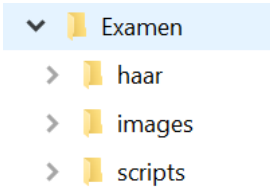
Université Chouaib Doukkali **AU : 2020/2021**
Faculté des Sciences
Département d'Informatique
El Jadida

OpenCV : Application 1

Dans ce problème, on considère l'arborescence suivante :



- Examen : dossier racine de votre travail
- scripts : c'est ici que vous allez enregistrer vos scripts
- images : ce dossier contient les images
- haar : ce dossier contient les fichiers haarcascade

Exercice 1

1. Afficher le contenu du dossier courant et du dossier "images"
2. En utilisant Numpy, générer trois images (trois matrices) de 200x300 pixels, l'une blanche, l'autre noire et la troisième grise.
3. Afficher la resolution des images
4. Afficher les images dans trois fenêtres différentes
5. Afficher les images dans une seule fenêtre alignées verticalement
6. Afficher les images dans une seule fenêtre alignées horizontalement

Exercice 2

1. A partir d'une image couleur enregistrée sur le disque (dossier Examen/Images), afficher :
   a. La matrice correspondante
   b. Les dimensions de l'image
   c. La résolution (nombre de pixels) de l'image
   d. Les trois couleurs de base du pixel situé à la position 0,0
2. Dans l'image de l'exercice précédent, modifier une zone de votre choix avec une couleur bleue
3. Dans l'image précédente, tracer un cercle, une ligne et un rectangle

Exercice 3

1. A partir d'une image contenant un groupe de personnes, encadrer les visages d'un rectangle vert
2. Enregistrer ensuite ces visages en couleur sur disque, sous forme de face1.jpg, face2.jpg, …
3. Afficher tous les visages enregistrés dans la question précédente

Correction proposée

```
import cv2
import numpy as np
import os
def fermer():
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

Exercice 1

```
#Afficher le contenu du dossier "images"
for file in os.listdir("./"):
    print(file)
#En utilisant Numpy, générer trois images (trois matrices) de 200x300 pixels,
#l'une blanche, l'autre noire et la troisième grise.
blanche = np.ones((200,300), dtype=np.uint8)*255
grise = np.ones((200,300), dtype=np.uint8)*127
noire = np.ones((200,300), dtype=np.uint8)*0
#Afficher la resolution des images
print("Résolution des trois images:",blanche.shape, grise.shape, noire.shape)
#Afficher les images dans trois fenêtres différentes
cv2.imshow("Image blanche", blanche)
cv2.imshow("Image grise", grise)
cv2.imshow("Image noire", noire)
fermer()
#Afficher les images dans une seule fenêtre alignées verticalement
verticale = np.vstack((blanche, grise, noire))
cv2.imshow("Alignement verticale", verticale)
fermer()
#Afficher les images dans une seule fenêtre alignées horizontalement
horizontale = np.hstack((blanche, grise, noire))
cv2.imshow("Alignement horizontale", horizontale)
fermer()
```

Exercice 2

```
#Question 1
image = cv2.imread("../images/real.jpg")
print("La matrice correspondante :", image)
print("Les dimensions :", image.shape)
print("La résolution :", image.size)
print("Les couleurs du pixel situé à la position 0,0 :", image[0,0])
#Question 2
image = cv2.imread("../images/real.jpg")
cv2.imshow("Image initiale", image)
image[:100,:100] = (255,0,0)
cv2.imshow("Image modifiée", image)
fermer()
# Question 3
image = cv2.imread("../images/real.jpg")
hauteur = image.shape[0]
```

```
largeur = image.shape[1]
cv2.imshow("Image initiale", image)
cv2.line(image,(0,0),(largeur, hauteur),(255,0,0),5)
cv2.rectangle(image,(100,100),(largeur-10, hauteur-10),(0,255,0),5)
cv2.circle(image,(largeur//2, hauteur//2),100, (0,0,255),5)
cv2.imshow("Image modifiée", image)
fermer()
```

Exercice 3

```
# Question 1
equipe = cv2.imread("../images/real.jpg")
haar = cv2.CascadeClassifier("../haar/frontalface.xml")
faces = haar.detectMultiScale(equipe)
for (x,y,w,h) in faces:
    cv2.rectangle(equipe,(x,y),(x+w, y+h),(0,255,0),1)
cv2.imshow("Equipe", equipe)
fermer()
# Question 2
equipe = cv2.imread("../images/real.jpg")
haar = cv2.CascadeClassifier("../haar/frontalface.xml")
faces = haar.detectMultiScale(equipe)
i = 1
for (x,y,w,h) in faces:
    cv2.rectangle(equipe,(x,y),(x+w, y+h),(0,255,0),1)
    face = equipe[y:y+h, x:x+w]
    cv2.imwrite("../images/face"+str(i)+".png", face)
    i+=1
fermer()
# Question 3
i=1
for image in os.listdir("../images/"):
    if image.startswith("face"):
        #print(image)
        cv2.imshow("face"+str(i),face)
        i+=1
fermer()
```