

# APÊNDICE G

Código Avaliação dos Algoritmos

```
1 import cv2
2 import os
3 import numpy as np
4 from PIL import Image
5
6 detectorFace = cv2.CascadeClassifier("
    haarcascade_frontalface_default.xml")
7 #reconhecedor = cv2.face.EigenFaceRecognizer_create()
8 #reconhecedor.read("classificadorEigen.yml")
9 #reconhecedor = cv2.face.FisherFaceRecognizer_create()
10 #reconhecedor.read("classificadorFisher.yml")
11 reconhecedor = cv2.face.LBPHFaceRecognizer_create()
12 reconhecedor.read("classificadorLBPH.yml")
13
14
15
16 totalAcertos = 0
17 percentualAcerto = 0.0
18 totalConfianca: float = 0.0
19
20 caminhos = [os.path.join('base_teste', f) for f in os.
    listdir('base_teste')]
21 for caminhoImagem in caminhos:
22     imagemFace = Image.open(caminhoImagem).convert('L')
23     imagemFaceNP = np.array(imagemFace, 'uint8')
24     facesDetectadas = detectorFace.detectMultiScale(
    imagemFaceNP)
25     for (x, y, l, a) in facesDetectadas:
26         idprevisto, confianca = reconhecedor.predict(
    imagemFaceNP)
27         path = os.path.split(caminhoImagem)[1]
28         idatual = float(path.replace("pessoa.", " ").
    replace(".jpg", " "))
29         print(str(idatual) + " foi classificado como " +
    str(idprevisto) + " - " + str(confianca))
30         if idprevisto == idatual:
31             totalAcertos += 1
32             totalConfianca += confianca
33             #cv2.rectangle(imagemFaceNP, (x, y), (x + l, y + a
    ), (0, 0, 255), 2)
34             #cv2.imshow("Face", imagemFaceNP)
35             #cv2.waitKey(1000)
36 percentualAcerto = ((totalAcertos / 30) *100)
37 print(percentualAcerto)
38 totalConfianca = totalConfianca / totalAcertos
```

```
39 print("Percentual de acerto: " + str(percentualAcerto))
40 print("Total confiança: " + str(totalConfianca))
41
```