

Tarea 3: Reconocimiento de rostros con OpenCV y Python

William José, Sicán Hernández, 202002896^{1,*}

¹Facultad de Ingeniería, Universidad de San Carlos, Ciudad Universitaria, Zona 12, Guatemala.

I. DESCRIPCION DE LA TAREA

La tarea consistio en crear un 3 scrips de python los cuales se encargarian de tomar muestras de los rostros, de hacer el entrenamiento para detectar los rostros y por ultimo crear un programa el cual se encargue de detectar el rostro ya sea de un video o de el uso de la camara de la computadora.

II. CAPTURA DE ROSTROS

Codigo de captura de rostros

```
capturandorostros.py  entrenandoRF.py  ReconocimientoFacial.py
1 import cv2
2 import os
3 import imutils
4
5 personaName = 'presentadora'
6 dataPath = 'C:/Users/willy/Desktop/tareasproyectos2025/Tarea3/Data' #Cambia a la ruta donde hayas almacenado Data
7 personPath = dataPath + '/' + personaName
8
9 if not os.path.exists(personPath):
10     print('Carpeta creada: ', personPath)
11     os.makedirs(personPath)
12
13 #cap = cv2.VideoCapture(0, cv2.CAP_DSHOW) #para capturar un video con la camara
14 cap = cv2.VideoCapture('C:/Users/willy/Desktop/tareasproyectos2025/imagenes para tareas/tarea3/presentadora.mp4')
15
16 faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
17 count = 0
18
19 while True:
20
21     ret, frame = cap.read()
22     if ret == False: break
23     frame = imutils.resize(frame, width=640) #tamaño del frame
24     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
25     auxFrame = frame.copy()
26
27     faces = faceClassif.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)
28
29     for (x,y,w,h) in faces:
30         cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
31         rostro = auxFrame[y:y+h,x:x+w]
32         rostro = cv2.resize(rostro, dsize=(150,150), interpolation=cv2.INTER_CUBIC)
33         cv2.imwrite(personPath + '/rostro-{}.jpg'.format(count), rostro)
34         count = count + 1
35         cv2.imshow('winname: ' + frame, frame)
36
37     k = cv2.waitKey(1)
38     if k == 27 or count >= 300: #numero de capturas
39         break
40
41 cap.release()
42 cv2.destroyAllWindows()
```

Figura 1: Captura de rostros p1

```
capturandorostros.py  entrenandoRF.py  ReconocimientoFacial.py
23 frame = imutils.resize(frame, width=640) #tamaño del frame
24 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
25 auxFrame = frame.copy()
26
27 faces = faceClassif.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)
28
29 for (x,y,w,h) in faces:
30     cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
31     rostro = auxFrame[y:y+h,x:x+w]
32     rostro = cv2.resize(rostro, dsize=(150,150), interpolation=cv2.INTER_CUBIC)
33     cv2.imwrite(personPath + '/rostro-{}.jpg'.format(count), rostro)
34     count = count + 1
35     cv2.imshow('winname: ' + frame, frame)
36
37 k = cv2.waitKey(1)
38 if k == 27 or count >= 300: #numero de capturas
39     break
40
41 cap.release()
42 cv2.destroyAllWindows()
```

Figura 2: Captura de rostros p2

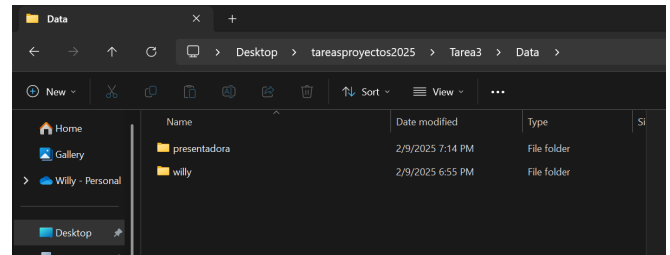


Figura 3: Carpetas de los datos obtenidos

III. ENTRENAMIENTO DE LOS MODELOS DE CAPTURA DE ROSTROS

```
capturandorostros.py  entrenandoRF.py  ReconocimientoFacial.py
1 import cv2
2 import os
3 import numpy as np
4
5 dataPath = 'C:/Users/willy/Desktop/tareasproyectos2025/Tarea3/Data' #Cambia a la ruta donde hayas almacenado Data
6 peopleList = os.listdir(dataPath)
7 print('Lista de personas: ', peopleList)
8
9 labels = []
10 facesData = []
11 label = 0
12
13 for nameDir in peopleList:
14     personPath = dataPath + '/' + nameDir
15     print(' Leyendo las imagenes')
16
17     for fileName in os.listdir(personPath):
18         print('Rostros: ', nameDir + '/' + fileName)
19         labels.append(label)
20         facesData.append(cv2.imread(personPath + '/' + fileName, 0))
21         #image = cv2.imread(personPath + '/' + fileName, 0)
22         #cv2.imshow('image', image)
23         #cv2.waitKey(10)
24         label = label + 1
25
```

Figura 4: Entrenador de reconocimiento facial p1

```
capturandorostros.py  entrenandoRF.py  ReconocimientoFacial.py
24 label = label + 1
25
26 #print('Labels= ', labels)
27 #print('Número de etiquetas 0: ', np.count_nonzero(np.array(labels)==0))
28 #print('Número de etiquetas 1: ', np.count_nonzero(np.array(labels)==1))
29
30 # Métodos para entrenar el reconocedor
31 face_recognizer = cv2.face.EigenFaceRecognizer_create()
32 #face_recognizer = cv2.face.FisherFaceRecognizer_create()
33 #face_recognizer = cv2.face.LBPHFaceRecognizer_create()
34
35 # Entrenando el reconocedor de rostros
36 print("Entrenando...")
37 face_recognizer.train(facesData, np.array(labels))
38
39 # Almacenando el modelo obtenido
40 face_recognizer.write('modeloEigenFace.xml')
41 #face_recognizer.write('modeloFisherFace.xml')
42 #face_recognizer.write('modeloLBPHFace.xml')
43 print("Modelo almacenado...")
```

Figura 5: Entrenador de reconocimiento facial P2

* e-mail: 3058096360301@ingenieria.usac.edu.gt

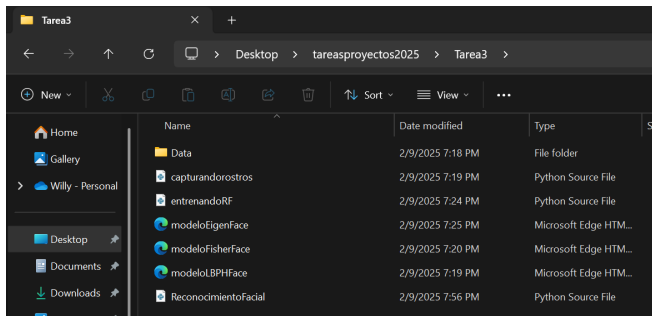


Figura 6: Los 3 modelos ya entrenados en la carpeta

IV. RECONOCIMIENTO FACIAL

Código de reconocimiento facial

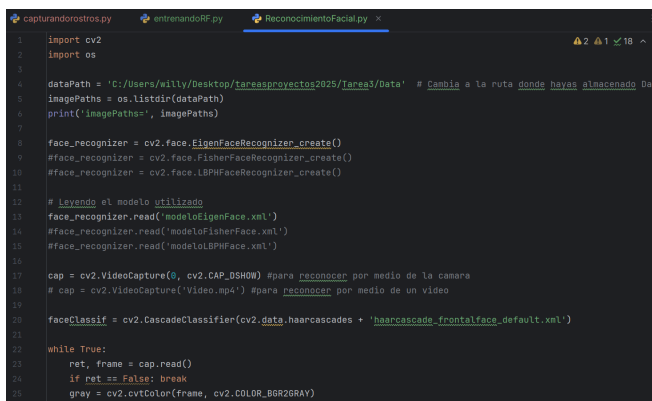


Figura 7: Reconocimiento P1

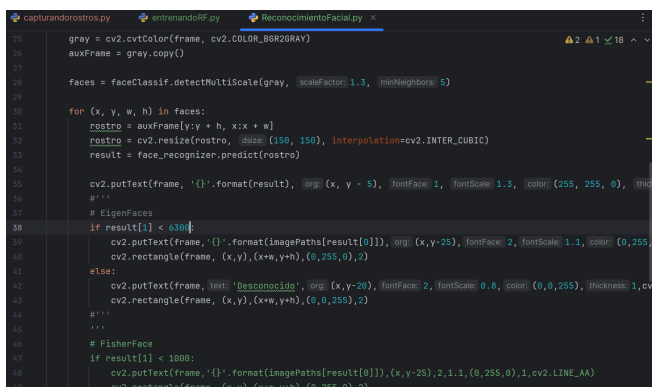


Figura 8: Reconocimiento P2

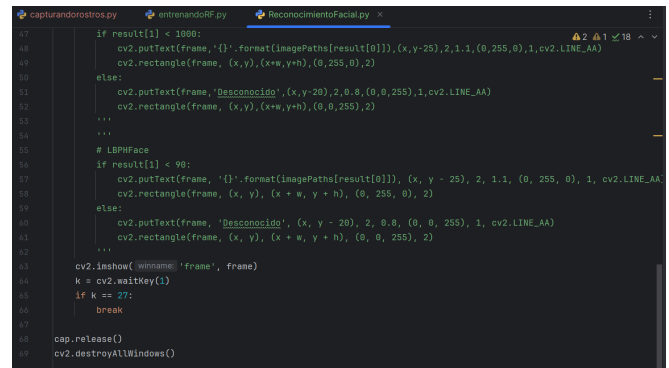


Figura 9: Reconocimiento P3

Vista del funcionamiento final

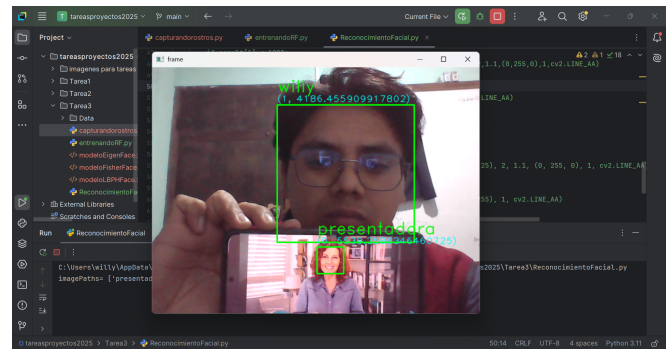


Figura 10: Detección final

V. REPOSITORIO

<https://github.com/WilliamSican/Tareas-Proyectos.git>