

APÊNDICE B

Código de Captura de Imagens

```
1 import cv2
2 import numpy as np
3
4 classificador = cv2.CascadeClassifier("
  haarcascade_frontalface_default.xml")
5 classificadorOlho = cv2.CascadeClassifier ("
  haarcascade_eye.xml")
6 camera = cv2.VideoCapture(0)
7 amostra = 1
8 numeroAmostras = 25
9 id = input('Informe seu identificador: ')
10 largura, altura = 220, 220
11 print("Capturando as Faces .. ")
12
13
14 while(True):
15     conectado, imagem = camera.read()
16     imagemCinza = cv2.cvtColor(imagem, cv2.COLOR_BGR2GRAY)
17     print(np.average(imagemCinza))
18     facesDetectadas = classificador.detectMultiScale(
  imagemCinza,
19
  scaleFactor=1.5,
20
  minSize=(150,150))
21
22     for(x,y,l,a) in facesDetectadas:
23         cv2.rectangle(imagem, (x, y), (x + l, y + a), (0,
  0, 255), 2)
24         '''regiao = imagem[y:y + a, x:x + l]
25         regiaoCinzaOlho = cv2.cvtColor(regiao, cv2.
  COLOR_BGR2GRAY)
26         olhosDetectados = classificadorOlho.
  detectMultiScale(regiaoCinzaOlho)
27         for (ox, oy, ol, oa) in olhosDetectados:
28             cv2.rectangle(regiao, (ox, oy), (ox + ol, oy
  + oa), (0,255,0), 2)'''
29
30         if cv2.waitKey(1) & 0xFF == ord('q'):
31
32             '''if np.average(imagemCinza) > 110:'''
33             imagemFace = cv2.resize(imagemCinza [y:y + a,
  x: x + l], (largura, altura))
34             cv2.imwrite("Photos/pessoa." + str(id) + "." +
  str(amostra) + ".jpg", imagemFace)
```

```
35         print("[foto " + str(amostra) + "capturada com  
sucesso]")  
36         amostra +=1  
37  
38     cv2.imshow("Face", imagem)  
39     cv2.waitKey(1)  
40     if (amostra >= numeroAmostras + 1 ):  
41         break  
42  
43 print("Faces Capturadas com Sucesso")  
44 camera.release()  
45 cv2.destroyAllWindows()  
46  
47
```