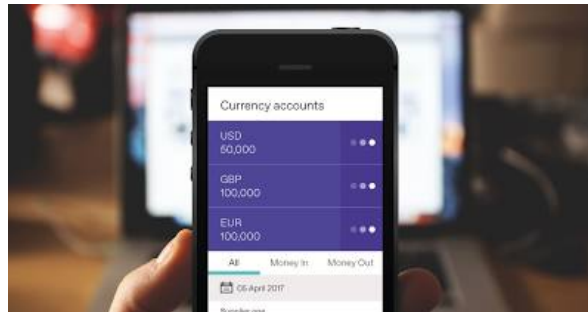
[Home](#)[+=1](#)[Support the Content](#)[Community](#)[Log in](#)[Sign up](#)

Haar Cascade Object Detection Face & Eye OpenCV Python Tutorial

In this OpenCV with Python tutorial, we're going to discuss object detection with Haar Cascades. We'll do face and eye detection to start. In order to do object recognition/detection with cascade files, you first need cascade files. For the extremely popular tasks, these already exist. Detecting things like faces, cars, smiles, eyes, and license plates for example are all pretty prevalent.

First, I will show you how to use these cascade files, then I will show you how to embark on creating your very own cascades, so that you can detect any object you want, which is pretty darn cool!



Open multiple local
currency accounts in GBP,
EUR & USD – for Free.

Ad WorldFirst

Sign Up



You can use Google to find various Haar Cascades of things you may want to detect. You shouldn't have too much trouble finding the aforementioned types. We will use a [Face cascade](#) and [Eye cascade](#). You can find a few more at the root directory of Haar cascades. Note the license for using/distributing these Haar Cascades.

Let's begin our code. I am assuming you have downloaded the `haarcascade_eye.xml` and `haarcascade_frontalface_default.xml` from the links above, and have these files in your project's directory.

```
import numpy as np
import cv2

# multiple cascades: https://github.com/Itseez/opencv/tree/master/data/haarcascades

#https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml
```

```
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')  
#https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_eye.xml  
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')  
  
cap = cv2.VideoCapture(0)
```

Here, we begin with import cv2 and numpy, then we load in our face and eye cascades. Simple enough so far.

```
while 1:  
    ret, img = cap.read()  
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)  
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
```

Now we begin our typical loop, the only new thing here is the creation of faces. For more information, visit the [documentation for the detectMultiScale functionality](#). Basically, it finds faces! We also want to find eyes, but, in a world of false positives, wouldn't it be prudent to logically make it so that we only find eyes in faces? Let's hope we're not looking for eyes that aren't in faces! In all seriousness, "eye detection" probably wouldn't find an eyeball laying around. Most eye detection uses the surrounding skin, eye lids, eye lashes, and eye brows to also make the detection. Thus, our next step is to break down the faces first, before getting to the eyes:

```
for (x,y,w,h) in faces:  
    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)  
    roi_gray = gray[y:y+h, x:x+w]  
    roi_color = img[y:y+h, x:x+w]
```

Here, we're finding faces, their sizes, drawing rectangles, and noting the ROI. Next, we poke around for some eyes:

```
eyes = eye_cascade.detectMultiScale(roi_gray)  
for (ex,ey,ew,eh) in eyes:  
    cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
```

If we find those, we'll go ahead and make some more rectangles. Next we finish up:

```
cv2.imshow('img',img)
k = cv2.waitKey(30) & 0xff
if k == 27:
    break

cap.release()
cv2.destroyAllWindows()
```

Full code:

```
import numpy as np
import cv2

# multiple cascades: https://github.com/Itseez/opencv/tree/master/data/haarcascades

#https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
#https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_eye.xml
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

cap = cv2.VideoCapture(0)

while 1:
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

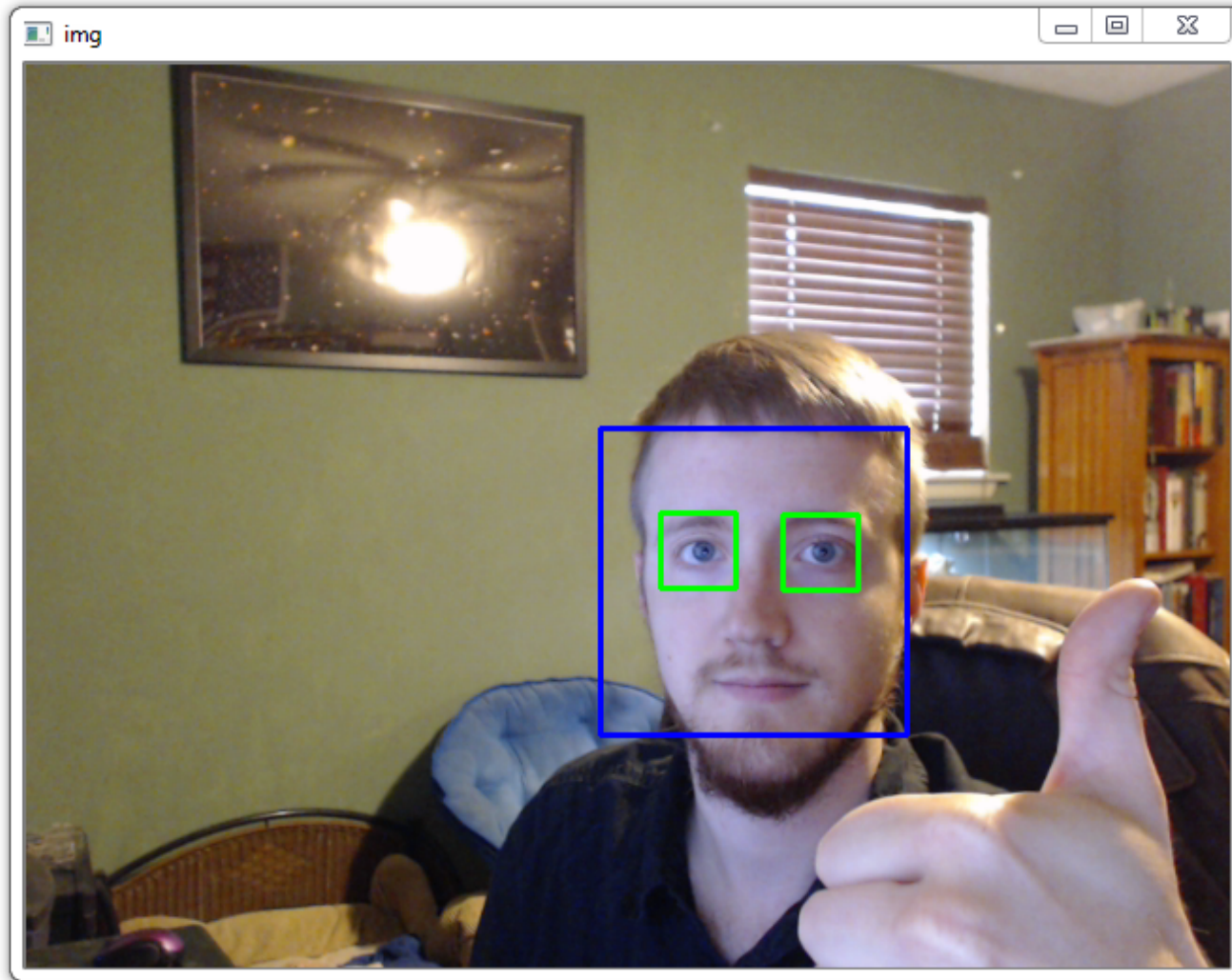
    for (x,y,w,h) in faces:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
```

```
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

cv2.imshow('img',img)
k = cv2.waitKey(30) & 0xff
if k == 27:
    break

cap.release()
cv2.destroyAllWindows()
```

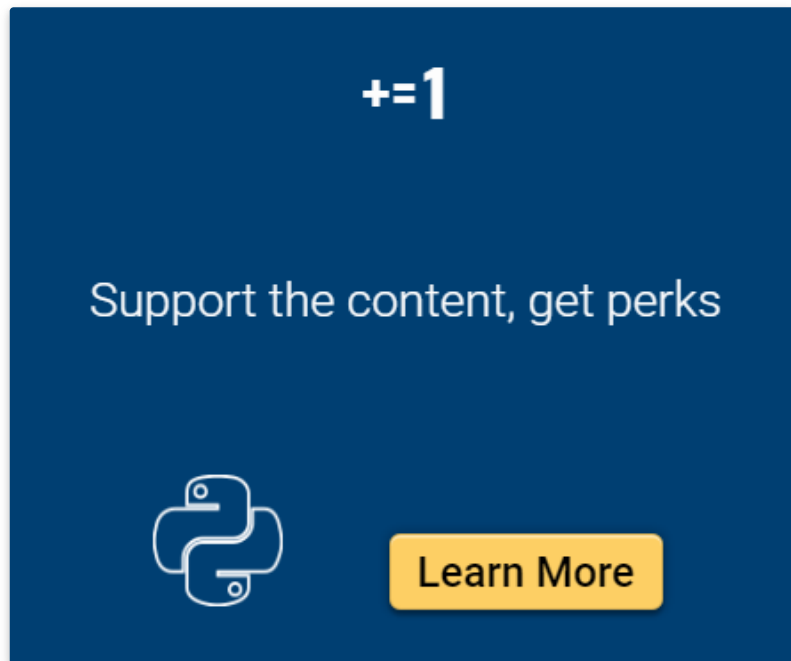
Result:



Not bad. You may notice I had to remove my glasses. These caused some trouble. My mouth also was detected pretty often as eyes, sometimes even a face, but you get the idea. Facial hair and other things can often fool rudimentary face detection, and even advanced. Skin color can also cause a lot of trouble, as we often try to simplify images as much as possible, thus losing a lot of color values. There's even a mini-industry around avoiding face detection and recognition. The website [CVDazzle](https://cvdazzle.com/) would be an example of this. Some of these are pretty outlandish, but they can work pretty well. You could also

always go the route of complete facial reconstructive surgery to avoid automated tracking and detection, so there's always that, but that's a bit more permanent. A hair-do is a bit more temporary and easy to do.

Alright so detecting faces, eyes, and cars is alright, but we are programmers. We want to be able to do anything. Turns out, things can get pretty confusing, and somewhat challenging to build your own Haar Cascades, but other people do it... and so can you! That's what we'll be talking about in the next tutorial.



OpenCV with Python Intro and loading Images tutorial

Loading Video Source OpenCV Python Tutorial

Drawing and Writing on Image OpenCV Python Tutorial

Image Operations OpenCV Python Tutorial

Image arithmetics and Logic OpenCV Python Tutorial

Thresholding OpenCV Python Tutorial

Color Filtering OpenCV Python Tutorial

Blurring and Smoothing OpenCV Python Tutorial

Morphological Transformations OpenCV Python Tutorial

Canny Edge Detection and Gradients OpenCV Python Tutorial

Template Matching OpenCV Python Tutorial

GrabCut Foreground Extraction OpenCV Python Tutorial

Corner Detection OpenCV Python Tutorial

Feature Matching (Homography) Brute Force OpenCV Python Tutorial

MOG Background Reduction OpenCV Python Tutorial

Haar Cascade Object Detection Face & Eye OpenCV Python Tutorial

Creating your own Haar Cascade OpenCV Python Tutorial



FREE EBOOK

Developer First Look: Competitive Mobile Gaming

Get the eBook



Amazon GameOn eBook

Learn how Amazon GameOn can help you cultivate a passionate fan-base. Get the eBook today.
amazonappservices.com



DOWNLOAD

You've reached the end!

Contact: Harrison@pythonprogramming.net.

Support this Website!

[Hire me](#)

[Facebook](#)

[Twitter](#)

[Google+](#)

Legal stuff:

[Terms and Conditions](#)

[Privacy Policy](#)

© OVER 9000! [PythonProgramming.net](https://pythonprogramming.net)

Programming is a superpower.