Projet Module NLP : Reconnaissance De Texte (OCR) Avec Tesseract Et Opencv

Les Membres du groupe : Abdelmoutalib Khezane

Oussama Rchouk

Dans ce Projet, nous allons voir comment utiliser Tesseract pour reconnaître le texte d'une image. Tesseract est l'OCR (reconnaissance optique de caractères) le plus populaire, il est open source et il est développé par google depuis 2006.

Dans ce Mini Projet spécifique, nous verrons :

1. Comment installer Tesseract sur (Windows)
2. Lire du texte à partir d'une image
3. Ajustez tesseract pour améliorer la reconnaissance de texte

1. Installez Tesseract pour travailler avec Python et Opencv

Avant de procéder à l'installation de Tesseract, il est important de comprendre tous les outils que nous allons utiliser et le but de chacun d'eux.

Voici les outils dont nous avons besoin :

- Python et Opencv : nous allons utiliser le langage de programmation python et Opencv pour charger l'image, et faire un prétraitement d'image (par exemple supprimer les zones où il n'y a pas de texte, supprimer du bruit, appliquer un filtre d'image pour rendre le texte plus lisible) .
- Tesseract : c'est le moteur OCR, donc le cœur de la véritable reconnaissance de texte. Il prend l'image et en retour nous donne le texte.
- Pytesseract : c'est la liaison tesseract pour python. Avec cette bibliothèque, nous pouvons utiliser le moteur tesseract avec python avec seulement quelques lignes de code.

1.1 Installer Python et Opencv

Tout d'abord, assurons-nous que python et Opencv sont installés. Sinon, vous pouvez installer Opencv et Python sous Windows.

1.2 Installer Tesseract

Windows

Ensuite, c'est le moment d'installer Tesseract.

Si vous avez Windows, allez sur cette page https://github.com/UB-Mannheim/tesseract/wiki , téléchargez et installez tesseract 64 bits.

```
C:\Users\Master-Admin>python -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/ca/31/b88ef447d595963c01060998cb329251648acf4a067721b0452c45527eb8/pip-21.2.4-py3-none-any.whl (1.6MB)
     |                                | 1.6MB 819kB/s
Installing collected packages: pip
  Found existing installation: pip 19.2.3
    Uninstalling pip-19.2.3:
      Successfully uninstalled pip-19.2.3
Successfully installed pip-21.2.4

C:\Users\Master-Admin>pip3 install pytesseract
Collecting pytesseract
  Using cached pytesseract-0.3.8.tar.gz (14 kB)
```

Vérification installation Tesseract :

```
C:\Users\Master-Admin>pip3 install pytesseract
Requirement already satisfied: pytesseract in c:\users\master-admin\appdata\local\programs\python\python38\lib\site-packages (0.3.7)
Requirement already satisfied: Pillow in c:\users\master-admin\appdata\local\programs\python\python38\lib\site-packages (from pytesseract) (8.3.1)
```

1.3 Installer PyTesseract

Pytesseract est une bibliothèque indispensable si nous voulons utiliser tesseract avec Python. Il peut être facilement installé comme n'importe quelle autre bibliothèque python à l'aide de la commande pip.

Copiez donc les commandes suivantes sur votre terminal.

pip installer pytesseract

pip3 installer pytesseract

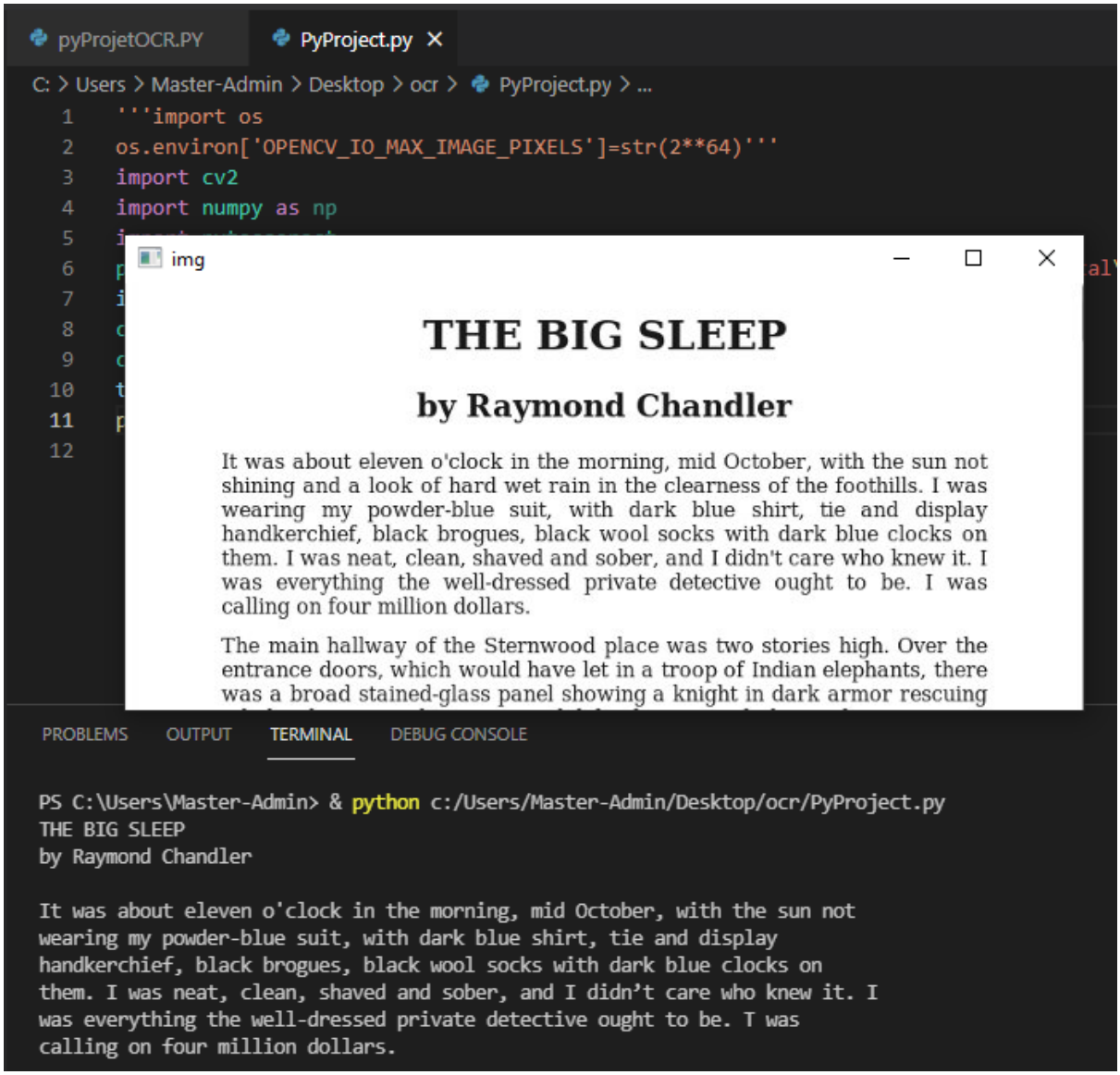
install opencv :

```
  Downloading pytesseract-0.3.7.tar.gz (13 kB)
Collecting Pillow
  Downloading Pillow-8.3.1-1-cp38-cp38-win_amd64.whl (3.2 MB)
     |                                | 3.2 MB 1.1 MB/s
Using legacy 'setup.py install' for pytesseract, since package 'wheel' is not installed.
Installing collected packages: Pillow, pytesseract
    Running setup.py install for pytesseract ... done
Successfully installed Pillow-8.3.1 pytesseract-0.3.7

C:\Users\Master-Admin>pip install pytesseract
Requirement already satisfied: pytesseract in c:\users\master-admin\appdata\local\programs\python\python38\lib\site-packages (0.3.7)
Requirement already satisfied: Pillow in c:\users\master-admin\appdata\local\programs\python\python38\lib\site-packages (from pytesseract) (8.3.1)

C:\Users\Master-Admin>pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.5.3.56-cp38-cp38-win_amd64.whl (34.9 MB)
     |                                | 34.9 MB 47 kB/s
Collecting numpy>=1.17.3
  Downloading numpy-1.21.2-cp38-cp38-win_amd64.whl (14.0 MB)
     |                                | 14.0 MB 107 kB/s
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.21.2 opencv-python-4.5.3.56

C:\Users\Master-Admin>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.4.3-cp38-cp38-win_amd64.whl (7.1 MB)
     |                                | 7.1 MB 57 kB/s
```

2. Lire du texte à partir d'une image

Une fois l'installation terminée, il est temps pour nous d'essayer Tesseract pour lire le texte à partir d'une image.

J'ai cette page numérisée d'un livre (« The big sleep»). Essayons d'extraire le texte de cette image.



Tout d'abord nous commençons par importer les librairies Opencv, Numpy et Pytesseract.

Ensuite, à la ligne 5, vous devez indiquer où le moteur Tesseract est installé. La configuration ci-dessous convient si vous utilisez Windows, au lieu de cela si vous êtes sur Mac ou Linux, vous devriez vous référer à la documentation officielle pour voir comment la configurer.

```
'''import os
os.environ['OPENCV_IO_MAX_IMAGE_PIXELS']=str(2**64)'''
import cv2
import numpy as np
import pytesseract
pytesseract.pytesseract.tesseract_cmd = r"C:\\Users\\Master-
Admin\\AppData\\Local\\Programs\\Tesseract-OCR\\tesseract.exe"
img = cv2.imread("C:\\Users\\Master-Admin\\Desktop\\ocr\\bigsleep.jpg",0)
```

Ensuite, nous chargeons simplement l'image et extrayons le texte à l'aide de Pytesseract.

```
cv2.imshow("img",img)
cv2.waitKey(0)
text = pytesseract.image_to_string(img)
print(text)
```

Et on obtient la sortie :

THE BIG SLEEP

by Raymond Chandler

It was about eleven o'clock in the morning, mid October, with the sun not

shining and a look of hard wet rain in the clearness of the foothills. I was

wearing my powder-blue suit, with dark blue shirt, tie and display

handkerchief, black brogues, black wool socks with dark blue clocks on

them. I was neat, clean, shaved and sober, and I didn’t care who knew it. I

was everything the well-dressed private detective ought to be. T was

calling on four million dollars.

‘The main hallway of the Sternwood place was two stories high. Over the

entrance doors, which would have let in a troop of Indian elephants, there

3. Ajustez tesseract pour améliorer la reconnaissance de texte

Dans la section 2, nous avons vu à quel point il est facile d'exécuter Tesseract en utilisant python, et le résultat était vraiment bon car le moteur fonctionnait très bien et la reconnaissance de texte était presque parfaite.

Nous devons dire aussi que ce n'était pas un gros défi, car le texte était très clair.

Maintenant, nous allons rendre la reconnaissance de texte plus difficile, en donnant à Tesseract une image (au lieu de numériser une page de livre) , où l'orientation du texte n'est pas exactement horizontale mais il y a une pente, où l'éclair change sur différentes parties de la page et où il y a des éléments qui ne sont pas du texte et ne devraient pas être là.

Comment Tesseract se comportera-t-il avec de telles images ?

En utilisant le code pyton que nous avons utilisé pour l'image précédente, nous obtenons par défaut un résultat terrible . Le texte n'est presque pas reconnu du tout.

Nous pouvons faire ce qu'on appelle le « prétraitement d'image ». Ce sont des opérations que nous allons faire avant d'essayer de détecter le texte, en améliorant au maximum l'image pour rendre le texte plus clair.

Il y a beaucoup d'opérations que nous pouvons faire pour améliorer l'image avant de la donner à l'OCR. Ci-dessous, je vais juste vous donner quelques idées.

Convertir l'image en niveaux de gris : nous n'avons besoin que du texte, nous ne nous soucions pas des couleurs. Le texte est plus simple à identifier sur une image en niveaux de gris.

Changer la taille de l'image : j'ai vu que tesseract ne fonctionnait pas aussi bien quand les images sont trop grandes, donc on peut utiliser comme résolution maximale, celle où le texte est clair pour être lu à nos yeux.

Suppression du bruit : à quoi ressemble un texte clair, si nous l'agrandissons, l'image, pixel par pixel, n'est peut-être pas aussi claire mais il peut y avoir du bruit. On peut donc par exemple flouter l'image.

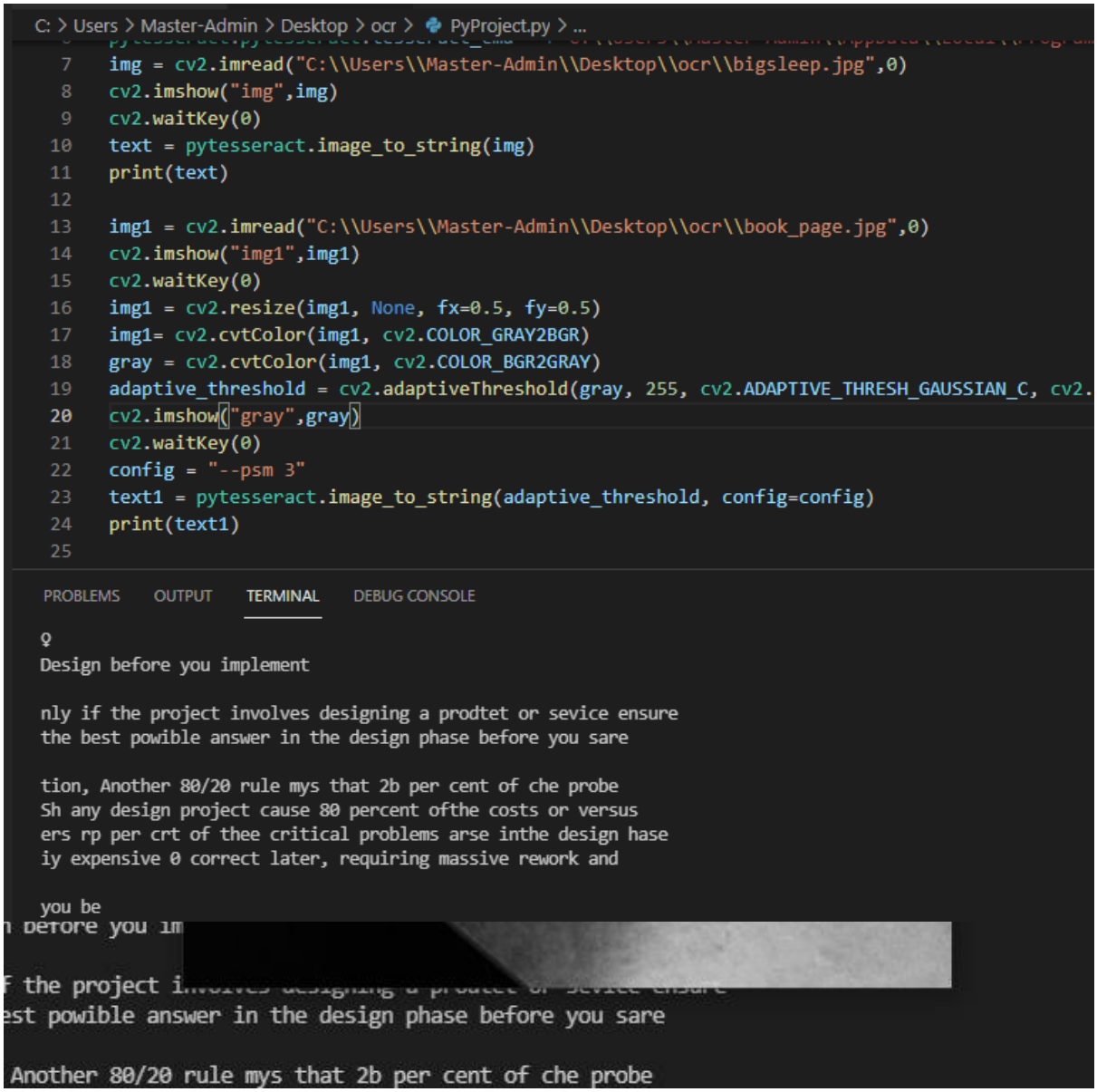
Convertir l'image en noir et blanc : en convertissant l'image en noir et blanc, nous définissons l'arrière-plan en blanc et le texte en noir. De cette façon, nous résolvons le problème de reconnaître le texte lorsque l'éclair change constamment sur l'image.

Dans cette image spécifique, je vais en appliquer quelques-uns. Premièrement, je redimensionnerai l'image pour la rendre plus petite, puis je la convertirai en niveaux de gris et enfin je créerai un seuil, en utilisant la méthode du seuil adaptatif pour la convertir en noir et blanc.

Voici le code utilisé pour cela :

```
img1 = cv2.imread("C:\\Users\\Master-Admin\\Desktop\\ocr\\book_page.jpg",0)
cv2.imshow("img1",img1)
cv2.waitKey(0)
img1 = cv2.resize(img1, None, fx=0.5, fy=0.5)
img1= cv2.cvtColor(img1, cv2.COLOR_GRAY2BGR)
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
adaptive_threshold = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_GAUS
SIAN_C, cv2.THRESH_BINARY, 85, 11)
cv2.imshow("gray",gray)
cv2.waitKey(0)
```

Et voilà à quoi ressemble l'image après les opérations de prétraitement :



Résultat du traitement

Comme vous le voyez après le prétraitement, le texte est beaucoup plus facile à lire, et cela fait une énorme différence pour le moteur OCR.

En plus des opérations de prétraitement d'image, nous pouvons régler Tesseract.

Tesseract dispose de 10 modes de segmentation de page (PSM) différents que nous pouvons sélectionner manuellement :

0 = Orientation and script detection (OSD) only.

1 = Automatic page segmentation with OSD.

2 = Automatic page segmentation, but no OSD, or OCR

3 = Fully automatic page segmentation, but no OSD. (Default)

4 = Assume a single column of text of variable sizes.

5 = Assume a single uniform block of vertically aligned text.

6 = Assume a single uniform block of text.

7 = Treat the image as a single text line.

8 = Treat the image as a single word.

9 = Treat the image as a single word in a circle.

10 = Treat the image as a single character.

Nous pouvons choisir le mode à utiliser en ajoutant une valeur de configuration au code python.

Après « –psm », vous devez mettre une valeur de 0 à 10, en fonction de la segmentation de page que vous souhaitez utiliser.

```
config = "--psm 3"
text1 = pytesseract.image_to_string(adaptive_threshold, config=config)
print(text1)
```