1. Introduction à la détection des visages

La détection des visages est une technologie informatique qui identifie les visages humains dans des images fixes ou animées.

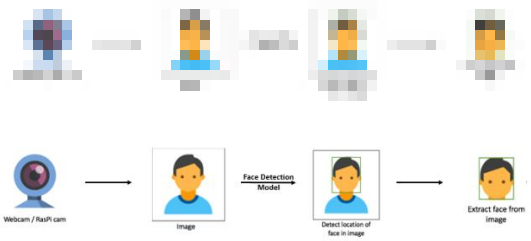
- Durant ce travail, vous nous allez apprendre la ***détection de visage en utilisant OpenCV (Python) . La détection des visages*** est une technologie tendance presque utilisée dans tous les domaines de nos jours, de la sécurité, de la recherche, de l'analyse, de la reconnaissance, des appareils intelligents, de l'automatisation et bien d'autres choses.
- Vous utiliserez le module ***OpenCV*** , qui est une bibliothèque de vision par ordinateur avec le langage python pour détecter les visages humains.
- Vous allez utiliser votre webcam PC pour obtenir le flux vidéo.

Définition -

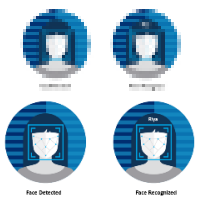
La détection des visages est un terme de technologie informatique utilisé lorsque le logiciel est utilisé pour déterminer l'existence, l'emplacement et la taille d'un visage humain sur une photo particulière. Le logiciel est suffisamment intelligent pour détecter les traits du visage, tout en ignorant d'autres objets comme les arbres, les bâtiments et les corps.

2. Fonctionnement de la détection des visages

- Bien que le processus soit quelque peu complexe, les algorithmes de détection de visage commencent souvent par rechercher des yeux humains ou un visage frontal.
- Les yeux constituent ce qu'on appelle une région de vallée et sont l'une des caractéristiques les plus faciles à détecter.
- Une fois les yeux détectés, l'algorithme pourrait alors tenter de détecter les régions du visage, notamment les sourcils, la bouche, le nez, les narines et l'iris.
- Une fois que l'algorithme présume qu'il a détecté une région faciale, il peut alors appliquer des tests supplémentaires pour valider s'il a effectivement détecté un visage.



Détection de visage versus reconnaissance de visage



La Face Detection:

- Détecte le visage dans l'image.
- Recherche le visage humain général comme un segment dans l'image entière. La sortie peut être un ou plusieurs.
- La sortie sera un rectangle ou des rectangles sur les faces de l'image.

La Face Recognition:

- Reconnaître la face d'entrée de la base de données déjà formée avec le score de correspondance le plus élevé.
- Une seule face doit être donnée en entrée et la sortie sera un nom, un nom de classe ou une face inconnue.

3. Outils et technologies utilisés

- **OpenCV**
- **Programmation Python**

OpenCV

OpenCV (Open Source Computer Vision) est une bibliothèque de fonctions de programmation principalement destinées à la vision par ordinateur en temps réel. En langage simple, c'est une bibliothèque utilisée pour le traitement d'images. Elle est principalement utilisée pour effectuer toutes les opérations liées aux images.

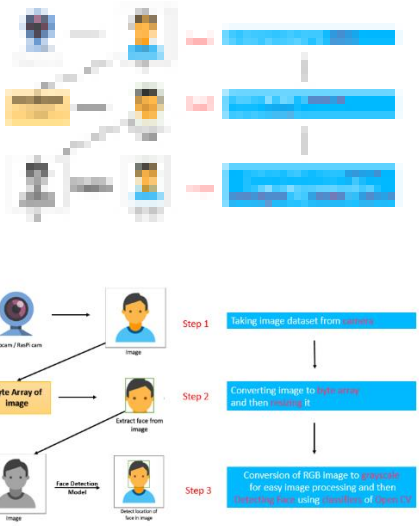
Ce que peut faire OpenCV:

- Lire et écrire des images.
- Détection des visages et de ses caractéristiques.
- Détection de formes telles que cercle, rectangle, etc. dans une image.

Par exemple, Détection d'une pièce dans les images.

- Reconnaissance de texte dans les images. ex. lecture des plaques d'immatriculation
- Modification de la qualité et des couleurs de l'image, par exemple Instagram, CamScanner.
- Développement d'applications de réalité augmentée.

Étapes derrière le travail



4. Codage et compréhension

```
#Installing OpenCV library
!pip install opencv-python

#Importing Library
import cv2

#Input your name to display while detection
name = input("Enter your name here:")
```

- OpenCV doit être installé et importé.
- Saisissez votre nom.

Accéder à votre webcam

```
#This block of code is to access the camera, to get it's video feed
#So as to use it next for face detection

# capture frames from a camera
cap = cv2.VideoCapture(0)

#To Get video output from your camera
while 1:
  #ret stores the continuous video feed
  ret, img = cap.read()

  #To show the video window
  cv2.imshow('img',img)
```

Utilisez la fonction ***cv2.imshow ()*** pour ***afficher*** une image dans une fenêtre. La fenêtre s'adapte automatiquement à la taille de l'image. Le premier argument est un nom de fenêtre qui est une chaîne. Le deuxième argument est notre image.

```
#Important to break the loop, press q
  #else it will be an infinite loop,
  #always put this at the end of your code while using camera
  if cv2.waitKey(100) &amp;amp; 0xff == ord('q'):
    break

# Releases the camera
cap.release()

# De-allocate any associated memory usage
cv2.destroyAllWindows()
```

C'est ainsi que vous utilisez votre caméra, depuis l'ouverture, l'accès à son flux jusqu'à sa libération dans OpenCV.

```
#The OpenCV Classifier for face
#Must be present at the same location as your this ipynb file
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

- Une cascade de Haar est essentiellement un classificateur qui est utilisé pour détecter des objets particuliers de la source.
- Le **haarcascade_frontalface_default.xml** est une cascade de haar conçue par OpenCV **pour détecter la face frontale** .

Detecting Faces

```
cap = cv2.VideoCapture(0)
# loop runs if capturing has been initialized.
while 1:
   ret, img = cap.read()
   # convert to gray scale of each frames
   gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

- Pour la conversion de **B G R** en **Gray** , nous utilisons les drapeaux ***cv2.COLOR_BGR2GRAY***
- Les niveaux de gris réduisent simplement la complexité d'une valeur de pixel 3D (R, G, B) à une valeur 1D, car de nombreuses tâches ne fonctionnent pas mieux avec des pixels 3D (par exemple, la détection des contours).

```
# Detects faces of different sizes in the input image
   faces = face_cascade.detectMultiScale(gray, 1.3, 5)

   # Draws rectangle around the faces
   for (x,y,w,h) in faces:
      cv2.rectangle(img,(x,y),(x+w,y+h),(255,255,0),2)

      # To put the text on video feed .i.e. Your Name
      cv2.putText(img,name, (x - 1, y - 1), cv2.FONT_HERSHEY_PLAIN,4,(0, 255, 0))
```

- ***detectMultiScale ()*** détecte des objets de différentes tailles dans l'image d'entrée. Les objets détectés sont renvoyés sous forme de liste de rectangles.
- Pour dessiner un rectangle, ***cv2.rectangle ()*** est utilisé et vous avez besoin d'un coin supérieur gauche et d'un coin inférieur droit du rectangle. Cette fois, nous allons dessiner un rectangle vert sur le visage détecté.
- ***c v2.putText ()*** met du texte sur la vidéo.

```
cv2.imshow('img',img)
   if cv2.waitKey(100) &amp;amp; 0xff == ord('q'):
      break
# Close the window
cap.release()
# De-allocate any associated memory usage
cv2.destroyAllWindows()
```

- ***cv2.waitKey ()*** prend un personnage pour arrêter la prise de vidéo, si vous ne le mettez pas, ce sera une **boucle infinie**
- ***cap.release () libère*** la caméra occupée
- ***cv2.destroyAllWindows ()*** ferme toutes les fenêtres ouvertes.

Travail à faire :

Rechercher sur internet :

- Des tutoriels qui parlent de la reconnaissance faciale sous python.
- Des exemples de mise en œuvre.