

RAPPORT DU HACKATHON

Systeme de surveillance d'examen en Ligne



G12, Lounis OUMOHAND, Nabil BENSAFIA

Aomar OSMANI- CLASSE DE Master 2 PLS+EID

Année Universitaire 2021/2022

Cet événement de e-sport avec 2 joueurs : Lounis et Nabil , nous a permis de trouver des idées originales.

Nous avons commencé par la présentation des pitches et les défis de chacun pour ensuite se lancer et réaliser le maximum en essayant toujours de trouver des idées brillantes et innovantes.

I-Fonctionnalités:

Au début nous sommes tenter par :

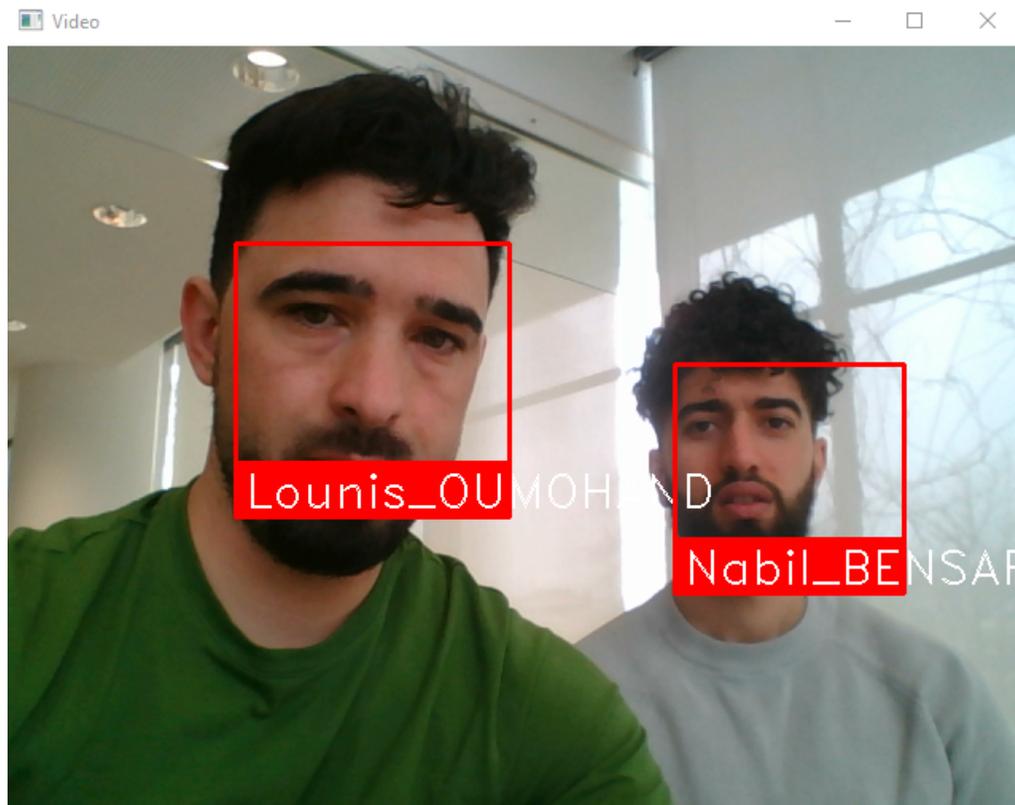
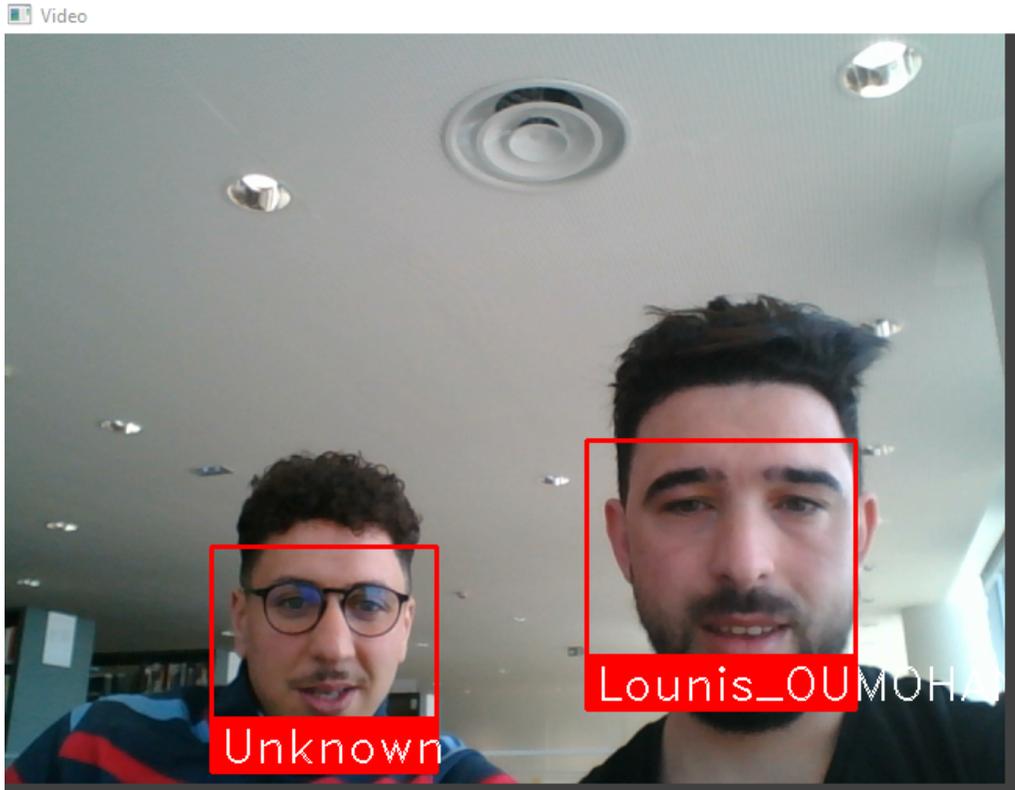
1. Reconnaissance faciale de l'étudiant avec son nom et prénom

Le modèle de base est le suivant : nous avons pris plusieurs photos.

nous avons organisés ces photos dans un dépôt pour qu'elles soient utilisables par un algorithme d'apprentissage Photo

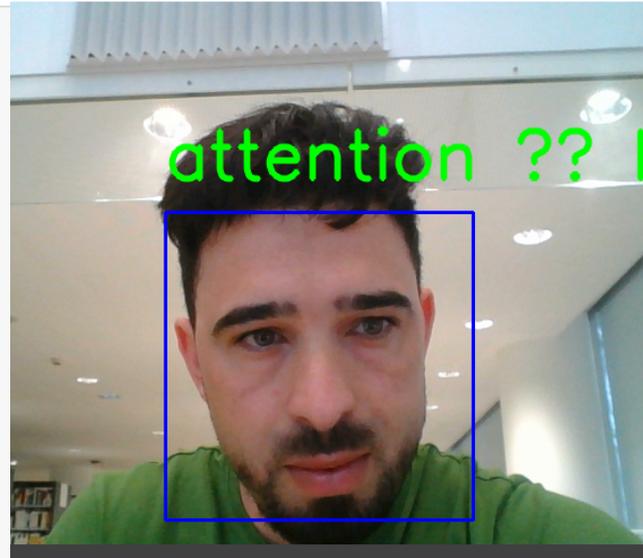


2-Vidéo temps réel



Etape 2 detection lorsque il triche en parlant :

```
1 import numpy as np
2 import cv2
3
4 faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
5 smileCascade = cv2.CascadeClassifier('haarcascade_smile.xml')
6
7 cap = cv2.VideoCapture(0)
8
9 while True:
10     ret, img = cap.read()
11     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
12     faces = faceCascade.detectMultiScale(
13         gray,
14         scaleFactor=1.3,
15         minNeighbors=5,
16         minSize=(30, 30)
17     )
18
19     for (x,y,w,h) in faces:
20         cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
21         roi_gray = gray[y:y+h, x:x+w]
22
23         smile = smileCascade.detectMultiScale(
24             roi_gray,
25             scaleFactor= 1.5,
26             minNeighbors=15,
27             minSize=(25, 25),
28         )
29
30         for i in smile:
31             if len(smile)>1:
32                 cv2.putText(img,"attention à la triche ",(x,y-30),cv2.FONT_HERSHEY_SIMPLEX,
33                     2,(0,255,0),3,cv2.LINE_AA)
34
35     cv2.imshow('video', img)
36     k = cv2.waitKey(30) & 0xff
37     if k == 27: # press 'ESC' to quit
38         break
39
40 cap.release()
41 cv2.destroyAllWindows()
```



MATÉRIELS

1. Open Cv, dlib
2. Keras
3. Tensor Flow
4. caméra de l'ordinateur

Environnements

Jupyter Notebook avec Anacondaquipe

Membre	Action
Lounis OUMOHAND Nabil BENSALIA	Reconnaissance faciale Identification de l'étudiant avec son nom et prenom en photo + temps réel reconnaissance d'un visage avec ses mouvement. Head pose estimation avec un repère OXYZ

RÉSULTATS

Dans un premier temps, nous nous sommes intéressés à la détection de l'identité de l'étudiant à partir d'une photo que l'on fournit au code.

Ensuite, nous avons adapté le code de façon à ce qu'il puisse reconnaître l'identité de l'étudiant à partir d'une vidéo lancée par la caméra de son pc portable. L'identification de l'étudiant se fait en temps réel.

A ce niveau, nous avons répondu à la première problématique de triche qui consiste à s'assurer que c'est bel et bien l'étudiant concerné qui est en face de son ordinateur pour passer l'examen.

Il reste donc l'amélioration du code pour venir annoter une vidéo et lever des drapeaux au bon moment pour alléger l'attention du surveillant ainsi que le développement de l'aspect commercial du produit.

RÉFÉRENCES

1. https://github.com/natanielruiz/deep-head-pose/blob/master/code/test_on_video.py
2. <https://stackoverflow.com/questions/>
3. <https://github.com/opencv/opencv/tree/master/data/haarcascades>