

**Travaux pratiques RPE N°1
Reconnaissance Automatique Ecriture**

Ce TP porte sur la reconnaissance automatique de caractères arabes isolés et les chiffres de 0 à 9.

- Il faut télécharger et installer :

-installer python

- installer les librairies suivantes : utiliser la commande pip install

Numpy, pandas, opencv, matplotlib, pandas , sklearn , random

Voici un exemple en python avec la librairie opencv

- 1- Il charge une image contenant des caractères arabes et la convertit en niveau de gris puis fait un prétraitement : il détermine les contours de chaque caractère.
- 2- Il est demandé d'écrire (manuscrite) sur une feuille blanche les sourates fatiha, ikhlas, al falak, al nass (avec un stylo noir) comme l'exemple indiqué au cours.
- 3- Il est demandé d'écrire (manuscrite) sur trois feuilles blanches les lettres de la langue arabes (avec un stylo noir) chacune sur une ligne comme l'exemple suivant et la scanner ou la photographier avec votre portable et la sauvegarder en png :

ل ل ل ل ل ل ل ل ل ل ل ل ل ل
ب ب ب ب ب ب ب ب ب ب ب ب ب ب

Et la meme chose avec toutes les lettres de l'arabe

Travail a rendre

Envoyer moi vos images scannées le 17/06/2021 à ce mail (fichiers zippes) :
haytdjellali@yahoo.fr

Le premier jour après les examens normalement le (18/06/2021) : validation du TP à 10h30 .

Les binômes sont autorisés (envoyer la liste des binôme avant le 08 juin par le délégué (finalisée : plus de changement).



Figure 1 : lettre arabes

- 4- Ecrire ce programme sous python et exécutez le avec une image contenant cinq lettres arabes séparés scanné par vous (portable avec appareil photo) après l'avoir écrit sur une feuille.

Voici les images exemple et le résultat :

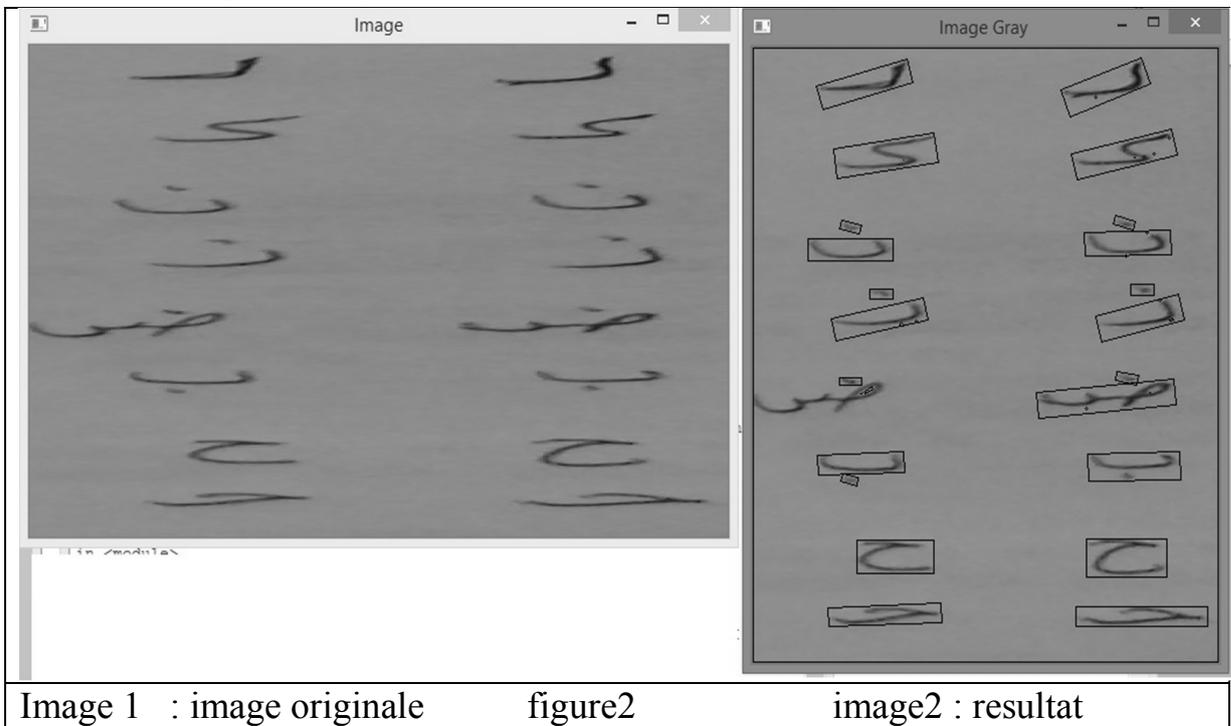


Image 1 : image originale

figure2

image2 : resultat

```
tplati_char_arabic.py - C:\PythonCode2019\digitRecognition_dh\tplati_char_arabic.py (3.7.1)
File Edit Format Run Options Window Help
#The function cv2.moments() gives a dictionary of all moment values ca
import cv2
import numpy as np
from matplotlib import pyplot as plt
import os

#liste des fichiers à traiter
listfile=["mots_ar1.png",|"sourate1.png", "sourate2.png", "mots_ar1.png",

path= "C:/PythonCode2019/digitRecognition_dh/baseImageCaractere"

# donner ou se trouve vos images à traiter
path=input("Give the Complete Path please: ")

# le dossier ou il va mettre le resultat de la segmentation de l'image
pathres="C:/PythonCode2019/digitRecognition_dh/baseImageCaractere"

#C:\PythonCode2019\digitRecognition_dh\baseImageCaractere

#pathres="C:/PythonCode2019/digitRecognition_dh/iati/AhctiNabil/"
#C:/PythonCode2019/digitRecognition_dh/iati/AhctiNabil/ahcti_part1.j

scriptor=input("Scriptor Name Please:")
# creer le sous dossier indiqué
pathsave=pathres+"/"+scriptor
```

Figure3 : code python extraction de contours des caracteres

Voici un exemple d'image mais faites la sans les traits des colonnes et lignes svp.

TP code a analyser :

```
#The function cv2.moments() gives a dictionary of all moment values calculated.
```

```
import cv2
```

```
import numpy as np
```

```
from matplotlib import pyplot as plt
```

```
import os
```

```
#liste des fichiers à traiter
```

```
listfile=["mots_ar1.png","sourate1.png","sourate2.png","mots_ar1.png","ahcti_part1.png","ahcti_part2.png","aouag1_part2.png","char9_1.png","arab1.png","char2.png","char4.png","char5.png","char7.png","char20_1.png","",""]
```

```
path= "C:/PythonCode2019/digitRecognition_dh/baseImageCaractere"
```

```
# donner ou se trouve vos images à traiter
```

```
path=input("Give the Complete Path please: ")
```

```
# le dossier ou il va mettre le resultat de la segmentation de l'image
```

```
pathres="C:/PythonCode2019/digitRecognition_dh/baseImageCaractere"
```

```
scriptor=input("Scriptor Name Please:")
```

```
# creer le sous dossier indiqué
```

```
pathsave=pathres+"/"+scriptor
```

```
os.makedirs(pathsave)
```

```
name = listfile[0];
```

```
file = pathres + "/" + name # or path+name
```

```
print("le fichier = \n",file )
```

```
img = cv2.imread(file)
```

```
print("file: ... ", file)
```

```
cv2.imshow('Image', img)
```

```
#convertir l'image : img en niveau de gris
```

```
imgray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
#redimmnsionner image imgray en largeur 400, hauteur 500
```

```

imres = cv2.resize(imgray,(400,400), interpolation= cv2.INTER_AREA)
cv2.imshow('Image', imres)

# calcul du seuil
ret,thres = cv2.threshold(imres,120,255,0)#(imres, 127,255,0)

#trouver les contours
contours,hierarchy = cv2.findContours(thres, 1, 2)#thres,1,2
nbCont= len(contours); print("Nombres de Contours " + str(nbCont))
cnt = contours[0] ; M = cv2.moments(cnt); print ("M= " ,M)

#-----

img2 = cv2.imread(file)
gray2 = cv2.cvtColor(img2,cv2.COLOR_BGR2GRAY)
ret, thresh2 =
cv2.threshold(gray2,0,255,cv2.THRESH_BINARY_INV+cv2.THRESH_OTSU)

#-----
cp=1;t=0; a=12;b=4; #a=b
for i in range(0,nbCont):
    x,y,w,h = cv2.boundingRect(cnt)
    if w>8 and h>8:
        img3 = cv2.rectangle(imres,(x,y),(x+w,y+h),(0,255,0),2)
        thres[y:y+h, x:x+w]
        char= thres[y:y+h, x:x+w]
        print("taille du caractere = ", char.shape[0]," ",char.shape[1])
        if char.shape[0]>3 and char.shape[1]>3:
            X=x-b; Y=y-b
            #cv2.imwrite("D"+str(cp)+".jpeg",thres[Y:Y+h+5, X:X+w+5]) bien
            pat= thres[Y:Y+h+a, X:X+w+a]
            if pat.shape[0]>8 and pat.shape[1]>8: #8 avant
                print("save image ")
                cv2.imwrite(pathsave+"/"+scriptor+str(cp)+".jpeg",thres[Y:Y+h+a,
X:X+w+a])
            #cv2.imshow('Image', img3)
            cp=cp+1
            print("cp= ",cp)

```

```
cnt = contours[i]
print("New Contour = ",str(i))
if i==nbCont:
    print("Quit....")
    break
```

```
cv2.imshow('Image', img2)
#cv2.imshow('Image', img3)
#cv2.waitKey(0)
cv2.destroyAllWindows()
```