

DS878-N Processamento Digital de Imagens

Julio Cesar Mariotto Junior

Python e Opencv

Curitiba

2021

O objetivo desta tarefa era instalar o OpenCV com o Python seguindo os passos deste artigo <https://blog.cedrotech.com/opencv-uma-breve-introducao-visao-computacional-com-python/>

Conforme a ultima aula, já havia conseguido instalar o Python e a biblioteca do OpenCV.

```
Administrador: C:\Windows\system32\cmd.exe

C:\Program Files\Python>python --version
Python 3.10.0

C:\Program Files\Python>python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\program files\python\lib\site-packages (21.3.1)

C:\Program Files\Python>
```

Passando os passos seguintes, encontrei dificuldades em rodar o código proposto no artigo.

```
1  import numpy as np;
2  import cv2 as cv;
3
4  face_classifier = cv.CascadeClassifier('haarcascade_frontalface_default.xml')
5
6  image = cv.imread('image.jpg')
7  image_gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
8
9  faces = face_classifier.detectMultiScale(image_gray, 1.3, 5)
10
11  for(x,y,w,h) in faces:
12      cv.rectangle(image,(x,y),(x+w,y+h),(255,0,0),2)
13
14
15  cv.imshow('imagem', image)
16  cv.waitKey(0)
17  cv.destroyAllWindows()
```

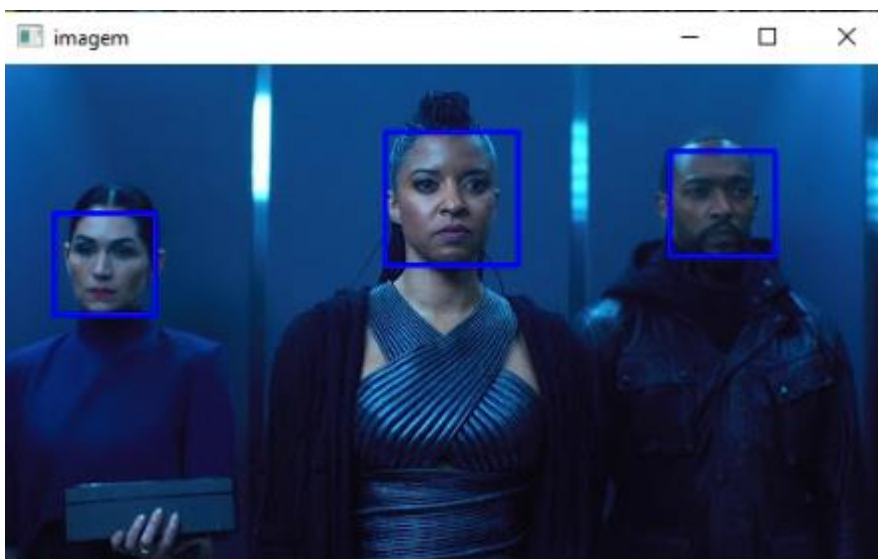
```
C:\Program Files\Python>python app2.py
Traceback (most recent call last):
  File "C:\Program Files\Python\app2.py", line 9, in <module>
    faces = face_classifier.detectMultiScale(image_gray, 1.3, 5)
cv2.error: OpenCV(4.5.4-dev) D:\a\opencv-python\opencv-python\opencv\modules\objdetect\src\cascadedetect.cpp:1689: error: (-215:Assertion failed) !empty() in function 'cv::CascadeClassifier::detectMultiScale'

C:\Program Files\Python>
```

Depois de buscar um pouco na internet, achei uma solução que funcionou para o problema, faltava adicionar uma propriedade no parâmetro da CascadeClassifier.

```
1 import numpy as np;
2 import cv2 as cv;
3
4 face_classifier = cv.CascadeClassifier(cv.data.harcascades + 'haarcascade_frontalface_default.xml')
5
6 image = cv.imread('image.jpg')
7 image_gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
8
9 faces = face_classifier.detectMultiScale(image_gray, 1.3, 5)
10
11 for(x,y,w,h) in faces:
12     cv.rectangle(image, (x,y), (x+w,y+h), (255,0,0), 2)
13
14
15 cv.imshow('imagem', image)
16 cv.waitKey(0)
17 cv.destroyAllWindows()
```

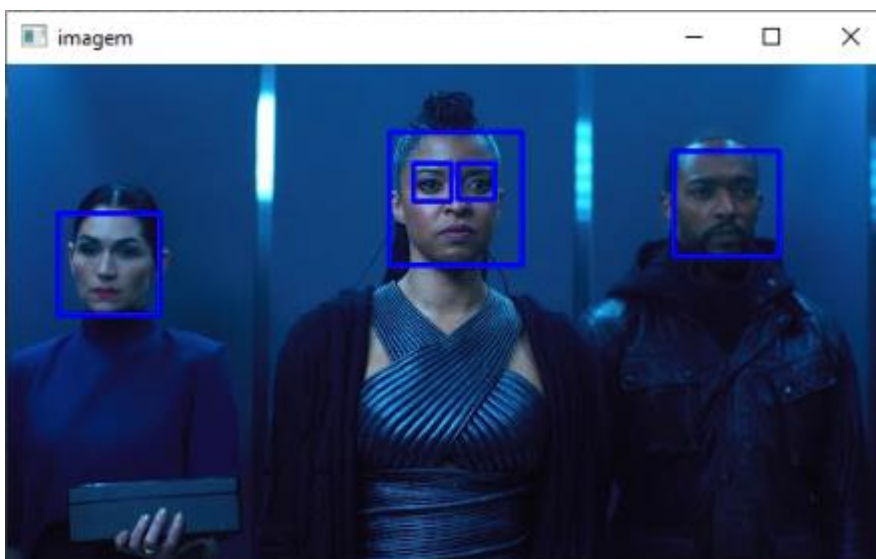
Depois de arrumar o código, este programa me retornou esta tela.



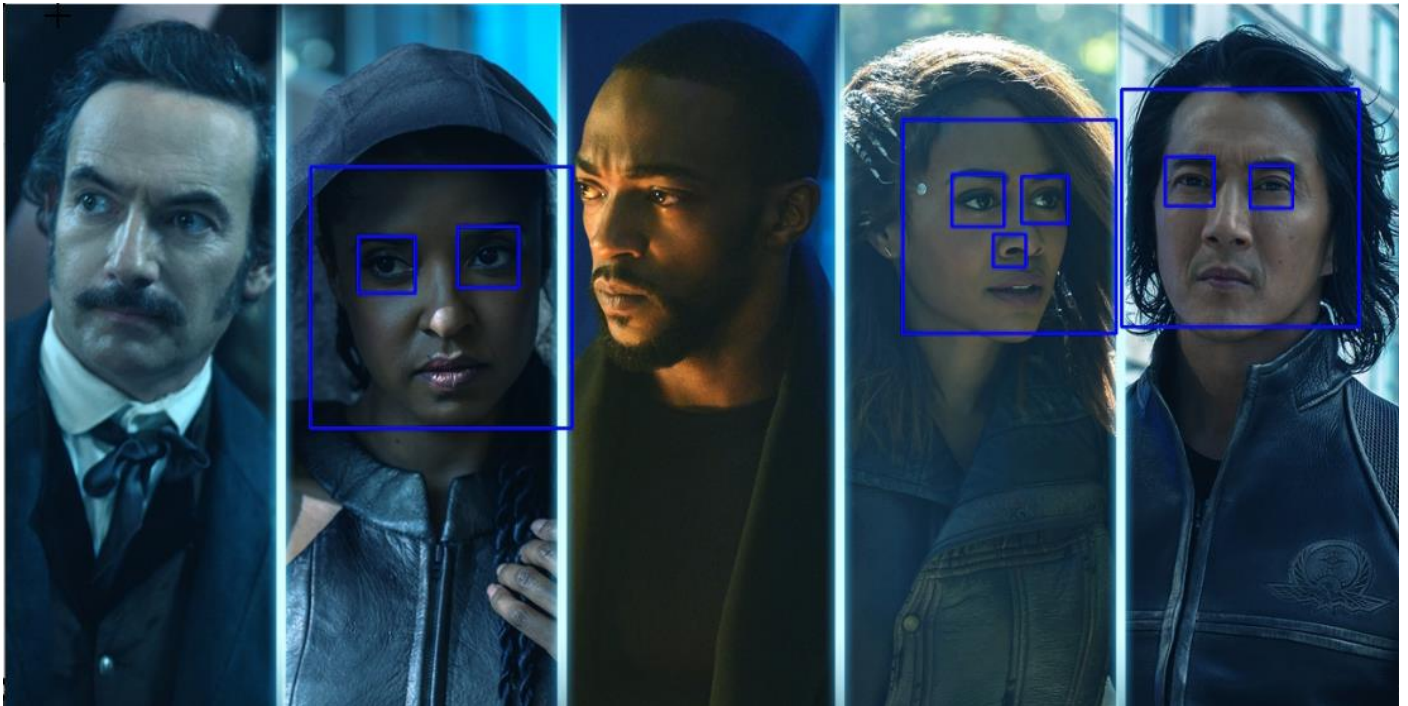
Seguindo o artigo implementei a segunda parte do código.

```
1  import numpy as np;
2  import cv2 as cv;
3
4  face_classifier = cv.CascadeClassifier(cv.data.harcascades + 'haarcascade_frontalface_default.xml')
5  eye_classifier = cv.CascadeClassifier(cv.data.harcascades + 'haarcascade_eye.xml')
6
7  image = cv.imread('image.jpg')
8  gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
9
10 faces = face_classifier.detectMultiScale(gray, 1.3, 5)
11
12 for(x,y,w,h) in faces:
13     cv.rectangle(image,(x,y),(x+w,y+h),(255,0,0),2)
14     roi_gray = gray[y:y+h, x:x+w]
15     roi_color = image[y:y+h, x:x+w]
16     eyes = eye_classifier.detectMultiScale(roi_gray)
17     for(ex,ey,ew,eh) in eyes:
18         cv.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(255,0,0),2)
19
20 cv.imshow('imagem', image)
21 cv.waitKey(0)
22 cv.destroyAllWindows()
```

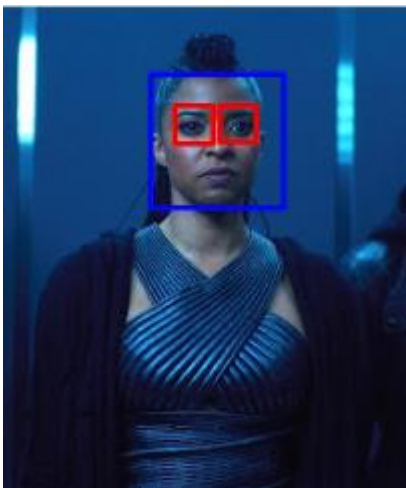
Que me retornou esta tela.



Como o programa só identificou os olhos da pessoa mais a frente, tentei rodar o código com outra imagem.



Ainda sim, foi possível melhorar um pouco o código diferenciando as cores dos quadrados para rosto e olhos.



Após analisar os resultados, percebi que para o algoritmo ter um resultado eficiente, a imagem precisa estar em uma resolução boa, com um bom ângulo, foco e a pessoa na imagem precisa estar um pouco próxima.