

DÉTECTION FACIALE

Reconnaissance et détection faciale

Chloé Josse BTS SN1

SOMMAIRE

- Reconnaissance / détection faciale
- Système existant
- Avantages et inconvénients
- Détection faciale sous Python
- Conclusion

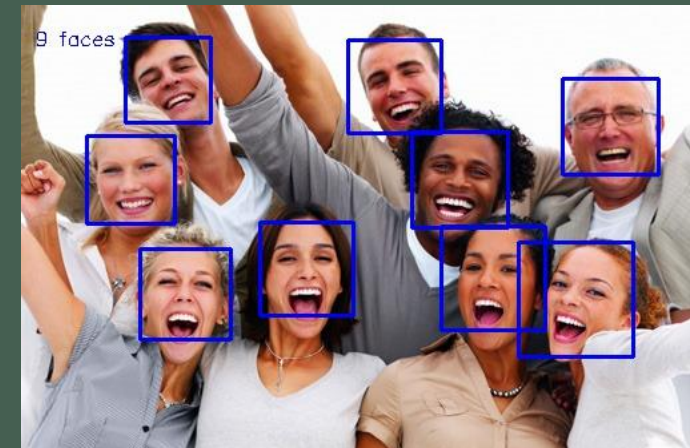
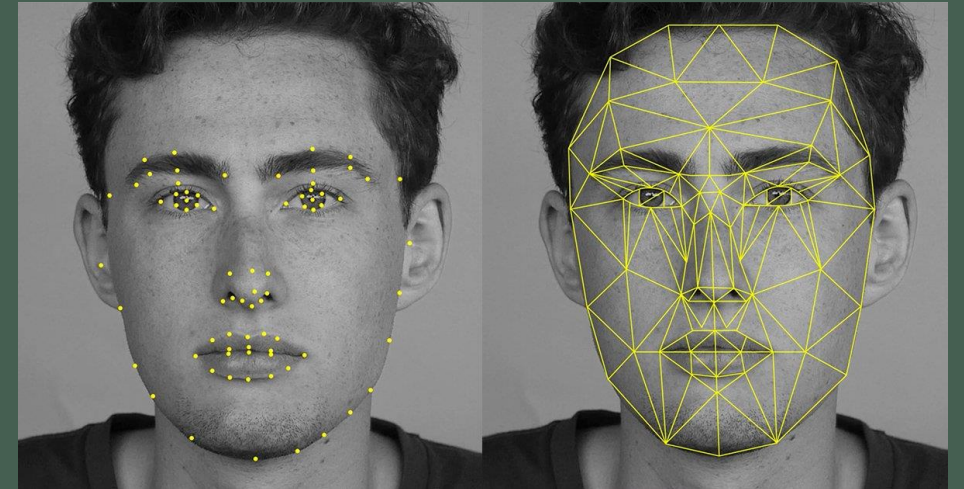
DÉFINITION

- Reconnaissance faciale

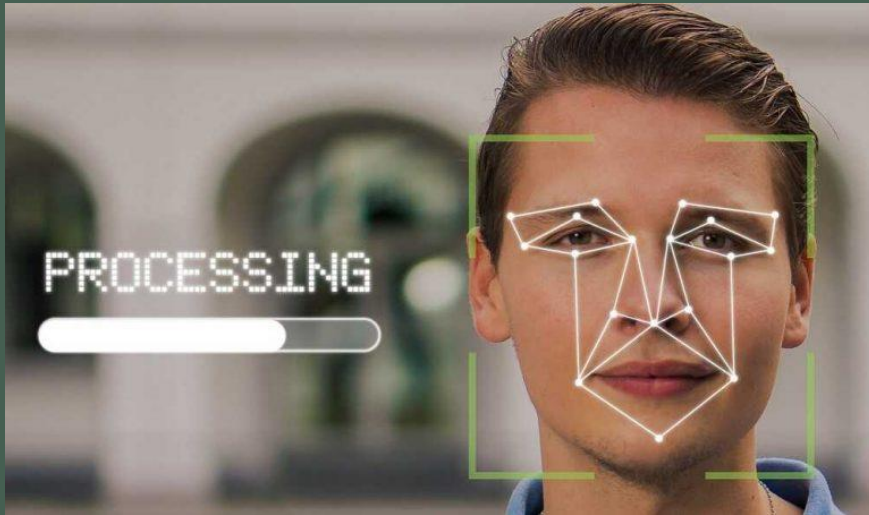
Application logicielle biométrique capable d'identifier de manière unique une personne grâce à une base de données

- Détection faciale

Système capable d'identifier un visage humain sur une photo ou une vidéo, mais ne peut pas dire à qui correspond le visage



SYSTÈME EXISTANT



- Sécurité : Contrôle d'identité biométrique grâce à la reconnaissance faciale
- Services bancaires, marketing et publicité
- Protection biométrique (serrure biométrique, face ID...)



AVANTAGES ET INCONVÉNIENTS

Avantages :

- Sécurité
- Meilleure identification
- Facilité d'utilisation (smartphone...)

Inconvénients :

- Menace sur la vie privée
- Réduction des libertés personnelles
- Violation des droits des personnes
- Possibilité de fraude



DÉTECTION FACIALE SOUS PYTHON

```
facedetimage.py x facedetvideo.py x
1 import cv2
2
3 imagePath = "files/groupe1.jpg" # Chemin de l'image
4
5 #https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_alt.xml
6 cascadeClassifierPath = "haarcascade_frontalface_alt.xml" # Chemin du Classifieur pour le visage
7
8 cascadeClassifier = cv2.CascadeClassifier(cascadeClassifierPath)
9
10 image = cv2.imread(imagePath)
11
12 grayImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) # Conversion de la vidéo en noir et blanc
13
14 detectedFaces = cascadeClassifier.detectMultiScale(grayImage) # Détection des visages
15
16 for(x,y,width,height) in detectedFaces:
17     cv2.rectangle(image, (x, y), (x+width, y+height), (0,255,0), 5)
18
19 cv2.imwrite('resultat.jpg', image) # crée un fichier "resultat" avec l'image de imagePath.
20
21 cv2.imshow("resultat", image)
22 cv2.waitKey(0)
```



DÉTECTION FACIALE SOUS PYTHON

```
1 import cv2
2 import datetime
3
4 # https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade\_frontalface\_alt.xml
5 cascadeClassifierPath = 'haarcascade_frontalface_alt.xml' # Chemin du Classifieur pour le visage
6 cascadeClassifier = cv2.CascadeClassifier(cascadeClassifierPath)
7
8 #cap = cv2.VideoCapture(0) # On récupère la webcam
9 cap = cv2.VideoCapture("files/video.mp4") # On récupère une vidéo
10
11 while(cap.isOpened()):
12     _, frame = cap.read()
13     grayImage = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) # Conversion de la vidéo en noir et blanc
14     detectedFaces = cascadeClassifier.detectMultiScale(grayImage, scaleFactor=1.1, minNeighbors=10, minSize=(20, 20)) # Détection des visages
15
16     for(x,y, width, height) in detectedFaces:
17         cv2.rectangle(frame, (x, y), (x+width, y+height), (0,255,0), 3) # Dessin d'un rectangle autour du visage
18
19     font = cv2.FONT_HERSHEY_SIMPLEX # police d'écriture pour la date
20     text = str(datetime.datetime.now()) # texte de la date
21     frame = cv2.putText(frame, text, (10,50), font, .7, (0,255,255), 2, cv2.LINE_AA) # Ajout de la date
22
23     cv2.imshow("result", frame)
24     if cv2.waitKey(1) == ord('q'):
25         break
26
27 cap.release()
28 cv2.destroyAllWindows()
```

CONCLUSION

- Fonction de plus en plus utilisé de nos jours, qui commence à être très présente dans nos sociétés
- Découverte de Bibliothèque OpenCV et cv2, des fonctionnalités Haar

