

RAPPORT DE SEANCE N°2

Le 08/12/2023

Aimée BOUKAMBA

ROBO3

Lors de cette seconde séance, j'ai décidé de m'intéresser au programme informatique principal de mon robot. Il était question de s'attaquer à la fonction de détection faciale mais également de reconnaissance faciale à l'aide de Python et de open cv.

J'ai écrit deux codes informatiques. L'un que j'ai testé avec la caméra Logitech détectait en temps réel les visages et indiquait dans un cadre entourant le potentiel visage le pourcentage de certitude du programme.

```
test.py -> _
1  import cvzone
2  from cvzone.FaceDetectionModule import FaceDetector
3  import cv2
4  import os
5  import numpy as np
6
7
8  # Initialize the webcam
9  # '2' means the third camera connected to the computer, usually 0 refers to the built-in webcam
10 cap = cv2.VideoCapture(2, cv2.CAP_DSHOW)
11 fourcc = cv2.VideoWriter_fourcc(*'XVID') # codec
12
13
14 fps = cap.get(cv2.CAP_PROP_FPS)
15
16
17
18
19
20 # Initialize the FaceDetector object
21 # minDetectionCon: Minimum detection confidence threshold
22 # modelSelection: 0 for short-range detection (2 meters), 1 for long-range detection (5 meters)
23 detector = FaceDetector(minDetectionCon=0.5, modelSelection=1)
24
25 # Run the loop to continually get frames from the webcam
26 while True:
27     # Read the current frame from the webcam
28     # success: Boolean, whether the frame was successfully grabbed
29     # img: the captured frame
30
31     success, img = cap.read()
32
```

```

34     # Detect faces in the image
35     # img: Updated image
36     # bboxes: List of bounding boxes around detected faces
37     img, bboxes = detector.findFaces(img, draw=False)
38
39     # Check if any face is detected
40     if bboxes:
41         # Loop through each bounding box
42         for bbox in bboxes:
43             # bbox contains 'id', 'bbox', 'score', 'center'
44
45             # ---- Get Data ---- #
46             center = bbox["center"]
47             x, y, w, h = bbox["bbox"]
48             score = int(bbox['score'][0] * 100)
49
50             # ---- Draw Data ---- #
51             cv2.circle(img, center, 5, (255, 0, 255), cv2.FILLED)
52             cvzone.putTextRect(img, f'{score}%', (x, y - 10))
53             cvzone.cornerRect(img, (x, y, w, h))
54
55     # Display the image in a window named 'Image'
56     cv2.imshow("Image", img)
57     # Wait for 1 millisecond, and keep the window open
58     if cv2.waitKey(1) == ord('q'):
59         break
60
61     # Assuming 'frame' is the frame you want to save
62     # Write out the frame
63
64     # Release the Videowriter when finished
65
66     cap.release()
67     cv2.destroyAllWindows()

```

L'autre fait simplement une comparaison entre deux images et renvoie un booléen si les deux visages appartiennent à la même personne.

TESTS

debbouze.jpg

debbouze2.jpg

messi.jpg

test1.py

test2.py1

test3.py2

test42

test2.py > ...

1 #Appel des librairies

2 import cv2

3 import face_recognition

4

5 # Face encoding

6 img = cv2.imread("debbouze.jpg")

7 rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

8 img_encoding = face_recognition.face_encodings(rgb_img)[0]

9

10

11 img2 = cv2.imread("debbouze2.jpg")

12 rgb_img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)

13 img_encoding2 = face_recognition.face_encodings(rgb_img2)[0]

14

15

16 result = face_recognition.compare_faces([img_encoding], img_encoding2)

17 print("Result: ", result)

18

19 # Encode faces from a folder

20 sfr = SimpleFacerec()

21 sfr.load_encoding_images("images/")

Dans la deuxième partie de la séance, nous avons dessiné sur fusion 360 et découpé rau FabLab la base du robot.

