of this image in the document, and sort by similarity. By implementing this tool, we used the MVC to create the project structure, We also used the way to transmit parameters by the command line to facilitate the work of the testers.

Keywords

iwordspotting, ICDAR, C#, OpenCV, MVC

Tuteur académique

Nicolas RAGOT

Étudiants

Alafate ABULIMITI (DI4)

Yuanyuan LI (DI4)