



TP 2 :

Segmentation d'images par région pour la reconnaissance de caractères manuscrits

Presented To

Mr SOUALAH

Presented By

Anis BENINI

Sommaire

Objectif.....	4
I. Introduction et Prétraitement.....	4
1. Introduction.....	4
2. Étapes de prétraitement.....	4
II. Méthode de Segmentation : Croissance de Régions.....	6
1. Présentation de la méthode.....	6
2. Implémentation de l'algorithme.....	6
III. Post-Traitemet.....	7
1. Nettoyage des régions.....	7
2. Extraction des sous-images.....	7
3. Validation visuelle.....	8
4. Plus.....	9
IV. Conclusion.....	12

Objectif

L'objectif principal de ce travail pratique était de segmenter une image contenant des caractères manuscrits afin d'isoler chaque caractère en utilisant une méthode de segmentation par région. Ce processus comprenait les étapes suivantes :

- Le prétraitement de l'image pour la rendre plus adaptée à la segmentation.
- La segmentation en elle-même à l'aide d'un algorithme de croissance de régions.
- Le post-traitement pour obtenir des régions exploitables destinées à la reconnaissance de texte.

Ce rapport détaille chaque étape, les résultats obtenus.

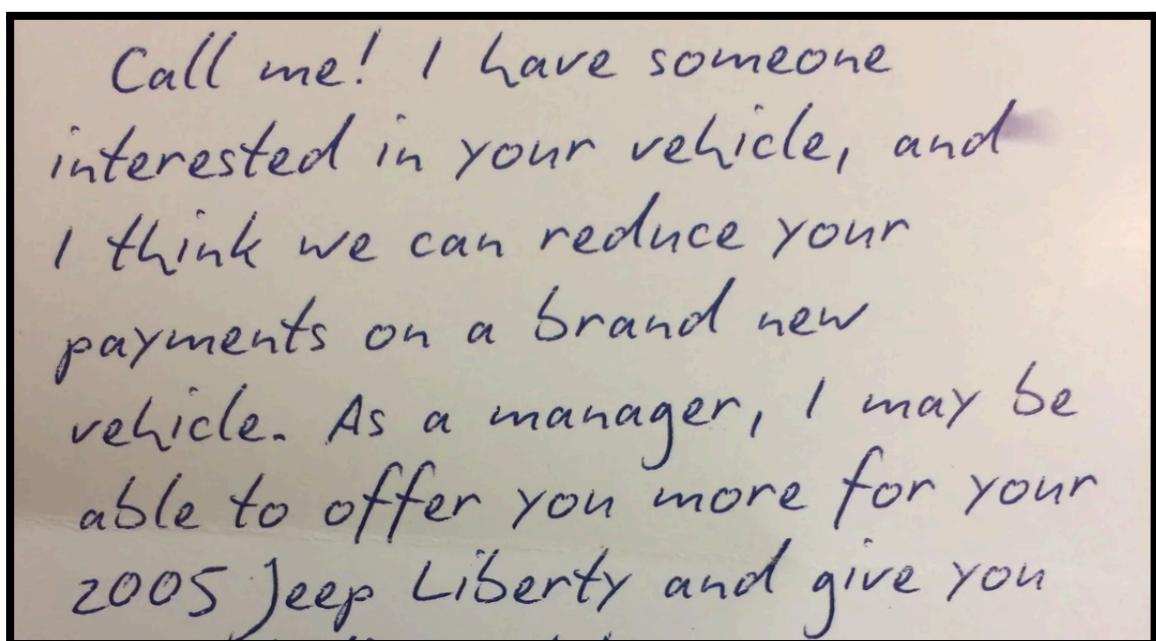
I. Introduction et Prétraitement

1. Introduction

La segmentation d'images joue un rôle crucial dans la reconnaissance de texte manuscrit. Cette méthode consiste à diviser une image en parties homogènes (ou "régions"), correspondant à chaque caractère manuscrit. Le prétraitement est une étape essentielle pour améliorer la qualité de la segmentation.

2. Étapes de prétraitement

- Chargement de l'image : Une image d'exemple contenant du texte manuscrit scanné a été chargée.



- **Conversion en niveaux de gris** : L'image a été convertie en niveaux de gris pour simplifier le traitement.

Call me! I have someone interested in your vehicle, and I think we can reduce your payments on a brand new vehicle. As a manager, I may be able to offer you more for your 2005 Jeep Liberty and give you

- **Réduction du bruit** : Un filtre médian a été appliqué pour éliminer les artefacts tout en préservant les bordures des caractères.

Image Filtré

Call me! I have someone interested in your vehicle, and I think we can reduce your payments on a brand new vehicle. As a manager, I may be able to offer you more for your 2005 Jeep Liberty and give you

- **Normalisation de l'image** : Une égalisation de l'histogramme a été utilisée pour améliorer le contraste, facilitant ainsi la distinction des caractères.

Image normalisé

Call me! I have someone interested in your vehicle, and I think we can reduce your payments on a brand new vehicle. As a manager, I may be able to offer you more for your 2005 Jeep Liberty and give you

- **Image binarisé** : est une image composée uniquement de deux niveaux de gris : le noir (généralement représenté par la valeur 0) et le blanc (représenté par la valeur maximale, souvent 255 dans les images 8 bits).

En d'autres termes, chaque pixel de l'image prend uniquement une valeur parmi ces deux options, ce qui simplifie considérablement le traitement de l'image.

Image Binarisée

Call me! I have someone interested in your vehicle, and I think we can reduce your payments on a brand new vehicle. As a manager, I may be able to offer you more for your 2005 Jeep Liberty and give you

II. Méthode de Segmentation : Croissance de Régions

1. Présentation de la méthode

L'algorithme de croissance de régions est une approche classique pour diviser une image en parties homogènes. Il commence par des "seeds" (pixels initiaux) et étend les régions selon un critère d'homogénéité.

Il fonctionne bien sur des images avec un contraste clair entre le fond et les caractères, le prétraitement améliore la qualité globale de la segmentation.

2. Implémentation de l'algorithme

Génération des SEEDS : Les pixels initiaux (seeds) ont été générés automatiquement en binarisant l'image normalisée et en extrayant les contours des composants connectés.



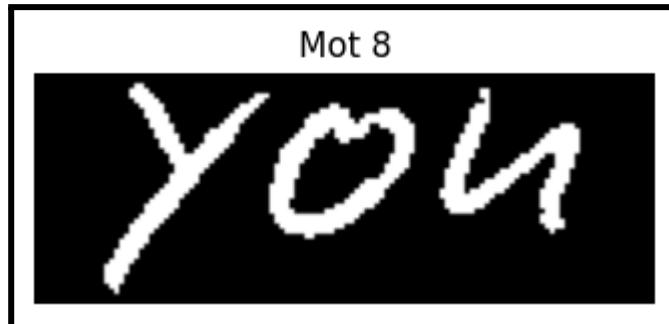
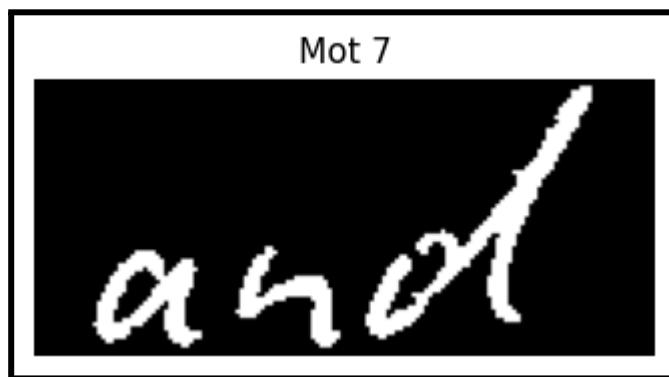
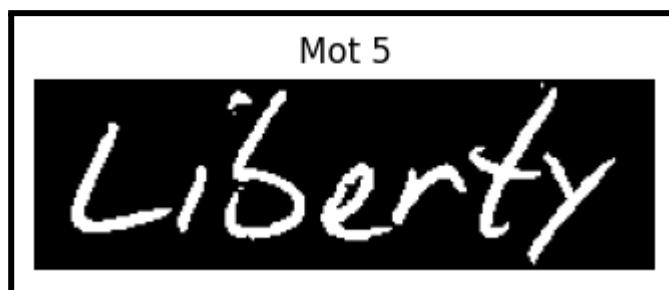
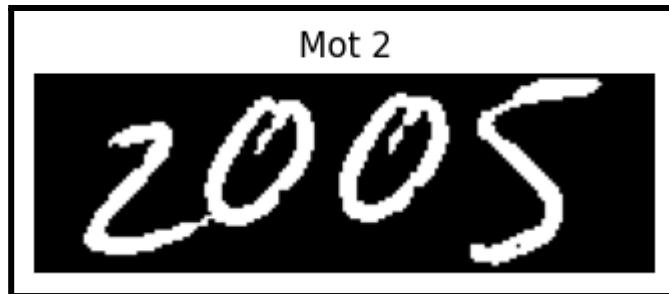
III. Post-Traitement

1. Nettoyage des régions

Suppression des petites régions : Les régions dont la taille était inférieure à un seuil prédéfini ont été supprimées pour éliminer le bruit.

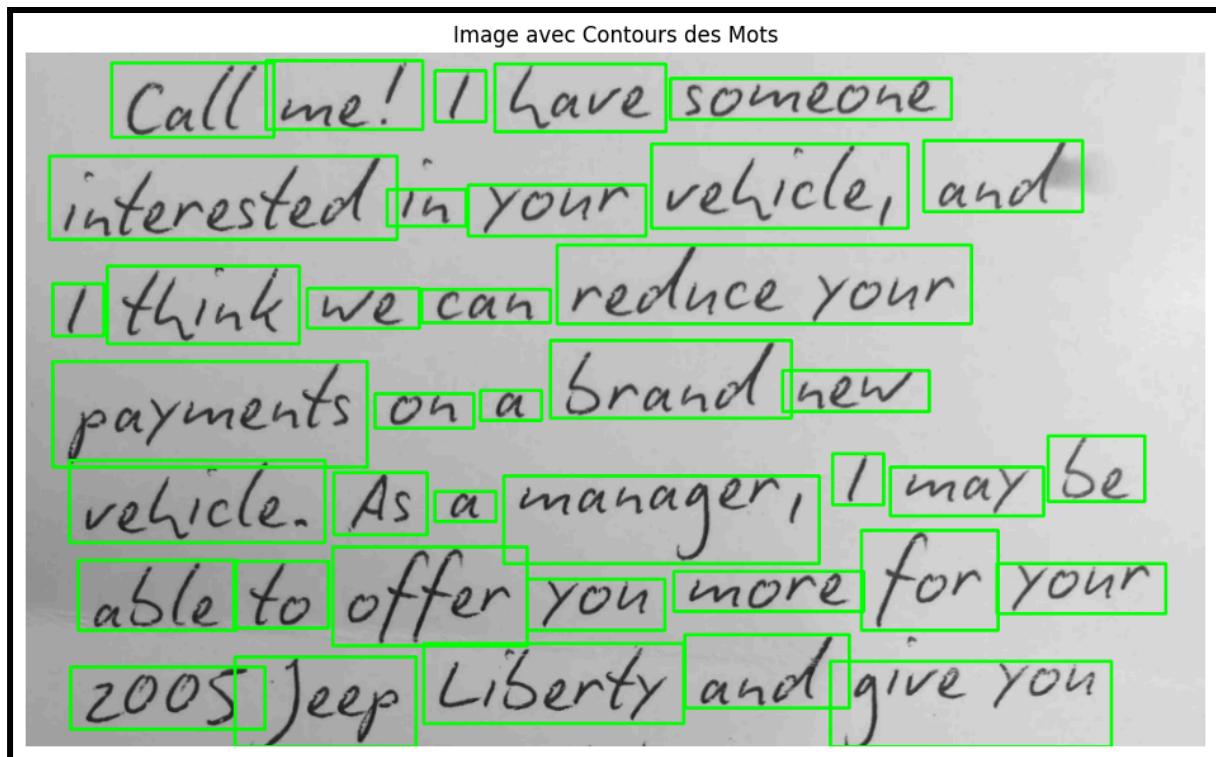
2. Extraction des sous-images

- **Boîtes englobantes** : Chaque caractère segmenté a été extrait et placé dans une boîte englobante ajustée à ses bordures.



3. Validation visuelle

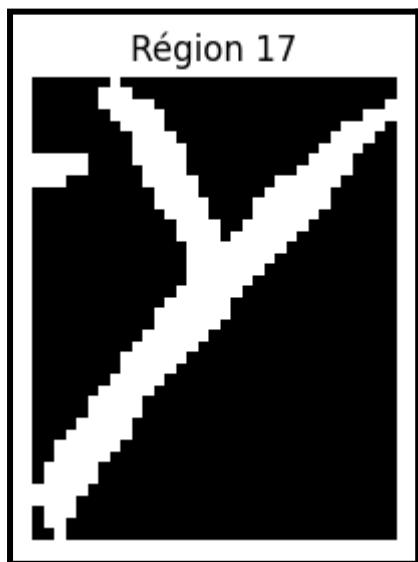
- Affichage des sous-images : affiché pour valider visuellement la qualité de la segmentation.



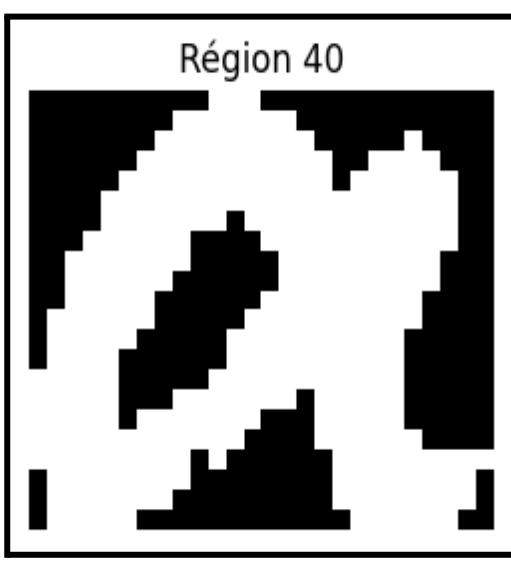
➤ Plus :

- 1) J'ai aussi réaliser un travail qui consiste à segmenter le texte lettre par lettre
vous allez trouver le travail dans "WordByWord.ipynb"

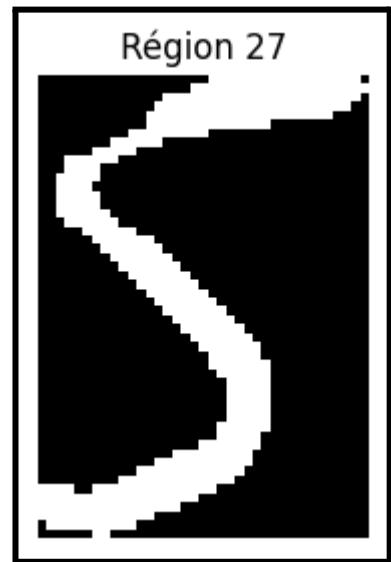
- Voici un affichage avec la même image utilisé dès le début :



“y”

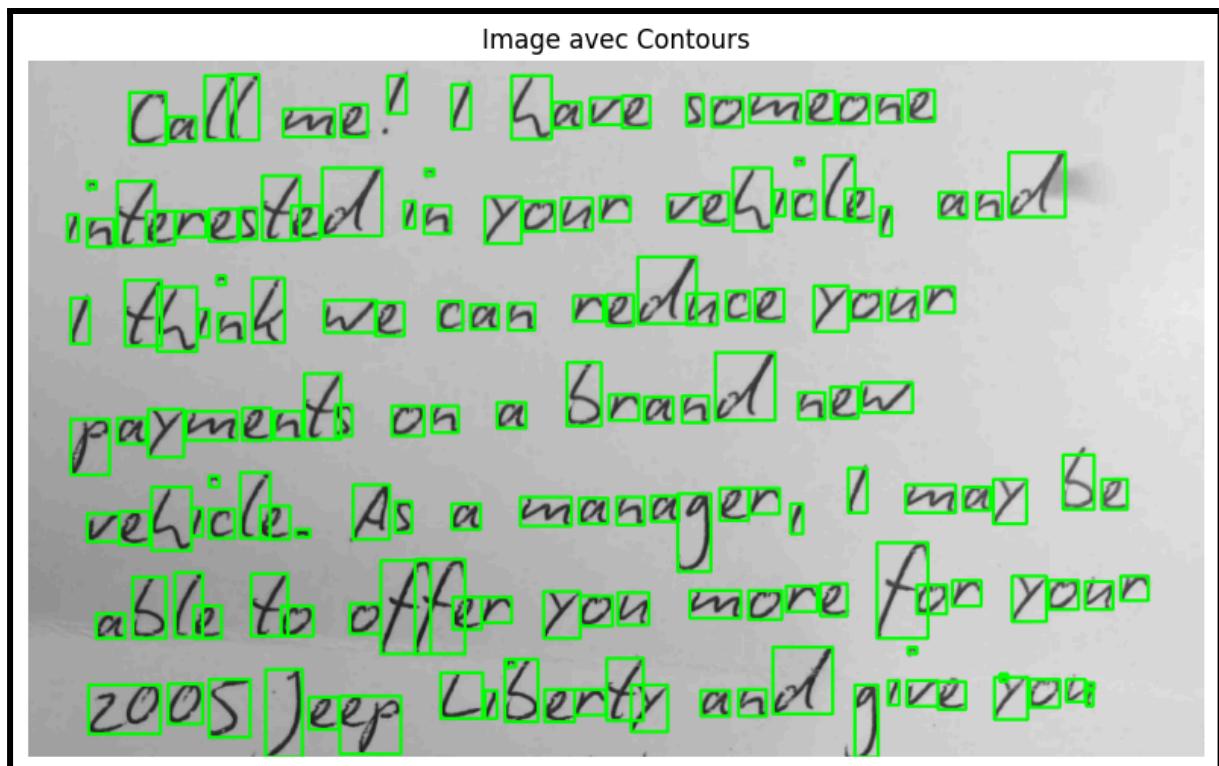


“a”



“5”

- Voici le résultat sur l'image :



2) J'ai aussi trouver une bibliothéque qui aide à segmenter les textes en arabe :



```
1 !pip install ArabicOcr
```



```
1 from ArabicOcr import arabicocr
2 image_path=r"Z:\Study\Mrsoualah\TP2-A-Remettre\img\quran.png"
3 out_image='Z:\Study\Mrsoualah\TP2-A-Remettre\img\out.png'
4 results=arabicocr.arabic_ocr(image_path,out_image)
5 print(results)
6
7 words=[]
8 for i in range(len(results)):
9     word=results[i][1]
10    words.append(word)
11 with open ('file.txt','w',encoding='utf-8') as myfile:
12     myfile.write(str(words))
```

Dans le cadre de l'analyse de textes en arabe, l'utilisation de la reconnaissance optique de caractères (OCR) devient essentielle pour extraire automatiquement des informations à partir d'images contenant des textes manuscrits ou imprimés.

Je sais que ce n'est pas exactement le travail demandé, mais voici ce que fait ce script. Il utilise la bibliothèque **ArabicOcr**, un outil conçu pour effectuer la reconnaissance optique de caractères (OCR) sur des textes écrits en arabe.

Le script prend en entrée une image (ici, un extrait du Coran) et en extrait les mots arabes. L'image est d'abord traitée par la fonction **arabic_ocr**, qui retourne les résultats sous forme de liste contenant des informations sur les mots reconnus. Chaque mot est extrait et ajouté à une liste, puis sauvegardé dans un fichier texte en utilisant l'encodage UTF-8 pour assurer une gestion correcte des caractères arabes. Cette approche permet de transformer des images contenant du texte arabe en données exploitables pour une analyse ou un traitement ultérieur.

- Voilà le résultat qu'on va trouver dans l'image OUT.png :

Using CPU. Note: This module is much faster with a GPU.
[INFO] OCR'ing input image...
[INFO] 0.2569: (نزل به آرُوح الْأَمِينِ:
[INFO] 0.1503: وَإِنَّهُ دَلِيلَ رَبِّ
على قلبك لِتَكُونَ مِنَ الْمُنذِرِينَ بِلسانِ عَرَبِيٍّ مُّبِينٍ
[INFO] 0.0894: هَذَا:
[INFO] 0.8469: هَذَا:
[INFO] 0.9958: هَذَا:
[INFO] 0.7159: الْفَضَائِلُ الْفَاحِرُ:
[INFO] 0.9871: هَذَا:
[INFO] 0.6577: اجْتَمَعَتْ:
[INFO] 0.5204: كَيْفَ:



IV. Conclusion

Ce travail pratique a permis d'explorer et d'implémenter une méthode de segmentation par croissance de régions, adaptée à l'isolation des caractères manuscrits dans une image scannée. Les principales réalisations de ce projet incluent :

- Une phase de prétraitement rigoureuse comprenant la conversion en niveaux de gris, la réduction de bruit et la normalisation des contrastes, garantissant des conditions optimales pour la segmentation.
- La mise en œuvre de l'algorithme de croissance de régions, avec une sélection efficace des seeds et une gestion adéquate des collisions entre régions adjacentes, permettant une séparation précise des caractères.
- Une étape de post-traitement réussie, incluant la suppression des petites régions indésirables et l'extraction de sous-images nettes pour chaque caractère, préparées pour des applications ultérieures comme la reconnaissance de texte.

Cette approche démontre la pertinence des techniques de segmentation par région dans le cadre de la reconnaissance de texte manuscrit et pose une base solide pour des améliorations futures, notamment l'intégration de techniques modernes pour renforcer la robustesse et la précision du processus.