REAL TIME FACE DETECTION USING OPENCV

Face Detection is a computer technology related to computer vision, image processing and deep learning that deals with detecting instances of objects in images and videos. We will do object detection in this article using something known as haar cascades.

What are Haar cascades?

Haar Cascade classifiers are an effective way for object detection. This method was proposed by Paul Viola and Michael Jones in their paper Rapid Object Detection using a Boosted Cascade of Simple Features. Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier.

- Positive images – These images contain the images which we want our classifier to identify.
- Negative Images – Images of everything else, which do not contain the object we want to detect.

What is Opencv?

OpenCV (Open source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being an Apache 2 licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

Requirements:

- Make sure you have python, Matplotlib and OpenCV installed on your pc (all the latest versions).
- The haar cascade files can be downloaded from the OpenCV Github repository.
- A working webcam

The Code:

- creating a face cascade

```
import cv2
import sys

cascPath = sys.argv[1]
faceCascade = cv2.CascadeClassifier(cascPath)
```

- Setting the video source to the default webcam

```
video_capture = cv2.VideoCapture(0)
```

- Capturing the video

  while True:
  # Capture frame-by-frame
  ret, frame = video_capture.read()

  The read() function reads one frame from the video source, which in this example is the webcam. This returns:

  1. The actual video frame read (one frame on each loop)
  2. A return code

- searching for the face in our captured frame.

```
# Capture frame-by-frame
ret, frame = video_capture.read()

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize=(30, 30),
    flags=cv2.cv.CV_HAAR_SCALE_IMAGE
)

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

# Display the resulting frame
cv2.imshow('Video', frame)
```

- Wait key

  if cv2.waitKey(1) & 0xFF == ord('q'):
  break
  We wait for the ‘q’ key to be pressed. If it is, we exit the script.

- A final cleaning up

```
# When everything is done, release the capture
video_capture.release()
cv2.destroyAllWindows()
```