**Face Detection**

Can’t have dog pictures here, sadly :(

Image for post



[Photo by Free-Photos from Pixabay](https://pixabay.com/photos/young-woman-female-youth-healthy-1208208/)

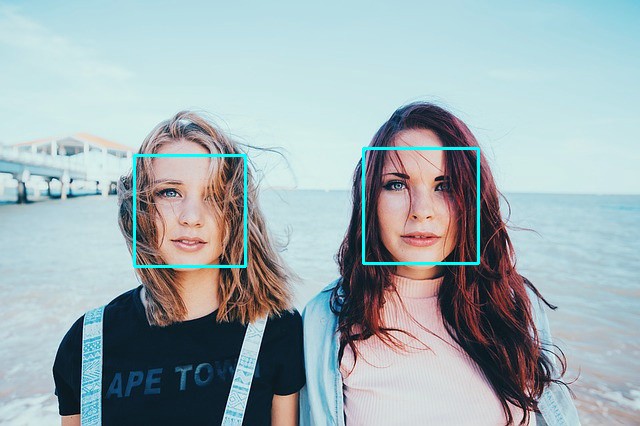
import cv2image\_path = "./Path/To/Photo.extension"  
face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')image = cv2.imread(image\_path)  
gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  
faces = face\_cascade.detectMultiScale(  
 gray,  
 scaleFactor= 1.1,  
 minNeighbors= 5,  
 minSize=(10, 10)  
)faces\_detected = format(len(faces)) + " faces detected!"  
print(faces\_detected)# Draw a rectangle around the faces  
for (x, y, w, h) in faces:  
 cv2.rectangle(image, (x, y), (x+w, y+h), (255, 255, 0), 2)viewImage(image,faces\_detected)

The detectMultiScale function is a general function that detects objects. Since we’re calling it on the face cascade, that’s what it detects.

**The detectMultiScale function takes 4 parameters**

* The first parameter is the grayscale image.
* The second parameter is the scaleFactor. Since some faces may be closer to the camera, they would appear bigger than the faces in the back. The scale factor compensates for this.
* The detection algorithm uses a moving window to detect objects. minNeighbors defines how many objects are detected near the current one before it declares the face found.
* minSize, meanwhile, gives the size of each window.

Image for post



2 faces detected!

**Contours—A method for Object Detection**

Using color-based [image segmentation](https://www.fritz.ai/features/image-segmentation.html), you can detect objects.   
cv2.findContours &cv2.drawContours are two functions that help you with that.

Recently, I’ve written a very detailed articled called [Object detection via color-based image segmentation using Python](https://towardsdatascience.com/object-detection-via-color-based-image-segmentation-using-python-e9b7c72f0e11). Everything you need to know about contours is there.

**Finally, Saving the image**

import cv2  
image = cv2.imread("./Import/path.extension")  
cv2.imwrite("./Export/Path.extension", image)