

```

import numpy as np
import cv2
from cv2 import cvtColor
import os, sys
import time
import pickle
import face_recognition
from keras.preprocessing.image import image
import tensorflow as tf
from keras.models import load_model
import RPi.GPIO as gpio

#0 trecho acima carrega todas as bibliotecas necessárias para o funcionamento do
sistema.
#A linha abaixo abre a janela de captura de video.
cap=cv2.VideoCapture(0)
#Carregando a base de dados do face_recognition
data=pickle.loads(open("enc_TCC.pickle","rb").read())
st=0
a=0
# carregando os pesos das redes neurais de cada usuário
henr_model=load_model('henrique_saved.h5')

iran_model=load_model('iranilson_saved.h5')

wands_model=load_model('wands_saved.h5')
#carregando as faces padrões de cada usuário para comparações adicionais
print("1...",")
henr=cv2.imread("henrique.jpg",3)#le a imagem padrão de henrique
boxhenr=face_recognition.face_locations(henr,model="hog")#localiza a face
henr_encod=face_recognition.face_encodings(henr,boxhenr)# extrai as características
print("2...",")
iran=cv2.imread("iranilson.jpg",3)
boxiran=face_recognition.face_locations(iran,model="hog")
iran_encod=face_recognition.face_encodings(iran,boxiran)
print("3...")
wands=cv2.imread("wands.jpg",3)
boxwands=face_recognition.face_locations(wands,model="hog")
wands_encod=face_recognition.face_encodings(wands,boxwands)
gpio.setmode(gpio.BOARD)
gpio.setup(12, gpio.OUT)
gpio.output(12, gpio.LOW)

print("VALENDO!!!!!!!!!!!!!!!!!!!!")
#Enquanto a janela de captura estiver aberta...
while(cap.isOpened()):
    ret, frame = cap.read()
    frameout=frame.copy()
    #captura o frame pela camera
    boxes=face_recognition.face_locations(frame,model="hog")
    #usa o algoritmo HOG para encontrar a face
    encodings=face_recognition.face_encodings(frame,boxes)
    #utilizando a area da face recortada para extrair as características
    names=[]
    # o trecho abaixo verifica se a face pertence a algum usuário contido na base de
    dados.
    for encoding in encodings:
        matches=face_recognition.compare_faces(data["encodings"],encoding)
        name="Unknown"
        if True in matches:
            matchedidxs=[i for (i, b) in enumerate(matches) if b]
            counts= {}
            for i in matchedidxs:
                name=data["names"][i]
                counts[name]=counts.get(name,0)+1
            name=max(counts, key=counts.get)
            names.append(name)

```

```

for ((top, right, bottom, left),name) in zip(boxes,names):
# se o nome for classificado como intruso escreve o nome intruso em cima da caixa
    if name=="intruso":
        cv2.rectangle(frame, (left, top), (right,bottom), (0,0,255),3)
        y= top - 15 if top - 15>15 else top+15

        cv2.putText(frame, name, (left,y), cv2.FONT_HERSHEY_SIMPLEX, 0.75,
(0,0,255), 2)
        cv2.putText(frame, 'Pressione "s" para acessar via senha', (100,460),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,255,255), 2)

        if name=="henrique": #se o nome for igual a "henrique"
            dist=face_recognition.face_distance(encoding,henr_encod)# calcula a
distancia entre a face extraída a face padrão
            crp= frame[int(boxes[0][0]):int(boxes[0][2]),int(boxes[0]
[3]):int(boxes[0][1])]#recorta a imagem
            crp= cv2.resize(crp,(224,224),interpolation=cv2.INTER_LANCZOS4)#
redimensiona a imagem para 224 por 224 pixels
            cv2.imwrite("out0.jpg",crp)#salva a imagem
            img=image.load_img("out0.jpg", target_size=(224,224))#carrega a face
recortada para ser utilizada como entrada da CNN-keras
            img=np.expand_dims(img, axis=0)#expande as dimensoes (recomendações
keras)
            result=henr_model.predict_proba(img)#verifica se a face recortada
pertence ou não ao "henrique"
            if result[0][0]==1.0 and dist<0.45:# se o resultado for positivo e a
distancia com relação a face padrão for inferior a 0.45 prossegue
                cv2.rectangle(frame, (left, top), (right,bottom), (0,255,0),3)
                y= top - 15 if top - 15>15 else top+15
                cv2.putText(frame, name, (left,y), cv2.FONT_HERSHEY_SIMPLEX, 0.75,
(0,255,0), 2)# Escreve o nome henrique em cima da caixa
                gpio.output(12, gpio.HIGH)#ativa o rele
                time.sleep(5)#espera 5 segundos
                gpio.output(12, gpio.LOW)#desliga o rele
            else: #se não, conclui que é intruso
                cv2.rectangle(frame, (left, top), (right,bottom), (0,0,255),3)
                y= top - 15 if top - 15>15 else top+15

                cv2.putText(frame, 'intruso', (left,y), cv2.FONT_HERSHEY_SIMPLEX,
0.75, (0,0,255), 2)#escreve intruso em cima da caixa
                cv2.putText(frame, 'Pressione "s" para acessar via senha',
(100,460), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,255,255), 2)# escreve em baixo do
video a opção de acessar via senha
# o bloco abaixo tem a mesma função, porém para o usuário "iranilson"
                if name=="iranilson":
                    dist=face_recognition.face_distance(encoding,iran_encod)
                    crp= frame[int(boxes[0][0]):int(boxes[0][2]),int(boxes[0]
[3]):int(boxes[0][1])]
                    crp= cv2.resize(crp,(224,224),interpolation=cv2.INTER_LANCZOS4)
                    cv2.imwrite("out1.jpg",crp)
                    img=image.load_img("out1.jpg", target_size=(224,224))
                    img=np.expand_dims(img, axis=0)
                    result=iran_model.predict_proba(img)
                    if result[0][0]==1.0 and dist<0.45:
                        cv2.rectangle(frame, (left, top), (right,bottom), (0,255,0),3)
                        y= top - 15 if top - 15>15 else top+15
                        cv2.putText(frame, name, (left,y), cv2.FONT_HERSHEY_SIMPLEX, 0.75,
(0,255,0), 2)
                        gpio.output(12, gpio.HIGH)
                        time.sleep(5)
                        gpio.output(12, gpio.LOW)
                    else:
                        cv2.rectangle(frame, (left, top), (right,bottom), (0,0,255),3)
                        y= top - 15 if top - 15>15 else top+15

                        cv2.putText(frame, 'intruso', (left,y), cv2.FONT_HERSHEY_SIMPLEX,

```

```

0.75, (0,0,255), 2)
    cv2.putText(frame, 'Pressione "s" para acessar via senha',
(100,460), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,255,255), 2)
# o bloco abaixo tem a mesma função, porém para o usuário "wanderson"
    if name=="wands":
        dist=face_recognition.face_distance(encoding,wands_encod)
        crp= frame[int(boxes[0][0]):int(boxes[0][2]),int(boxes[0]
[3]):int(boxes[0][1])]
        crp= cv2.resize(crp,(224,224),interpolation=cv2.INTER_LANCZOS4)
        cv2.imwrite("out2.jpg",crp)
        img=image.load_img("out2.jpg", target_size=(224,224))
        img=np.expand_dims(img, axis=0)
        result=wands_model.predict_proba(img)
        if dist<0.45:
            cv2.rectangle(frame, (left, top), (right,bottom), (0,255,0),3)
            y= top - 15 if top - 15>15 else top+15
            cv2.putText(frame, name, (left,y), cv2.FONT_HERSHEY_SIMPLEX, 0.75,
(0,255,0), 2)
            gpio.output(12, gpio.HIGH)
            time.sleep(5)
            gpio.output(12, gpio.LOW)
        else:
            cv2.rectangle(frame, (left, top), (right,bottom), (0,0,255),3)
            y= top - 15 if top - 15>15 else top+15

            cv2.putText(frame, 'intruso', (left,y), cv2.FONT_HERSHEY_SIMPLEX,
0.75, (0,0,255), 2)
            cv2.putText(frame, 'Pressione "s" para acessar via senha',
(100,460), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,255,255), 2)

    if ret==True:
        #out.write(frame)

        esca=0#essa variavel informa ao sistema que a tecla e foi
pressionada
        cv2.imshow('frame',frame)#mostra o frame
#As instruções abaixo se referem ao mecanismo da senha
        if cv2.waitKey(1) & 0xFF == ord('s'):# se a tecla "s" for pressionada
(entra no modo senha)
            cv2.putText(frameout, 'Pressione "e" para sair', (100,460),
cv2.FONT_HERSHEY_SIMPLEX, 0.75, (0,255,255), 2)# informa na parte inferior do
video que para sair é necessário segurar a tecla "e"
            cv2.imshow('frame',frameout)#mostra o frame
            while cv2.waitKey(1) & 0xFF != ord('l'):# enquanto o digito "l" nao
for pressionado por mais de um segundo
                print("digite...")
                if cv2.waitKey(1) & 0xFF == ord('e'):# se a tecla "e" for
pressionada
                    esca+=1# incrementa a variável "esca"
                    break
            while cv2.waitKey(1) & 0xFF != ord('x'):# enquanto o digito "x" nao
for pressionado por mais de um segundo...
                print("digite...")
                if cv2.waitKey(1) & 0xFF == ord('e'):
                    esca+=1
                    break
            while cv2.waitKey(1) & 0xFF != ord('3'):# enquanto o digito "3" nao
for pressionado por mais de um segundo...
                print("digite...")
                if cv2.waitKey(1) & 0xFF == ord('e'):
                    esca+=1
                    break
            if esca==0:#se todos os digitos estao corretos

```

```
        gpio.output(12, gpio.HIGH)#ativa o rele
        time.sleep(5)#espera 5 segundos
        gpio.output(12, gpio.LOW)#desliga o rele
        print("liberado!!!)

    else:
        break

cap.release()

cv2.destroyAllWindows()
```