THE DEVELOPER'S CONFERENCE

Reconhecimento Facial no Raspberry Pi

Felipe Kühne
Líder Técnico - DBServer

◆ Reconhecimento facial?

THE DEVELOPER'S CONFERENCE

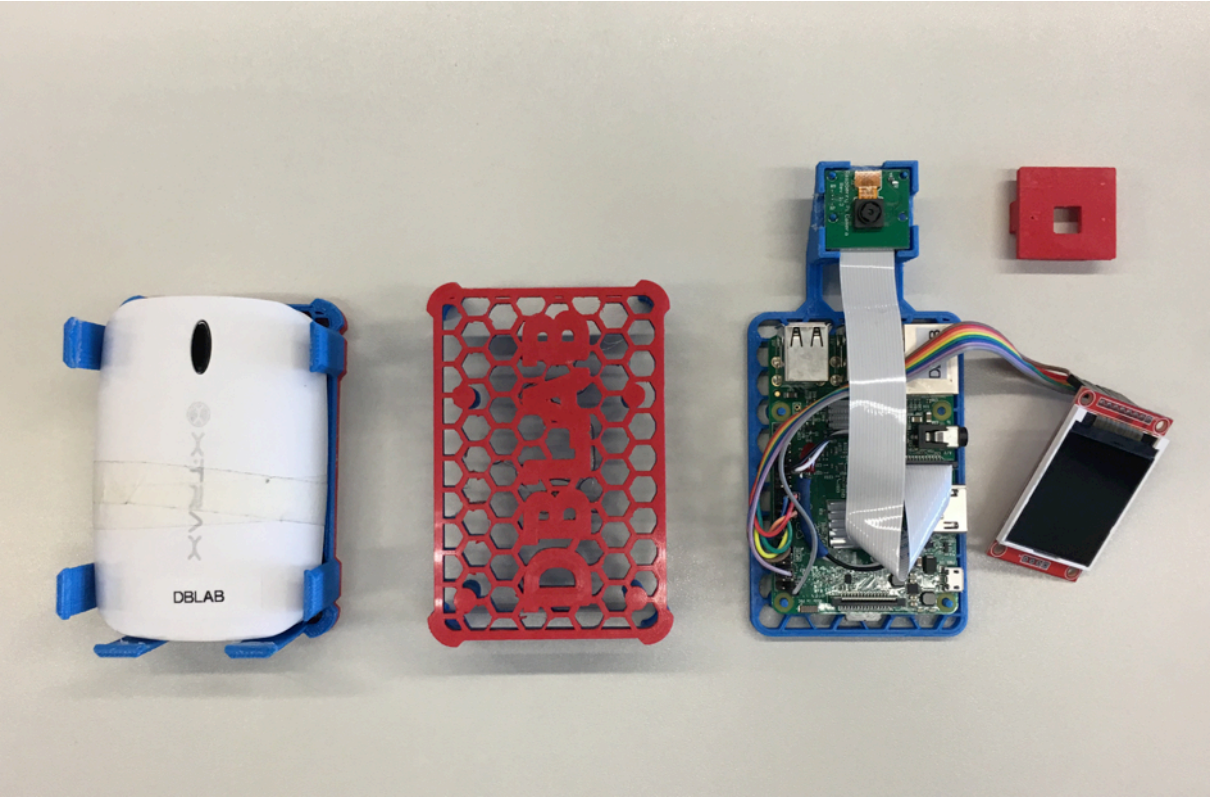
① Ler uma **imagem**

② Identificar as **faces**

③ Extrair as **características** de cada face

④ **Comparar** as características com outras características de identidades já conhecidas

Hardware



- Raspberry Pi 3B
- Câmera 5MP
- Display ST7735 1.8”
- Bateria (ou fonte)
- Case 3D

◆ Software

THE DEVELOPER'S CONFERENCE

face_recognition 1.2.3

```
pip install face_recognition
```

luma.lcd 2.2.0

```
pip install luma.lcd
```

Pillow 6.2.1

```
pip install Pillow
```

face_recognition

THE DEVELOPER'S CONFERENCE

```
import face_recognition
known_image = face_recognition.load_image_file("biden.jpg")
unknown_image = face_recognition.load_image_file("unknown.jpg")

biden_encoding = face_recognition.face_encodings(known_image)[0]
unknown_encoding = face_recognition.face_encodings(unknown_image)[0]

results = face_recognition.compare_faces([biden_encoding], unknown_encoding)
```

◆ Desenvolvimento

THE DEVELOPER'S CONFERENCE

① Ler uma **imagem**

② Localizar as **faces**

③ Extrair as **características** de cada face

④ **Comparar** as características com outras características de identidades já conhecidas

① Ler uma **imagem**

THE DEVELOPER'S CONFERENCE

Thread para leitura da webcam:

```
stream = cv2.VideoCapture()
frame = stream.read()
```

taxas independentes!

Fluxo principal:

```
capture = WebCamVideoStream().start()

while capture.isOpened():
        frame = capture.read()

        …
```

① Ler uma **imagem**

THE DEVELOPER'S CONFERENCE

- Câmera: 5MP / R$20 (similar no eBay)
- Leitura da imagem a 640 x 480 pixels
- Processamento a cada dois frames
    - *Downscale* de 4x (160 x 120 pixels)

② Localizar as **faces**

THE DEVELOPER'S CONFERENCE

② Localizar as **faces**

THE DEVELOPER'S CONFERENCE

② Localizar as **faces**

THE DEVELOPER'S CONFERENCE

```
face_locations = face_recognition.face_locations(frame)

face_count = len(face_locations)
```

② Localizar as **faces**

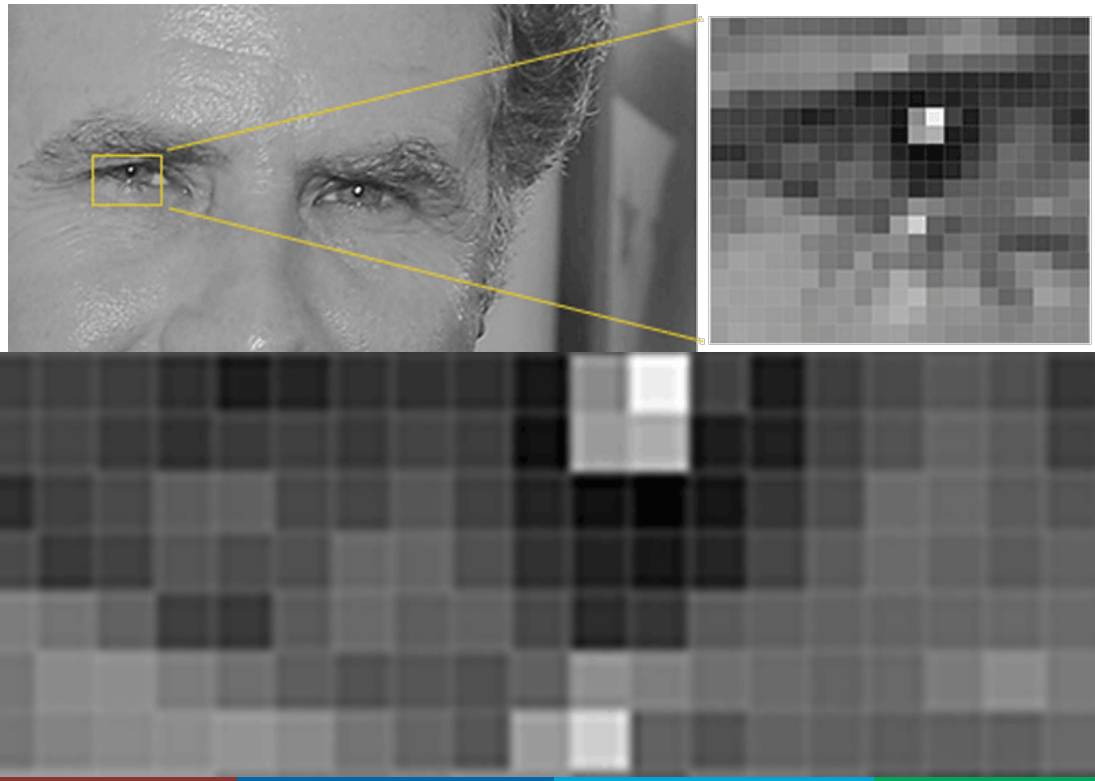
THE DEVELOPER'S CONFERENCE

A **Dlib** dispõe de dois métodos para identificação de faces:

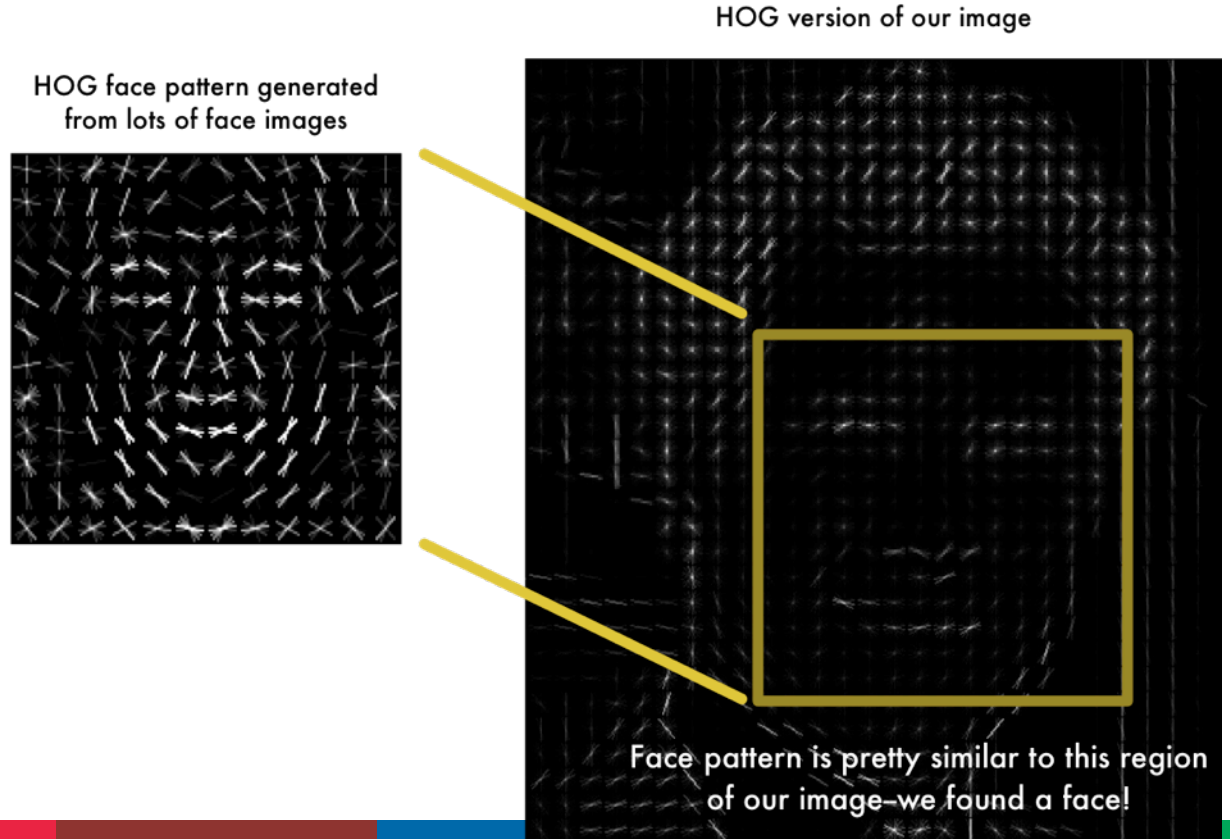
- **HOG** (*Histogram of Oriented Gradients*)
  - Leve e rápida
  - 0,02s
- **CNN** (*Convolutional Neural Network*)
  - Mais precisa
  - Computacionalmente pesada
  - 0,64s

② Localizar as **faces (HOG)**

THE DEVELOPER'S CONFERENCE



② Localizar as **faces (HOG)**



③ Extrair as **características**

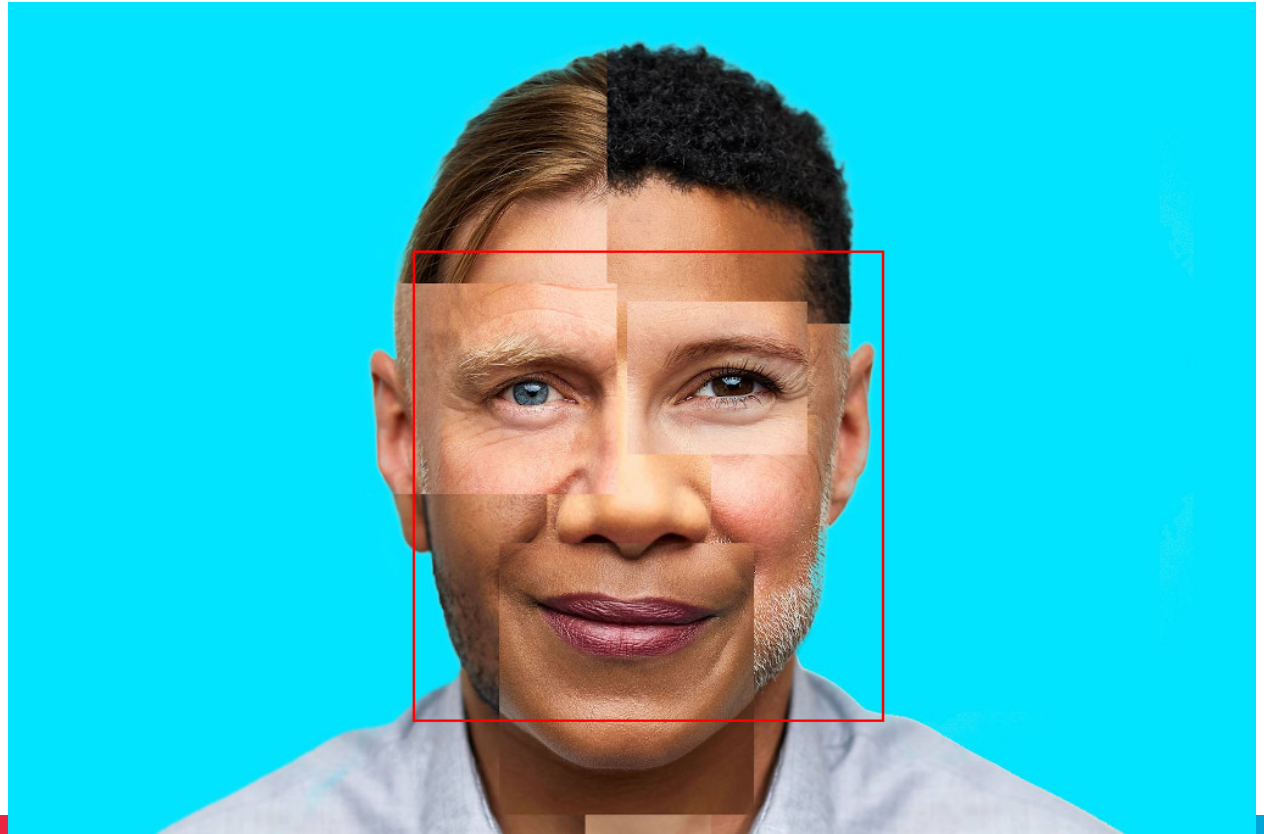
THE DEVELOPER'S CONFERENCE

```
for face_location in face_locations:

        face_encodings = face_recognition.face_encodings(
                frame, face_location)
```

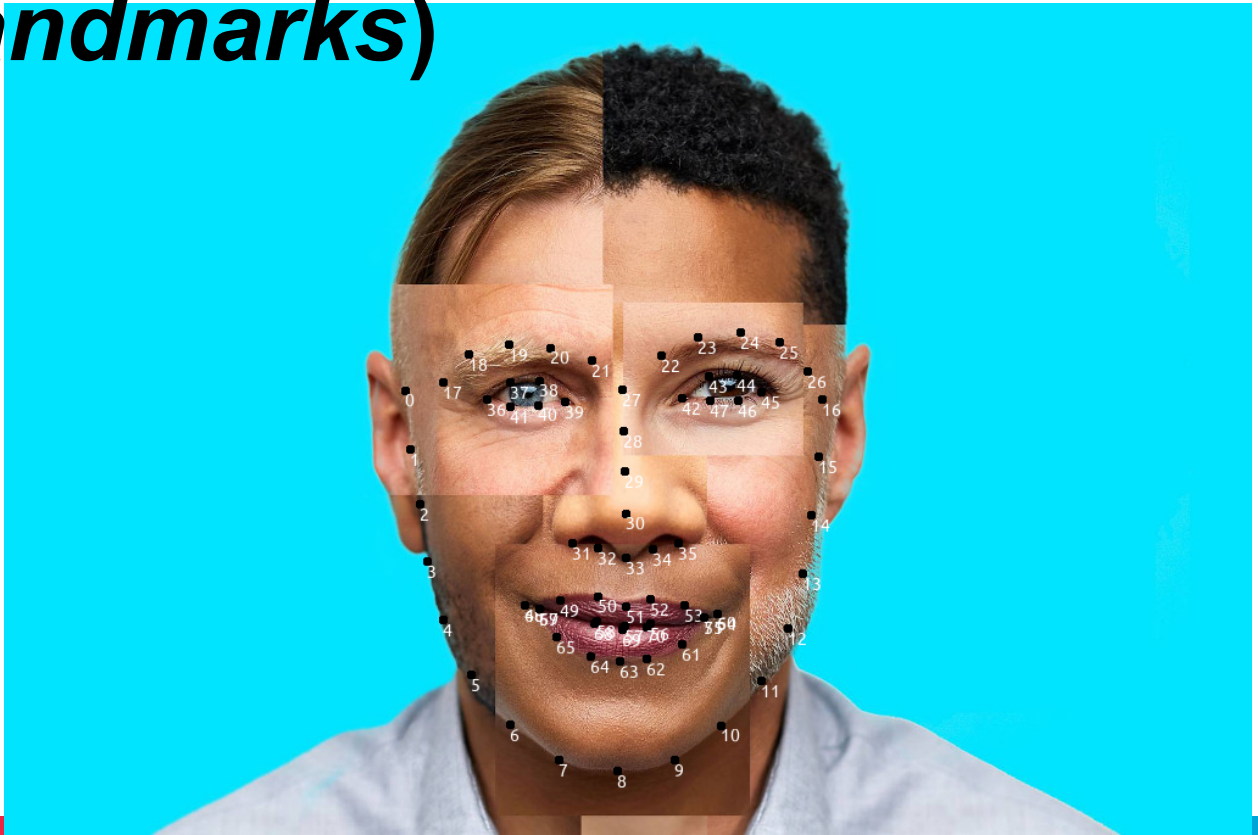
③ Extrair as **características**

THE DEVELOPER'S CONFERENCE



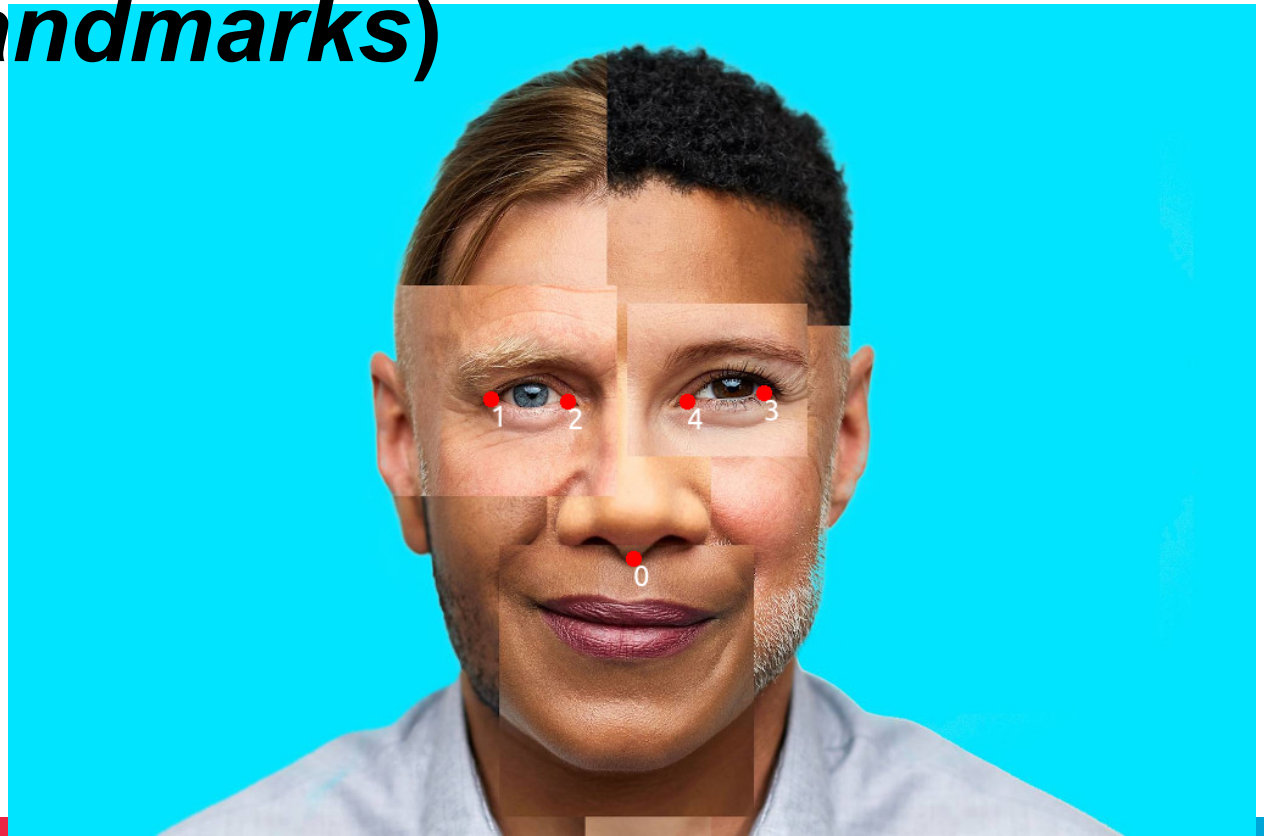
③ Extrair as **características** (***landmarks***)

THE DEVELOPER'S CONFERENCE

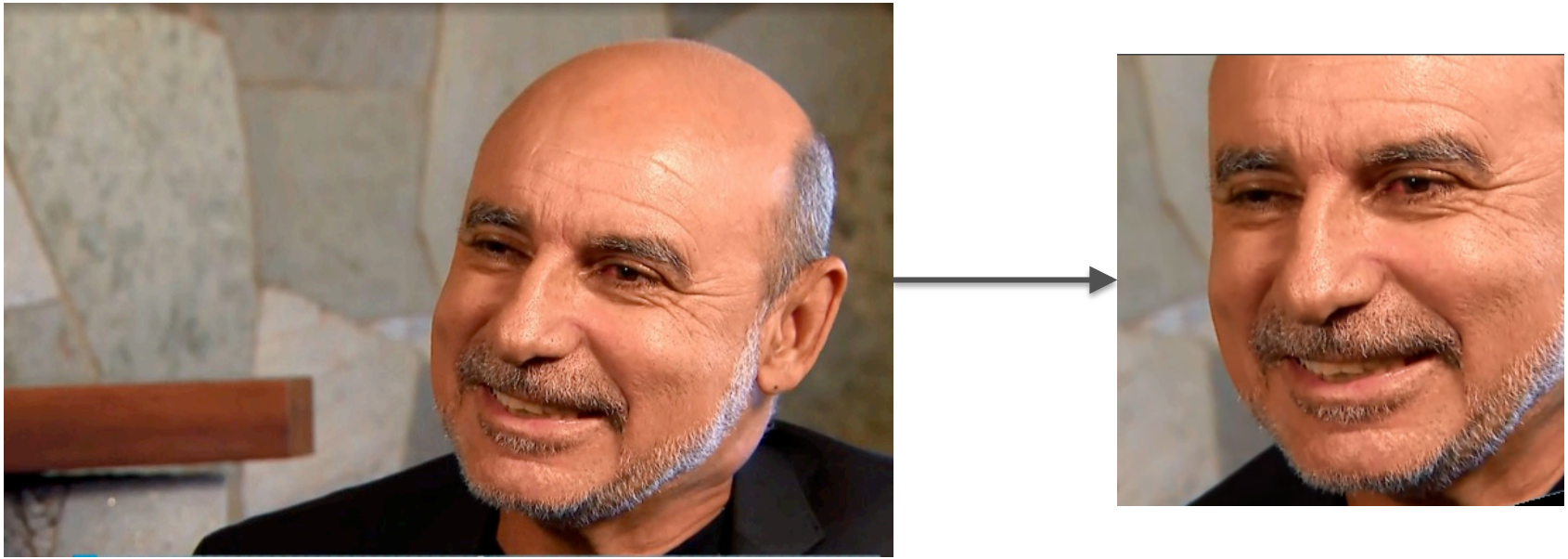


③ Extrair as **características** (***landmarks***)

THE DEVELOPER'S CONFERENCE



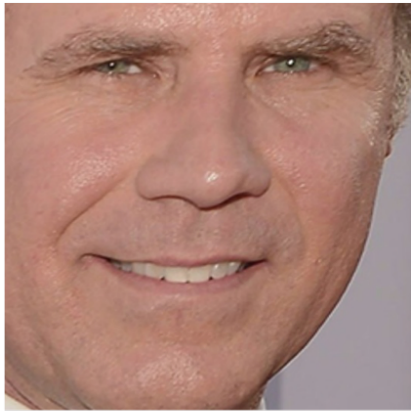
③ Extrair as **características (alinhamento)**



③ Extrair as **características** (***encodings***)

THE DEVELOPER'S CONFERENCE

Input Image



128 Measurements Generated from Image

0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

③ Extrair as **características** (***encodings***)

THE DEVELOPER'S CONFERENCE

Rede pré-treinada da **Dlib** para extração das características:

- **Resnet-34** com 29 camadas convolucionais
- Dataset de treinamento: 3M faces
- Acurácia de **99,38%** (*Labelled Faces in the Wild*)

④ **Comparar** com faces conhecidas (***distância euclidiana***)

THE DEVELOPER'S CONFERENCE

```
for face in face_encodings:

      distances = np.linalg.norm(
            known_face_encodings - face, axis=1)

      match = np.argmin(distances)

      if distances[match] <= MIN_DISTANCE_TO_MATCH:
            name = known_face_names[match]
```

MQTT (Eclipse Paho)

THE DEVELOPER'S CONFERENCE

```
mqttClient = mqtt.Client()
mqttClient.connect('emqx.dbserver.com.br', 1883)
mqttClient.loop_start()

...

mqttClient.publish('dblab/mqtt/facerec/name', name)
```

Display ST7735 (Luma)

THE DEVELOPER'S CONFERENCE

```
font = ImageFont.truetype('miscfs_.ttf', 17)

serial = spi(port=0, device=0,
             bus_speed_hz=32000000,
             gpio_DC=23, gpio_RST=24)

device = st7735(serial,
                width=160, height=128, rotate=2,
                gpio_LIGHT=18, active_low = False)

terminal = terminal(device, font12)
```

Display ST7735 (Luma)

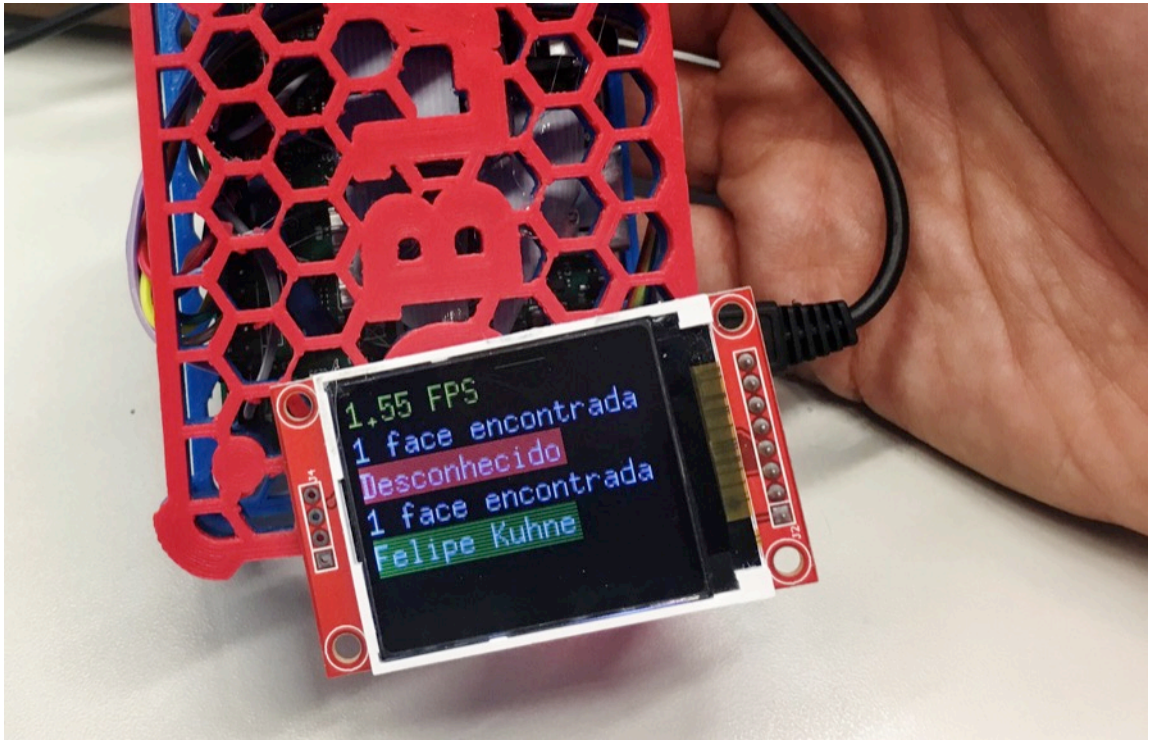
THE DEVELOPER'S CONFERENCE

```
term12.background_color("black")
term12.foreground_color("white")

term12.println(text="Iniciando coleta de identidades")
```

DEMO

THE DEVELOPER'S CONFERENCE



DEMO

THE DEVELOPER'S CONFERENCE

```
fkuhne@lordvader:~/dblab/gitlab/face-recognition$ python generateFeatures.py -i ~/Desktop/TDC-faces -o ~/Desktop/TDC-faces/tdc-faces.py -j 100 -v
Iniciando...
import done
Iniciando...
/home/fkuhne/Desktop/TDC-faces/tdc-faces.py sera re-criado.
Diretorio de entrada: /home/fkuhne/Desktop/TDC-faces
Arquivo de saida:     /home/fkuhne/Desktop/TDC-faces/tdc-faces.py
Jitter:               100
Force:                True
Append:               False
Coletando caracteristicas das faces (15 arquivos)...
        Felipe-Kuhne-4.png... OK
        Giovanni-Comunello-5.png... OK
        Giovanni-Comunello-3.png... OK
        Felipe-Kuhne-1.png... OK
        Felipe-Kuhne-5.png... OK
        Pablo-Oliveira-4.png... OK
        Giovanni-Comunello-2.png... OK
        Felipe-Kuhne-3.png... OK
        Pablo-Oliveira-2.png... OK
        Pablo-Oliveira-5.png... OK
        Giovanni-Comunello-1.png... OK
        Pablo-Oliveira-3.png... OK
        Felipe-Kuhne-2.png... OK
        Giovanni-Comunello-4.png... OK
        Pablo-Oliveira-1.png... OK
Arquivo de caracteristicas /home/fkuhne/Desktop/TDC-faces/tdc-faces.py criado com sucesso.
        Tamanho: 18100 bytes
        Imagens: 15 (validas: 15)
```

Referências

THE DEVELOPER'S CONFERENCE

- https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
- https://github.com/ageitgey/face_recognition
- http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html

Referências adicionais

THE DEVELOPER'S CONFERENCE

- https://towardsdatascience.com/real-time-face-recognition-with-cpu-983d35cc3ec5
- https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB
- https://pypi.org/project/mtcnn
- https://github.com/sthanhng/yoloface
- https://towardsdatascience.com/faced-cpu-real-time-face-detection-using-deep-learning-1488681c1602
- https://cmusatyalab.github.io/openface
- https://www.pyimagesearch.com/2015/12/21/increasing-webcam-fps-with-python-and-opencv

OBRIGADO!

THE DEVELOPER'S CONFERENCE

Felipe Kühne

fkuhne@dbserver.com.br

https://dblab.io

https://github.com/fkuhne/tdcpoa19

THE DEVELOPER'S CONFERENCE