

## RECONOCIMIENTO FACIAL OPENCV Y PYTHON

Es necesario entender la diferencia entre la detección y reconocimiento de rostros, ya que son términos que muchas veces se usan como si se refiriera a una misma acción. La diferencia es simple, la detección de rostros busca caras dentro de una imagen o dentro fotogramas en caso de emplear un video. El reconocimiento facial en cambio emplea la detección de rostros y buscará identificar a la persona que le pertenece aquel rostro detectado.

Por lo tanto primero debemos emplear detección facial, y luego sobre el rostro detectado pasaremos al reconocimiento facial. Para este último procedimiento se necesita que previamente se haya entrenado un clasificador de rostros de las personas que se deseen reconocer.

Para este ejemplo utilizaremos el método LBPH (Histograma de patrones binarios locales). Es un algoritmo de extracción de características de textura de imagen, una característica local, una **clasificación de la foto y la cara a recuperar** utilizado en el estudio de un algoritmo de extracción con mas características.

[https://docs.opencv.org/4.2.0/da/d60/tutorial\\_face\\_main.html#tutorial\\_face\\_eigenfaces](https://docs.opencv.org/4.2.0/da/d60/tutorial_face_main.html#tutorial_face_eigenfaces)

### REALIZAR MODELO DE RECONOCIMIENTO FACIAL

Lo primero es crear un script para capturar los rostros de personas que vamos a reconocer, luego pasaremos con el entrenamiento con el método seleccionado. Una vez entrenado obtenemos un modelo para finalmente en otro script poder leerlo y probar el reconocimiento facial.

Creamos una carpeta llamada ReFacial.

Dentro creamos una carpeta llamada "Data" donde almacenamos los rostros de las personas que vamos a reconocer

Luego nos vamos a nuestro IDE favorito, y creamos un archivo Python llamado

**capturandoRostros.py**

```

import cv2
import os
import imutils

#Crear Carpeta con el nombre de la persona sino esta crearla
personName = 'Jonatan'
dataPath = 'C:/Users/Redes Cableadas/Desktop/IA/cnca/ReFacial/Data' #Cambia
a la ruta donde hayas almacenado Data
personPath = dataPath + '/' + personName

if not os.path.exists(personPath):
    print('Carpeta creada: ',personPath)
    os.makedirs(personPath)

cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)#abrir web cam

faceClassif =
cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default
.xml')#algoritmo identificar rostro
count = 0 #iniciar contador

#leemos cada fotogramas de la imagen y lo redimensionamos para que no sea
muy grandes,
while True:

    ret, frame = cap.read()
    if ret == False: break
    frame = imutils.resize(frame, width=640)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = frame.copy()

    faces = faceClassif.detectMultiScale(gray,1.3,5)
#Guardamos la foto de la web cam, con 150 pixeles, lo guardamos en la
carpeta con el nombre rostro.
    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
        rostro = auxFrame[y:y+h,x:x+w]
        rostro = cv2.resize(rostro,(150,150),interpolation=cv2.INTER_CUBIC)
        cv2.imwrite(personPath + '/rostro_{}.jpg'.format(count),rostro)
        count = count + 1
    cv2.imshow('frame',frame)

#Se guardaran 300 rostros de forma automática al abrir la ventana de la web
cam

```

```

k = cv2.waitKey(1)
if k == 27 or count >= 300:
    break

cap.release()
cv2.destroyAllWindows()

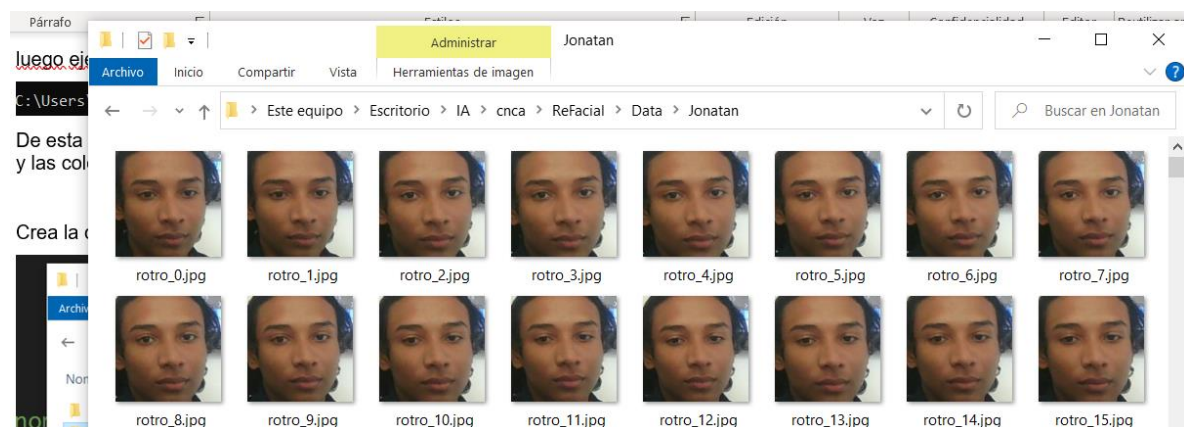
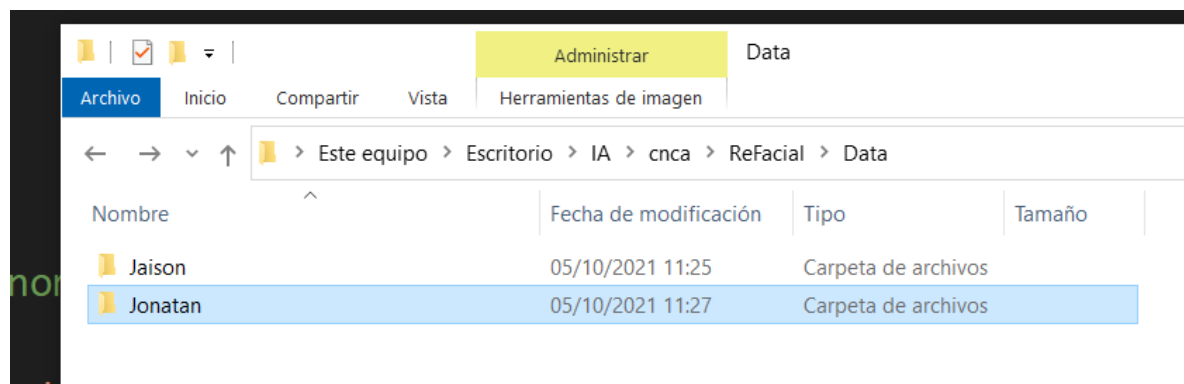
```

luego ejecutamos el alrchivo

```
C:\Users\Redes Cableadas\Desktop\IA\cnca\ReFacial>python capturandoRostros.py
```

De esta forma abre la web cam durante unos segundos el script guardara 300 fotos y las colocara en la carpeta.

Crea la carpeta con las 300 fotos dentro



Si deseo guardar otro rostro, se debe cambiar el nombre de la carpeta en el script fila 6. Y ejecutar. Luego al abrir la web cam colocar la otra persona.

Luego creamos el script para entrenar las fotos llamado entrenandoRF.py

```
import cv2
import os
import numpy as np

#asignamos la ruta y alistamos las carpetas creadas
dataPath = 'C:/Users/Redes Cableadas/Desktop/IA/cnca/ReFacial/Data' #Cambia
a la ruta donde hayas almacenado Data
peopleList = os.listdir(dataPath)
print('Lista de personas: ', peopleList)

#creamos listas vacias para colocarle etiquetas a los rostros
#la primera persona tendra 0, la segunda 1 y asi sucesivamente
labels = []
facesData = []
label = 0

#para leer cada una de las imagenes
for nameDir in peopleList:#leer las carpetas
    personPath = dataPath + '/' + nameDir
    print('Leyendo las imágenes')

    for fileName in os.listdir(personPath):#leer las imagenes dentro de las
carpetas
        print('Rostros: ', nameDir + '/' + fileName)
        labels.append(label)
        facesData.append(cv2.imread(personPath+'/'+fileName,0))

    label = label + 1 #incrementar label

# Métodos para entrenar el reconocedor
face_recognizer = cv2.face.LBPHFaceRecognizer_create()

# Entrenando el reconocedor de rostros
print("Entrenando...")
face_recognizer.train(facesData, np.array(labels))#entrenar todas las
imagenes

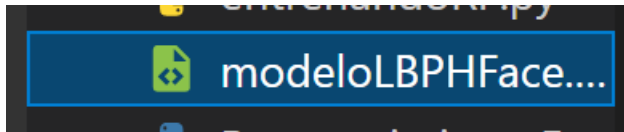
# Almacenando el modelo obtenido
face_recognizer.write('modeloLBPHFace.xml')
print("Modelo almacenado...")
```

Lo ejecutamos

```
: \Users\Redes Cableadas\Desktop\IA\cnca\ReFacial>python entrenandoRF.py
```

Esperemos que ejecute y el crea un archivo llamado modeloLBPHFace.xml

Este tendrá todos los parámetros para comparar las imágenes con otras



Luego creamos otro Script para revisar si el modelo funciona llamado

### **ReconocimientoFacial.py**

```
import cv2
import os

dataPath = 'C:/Users/Redes Cableadas/Desktop/IA/cnca/ReFacial/Data' #Cambia
a la ruta donde hayas almacenado Data
imagePaths = os.listdir(dataPath)
print('imagePaths=',imagePaths)

#llamamos al algoritmo
face_recognizer = cv2.face.LBPHFaceRecognizer_create()

#leemos el modelo creado
face_recognizer.read('modeloLBPHFace.xml')

cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)

faceClassif =
cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default
.xml')
#leemos cada fotogramas de la imagen y lo redimensionamos para que no sea
muy grandes,
while True:
    ret,frame = cap.read()
    if ret == False: break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = gray.copy()
```

```

    faces = faceClassif.detectMultiScale(gray,1.3,5)
#Redimensionar las imagenes a 150 pixeles
    for (x,y,w,h) in faces:
        rostro = auxFrame[y:y+h,x:x+w]
        rostro = cv2.resize(rostro,(150,150),interpolation= cv2.INTER_CUBIC)
        result = face_recognizer.predict(rostro)#Predic predice una etiqueta
y la imagen de confianza en este caso (rostro)

        cv2.putText(frame,'{}'.format(result),(x,y-
5),1,1.3,(255,255,0),1,cv2.LINE_AA)#visualziar dos valores y rectangulo que
rodea objetos

        # LBPHFace
        if result[1] < 55:
            cv2.putText(frame,'{}'.format(imagePaths[result[0]]),(x,y-
25),2,1.1,(0,255,0),1,cv2.LINE_AA)
            cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
        else:
            cv2.putText(frame,'Desconocido',(x,y-
20),2,0.8,(0,0,255),1,cv2.LINE_AA)
            cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)

    cv2.imshow('frame',frame)
    k = cv2.waitKey(1)
    if k == 27:
        break

cap.release()
cv2.destroyAllWindows()

```

lo ejecutamos

```
C:\Users\Redes Cableadas\Desktop\IA\cnca\ReFacial>python ReconocimientoFacial.py_
```

Nos abre la web cam, el sistema nos debe reconocer